Probabilistic Sensor Models for Virtual Validation – Use Cases and Benefits

Dr. Robin Schubert
Co-Founder & CEO
BASELABS GmbH
BASELABS enables data fusion results.
Who we are

- Data fusion enthusiasts
- Team of 25 engineers, software developers and managers
- Software supplier to OEMs and automotive suppliers
- Active contributors to advances in research

What we do

- Focus on multiple sensor scenarios – with software products and projects
- Enable data fusion as a key technology for automated driving

We partner

- Vector is strategic partner with 49% of shares
- Partnership with simulation providers for improved virtual validation
A brief survey on data fusion
Example of a 360° data fusion system for automated driving

BASELABS’ mission is to provide a unified, ambiguity-free, reliable representation of the vehicle's environment
Data fusion in a nutshell

A brief summary of Bayesian filtering and multi objects tracking

- **World Model**
  - Posterior state vector $\hat{x}_{k-1}$
  - Prior state vector $\hat{x}_k$

- **Sensor Model**
  - Sensor data hypothesis $\hat{z}_k$

- **Sensor Data**

**Evaluation**

**Update**
Virtual validation in a nutshell

A brief summary of model/software/hardware-in-the-loop (xIL) validation

World Model

Scenario

Simulated entities

Sensor Model

Sensor data

DUT = Device under Test

Feedback

Scenario → Simulated entities → Sensor data → DUT → Feedback

Apply & Innovate 2016
Data fusion revisited

Properties of a good sensor model in Bayesian filtering

- High similarity between simulated and real sensor data (in case of valid hypothesis)
  - Similar level of preprocessing
  - Mathematical sensor model needs to capture sensor behavior
  - Parameters of the model need to capture properties of specific sensor device
  - Model needs to deal with data variability
- Fast comparison between prediction and data

→ Probabilistic sensor models
  (deterministic & probabilistic part)
Example: Sensor model for a multi objects tracking system

Modelled phenomena

World model

- 3 object hypotheses with position, direction and speed

Sensor model: We expect

- 3 sensor detections
- of which some may not be detected (false negatives)
- whereas we might get some clutter (false positives) and
- the detections may not be precisely where expected (noise) plus
- they might be delayed (latency)

This expectation is evaluated against the actual sensor data
Use cases and requirements for virtual validation

(Focus on simulation of ADAS–related sensor data)
Use Case 1: „Lucky–Path–Testing“

- Test of developed system under optimal conditions
- Rationale: „If it does not even work with perfect data yet, I do not need to start testing with real data.“

Potential target audience:
- Developers of virtual proof–of–concept implementations
- Testers at the beginning of the test process
Use Case 2: „Virtual Validation“

- Ensure the correct functionality of the system
- Complement or even partial replacement of field testing
- Objective: Simulation should be as realistic as possible
- Limitation: Real-time requirements of test system (in particular for HiL-setups)

- Most realistic sensor models: Simulation on physical level (EM-waves)

Would physical simulation models be the solution if they were fast enough?

Yes and No (a.k.a. it depends)
The required level of simulated sensor models depends on the DUT's input.
The required level of simulated sensor models depends on the DUT’s input

Target group A: Developers and testers of image processing (detectors/classifiers)

- Input of DUT: Camera images
- Requirement for virtual validation: Simulated camera images that are as realistic as possible
- Optimal solution: Sensor models on physical level (image rendering)
The required level of simulated sensor models depends on the DUT’s input

Target group A2: Developers and testers of complete data processing chain including image processing

- Input of DUT: Camera images
- Requirement for virtual validation: Simulated camera images that are as realistic as possible
- Optimal solution: Sensor models on physical level (image rendering)
The required level of simulated sensor models depends on the DUT’s input

Target group B: Function developers and testers

- Input of DUT: Track list (e.g., list of vehicles in front of ego vehicle)
- Requirement for virtual validation: Track lists that are as realistic as possible
- Optimal solution: Sensor models on physical level (image rendering)
  Sensor model that behaves like a realistic tracker

Smart sensors (e.g., MobilEye cameras or recent radars) do not even output raw data, just track lists.
The required level of simulated sensor models depends on the DUT‘s input

Target group B2: Tracking/Data Fusion developers and testers

- Input of DUT: Detections (e.g., list of image regions containing vehicles)
- Requirement for virtual validation: List of detections that are as realistic as possible
- Optimal solution: Sensor models on physical level (image rendering)  
  Sensor model that behaves like a realistic detector/classifier
Intermediate Summary: Claims and Assumptions

Idealized (=error free) sensor models cannot be more than a starting point

Sensor models on physical level are a valuable solution for developers and testers of systems based on sensor raw data (e.g., detectors/classifiers)

Sensor models on physical level are not suitable for developers and testers of data fusion, tracking, or functions (as well as users of smart sensors)

For this target group, sensor models should simulate a realistic behavior on the correct processing level
Probabilistic sensor models

BASELABS Models for Carmaker
There is no need to re-invent the wheel

Probabilistic sensor models from the tracking/data fusion domain are designed for these requirements.
BASELABS Models for CarMaker (planned)

- Probabilistic sensor models from BASELABS
- Development based on extensive experience using real ADAS sensors
- Models can simulate detector and tracking output
- Tight integration IPG Carmaker (including visualization, parameterization, simulation, and licensing)
Modelled sensors

ADAS sensors with typical interfaces

Radar Detector
- Provides an interface typical for automotive radar sensors
- Delivers measurements in Polar coordinates relative to the sensor position
- Models specific errors of radar sensors delivering detections

Smart Camera
- Provides an interface typical for automotive smart camera systems
- Delivers measurements in Cartesian coordinates relative to the sensor position

Errors common to sensors
- Latency
True positives with measurement noise

Simulation of a more realistic radar sensor

Phenomena modelled by BASELABS Models:

- False negatives (Objects that are not detected by the sensor)
- False positives (Detections that do not originate from real objects)
- True positives with measurement noise: Stochastic variations of measured signals (range, velocity, azimuth angle)
- Cardinality: One object generates more than one detection
Cardinality

Simulation of a more realistic sensor

Phenomena modelled by BASELABS Models:

- Cardinality: One object generates more than one detection
  - Depends on the size of the object (trucks, cars, …)
  - Depends on the distance of the object
  - Depends on characteristics of specific sensor
  - Enables simulation of data fusion for extended object tracking
Latency

Simulation of a more realistic sensor

Phenomena modelled by BASELABS Models for all sensors:

- Latency: Object detections represent time in the past (due to delay caused by the signal processing and data transmission in real sensors)
With BASELABS Models for CarMaker, the developed perception algorithm or ADAS function can be tested in real-time with more realistic sensor data!
Parameterization of probabilistic sensor models

Simulating the statistical behavior of specific sensor devices
The parameters describing the quality of the data of the probabilistic sensor model can be adopted to simulate different sensors.

- General option for the selection of model parameters
  - Based on the user’s experience
  - Based on validation of real sensor to determine its statistical parameters

Main question: what are the right parameters?
Tool–supported inspection of real sensor data

BASELABS Validate

- Integrated annotation of sensor and reference data
- Visual inspection and comparison of sensor data and reference data
- Automatic calculation of geo–referenced sensor performance metrics
Integrated calculation of statistical parameters

- “Heat map“ of detections on the reference objects
- Exemplary error distribution for azimuth, range, and range rate of a radar sensor
- Derivation of statistical parameters, e.g.
  - Measurement noise
  - Detection rate
  - Cardinality
  - Latency
What is your opinion on the topic?