# Deliverable D5.1

## System performance assessment specification

| | |
|---|---|
| Dissemination level | CO/RE/PP/PU |
| Version | 1.1 Final |
| Lead contractor | FICOSA |
| Due date | 01.06.2016 |
| Version date | 23.12.2016 |

This project is co-funded
by the ECSEL JU

# Document information

AUTHORS

Andrea Saccagno – FICOSA
Werner Ritter – DAI
Pablo Mejuto - CTAG
Bernd Schäufele – FOKUS
Fabio Tango – CRF
Chistian Schyr – AVL_DE
Claus Christmann - BOSCH
Matti Kutila - VTT
Florian Kuhnt - FZI
Peter Wolf - FZI
Hendrik Königshof – FZI
Christian Waldschmidt - UULM

Coordinator

Dr. Werner Ritter
Daimler AG
Wilhelm-Runge-Straße 12
89082 Ulm
Germany
Phone: +49 731 505 2140
Email: Werner.R.Ritter@Daimler.com

Project funding

H2020-EU.2.1.1.7.
ECSEL Programme
Contract No. 661933

# Legal Disclaimer

# Revision and history chart

| Version | Date | Comment |
|---|---|---|
| 0.1 | 14.04.2016 | Table of content Planning |
| 0.2 | 14.06.2016 | Revised document structure |
| 0.3 | 29.06.2016 | Added test cases 3.4.3 and 3.4.4 for LIDAR |
| 0.4 | 01.06.2016 | Added test cases 1 and 2 for cameras |
| 0.5 | 04.06.2016 | Added contributes form CRF and interior camera for driver monitoring |
| 0.6 | 18.07.2016 | Added contributed from FOKUS on monitoring software |
| 0.7 | 20.07.2016 | Added FZI parts to Monitoring Algorithm, added test cases 9 and 10, small corrections |
| 0.8 | 22.07.2016 | Added paragraphs about offline and online assessment to FOKUS part |
| 0.9 | 26.07.2016 | Corrected some typos and revised titles and indexes. |
| 0.10 | 28.07.2016 | Extensions to Test case 8-10 |
| 0.11 | 28.07.2016 | Language, Typos etc. |
| 0.12 | 28.07.2016 | Added contribution from AVL DE on test methodology |
| 0.13 | 31.08.2016 | Revised text introduction par. 4.1. Text language set to British English. |
| 0.14 | 29.07.2016 | Added scenario head-on collision for use case 10 non-compliant behaviour |
| 1.0 | 31.08.2016 | Final - Revised according peer reviewers indications. |
| 1.1 | 01.12.2016 | New issue taking account requests from Commission |

# Table of contents

**1 Summary** **7**

**2 Introduction** **9**

**3 System Performance Assessment (SPAM)** **10**

3.1 Architecture and integration 10

3.1.1 Interfaces and metrics 12

3.1.2 Information processing. Degradation levels and global platform status 15

3.1.3 SPAM integration 16

3.2 Monitoring Algorithm 17

3.2.1 Logging and Monitoring 17

**4 System Performance Assessment Module validation** **22**

4.1 Test Case 1: Camera test with Videos 22

4.1.1 Test case 1 degradation levels 23

4.1.2 Test case 1 degradation modelling 23

4.2 Test Case 2: Comparison of camera tests on vehicle and videos 25

4.2.1 Test case 2 degradation levels 26

4.2.2 Test case 2 degradation modelling 26

4.3 Test Case 3: Interior camera system for driver monitoring 26

4.3.1 Test case 3 degradation levels 27

4.4 Test Cases for LIDAR 27

4.4.1 Test Case 4: LIDAR tests in fog chamber 27

4.4.2 Test Case 5: LIDAR tests in real traffic 28

4.5 Test Cases for RADAR 29

4.5.1 Test Case 6: RADAR tests in laboratory 29

4.5.2 Test Case 7: RADAR tests in real traffic 29

4.6 Test Case 8: Mapping and Localization tests 30

4.7 Test Case 9: Sensor Uncertainties 30

4.8 Test Case 10: Non-compliant Behaviours 32

**5 Conclusions** **36**

# List of figures

Figure 1.1: The overall RobustSENSE system architecture 7

Figure 2.1: Interaction of Task 5.1 with other task and related documents 9

Figure 3.1: The System Performance Assessment Module relationship with Sensor, Data Fusion and Understanding and Planning layers. 10

Figure 3.2: The System Performance Assessment Module (SPAM) architecture 13

Figure 3.3: Integration of the SPAM in the CAN network 16

Figure 3.4: Measurands online overview 18

Figure 3.5: Workflow of the Logging libraries 18

Figure 3.6: End-to-End workflow of the logging facilities 19

Figure 4.1: Scenario model with static and dynamic objects 23

Figure 4.2: Simulated video data streams 24

Figure 4.3: Image distortion and blending 24

Figure 4.4: Simulated rain condition on camera 25

Figure 4.5: The Camera set considered for the vehicle exterior. 26

Figure 4.6: Fog chamber for validating performance of LIDAR sensor in foggy conditions 28

Figure 4.7: Validation software and facilities of the test car 29

Figure 4.8: Sensor uncertainties test case. 31

Figure 4.9 Sensor uncertainties test case: opposing vehicle entering the street. 32

Figure 4.10: Non-compliant behaviour I. 33

Figure 4.11: Non-compliant behaviour II. 34

Figure 4.12: Overtaking in oncoming traffic 35

Figure 4.13: Wrong-way driver on highway 35

# 1 Summary

Autonomous driving is one of the technological mega-trends in automotive industry today. Semi-autonomous vehicles are planned to be introduced in the market in a few years. Today, automation is taking role also in normal driving (e.g. highway, intersection, etc.). The main challenge, which still remains, is the reliability and robustness of the sensor systems in all possible environmental conditions, which at the same time can be a challenge for a human driver also. RobustSENSE project aims to define a platform for tackling this problem generating a reliable situation map for vehicle control units.

The platform considered is sketched in the block diagram of Figure 1.1. This figure is described in detail in other Deliverable reports, in particular in D2.1 “Initial system specification”.

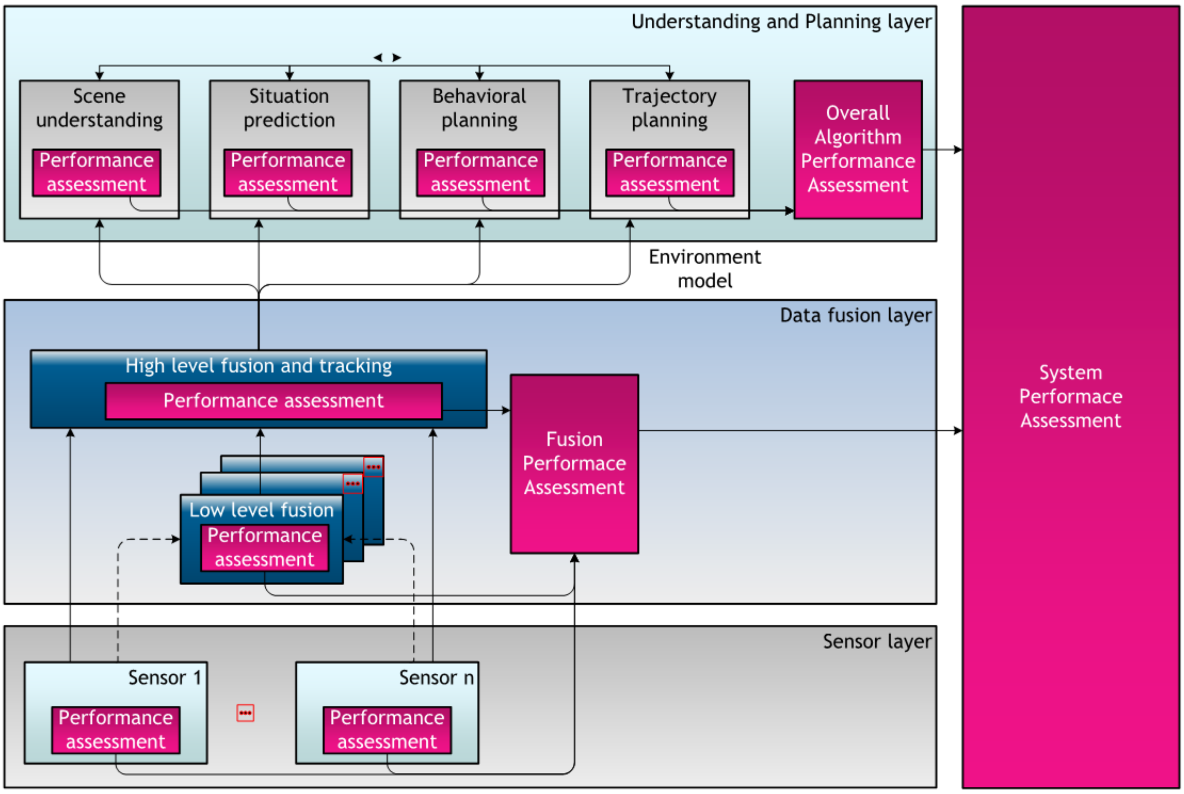



Figure 1.1: The overall RobustSENSE system architecture

RobustSENSE architecture is organised in three different layers regarding their own roles:

1. **Sensor layer**: Sensor level components (hardware and software) and their output signals.
2. **Data fusion layer**: Having both low level sensor fusion for raw sensor data and high-level fusion modules for fusing object level data
3. **Understanding and planning layer**: This layer takes care of scene understanding and decision making concerning desired vehicle control and intervention functions and planning of the correct trajectory.

All layers include performance assessment sub-modules for checking if performance over the time remains and corresponds with the initial parameters.

The present document addresses the description of the specific performance assessment sub-module at whole system level. This sub-module is planned to get the overall assessment of the vehicle capability to survey in the actual driving conditions, thus to assess the actual robustness of the specified system.

# 2 Introduction

The aim of the RobustSENSE project is to:

*Develop a sensor platform for automated and autonomous driving, that overcomes the limitations of existing sensors and provides enhanced sensing capabilities.*

To address this scope, RobustSENSE architecture prescribes that the performance of sensors, algorithms and system as a whole is monitored in order to determine the global platform status and its level of reliability during execution.

This report describes the functionality of the system performance component and all the interfaces to the vehicle. It describes the result of the activities done by partners in Task T5.1. Figure 2.1 shows the interaction of T5.1 with other workpackages and tasks:

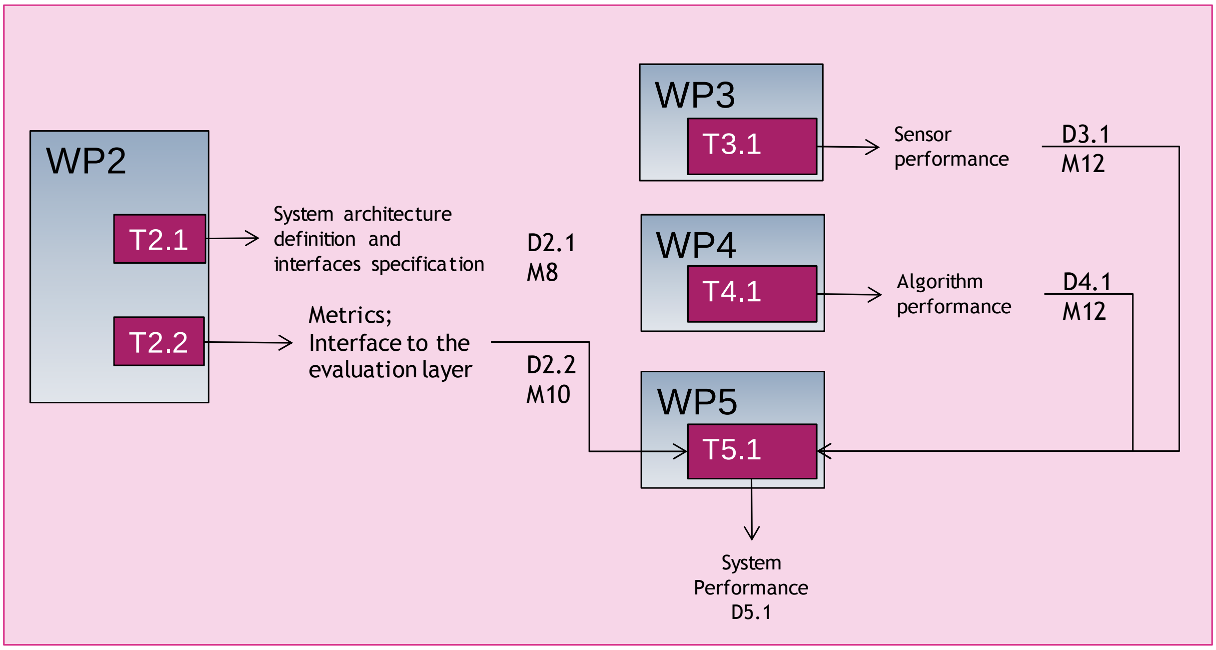



Figure 2.1: Interaction of Task 5.1 with other task and related documents

# 3 System Performance Assessment (SPAM)

The base concept of RobustSENSE is that performance assessments have to be done at any level: sensors provide an assessment on their own performance, which is taken in account in the fusion level, which produces an assessment on its own performance, which is in turn taken in account by the understanding and planning layer which produces the overall one relative to the layers itself. Finally this assessments and the one from the data fusion layer are used to produce the Performance assessment at System Level. This latter is evaluated by the algorithm described in the following chapters.

## 3.1 Architecture and integration

In order to be able to perform an overall system performance assessment in RobustSENSE is planned the definition of a specific software module aimed to continuously evaluate the performance status of the whole vehicle by processing inputs from the specific Performance Assessment modules of the Data Fusion and Understanding and Planning layer (see Figure 3.1).

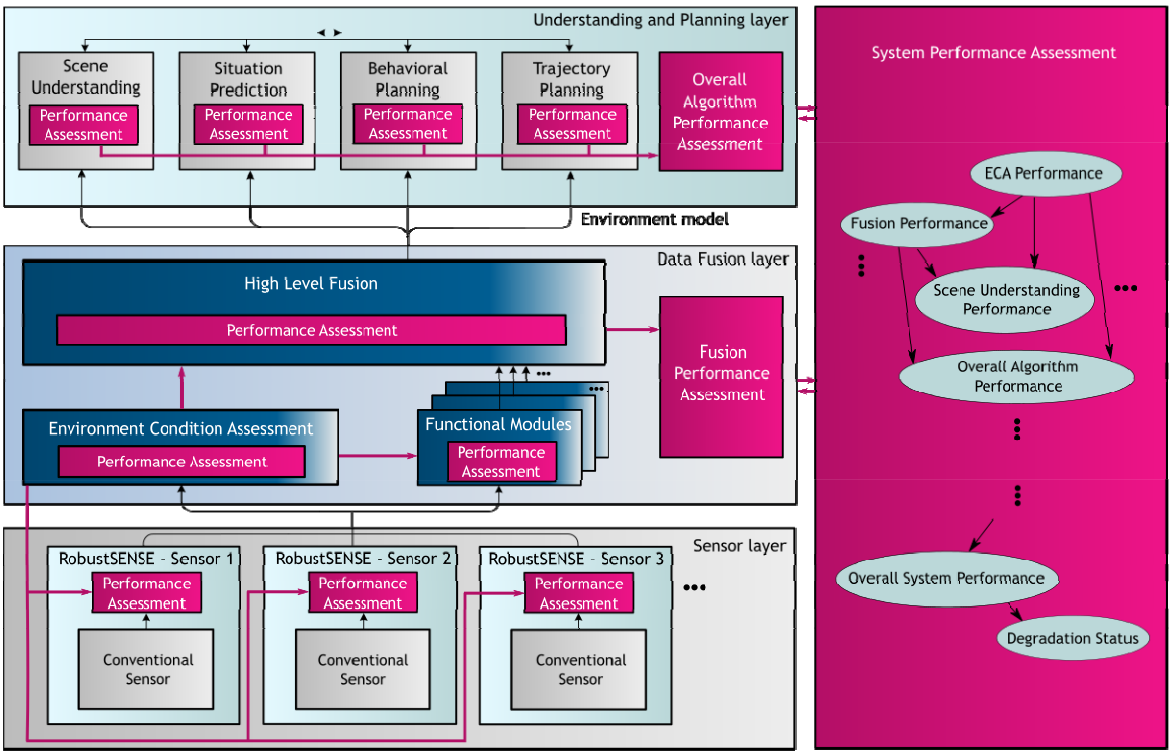



Figure 3.1: The System Performance Assessment Module relationship with Sensor, Data Fusion and Understanding and Planning layers.

The Performance Assessment in the Data Fusion layer is performed for the individual modules and the overall layer. Every functional module is responsible for a specific task and uses the data and performance assessment of various sensors for its estimations. The Environment Condition Assessment module estimates the current environmental conditions and feeds this data back to the Sensor layer. The sensors may consider this condition in their measurement models during their individual performance assessment. The functions which should result not fully available in harsh conditions are tagged as in a “degraded” mode. The High Level Fusion uses the estimates of the functional modules and the Environment Condition Assessment module and outputs a probabilistic environment description containing detected objects, the street topology as well as the map with localization, taking into account the degraded modes too. The performance assessment of the High Level Fusion takes the performance assessment of the functional modules and the Environment Condition Assessment into account and is propagated to the Fusion Performance Assessment of the Data Fusion layer. There the validity of the environment model is evaluated in a holistic manner based on the confidence, existence probability and consistency of the fused data. Since each sub-module performs in a probabilistic manner, the confidence can be derived, e.g. from filter covariance. This evaluation of the environment model’s validity is passed on to the System Performance Assessment Module (SPAM).

Similarly, in the Understanding and Planning layer a performance assessment occurs in each single module (Scene Understanding, Situation Prediction, Behavioural Planning, Trajectory Planning) to indicate properly their overall function. Those assessments are meant to be in a rather qualitative manner. Performance metrics like variance values of predicted future states are combined to human-readable properties like “the prediction is very inconsistent”. The Overall Algorithm Performance Assessment module aggregates all this data and propagates its input to the SPAM.

In this way the performance properties of the Data Fusion and Understanding and Planning layer are further logically combined in the SPAM and degradation decisions are derived. Discrete Bayesian methods can be utilized to derive descriptive system performance values online that are then passed back to the modules in the Data Fusion and Understanding and Planning layer. Thus every module becomes aware of the performance and degradation of the overall system and its distinct elements in these layers. This allows the utilization of sophisticated processing algorithms that consider the source of any degradation. Moreover, information from high level modules can be passed back to low level modules. For example, the Environment Condition Assessment modules can profit from performance assessments of scene understanding that can give a hint on the current environment condition. Based on this, the modules will be able to evaluate the degradation and consequently adapt their algorithms or switch to alternative algorithms to fit the current perception situation of the environment, whenever it is possible.

In Table 3.1 are summarised the inputs and outputs between the SPAM and the Data Fusion and Understanding and Planning layers.

Table 3.1: Inputs and outputs of the SPAM

| Layer | Input:<br>info provided to SPAM | Output:<br>Info provided from SPAM |
|---|---|---|
| Data Fusion | Environment model vaildity:<br>1. Confidence level of Fused data<br>2. Existence probability<br>3. Consistency of fused data | Degradation levels of the overall system |
| Understanding and planning | • Performance metrics (variance values of predicted future states)<br>• Status of the prediction (consistent, inconsistent) | |

### 3.1.1 Interfaces and metrics

In accordance with Deliverable 2.1, the System Performance Assessment module has two interfaces as input: IF5.1 from Overall Algorithm Performance Assessment and IF5.2 from Fusion Performance Assessment. The output of the System Performance Assessment module is provided by interface IF5.3 to the Understanding and Planning layer, by interface IF 5.4 and IF 5.5 to the Data Fusion layer and by interfaces IF5.6.1 through IF5.6.8 to the sensor modules. Eventually IF5.10 provides the output from the System Performance Assessment to Offline Assessment Collection with the data produced by the SPAM.

The implementation of the input interfaces IF5.1 and IF5.2 to the System Performance Module is realized by the logging libraries created by the code generator described in section 3.1. Even though the specific methods and message content is different for the interfaces, they are created in a uniform way. All input interfaces to the System Performance Assessment module is realized through a software library containing methods to collect all metrics from a specific component.

The output interfaces are also provided in a uniform way. As soon as a new value is collected through IF5.1 or IF5.2 it is provided to all other components trough a UDP socket channel. Each component can use this interface to receive the latest metrics from the System Performance Assessment module.

Four sensors are considered in RobustSENSE: RADAR, Camera, LIDAR, and localization. Each of these sensors sends raw data into each individual data processing module, which form a block with the sensor and its performance assessment module, which evaluates the data validity.

The inputs from the sensor performance assessment module are described in deliverable D2.2. According the type of sensors, they can be

- Gradual measures (for example the camera image state, or lidar confidence index or localization uncertainty)
- Binary measures (like sensor working or not, for example)

The Inputs from Fusion performance assessment module are described in deliverable D2.2. as well, and are related to :

- Fused existence probability
- Dynamic stated uncertainties
- Unobservable regions

The basic architecture of the SPAM is shown in the following Figure 3.2.

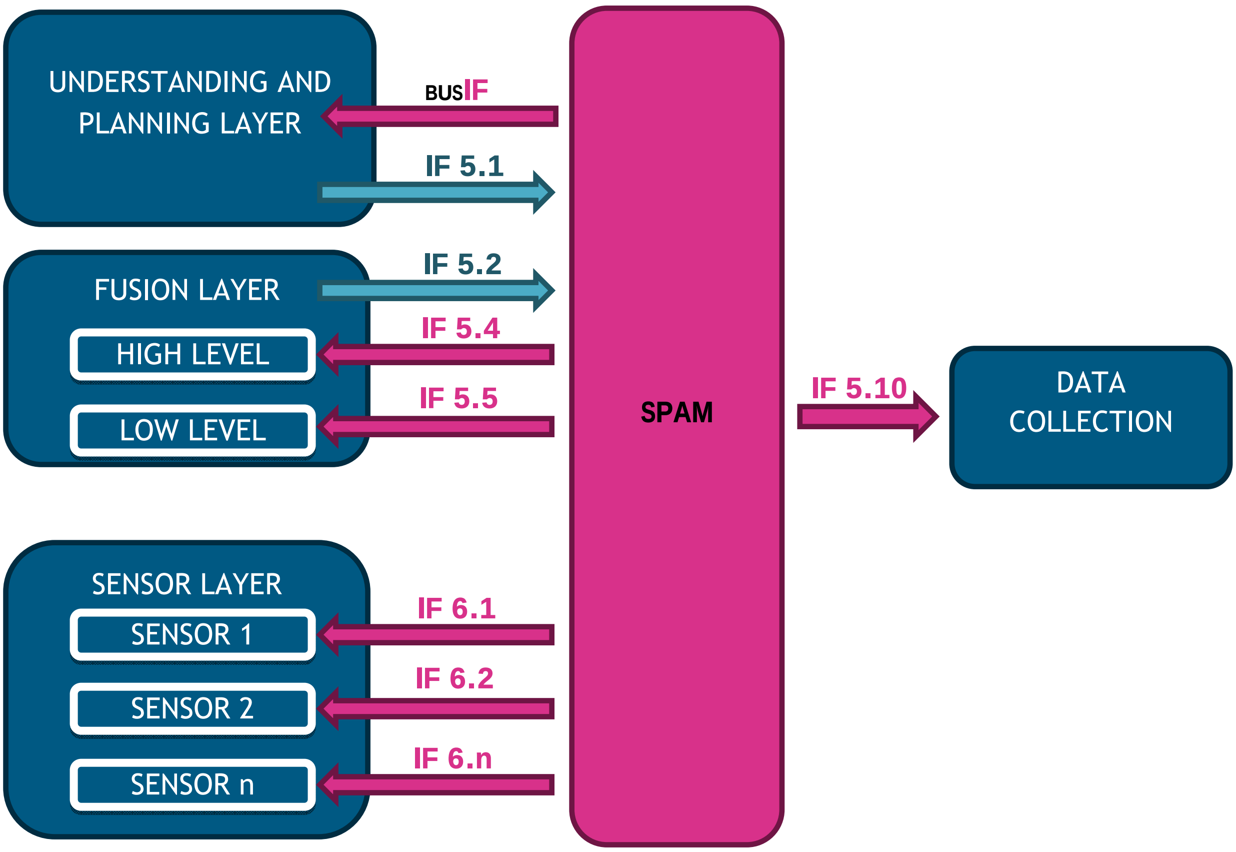



Figure 3.2: The System Performance Assessment Module (SPAM) architecture

The description of the SPAM interfaces is given in Table 3.2.

Table 3.2: Description of SPAM interfaces

| **Interface** | **Link with SPAM** | **Description** |
|---|---|---|
| IF 5.1 | From Understanding and Planning layer | Evaluation from the Overall Algorithm Performance Assessment module. It carries sensor-independent data. |
| IF5.2 | From Performance Assessment Module in the data fusion layer | Quality of the low level fusion and the high level fusion and tracking modules. This allows the SPAM to evaluate the overall performance and reliability of the current sensor array. It carries sensor-independent data. |
| IF5.3 | To Understanding and Planning layer | Back-propagation of decisions and metrics from the system performance assessment, to the four Understanding and Planning modules. They can use it to improve and/or optimize their algorithms for the current perception situation of the environment and its evaluation in the online SPAM. |
| IF 5.4 | To Data fusion Layer (High Level) | Interface to the high level fusion and tracking block. There is generated an environment model. |
| IF 5.5 | To Data fusion Layer (Low Level) | Interface to the lower part of it, i.e. to the low level fusion block. Here are produced fused data with all the data generated by the sensors along with a multi-object tracking. |
| IF 5.10 | To data collection for offline analysis | The interfaces here described are the ones that will enable the performance assessment module to analyse the whole system offline. Data transmitted is sensor-independent, and will allow testing component-wise and system-wise. |
| IF 6.1<br>IF 6.2<br>IF 6.n | To Sensors | The decisions created by the System Performance Assessment module are back propagated to the sensors. This to allow the sensors to eventually switch to alternative data processing that can only be derived knowing the performance of the higher levels. |

### 3.1.2 Information processing. Degradation levels and global platform status

The main task of the SPAM is to provide **degradation levels** and a **global platform status**. Both measures are calculated taking into account metrics of the RobustSENSE platform from the three levels of the architecture as shown in Figure 1.1:

- Sensor
- Fusion
- Understanding and planning layer

These metrics, listed in Chapter 3.1.1, give information to the System Performance Assessment Module about the performance or status of every level or single sensor that is taken into account to calculate the levels of degradation and the global platform status.

Both, the levels of degradation and the global platform status are taken into account on the Understanding and Planning layer to optimize and improve the algorithms and choose the best decision to be made during the automated driving in safe conditions. Also back propagation to sensor and fusion levels will be done in order to have a better input of the situation of the vehicle.

The degradation levels depend on the vehicle and functions available on it. This is in accordance to the platform available in every vehicle, sensors integrated and automated functions available. These automated functions will be executed in three main stages depending on the information of the SPAM:

- Degradation level 1→Full availability, no degradation
- Degradation levels 2 to n → Function executed with degradation of some features
- Degradation level n+1 → Unavailable function

Degradation levels 2 to n mean that the function will be executed, while in the same weather conditions would be unavailable without RobustSENSE, but with degradation in some aspects, as maximum speed, lower level of automation, required intervention of the driver, etc.

### 3.1.3 SPAM integration

The SPAM software will be fully integrated in the original platform without any required change in the original physical architecture, more than the interfaces needed for the communication. This software will take information available from the different CAN buses of the vehicle and will publish data on the private CAN bus devoted to development for the autonomous functions. The figure below shows how the SPAM is integrated in the CAN network of the vehicle and a sensor platform of two sensors, radar and Lidar, as an example:

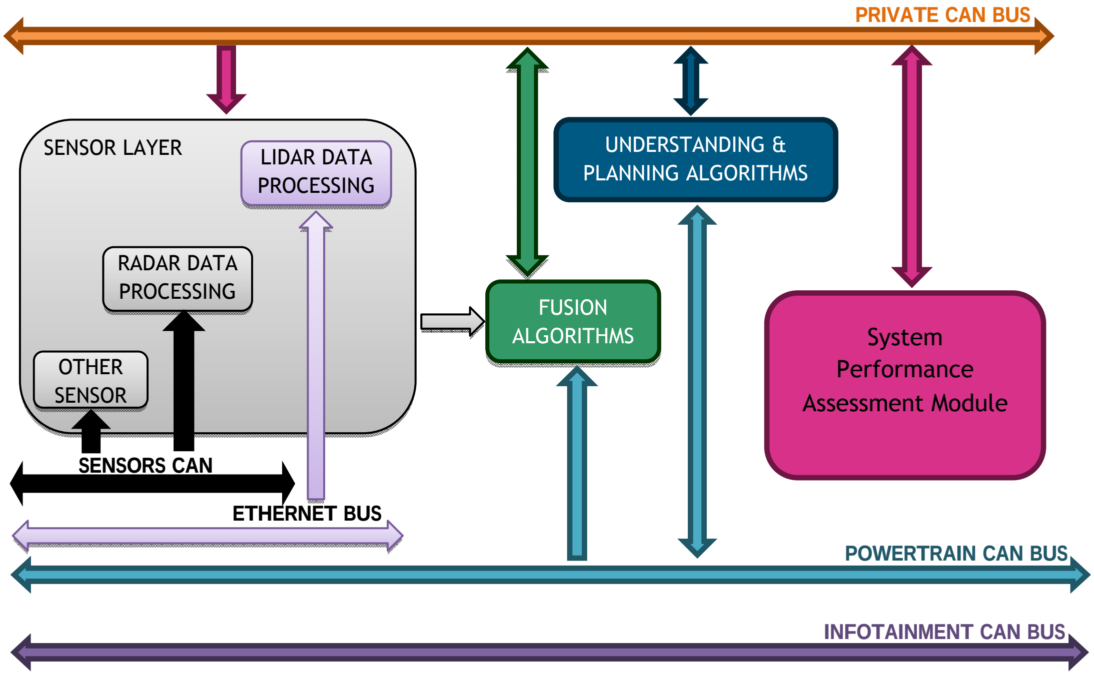



Figure 3.3: Integration of the SPAM in the CAN network

The powertrain bus provides all the information from the vehicle as speed, steering wheel angle or yaw rate. The Ethernet bus is used for the communication with the Lidar, as this is the kind of communication possible with this sensor. The sensor CAN Bus provides communication with all the sensors with this type of communication.

The communication among the other modules, fusion and Understanding and Planning, are done according to the RobustSENSE architecture. The Private CAN Bus provides the communication link between these modules and with the System Performance Assessment. All the data exchange and the output of the SPAM will be available in this bus to the other modules, including the sensor layer.

Finally, the connection of the Understanding and Planning layer with the Powertrain bus allows the platform to take the control of the vehicle during the autonomous operation.

The information on the Infotainment CAN Bus, although is part of the CAN network of the vehicle, is not used for the purpose of the RobustSENSE developments.

As shown in Figure 3.3 the integration of the SPAM in the CAN network is simple through a dedicated CAN Bus where all the information is available.

## 3.2 Monitoring Algorithm

In RobustSENSE each component monitors its status permanently and provide this information to other modules. This distribution of information is performed by the overall assessment that allows all components to access these metrics. The metrics data are used for both online assessment and offline assessment.

### 3.2.1 Logging and Monitoring

For the overall assessment, the Logging and Monitoring component is developed. The Logging and Monitoring component serves two purposes. It allows to evaluate the tests after completion, i.e. in offline assessment, and for online assessment to monitor them during execution.

As there is a heterogeneous system landscape in RobustSENSE the Logging and Monitoring facility ensures that data of different systems can be compared. The unified data collection system guarantees that all data for online monitoring and offline validation follows the same data scheme. Hence the developers of other modules, and evaluators afterwards can directly use this data, without having to align different data formats.

The Logging and Monitoring software for each component is generated automatically based on an XML definition of the measurands of each component. To this end a webpage is provided that transforms the measurands definition to a library for different platforms, such as Java VM, shared library for X86 on Linux etc. The webpage also provides an overview of the measurands of all components, as can be seen in Figure 3.1. Besides the log definitions for all components can be downloaded as well as an aggregation of all measurands.

From this platform, the Logging and Monitoring software can be downloaded. The Logging and Monitoring software of each components uses a library that gathers the measurands from all log components. This workflow is shown in Figure 3.5. Eventually, the main Logging and Monitoring tool will provide that data to other components and store it on the hard drive.

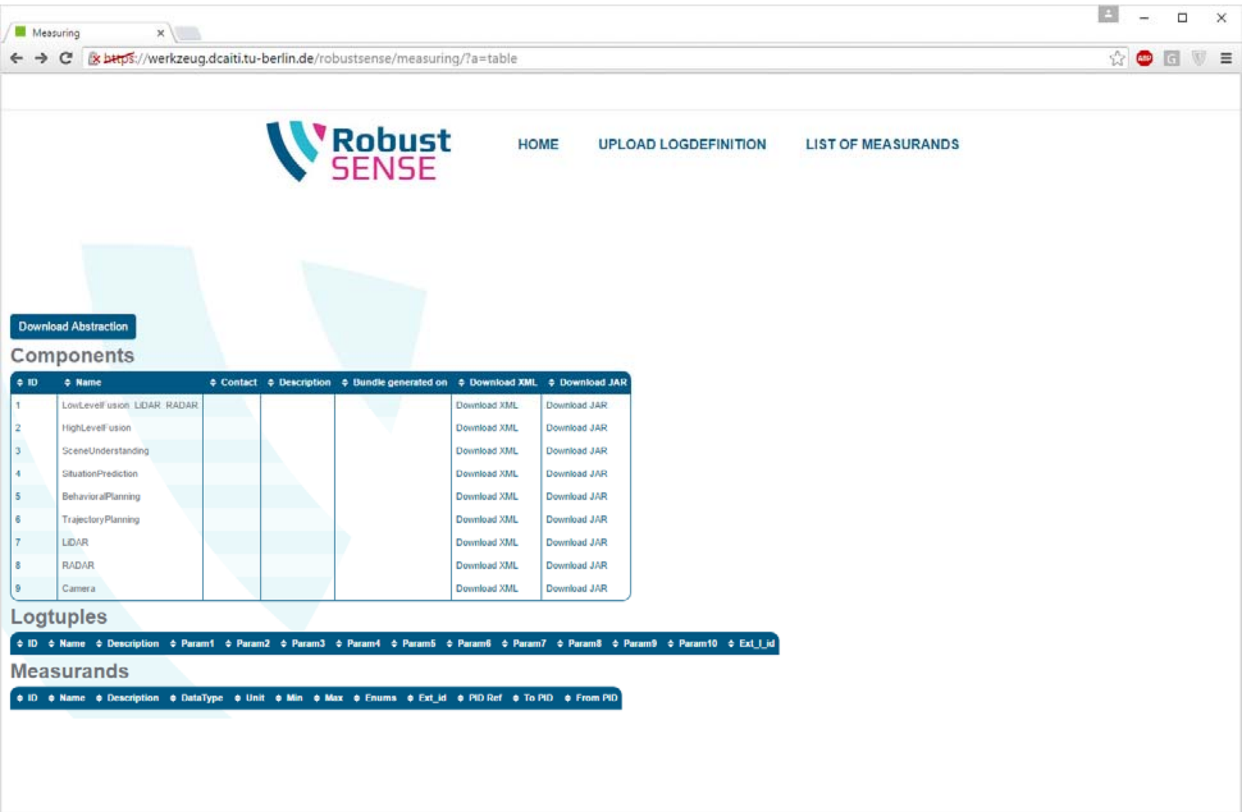



Figure 3.4: Measurands online overview

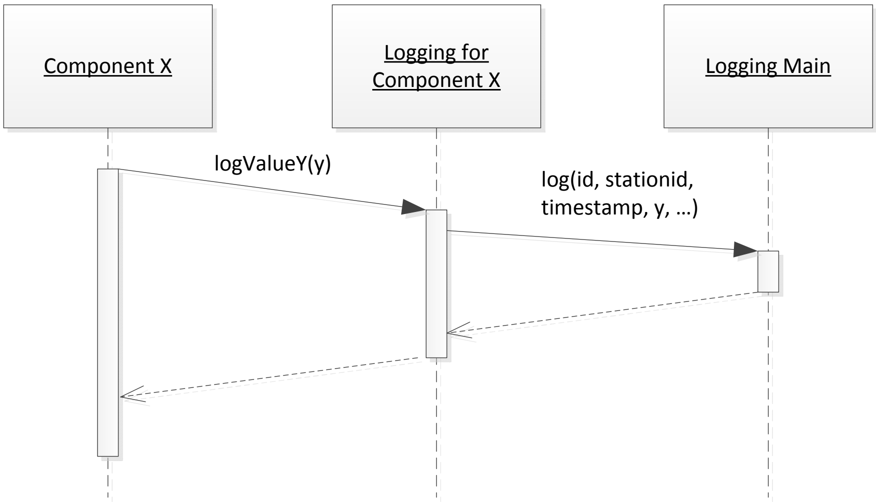



Figure 3.5: Workflow of the Logging libraries

Hence, there are two software components that are developed, one being the website that stores the measurand definitions and generates the logging software per component. The other one is the main logging library that collects the data of the generated logging libraries. The developers of the functional components, e.g. data fusion, have little to no additional implementation effort. The only task to be performed by each component developer is to define the measurands of the component and to integrate the generated library.

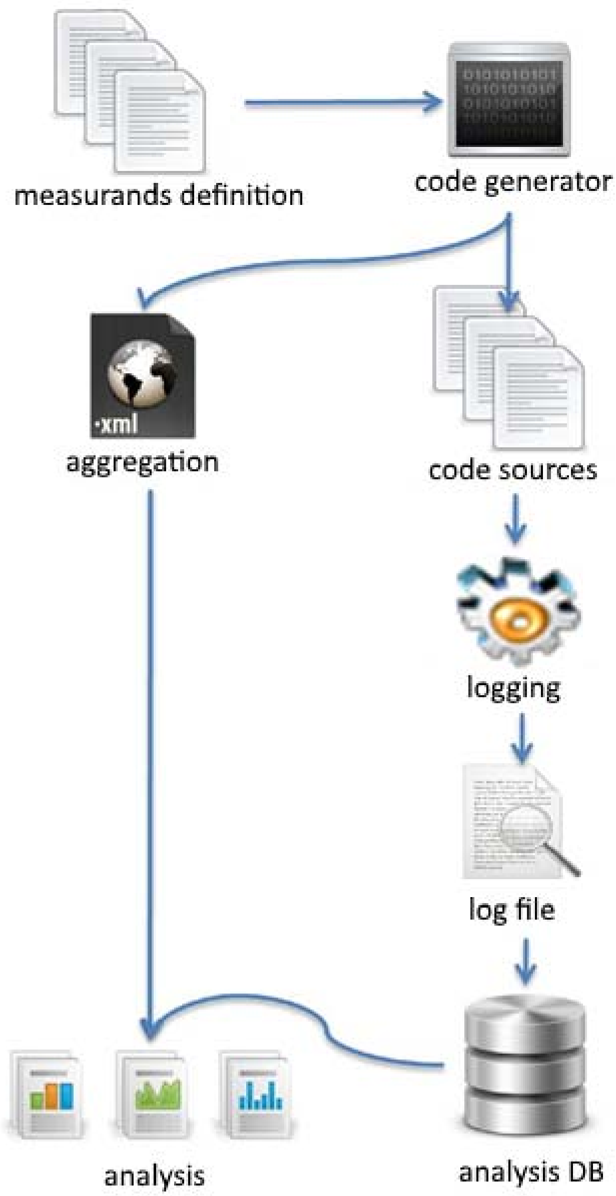



Figure 3.6: End-to-End workflow of the logging facilities

```
<logtuple>
    <name>position</name>
    <description>A geoposition, encoded as WSG84, decimal notation</description>
    <measurands>
        <measurand>latitude</measurand>
        <measurand>longitude</measurand>
        <measurand>heading</measurand>
        <measurand>speed</measurand>
    </measurands>
</logtuple>
```

Listing 1: Sample XML code for a log tuple

The overall workflow for the users of the Logging and Monitoring facilities is depicted in Figure 3.6. The developers of components define the measurands in an XML file, which they can upload to a website. The website uses the file as input for a code generator. From the code generator, the source files are created and compiled to a logging library for different platforms. The library can be used eventually by the developer of the original component. During runtime it creates log files and allows to access the metrics through monitoring. The monitoring can be used by other components to assess performance during runtime. For offline evaluation, the created log files can be transferred to a database in order to analyse the performance of the system.

```xml
<intMeasurand>
	<name>destinationId</name>
	<description>The identifier of the destination</description>
	<unit>id</unit><!-- optional -->
	<min_value>0</min_value><!-- optional -->
	<max_value>32767</max_value><!-- optional -->
</intMeasurand>
```

Listing 2: Sample XML code for an Integer measurand.

The measurand definitions can consist of single values and log tuples that comprise single values. The maximum number of values in a log tuples is ten. Single values can either be numeric, i.e. of Integer or Double type, Boolean or an Enumeration of values. Moreover, String values can be defined. However, String values cannot be added to tuples and as their automatic analysis is limited, they should only be used for exceptional values. An example of a log definition for a tuple and an Integer value are provided in Listing 1 and Listing 2.

This software can be used for online and offline assessment. For online assessment, all data that is gathered from the system is provided to all other components of the system. This is performed via a UDP interface. Using components receive all the metrics of other components through this interface from the System Performance Assessment. The receiving component can use this information for adaptation, e.g. using the reliability of a sensor in data fusion.

For offline assessment, the logging and monitoring software creates log files. All metrics from all components are stored in joint log files in temporal order. For offline assessment, the data in these log files can be analysed. The assessment can be performed directly on the files or after the import into a database. The logging and monitoring facility ensures that all metrics from all components are recorded in a uniform format. Also the logs created by different partners on different vehicles comply to this common format. Thereby in offline assessment, the data from different vehicles and different platforms can be compared.

# 4 System Performance Assessment Module validation

The correct working of the assessment processes has to be checked in order to assure that the architecture released is working correctly and it is able to fulfil the project goals. For this reason it is planned to perform a specific validation through tests done mostly at laboratory level, in order to have controlled condition which can be reproduced. Some validation will be done with vehicles, in particular for camera sensors and radars, trying to cover as much weather conditions as possible, taking in account the testing track available and the available time within the project end.

Testing methodology is outlined in Deliverable D2.2 “Metrics and validation criteria” and tests detailed descriptions and results are described in deliverable D5.2 “Laboratory and environmental tests reports” and in Deliverable D5.5 “Demonstrator validation report”. Finally, Deliverable D3.3 “Package of generic sensor models” describes the applied methodology, tools and test cases to utilize augmented generic sensor models for integration and validation purposes of RobustSENSE sensor platforms.

In order to ensure robust and reliable operation of the RobustSENSE autonomous vehicles, they must be able to handle several predefined use cases.

In the following sections, the test-cases for the sensors used inside WP5 are described. The sensors considered within the project are cameras, LIDARs and RADARs. Mapping and localization are treated as another sensor, too. Moreover, car data are used in some use cases, like in driver monitoring where are used info from car data and a specific camera to observe the driver, installed inside the vehicle. The tests will be aimed to verify that the sensors are able to provide the metrics related to their malfunctions and their uncertainties (occlusion, short time failure, etc.) and that the system is able to correctly log and handle them according the specifications.

Tests cases targeted to check sensor uncertainties and incompliant behaviour were added too, following the guidelines provided in deliverable D4.1 “Specification of Situation Understanding and Trajectory Planning Modules”.

## 4.1 Test Case 1: Camera test with Videos

Cameras performances will be tested using videos. An initial session will be performed using videos produced by simulations, to be eventually integrated and completed by collecting videos in harsh weather conditions suitable for tests. In particular it will be considered situations with dirtiness on the optics and some cases with rain. The aim of the test is to verify that the optic cleaning analysis software operates correctly and measure how much it is able to identify conditions impairing the image quality. Furthermore, it will be checked that the optical cleaning system is operated and estimated if its action permits to recover the image.

The tests will be used to check the data defined in D2.2 for the sensor performance assessment are properly generated and transmitted to the fusion layer.

### 4.1.1 Test case 1 degradation levels

As indicated in Deliverable D2.2 “Metrics and Validation criteria” if the state is not recovered by the action of the cleaning system, and taking in account also the position of the cells in the image, an image state metrics is provided to the info fusion layer and to the System Performance Assessment.

The image state can assume 3 levels: clean vision / partially working / vision impaired.

| Name | Description | Data type | Unit | Values | Used by |
|---|---|---|---|---|---|
| **ImageState** | Global state of the image | Float | 0.5 | 0 (image clean)<br>0.5 (Partially working)<br>1 (Vision Impaired) | Fusion modules |

### 4.1.2 Test case 1 degradation modelling

A scenario simulation model is used to configure and simulate, time- or event-based, all relevant interactions between static and dynamic objects (obstacles) like traffic lights, vehicles, pedestrians, etc. All dynamic objects are referenced within a road model based on the standard OpenDRIVE.

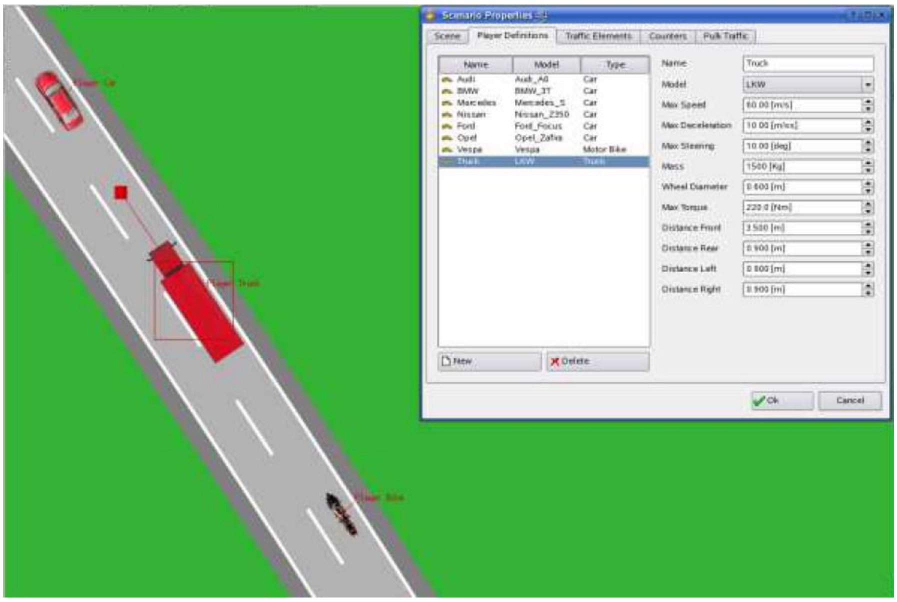


Figure 4.1: Scenario model with static and dynamic objects

Sensor models which can be parameterized are used to generate a live video data stream with a high level of realism like reflectivity of road objects and vehicles. Image distortion and blending effects can be incorporated into the sensor simulation of the camera system.

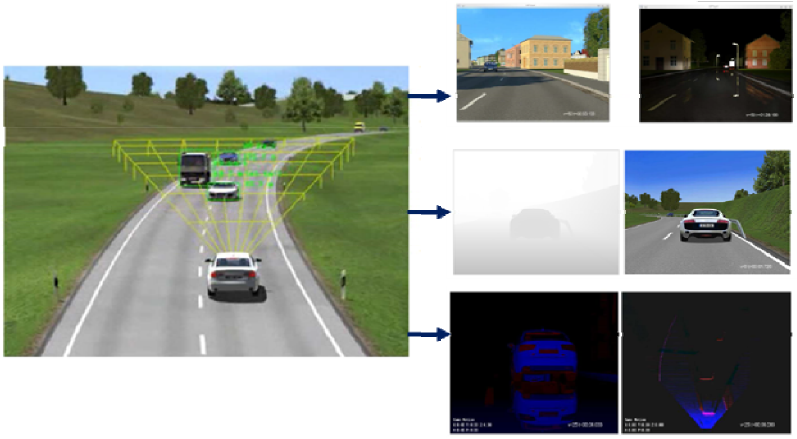


Figure 4.2: Simulated video data streams

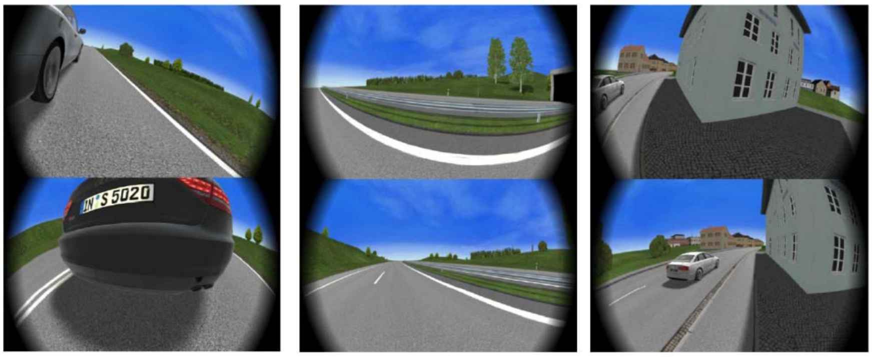


Figure 4.3: Image distortion and blending

The generic camera sensor model will be extended with plug-in functions in high-dynamic-range resolution at pixel level to simulate harsh weather conditions onto the camera like rain droplets and dirtiness on the optics.

Figure 4.4: Simulated rain condition on camera

A set of videos will be produced by simulations representing some significant traffic situations in harsh weather conditions. They will be used to check the capability of external camera sensor to identify the situation and provide the correct metrics.

Examples for traffic and weather situations are:

- High-Density traffic on multi-lane road in rainy conditions
- Low-density traffic on multi-lane road at very low visibility in snow conditions with strong side-winds
- Low-density traffic on single-lane road through chain of tunnels. Low lightning conditions in tunnels and bright sunlight outside the tunnels.

## 4.2 Test Case 2: Comparison of camera tests on vehicle and videos

The camera systems defined for the installation on vehicle will be installed on the prototype and tests will be performed in bad weather conditions. These tests will be focused in particular on verifying the side cameras, the most exposed ones, and to assess the capability to detect approaching vehicles.

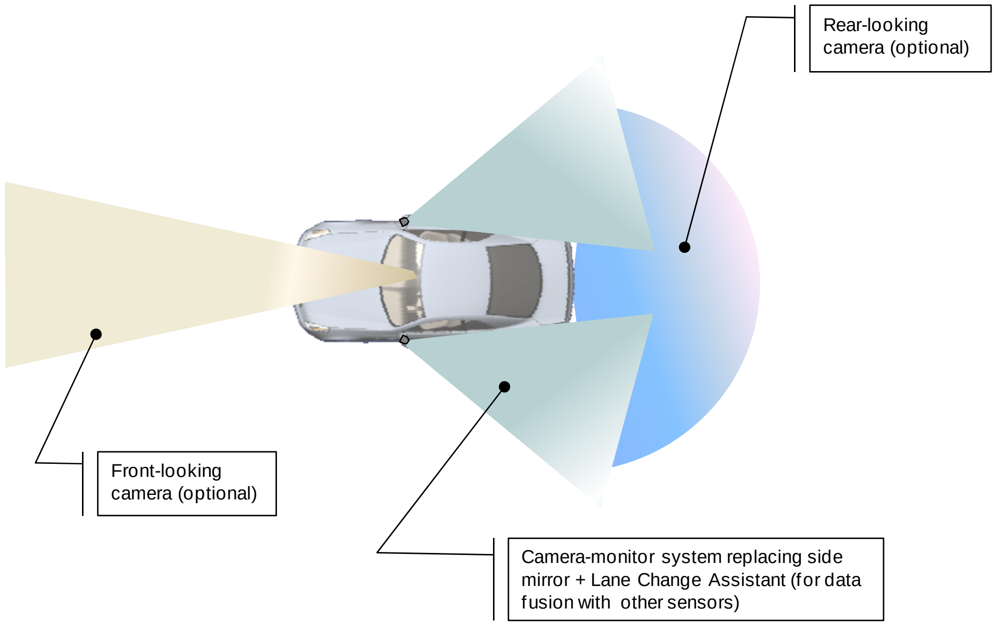



Figure 4.5: The Camera set considered for the vehicle exterior.

The performance obtained on real vehicle will be compared with the one obtained with videos. These tests will be done mostly in the weather conditions available at test time, due to the difficulty to reproduce all possible weather conditions on the available testing facilities, even if efforts will be done to try to cover as much environmental condition as possible. It should be noticed that the main purpose of these tests is to verify that the camera sensor system is working as planned and reacting coherently with the behaviour shown in the tests performed using videos, providing the correct metrics as defined in D2.2.

### 4.2.1 Test case 2 degradation levels

The three degradation levels are: properly working / partially working / not appropriate (not working at all).

### 4.2.2 Test case 2 degradation modelling

No further modelling. (The video considered for test case 1 is reproduced in realistic driving conditions).

## 4.3 Test Case 3: Interior camera system for driver monitoring

The driver monitoring system is developed in the project as a use case within WP5. The RobustSENSE new sensors concept is necessary for the upcoming semi-autonomous vehicles and for this class of vehicles a proper monitoring of the driver behaviour and intentions is very important. In this case the “robustness” is addressed by using a stereo camera able to work also in the near infrared (NIR). It allows to elaborate a 3D model of the driver face, which should make the detection of driver’s face and gaze direction more robust against head movements and illumination changes and reflections on eyeglasses. A further internal IR camera looking at the driver thorax can be used to extract from the images the driver respiration rate, which is a parameter directly related to drowsiness. These pieces of information are used, in combination with car data, to elaborate an indication about driver drowsiness and distraction. The IR camera system is installed internally to the vehicle, and thus is not exposed to the external atmospheric agents. For this reason it does not make use of the algorithm in development for the external cameras to detect image degradation due to dirt or raindrops. The stereo IR camera provides instead info about its correct internal working and detection of the driver distraction or drowsiness. This info is used internally to driver monitoring software environment, which elaborates the driver status together with a confidence level. For the driver monitoring thus it will be tested the correct electrical functionality and communication, and performed functional tests at simulator, checking the correct identification of driver distraction (eye gaze not directed to the road, driver interacting with infotainment and/or other console systems, driver drowsy). Similar tests (apart from drowsiness, due to legal issues to drive in drowsy conditions) will be performed on the demo vehicle to assure that it operates correctly and coherently with the simulator tests.

### 4.3.1 Test case 3 degradation levels

As indicated in deliverable D2.2, the interior camera system provides a single metrics about its performance:

| Name | Description | Data type | Unit | Minimum | Used by |
|---|---|---|---|---|---|
| **InteriorCamera Working** | Indicates if the Stereo Camera is up and running correctly. | Boolean | 1 | 0 | Fusion modules |

## 4.4 Test Cases for LIDAR

### 4.4.1 Test Case 4: LIDAR tests in fog chamber

The LIDAR performance will be validated in the test chamber in which fog can artificially been produced. The LIDAR performance evaluation will be conducted with installing the laser scanner to the fog chamber and thereafter, measuring strength of the light beam. The validation will be repeated in different near-infrared band wavelengths: 905, 1300, 1400, 1550, 1600 nm.

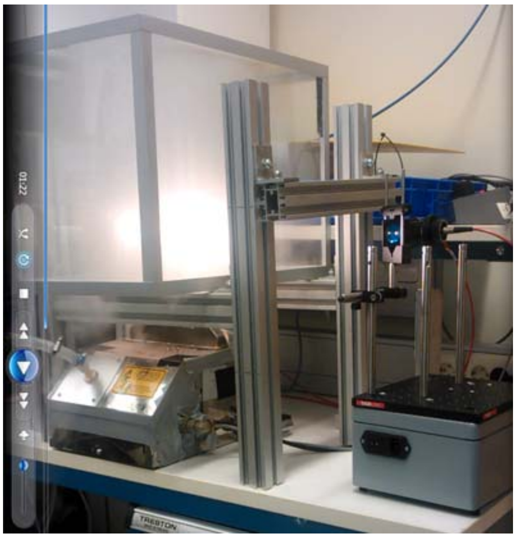


Figure 4.6: Fog chamber for validating performance of LIDAR sensor in foggy conditions

The aim of the tests is to check the amount of penetration of the laser beam and therefore, the detection range drop in different levels of fog. In practice, the test will be performed with fog densities ranging from no fog, to light and dense fog. The precise levels of fog density are still in definition at present document date in order to match some preliminary measures made in stationary conditions in real conditions in the airport of the Sodankylä, Lapland (Finland) by VTT.

The test chamber will also be used for assessing the Confidence Index value [Sawade et. al. 2016] which is calculated by the performance assessment module. The validity of the Index is estimated in laboratory conditions in five different fog density levels.

### 4.4.2 Test Case 5: LIDAR tests in real traffic

The new developed LIDAR sensor will be mounted to the front bumper of the test vehicle of VTT and the tests will be carried out for sensor performance in adverse weather conditions (rain and snow). The tests will be focused on assessing performance of detecting objects in front of the vehicles. In particular, the tested objects are pedestrians (due to the importance of detecting them and being harder to identify with respect to static obstacles) and how far ahead of the vehicle they are detected. In addition, the performance in real traffic will be compared to the measures done in the lab fog chamber (fog, rain and snow have all the effect of reducing the LIDAR range, as the static measures in real winter conditions showed). These tests are only carried out in clear weather without fog and its purpose is to validate the lab environment against air in the city.

The tests will also measure LIDAR performance assessment module against the ideal conditions. The baseline results of the performance assessment will be captured in clear day and comparison is done against the rainy and snowy days.


Figure 4.7: Validation software and facilities of the test car

## 4.5 Test Cases for RADAR

### 4.5.1 Test Case 6: RADAR tests in laboratory

The polarimetric radar sensors will be validated in a first step in an anechoic chamber in order to compare the performance (according the metrics described in Deliverable D2.2) with standard sensors and to check the quality of the sensor calibration. In a second step, the sensor performance in real world outdoor scenarios is validated.

### 4.5.2 Test Case 7: RADAR tests in real traffic

For the real traffic scenarios, one of the polarimetric radar sensors will be mounted on a test vehicle. The focus will be on clutter and noise evaluation under good weather conditions to assess the laboratory tests by comparing both. In these scenarios different targets are considered like curb stones and vegetation. The street and roadside clutter will be considered as well. All outdoor real world tests will be performed under dry conditions and wet conditions to work out the different noise and clutter levels at different weather conditions to finally assess the polarimetric sensors and the sensor performance measures.

With these real traffic tests, the sensor performance measures will be tested as well.

## 4.6 Test Case 8: Mapping and Localization tests

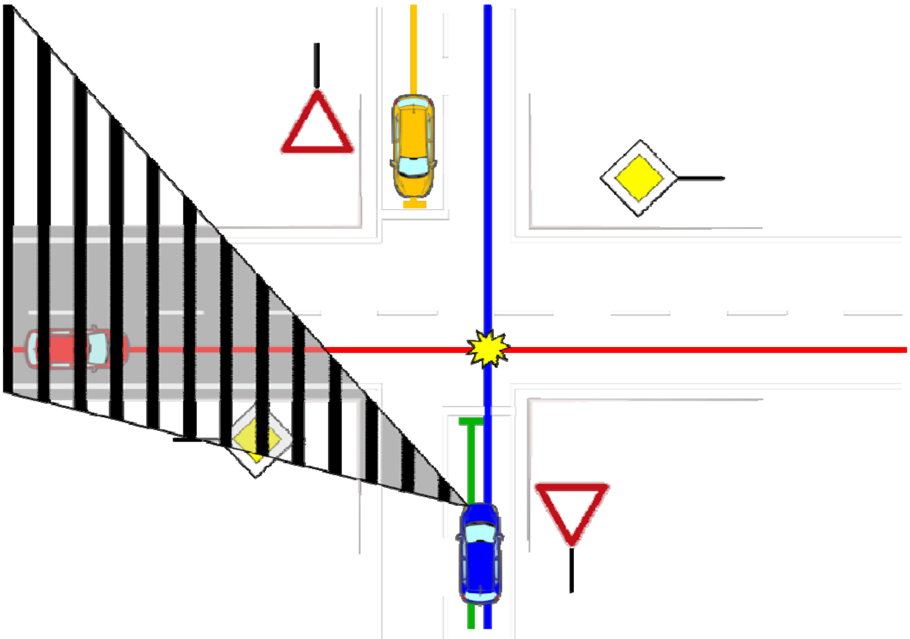
In the project it is planned to use perception based or *perception augmented* localization systems. These typically utilize vision or LIDAR sensors information combined with the ones from GNSS and IMU.

Vision based or vision augmented localization relies strongly on the illumination and weather conditions. The localization quality will decrease if these considerably differ from the conditions during mapping. Depending on the uncertainty metrics defined in D2.2, the map information can be disregarded and object fusion can rely more on the real sensor data in these cases. It is thus necessary to test on one hand the previously described performance measurements of the LIDARs and cameras and on the other hand, the localization quality in case of different weather conditions compared to the mapping process. That means generating a map of a test track on a clear day and driving again this track on a rainy, snowy or foggy day and evaluating the localization quality and the performance of updating the map in case of changes in the static environment.

## 4.7 Test Case 9: Sensor Uncertainties

In order to test the robustness gained by the Scene Understanding and Planning Layer a scenario is created that requires the system to be able to detect inconsistencies within the environment model and derive poor performance from system components.

At an intersection of a priority road with a non-priority road (see Figure 4.8), the test vehicle (blue) will be arriving from the non-priority road. One of the sensors will be temporarily failing so that one of the main street's sides is not observable. Without any other observations it would be very hard to distinguish if another vehicle (red) is on the priority road and thus safely entering the priority road would not be possible without detecting the inconsistencies of the scene.

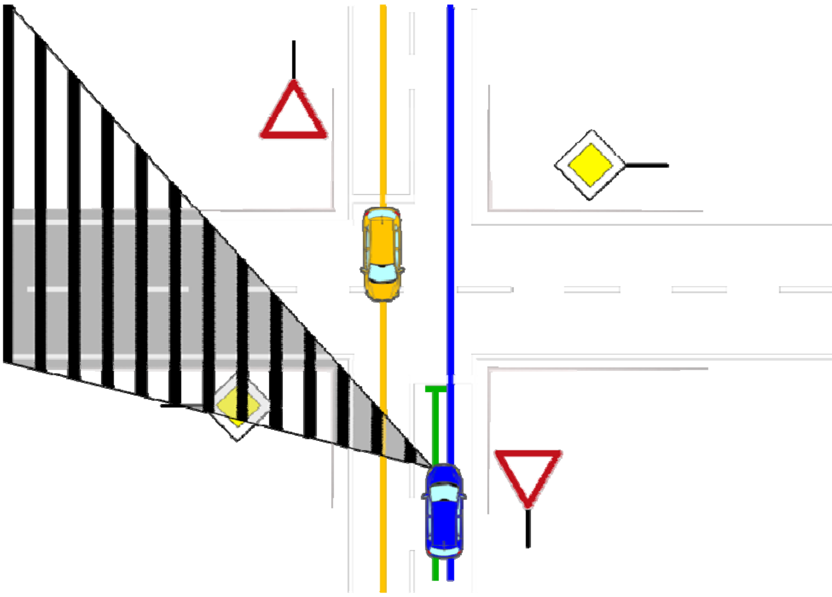


Due to a temporary sensor failure, the test vehicle (blue) cannot observe the whole intersection and is dependent on additional observations. Since the other observed vehicle (orange) is waiting, the failure of the sensor might be derived and it might be inferred that another vehicle (red) is on the priority road.

Figure 4.8: Sensor uncertainties test case.

It will be tested, if with the Scene Understanding and Planning Layer an observation of a vehicle (orange) from the opposing side road can improve robustness of the system. In the case that a vehicle is waiting on the opposing side road a robust system should be capable to infer that there has to be a reason why the vehicle is not driving. Since with high likelihood there is traffic on the priority road, on one hand the test vehicle should not enter the priority road and on the other hand, the system should even be able to derive the sensor failure. Therefore, compared to a system without scene understanding, the existence probability of the red vehicle should increase and the behaviour uncertainty of the orange vehicle should decrease because of the consistent scene estimation. The situation prediction should anticipate that with a high probability the red vehicle will cross the intersection.

In the case that the opposing vehicle is entering the street, the test vehicle should derive, that the priority road is free with a high likelihood (see Figure 4.9). However, the necessary boundary conditions have to be evaluated carefully before entering the intersection. In this way the unobserved area can be estimated, even if the system is already aware of the sensor failure and assumes it could be missing something on the priority road. Compared to a system without scene understanding the existence probability of a car in the unobserved area should decrease and the probability of a free road should increase. Again, the behaviour uncertainty of the orange vehicle should decrease because of the consistent scene estimation. In both, behaviour and trajectory planning, the costs for planned solutions should decrease because the plans are less conservative and more comfortable, as the blue vehicle is not waiting any longer.



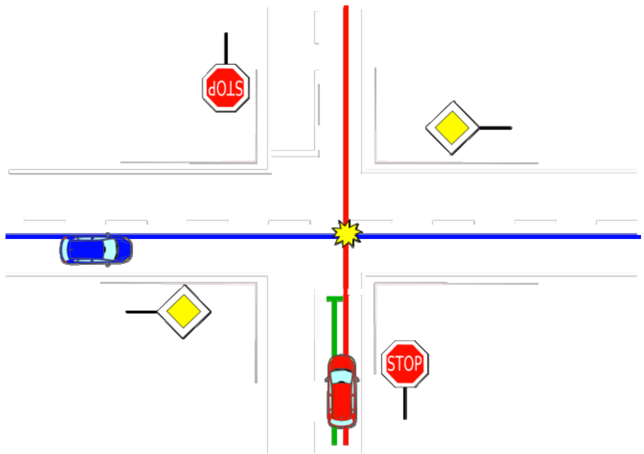
The same initial situation as in

Figure 4.9 Sensor uncertainties test case: opposing vehicle entering the street.

In both cases (Figure 4.8, Figure 4.9) the overall uncertainty should decrease or decrease faster over time than without scene understanding. Based on the overall uncertainty value the System Performance Assessment module is able to evaluate the performance of the Scene Understanding module. The higher the uncertainty the less consistent the scene and vice versa.

## 4.8 Test Case 10: Non-compliant Behaviours

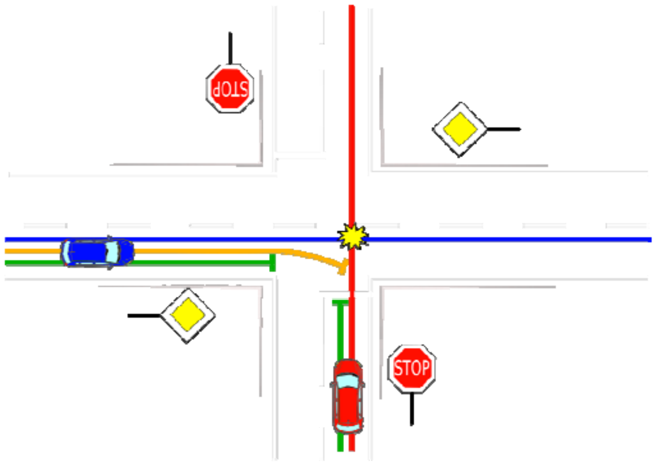
The lower layers cannot handle non-compliant behaviours explicitly. To test if the Scene Understanding and Planning modules are able to detect possible driving behaviours and estimate the compliance of traffic participants a test scenario is created. In this scenario (see Figure 4.10), the test vehicle (blue) is driving on a priority route while a car (red) is arriving on a side route. Usually the red car is predicted to stop (green) if it recognizes the test vehicle. Nevertheless, in the case that the red car cannot detect the test vehicle or its driver has a non-compliant driving style, it may enter the intersection resulting in a non-compliant behaviour.




Although the test vehicle (blue) is driving on a priority road, the other vehicle (red) might enter the intersection.

Figure 4.10: Non-compliant behaviour I.

It is tested if with the Scene Understanding and Planning Layer the test vehicle is aware of such situations and detects the non-compliant behaviour as early as possible in a probabilistic manner. The test vehicle should consider the non-compliance of the red vehicle's driver for prediction and planning. Safe manoeuvre plans that also consider the case of non-compliant behaviour have to be evaluated (see Figure 4.11). The planned trajectories should always lead to save states, but do not lead to too conservative behaviours.




Non-compliant behaviour of the other vehicle (red) has to be considered when planning manoeuvres. This should lead to save states, but not to too conservative behaviour.

Figure 4.11: Non-compliant behaviour II.

Compared to a system without the Scene Understanding and Planning layer the non-compliance of the red vehicle should be considered. In Scene Understanding, the existence probability of the incompliant behaviour (“Don’t Stop”-behaviour) should increase, especially if the red vehicle does not slow down before the intersection. In Situation Prediction, this should lead to an increased probability that the red vehicle will cross the main road. Behaviour Planning should consider the two likely behaviours of the red vehicle and provide two planned behaviours for the test vehicle with proper probabilities based on the likelihood of non-compliant behaviour: Either stop at the intersection or drive through. Trajectory planning should then be able to at least select the most conservative behaviour and plan a proper trajectory. Additionally, it can be able to consider both behaviours of the red vehicle and plan a trajectory that allows driving through the intersection but is still able to lead to a safe state, if the red vehicle enters the intersection.

Again, the overall uncertainty is a good metric for the performance of the Scene Understanding module: If the module is able to identify clearly one of the two behaviours of the red vehicle (e.g. the red vehicle is too fast to stop before the intersection), the overall uncertainty will decrease. If the two behaviours of the red vehicle however have almost the same probability, the overall uncertainty will increase and the Scene Understanding module can be estimated as less performing.

Additional scenarios for the validation of the system performance related to uncertainty and non-complaint behaviour are head-on collision types. These scenarios appear in two typical variations:

- Overtaking in oncoming traffic: A vehicle in the oncoming traffic is overtaking late or with too less difference speed, thus creating a risk not to finish the overtaking manoeuvre as shown in Figure 4.12. The robust sensor platform has to provide the necessary environmental information to support the optimum trajectory planning for crash avoidance.

Figure 4.12: Overtaking in oncoming traffic

- Wrong-way driver on multi-lane highway: A vehicle has entered the highway in the wrong direction and keeps on driving. The robust sensor platform has to provide the necessary environmental information to support the optimum trajectory planning for crash avoidance.

Figure 4.13: Wrong-way driver on highway

# 5 Conclusions

The goal of RobustSENSE is to develop a sensor platform able to operate under harsh conditions extending the operational range beyond the current sensors limitations. In order to achieve this at all times, the system must be able to self-assess its current status and correct operation. To assess the status of the overall system, each single component must assess its own operation. This information is used by other components to adapt to degrading sensor performance or adverse conditions. For that the available metrics of each component must be known to all other component.

In this document, it is described how the performance elaborated at each level are combined and monitored at system level in order to assess the overall system performance. An initial plan to test the overall assessment system is provided as well.

# References

Sawade, O., Schäufele, B., Kutila, M., et. al. (2016). Metrics and validation criteria. RobustSENSE deliverable 2.2.

# List of abbreviations and acronyms

| Abbreviation | Meaning |
|---|---|
| GNSS | Global Navigation Satellite System |
| IMU | Inertial Measurement Unit |