

Review

Not peer-reviewed version

A Novel Approach for Modelling and Developing Virtual Sensors Utilized in the Simulation of an Autonomous Vehicle

[István Barabás](#) , [Calin Iclodean](#) ^{*} , [Horia Beles](#) ^{*} , [Csaba Antonya](#) , [Andreia Molea](#) , [Florin Bogdan Scurt](#)

Posted Date: 19 March 2025

doi: 10.20944/preprints202503.1400.v1

Keywords: virtual sensors; ideal sensor; Hi-Fi sensor; RSI sensor; virtual vehicle model; autonomous driving; deep learning; machine learning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Review

A Novel Approach for Modelling and Developing Virtual Sensors Utilized in the Simulation of an Autonomous Vehicle

István Barabás ¹, Calin Iclodean ^{1,*}, Horea Beles ^{2,*}, Csaba Antonya ³, Andreia Molea ¹ and Florin Bogdan Scurt ²

¹ Department of Automotive Engineering and Transports, Technical University of Cluj-Napoca Romania, Muncii Bd. 103-105, 400114, Cluj-Napoca, Romania

² Department of Mechanical Engineering and Automotive, University of Oradea, Universitatii Str. 1, Oradea, 7, 410087, Bihor County, Romania

³ Department of Automotive and Transport Engineering, Transilvania University of Brasov, Eroilor Bd. 29, 500036, Brasov, Romania, Romania

* Correspondence: calin.iclodean@auto.utcluj.ro (C.I.); horia.beles@uoradea.ro (H.B.)

Abstract: A virtual model enables the study of reality in a virtual environment using a theoretical model, which is a digital image of a real model. The complexity of the virtual model must correspond to the reality of the evaluated system, becoming as complex as necessary nevertheless as simple as possible, allowing for computer simulation results to be validated by experimental measurements. The virtual model of the autonomous vehicle was created using the CarMaker software package, which was developed by the IPG Automotive company and is extensively used in both the international academic community and the automotive industry. The virtual model simulates the real-time operation of a vehicle's elementary systems at the system level and provides an open platform for the development of virtual test scenarios in the Autonomous Vehicles, ADAS, Powertrain, and Vehicle Dynamics application areas. This model included the following virtual sensors: slip angle sensor, inertial sensor, object sensor, free space sensor, traffic sign sensor, line sensor, road sensor, object by line sensor, camera sensor, global navigation sensor, radar sensor, lidar sensor and ultrasonic sensor. Virtual sensors can be classified based on how they generate responses: sensors that operate on parameters derived from measurement characteristics, sensors that operate on developed modeling methods, and sensors that operate on applications.

Keywords: virtual sensors; ideal sensor; Hi-Fi sensor; RSI sensor; virtual vehicle model; autonomous driving; deep learning; machine learning

1. Introduction

Sensors are electronic devices that generate electrical signals in response to various environmental stimuli [1]. Sensors' operating principles are determined by how information is recorded, and sensors can be classified as resistive, piezoelectric, capacitive, optical, magnetic, QTC (Quantum Tunnelling Composite), triboelectric effect, FET (Field-Effect Transistor), and so on [2]. Solid-state sensors are integrated devices that pre-process an analog or digital signal before delivering a sensory response that embedded systems can process.

A virtual sensor is a software emulation of a physical sensor that uses real data, mathematical models, fuzzy logic, neural networks, genetic algorithms, and ML (Machine Learning) and AI (Artificial Intelligence) models to estimate parameter values and anticipate scenarios [3]. ML algorithms comprise two types of mechanisms: interactive learning, in which the virtual sensor identifies relevant elements in data streams, and automatic teaching, in which the human factor proactively contributes relevant elements to training the learning process [4]. Virtual sensors are data-

driven prediction models that identify the physical characteristics of the system into which they are integrated, providing a viable substitute for real sensors while also being a cost-effective solution [5–9]. Another significant advantage of virtual sensors is the total reduction in vehicle mass, which includes both the mass of the physical sensors that have been replaced by virtual sensors and the mass of the wiring, insulators, and connectors that connect the physical sensors to the embedded systems, resulting in fuel and/or energy savings, in addition to a reduction in the volume of polluting emissions.

Virtual sensors have been developed by converting real sensor models into virtual models using real-world behavioral models, so that the generated output variables are represented with only input parameters relevant to the process being studied. Virtual sensor models are algorithms or mathematical models that estimate physical quantities using input data obtained from real sensors, statistical analysis, or input data generated by AI [10]. Virtual sensors are developed using software applications and do not require any additional or specific hardware components to function. According to [11], a virtual sensor is an only software sensor (no hardware components) that can generate signals autonomously by integrating data/signals obtained synchronously or asynchronously from physical sensors or even other virtual sensors. Virtual sensor models developed by [12] and implemented in the software architecture of real vehicles include, for example, the tire wear virtual sensor and the brake wear virtual sensor, which monitor wear, lifespan, and potential anomalies during the operation of some of the vehicle's systems and components (headlight levelling, tire pressure, tire temperature, e-motor temperature, brake temperature, suspension displacement, and so on) [13].

Virtual sensors offer a wide range of applications in vehicle active safety systems, including ABS (Antilock Braking System), AWD (All-Wheel Drive), ACC (Adaptive Cruise Control), and SAS (Smart Airbag System). The integration of virtual sensor parameters into the previously specified systems contributes to the management of the vehicle's optimal operating state in order to improve functional performance and reduce fuel/energy consumption, which is correlated with a reduction in pollutant emissions [14].

Implementing virtual sensors on real vehicles improves the accuracy of monitored data while also expanding coverage to locations where physical sensors are unavailable. When integrated into an embedded system, virtual sensors perform pre-processing, error correction, merging, and optimization of input data sets. Essentially, virtual sensors utilize an algorithm or mathematical apparatus to process input data and produce high-complexity output data sets that match specified requirements, as demonstrated by Hu et al. [15].

Unlike physical sensors, which must be added to precise positions inside a vehicle's structural architecture in order to function properly, virtual sensors depend on data sets collected from the vehicle's embedded systems. Based on this information, virtual sensors calculate specified parameters without the need for extra hardware. The development of a virtual sensor necessitates the implementation of a functional algorithm for the system under consideration, which is based on a statistical model that reliably anticipates the essential parameters being studied [16]. Because virtual sensors are made up of software components, firmware upgrades can be accomplished remotely via the OTA (Over The Air) approach, eliminating the need for physical interventions to remove and install these sensors.

Virtual sensors may improve data accuracy and resolution by merging information from numerous sources (other sensors, electronic control units, actuator feedback) using advanced data fusion and processing algorithms [17,18]. These sensors might be very simple or extremely sophisticated, depending on the activities and consequences they simulate: stimulus, electrical requirements, ambient environment, operational restrictions, and functional safety [19]. However, the performance of virtual sensors could decrease with time because of changes in nonlinear dynamics and the complexity of physical processes in the environment, as well as nonlinear interactions between input and output variables [6,20]. Virtual sensors increase the accessibility of data from physical sensors, facilitating collaboration at the sensor, equipment, and organizational

levels (allowing service providers to offer solutions based on the same hardware), allowing for more efficient use of the same hardware resources in interconnected systems, such as IoT (Internet of Things). Virtual sensors take data acquired by physical sensors and incorporate it into complicated software applications, where it is merged with other sources of information (databases) and processed by specialized algorithms to produce meaningful results [1,21].

Figure 1.1 illustrates three combinations of connected virtual and real sensors:

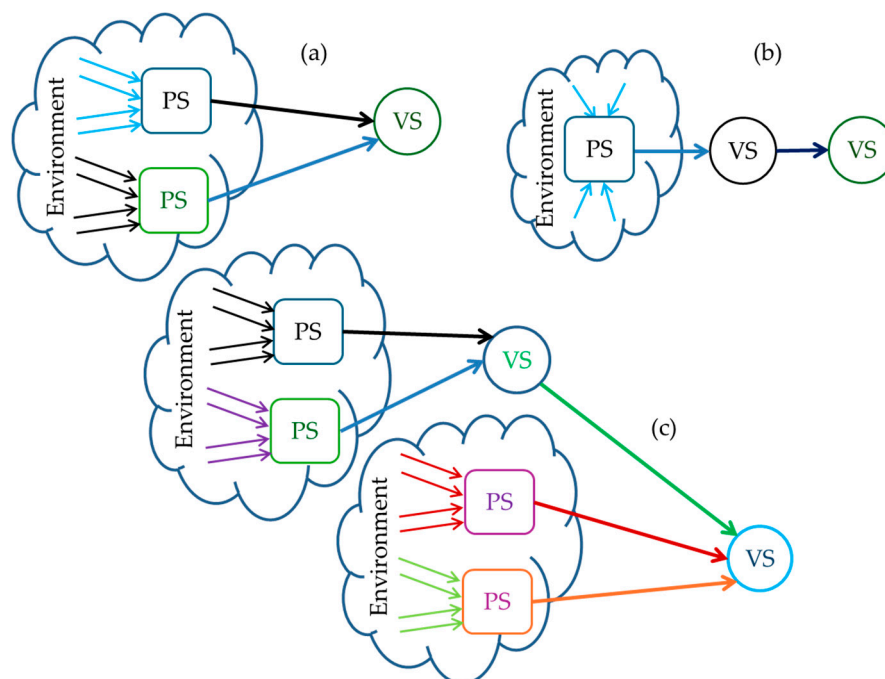


Figure 1.1. Different configurations of VS (Virtual Sensors) and PS (Physical Sensors).

- Virtual sensors depend only on the data from physical sensors. ESC (Electronic Stability Control) uses physical sensors like gyroscopes, accelerometers, wheel speed sensors, and virtual sensors to estimate yaw/slip angle, allowing the vehicle to maintain control in low-grip conditions or dangerous turns;
- Virtual sensors depend entirely on information from other virtual sensors. In the case of FCW (Forward Collision Warning), AEB (Automatic Emergency Braking), virtual sensor is used to predict the trajectory of the vehicle and evaluate the distance to other vehicles;
- Virtual sensors depend on data from both physical and virtual sensors. This configuration can be found in the DMS (Driver Monitoring System), which uses physical sensors like a video camera and/or pressure sensors in the steering wheel and/or seat, and virtual sensors like those for estimating the driver's level of attention and detecting the intention to leave the lane.

Finally, there is a requirement for using a suitable combination of physical and virtual sensors, in addition to maintaining functional algorithms for virtual sensors up to date [22].

Tactile Mobility [23] is a platform for monitoring, processing, and storing data from specific types of physical sensors that are installed in smart and interconnected vehicles in over 70 locations globally. This platform utilizes the data to create virtual sensors that, based on recorded scenarios, generate output parameters designed to improve the safety and performance of these vehicles [23]. The Tactile Mobility platform's solution incorporates a software program into the vehicle's built-in command and control systems, improving the operating regime by delivering information on road traffic, road conditions, tire grip and condition, vehicle mass, and so on.

Another platform that enables the use of virtual sensors in the automobile sector is the Compredict Virtual Sensor Platform [24], which calibrates, verifies, and implements these sensors on a wide range of real vehicle models. Thus, the Compredict platform can generate virtual models

based on Cloud-stored input data for the following virtual sensor categories: suspension travel, brake wear, brake temperature, wheel force transducer, vehicle mass, strain gauge, tire wear, tire pressure, tire temperature, LV (Low Voltage) battery health, HV (High Voltage) battery health, and battery anomaly.

The development of virtual sensors is accelerating; according to a market study conducted by Mordor Intelligence for the period 2025-2030 in [25], the virtual sensor market will be worth 1.37 billion USD in 2025 and 5.35 billion USD by 2030. The advancement of smart manufacturing technologies, specifically the digitization of industrial processes in addition to the digitalization and validation of real vehicle models, contributes to the development of the virtual sensor sector.

Virtual sensors are critical for modern vehicles in terms of improving autonomous driving capabilities, safety, and efficiency. Table 1.1 shows the sensors' progression and their implementation in a vehicle's constructive architecture from level 1 to level 5 automation (according to SAE J3016™) [26,27]. The sensors used in the equipment of autonomous vehicles are constrained by their physical dimensions, mass, the necessity to be positioned in less accessible sections of the vehicle's structural architecture, and the cost of these sensors [28]. It is evident that the number of sensors increases as automation levels rise with ultrasonic sensors and Lidar 2D/3D indicating the most significant numerical increases.

Table 1.1. Sensor's evolution depending on the driving automation (SAE J3016™).

| Level 1 | | Level 2 | | Level 3 | | Level 4 | | Level 5 (estimate) | |
|-------------------|-------|-------------------|-------|-------------------|-------|-------------------|-------|--------------------|-------|
| Model | Units | Model | Units | Model | Units | Model | Units | Model | Units |
| Ultrasonic | 4 | Ultrasonic | 8 | Ultrasonic | 8 | Ultrasonic | 8 | Ultrasonic | 10 |
| Radar Long Range | 1 | Radar Long Range | 1 | Radar Long Range | 2 | Radar Long Range | 2 | Radar Long Range | 2 |
| Radar Short Range | 2 | Radar Short Range | 4 | Radar Short Range | 4 | Radar Short Range | 4 | Radar Short Range | 4 |
| Camera mono | 1 | Camera mono | 4 | Camera mono | 2 | Camera mono | 3 | Camera mono | 3 |
| - | - | - | - | Camera stereo | 1 | Camera stereo | 1 | Camera stereo | 2 |
| - | - | - | - | Infra-Red | 1 | Infra-Red | 1 | Infra-Red | 2 |
| - | - | - | - | Lidar 2D/3D | 1 | Lidar 2D/3D | 4 | Lidar 2D/3D | 4 |
| - | - | - | - | Global Navigation | 1 | Global Navigation | 1 | Global Navigation | 1 |
| Total | 8 | Total | 17 | Total | 20 | Total | 24 | Total | 28 |
| 2012 | | 2016 | | 2018 | | 2020 | | estimated by 2030 | |

2. Classification of the Virtual Sensor

2.1. Virtual Sensor Model

According to research teams, the key real sensors frequently utilized in the constructive architecture of autonomous vehicles, sensors that form the basis for defining the virtual sensors used in modeling the virtual vehicle model, are as follows [29]:

1. Camera sensors generate synthetic data on the recognition and classification of objects in the area [30–32], in addition to the vehicle's positioning and orientation relative to close to objects and V2V (Vehicle-to-Vehicle) communication [33] based on the VLC (Visible Light Communication) principle [34]. The advantages of the camera sensor include the ability to provide data in real time, low latency in data acquisition and processing, adaptability to extreme lighting conditions (low lighting, bright lighting), accurate estimation of object position and orientation, and low production and implementation costs. The constraints of camera sensors include the need for direct view of surrounding objects, susceptibility to unexpected changes in lighting conditions, and the need for greater computer capacity due to the large quantities of data that are constantly generated;

2. Radar sensor generates data based on the reflection duration of radio waves ToF (Time of Flight) when detecting nearby target vehicles [35,36] and uses ML methods to estimate the current and future positions of nearby vehicles [37], respectively using DL (Deep Learning) methods to avoid collisions [38]. The benefits of radar sensors include the capacity to provide the location of target vehicles in real time, flexibility to severe weather conditions (rain, snow, fog), and low manufacturing and installation costs. The constraints of radar sensors include the requirement for increased computer capacity due to the massive volumes of data generated on a continuous basis, as well as a reliance on extra hardware systems and software;
3. Lidar sensors provide a system based on generating a point cloud through 2D and 3D laser scanning for real-time localization of static and dynamic objects in proximity [39,40] and applies the YOLO (You Only Look Once) picture segmentation technique [41]. The advantages of lidar sensors include the ability to localize static and moving objects in proximity precisely. The disadvantages of lidar sensors include the need for greater computer power due to the large quantity of data generated continuously, sensitivity to bad weather conditions (rain, snow, fog), and high manufacturing and implementation costs.

Sensor fusion is the process of combining sensor signals [42,43] using CNN (Convolutional Neural Network) neural networks, processing these signals with DL-type AI elements [44,45], detecting nearby objects in real time [46,47], and then making predictions about the evolution of these objects [48–51].

The virtual sensor models presented in this review were created, tested, and calibrated using the CarMaker simulation application from IPG Automotive, which is extensively used in the automotive industry for virtual vehicle model development at all stages. CarMaker is a platform that enables the development of any virtual test scenarios that are connected to other software applications [52].

Yeong et al. [53] classified physical and virtual sensors as smart or non-smart. Smart sensors are directly related to the IoT concept, and they are systems made up of interconnected devices that may collect and transport data remotely without the need for human involvement. A smart sensor is an IoT device that can condition and select incoming signals, process and interpret the generated data, and make decisions without the assistance of a separate processing unit [54].

Virtual sensors can be classed using the following criteria [55–57]:

1. Sensor fidelity could be classified as high, medium, or low-income;
2. Method for collecting information from the environment:
 - a) A deterministic strategy based on the simulation application's mathematical apparatus and involves the usage of a vast volume of input parameters to represent the ideal behavior and response of the virtual sensor as accurately as possible;
 - b) A statistical technique based on statistical distribution functions, which include the normal, binomial, Poisson, or exponential distribution;
 - c) The electromagnetic field propagation approach simulates electromagnetic wave propagation using Maxwell's equations.
3. The objective of using sensors is to develop a vehicle's operating mode based on observed metrics and to perform diagnostics using AI-based maintenance techniques to define the smart maintenance regime.

Virtual systems developed for simulation applications use additional virtual sensors that are intended to replace certain partial functionalities of the main real sensors to reduce the volume of input data, reduce computing power requirements, calibrate the main sensor, and provide an optimized output data stream [28].

CarMaker classifies virtual sensors into three types: Ideal Sensors, Hi-Fi (High Fidelity) Sensors, and RSI (Raw Signal Interface) Sensors (Figure 2.1). These virtual sensor models are intended to maximize the performance of the virtual vehicle model on which they are installed, as well as to assist the command-and-control system in developing and expanding the specific capabilities of each sensor to a higher class of sensors [58,59].

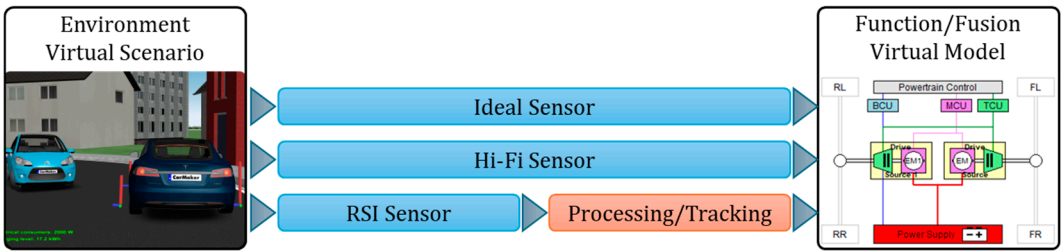


Figure 2.1. Virtual sensors classification.

The virtual model developed in CarMaker incorporates the following command and control systems for advanced assistance functions ADAS (Advanced Driver Assistance Systems): ACC, EBA (Emergency Brake Assist), LDW (Lane Departure Warning), LKA (Lane Keeping Assist), PA (Park Assist), ILA (Intelligent Light Assist), and TSA (Traffic Sign Assist). All these embedded systems evaluate and interpret data about the motor and/or vehicle's operating mode by combining various virtual sensor models [60].

Table 2.1 illustrates the functional properties of the main virtual sensor models, in addition to the range of applications and resources used by the system via the CPU (Central Processing Unit) and GPU (Graphics Processing Unit) [26,61].

Table 2.1. Virtual sensor models overview [62].

| Application | Model | Sensor | Description | Based |
|------------------|-------|-------------------|---|-------|
| Vehicle dynamics | Ideal | Slip Angle | Information about the vehicle's side slip angle | CPU |
| | Ideal | Inertial | Information about inertial body movements | CPU |
| | Ideal | Object | Detect objects defined as traffic objects | CPU |
| | Ideal | Free Space | Detect free and occupied spaces between objects defined as traffic objects | CPU |
| | Ideal | Traffic Sign | Detect traffic signs along the road | CPU |
| | Ideal | Line | Detect other road markings | CPU |
| | Ideal | Road | Provides road information as digital data | CPU |
| | Ideal | Collision | Detect contacts of the vehicle with other traffic objects | CPU |
| | Ideal | Object-by-Line | Detect traffic objects moving along selected road lanes | CPU |
| | Hi-Fi | Camera | Detect objects defined as traffic objects, traffic signs and traffic lights | CPU |
| | Hi-Fi | Global Navigation | Simulate GPS (Global Positioning System) satellites and their visibility for the vehicles | CPU |
| | Hi-Fi | Radar | Detect objects defined as traffic objects based on the SNR (Signal-to-Noise Ratio) | CPU |
| | RSI | Ultrasonic RSI | Simulate the propagation of the sound pressure waves through the virtual environment | GPU |
| | RSI | Lidar RSI | Lidar sensor simulating the propagation of laser light pulses through the virtual environment | GPU |
| | | | | |

2.1.1. Ideal Sensors

The role of ideal sensors in the CarMaker simulation program is to collect information from the simulation environment and transmit it to an embedded system. Ideal sensors are virtual entities developed using software that are independent of technology (Figure 2.2) that equip the virtual vehicle model with the following: Slip Angle, Inertial, Object, Free Space, Traffic Sign, Line, Road and Object by Line Sensor. Physical impacts that occur in the real environment in the case of a model integrated into the HiL (Hardware-in-the-Loop) system have no effect on these ideal sensors that are

integrated into the SiL (Software-in-the-Loop) model and don't generate information similar to a real sensor.

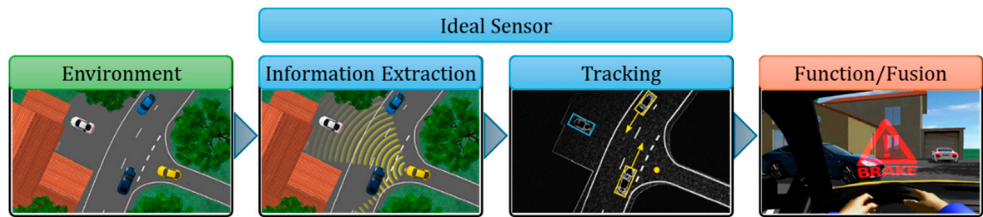


Figure 2.2. Ideal sensors.

2.1.2. Hi-Fi Sensors

Hi-Fi sensors filter the information supplied to the embedded system and provide data on the physical impacts that occur in the real environment, particularly the detection and classification of static and dynamic objects in the area. The virtual vehicle model is equipped with the following Hi-Fi sensors (Figure 2.3): Camera, Global Navigation and Radar sensors. Hi-Fi sensors have a role in reducing the impacts of false positives and false negatives that can occur in object perception and identification due to scenarios where part of the objects overlap, or environmental conditions prevent exact identification [63].

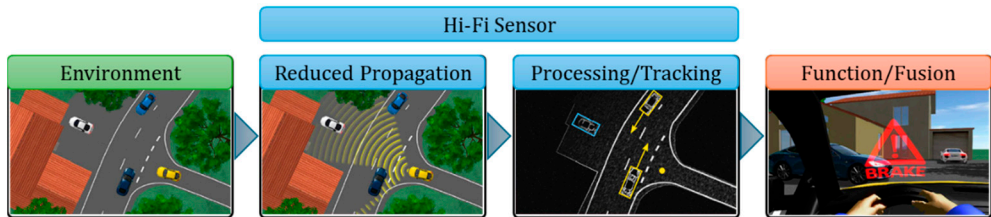


Figure 2.3. Hi-Fi sensors.

2.1.3. RSI Sensors

RSI sensors provide raw data and function identically to real sensors. The system filters, extracts, and interprets the data sent by the RSI sensors. Processing the information assigned by the RSI sensors necessitates high computational power, particularly for graphics processing provided by the GPU. There are RSI sensor types that do a post-processing of input in order to reduce the computing load on the embedded system. In the CarMaker simulation program, the RSI sensors identify objects in traffic and proximity, as well as all 3D surfaces in the surrounding environment. The utility IPGMovie, which is integrated with CarMaker, provides raw information for all these images. The virtual vehicle model is equipped with the following RSI sensors (Figure 2.4): ultrasonic RSI and lidar RSI [63].

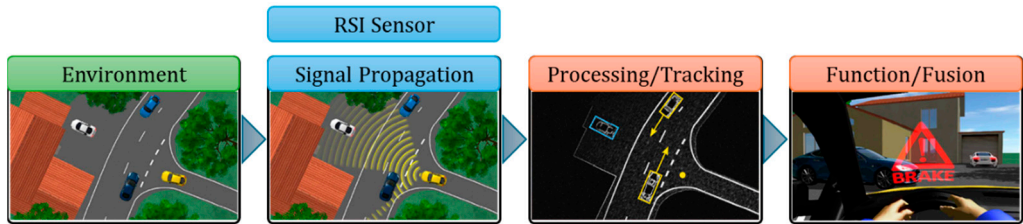


Figure 2.4. RSI sensors.

The use of RSI sensors in a virtual environment requires modeling the properties of the materials that compose the objects in their vicinity, namely relative electric permittivity for electromagnetic

waves and scattering effects. The direction and intensity of the field of waves reflected off 3D surfaces are significantly influenced by the material's characteristic properties [64].

RSI sensors process 3D images and offer real-time output for embedded systems, namely images and videos for the IPGMovie and/or MovieNX simulated scenario rendering system in CarMaker (Figure 2.5) [65].

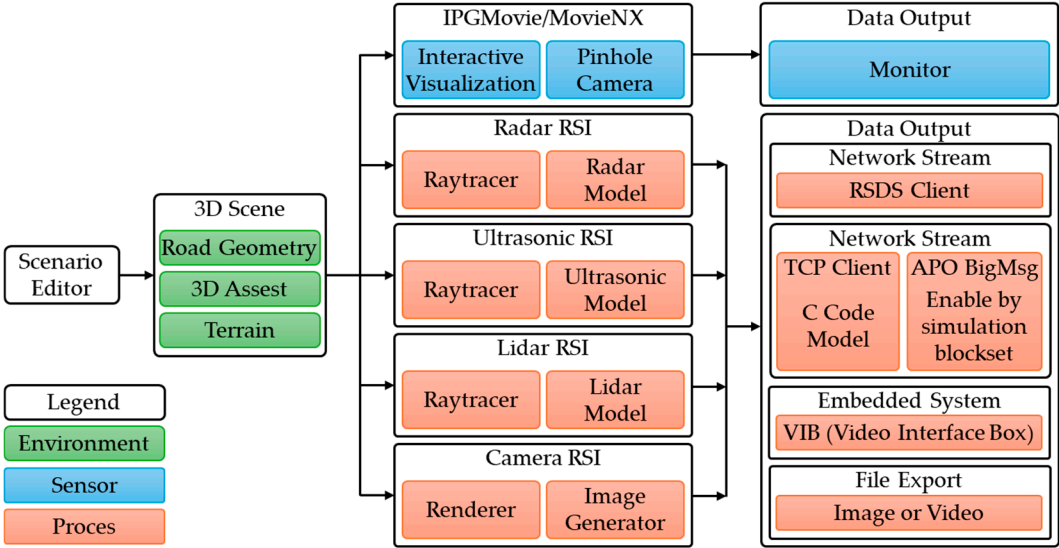


Figure 2.5. Interfaces and output format for RSI sensors.

2.2. Virtual Vehicle Model

A virtual vehicle model is a prototype that precisely replicates the characteristics of the elements and systems of a real model using mathematical and physical models. After validating the model, the simulation method enables the virtual vehicle to run in any user-defined scenario in a short period of time and at a low cost [19,66].

A virtual vehicle model, also referred to as a DTw (Digital Twin), is a digital image of a physical vehicle. Renard et al. [67] define DTw as an entity made up of a real model in a real space, a virtual model in a virtual space, and the data links that connect the real and virtual models. According to [68], the notion of DTw has gained popularity since 2017, with applications in a variety of academic and industry domains. DTw systems, thanks to bidirectional communication (Figure 2.6), allow the virtual model to be updated when the real model's state changes, and vice versa [68–71].

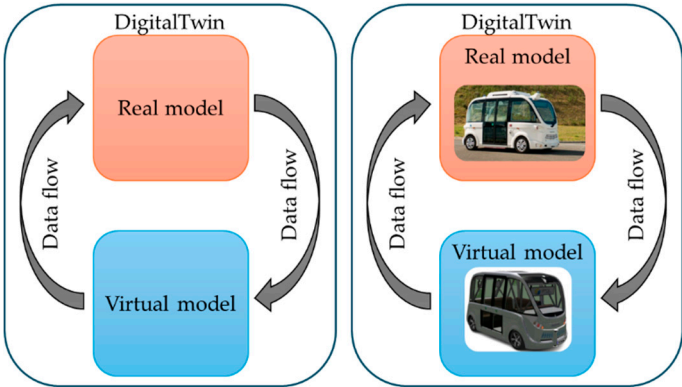


Figure 2.6. The basic structure of the DTw concept.

Navya Autonom® Shuttle is an autonomous shuttle bus vehicle designed for public passenger transportation and based on the architecture of a fully electric vehicle. Navya Autonom® Shuttle was introduced by the French start-up Navya in October 2015, and the main technical requirements

(navya.tech) were utilized for developing the virtual model in the CarMaker implementing virtual sensors (Figure 2.7). The results of the implemented scenarios were generated using the IPGControl application, which is a tool to generate diagrams based on computer simulation results [26,72–75].

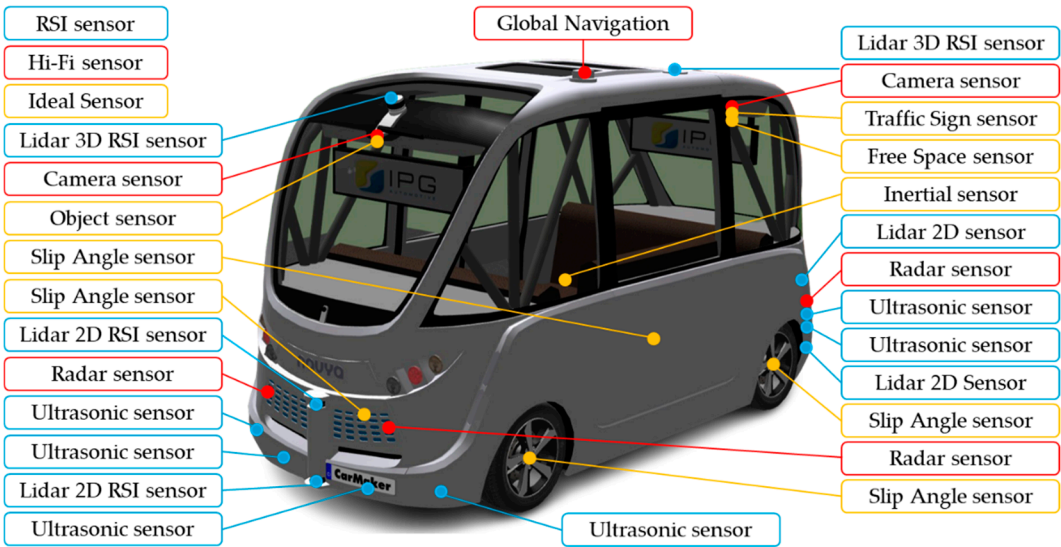


Figure 2.7. Virtual vehicle model for autonomous shuttle bus.

2.3. Virtual Environmental Model

The virtual environmental model consists of a virtual road and a virtual environment. The virtual road in which the virtual model of the autonomous vehicle travels was defined by digitizing the real route in Lyon, France using geographical coordinates (latitude, longitude, and altitude) extracted from Google Earth (Figure 2.8) [26,76,77]. The digitized route was converted to SRTM (Shuttle Radar Topography Mission) coordinates [77] using the GPSPRUNE application [78]. The route with the altitude profile was loaded into CarMaker's IPGRoad utility [79], which defined the following parameters in addition to the geographical coordinates (latitude, longitude, altitude): dimensions (length, width), connection angle, curvature, inclination, speed limit, and friction coefficient.

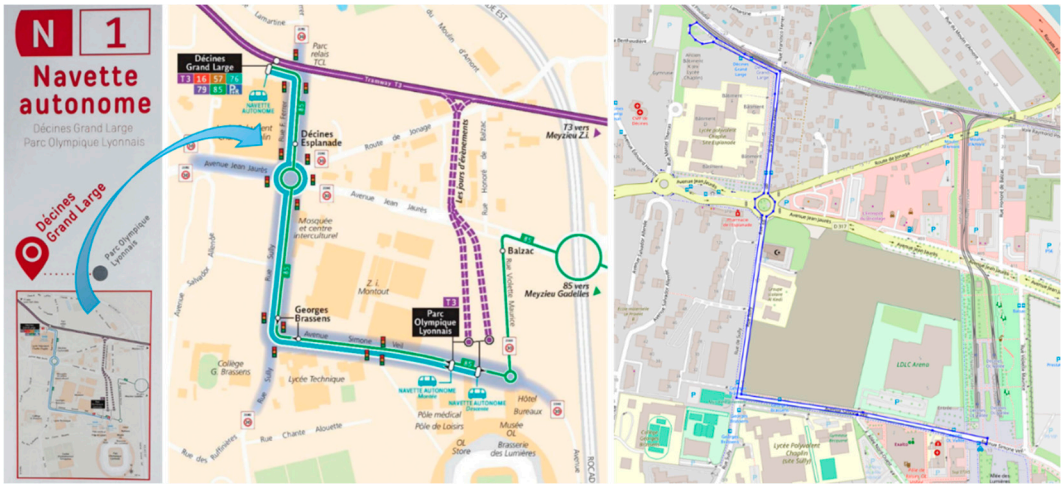


Figure 2.8. Virtual road vs. real road (TCL Lyon vs. GPSPRUNE - photo author (C.I.)).

The virtual environment for the computer simulations was created using the CarMaker application's Environment utility, which allowed the following atmospheric conditions to be defined: reference temperature, air density, air pressure, air humidity, cloud model, cloud intensity, fog, visibility, rain rate, wind velocity, and wind angle [61].

The autonomous driving system operating algorithm in CarMaker's Vehicle Control section defined virtual driver behavior by addressing different driving styles corresponding to a human driver's reaction speed and performance criteria under optimal energy consumption conditions [80].

Virtual traffic was defined using data collected on the flow of vehicles and pedestrians on the digitized route, traffic rules, peak hours and traffic congestion, and intelligent transportation systems (in the roundabout on the route, a traffic light is connected to autonomous vehicle via V2I (Vehicle-to-Infrastructure) technology).

3. Characteristics of the Virtual Sensor

3.1. Characteristics of Ideal Sensor

3.1.1. Slip Angle Sensor

The slip angle sensor monitors the lateral slip angle between the steering wheel angle and the vehicle's direction of motion. The slip angle sensor is located near the vehicle's steering wheel [26,61,65].

The yaw angle is helpful in active safety systems because it controls cornering stability, prevents vehicle rollovers, and avoids lane departure. Controlling the yaw angle is required because a big yaw angle reduces the tires' capacity to create lateral forces and greatly impairs the effectiveness of the vehicle control system. Beside yaw angle, also yaw rate are required variables for vehicle stability management [81,82].

The Pacejka model [83], a lateral force model, describes the complexity of the interaction between the tire and the road surface during dynamic maneuvers that are specific to autonomous vehicles. The Pacejka model is a semi-empirical mathematical model that represents the behavior of forces and moments created by a tire in contact with the road surface. It additionally provides a nonlinear representation of lateral forces, accounting for both large slip angles and normal forces. The Pacejka model (Figure 3.1) is commonly used in vehicle simulation and control, particularly for the development of advanced support systems and autonomous vehicles. The Pacejka model includes the following broad shape [82–84]:

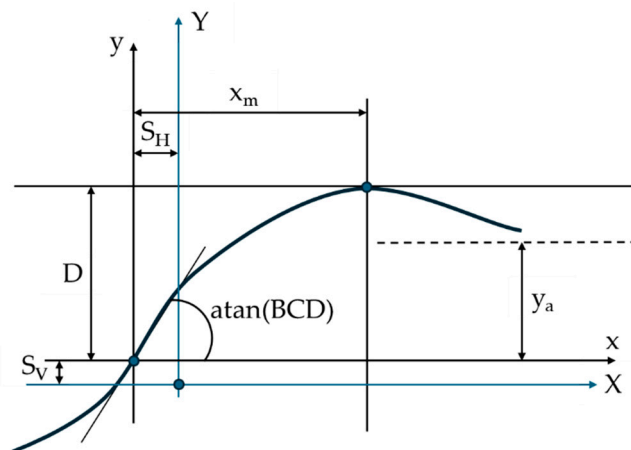


Figure 3.1. The Pacejka model.

$$y(x) = D \cdot \sin \{C \cdot \arctan[(1 - E)Bx - E \cdot \arctan(Bx)]\}, \quad (1)$$

with:

$$Y(X) = y(x) + S_v, \quad (2)$$

$$x = X + S_H, \quad (3)$$

where $y(x)$ represents F_x , F_y , M_z , B is the stiffness factor, C is the shape factor, D is the peak value, E is the curvature factor, S_H is the horizontal displacement, and S_V is the vertical displacement.

The Pacejka model generates a curve that passes through the origin ($x=y=0$), reaches a maximum value, and then tends to a horizontal asymptote. To create a more accurate representation of tire behavior, the Pacejka model allows the curve's position to be adjusted by inserting two translations, (S_H) and (S_V). These translations allow for the adjustment of potential asymmetries in the experimental data, resulting in a better fit between the model and reality. For specific coefficients B , C , D , and E , the curve exhibits anti-symmetry with respect to the origin. The coefficient D specifies the peak value, whereas the product B , C , D determines the curve's beginning slope. The coefficient C changes the operating range limits in the general formula, determining the shape of the curve. The stiffness factor is defined as the coefficient B , which is calculated using the inclination reported to the origin. The coefficient E is introduced to manage the curvature peak while also controlling its horizontal position. The shape factor C can be derived from the peak height and horizontal asymptote using the following formula [82,83,85]:

$$C = 1 \pm \left(1 - \frac{2}{\pi} \arcsin \frac{y_a}{D}\right). \quad (4)$$

Curvature factor E is computed from B and C for the position x_m of the peak value using the following equation (if $C > 1$) [86]:

$$E = \frac{Bx_m - \tan \left[\frac{\pi}{(2C)} \right]}{Bx_m - \arctan(Bx_m)}. \quad (5)$$

The specific force is expressed as follows in both the longitudinal and transverse directions [87]:

$$\Gamma = D \cdot \sin \{ C \cdot \arctg [B \cdot s - E \cdot (B \cdot s - \arctg(B \cdot s))] \}. \quad (6)$$

(c) The lateral dynamics of motor vehicles are described using the three-degree-of-freedom mathematical model, as shown in Figure 3.2. [81,88].

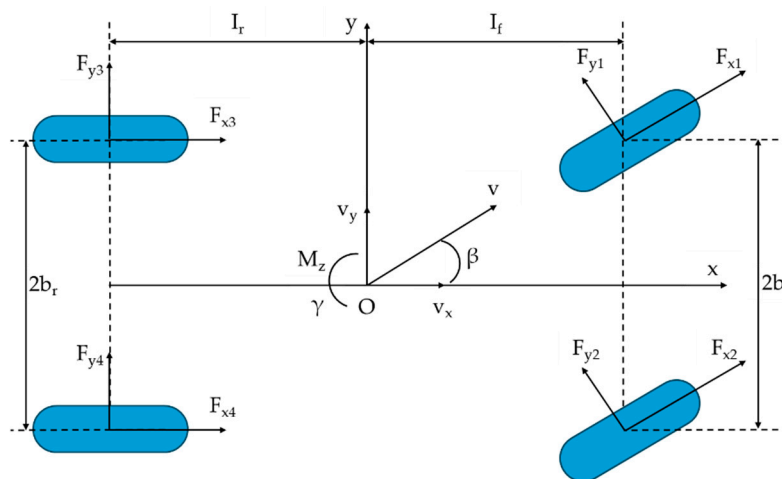


Figure 3.2. The three-degree-of-freedom model.

The three-degree-of-freedom mathematical model accounts for pitch, yaw, and roll motions [88–90] expresses the mathematical model equation as follows:

$$\dot{v}_x = \gamma v_y + \frac{1}{m} [(F_{x1} + F_{x2}) \cos \delta - (F_{y1} + F_{y2}) \sin \delta + F_{x3} + F_{x4}], \quad (7)$$

$$\dot{V}_y = -\gamma v_x + \frac{1}{m} [(F_{x1} + F_{x2}) \sin \delta + (F_{y1} + F_{y2}) \cos \delta + F_{y3} + F_{y4}], \quad (8)$$

$$\dot{\gamma} = \frac{1}{I_z} [(F_{x1} + F_{x2}) l_f \sin \delta - (F_{y3} + F_{y4}) l_r + (F_{y1} + F_{y2}) l_f \cos \delta + (F_{y1} - F_{y2}) b_f \sin \delta - (F_{x1} - F_{x2}) b_f \cos \delta - (F_{x3} - F_{x4}) b_r]. \quad (9)$$

The equations for a vehicle is as follows: v_x is the longitudinal speed, v_y is the lateral speed, γ is the yaw rate, m is the mass, δ is the steering angle of the front wheels, I_z is the moment of inertia, F_{xj} and F_{yj} ($j=1,2,3,4$) are the longitudinal and lateral forces of the tires, l_f and l_r are the distances from the vehicle's center of gravity to the front and rear axles, and b_f and b_r are half of the tread of the front and rear axles, respectively.

To run the simulations in CarMaker, the virtual vehicle model dynamics library was utilized, which calculates the wheel slip angle using the force and moment equilibrium equations as the derivative of the virtual vehicle's slip and yaw ratio [91,92]:

$$\dot{\beta} = -\gamma + \frac{2C_f}{mv_x} \left(\delta - \beta - \frac{l_f \gamma}{v_x} \right) + \frac{2C_r}{mv_x} \left(-\beta + \frac{l_r \gamma}{v_x} \right), \quad (10)$$

$$\dot{\gamma} = \frac{2C_f l_f}{I_z} \left(\delta - \beta - \frac{l_f \gamma}{v_x} \right) + \frac{2C_r l_r}{I_z} \left(\beta + \frac{l_r \gamma}{v_x} \right). \quad (11)$$

For β vehicle slip angle and γ vehicle yaw rate, C_f vehicle front tire cornering stiffness, m vehicle mass and v_x longitudinal velocity. Vehicle parameters include δ front-steering angle, β body slip angle, l_f distance from center of gravity to front axle, and v_x longitudinal velocity, C_r vehicle rear tire cornering stiffness, l_r distance between the center of gravity and the rear axle, I_z moment of inertia for the vehicle's yaw axis.

3.1.2. Inertial Sensor

The inertial sensor determines the vehicle's position, speed, and acceleration. It is based on a three-axis accelerometer (x, y, z) that outputs information about the vehicle's translational speed, translational acceleration, and rotational acceleration. The inertial sensor is located in the center of the vehicle [26,61,65].

Inertial sensors, coupled with the slip angle sensor, comprise the inertial positioning system IMU (Inertial Measurement Unit), which also incorporates a three-axis accelerometer. Inertial measurements include linear acceleration, angular velocity, and angular acceleration. The dynamic parameters (roll, pitch, and yaw rate) measured by the inertial sensor are incorporated in the following relationships [93]:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} 0 & -\dot{\phi} & -\dot{\theta} & -\dot{\psi} \\ \dot{\phi} & 0 & \dot{\psi} & -\dot{\theta} \\ \dot{\theta} & -\dot{\psi} & 0 & \dot{\phi} \\ \dot{\psi} & \dot{\theta} & -\dot{\phi} & 0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \quad (12)$$

In three-dimensional rotation computations, the quaternion $[q_0 q_1 q_2 q_3]^T = Q$ represents the ϕ roll, θ pitch, and ψ yaw rate. Based on the Taylor series results, the quaternion solution from instant k to moment $k+1$ is:

$$Q(k+1) = \left(I \left(1 - \frac{\Delta \theta^2}{8} \right) + \frac{\Delta \theta}{2} \right) \cdot Q(k), \quad (13)$$

$$\Delta \theta = \int_k^{k+1} \begin{bmatrix} 0 & -\dot{\phi} & -\dot{\theta} & -\dot{\psi} \\ \dot{\phi} & 0 & \dot{\psi} & -\dot{\theta} \\ \dot{\theta} & -\dot{\psi} & 0 & \dot{\phi} \\ \dot{\psi} & \dot{\theta} & -\dot{\phi} & 0 \end{bmatrix} dt \approx \begin{bmatrix} 0 & -\Delta \theta_\phi & -\Delta \theta_\theta & -\Delta \theta_\psi \\ \Delta \theta_\phi & 0 & \Delta \theta_\psi & \Delta \theta_\theta \\ \Delta \theta_\theta & -\Delta \theta_\psi & 0 & \Delta \theta_\phi \\ \Delta \theta_\psi & \Delta \theta_\theta & -\Delta \theta_\phi & 0 \end{bmatrix}. \quad (14)$$

For dt sampling time, $\Delta \theta = [\Delta \theta_\phi \Delta \theta_\theta \Delta \theta_\psi]^T$ angular increments include $\Delta \theta_\phi$, $\Delta \theta_\theta$, $\Delta \theta_\psi$ roll, pitch, and yaw angles.

Inertial sensors IMU and VDM (Vehicle Dynamic Model) can be used to explain the performance and safety of autonomous vehicles by properly determining slip angle and altitude. VDM is composed of two basic parts: a delayed estimator and a predictor (Figure 3.3) [94]. The delayed estimator includes two types of estimators: those based on IMU data and those based on vehicle dynamic models. IMU estimators directly estimate variables like speed and attitude, but VDM estimators use mathematical models based on measurements from other sensors, like wheel speed sensors.

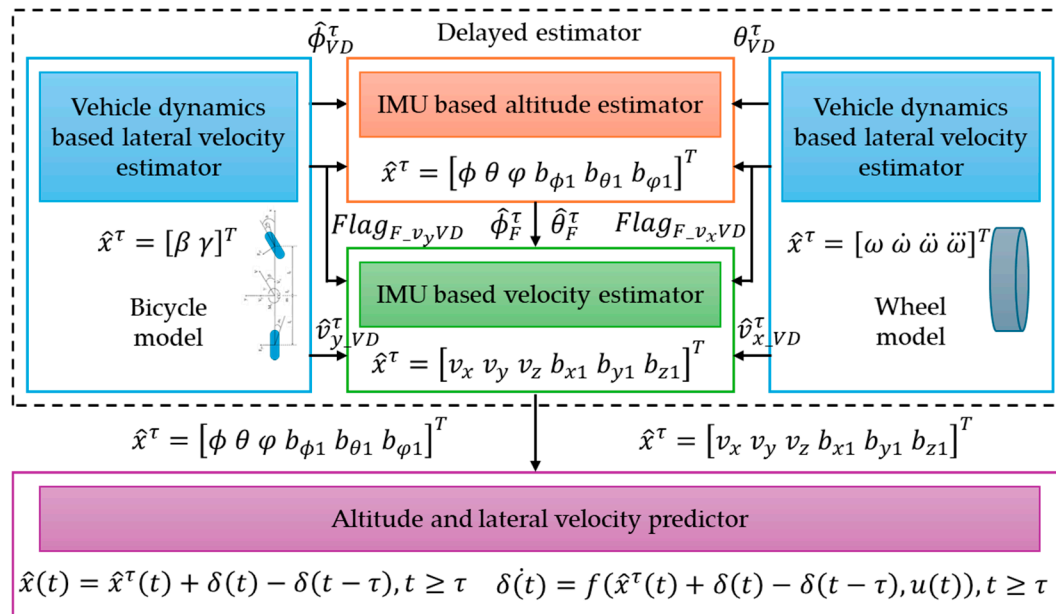


Figure 3.3. Vehicle Dynamic Model.

Under normal driving conditions, data from VDM estimators are utilized to correct errors that may develop in IMU estimates, using a Kalman filter to predict roll and pitch angles [94].

The vehicle dynamics models provide a more in-depth understanding of the vehicle's overall behavior, which enhances estimate accuracy. Dynamic models may become less accurate under harsh driving situations, such as hard breaking or quick bends. In such cases, the IMU estimators are temporarily separated from the VDM estimators to prevent error propagation. To synchronize the input from the two estimators, a delay is added to the VDM-based estimate. To predict the system's current state, the predictor uses delayed estimates as well as information about the vehicle's controls. This enables a more precise estimate of the slip angle and attitude, even in dynamic conditions [95].

3.1.3. Object Sensor

Scanning the environment is an important stage in an autonomous vehicle since it offers information that allows it to perceive and understand its surroundings. This first stage is critical for obtaining a thorough and up-to-date image of the traffic situation, allowing the vehicle to make informed decisions and travel safely. To perform this comprehensive and multidimensional scanning, autonomous vehicles employ a complex network of sensors, each with particular expertise in providing information. The Object Sensor is a software component that simulates the operation of a real video camera in an autonomous vehicle. It detects objects in traffic and estimates their distance, with the nearest object considered the target. The data is utilized to make decisions in autonomous driving. Object Sensor employs image processing techniques and AI to identify and track objects such as vehicles, pedestrians, and bicycles, and the distance between them can be calculated using trigonometric calculations [96].

Bewley et al. in [97] describe a simple and efficient technique to real-time multi-object tracking. This method focuses on the quick connection of items observed in consecutive frames, highlighting the need for accurate detection for quality tracking. Using conventional methods such as the Kalman

filter and the Hungarian algorithm, the method achieves accuracy comparable to complex systems. This method's simplicity and efficiency make it ideal for real-time applications including pedestrian tracking in autonomous driving systems.

To predict the position of an object in the frame, a linear motion model with constant velocity is utilized, which is unaffected by other objects or camera movement.

$$X = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T, \quad (15)$$

where u and v represent the horizontal and vertical location of the target center in pixels, and s and r indicate the scale (area) and aspect ratio of the target's bounding box, respectively [97,98].

The DPM (Deformable Part Model) algorithm, which detects surrounding vehicles and pedestrians, is an advanced object recognition method. DPM looks for and evaluates the characteristics of target objects in the images collected by the video camera, where objects are defined as a collection of parts organized in a deformable configuration. Each part represents the local attributes of an object's appearance, whereas links between pairs of parts defined the deformable configuration. The DPM algorithm learns to identify objects of interest from the background by comparing positive and negative examples. Therefore, the algorithm develops a collection of filters that respond to the object's specific properties, such as edges, corners, and textures. A filter is built using a rectangular template defined by a matrix of dimensional vectors d . The response of the filter F at a position (x, y) on a feature map G is the "local dot product" of the filter and a sub-window of the feature map at (x, y) [99,100].

$$\sum_{x', y'} F[x', y'] \cdot G[x + x', y + y']. \quad (16)$$

A feature pyramid is used to specify an object's size and position in a picture. A pyramid is a series of feature maps with varying resolutions. In practice, pyramids are created by computing a conventional image pyramid, smoothing, and repeatedly down-sampling. A feature map is then generated from each point in the picture pyramid, as depicted in Figure 3.4 [101,102].

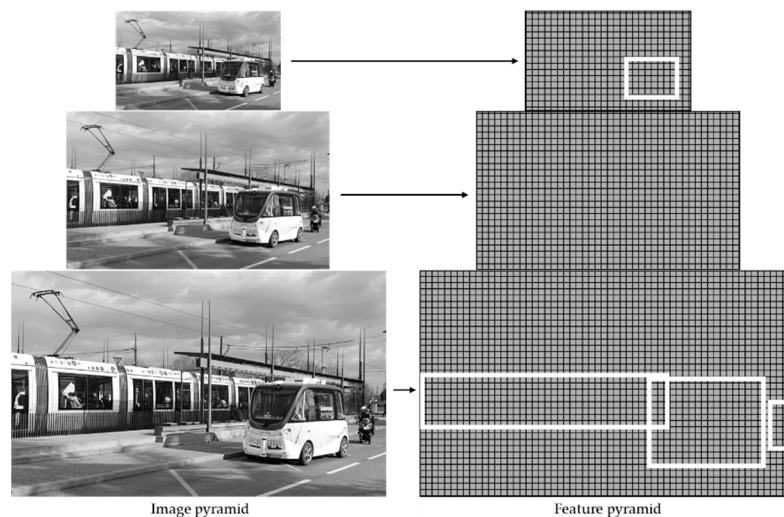


Figure 3.4. A feature pyramid getting an instantiation of a person model within it. The part filters are positioned at double the spatial resolution of the root location.

In the CarMaker virtual environment, the Object sensor is a virtual sensor that detects objects in traffic and calculates their distance. The most appropriate target object is determined by its proximity to the sensor [26,61]. The Object sensor transmits information to the ACC system, which is responsible for automatically adapting the acceleration in the vehicle's movement so that it maintains a consistent speed in comparison to the vehicles in front of it [66].

Figure 3.5 illustrates the structure of the ACC system, which includes the object sensor. A cluster of two sensors collects information about detected objects, one using an antenna for objects at a distance (long range antenna) and one using an antenna for objects in close proximity to the vehicle (short range antenna) and generates a list of intercepted objects. Raw sensor data is used to locate and track detected objects (object sensor detected objects list), with tracking algorithms performing data fusion to ensure control over the vehicle's cruising speed while maintaining a safe distance from the relevant detected objects [65].

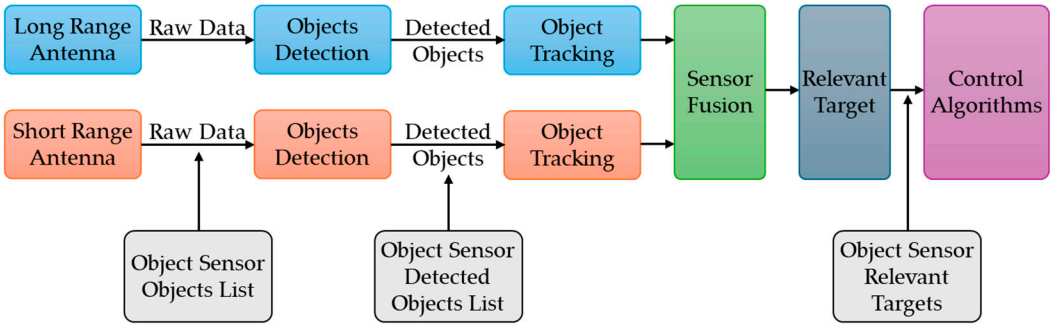


Figure 3.5. Object sensor integrated into the ACC system.

The object list in the object sensor interface transmits data about the objects in the CarMaker application database that are detected by the sensor's detection field. These objects are identified based on the following characteristics: object ID (Identifier), object dimensions, object orientation, distance to the object, and object speed, which correspond to the angle of incidence between the sensor beam and the object.

The algorithm for detecting the closest object is based on scanning all objects within range and selecting the relevant target objects for the object sensor by identifying the trajectory and movement lane, respectively by calculating the movement speed and distance to the object (Figure 3.6) [65].

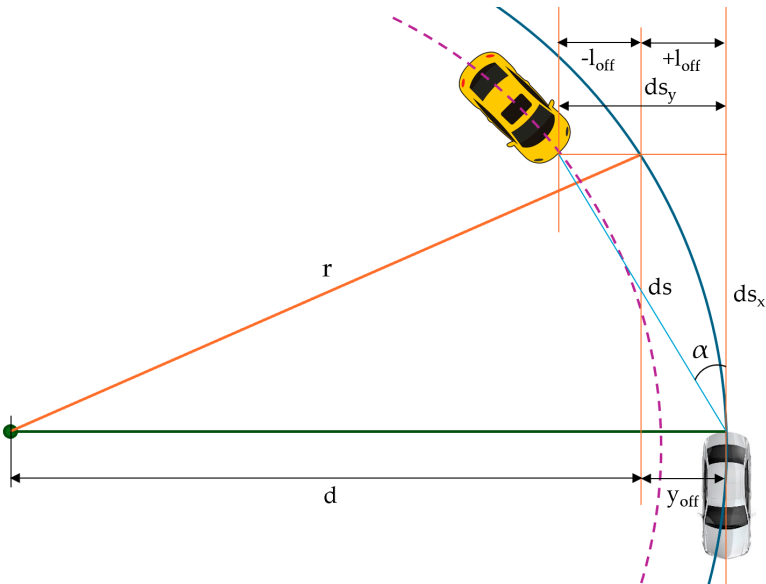


Figure 3.6. Object trajectory identification algorithm.

In Figure 3.6 ds represents the projected distance to target vehicle, α angle of the target vehicle in sensor frame, ds_x component in X direction of projected distance, ds_y component in Y direction of projected distance in sensor frame, r turning radius, y_{off} imaginary vehicle offset in sensor frame at

the target position, l_{off} half of vehicle lane width, a_y lateral acceleration of vehicle, and v the vehicle speed. Target selection algorithms can configure sensor response in two modes:

- Nearest object, this is the closest visible object that is considered a relevant target;
- Nearest object in the path, this is the closest object within an interval of an estimated vehicle trajectory.

The following relationships characterize components in object sensors:

$$ds_x = ds \cdot \cos(\alpha), \quad (17)$$

$$Ds_y = ds \cdot \sin(\alpha). \quad (18)$$

The following relationships describe the vehicle's offset in the sensor's perception of the position of adjacent objects, and thus the limits of the vehicle's trajectory.

$$y_{off} = r - d = (r - \sqrt{r^2 - ds_x^2}) \cdot \text{sign}(a_y), \quad (19)$$

$$L_{off} = \frac{\text{vehicle}_{lane}}{2} + \text{lane}_{offset}, \quad (20)$$

$$(y_{off} - l_{off} < ds_y) \cap (ds_y < y_{off} + l_{off}). \quad (21)$$

The CarMaker Object Sensor module generates an item list for each configured sensor, with quantities for each traffic object in the sensor's view. After scanning the environment, the list of objects that can be recognized around the virtual vehicle model will include the following markers [65]:

- Object ID a name or a code used to identify an object;
- Path direction (reference and closest point);
- Relative distance and velocity (between the reference and the nearest positions);
- Relative orientation in the axle x-y-z (the reference point);
- Sensor frame's x-y-z distances (between the reference and the nearest point);
- Sensor frame's x-y-z velocity (between the reference and the nearest point);
- Flag object has been identified (in the sensor viewing area);
- Flag object has been identified (in the observation area);
- Incidence angles between the sensor beam and the object being detected from proximity;
- Width, height, length of the object, and height above the ground.

3.1.4. Free Space Sensor

The free space sensor is a software component that detects free space around the vehicle and uses this information to plan routes and avoid obstacles. The sensor creates an accurate map of the surrounding environment by combining data from several sensors, including cameras, lidar and radar. The sensor data is analyzed to identify barriers and compute their distances. The free space sensor information is utilized to build safe and efficient trajectories, making it an important component in assuring the safety and efficiency of autonomous vehicles.

The free space sensor is an extension of the object sensor, with the sensor beam separated into horizontal and vertical segments. Each segment determines the closest point of the observed objects in traffic, in addition to the vehicle's angle with respect to these objects and their respective speeds. The sensor scans the environment and determines the free and occupied area in the vicinity, guiding the vehicle's progress through it [26,62].

The free space sensor plus is an extension of the sensor that detects all around items using a separate computational approach based on 3D image analysis. Three-dimensional image analysis of objects in proximity uses two filtering methods: the closest point on the object surface (nearest) and the strongest point on the object surface (strongest). The nearest finds the point on the object surface (represented by a pixel in the generated image) that is closest to the sensor position. The strongest

determines the point on the object's surface with the least reflection angle relative to the incident vector for each pixel within the sensor's detecting range [65].

Open space identification methods use either 2D models (camera pictures) or 3D models (point clouds obtained from lidar sensors or stereo cameras). 2D approaches segment the road using low-level cues including color and texture, but they may fail if the road textures are inconsistent. However, 3D algorithms may have difficulties recognizing modest height variations, such as those between the road and the sidewalk. The hybrid method combines the benefits of 2D, and 3D modeling to overcome the limits of each methodology and provide more robust open space identification [103–105]. Therefore, the use of 3D information obtained from the input 3D point cloud renders road plane recognition more efficient. The road plane is determined in a parametric space, which includes the plane distance from the center of the room as well as the angle between the plane normal and the room main axis. The plane in the parametric space is described by the following equation [104]:

$$Z \cdot \sin\theta - y \cdot \cos\theta = d \cdot \cos\theta. \quad (22)$$

This estimate suggests that the camera height and direction to the road remain constant (Figure 3.7). Encoders mounted on the vehicle wheel are used to correct camera and point cloud translations obtained by simultaneous localization and mapping to a metric space. The distance scale obtained from the encoders is utilized to adjust the camera translation scale, which then automatically scales the point cloud to metric space. This is required since it assists in parametrically altering the plane given a known initialization of d , dependent on the camera height [105].

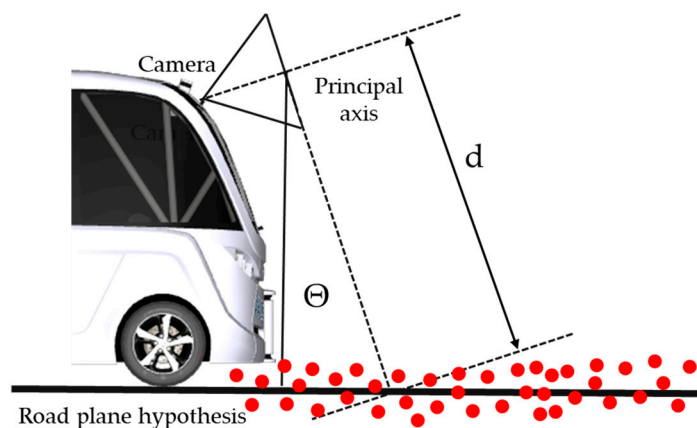


Figure 3.7. Detecting the road plane from a point cloud.

3.1.5. Traffic Sign Sensor

The traffic sign sensor recognizes pre-selected signs within its defined range and sight region. The sensor determines if the detected traffic sign is directly facing the vehicle and ranks the detected signs in ascending order of distance from it before identifying and classifying them. The information supplied to embedded command and control systems about detected traffic signs aids in the comprehension and interpretation of the traffic rules and conduct specified by it [26,62]. The traffic sign sensor is an ideal camera, equipped with an algorithm for recognizing traffic signs and the colors of traffic lights within its field of vision, which uses the identifiers assigned to traffic signs and traffic lights in close proximity to locate, classify, and interpret their operation [106].

HD (High-Definition) maps can provide insight into the environment in which road traffic evolves. HD maps give precise information about the environment where static road traffic occurs, including details about roads and obstacles, across a radius of more than 200 meters, even in locations with no direct vision (in bends). This information, when combined with data from cameras and lidar sensors, allows for exact vehicle localization. Currently, creating HD maps requires a professional technique that includes specialized topography and mapping with a MMS (Mobile Mapping System).

These maps are created by integrating road pictures with 3D data extracted from point clouds. Zhang et al. in [107] developed an architecture for real-time HD map production that includes an industrial camera, a cutting-edge GNSS (Global Navigation Satellite System)/IMU, and a high-performance computer platform on board the vehicle (Figure 3.8). The semantic SLAM (Simultaneous Localization And Mapping) technology, which is based on an enhanced BiSeNet (Bilateral Segmentation Network), is used to extract semantic data from point clouds, including information about the traffic situation.

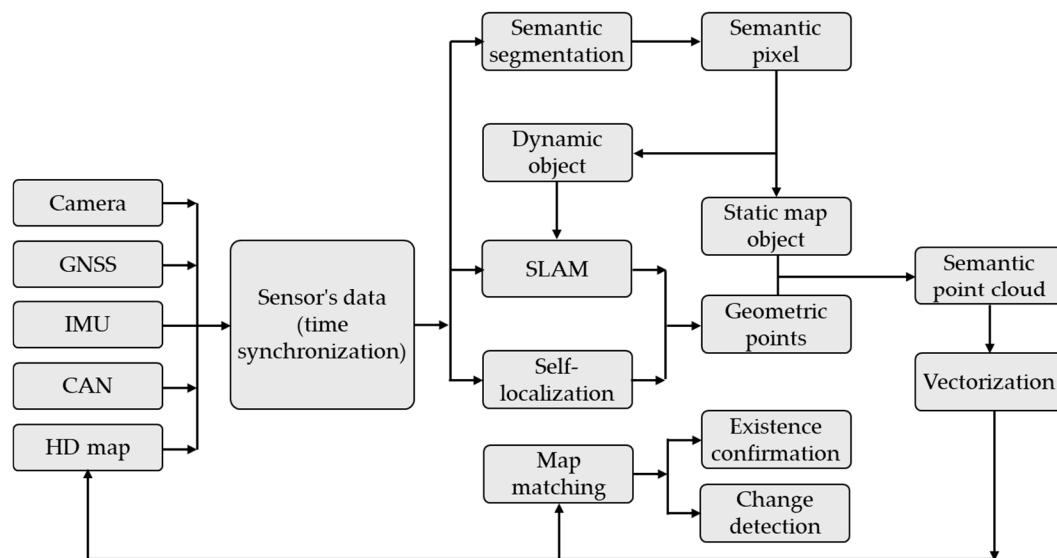


Figure 3.8. The architecture of the HD map.

SLAM is expensive and takes a long time to create HD maps. Furthermore, HD maps may contain inconsistencies between recorded road signs and real-time local modifications. In addition to supporting drivers, intelligent object identification systems can help with roadside maintenance, including road signs, lane markings, and guardrails. Road sign recognition systems, for example, can check for potential anomalies using autonomous vehicles, as human inspection of a complete road infrastructure is difficult.

As a result, the traffic sign recognition technique is an essential component of both autonomous driving systems and road management systems. Methods for recognizing road signs have centered on researching key aspects such as color and shape [108]. These feature-based approaches are particularly sensitive over long distances and in poor light. The usage of object detection models based on CNN has recently become popular in road sign recognition systems. DL based object identification algorithms, such as YOLO models, aid in the correct recognition of road signs in traffic. YOLO model-based studies for road sign identification have demonstrated great performance when using publicly available reference datasets [59,109,110]. Figure 3.9 shows a YOLO model-based arrangement for traffic sign identification [59].

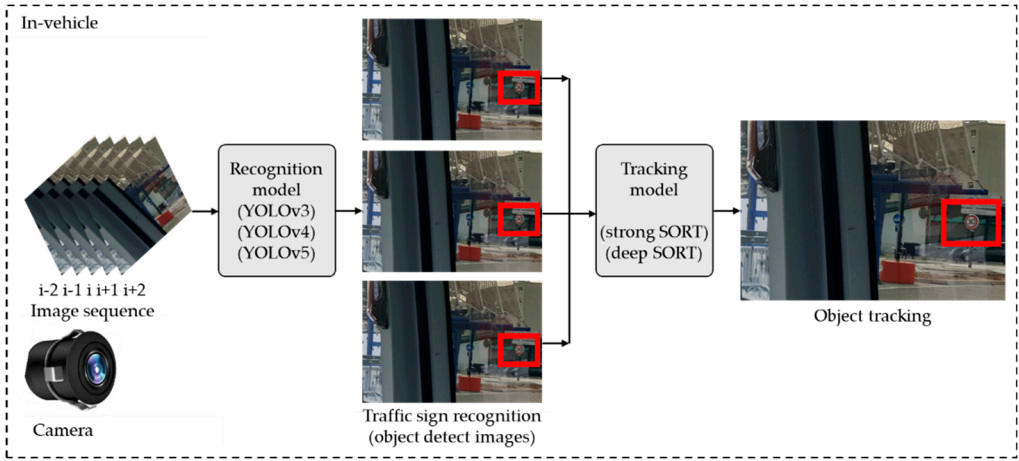


Figure 3.9. Traffic sign recognition using YOLO models.

3.1.6. Line Sensor

The line sensor detects road markings on the roadway in the direction of driving and organizes them into left and right lines, recording information about each. The recorded data includes the lateral distance to the given reference points, the type of lines, their width and height, and the color code. To detect roadway lines, the sensor generates a sequence of planes based on seven points (five vertical planes and three inclined horizontal planes) deployed along the travel direction. Road markings on the roadway are identified by recording the intersection points of the vertical and horizontal planes created by the sensor and the lines on the road surface [65]

The tread detection algorithm, illustrated in Figure 3.10, starts by capturing an image of the road. The image is processed in multiple phases. Initially, the image is transformed from RGB to grayscale, then noise is reduced using a symmetric 2D Gaussian filter. The image is then processed to improve contrast in order to recognize road markings. A Sobel operator can be used to detect edges. Finally, a binary image highlighting the tread markings is generated (Figure 3.11) [111].

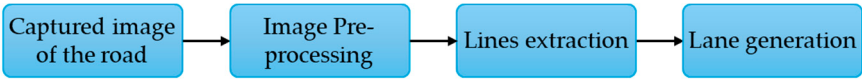


Figure 3.10. The tread detection algorithm.

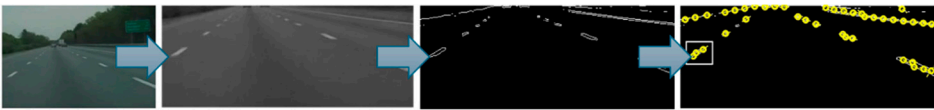


Figure 3.11. Vector direction of markings on a road.

If a combined laser scanning and video camera system is used for lane marking identification, the method is based on a top-hat transformation, which is preprocessed using the vertical-gradient Prewitt operator to generate a binary image. The binary image is then processed with a PPHT (Progressive Probabilistic Hough Transform) to detect lane markers. Figure 3.12 displays the lane marking detection algorithm [91].

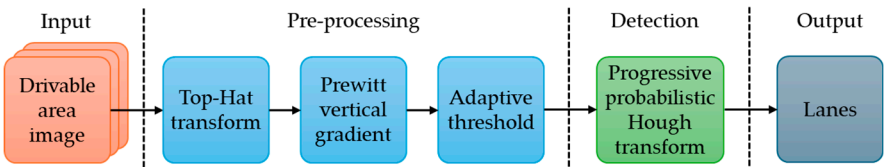


Figure 3.12. Lane marking detection algorithm.

A Top-Hat transformation is used to increase image contrast while reducing interference from non-tread marks. The Top-Hat transformation is a mathematical model that recovers small-sized bright objects and details from pictures using the following relationship:

$$h = f - (f \circ b), \tag{23}$$

where f is the source image, h is the final image after performing the Top-Hat transformation, and “ \circ ” is an operator that is realized by the Top-Hat transformation and controlled by the choice of the structuring element (b). The size of the structural element b determines how many elements are extracted from the image [91].

The Prewitt vertical gradient operator uses the following mathematical model to detect vertical edges in an image [112]:

$$G(x,y) = |I(x + 1,y - 1) + I(x + 1,y) + I(x + 1,y + 1) - I(x - 1,y - 1) - I(x - 1,y) - I(x - 1,y + 1)|, \tag{24}$$

where $I(x,y)$ represents the pixel intensity at the coordinates (x,y) .

PPHT reduces the amount of computing required to correctly detect the markings, utilizing a linear mathematical model:

$$y = mx + b, \tag{25}$$

where m is the slope of the line and b is the intercept at the origin.

3.1.7. Road Sensor

The road sensor determines the following roadway attributes up to a specified distance: road curvature, road marker attributes (speed limits), longitudinal and lateral slope, and the distance and angle of deviation when driving. This data is sent to embedded command and control systems to perform the following functions: LK (Lane Keeping), LDW, AD (Autonomous Driving), SD (Sign Detection), EM (Energy Management), FC (Fuel Consumption), WLD (Wheel Lifting Detection), and PT (Powertrain) (Table 3.1) [65].

The road sensor is located in the middle of the vehicle’s front wheels. The sensor’s technique for detecting roadway features is based on projecting a point along the route reference line. The deviation represents the lateral offset of the projected point from the route reference line.

Table 3.1. Road sensor functions.

| Function | LK | LDW | AD | SD | EM | FC | WLD | PT |
|----------------------------|----|-----|----|----|----|----|-----|----|
| Road curvature | | | | | | | | |
| Longitudinal/lateral slope | | | | | | | | |
| Deviation angle/distance | | | | | | | | |
| Lane information | | | | | | | | |
| Road point position | | | | | | | | |
| Road marker attributes | | | | | | | | |

3.1.8. Object by Line Sensor

Object by line sensors detect and transmit information on traffic lanes and traffic objects passing through them by assigning POI (Point Of Interest) points to each of these objects.

The route’s number of lanes is divided into LaneScope sections, which include the road axis (LandScope Center), the left (LandScope Left), and the right (LandScope Right).

LaneScopes are used to structure information about objects and traffic lanes (Figure 3.13), with S_{min} , S_{max} , t_{min} , and t_{max} defining the offsets of the extremities of traffic objects along the route [65].

The LaneScope center is calculated using the POI position. The LaneScope center course is considered to be the basis for establishing the left and right LaneScope. The center lane course is determined by track lists generated by the computation method. The algorithm determines whether

there is a lane where the POI is located. It then creates two lanes (successor/predecessor) beginning with this lane. If no acceptable successor/predecessor route is found, the relevant lane for the object by line sensor is terminated.

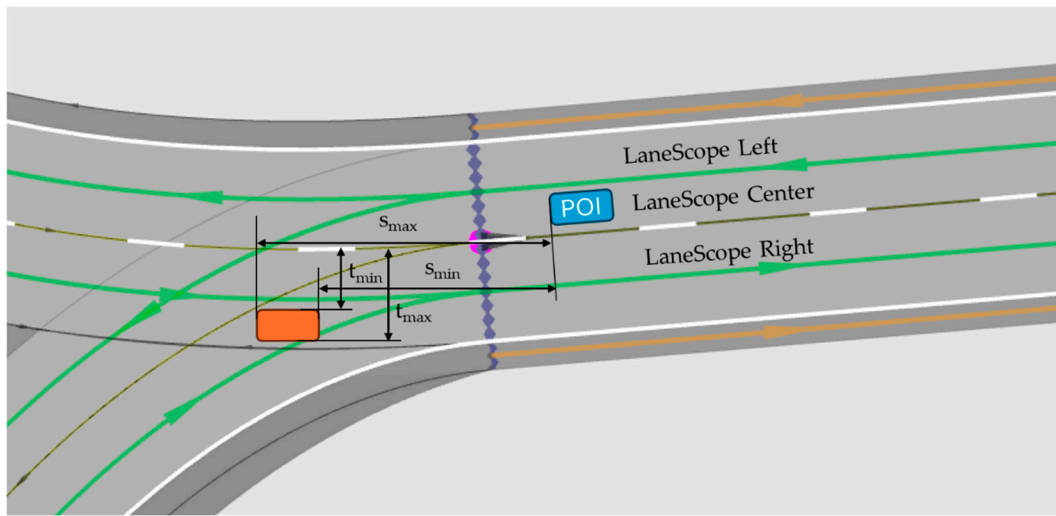


Figure 3.13. Characteristics of object by line sensor.

Lindenmaier et al. in [113] used a GNN (Global Nearest Neighbor) approach to assign detected objects to the reference object category in an interval d_c , respectively $d_{c,lat}$, resulting in a minimal global association distance. The Mahalanobis distance relation $d_{MH}(x_i, z_j)$ between the reference object x_i and the detection object z_j defines the distance matrix $D_{ij} \in \mathbb{R}^{+(N,M)}$, which serves as the foundation of the GNN method.

$$D_{ij} = d_{MH}(x_i, z_j) = (x_i^{pos} - z_j^{pos})^T \cdot S^{-1} \cdot (x_i^{pos} - z_j^{pos}), \quad (26)$$

where N and M are the number of reference and detected objects, and x_i^{pos} and z_j^{pos} are the position vectors of the respective objects. The matrix of covariance S is calculated using the cutoff distance ratio $d_r = d_c/d_{c,lat}$ as follows [114]:

$$S = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & d_r^2 \end{bmatrix} \cdot \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}^T, \quad (27)$$

where α represents the angle of the road path at the longitudinal distance of the detected object x_i .

3.2. Characteristics of Hi-Fi Sensor

3.2.1. Camera Sensor

The virtual vehicle model is equipped with a sensor camera that is positioned front-to-back to provide a circular image of its surroundings. The sensor camera's objective is to constantly monitor the movement of the virtual vehicle model along the selected route's traffic lane in order to detect and classify static and dynamic objects in the surroundings, as well as recognize traffic signs and traffic light colors. Monocular cameras take 2D images without determining the distances to the monitored objects, whereas stereo cameras may determine the distance by measuring the difference between two images taken from various perspectives [72,115].

Environmental conditions may reduce visibility and affect the identification of nearby objects. The influence of these elements is determined using the following relationship:

$$A_{Env} = \max\left(1.0 - \frac{\text{RainRate}}{\text{RainRate}_{\max}}, 0\right) \cdot \min\left(\frac{\text{VisRangeInFog}}{\text{VisRange}_{\max}}, 1\right), \quad (28)$$

where RainRate is the rain's intensity, and VisRangeInFog is the direct visibility under foggy conditions.

The camera sensor's maximum error in measuring the distance ($\text{dist}_{\text{Err,max}}$) to nearby objects will be computed using the following formula:

$$\text{dist}_{\text{Err,max}} = \frac{\text{dist}^2}{f \cdot b} \cdot d_{\text{Err}}, \quad (29)$$

where dist is the actual distance to the monitored object, f is the focal length, and b is the baseline. d_{Err} represents the disparity error.

The x and y coordinates of the image of the item acquired by the camera are determined using the following formulas [116,117]:

$$x = \frac{h \cdot (x'^2 + f^2 - y' \cdot f \cdot \tan \alpha) \cdot \sin \beta}{\sqrt{x'^2 + f^2} \cdot (f \cdot \tan \alpha + y')}, \quad (30)$$

$$Y = \frac{h \cdot (x'^2 + f^2 - y' \cdot f \cdot \tan \alpha) \cdot \cos \beta}{\sqrt{x'^2 + f^2} \cdot (f \cdot \tan \alpha + y')}, \quad (31)$$

where h is the camera's height from the ground, f is its focal length, and α is the angle between the camera's optical axis and the horizontal line to the target. Breaking down the relationships results [118]:

$$\begin{aligned} \tan^{-1} \frac{f \cdot \tan \alpha}{\sqrt{x'^2 + f^2}} + \tan^{-1} \frac{y'}{\sqrt{x'^2 + f^2}} &= \tan^{-1} \frac{\frac{f \cdot \tan \alpha}{\sqrt{x'^2 + f^2}} + \frac{y'}{\sqrt{x'^2 + f^2}}}{1 - \frac{f \cdot \tan \alpha}{\sqrt{x'^2 + f^2}} \cdot \frac{y'}{\sqrt{x'^2 + f^2}}} \\ &= \tan^{-1} \frac{\sqrt{x'^2 + f^2} \cdot (f \cdot \tan \alpha + y')}{x'^2 + f^2 - y' \cdot f \cdot \tan \alpha} = \tan^{-1} \frac{h}{d}, \end{aligned} \quad (32)$$

$$\frac{\sqrt{x'^2 + f^2} \cdot (f \cdot \tan \alpha + y')}{x'^2 + f^2 - y' \cdot f \cdot \tan \alpha} = \frac{h}{d}, \quad (33)$$

$$d = \frac{h(x'^2 + f^2 - y' \cdot f \cdot \tan \alpha)}{\sqrt{x'^2 + f^2} \cdot (f \cdot \tan \alpha + y')}. \quad (34)$$

3.2.2. Global Navigation Sensor

The global navigation sensor locates a vehicle by using the positions of at least four geostationary GPS satellites. The sensor determines real-time positioning in geographic coordinates using information about the transmission time of the extrapolation signal emitted by the satellites and received by the vehicle's receiver (x, y, z, t) [119].

CarMaker can represent any position of the virtual vehicle model in the global road architecture as a geographic point on the Earth's surface. This uses the GCS (Geographic Coordinate System) coordinate system, which consists of latitude, longitude, and altitude.

The origin of the road frame on the Earth's surface is determined using the FlatEarth projection method using GCS reference points (Figure 3.14). This projection method ignores the Earth curvature around the GCS reference point when calculating the relative position $\overrightarrow{\text{RefPo}}$ of point P in the road frame. The elevation value h at point P is calculated as follows [65]:

$$h = h_R + \overrightarrow{\text{RefPo}}[z]_0. \quad (35)$$

The relative latitude $\Delta\phi$ can be calculated as follows:

$$\Delta\Phi = \text{asin}\left(\frac{\Delta y}{R_N(\phi_R) + h}\right) \approx \frac{\Delta y}{R_N(\phi_R) + h} = \frac{\overrightarrow{\text{RefPo}}[y]_0}{R_N(\phi_R) + h}, \quad (36)$$

where $R_N(\phi_R)$ is the radius of the Earth ellipsoid in the north, which is determined by the latitude of the GCS reference point. The latitude ϕ of point P is transformed into:

$$\Phi = \phi_R + \Delta\phi. \quad (37)$$

Similarly, the longitude at point P can be determined using the following formula:

$$\lambda = \lambda_R + \frac{\overrightarrow{\text{RefPo}}[zx]_0}{(R_E(\phi_R) \cdot \cos(\phi_R))}, \quad (38)$$

where $R_E(\phi_R)$ is the radius of the Earth ellipsoid in the east, which is determined by the latitude of the GCS reference point. The factor $\cos(\phi_R)$ considers decreasing radius with increasing latitude.

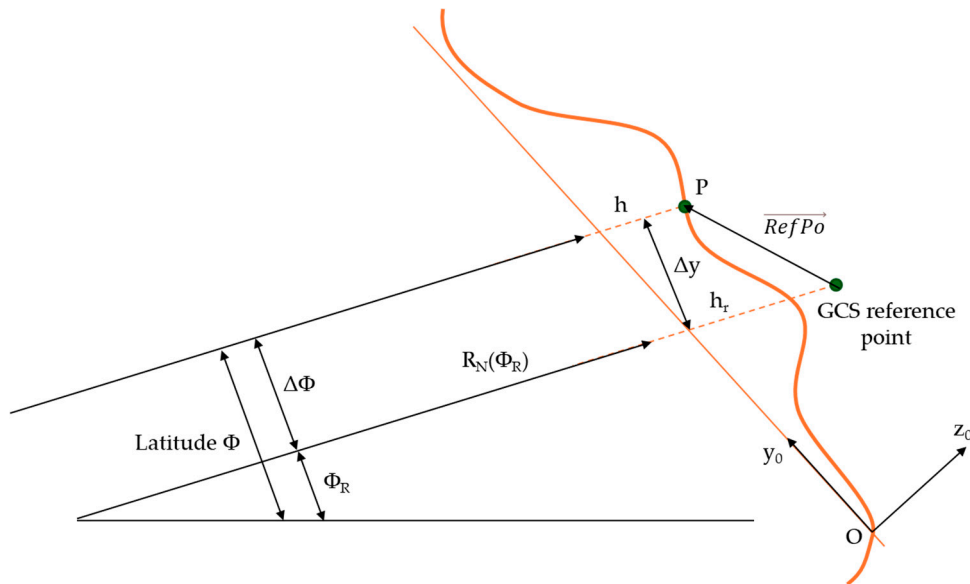


Figure 3.14. Calculation of the latitude for the global navigation sensor.

D-GNSS (Differential-Global Navigation Satellite System) with differential RTK (Real-Time Kinematic) correction reduces vehicle positioning errors by applying a differential correction relative to the coordinates of a reference base station, which then uses signals transmitted by satellites. The delivery of D-GNSS differential corrections with RTK from the reference base station to the vehicle takes place via mobile data connections [26].

CarMaker's virtual car model employs Cartesian coordinates (x, y, z), geocentric coordinates ECEF (Earth Centered Earth Fixed), and ellipsoidal coordinates (ϕ latitude, λ longitude, h elevation) for geostationary satellite positioning. GDOP (Geometric Dilution of Precision) refers to the accuracy of the computed position of the vehicle receiver in reference to the geostationary position of visible satellites.

3.2.3. Radar Sensor

The radar sensor detects static and dynamic objects on the virtual vehicle route using the SNR signal. The sensor detects things using cellular units and accounts for the effects of overlapping traffic items. The detected items are identified using particular RCS (Radar Cross Section) cross-section maps that take into account the angle of incidence and the signal reflected by the traffic objects. Depending on the signal-to-noise ratio, surrounding objects will be recognized, removed, or regarded as false negatives [26].

RCS is defined as the cross-section of a detected object that intercepts the most amount of power transmitted by the radar sensor. It is determined by the following parameters: the size and shape of the object, the antenna orientation angle, the frequency and polarization of the radar waves, the object's electromagnetic properties, and the surface structure. Radar sensor simulation is complex due to the dispersion effect of radar waves in virtual settings, and it is achieved using physically

interpretable characteristics such as the distance to nearby objects, the movement speeds of traffic items, and their angular locations.

Elster et al. in [120] used the DVM (Double Validation Metric) methodology to validate radar sensor data virtually. DVM uses the reliability and repeatability of radar sensor readings to effectively quantify deviations between distributions for various types of detected objects. The measurement data (M1, M2) is preprocessed and filtered, and the number of points resulting from the measurements is determined for each data set using the EDF (Empirical cumulative Distribution Function) methodology.

The deviation of mean values d_{bias} for measurement data M1 is calculated and compared to measurement data M2. The shape deviation of the distribution function d_{CAVM} is calculated to highlight the difference in signal scattering between M1 and M2 (Figure 3.15) [121].

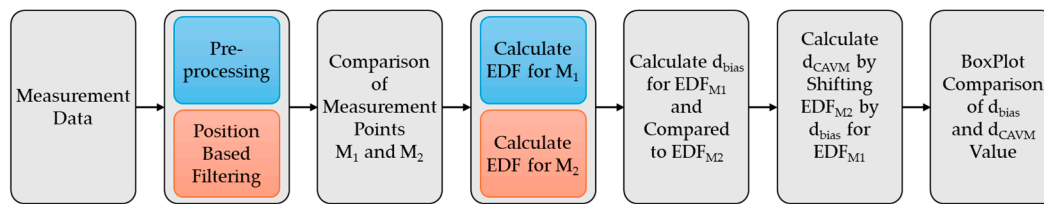


Figure 3.15. DVM methodology.

The radar equation defines the physical relationships connected to the features of the radar sensor, resulting in the received signal power $Q_s(r_s, u_s, \phi_s)$ [65,122,123].

$$|Q_{SP}(r_s, u_s, \phi_s)|^2 = \frac{P_{RX}}{P_{TX}} = \frac{\lambda_c^2 \cdot G_{TX}(u_s, \phi_s) \cdot G_{RX}(u_s, \phi_s) \cdot \sigma_s}{(4\pi)^3 \cdot r_s^4} \cdot F(r_s, u_s, \phi_s)^4, \quad (39)$$

where Q_s is the impulse response for s point scatterer, positioned at a r_s distance at angular position, for u_s elevation and ϕ_s azimuth, λ_c wavelength of the carrier frequency, G_{TX} gain of the transmitter antenna, G_{RX} gain of the receiver antenna, and σ_s reflection coefficient of the point scatterer.

A gain map is used to highlight antenna features, which is characterized by a unidirectional gain factor parameterized by azimuth and elevation in the scanning direction. The following equations describe field strength and antenna gain [124]:

$$f(\theta, \phi) = \frac{\sin\left(\pi \frac{a}{\lambda} \sin \theta \cos \phi\right) \sin\left(\pi \frac{b}{\lambda} \sin \theta \sin \phi\right)}{(\pi v_y)} \frac{1}{(\pi v_z)}, \quad (40)$$

The aperture dimensions include θ elevation, ϕ azimuth, a , and b major lobes.

The detection threshold can be determined by the least detectable value of the SNR_{min} and is calculated using the parameters minimal probability of detection P_{Dmin} and probability of false alarm P_{FA} [65]:

$$SNR_{min} = 2(\text{erfc}^{-1}(2P_{FA}) - \text{erfc}^{-1}(2P_{min}))^2. \quad (41)$$

The strength S of the received signal is determined by the radar equation:

$$S = \frac{PG^2\lambda^2RCS}{(4\pi)^3r^4} \cdot \frac{1}{L_A L_{atm}}, \quad (42)$$

where P is the transmitted power, G is the gain of the antenna, λ is the wavelength, r is the distance from the radar sensor to the object, L_A is the additional system losses and L_{atm} is the atmospheric losses.

The specific RCS cross-section maps for close objects are determined by radar sensor resolution, object size, direction of incidence, object occlusion, and object merging (Figure 3.16) [26].

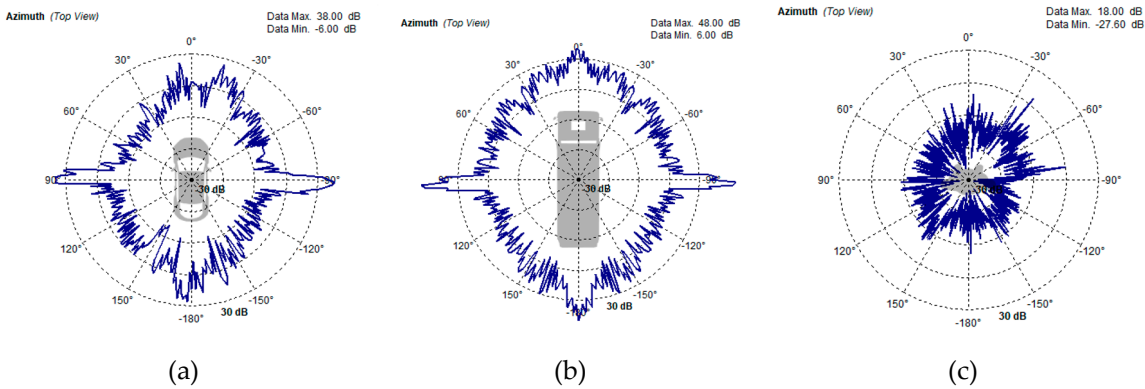


Figure 3.16. RCS of the various objects: (a) vehicle, (b) truck, (c) pedestrian.

The transmit gain map adjusts the power of the delivered signal to 3D objects. The transmit gain is calculated using linear interpolation of the parameterized gain map (Figure 3.17).

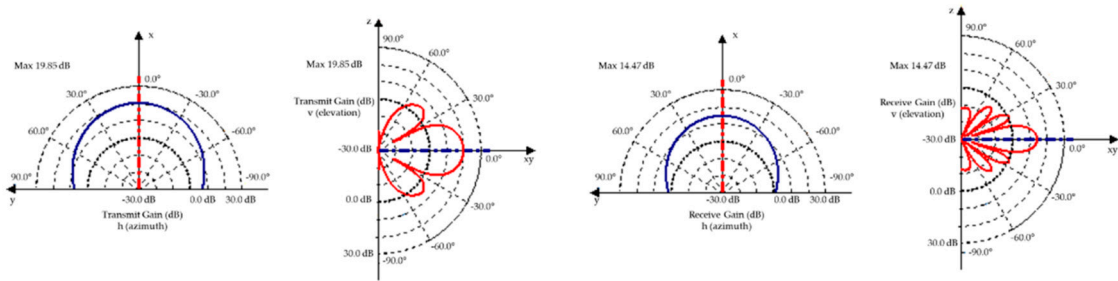


Figure 3.17. Transmit/receive (azimuth/elevation) gain map.

3.3. Characteristics of RSI Sensor

3.3.1. Lidar RSI Sensor

The lidar (light detection and ranging) sensor operates based on measuring the ToF for a beam of laser pointers with a wavelength of 905 nm that is emitted at objects in the area and computing the reception time of this beam reflected by these items. Lidars, including RSI sensors, generally use the ToF principle to determine distance. This involves generating a laser pulse and measuring its return time after hitting an object, as shown in Figure 3.18 [125].

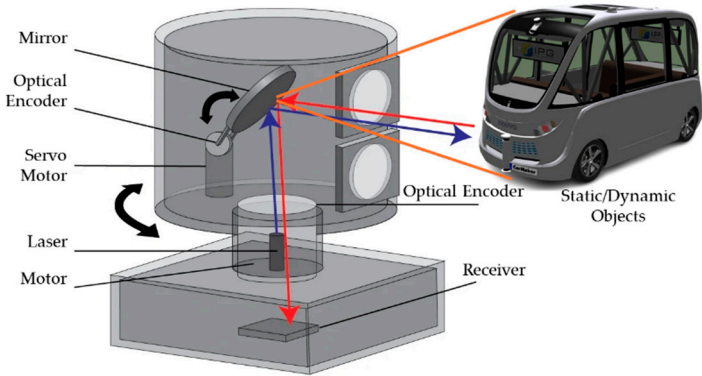


Figure 3.18. ToF principle.

The lidar 2D sensor captures information about nearby objects by sending a single laser beam onto a revolving mirror perpendicular to the axis of rotation. The lidar 3D sensor gathers information

about nearby objects by shooting a beam of laser rays through a rotating mechanism, resulting in a point cloud for the contour of these objects and the capacity to build high-precision 3D maps.

Lidar RSI sensors use a rotating equipment to guide a laser beam across their range of vision. This scanning is often accomplished using rotating polygonal mirrors that direct the beam with each facet, resulting in a balance of accuracy and speed. Alternatively, revolving prisms can refract the beam with great precision and stability. A recent technique uses MEMS (Micro-ElectroMechanical Systems) technology, in which small mirrors direct the beam, allowing for more compact designs but potentially limiting the range and scan angle.

Lidar generates a 3D representation of the viewed reality by measuring the ToF at multiple points within their FoV (Field of View). The set of points is referred to as a point cloud. The n th measurement point (p_n) in the lidar reference system $\{L\}$ can be expressed as follows [126]:

$$\{L\}_{p_n} = \left[\{L\}_{x_n}, \{L\}_{y_n}, \{L\}_{z_n} \right]^T = \frac{c}{2} t_{ToF,n} \cdot \{L\}_{\hat{s}_n}, \quad (43)$$

where c is the speed of light in air, $t_{ToF,n}$ is the ToF measure, and \hat{s}_n is the unitary vector indicating the scanning direction of the lidar in its reference system $\{L\}$. Equation (43) shows that accurate and exact point clouds require both the ToF measurement ($t_{ToF,n}$) and the scanning direction (\hat{s}_n). The scanning direction of the lidar, designated as \hat{s} , is expressed in the laser source's reference system $\{I\}$. The mirror's tilt angles (α and β) correspond to horizontal and vertical directions, respectively. This relationship is further developed in the following equations [126]:

$$\{I\}_{\hat{s}} = \{I\}_i - (\gamma(\alpha, \beta, \varphi) - \sqrt{(\gamma(\alpha, \beta, \varphi)^2)} \cdot \{I\}_{\hat{n}}, \quad (44)$$

$$\{I\}_{\hat{n}} = \begin{bmatrix} \sin\alpha \cdot \cos\beta \\ \cos\varphi \cdot \sin\beta + \sin\varphi \cdot \cos\alpha \cdot \cos\beta \\ \sin\varphi \cdot \sin\beta - \cos\varphi \cdot \cos\alpha \cdot \cos\beta \end{bmatrix}, \quad (45)$$

$$\gamma(\alpha, \beta, \varphi) = \{I\}_{\hat{n}3} = \sin\varphi \cdot \sin\beta - \cos\varphi \cdot \cos\alpha \cdot \cos\beta. \quad (46)$$

In the laser source reference system $\{I\}$, γ represents the third component of the normal vector \hat{n} , whereas α and β represent the scanning tilt angles, direction, and mirror surface. Each point in the cloud is described by its (x , y , z) coordinates, which are computed from the measured distance, the horizontal angle (determined by the revolving mirror or prism), and the vertical angle (typically fixed but variable in multi-emitter systems).

For the CarMaker virtual vehicle model, lidar 2D and 3D sensors are defined as lidar RSI sensors that provide information about nearby objects and behave just like genuine sensors. The calibration of the lidar virtual sensor was performed out using simulation data from various surroundings that was correlated with the motion sensor [127].

Lidar 2D sensors are used in pairs of two units to reduce coverage gaps and ensure continuous visibility throughout the entire surface of a horizontal scanning plane (P2D) with a viewing angle of 180° and a field of view centered on the virtual vehicle model's median longitudinal plane. Lidar 3D sensors scan on 16 channels, following a planes (P3D) with successive inclinations along an increasing angular axis with respect to the upper horizontal scanning plane. They have a viewing angle of 30° and an opening of $\pm 15^\circ$ with respect to the median longitudinal plane of the virtual vehicle model (Figure 3.19).

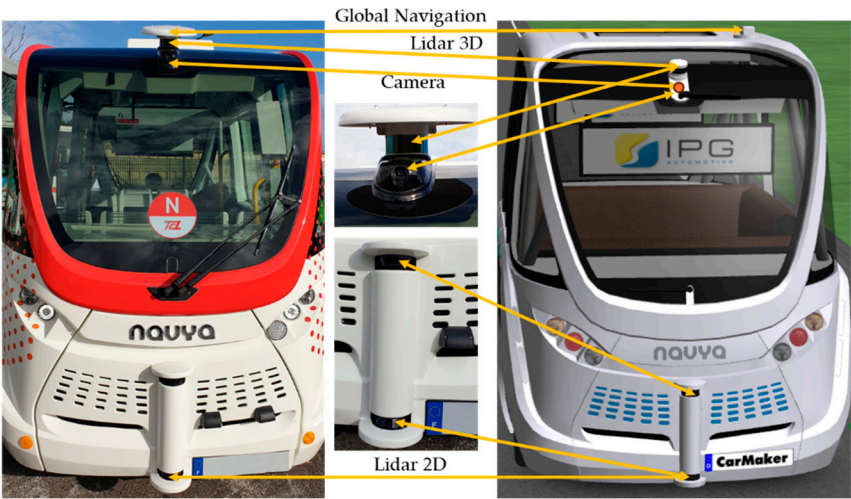


Figure 3.19. RSI sensor distribution on the virtual vehicle model's body structure.

- The interaction modes of the lidar RSI sensor are classified using the following criteria [65]:
1. Diffuse reflected laser rays are distributed uniformly, regardless of the direction of the incident ray (Lambertian reflection), with the intensity of the reflected ray decreased by the inverse of the angle between the incident ray and the normal of the reflective surface;
 2. Retroreflective means that incident laser rays are reflected back in the direction of the emitter, with the intensity of the reflected ray being reduced based on the reflectance parameters and incident angle;
 3. Specular, the incident and reflected laser rays form identical angles with the normal of the reflective surface, and the incident and reflected laser rays are in the identical plane;
 4. Transmissive: the incident laser photons keep their course but are attenuated by transmission.

3.3.2. Ultrasonic RSI Sensor

CarMaker's ultrasonic RSI model is based on the signal chain of a real ultrasonic sensor. The process begins with a transmitter illuminating the environment and emitting sound waves (Figure 3.20). This method divides sound waves into a finite number of rays. This provides for a balance between simulation speed and physical accuracy. Environmental circumstances with physical effects on wave propagation are considered into account, and the Helmholtz equation is used to compute electrical scatter fields on object surfaces utilizing parameterizable material properties to predict reflections. Every suitable ray returns to the receiver [128].

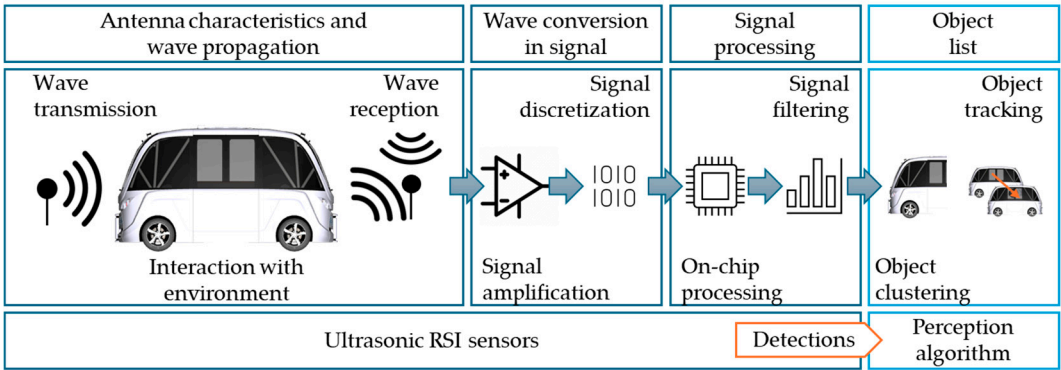


Figure 3.20. Signal chain of the ultrasonic RSI sensor models.

For all detections, the sensor model returns the sound pressure level as well as the flight time. It accounts for dense packaging and associated effects by optionally sensing cross-echoes between

sensors. Interfaces enable the replacement of individual steps with user-defined code, and clustering on available GPUs improves performance [128], as seen in Figure 3.21.

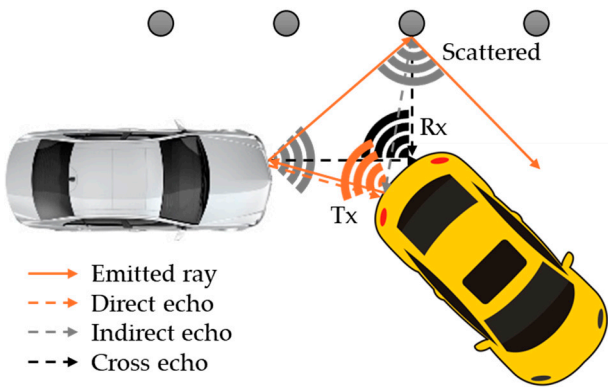


Figure 3.21. A diagram of the ray tracing algorithm used to simulate a sound wave.

The ultrasonic RSI sensor uses mechanical acoustic pressure waves that are reflected by obstacles in the immediate vicinity of the virtual vehicle model using the ToF principle, and the distance to the respective obstacles is accurately calculated using the SPA (Sound Pressure Amplitude) (Figure 3.22). The sensor considers the impacts of overlapping objects in the area, the effects of physical propagation, and the classification as false positives or false negatives of observed objects. The propagation modes of acoustic pressure waves are classified using the following criteria [65]:

1. Direct Echo: the acoustic pressure wave is reflected once by an object in close proximity and received by the emitting sensor;
2. Indirect Echo: the acoustic pressure wave is reflected at least twice by objects or surfaces in the vicinity and received by the emitting sensor;
3. Repeated Echo: the emitting sensor receives the acoustic pressure wave after it has been reflected by nearby objects or surfaces;
4. Cross Echo: the reflected acoustic pressure wave is received by a sensor other than the originating sensor, resulting in a propagation mode known as cross echo;
5. Road Clutter occurs when the acoustic pressure wave is reflected by bumps or irregularities in the roadway.

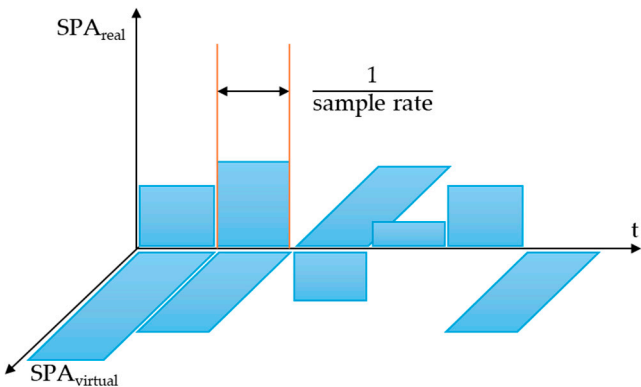


Figure 3.22. SPA full wave form.

Ultrasonic RSI sensors are installed in the vehicle's front bumper, side panels, and rear bumper.

4. Virtual Sensor Parametrization

4.1. Ideal Sensor Parametrization

The Ideal sensors incorporated in the virtual vehicle model are as follows: Slip Angle, Inertial, Object, Free Space, Traffic Sign, Line, Road and Object by Line.

To highlight the major parameters generated by the indicated Ideal sensors, a series of simulations have been performed on the proposed route (see Figure 2.8), digitized, and implemented in the IPGRoad/CarMaker application.

For the slip angle sensor (Figure 4.1a), one parameter has been chosen: Sensor.SAngle.SL00.Ang (Figure 4.1b). The analysis was performed on an area of the route traveled, starting with a straight road section, then a left turn and another straight road section.

The graphic illustrates how the slip angle sensor's values change when the vehicle approaches a left curve (second 7) and returns to a straight road segment (second 25). According to the data displayed on the graph in the image, the maximum slip angle is around 0.25 degrees.

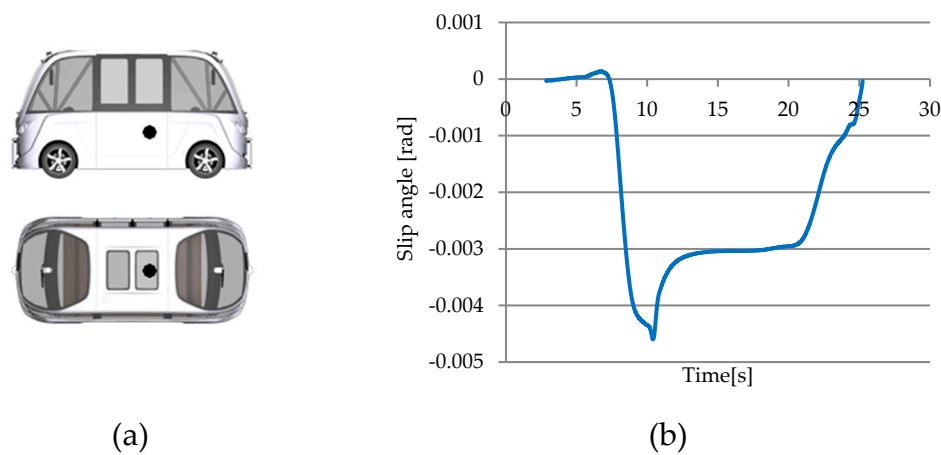


Figure 4.1. Slip angle sensor parameterization and generated parameter.

For the inertial sensor (Figure 4.2a), two parameters have been chosen: Sensor.Inertial.YRS.Vel_0.x (red curve) for the speed in the x direction and Sensor.Inertial.YRS.Acc_B.y (blue curve) for the lateral acceleration (Figure 4.2b).

The graphic shows that lateral acceleration increases from 0 m/s^2 when moving straight to 5.3 m/s^2 when turning. The vehicle's lateral acceleration reduces as it exits the turn, returning to near-zero m/s^2 levels. The sensor's observed lateral accelerations correspond with typical turns.

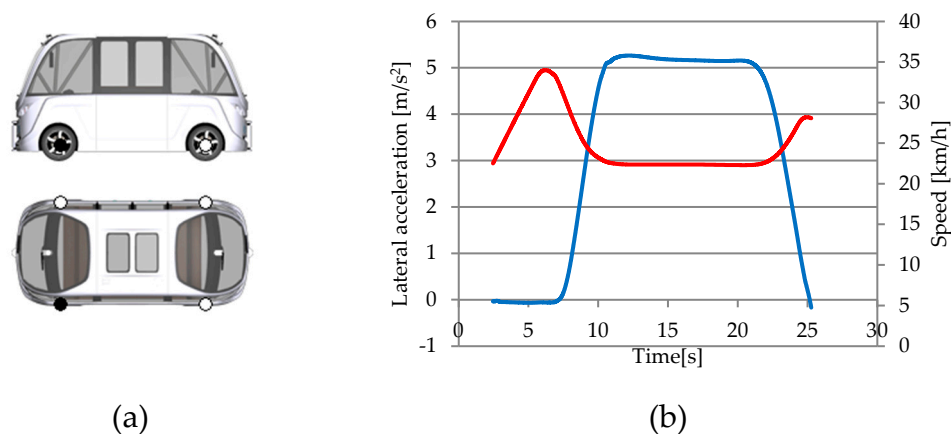


Figure 4.2. Inertial sensor parameterization and generated parameters.

For the object sensor (Figure 4.3a), two parameters have been chosen: Sensor.Object.RadarL.relvTgt.NearPnt.ds.x (red curve) represents the distance (ds) along the specified x-axis, corresponding to the direction of the sensor system axes, from the most relevant target (relvTgt), respectively the detected object that is most important to track Sensor.Object.RadarL.relvTgt.NearPnt.ds.y (blue curve) represents the distance (ds) along a specific y-axis parallel to the sensor system axes and the detected object's nearest point (NearPnt) (Figure 4.3b). This value represents the shortest distance between the sensor and the relevant target (a point on the target), which could be a vehicle, a pedestrian etc.

The graphic shows the variations in x and y coordinates over time for the nearest point on an obstacle detected by the sensor. The coordinates corresponding to the z axis (height) have not been shown since they contain relatively minor fluctuations that have no significant effect on the position of the obstacle.

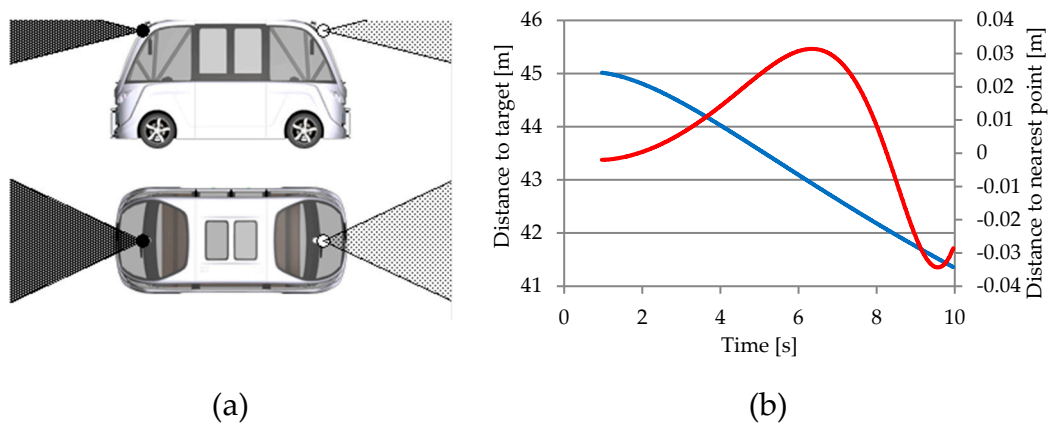


Figure 4.3. Object sensor parameterization and generated parameters.

For the free space sensor (Figure 4.4a), two parameters have been chosen: Sensor.FSpace.Front.Segm.0.ds.x (red curve), and Sensor.FSpace.Front.Segm.198.ds.x (blue curve) that represents the segmentation areas, includes the closest point of a detected traffic obstacle. The segmentation area is divided into four quadrants. Figure 4.4b shows how the various parameters change along the x-axis, with each parameter representing a quadrant of the segmentation area. The graphs illustrate the change in expected parameter values across the segmentation region over the same time frame. Analyzing the graph which corresponds to the parameter Sensor.FSpace.Front.Segm.198.ds.x, one can see no more variations of the specified parameter after second 52, indicating that no obstacle was identified in the associated quadrant.

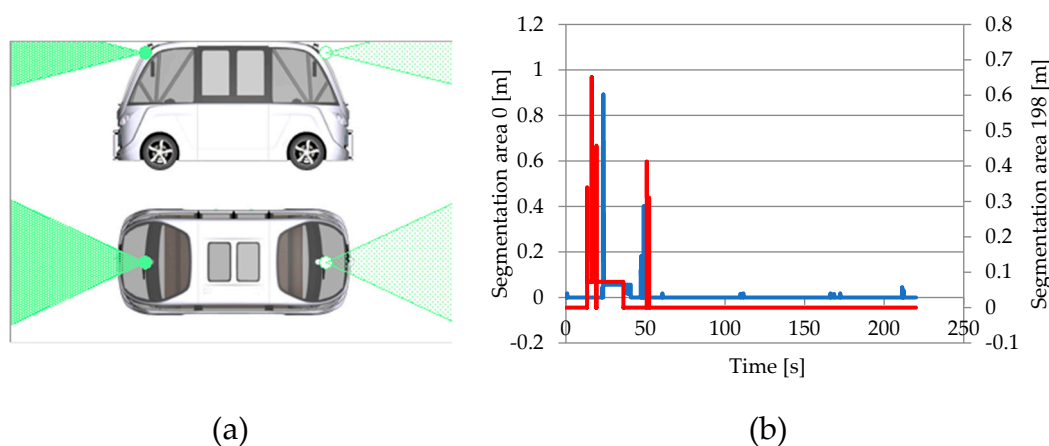


Figure 4.4. Free space sensor parameterization and generated parameters.

For the traffic sign sensor (Figure 4.5a), two parameters have been chosen: Sensor.TSign.FrontCam.SpeedLimit.0.ds.x (blue curve) indicates the distance between the detected indicator (Speed Limit sign) and the direction of travel (x), respectively Sensor.TSign.FrontCam.SpeedLimit.nSign (red curve) indicates the signal generated by the sensor when it detects a speed limitation sign (Figure 4.5b). When the distance between the vehicle (sensor) and the road sign is less than 30 meters, the signal reaches its maximum value.

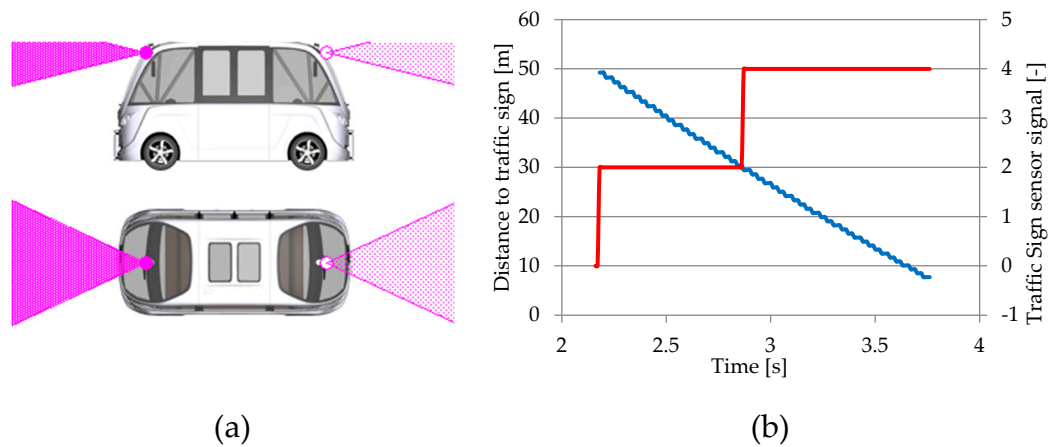


Figure 4.5. Traffic Sign sensor parameterization and generated parameters.

For the line sensor (Figure 4.6a), one parameter has been chosen: Sensor.Line.Front.RLines.1.Type (Figure 4.6b) indicates the type of longitudinal road marking located on the right side of the road in the virtual vehicle model's direction of travel. The graph shows a variation in the signal of the parameter in the range of 1 to 2. The parameter's value 1 indicates the presence of a longitudinal road marking made up of a simple dashed line, while the parameter's value 2 indicates the presence of a longitudinal road marking made up of a single continuous line.

Changing the parameter value from 1 to 2 moves the virtual vehicle model from a road section with a simple dashed line (1) to a road section with a continuous line (2). If the parameter value is constant (1 or 2), the type of longitudinal road marking does not differ; if the values fluctuate frequently a road section with many changes in longitudinal road marking will result.

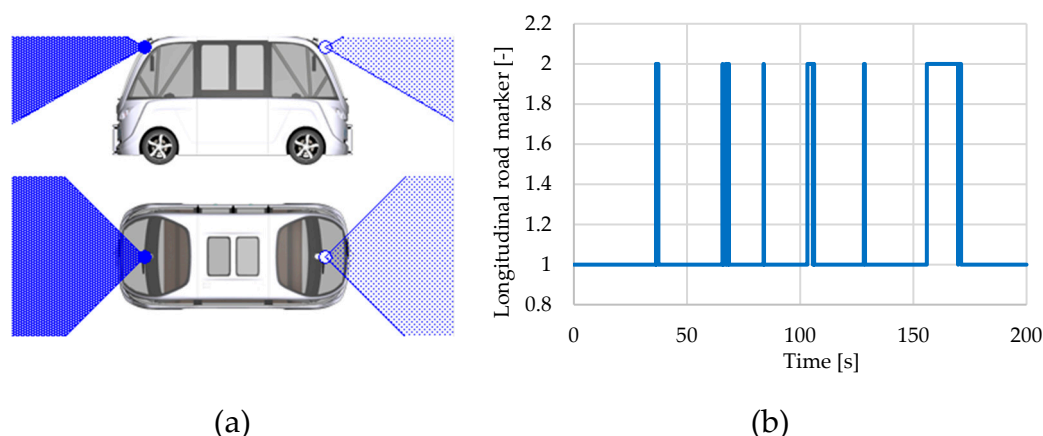


Figure 4.6. Line sensor parameterization and generated parameter.

Two parameters have been chosen for the Road sensor (Figure 4.7a): Sensor.Road.AlongR.Path.tx (blue curve) represents the vehicle displacement along the x-axis, and Sensor.Road.AlongR.Path.DevDist (red curve) represents the deviation from the planned route (Figure 4.7b). Over a route length of around 8 m, the deviation ranges between 0 and $7 \cdot 10^{-9}$ m.

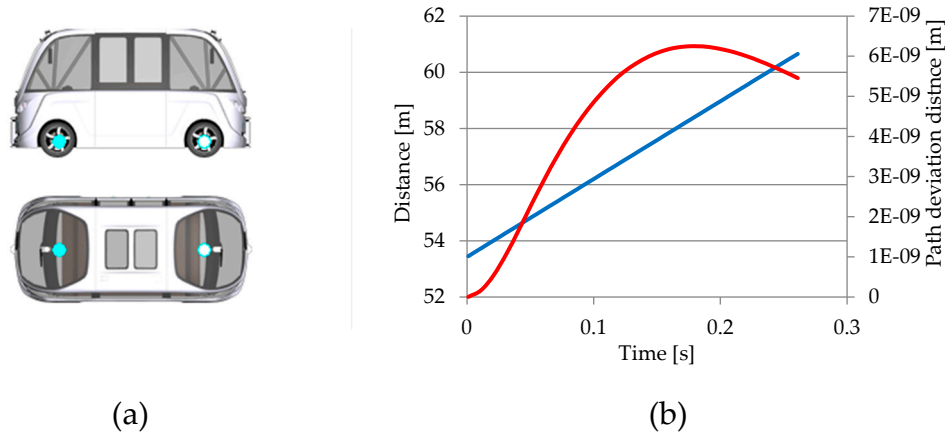


Figure 4.7. Road sensor parameterization and generated parameters.

For the object by line sensor (Figure 4.8a), one parameter has been chosen: Sensor.ObjByLine.OBL00.LinesC.0.ObjF.0.sMax (Figure 4.8b), that highlights the variation in maximum distance between the virtual vehicle model and the POI over a 2.5-second period. Figure 3.13 shows that S_{\max} decreases over time, indicating that the vehicle has approached the POI, where LinesC indicates the LaneScope Center section.

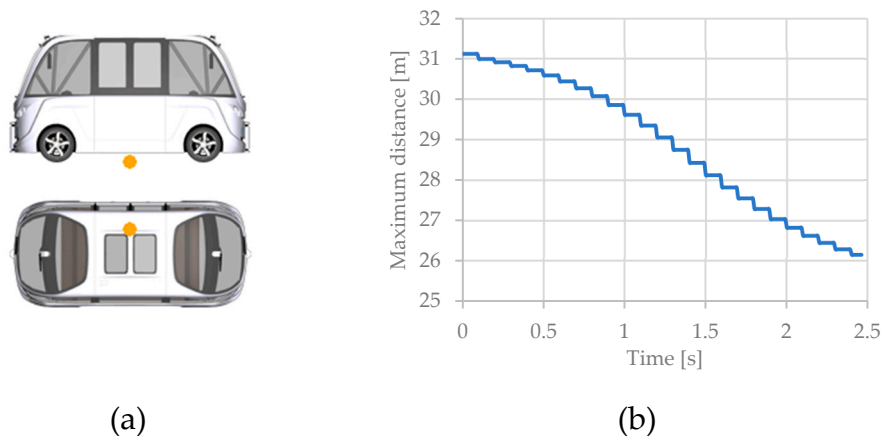


Figure 4.8. Object by line sensor parameterization and generated parameter.

4.2. Hi-Fi Sensor Parametrization

The Hi-Fi sensors incorporated in the virtual vehicle model are as follows: (a) Camera, (b) Global Navigation, (c) Radar.

To highlight the major parameters generated by the indicated Ideal sensors, a series of simulations have been performed on the proposed route (see Figure 2.8), digitized, and implemented in the IPGRoad/CarMaker application.

For the camera sensor (Figure 4.9a), two parameters have been chosen: Sensor.Camera.CA00.Obj.0.nVisPixels (x-axis) indicates the number of visible pixels of the object recognized by the sensor and Sensor.Camera.CA00.Obj.0.Confidence (y-axis) indicates the confidence degree of object detection (Figure 4.9b). The graph illustrates the efficacy of object detection based on their size in the image. If an object has many visible pixels but a low confidence level, the recognition algorithm becomes unreliable. The maximum confidence level for recognized items is 1.

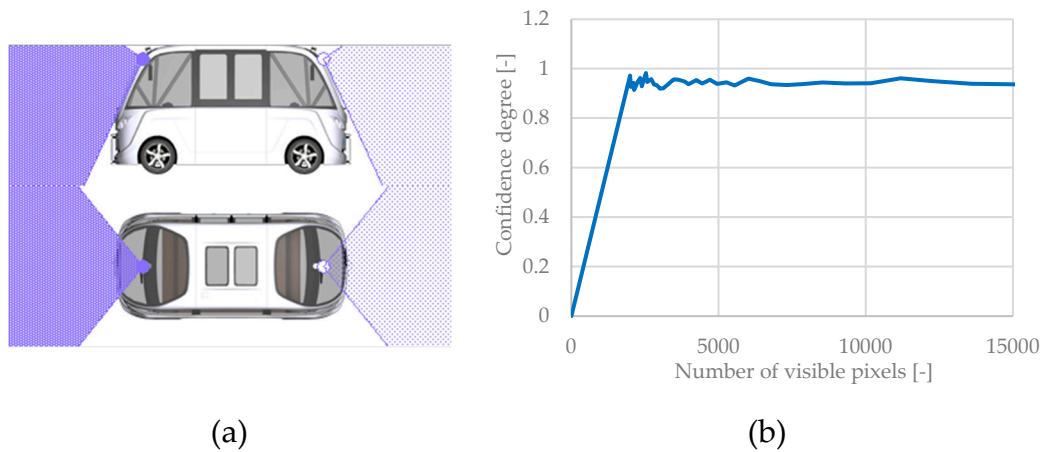


Figure 4.9. Camera sensor parameterization and generated parameter.

For the global navigation sensor (Figure 4.10a), one parameter has been chosen `Sensor.GNav.Receiver.SatNoDirectView` that highlights the number of satellites that are directly visible to the GNSS receiver in a specific time interval (Figure 4.10b).

The satellites' direct visibility influences measurement accuracy. The maximum number of visible satellites ranges from 4 to 10 and is influenced by tall buildings or underground passages that the vehicle passes through, and by extreme weather conditions.

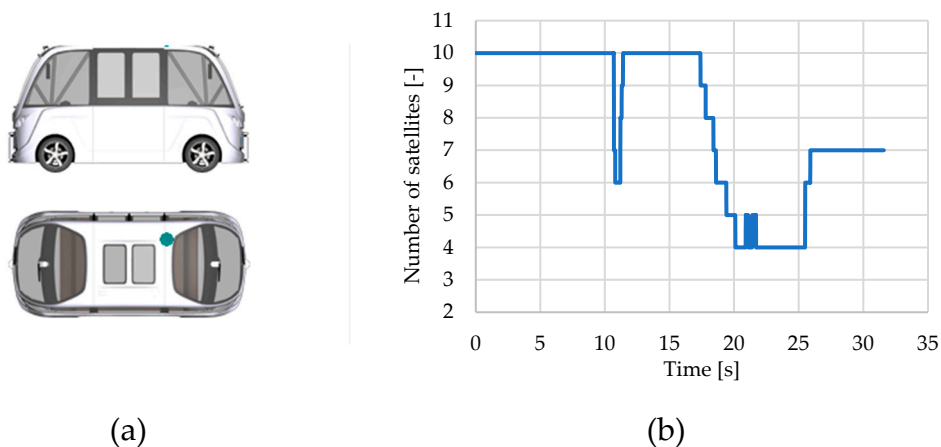


Figure 4.10. Global navigation sensor parameterization and generated parameter.

For the radar sensor (Figure 4.11a), two parameters have been chosen: `Sensor.Radar.RA00.Obj0.Dist` (blue curve) represents the variation of the signal-to-noise ratio (SNR) as a function of the distance from the object and `Sensor.Radar.RA00.Obj0.SNR` represents the signal to noise ratio (Figure 4.11b). The graph illustrates the radar signal's detection efficiency versus background noise as a function of distance to the target object. A higher signal-to-noise ratio suggests better detection. As the object moves away, the signal-to-noise ratio decreases, resulting in poorer detection.

Another parameter evaluated for the radar sensor is `Sensor.Radar.RA00.Obj0.RCS` (red curve) represents a variation of the RCS parameter (target object's radar signature) in the case of detection (Figure 4.11c). The graph shows significant changes in RCS parameter values, ranging from -18dBm^2 to 21dBm^2 . The analysis of the RCS parameter indicates that the radar sensor detects obstacles in the virtual vehicle model's travel environment, ranging from small (negative values for poor detection) to large (positive values for good detection).

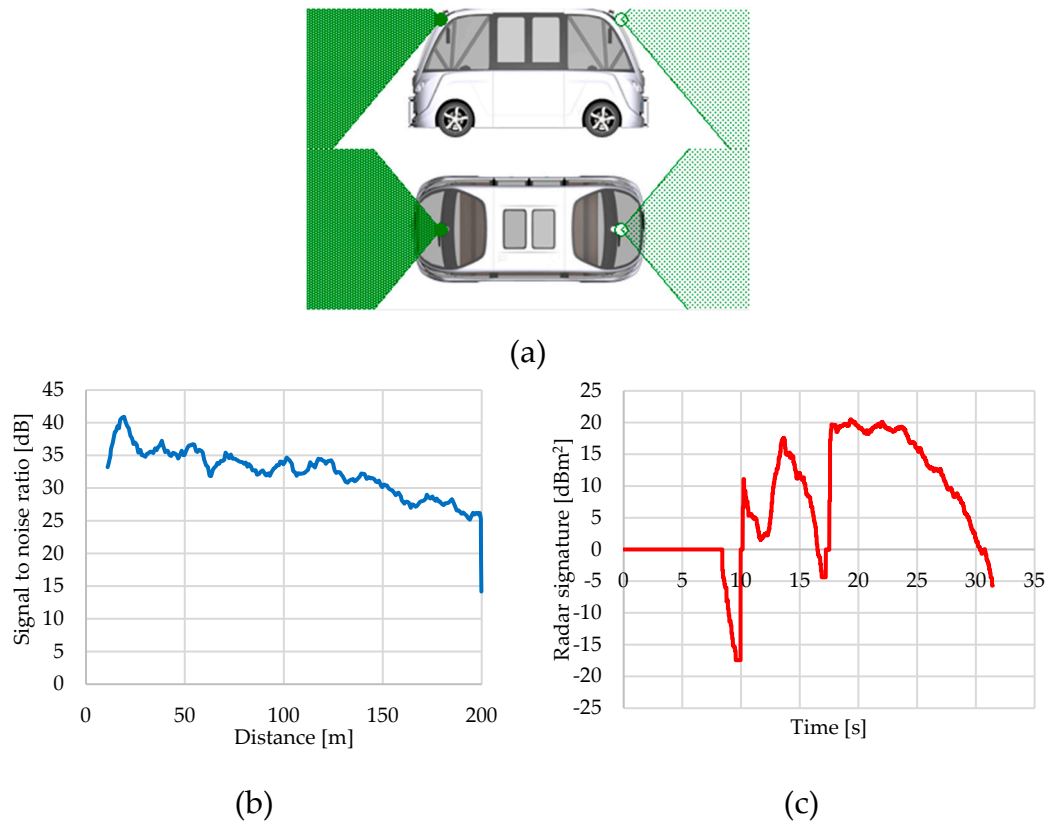


Figure 4.11. Radar sensor parameterization and generated parameters.

4.3. RSI Sensor Parametrization

The RSI sensors incorporated in the virtual vehicle model are as follows: (a) Lidar RSI, (b) Ultrasonic RSI.

To highlight the major parameters generated by the indicated Ideal sensors, a series of simulations have been performed on the proposed route (Figure 2.8), digitized, and implemented in the IPGRoad/CarMaker application.

For the lidar RSI sensor (Figure 4.12a), one parameter was chosen: Sensor.LidarRSI.LIRS00.nScanPoints indicates the number of points detected in an interval of time (Figure 4.12b). The graph's parameter variations allow for an investigation of scanning conditions in particular areas of the virtual vehicle model's travel path. A high parameter value results in a congested area with many detected obstacles, whereas a low parameter value results in an area with fewer detected obstacles.

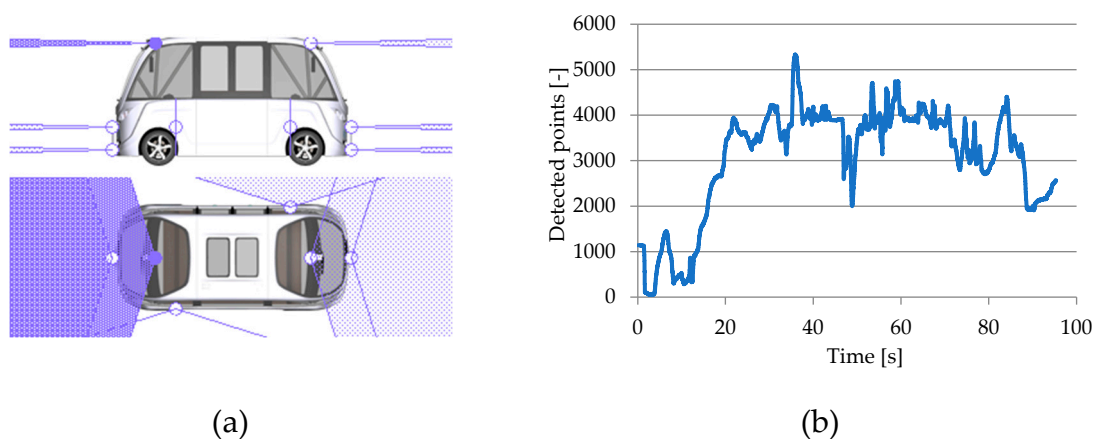


Figure 4.12. Lidar RSI sensor parameterization and generated parameter.

The ultrasonic RSI sensor (Figure 4.13a) has two parameters: Sensor.USonicRSI.Tx.USRS00.Rx.USRS00.nDetections (blue curve) and Sensor.USonicRSI.Tx.USRS01.Rx.USRS01.nDetections (red curve) represent the variation of obstacles detected by two ultrasonic sensors placed in different positions on the virtual vehicle model (Figure 4.13b). The graph illustrates how the ultrasonic sensors under consideration detect obstacles. If the variation curves for the two parameters overlap (blue/red), the considered ultrasonic sensors detect the same number of obstacles in a specific time interval, indicating the detection of a large object or a uniform surface.

If an ultrasonic sensor registers an increase in detections, it indicates that an obstacle has been recognized in the neighborhood; if the ultrasonic sensor detects nothing, it indicates that there may be blind zones around the virtual vehicle model.

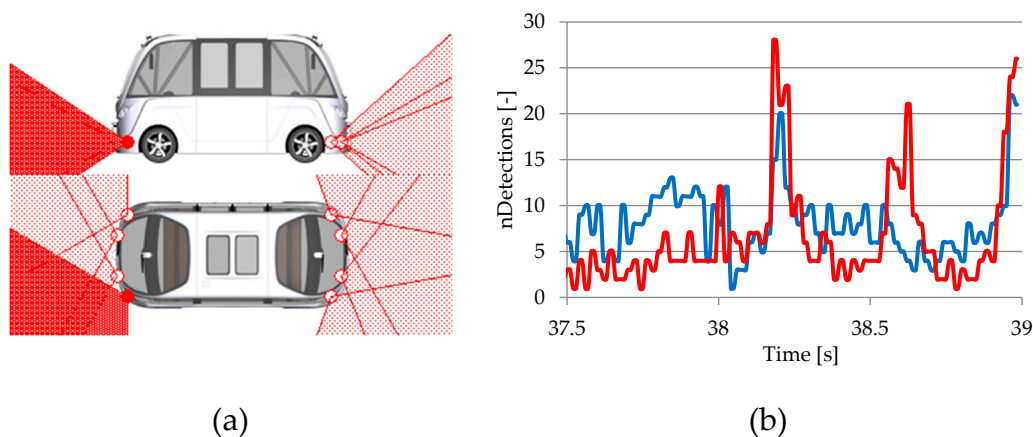


Figure 4.13. Ultrasonic RSI sensor parameterization and generated parameters.

5. Discussion

Replicating the behavior of real-world sensors in a virtual environment is a complex task. This involves capturing elements such as sensor noise, limitations in the field of view, uncertainty in physical parameters, and responses to varying environmental conditions. This can be achieved through advanced data-driven algorithms, AI-based simulation techniques, and innovative virtual sensor designs. Autonomous vehicles utilize a variety of sensors (cameras, radar, lidar, etc.), and combining data from multiple virtual and real sensors through sensor fusion techniques, each with its characteristics and potential errors, is a significant challenge [129]. Their performance can degrade due to changes in nonlinear dynamics, complex physical processes in the environment, and nonlinear interactions between input and output variables [130,131]. High-fidelity sensors can be computationally intensive to simulate real-world sensor data with high precision. This requires robust hardware and efficient algorithms to process the data in real-time [132]. Virtual sensors must operate in real time within the simulation environment, meaning they must generate and process data quickly enough to keep up with the simulation. This can be computationally demanding [133]. AI techniques enable the development of virtual sensors that can estimate physical parameters without traditional sensors, using data-driven models to emulate real-world conditions [10]. The integration of AI introduces potential uncertainties, necessitating validation to maintain trust in these systems [133].

Validating virtual sensor data against real-world data or physical sensor measurements can be difficult. Autonomous vehicles must handle unusual situations and realistically simulating these circumstances adds another layer of complexity. Virtual sensors in the overall software architecture of autonomous vehicles can be complicated, mainly if the architecture was not designed initially with virtual sensors in mind. The development and validation of virtual sensors can be time-consuming and costly, especially when dealing with complex sensor models or high-accuracy requirements.

6. Conclusions

Future studies could focus on improving the accuracy and robustness of virtual sensor models, particularly in complex and dynamic environments. This can address challenges related to nonlinear dynamics models with complex physical processes, research on environmental factors (such as weather conditions, temperature, and lighting), and detailed real-world behavioral models. Research into advanced AI and ML techniques for sensor fusion and data processing is also essential. Implementing these advancements is likely to result in more accurate and reliable data, thereby enhancing the overall performance of AV driving systems. Additionally, using DL architectures designed explicitly for particular tasks (like object detection or lane-keeping) would be highly beneficial. Efforts should also concentrate on developing robust data-driven system models capable of accurately identifying various physical characteristics. Such models can provide a viable and cost-effective alternative to traditional real sensors. Research should include testing virtual sensor performance in rare or unexpected situations, often referred to as edge cases. This requires testing and validating across a wide range of scenarios.

Validating and calibrating virtual sensors is also important and may involve establishing standardized testing procedures and metrics. Establishing industry standards for the development and validation would promote interoperability and facilitate the widespread adoption of this technology. It is also crucial to optimize algorithms and ensure that hardware can meet the computational demands of high-fidelity sensor interfaces, especially in complex scenarios. Exploring hybrid approaches that combine both virtual sensor data and real sensor data could leverage the strengths of each type, thus enhancing overall accuracy and reliability. Also, it is important to examine the necessary levels of simulation fidelity for various autonomous driving tasks to balance the need for accuracy with computational efficiency. Studying how virtual sensor data is presented to human operators in semi-autonomous systems can also be an important step, as it impacts their situational awareness and decision-making processes.

Author Contributions: Conceptualization, I.B. and C.I.; methodology, C.I.; software, C.I.; validation, C.I., H.B. and F.B.S.; formal analysis, I.B., C.I. and A.C.; resources, C.I.; data curation, H.B. and F.B.S.; writing—original draft preparation, C.I. and A.M.; writing—review and editing, I.B., C.I., H.B., C.A., A.M. and F.B.S.; visualization, C.I.; supervision, I.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Access to the data is available upon request. Access to the data can be requested via e-mail to the corresponding author.

Acknowledgments: The simulations presented in the paper were done using the software CarMaker supported by IPG Automotive GmbH, Karlsruhe, Germany.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---------|------------------------------------|
| ABS | Antilock Braking System |
| ADAS | Advanced Driver Assistance Systems |
| ACC | Adaptive Cruise Control |
| AD | Autonomous Driving |
| AEB | Automatic Emergency Braking |
| AI | Artificial Intelligence |
| AWD | All-Wheel Drive |
| BiSeNet | Bilateral Segmentation Network |

| | |
|--------|---|
| CNN | Convolutional Neural Network |
| D-GNSS | Differential-Global Navigation Satellite System |
| CPU | Central Processing Unit |
| DL | Deep Learning |
| DMS | Driver Monitoring System |
| DPM | Deformable Part Model |
| DTw | Digital Twin |
| DVM | Double Validation Metric |
| EBA | Emergency Brake Assist |
| ECEF | Earth Centered Earth Fixed |
| EDF | Empirical cumulative Distribution Function |
| EM | Energy Management |
| ESC | Electronic Stability Control |
| FC | Fuel Consumption |
| FCW | Forward Collision Warning |
| FET | Field-Effect Transistor |
| FoV | Field-Of-View |
| GCS | Geographic Coordinate System |
| GDOP | Geometric Dilution of Precision |
| GNN | Global Nearest Neighbor |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| GPU | Graphics Processing Unit |
| HD | High-Definition |
| Hi-Fi | High Fidelity |
| HiL | Hardware-in-the-Loop |
| HV | High Voltage |
| ID | IDentifier |
| ILA | Intelligent Light Assist |
| IMU | Inertial Measurement Unit |
| IoT | Internet of Things |
| LDW | Lane Departure Warning |
| Lidar | Light detection and ranging |
| LK | Lane Keeping |
| LKA | Lane Keeping Assist |
| LV | Low Voltage |
| MEMS | Micro-ElectroMechanical Systems |
| ML | Machine Learning |
| MMS | Mobile Mapping System |
| OTA | Over The Air |
| PA | Park Assist |
| PPHT | Progressive Probabilistic Hough Transform |
| POI | Point Of Interest |
| PS | Physical Sensors |
| PT | Powertrain |
| QTC | Quantum Tunnelling Composite |
| RCS | Radar Cross Section |
| RSI | Raw Signal Interface |
| RTK | Real-Time Kinematic |
| SAS | Smart Airbag System |
| SD | Sign Detection |
| SiL | Software-in-the-Loop |
| SLAM | Simultaneous Localization And Mapping |
| SNR | Signal-to-Noise Ratio |
| SPA | Sound Pressure Amplitude |
| SRTM | Shuttle Radar Topography Mission |
| ToF | Time of Flight |
| TSA | Traffic Sign Assist |
| V2I | Vehicle-to-Infrastructure |

| | |
|------|-----------------------------|
| V2V | Vehicle-to-Vehicle |
| VDM | Vehicle Dynamic Model |
| VLC | Visible Light Communication |
| VS | Virtual Sensors |
| WLD | Wheel Lifting Detection |
| YOLO | You Only Look Once |

References

1. Martin, D.; Kühl, N.; Satzger, G. Virtual Sensors. *Bus. Inf. Syst. Eng.* **2021**, *63*, 315–323. <https://doi.org/10.1007/s12599-021-00689-w>
2. Dahiya, R.; Ozioko, O.; Cheng, G.; *Sensory Systems for Robotic Applications*, Publisher: MIT Press, Cambridge, Massachusetts, 2022. <https://doi.org/10.1049/pbce097e>
3. Šabanovič, E.; Kojis, P.; Šukevičius, Š.; Shyrokau, B.; Ivanov, V.; Dhaens, M.; Skrickij, V. Feasibility of a Neural Network-Based Virtual Sensor for Vehicle Unsprung Mass Relative Velocity Estimation. *Sensors* **2021**, *21*, 7139. <https://doi.org/10.3390/s21217139>
4. Persson, J.A.; Bugeja, J.; Davidsson, P.; Holmberg, J.; Kebande, V.R.; Mihailescu, R.-C.; Sarkheyli-Hägele, A.; Tegen, A. The Concept of Interactive Dynamic Intelligent Virtual Sensors (IDIVS): Bridging the Gap between Sensors, Services, and Users through Machine Learning. *Appl. Sci.* **2023**, *13*, 6516. <https://doi.org/10.3390/app13116516>
5. Ambarish, P.; Mitradip, B.; Ravinder, D. *Solid-State Sensors (IEEE Press Series on Sensors)*, Publisher: Wiley-IEEE Press, 2023. doi:10.1002/9781119473114
6. Shin, H.; Kwak, Y. Enhancing digital twin efficiency in indoor environments: Virtual sensor-driven optimization of physical sensor combinations, *Automat. Constr.* **2024**, *161*, 105326. <https://doi.org/10.1016/j.autcon.2024.105326>
7. Stanley, M.; Lee, J. *Sensor Analysis for the Internet of Things*, Publisher: Morgan & Claypool Publishers, Arizona State University, 2018.
8. Stetter, R. A Fuzzy Virtual Actuator for Automated Guided Vehicles. *Sensors* **2020**, *20*, 4154. <https://doi.org/10.3390/s20154154>
9. Xie, J.; Yang, R.; Gooi, H.B.; Nguyen, H. PID-based CNN-LSTM for accuracy-boosted virtual sensor in battery thermal management system, *Appl. Energ.* **2023**, *331*, 120424. <https://doi.org/10.1016/j.apenergy.2022.120424>
10. Fabiocchi, D.; Giulietti, N.; Carnevale, M.; Giberti, H. AI-Driven Virtual Sensors for Real-Time Dynamic Analysis of Mechanisms: A Feasibility Study. *Machines* **2024**, *12*, 257. <https://doi.org/10.3390/machines12040257>
11. Kabadayi, S.; Pridgen, A.; Julien, C. *Virtual sensors: Abstracting data from physical sensors*. In IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, United States, Buffalo-Niagara Falls, NY, 26.06.2006-29.06.2006 (26 June 2006). doi:10.1109/WOWMOM.2006.115
12. Compredict, Available online: <https://compredict.ai/virtual-sensors-replacing-vehicle-hardware-sensors/> (Accessed February, 6 2025).
13. Prokhorov, D. *Virtual Sensors and Their Automotive Applications*, In 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, VIC, Australia, 05-08 December 2005. doi:10.1109/ISSNIP.2005.1595614
14. Forssell, U.; Ahlqvist, S.; Persson, N.; Gustafsson, F. *Virtual Sensors for Vehicle Dynamics Applications*. In: Krueger, S., Gessner, W. (eds) *Advanced Microsystems for Automotive Applications 2001*. VDI-Buch. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-18253-2_25
15. Hu, X.H.; Cao, L.; Luo, Y.; Chen, A.; Zhang, E.; Zhang, W. *A Novel Methodology for Comprehensive Modeling of the Kinetic Behavior of Steerable Catheters*. In IEEE/ASME Transactions on Mechatronics, August 2019, doi:10.1109/TMECH.2019.2928786
16. Cummins, Available online: <https://www.cummins.com/news/2024/03/18/virtual-sensors-and-their-role-energy-future> (Accessed February, 6 2025)
17. Bucaioni, A.; Pelliccione, P.; Mubeen, S. Modelling centralised automotive E/E software architectures, *Adv. Eng. Inform.* **2024**, *59*, 102289, <https://doi.org/10.1016/j.aei.2023.102289>

18. Zhang, Q.; Shen, S.; Li, H.; Cao, W.; Tang, W.; Jiang, J.; Deng, M.; Zhang, Y.; Gu, B.; Wu, K.; Zhang, K.; Liu, S. Digital twin-driven intelligent production line for automotive MEMS pressure sensors, *Adv. Eng. Inform.* **2022**, *54*, 101779. <https://doi.org/10.1016/j.aei.2022.101779>
19. Ida, N. *Sensors, Actuators, and Their Interfaces: A multidisciplinary introduction*, 2nd Ed. Publisher: The Institution of Engineering and Technology, 2020. <https://doi.org/10.1049/PBCE127E>
20. Masti, D.; Bernardini, D.; Bemporad, A. A machine-learning approach to synthesize virtual sensors for parameter-varying systems, *Eur. J. Control.* **2021**, *61*, 40–49. <https://doi.org/10.1016/j.ejcon.2021.06.005>
21. Paepae, T.; Bokoro, P.N.; Kyamakya, K. From fully physical to virtual sensing for water quality assessment: A comprehensive review of the relevant state-of-the-art. *Sensors* **2021**, *21*(21), 6971. <https://doi.org/10.3390/s21216971>
22. Tihanyi, V.; Tettamanti, T.; Csonthó, M.; Eichberger, A.; Ficzer, D.; Gangel, K.; Hörmann, L.B.; Klaffenböck, M.A.; Knauder, C.; Luley, P.; et al. Motorway Measurement Campaign to Support R&D Activities in the Field of Automated Driving Technologies. *Sensors* **2021**, *21*(6), 2169. <https://doi.org/10.3390/s21062169>
23. Tactile Mobility. Available online: <https://tactilemobility.com/> (Accessed February, 6 2025)
24. Compredict-Virtual Sensor Platform. Available online: <https://compredict.ai/virtual-sensor-platform/> (Accessed February, 6 2025)
25. Mordor Intelligence. Available online: <https://www.mordorintelligence.com/industry-reports/virtual-sensors-market> (Accessed February, 6 2025)
26. Iclodean, C.; Varga, B.O.; Cordoş, N. *Autonomous Driving Technical Characteristics*. In: *Autonomous Vehicles for Public Transportation, Green Energy and Technology*, Publisher: Springer, 2022, pp. 15–68. doi: https://doi.org/10.1007/978-3-031-14678-7_2
27. SAE. Available online: https://www.sae.org/standards/content/j3016_202104/ (Accessed February, 6 2025)
28. Muhovič, J.; Perš, J. Correcting Decalibration of Stereo Cameras in Self-Driving Vehicles. *Sensors* **2020**, *20*, 3241. <https://doi.org/10.3390/s20113241>
29. Hamidaoui, M.; Talhaoui, M.Z.; Li, M.; Midoun, M.A.; Haouassi, S.; Mekkaoui, D.E.; Smaili, A.; Cherraf, A.; Benyoub, F.Z. Survey of Autonomous Vehicles' Collision Avoidance Algorithms. *Sensors* **2025**, *25*, 395. <https://doi.org/10.3390/s25020395>
30. Cabon, Y.; Murray, N.; Humenberger, M. Virtual KITTI 2. arXiv e-prints **2020**, Art. no. arXiv:2001.10773. <https://doi.org/10.48550/arXiv.2001.10773>
31. Mallik, A.; Gaopande, M.L.; Singh, G.; Ravindran, A.; Iqbal, Z.; Chao, S.; Revalla, H.; Nagasamy, V. Real-time Detection and Avoidance of Obstacles in the Path of Autonomous Vehicles Using Monocular RGB Camera. *SAE Int. J. Adv. Curr. Pract. Mobil.* **2022**, *5*, 622–632. <https://doi.org/10.4271/2022-01-0074>
32. Zhe, T.; Huang, L.; Wu, Q.; Zhang, J.; Pei, C.; Li, L. Inter-Vehicle Distance Estimation Method Based on Monocular Vision Using 3D Detection. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4907–4919. doi.org/10.1109/tvt.2020.2977623
33. Rill, R.A.; Faragó, K.B. Collision avoidance using deep learning-based monocular vision. *SN Comput. Sci.* **2021**, *2*, 375. <https://doi.org/10.1007/s42979-021-00759-6>
34. He, J.; Tang, K.; He, J.; Shi, J. Effective vehicle-to-vehicle positioning method using monocular camera based on VLC. *Opt. Express* **2020**, *28*, 4433–4443. <https://doi.org/10.1364/OE.382482>
35. Choi, W.Y.; Yang, J.H.; Chung, C.C. Data-Driven Object Vehicle Estimation by Radar Accuracy Modeling with Weighted Interpolation. *Sensors* **2021**, *21*, 2317. <https://doi.org/10.3390/s21072317>
36. Muckenhuber, S.; Museljic, E.; Stettinger, G. Performance evaluation of a state-of-the-art automotive radar and corresponding modeling approaches based on a large labeled dataset. *J. Intell. Transport. S.* **2022**, *26*, 655–674. <https://doi.org/10.1080/15472450.2021.1959328>
37. Sohail, M.; Khan, A.U.; Sandhu, M.; Shoukat, I.A.; Jafri, M.; Shin, H. Radar sensor based Machine Learning approach for precise vehicle position estimation. *Sci. Rep.* **2023**, *13*, 13837. <https://doi.org/10.1038/s41598-023-40961-5>
38. Srivastav, A.; Mandal, S. Radars for autonomous driving: A review of deep learning methods and challenges. *IEEE Access* **2023**, *11*, 97147–97168. doi:10.1109/ACCESS.2023.3312382

39. Poulouse, A.; Baek, M.; Han, D.S. Point cloud map generation and localization for autonomous vehicles using 3D lidar scans. In Proceedings of the 2022 27th Asia Pacific Conference on Communications (APCC), Jeju, Republic of Korea, 19–21 October 2022; pp. 336–341. doi:10.1109/APCC55198.2022.9943630
40. Saha, A.; Dhara, B.C. 3D LiDAR-based obstacle detection and tracking for autonomous navigation in dynamic environments. *Int. J. Intell. Robot. Appl.* **2024**, *8*, 39–60. <https://doi.org/10.1007/s41315-023-00302-1>
41. Dazlee, N.M.A.A.; Khalil, S.A.; Rahman, S.A.; Mutalib, S. Object detection for autonomous vehicles with sensor-based technology using YOLO. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 129–134. <https://doi.org/10.18201/ijisae.2022.276>
42. Guan, L.; Chen, Y.; Wang, G.; Lei, X. Real-time vehicle detection framework based on the fusion of LiDAR and camera. *Electronics* **2020**, *9*, 451. <https://doi.org/10.3390/electronics9030451>
43. Kotur, M.; Lukić, N.; Krunic, M.; Lukač, Ž. Camera and LiDAR sensor fusion for 3d object tracking in a collision avoidance system. In Proceedings of the 2021 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2021; pp. 198–202. doi:10.1109/ZINC52049.2021.9499281
44. Choi, W.Y.; Kang, C.M.; Lee, S.H.; Chung, C.C. Radar accuracy modeling and its application to object vehicle tracking. *Int. J. Control. Autom. Syst.* **2020**, *18*, 3146–3158. <https://doi.org/10.1007/s12555-019-1066-5>
45. Simcenter. Available online: <https://blogs.sw.siemens.com/simcenter/the-sense-of-virtual-sensors/> (Accessed February, 6 2025)
46. Kim, J.; Kim, Y.; Kum, D. Low-level sensor fusion network for 3D vehicle detection using radar range-azimuth heatmap and monocular image. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–4 December 2020. https://doi.org/10.1007/978-3-030-69535-4_24
47. Lim, S.; Jung, J.; Lee, B.H.; Choi, J.; Kim, S.C. Radar sensor-based estimation of vehicle orientation for autonomous driving. *IEEE Sensors J.* **2022**, *22*, 21924–21932. doi:10.1109/JSEN.2022.3210579
48. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020, pp. 11621–11631. doi:10.1109/CVPR42600.2020.01164
49. Robsrud, D.N.; Øvsthus, Ø.; Muggerud, L.; Amendola, J.; Cenkeramaddi, L.R.; Tyapin, I.; Jha, A. Lidar-mmW Radar Fusion for Safer UGV Autonomous Navigation with Collision Avoidance. In Proceedings of the 2023 11th International Conference on Control, Mechatronics and Automation (ICCMA), Grimstad, Norway, 1–3 November 2023; pp. 189–194. doi:10.1109/ICCMA59762.2023.10375017
50. Wang, Y.; Jiang, Z.; Gao, X.; Hwang, J.N.; Xing, G.; Liu, H. RODnet: Radar object detection using cross-modal supervision. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 504–513. <https://doi.org/10.48550/arXiv.2003.01816>
51. Rövid, A.; Remeli, V.; Paufler, N.; Lengyel, H.; Zöldy, M.; Szalay, Z. Towards Reliable Multisensory Perception and Its Automotive Applications. *Period. Polytech. Transp. Eng.* **2020**, *48(4)*, 334–340. <https://doi.org/10.3311/PPtr.15921>
52. IPG, CarMaker. Available online: <https://www.ipg-automotive.com/en/products-solutions/software/carmaker/> (Accessed February, 6 2025)
53. Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors* **2021**, *21*, 2140. <https://doi.org/10.3390/s21062140>
54. Liu, X.; Baiocchi, O. A comparison of the definitions for smart sensors, smart objects and Things in IoT. In 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2016, pp. 1–4. doi:10.1109/IEMCON.2016.7746311
55. Peinado-Asensi, I.; Montés, N.; García, E. Virtual Sensor of Gravity Centres for Real-Time Condition Monitoring of an Industrial Stamping Press in the Automotive Industry. *Sensors* **2023**, *23*, 6569. <https://doi.org/10.3390/s23146569>
56. Stetter, R.; Witczak, M.; Pazera, M. Virtual Diagnostic Sensors Design for an Automated Guided Vehicle. *Appl. Sci.* **2018**, *8*, 702. <https://doi.org/10.3390/app8050702>
57. Lengyel, H.; Maral, S.; Kerebekov, S.; Szalay, Z.; Török, Á. Modelling and simulating automated vehicular functions in critical situations—application of a novel accident reconstruction concept. *Vehicles* **2023**, *5(1)*, 266–285. <https://doi.org/10.3390/vehicles5010015>

58. Dörr, D. *Using Virtualization to Accelerate the Development of ADAS & Automated Driving Functions*. IPG Automotive, GTC Europe München, 28 September 2017.
59. Kim, J.; Park, S.; Kim, J.; Yoo, J. A Deep Reinforcement Learning Strategy for Surrounding Vehicles-Based Lane-Keeping Control. *Sensors* **2023**, *23*, 9843. <https://doi.org/10.3390/s23249843>
60. Pannaggar, P.; Nilac, D.; Orucevic, F.; Eichberger, A.; Rogic, B. Advanced Lane Detection Model for the Virtual Development of Highly Automated Functions. arXiv:2104.07481, 2021. <https://doi.org/10.48550/arXiv.2104.07481>
61. *IPG Guide-User's Guide Version 12.0.1 CarMaker*, IPG Automotive, 2023.
62. Iclodean, C.; Varga, B.O.; Cordoş, N. *Virtual Model*. In: Autonomous Vehicles for Public Transportation, Green Energy and Technology, Publisher: Springer, 2022, pp. 195-335. https://doi.org/10.1007/978-3-031-14678-7_6
63. Schäferle, S. Choosing the correct sensor model for your application. IPG Automotive 2019. Available online: https://www.ipg-automotive.com/uploads/tx_pbfaqtickets/files/98/SensorModelLevels.pdf (Accessed February, 6 2025)
64. Magosi, Z.F.; Wellershaus, C.; Tihanyi, V.R.; Luley, P.; Eichberger, A. Evaluation Methodology for Physical Radar Perception Sensor Models Based on On-Road Measurements for the Testing and Validation of Automated Driving. *Energies* **2022**, *15*, 2545. <https://doi.org/10.3390/en15072545>
65. *Reference Manual Version 12.0.1 CarMaker*, IPG Automotive, 2023.
66. Iclodean, C. *Introducere în sistemele autovehiculelor*, Publisher: Risoprint, Romania, 2023.
67. Renard, D.; Saddem, R.; Annebique, D.; Riera, B. From Sensors to Digital Twins toward an Iterative Approach for Existing Manufacturing Systems. *Sensors* **2024**, *24*, 1434. <https://doi.org/10.3390/s24051434>
68. Brucherseifer, E.; Winter, H.; Mentges, A.; Mühlhäuser, M.; Hellmann, M. Digital Twin conceptual framework for improving critical infrastructure resilience. *at-Automatisierungstechnik* **2021**, *69*(12), 1062-1080. <https://doi.org/10.1515/auto-2021-0104>
69. Grieves, M.; Vickers, J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In Transdisciplinary perspectives on complex systems: New findings and approaches, Publisher: Springer, pp. 85-113. https://doi.org/10.1007/978-3-319-38756-7_4
70. Kritzinger, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline* **2018**, *51*(11), 1016-1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>
71. Shoukat, M.U.; Yan, L.; Yan, Y.; Zhang, F.; Zhai, Y.; Han, P.; Nawaz, S.A.; Raza, M.A.; Akbar, M.W.; Hussain, A. Autonomous driving test system under hybrid reality: The role of digital twin technology. *Internet Things* **2024**, *27*, 101301. <https://doi.org/10.1016/j.iot.2024.101301>
72. Iclodean, C.; Cordos, N.; Varga, B.O. Autonomous Shuttle Bus for Public Transportation: A Review. *Energies* **2020**, *13*, 2917. <https://doi.org/10.3390/en13112917>
73. Navya - Brochure-Autonom-Shuttle-Evo. Available online: <https://navya.tech/wp-content/uploads/documents/Brochure-Autonom-Shuttle-Evo-EN.pdf> (Accessed February, 6 2025)
74. Navya - Self-Driving Shuttle for Passenger Transportation. Available online: <https://www.navya.tech/en/solutions/moving-people/self-driving-shuttle-for-passenger-transportation/> (Accessed February, 6 2025)
75. Patentimages. Available online: <https://patentimages.storage.googleapis.com/12/0f/d1/33f8d2096f49f6/US20180095473A1.pdf> (Accessed February, 6 2025)
76. AVENUE Autonomous Vehicles to Evolve to a New Urban Experience Report. Available online: https://h2020-avenue.eu/wp-content/uploads/2023/03/Keolis-LyonH2020-AVENUE_Deliverable_7.6_V2-not-approved.pdf (Accessed February, 6 2025)
77. EarthData Search. Available online: <https://search.earthdata.nasa.gov/search?q=SRTM> (Accessed February, 6 2025)
78. GpsPrune. Available online: <https://activityworkshop.net/software/gpsprune/download.html> (Accessed February, 6 2025)
79. *InfoFile Description Version 12.0.1 IPGRoad*, IPG Automotive, 2023.

80. *User Manual Version 12.0.1 IPGDriver*, IPG Automotive, 2023.
81. Piyabongkarn, D.N.; Rajamani, R.; Grogg, J.A.; Lew, J.Y. Development and Experimental Evaluation of a Slip Angle Estimator for Vehicle Stability Control. *IEEE Trans. Control. Syst. Technol.* **2009**, *17*, 78-88. doi:10.1109/TCST.2008.922503
82. *CarMaker Reference Manual 12.0.2 CarMaker*, IPG Automotive, 2023.
83. Pacejka, H.B. *Tyre and Vehicle Dynamics*. 2nd Edition. Publisher: Elsevier's Science and Technology, 2006.
84. Salminen, H. Parametrizing tyre wear using a brush tyre model. Master Thesis, Royal Institute of Technology, Stockholm, Sweden, 15 December 2014. <https://kth.diva-portal.org/smash/get/diva2:802101/FULLTEXT01.pdf>
85. Pacjka, H.B.; Besselink, I.J.M. Magic Formula Tyre Model with Transient Properties. *Veh. Syst. Dyn.* **1997**, *27*(sup001), 234-249. <https://doi.org/10.1080/00423119708969658>
86. Pacejka, H.B. Chapter 4 - Semi-Empirical Tire Models. In *Tire and Vehicle Dynamics (Third Edition)*; Editor Pacejka, H.B.; Butterworth-Heinemann, 2012, pp. 149-209. <https://doi.org/10.1016/B978-0-08-097016-5.00004-8>
87. Guo, Q.; Xu, Z.; Wu, Q.; Duan, J. The Application of in-the-Loop Design Method for Controller. In 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, China, 23-25 May 2007, pp. 78-81, doi: 10.1109/ICIEA.2007.4318373
88. Chen, T.; Chen, L.; Xu, X.; Cai, Y.; Jiang, H.; Sun, X. Sideslip Angle Fusion Estimation Method of an Autonomous Electric Vehicle Based on Robust Cubature Kalman Filter with Redundant Measurement Information. *World Electr. Veh. J.* **2019**, *10*, 34. <https://doi.org/10.3390/wevj10020034>
89. Jin, L.; Xie, X.; Shen, C.; Wang, F.; Wang, F.; Ji, S.; Guan, X.; Xu, J. Study on electronic stability program control strategy based on the fuzzy logical and genetic optimization method. *Adv. Mech. Eng.* **2017**, *9*(5), 1-13. <https://doi.org/10.1177/1687814017699351>
90. Zhao, Z.; Chen, H.; Yang, J.; Wu, X.; Yu, Z. Estimation of the vehicle speed in the driving mode for a hybrid electric car based on an unscented Kalman filter. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2014**, *229*(4), 437-456. <https://doi.org/10.1177/0954407014546918>
91. Li, Q.; Chen, L.; Li, M.; Shaw, S.-L.; Nuchter, A. A Sensor-Fusion Drivable-Region and Lane-Detection System for Auto-nomous Vehicle Navigation in Challenging Road Scenarios. *IEEE Trans. Veh. Technol.* **2013**, *63*(2), 540-555. doi:10.1109/TVT.2013.2281199
92. Rana, M.M. Attack Resilient Wireless Sensor Networks for Smart Electric Vehicles. *IEEE Sens. Lett.* **2017**, *1*(2), 5500204. doi: 10.1109/LSSENS.2017.2690907
93. Xia, X.; Xiong, L.; Huang, Y.; Lu, Y.; Gao, L.; Xu, N.; Yu, Z. Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors. *Mech. Syst. Signal Process.* **2022**, *162*, 107993. <https://doi.org/10.1016/j.ymssp.2021.107993>
94. Sieberg, P.M.; Schramm, D. Ensuring the Reliability of Virtual Sensors Based on Artificial Intelligence within Vehicle Dynamics Control Systems. *Sensors* **2022**, *22*, 3513. <https://doi.org/10.3390/s22093513>
95. Xiong, L.; Xia, X.; Lu, Y.; Liu, W.; Gao, L.; Song, S.; Han, Y.; Yu, Z. IMU-Based Automated Vehicle Slip Angle and Attitude Estimation Aided by Vehicle Dynamics. *Sensors* **2019**, *19*, 1930. <https://doi.org/10.3390/s19081930>
96. Ess, A.; Schindler, K.; Leibe, B.; Van Gool, L. Object detection and tracking for autonomous navigation in dynamic environments. *Int. J. Robot. Res.* **2010**, *29*, 1707-1725. <https://doi.org/10.1177/0278364910365417>
97. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 5-28 September 2016. doi:10.1109/ICIP.2016.7533003
98. Banerjee, S.; Serra, J.G.; Chopp, H.H.; Cossairt, O.; Katsaggelos, A.K. An Adaptive Video Acquisition Scheme for Object Tracking. In 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, 02-06 September 2019. doi: 10.23919/EUSIPCO.2019.8902829
99. Ning, C.; Menglu, L.; Hao, Y.; Xueping, S.; Yunhong, L. Survey of pedestrian detection with occlusion. *Complex Intell. Syst.* **2021**, *7*, 577-587. <https://doi.org/10.1007/s40747-020-00206-8>

100. Liu, Z.; Chen, W.; Wu, X. Salient region detection using high level feature. In 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 10-12 December 2014. doi: 10.1109/ICARCV.2014.7064488
101. Felzenszwalb, P.; Girshick, R.; McAllester, D.; Ramanan, D. Visual object detection with deformable part models. *Commun. ACM.* **2013**, *56*(9), 97-105. <https://doi.org/10.1145/2494532>
102. Kato, S.; Takeuchi, E.; Ishiguro, Y.; Ninomiya, Y.; Takeda, K.; Hamada, T. An Open Approach to Autonomous Vehicles. *IEEE Micro* **2015**, *35*(6), 60-68 doi:10.1109/MM.2015.133
103. Broggi, A.; Cattani, S.; Patander, M.; Sabbatelli, M.; Zani, P. A full- 3D voxel-based dynamic obstacle detection for urban scenario using stereo vision. In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, Netherlands, 06-09 October 2013, pp. 71-76. doi:10.1109/ITSC.2013.6728213
104. Patra, S.; Maheshwari, P.; Yadav, S.; Arora, C.; Banerjee, S. A Joint 3D-2D based Method for Free Space Detection on Roads. arXiv:1711.02144 2018, <https://doi.org/10.48550/arXiv.1711.02144>
105. Vitor, G.B.; Lima, D.A.; Victorino, A.C.; Ferreira, J.V. A 2D/3D vision based approach applied to road detection in urban environments. In IEEE Intelligent Vehicles Symposium (IV 2013), Australia, Jun 2013, pp.952-957.
106. Heinz, L. *CarMaker Tips & Tricks No. 3-011 Detect Traffic Lights*. IPG Automotive, 2019. Available online: https://www.ipg-automotive.com/uploads/tx_pbfaqtickets/files/100/DetectTrafficLights.pdf (Accessed February, 6 2025)
107. Zhang, P.; Zhang, M.; Liu, J. Real-time HD map change detection for crowdsourcing update based on mid-to-high-end sensors. *Sensors* **2021**, *21*, 2477. <https://doi.org/10.3390/s21072477>
108. Bahlmann, C.; Zhu, Y.; Ramesh, V.; Pellkofer, M.; Koehler, T. A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information. In IEEE Proceedings of Intelligent Vehicles Symposium, Las Vegas, 6-8 June 2005, 255-260. <https://doi.org/10.1109/IVS.2005.1505111>
109. Fazekas, Z.; Gerencsér, L.; Gáspár, P. Detecting Change between Urban Road Environments along a Route Based on Static Road Object Occurrences. *Appl. Sci.* **2021**, *11*, 3666. <https://doi.org/10.3390/app11083666>
110. Liu, C.; Tao, Y.; Liang, J.; Li, K.; Chen, Y. Object detection based on YOLO network. In Proceedings of the 2018 IEEE 4th In-formation Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14-16 December 2018, pp. 799-803. doi: 10.1109/ITOEC.2018.8740604
111. Nuthong, C.; Charoenpong, T. Lane Detection using Smoothing Spline. In 3rd International Congress on Image and Signal Processing, Yantai, China, 16-18 October 2010, pp. 989-993. doi:10.1109/CISP.2010.5646935
112. Dou, J.; Li, J. Robust object detection based on deformable part model and improved scale invariant feature transform. *Optik* **2013**, *124*(24), 6485-6492. <https://doi.org/10.1016/j.ijleo.2013.03.174>
113. Lindenmaier, L.; Aradi, S.; Bécsi, T.; Törő, O.; Gáspár, P. Object-Level Data-Driven Sensor Simulation for Automotive Environment Perception. *IEEE Trans. Intell. Veh.* **2023**, *8*(10), 4341-4356. doi: 10.1109/TIV.2023.3287278
114. Bird, J.; Bird, J. *Higher Engineering Mathematics*, 5th edition; London, Routledge, 2006. <https://doi.org/10.4324/9780080470955>
115. Ainsalu, J.; Arffman, V.; Bellone, M.; Ellner, M.; Haapamäki, T.; Haavisto, N.; Josefson, E.; Ismailogullari, A.; Lee, B.; Ma-dland, O.; et al. State of the Art of Automated Buses. *Sustainability* **2018**, *10*, 3118. <https://doi.org/10.3390/su10093118>
116. Lian, H.; Li, M.; Li, T.; Zhang, Y.; Shi, Y.; Fan, Y.; Yang, W.; Jiang, H.; Zhou, P.; Wu, H. Vehicle speed measurement method using monocular cameras. *Sci. Rep.* **2025**, *15*, 2755 <https://doi.org/10.1038/s41598-025-87077-6>
117. Vivacqua, R.; Vassallo, R.; Martins, F. A Low Cost Sensors Approach for Accurate Vehicle Localization and Autonomous Driving Application. *Sensors* **2017**, *17*, 2359. <https://doi.org/10.3390/s17102359>
118. Xue, L.; Li, M.; Fan, L.; Sun, A.; Gao, T. Monocular Vision Ranging and Camera Focal Length Calibration. *Sci. Program.* **2021**, *2021*, 979111. <https://doi.org/10.1155/2021/979111>
119. Rosique, F.; Navarro, P.J.; Fernández, C.; Padilla, A. A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research. *Sensors* **2019**, *19*, 648. <https://doi.org/10.3390/s19030648>

120. Elster, L.; Staab, J.P.; Peters, S. Making Automotive Radar Sensor Validation Measurements Comparable. *Appl. Sci.* **2023**, *13*, 11405. <https://doi.org/10.3390/app132011405>
121. Roy, C.J.; Balch, M.S. A Holistic Approach to Uncertainty Quantification with Application to Supersonic Nozzle Thrust. *Int. J. Uncertain. Quantif.* **2021**, *2*, 363–381. <http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2012003562>
122. Magosi, Z.F.; Eichberger, A. A Novel Approach for Simulation of Automotive Radar Sensors Designed for Systematic Support of Vehicle Development. *Sensors* **2023**, *23*, 3227. <https://doi.org/10.3390/s23063227>
123. Maier, M.; Makkapati, V. P.; Horn, M. Adapting Phong into a Simulation for Stimulation of Automotive Radar Sensors. In 2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Munich, Germany, 15–17 April 2018, pp. 1–4. doi:10.1109/ICMIM.2018.8443493
124. Minin, I.V.; Minin, O.V. Lens Candidates to Antenna Array. In: *Basic Principles of Fresnel Antenna Arrays. Lecture Notes Electrical Engineering*, Springer, Berlin, Heidelberg, 2008; Volume 19, pp. 71–127. https://doi.org/10.1007/978-3-540-79559-9_2
125. Sensors Partners. Available online: LiDAR laser: what is LiDAR and how does it work? | Sensor Partners (Accessed March, 6 2025)
126. García-Gómez, P.; Royo, S.; Rodrigo, N.; Casas, J.R. Geometric Model and Calibration Method for a Solid-State LiDAR. *Sensors* **2020**, *20*(10), 2898; <https://doi.org/10.3390/s20102898>
127. Kim, G. Performance Index for Extrinsic Calibration of LiDAR and Motion Sensor for Mapping and Localization. *Sensors* **2022**, *22*, 106. <https://doi.org/10.3390/s22010106>
128. Schmoll, L.; Kemper, H.; Hagenmüller, S.; Brown, C.L. Validation of an Ultrasonic Sensor Model for Application in a Simulation Platform. *ATZelectronics worldwide* **2024**, *19*(5), 8–13. <https://link.springer.com/content/pdf/10.1007/s38314-024-1853-5.pdf>
129. Sen, S.; Husom, E.J.; Goknil, A.; Tverdal, S.; Nguyen, P. Uncertainty-Aware Virtual Sensors for Cyber-Physical Systems. *IEEE Software* **2024**, *41*, 77–87. doi:10.1109/MS.2023.3306873
130. Ying, Z.; Wang, Y.; He, Y.; Wang, J. Virtual Sensing Techniques for Nonlinear Dynamic Processes Using Weighted Probability Dynamic Dual-Latent Variable Model and Its Industrial Applications. *Knowl.-Based Syst.* **2022**, *235*, 107642. <https://doi.org/10.1016/j.knosys.2021.107642>
131. Yuan, X.; Rao, J.; Wang, Y.; Ye, L.; Wang, K. Virtual Sensor Modeling for Nonlinear Dynamic Processes Based on Local Weighted PSFA. *IEEE Sens. J.* **2022**, *22*, 20655–20664. doi:10.1109/JSEN.2022.3196011
132. Zheng, T. Algorithmic Sensing: A Joint Sensing and Learning Perspective. In Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services; Association for Computing Machinery: New York, NY, USA, June 18 2023, pp. 624–626. <https://doi.org/10.1145/3581791.3597509>
133. Es-haghi, M.S.; Anitescu, C.; Rabczuk, T. Methods for Enabling Real-Time Analysis in Digital Twins: A Literature Review. *Comput. Struct.* **2024**, *297*, 107342. <https://doi.org/10.1016/j.compstruc.2024.107342>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.