

Received January 5, 2022, accepted January 29, 2022, date of publication February 9, 2022, date of current version February 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150284

# Validation Framework for Generic Radar Sensor Models

MICHAL JASINSKI<sup>1,2</sup>, (Graduate Student Member, IEEE),

PAWEL SKRUCH<sup>1</sup>, (Senior Member, IEEE), AND

MATEUSZ KOMORKIEWICZ<sup>2</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Automatic Control and Robotics, AGH University of Science and Technology, 30-059 Krakow, Poland

<sup>2</sup>APTIV Services Poland S.A., 30-399 Krakow, Poland

Corresponding author: Michal Jasinski (jasinski@agh.edu.pl)

This work was supported by the Polish Ministry of Science and Higher Education under Project 0014/DW/2018/02 and carried out in cooperation with Aptiv Services Poland S.A.—Technical Center Krakow and AGH University of Science and Technology—Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering.

**ABSTRACT** Automotive radar sensors are vital in Advanced Driver Assistance Systems (ADAS). To be more precise, their ability to explicitly measure the relative velocity to its targets is essential in Adaptive Cruise Control (ACC) and Emergency Braking (EB) applications. Nevertheless, ADAS are getting more and more complex, due to constantly increasing demands regarding safety and performance. As a result, to speed up the development and validation time of ADAS, part of the testing is performed in simulations. Replacing some of the test drives by the runs in virtual environments not only reduces the cost of a product, but also helps in fully safe execution of dangerous corner cases. However, to enable reliable testing of radar-based ADAS in virtual environments, high-fidelity radar sensor models are required. In order to prove the reliability of a given model, a proper evaluation process has to be conducted. This paper presents an end-to-end, straightforward methodology for performance assessment and fine-tuning of radar sensor models. To show how the full pipeline of the framework can be executed, an exemplary radar sensor model has been incorporated. The successful fine-tuning of the model proves the usefulness of the introduced method.

**INDEX TERMS** Mathematical modelling, sensor modelling, simulation, radars, virtual validation.

## I. INTRODUCTION

In recent years the automotive industry has focused on the intensive development of Advanced Driver Assistance Systems (ADAS), e.g. Emergency Braking (EB) or Adaptive Cruise Control (ACC), that support a driver in dangerous situations and, if necessary, take the control of a car. Usually, such applications have the following parts: sensing, object detection and decision making. Sensing and object detection modules are oftentimes coupled, forming an automotive perception system. The perception module has two main roles. First, to gather measurements from a given sensor, such as camera, lidar or radar. Then, based on raw sensor measurements, to fit an oriented bounding box (OBB) to each object that is visible from the sensor perspective. A list of OBBs generated in a given scene is called object list. It is worth mentioning, that an automotive perception system can also use multi-sensor data as an input. In that case, a high-level

fusion of outputs from multiple object detectors needs to be performed.

Taking into account that complexity of ADAS is rapidly growing, an immense effort must be made in order to prove the robustness of a given system. Moreover, potential multi-sensor input makes the system and its validation even more complicated. To properly assess the reliability of an ADAS in all possible road conditions and scenarios, a million miles of test drives has to be conducted [1]. Performing such tests entirely on the real road might be too time-consuming and also expensive. Therefore, to reduce the cost of a product, virtual environments are incorporated. It is mainly due to the fact, that a high-fidelity simulation allows to verify the system behaviour in any given road scenario (and under different weather conditions) in a relatively short period of time. Also, the reliability of the product itself can be increased, since dangerous corner cases can be safely reproduced within a virtual environment. The process of testing a real system in a simulation is usually called the virtual validation.

The associate editor coordinating the review of this manuscript and approving it for publication was Sandra Costanzo.

In order to make the virtual validation reliable and robust, the realism of a simulation should be proved. In other words, 3D scene rendered by a given simulator have to precisely emulate a real scenario, including road profile, static objects, objects in move and weather conditions. However, a photo-realistic simulation itself is not enough to reliably test an ADAS, e.g. ACC. To make the simulation robust, a proper input to the ADAS, written in the same format as in the production setup, has to be provided. In other words, sensor models precisely imitating raw measurements of a real sensor have to be incorporated into the virtual environment.

One of the key sensors that has to be considered within the simulation is radar. Its measurements are sparse and noisy compared to data provided by camera or lidar sensors. On the other hand, an explicit measure of the relative velocity of a target is provided, which is vital in applications like EB or ACC. Also, weather conditions have a little influence on radar performance. Unfortunately, free to use, state-of-the-art simulation environments, e.g. [2], [3], lack accurate radar sensor models [4].

Radar modelling task is challenging, due to the amount of effects that highly influence its measurements, like micro Doppler effect caused by rotating wheels or multi-path propagation of the electromagnetic wave [5]. Nevertheless, radar sensor modelling topic can be handled on many abstraction layers. According to [6], three main approaches can be defined based on state-of-the-art research papers: black-box models [7]–[9], that are usually based on deep neural networks, physical models explicitly simulating all radar effects and components [10]–[13] and generic models that combine deterministic geometrical models with the noise injection in order to derive radar detections [14]–[16]. The choice of the approach depends on the application. However, taking into account the ratio of modeling and validation time to the expected quality, a well-designed generic radar sensor model (GRSM) can be treated (in most cases) as a good compromise. For instance, there is no need to gather huge amounts of training data that is a prerequisite for a neural-network-based solution. Also, the number of components to be emulated is significantly lower compared to physical models.

According to the best knowledge of the authors, robust radar sensor model being able to thoroughly reproduce radar measurements in real-time still does not exist, regardless of the virtual scene complexity. However, a novel GRSM introduced in [16] can be treated as a good starting point for further work, based on the results presented by the authors. Nevertheless, before improving a baseline model, there is a requirement to introduce some performance metrics, in order to compare subsequent sensor model versions.

As shown in a comprehensive survey [17], evaluation of simulation models is still an open research area. What is more, the verification method strictly depends on an application. The novelty of this paper is based on an introduction of an end-to-end methodology for GRSMs evaluation. The method provides a straightforward fidelity assessment of a

given GRSM using a set of simple metrics. As a result, new extensions added to the baseline model can be easily verified, by checking if the overall sensor model performance has improved. Also, thanks to the proposed method, fine-tuning of a given model is easily achievable.

In order to prove the usefulness of the approach and to show how the full pipeline of a performance assessment can be carried out, the proposed validation framework has been explicitly employed in the fine-tuning process of the GRSM introduced in [16]. The obtained results clearly show that the optimized model outperforms the one with the original settings. The straightforward validation methodology together with the successful fine-tuning of the GRSM is the main contribution of that work.

The structure of this paper is organized as follows. In Section II, a background of radar sensor modelling is presented. Section III shortly describes the incorporated GRSM. In Section IV the details of the proposed validation framework are depicted. Section V comprehensively explains the evaluation process and the results of the sensor model fine-tuning procedure.

## II. RADAR SENSOR MODELLING APPROACHES

As mentioned in [6], three main approaches for radar sensor modelling can be found in the literature. This section presents basics of automotive radar sensors operation and gives an overview of the radar modelling techniques.

### A. RADAR BACKGROUND

Radar (radio detection and ranging) is a sensor that transmits a set of electromagnetic waves, which bounce off from various objects and static scenery. The electromagnetic echo is captured by radar antennas and transformed from the time to frequency domain using a digital signal processing unit. Then, given the transformed signal, radar signal processing algorithms extract the desired information from the clutter. There are several radar sensor types but usually Frequency-Modulated Continuous-Wave (FMCW) radars are employed in automotive industry. FMCW systems are able to simultaneously measure radial distance to a given target and its relative velocity w.r.t. the host vehicle. Additionally, FMCW radars oftentimes use multiple transmitting and receiving antennas. Thanks to an array of antennas, a measure of an angle in the horizontal plane is enabled, calculated w.r.t. the radar boresight. It is worth mentioning, that range, relative velocity and angle to a point on a given object form a single radar detection. As a consequence, considering these three measurement sources, an FMCW radar can return multiple detections per single target (e.g. car). A result of a single scan through the radar field-of-view is then a point cloud of detections. Please note that in some automotive applications a radar detection can explicitly refer to an outcome of an object detection algorithm (a so-called tracking process), that is used to find an OBB for each object in the radar field-of-view, based on the raw point cloud. Nevertheless, in this paper a tracking part is separated from an FMCW scanning procedure

and its product is called object list rather than detection list.

Unfortunately, raw radar data (point cloud) obtained from the radar signal processing chain is sparse and still contain huge amount of ambiguities and noise. Therefore, a radar sensor model should take this fact into account and generate a synthetic point cloud of a high fidelity. What is more, in order to enable the virtual validation of ADAS, the sensor model integration with a given virtual environment is needed, to make sure that the synthetic measurements are calculated in real-time and explicitly based on simulation data. Additionally, a radar sensor model must have a configurable mounting position and orientation, to support cases where multiple radar sensors are mounted on a vehicle equipped with an ADAS. An example way of handling this, is to define mounting positions and orientations of each radar sensor (and its corresponding model in the virtual environment) w.r.t. to a fixed reference coordinate system. As a result, even though each radar sensor returns a point cloud w.r.t. its own coordinate system, transformations to the reference coordinate system is possible, explicitly defined by the mounting offsets. Despite the fact that an ultimate automotive simulator does not exist yet [4], several tools for virtual validation are available on the market, e.g. [2] or [3].

### B. BLACK-BOX RADAR MODELLING

Considering current advances in architectures and training processes of neural network algorithms, there is also a possibility to incorporate artificial intelligence in the radar modelling area. In other words, it is feasible to find a mapping from a simulation-based data to synthetic radar detections, based on training data. That approach is called black-box modelling, since there is no explicit implementation of the radar model components. Several black-box radar models have already been introduced in the literature [7]–[9] with satisfactory results. However, taking into account the complexity of the radar systems, development of a robust and scene-independent black-box radar model would require an immense amount of labelled data from wide range of road scenarios. Unfortunately, data gathering and labelling processes are both time-consuming and expensive. Therefore, a more generic approach should be used for the radar sensor modelling. In particular, there is a need of the model being more independent from real data, so that it can be used in virtual validation of ADAS in case real sensor suite is not yet ready for test drives campaigns.

### C. DETAILED RADAR MODELLING

An opposite way to create a radar sensor model is to explicitly simulate all components of a given radar system from the electromagnetic wave propagation to signal processing chain. Such models are usually called physical models or white-box models. The development of a white-box model itself is data-independent and only small amount of data is required to fine-tune a given model. That is a significant advantage compared

to black-box modelling, where a labelled training dataset is a prerequisite.

Physical models are usually based on ray-tracing algorithms [10]–[12]. As shown in [12] a real-time implementation of a physical model in a simulation environment is possible. However, a reliable emulation of the electromagnetic wave propagation using simulation data and other radar components require a bulk of computational resources. What is more, due to the fact that all components of an automotive radar system have to be explicitly emulated, the development of a white-box model requires huge expertise in radar systems. In general, as a consequence, it is difficult and time-consuming to implement and fine-tune a real-time capable white-box radar sensor model.

### D. GENERIC RADAR MODELLING

To simplify the development time of the radar sensor model an alternative approach can be used. The goal is to derive a set of synthetic radar detections (a point cloud) explicitly using high-level simulation data - without detailed modelling of the radar physics. To achieve that, a scattering-centers concept known from radar cross-section studies can be used. It states that the electromagnetic scattering from an electrically large target can be approximated with a sparse set of points - called scattering-centers (SCs) - located on that target [14]. Each point (a single SC) corresponds to a single radar detection, represented by range, relative velocity and angle values. In other words, instead of emulating directly the propagation of the electromagnetic wave together with the signal processing chain, synthetic radar detections can be obtained by generating SCs from a high-level simulation data (so-called ground truth, i.e. set of bounding boxes). Unfortunately, raw synthetic detections (SCs) do not encapsulate physics of a given radar sensor. Therefore, a noise model has to be additionally incorporated, to ensure that the synthetic point cloud has similar stochastic properties to the real sensor measurements.

In such a manner, generation of SCs is sensor-independent. Only noise model has to be fine-tuned to real measurements. As a result, a radar model developed using this approach can be called generic radar sensor model (GRSM). GRSMs can be easily used in rigorous hardware-in-the-loop (HIL) simulations of ADAS, due to their relatively low computational complexity. Also, they do not require large, labelled datasets as black-box models do. A well-designed white-box model will be in general more accurate than a GRSM. However, considering the development time and computational complexity, a GRSM can be sufficient for HIL applications, if only its noise model is well-designed.

### III. GENERIC RADAR SENSOR MODEL

The term generic radar sensor model (GRSM) introduced in the previous section and used further for the need of this work applies to any radar sensor model that is not based on neural networks and that mimics the behaviour of a real sensor without a detailed modelling of its physics. The goal of

a GRSM is to derive a set of synthetic radar detections (set of range, relative velocity and angle tripples) of a high-fidelity. The simulated point cloud has to be acquired using as low computational resources as possible in order to meet strong real-time requirements of the HIL simulations of ADAS.

To conduct a fine-tuning procedure of a GRSM using the validation framework, a novel GRSM introduced in [16] has been incorporated. The model is based on Open Simulation Interface (OSI) data format [18] and consists of two main parts:

- Deterministic Geometrical Model that generates set of SCs (raw synthetic detections) based on data provided by OSI,
- Stochastic Model that injects noise over time to raw synthetic data.

In this section, a brief description of OSI data format and details about the used GRSM are presented.

### A. OPEN SIMULATION INTERFACE DATA FORMAT

OSI defines a high-level abstraction layer for data exchange in virtual environments for virtual validation purposes. To be more precise, a well-organized structures are defined, where high-level simulation data is stored. In OSI-based simulation, each sensor model is fed with data stored in an *OSI::SensorView* structure. That is a so-called ground-truth that contains information about all objects in a given scene, where vehicles and static scenery are represented by oriented bounding boxes (OBBs). Each object in *OSI::SensorView* is defined w.r.t. a global coordinate system of a given simulator. The output from each sensor model executed in a given simulator is then written in an *OSI::SensorData* structure, which stores synthetic sensor measurements, defined w.r.t. a sensor coordinate system. It is worth mentioning that OSI, apart from OBBs and synthetic detections, defines structures for storing simulation metadata, such as: object classification, environmental conditions, lanes description, lane assignment, etc.

### B. DETERMINISTIC GEOMETRICAL MODEL

The goal of the geometrical model is to generate a set of scattering-centers (SCs). Currently the geometrical model generates detections only for vehicles - namely, for each vehicle type defined in the *OSI::VehicleClassification* structure.

SC point cloud play a role of a spatial simulation of the electromagnetic wave scattering. Therefore, the location of points on a given object have to be accurately aligned with the places, from which the electromagnetic wave is most often scattered back to the sensor. As a result, each object type within the model has a unique set of SCs that take into account the shape and curvatures of that object.

To handle this properly, a separate 3D vehicle model is generated for each OSI vehicle type defined in the *OSI::VehicleClassification* structure [18]. Each model contains a set of interconnected triangular polygons, scaled to the default dimensions of the given type. The example sets of polygons generated for *OSI::TYPE\_SMALL\_CAR* and

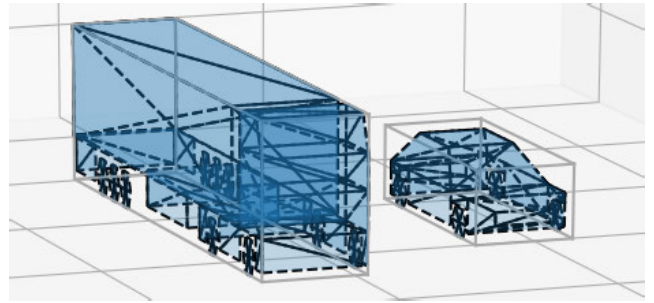


FIGURE 1. Examples of vehicle models generated within the sensor model.

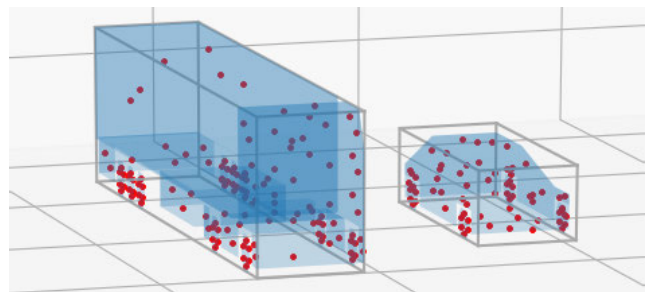


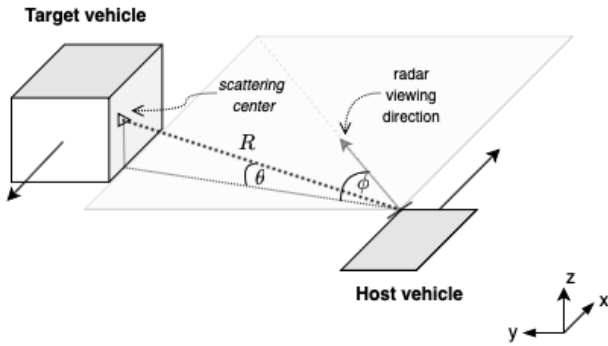
FIGURE 2. Scattering-centers generated for small car and truck with semitrailer.

*OSI::TYPE\_SEMITRAILER* (truck with semitrailer) are presented in Figure 1.

Also, in order to properly generate detections the following metadata must be defined for each polygon: its middle position, normal vector, area and material type. The middle position of a given polygon explicitly represents a single SC. The SCs corresponding to the polygons of *OSI::TYPE\_SMALL\_CAR* and *OSI::TYPE\_SEMITRAILER* are shown in Figure 2.

To sum up, using the polygons generated for all OSI vehicle types, the following procedure represents deterministic part of the GRSM:

- 1) For each vehicle, rendered at the given time moment by a simulation environment like [2], the information about its OBB and type is extracted. Data collected from all objects form a ground truth message, which is stored in a *OSI::SensorView* message.
- 2) Then, for each vehicle in the ground truth, the triangular polygons corresponding to the type of that vehicle are scaled to the dimensions of the OBB.
- 3) In the next step, range  $r$ , azimuth  $\phi$  and elevation  $\theta$  are calculated separately for all of the generated polygons. It is worth mentioning that the calculations are performed w.r.t. the sensor coordinate systems (SCS), defined by the viewing angle of the sensor model. This case is presented in Figure 3.
- 4) Also, for each SC, the relative velocity along the viewing angle is calculated.
- 5) Finally, using  $(r, \phi, \theta, v)$  gathered from all the vehicles available in the ground truth, the field-of-view and



**FIGURE 3.** Radar detection definition w.r.t. the viewing angle for a given scattering center.

occlusions filters are applied, in order to get rid of the detections (SCs) that are not visible from the sensor model perspective. In other words, each point that is not covered by the sensor range or is shadowed by an obstacle is deleted and not acknowledged as detection. The remaining elements form a set of raw sensor model detections.

It is worth noting down, that even though the current GRSM version is able to generate point cloud of detections only for vehicles, the model can be easily extended to support also other targets detectable by radar sensors, e.g. guardrails, bicycles, poles, etc. In order to achieve this it is necessary to build a 3D geometrical model (consisting of triangular polygons) for a given target. Also, the high-level data of this object has to be extracted from the simulation environment and stored properly in an OSI message. In other words, if the SCs are provided for an object, the GRSM is able to generate a set of detections based on them.

### C. STOCHASTIC MODEL

Taking into account the stochastic properties of the real system in sensor model detections, the obtained deterministic measurements have to be modified. First, a radar-specific noise is injected to range, azimuth and elevation estimates. The noise is a normally distributed random variable with mean equal zero and variance defined by using both radar system specification and the estimated signal-to-noise ratio [16]. Also, to mimic the sparsity of the real data, a statistic test is executed separately for each SC that removes some of the elements from the raw detection list. In particular, the  $i_{th}$  SC is acknowledged as a detection only when the measured signal  $x_i$ , calculated from that SC, exceeds the current threshold  $th_i$  [16]:

$$x_i > th_i \quad (1)$$

$$x_i = \sqrt{P_{si}} + N(0, \sigma_{noise}^2) \quad (2)$$

$$th_i = \sqrt{2\sigma_{noise}^2} \operatorname{erf}^{-1}(1 - 2P_{FA}) \quad (3)$$

where:

- $P_{si}$  represents an estimation of the received signal power amplified by the radar system gain,

- $N(0, \sigma_{noise}^2)$  is the probability distribution of the system noise - normal random variable with zero mean and variance equal to  $\sigma_{noise}^2$ ,
- $\operatorname{erf}^{-1}$  stands for the inversion of the error function that depends on the probability of false alarm  $P_{FA}$ .

The values of all the parameters defined above highly influence the fidelity of the sensor model point cloud, e.g. its sparsity and the distribution of points. Therefore, in order to increase the plausibility of the sensor model output as much as possible, a proper fine-tuning algorithm has to be executed. The role of that procedure is to modify the values of the parameters (selected initially from the radar system specification) using real radar measurements as a reference. The procedure has to be executed until the score from a given set of metrics meets the desired criteria. In particular, in the used GRSM the number and the distribution of points depend on both  $P_{si}$  and  $th_i$ . In each  $P_{si}$  estimation, the structure of a given target is reflected, since the  $P_{si}$  value is mostly based on the radar cross section value that is calculated using area, material and an angle to the radar boresight of the  $i_{th}$  SC. However, it turns out that both  $P_{si}$  and  $th_i$  are affected by the value of the noise figure parameter  $F_n$  [dB]. As a result, the location and the sparsity of SCs can be explicitly controlled via the  $F_n$  parameter, so that the resulted point cloud accurately emulates real measurements.

### IV. VALIDATION FRAMEWORK

Development and testing of ADAS in simulation environments (virtual validation) requires accurate sensor models emulating a given automotive perception system. However, in order to properly assess a sensor model performance, synthetic data has to be compared to data from sensor logs, i.e. synthetic detections returned from a GRSM have to be set against detections from a real radar sensor.

To answer this problem, a straightforward framework is introduced in this paper. Using that methodology, a given GRSM can be explicitly fine-tuned based on real measurements. As a consequence, the fidelity of the GRSM itself can be easily boosted - thus increasing the reliability of the whole simulation environment. The proposed methodology extends the idea presented in [6]. The concept adopts the property of automotive radar systems, where additionally an object list (set of OBBs) generated on top of a point cloud of detections is provided. The assumption of the method is high-fidelity radar detections (point cloud) result in realistic object list. Therefore, a GRSM performance evaluation metrics can be explicitly defined based on the comparison of the real radar object list to the object list generated using simulated detections.

To be more precise, a set of object-based metrics are defined that return an evaluation score of a GRSM. On top of this it is possible to explicitly compare detections point clouds as shown in the original GRSM paper [16]. In such a manner, object-based comparison can be treated as an additional source of information in the sensor model fine-tuning process.

**A. COORDINATE SYSTEMS DEFINITION**

In order to compare real and synthetic data, sensor measurements need to be gathered. In particular two sources of data are essential:

- Object list from a radar object detector for calculating object-based metrics,
- OSI-based labels obtained from a reference sensor to generate a GRSM detections.

However, in order to reliably collect the data, mounting positions for all radar sensors have to be properly selected on a test vehicle. Most importantly, it is necessary to choose a reference coordinate system. In this case a vehicle coordinate system (VCS) is selected. VCS is defined as a right-handed Cartesian coordinate system, with x-axis pointing to the front, y-axis pointing to the left and z-axis pointing upwards. Its origin is set in the middle of the front bumper of the test vehicle, in the ground level. Then, using VCS, all sensor mounting positions and orientations can be defined. In other words, each sensor provides its measurements w.r.t. its sensor coordinate system (SCS). Nevertheless, an SCS origin is defined w.r.t. VCS, in the sensor mounting position and according to the sensor orientation.

Different radar sensors setups are used in vehicles equipped with ADAS. In that case it is assumed that a vehicle has four corner radar sensors: front left (FL), front right (FR), rear left (RL) and rear right (RR). Also, on top of the radar sensors, an object detection algorithm is executed in real-time during the test drive. The object detector returns an object list expressed w.r.t. VCS. The configuration of the radar sensors on the host vehicle is shown in Figure 4

**B. DATA DESCRIPTION**

In order to evaluate a GRSM, both real radar measurements and ground truth (labels written as oriented bounding boxes) are required. To achieve this goal, a test drive has been performed on a two-way highway road. Then, to acquire proper ground truth, the data from the reference sensor has been labelled. Finally, radar measurements and ground truth have been synchronized.

1) GROUND TRUTH

At a given time frame, the ground truth data contain a set of oriented bounding boxes (OBB), which are defined using the following parameters:

- position:  $[x, y, z]$  [m] of the OBB center expressed w.r.t. VCS,
- orientation:  $[yaw, pitch, roll]$  [rad], defined w.r.t. VCS using the right-handed convention,
- dimensions:  $[length, width, height]$  [m],
- object type: mapped to the elements of the, *OSI::VehicleClassification* structure.

An example frame of the recorded ground truth is presented in Figure 5.

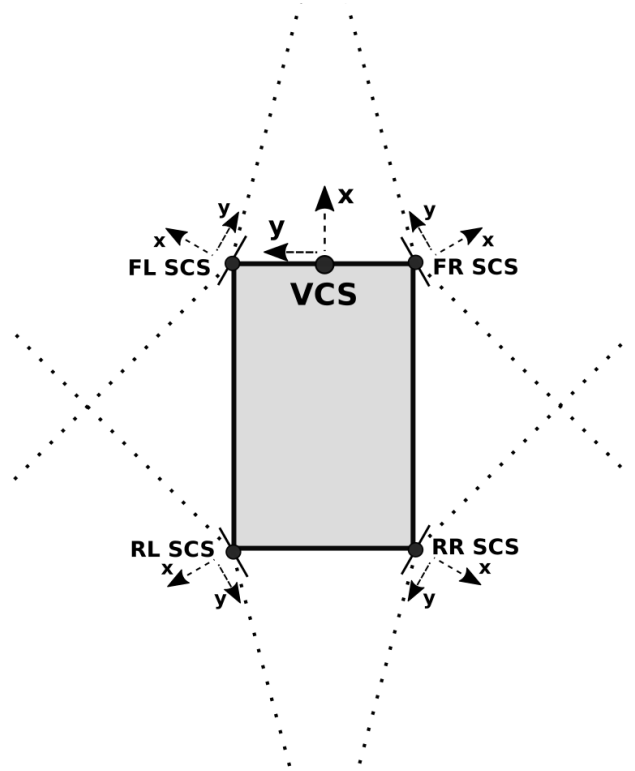


FIGURE 4. Host vehicle with radar sensors.

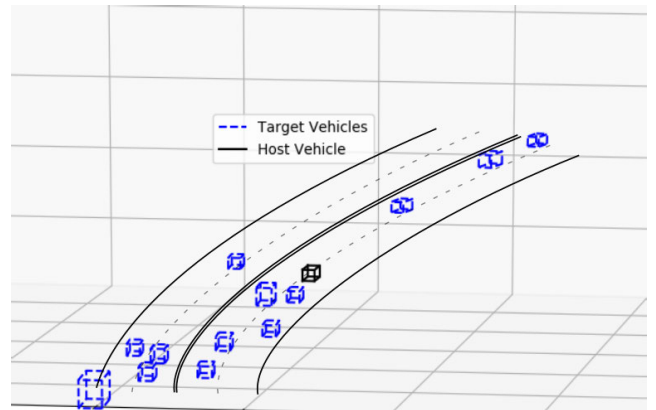


FIGURE 5. Example frame of data with host vehicle and labels.

2) RADAR MEASUREMENTS

A radar sensor returns a point cloud of detections. A single detection is defined by:

- range  $R$  [m] (radial distance to a given SC),
- azimuth  $\phi$  [rad],
- elevation  $\theta$  [rad],
- radial velocity [m/s] (a change of range in the direction of the incident angle).

Additionally, the data from a radar-based object detection algorithm (oftentimes referred as tracker) has been recorded. The incorporated object detector takes as input detection lists

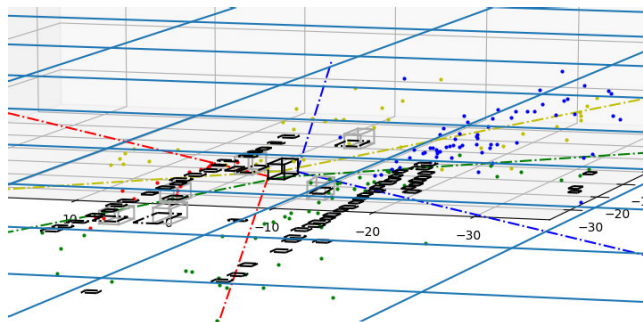


FIGURE 6. Example frame of real data: radar measurements and labels.

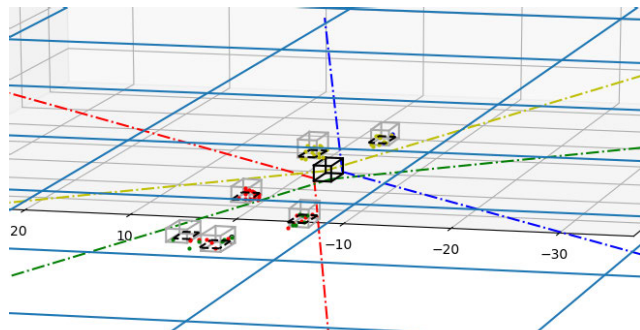


FIGURE 7. Example frame of synthetic data together with labels.

from four corner radars and returns a set of two-dimensional OBBs (object list, tracks). Each OBB is defined using:

- position:  $[x, y]$  [m] of the 2D OBB center, expressed w.r.t VCS,
- orientation:  $yaw$  [rad] defined w.r.t VCS using the right-handed convention,
- dimensions:  $[length, width]$  [m].

An example frame of all types of recorded data, including radar object list, radar detections and labels, is visualized in Figure 6, where:

- to each radar a color is assigned: red - rear right, green - rear left, blue - front right and yellow - front left,
- dashed lines represent the field-of-view area of a given radar sensor,
- dots represent radar detections,
- black-dashed boxes are the radar tracks,
- gray boxes represent labels.

As it is visible on the graph, objects generated from the tracker (black-dashed boxes) and labels (gray boxes) are accurately aligned. As a consequence, the data is valid in terms of synchronization and extrinsic calibration. Also, an important fact can be observed in the presented plot - radar sensors return huge amount of detections on road edges. This is captured by the tracker that generates a set of objects (tracks) on top of that detections.

### C. RESIMULATION

Apart from real tracks, a synthetic object list is required to evaluate a GRSM using the proposed methodology. To be able to associate real and synthetic tracks, the simulated data has to be generated using the recorded ground truth (labels). This can be achieved in two steps:

- 1) Generation of synthetic radar detections (point cloud) by feeding the GRSM with recorded labels.
- 2) Feeding the simulated detections to radar tracking algorithm.

This process is called resimulation, due to the fact that the production radar tracking algorithm is executed in offline mode using the simulated detections. Also, it is worth to mention, that the GRSM should be executed four times - each

time using the extrinsic calibration of one of the four radar sensors.

An example frame of resimulated data (sensor model detections and synthetic object list) is presented in Figure 7, where the meaning of colors is the same as in the case of the figure 6. As it can be noticed in the graph, radar objects are generated only on the OBBs from ground truth. However, this is an expected behaviour, since the current version of the GRSM does not generate detections on static scenery.

### D. END-TO-END METHODOLOGY

Taking into account all of the components, the simplified flow of the framework can be defined as follows:

- 1) Data gathering from a test drive, such as: radar detections, radar object list and measurements from a reference sensor, e.g. camera or lidar.
- 2) Reference data labelling in order to produce a list of OBBs for each object in the recorded scenario. The annotated object list at each time frame is called ground truth. Also, each OBB in the ground truth is called label.
- 3) Synchronization of radar data and ground truth
- 4) Generation of synthetic detections from a GRSM using ground truth as input
- 5) Generation of synthetic radar object list using GRSM detections
- 6) Comparison of real and synthetic object list using a set of defined metrics.
- 7) Fine-tuning of GRSM parameters to obtain the best possible evaluation score

The pipeline listed above is presented with details in Figure 8.

where:

- $label^{t,i}$  is an  $i_{th}$  OBB in the ground truth at time frame  $t$ , represented by its type, dimensions, position w.r.t. VCS and orientation w.r.t. VCS,
- $sc^{t,i,j}$  is a  $j_{th}$  scattering-center generated for the  $i_{th}$  label,
- extrinsic calibration stores mounting position and orientation of the radar sensor  $k \in [1, 4]$ ,
- $R, \phi, \theta, v$  are the range, azimuth, elevation and relative velocity estimates respectively for the  $j_{th}$

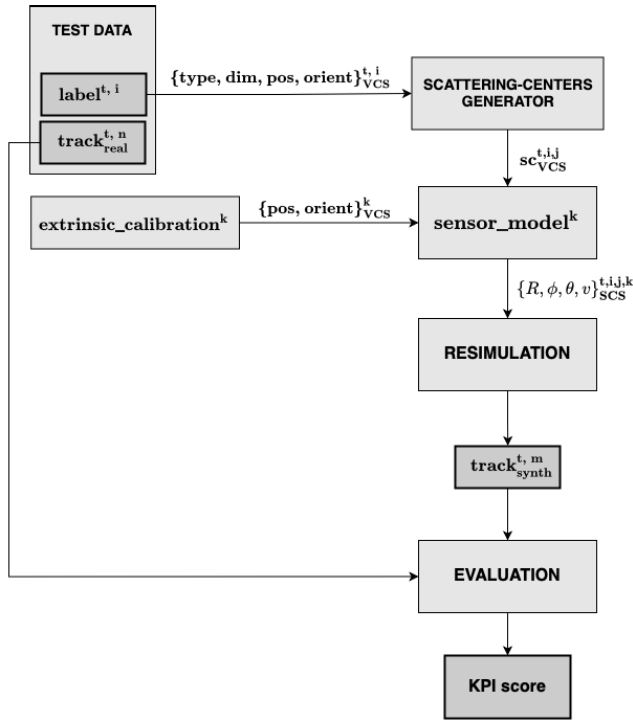


FIGURE 8. Block diagram of the framework.

scattering-center of the  $k_{th}$  GRSM (a single detection expressed w.r.t.  $k_{th}$  SCS),

- $track_{real}^{t,n}$  is an  $n_{th}$  element of the real radar object list,
- $track_{synth}^{t,m}$  is an  $m_{th}$  element of the synthetic radar object list.

To put it in a different way, the input of the framework at a given time moment is a set of labels and the extrinsic calibration of the real sensors. The output from the system is an evaluation score of a GRSM. It is calculated based on a set of object-based metrics that take into account the associated real and synthetic radar tracks. The score can be explicitly used to fine-tune the parameters of the given GRSM. It is worth mentioning, that to robustly assess the fidelity of a given GRSM, a point-cloud-based comparison should be executed together with an object-based evaluation, e.g. using the methodology proposed in the original GRSM paper [16] or a metric based on a deep neural network classifier [19]. That is because, the object-based evaluation may not be enough to check if the sensor model is generating a proper amount of radar-specific noise or if the sparsity of the data is comparable to the real measurements. Nevertheless, as it is shown in this paper, the object-based evaluation provides highly valuable feedback that can be incorporated in the sensor model fine-tuning process. If necessary, a point-cloud-based comparison can be easily added to the framework,

### V. SENSOR MODEL EVALUATION

Two sets of radar object lists (tracks) are available for the framework at a given time moment: real and synthetic. Real object list is taken from the data gathered during the test drive.

This object list is treated as a reference, because it encapsulates the desired radar-specific noise. In particular, the real tracks will not be accurately sticking to their associated ground truth OBBs. Instead, their parameters, e.g. positions, will be floating proportionally to the current amount of noise.

It is assumed, that a GRSM provides a high-fidelity point cloud when the synthetic object list, obtained from the resimulation, is as close as possible to the reference, in the meaning of the generated amount of noise. Let us treat the noise encapsulated in an object list as a random variable  $X$  that is normally distributed with mean  $\mu$  and standard deviation  $\sigma$ . Therefore, two random variables can be defined for real and synthetic object lists respectively:  $X_{real} \sim N(\mu_{real}, \sigma_{real}^2)$  and  $X_{synth} \sim N(\mu_{synth}, \sigma_{synth}^2)$ . The goal of the fine-tuning is to change the GRSM parameters in the way, that the distribution of  $X_{synth}$  gets as close as possible to  $X_{real}$ . However, to enable such fine-tuning, the metrics for comparing probability distributions have to be well-designed and the sensor model itself is required to have a configurable noise profile. For the GRSM used in this paper, the level of clutter is explicitly controlled via the noise figure parameter  $F_n[dB]$  and therefore its fine-tuning (evaluation) is possible.

### A. DATA ASSOCIATION

As it was mentioned before, radar data is extremely sparse and contain a lot of clutter. This fact influences radar tracks that are noisy compared to ground truth OBBs. However, looking at the deviations between radar tracks and the associated ground truth OBBs, it is possible to identify the parameters of both: noise probability distributions  $X_{real}$  and  $X_{synth}$ .

In order to make the comparison between probability distributions reliable, each label has to be associated with only one real and only one synthetic radar track. As a result, the associated data is a set of  $\{label^i, track_{real}^i, track_{synth}^i\} : i \in [1, M]$  triples, collected from all available data. The number of elements in this set,  $M$ , is a sum of all successful associations:

$$M = \sum_{t=1}^T M_t \tag{4}$$

where:

- $T$  is the number of recorded time frames
- $M_t$  represents successful associations count in the time frame  $t$

A synthetic object is associated with a label and a real object (an  $i_{th}$  triple) when:

$$\|p_s - p_l\| < \epsilon \tag{5}$$

$$\|p_s - p_r\| < \epsilon \tag{6}$$

where:

- $p_s, p_r, p_l$  are the positions of the synthetic object, real object and label respectively
- $\epsilon$  is set to a constant value

An example subset of associated data is presented in Figure 9, with the synthetic objects marked orange, real



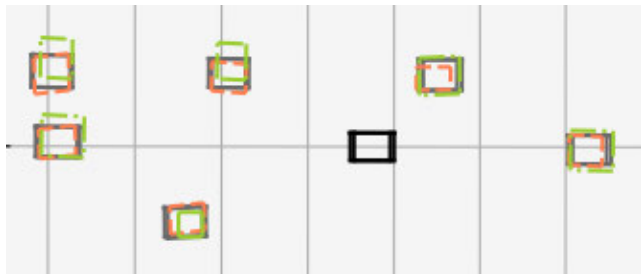


FIGURE 9. Example frame of associated data.

objects marked green, labels marked gray and host marked black.

### B. PROBABILITY DISTRIBUTION IDENTIFICATION

As it was mentioned above, the noise in a radar object list is described as a univariate normal random variable  $X \sim N(\mu, \sigma^2)$ . Therefore, the noise measure has to be represented by a real number. It is rather difficult, since the noise should be intuitively described using the multivariate normal distribution. This is due to the fact, that unknown and independent errors are injected to object position, orientation, dimensions, velocity and more.

However, using a label (OBB) as a reference, it is possible to calculate the Intersection Over Union (IoU) metric for a given radar track [20], associated to this label. IoU is a single-number measure of how well two OBBs fit to each other. It returns 1 in case of perfect overlapping and 0 when the OBBs do not intersect. It can be assumed, that IoU is a joint estimation of all errors injected to the parameters of a given OBB (radar track). As a result, the normal random variable  $X$  can be explicitly represented by the IoU measure, with quite a good accuracy.

In order to identify the probability distribution of the noise injected to a given radar object list (either real or synthetic), the IoU metric has to be computed for all the associated items. Then, based on the IoU results, the mean and variance of  $X$  can be easily calculated.

### C. EARTH MOVER DISTANCE

Taking the probability distributions of  $X_{real}$  and  $X_{synth}$ , representing the IoU measures from real and synthetic data respectively, it is possible to measure how close these two distributions are to each other. To do this, the first Wasserstein distance can be incorporated. It is known as the so-called Earth-Mover-Distance, due to the fact that it describes how much work is required to transfer a given distribution  $\mu$  into another distribution  $\nu$ .

Let  $\mu$  and  $\nu$  denote probability measures on  $\mathfrak{R}$  that are regular and finite. The first Wasserstein distance  $W(\mu, \nu)$  is defined as follows [21]:

$$W(\mu, \nu) = \inf_{\pi \in \Gamma(\mu, \nu)} \int_{\mathfrak{R} \times \mathfrak{R}} \|x - y\| d\pi(x, y) \quad (7)$$

where  $\Gamma(\mu, \nu)$  is a set of all joint probability measures on  $\mathfrak{R} \times \mathfrak{R}$  whose marginals are  $\mu$  and  $\nu$ .

The Wasserstein metric is extremely powerful and it has been already incorporated in various research works, e.g. in training of Generative Adversarial Networks [22] and in clustering of automotive test scenes [23]. Here, the first Wasserstein distance is explicitly used to fine-tune the parameters of the GRSM.

### D. FINE-TUNING

The goal of the fine-tuning is to optimize the noise parameters of the sensor model with respect to the Wasserstein distance, so that the synthetic distribution gets as close as possible to the real distribution. As it has been mentioned before, the noise in the GRSM is explicitly controlled by the noise figure parameter  $F_n$ [dB]. It is a gain that is injected to both measured signal  $x_i$  and current threshold  $th_i$ , during the process of calculating the parameters of a synthetic detection [16]. Note that when  $F_n$  is set to 0, the noise is still injected and statistical test is performed. However, there is no additional gain added to both quantities. When  $F_n \gg 0$ , sensor model is expected to add a lot of noise to each SC parameter. Also, the final set of synthetic detections (point cloud) should be sparse. In the opposite case ( $F_n \ll 0$ ), the set of detections gets close to the one returned by the deterministic geometrical model.

In other words, the goal of the fine-tuning is to find an  $F_n$  value so that the distribution of the set of synthetic detections is of a high-fidelity. It means that the synthetic radar object list should contain similar amount of noise compared to real radar object list. It is worth adding, that the quality of a radar object list is explicitly related to the sparsity of radar detections. In case there are no thresholding and noise applied to the synthetic detections, the radar object list would be almost perfectly aligned with the corresponding labels. On the other hand, when the set of synthetic detections gets very sparse, the radar tracker would have huge problems with accurate estimation of tracks.

To fine-tune the GRSM parameters 500 frames of data have been used. Then, the full pipeline of the validation framework has been executed, separately for each  $F_n \in [-10, 10]$ . In other words, in each iteration all sensor models have been configured given the current  $F_n$  value. In every iteration of the framework pipeline, real objects, synthetic objects and labels have been associated. Then, for each association two IoUs have been computed: real object w.r.t. label and synthetic object w.r.t. label. Finally, using these two sets of IoUs, collected separately for real and synthetic objects, the Wasserstein metric has been calculated. Its value is the evaluation score for the given noise setting.

### E. EVALUATION RESULTS

The Wasserstein metric calculated for all  $F_n$  values using the dataset mentioned above is presented in Figure 10. As expected, a local minimum can be found on the graph. The optimal noise figure value, that has the lowest value of

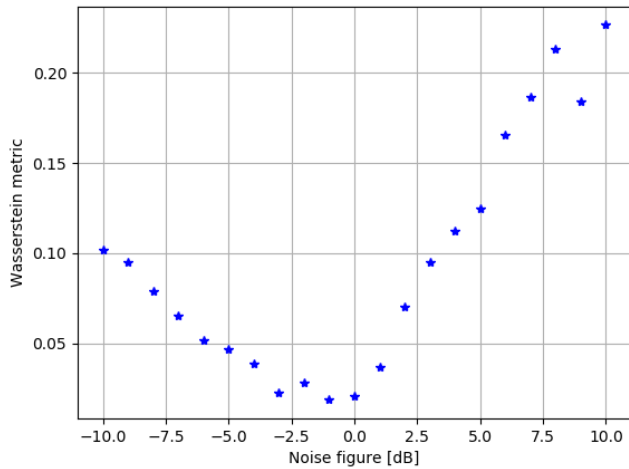


FIGURE 10. Sensor model fine-tuning using wasserstein metric.

TABLE 1. Lateral error - minimum, maximum and mean deviations from real data distribution.

$F_n$	min	max	mean
-8 dB	0.00005	1.42653	0.47663
-1 dB	0.00222	2.20116	0.60624
8 dB	0.00376	2.79022	1.25919

TABLE 2. Longitudinal error - minimum, maximum and mean deviations from real data distribution.

$F_n$	min	max	mean
-8 dB	0.000005	1.055016	0.242126
-1 dB	0.000518	0.947814	0.300391
8 dB	0.001888	2.025263	0.400833

the Wasserstein distance, is  $F_n = -1$  [dB]. Consequently, to obtain results of the highest fidelity, the GRSM should be configured with this  $F_n$  value.

To show how the GRSM noise figure settings affect the quality of radar objects, lateral and longitudinal errors (figures 11 and 12 respectively) have been plotted for a single object located in the front of the host vehicle, for three different  $F_n$  values: -8, -1 and 8 decibels. Additionally, to quantitatively show the differences, minimum, maximum and mean deviations from real data have been computed for those decibels values - tables 1 and 2 for lateral and longitudinal deviations respectively. As it can be noticed on the red plot ( $F_n = 8$  [dB]), when the level of noise injected to synthetic radar detections generated by the GRSM is high, the errors on both lateral and longitudinal axes are significant. On the other hand, in case synthetic detections are ideal, the error encapsulated in the radar object (blue plot,  $F_n = -8$  [dB]) does not correspond to real data (black plot). Finally, the distribution of errors encapsulated in the radar object generated for the noise settings equal to the optimal value (green plot,  $F_n = -1$  [dB]) is close to real data.

In other words, the obtained results prove that the stochastic properties of the radar object list are affected by the quality of the radar detections. As a result, the fidelity of a GRSM,

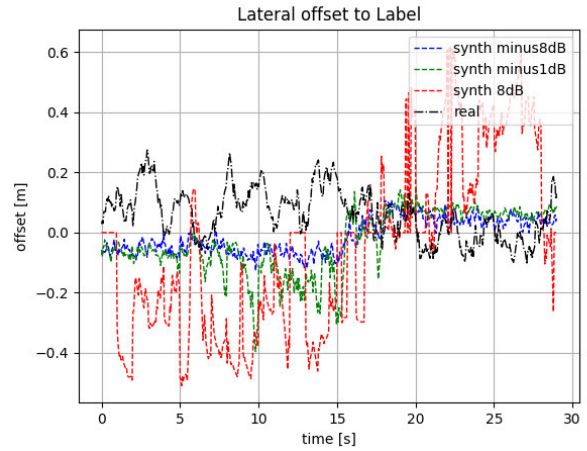


FIGURE 11. Lateral error over time for a single radar object (recorded for a few different noise parameters) to its associated label.

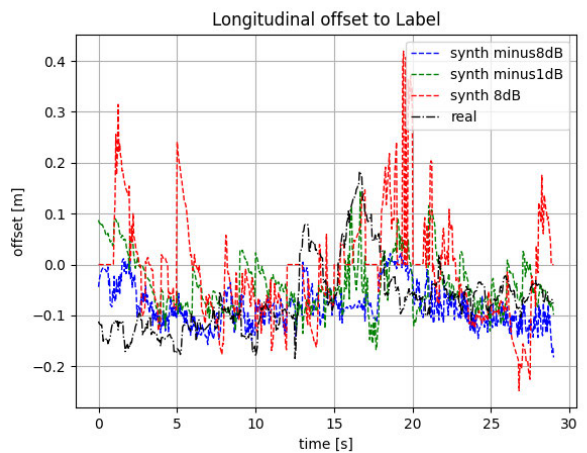


FIGURE 12. Longitudinal error over time for a single radar object (recorded for a few different noise parameters) to its associated label.

being used in the virtual validation process of an ADAS, can be explicitly increased by fine-tuning its noise level. This shows the usefulness of the proposed evaluation methodology. It is still necessary to perform point-cloud-based evaluation in sensor model validation process. However, object-based assessment can be treated as an unequivocal source of feedback of a GRSM fidelity. Namely, when the distribution of synthetic detections is set appropriately (via the noise figure parameter), the corresponding synthetic object list accurately emulates real tracks. Such proof of the reliability of the synthetic data is needed before incorporating a sensor model into the virtual validation process.

## VI. CONCLUSION

In this paper a robust and straightforward end-to-end methodology for validating generic radar sensor models has been presented. The goal of the method is to assess the performance of a given sensor model on an object-list level, due to the fact that objects are crucial in decision-making processes

in Advanced Driver Assistance Systems. To show how the full pipeline of the framework can be executed, an exemplary radar sensor model has been integrated. Thanks to the presented framework and incorporation of a well-defined metric a successful sensor model fine-tuning has been performed.

## REFERENCES

- [1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transp. Res. A, Policy Pract.*, vol. 94, pp. 182–193, Dec. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0965856416302129>
- [2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," 2017, *arXiv:1711.03938*.
- [3] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," 2017, *arXiv:1705.05065*.
- [4] T. Sulkowski, P. Bugiel, and J. Izydorczyk, "In search of the ultimate autonomous driving simulator," in *Proc. Int. Conf. Signals Electron. Syst. (ICSES)*, Sep. 2018, pp. 252–256.
- [5] M. Holder, P. Rosenberger, H. Winner, T. Dhondt, V. P. Makkapati, M. Maier, H. Schreiber, Z. Magosi, Z. Slavik, O. Bringmann, and W. Rosenstiel, "Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2616–2622.
- [6] M. Jasinski, "A generic validation scheme for real-time capable automotive radar sensor models integrated into an autonomous driving simulator," in *Proc. 24th Int. Conf. Methods Models Automat. Robot. (MMAR)*, Aug. 2019, pp. 612–617.
- [7] T. A. Wheeler, M. Holder, H. Winner, and M. Kochenderfer, "Deep stochastic radar models," 2017, *arXiv:1701.09180*.
- [8] A. Scheel and K. Dietmayer, "Tracking multiple vehicles using a variational radar model," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3721–3736, Oct. 2019.
- [9] N. Hirsenkorn, T. Hanke, A. Rauch, B. Dehlink, R. H. Rasshofer, and E. Biebl, "Virtual sensor models for real-time applications," *Adv. Radio Sci.*, vol. 14, pp. 31–37, Sep. 2016.
- [10] H. Ling, R.-C. Chou, and S.-S. Lee, "Shooting and bouncing rays: Calculating the RCS of an arbitrarily shaped cavity," *IEEE Trans. Antennas Propag.*, vol. 37, no. 2, pp. 194–205, Feb. 1989.
- [11] C.-F. Yang, B.-C. Wu, and C.-J. Ko, "A ray-tracing method for modeling indoor wave propagation and penetration," *IEEE Trans. Antennas Propag.*, vol. 46, no. 6, pp. 907–919, Jun. 1998.
- [12] N. Hirsenkorn, P. Subkowski, T. Hanke, A. Schaermann, A. Rauch, R. Rasshofer, and E. Biebl, "A ray launching approach for modeling an FMCW radar system," in *Proc. 18th Int. Radar Symp. (IRS)*, Jun. 2017, pp. 1–10.
- [13] C. Schüßler, M. Hoffmann, J. Bräunig, I. Ulimann, R. Ebel, and M. Vossiek, "A realistic radar ray tracing simulator for large MIMO-arrays in automotive environments," *IEEE J. Microw.*, vol. PP, pp. 1–13, 09 2021.
- [14] K. Schuler, D. Becker, and W. Wiesbeck, "Extraction of virtual scattering centers of vehicles by ray-tracing simulations," *IEEE Trans. Antennas Propag.*, vol. 56, no. 11, pp. 3543–3551, Nov. 2008.
- [15] J. Guo, W. Deng, S. Zhang, S. Qi, X. Li, C. Wang, and J. Wang, "A novel method of radar modeling for vehicle intelligence," *SAE Int. J. Passenger Cars Electron. Electr. Syst.*, vol. 10, no. 1, pp. 50–56, Sep. 2016.
- [16] A. Martowicz, A. Gallina, and G. Karpel, "Uncertainty propagation for vehicle detections in experimentally validated radar model for automotive application," in *Proc. 24th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2019, pp. 606–611.
- [17] S. Riedmaier, B. Danquah, B. Schick, and F. Diermeyer, "Unified framework and survey for model verification, validation and uncertainty quantification," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 2655–2688, Jun. 2021.

- [18] T. Hanke, N. Hirsenkorn, C. van Driesten, P. Garcia-Ramos, M. Schiementz, S. Schneider, and E. Biebl, "Open simulation interface - a generic interface for the environment perception of automated driving functions in virtual scenarios," 2017.
- [19] A. Ngo, M. P. Bauer, and M. Resch, "Deep evaluation metric: Learning to evaluate simulated radar point clouds for virtual testing of autonomous driving," in *Proc. IEEE Radar Conf. (RadarConf)*, May 2021, pp. 1–6.
- [20] A. Rosebrock. (2016). *Intersection Over Union (IOU) for Object Detection*. Accessed: Mar. 6, 2019. [Online]. Available: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
- [21] A. Ramdas, N. G. Trillos, and M. Cuturi, "On Wasserstein two-sample testing and related families of nonparametric tests," *Entropy*, vol. 19, no. 2, 2017, Art. no. 47. [Online]. Available: <https://www.mdpi.com/1099-4300/19/2/47>, doi: [10.3390/e19020047](https://doi.org/10.3390/e19020047).
- [22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
- [23] P. Kowalczyk, P. Bugiel, J. Izydorczyk, and M. Szelest, "Clustering of test scenes by use of Wasserstein metric and analysis of tracking quality in optically degraded videos," Wydawnictwo Politechniki Śląskiej, Gliwice, Poland, Mar. 2021, doi: [10.5281/zenodo.4628038](https://doi.org/10.5281/zenodo.4628038).



Polish Chapter of the IEEE Vehicular Technology Society.

**MICHAL JASINSKI** (Graduate Student Member, IEEE) received the master's degree in automatic control and robotics. He is currently pursuing the Ph.D. degree with the AGH University of Science and Technology, Cracow, Poland. He works in cooperation with the automotive industry, where he is involved in research and development projects for advanced safety systems. His research interests include sensor modeling, virtual validation, and machine learning. He is a member of a



**PAWEŁ SKRUCZ** (Senior Member, IEEE) is a Professor of control engineering with the AGH University of Science and Technology, Cracow, and an Advanced Engineering Manager AI & Safety with the Aptiv Technical Center, Cracow. His current research interests include dynamical systems, autonomous systems, artificial intelligence, machine learning, modeling and simulation, and applications of control theory to software systems.



**MATEUSZ KOMORKIEWICZ** (Senior Member, IEEE) is a Senior Computer Vision and Artificial Intelligence Engineer with the Advanced Engineering Department, Aptiv. His research interests include deploying AI solutions on SoC devices and nonstandard ML usage in automotives.

...