D 6.4 – DEFINITION AND PLANNING OF EVALUATION TRIALS

Authors Marta Kasprzak (RTC), Anna Olejniczak-Serowiec (RTC), Adam Dąbrowski (RTC), Wojciech Jaworski (RTC), Alessandro Zanella (CRF), Gerasimos Arvanitis (UPAT), Aris Lalos (ISI), Apostolos Fournaris (ISI), Pavlos Kosmides (CTL), Petros Kapsalas (PASEU)

Work Package WP6 – Industry Driven Trial and Evaluation

Abstract

This report contains the output of Task 6.1 which lays the ground for the evaluation trials. It consists of designing and planning the way the pilot studies will be organized, supported, and managed throughout the duration of the project. The deliverable is organized around two use case groups. First one Human-Robot Interaction in manufacturing environment includes following use cases: a design operation continuum evaluation and resilience and safety. Second group of connected and autonomous L3-L4 vehicles use cases consists of: human in the loop control, cybersecurity issues and cooperative awareness.

Funded by the Horizon 2020 Framework Programme
of the European Union

Deliverable Information

Work Package	WP6 Industry Driven Trial and Evaluation
Task	T6.1 Pilot trials specification and assessment protocol
Deliverable title	Definition and planning of evaluation trials
Dissemination Level	Public
Status	Final
Version Number	2.0
Due date	30/06/2021

Project Information

Project start and duration	1.01.2020-31.12.2022
Project Coordinator	Industrial Systems Institute, ATHENA Research and Innovation Center 26504, Rio-Patras, Greece
Partners	1. ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS (ISI) - Coordinator 2. FUNDACIO PRIVADA I2CAT, INTERNET I INNOVACIO DIGITAL A CATALUNYA (I2CAT) 3. IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD (IBM ISRAEL) 4. ATOS SPAIN SA (ATOS) 5. PANASONIC AUTOMOTIVE SYSTEMS EUROPE GMBH (PASEU) 6. EIGHT BELLS LTD (8BELLS) 7. UNIVERSITA DELLA SVIZZERA ITALIANA (USI), 8. TAMPEREEN KORKEAKOULUSAATIO SR (TAU) 9. UNIVERSITY OF PELOPONNESE (UoP) 10. CATALINK LIMITED (CATALINK) 11. ROBOTEC.AI SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA (RTC) 12. CENTRO RICERCHE FIAT SCPA (CRF) 13. PANEPISTIMIO PATRON (UPAT)
Website	www.cpsosaware.eu

Control Sheet

Version	Date	Summary of changes	Author
0.1	13/05/2021	Table of Content distributed to the Consortium	*Michał Niezgoda (RTC)*
1.0	17/06/2021	Final version for internal review	*Marta Kasprzak (RTC),* *Anna Olejniczak-Serowiec (RTC),* *Adam Dąbrowski (RTC),* *Wojciech Jaworski (RTC),* *Alessandro Zanella (CRF),* *Gerasimos Arvanitis (UPAT),* *Aris Lalos (ISI),* *Apostolos Fournaris (ISI),* *Pavlos Kosmides (CTL),* *Petros Kapsalas (PASEU)*
1.1	24/06/2021	Reviewed version	*Pekka Jääskeläinen (TAU), Pavlos Kosmides (CTL)*
2.0	29/06/2021	Final version after review	*Marta Kasprzak (RTC)*

	Name
Prepared by	RTC
Reviewed by	CTL, TAU
Authorised by	RTC

Date	Recipient
29/06/2021	Project Consortium
30/06/2021	European Commission

Table of contents

List of figures

List of tables

List of acronyms

ADAS	Advanced Driver Assistance Systems
ADS	Autonomous Driving Systems
AR	Augmented Reality
CAN	Controller Area Network
CAV	Connected Autonomous Car
CL	Cooperative Localization
CPSoS	Cyber Physical System of Systems
CPS	Cyber Physical System
CPU	Central Processing Unit
DDAW	Driver Drowsiness and Attention Warning
DDoS	Distributed Denial of Service
DGPS	Differential Global Positioning System
DMH	Digital Human Model
DoS	Denial of Service
DSM	Driver State Monitoring
EAWS	Ergonomic Assessment Worksheet
ECU	Electronic Control Unit
EEG	Electroencephalography
EKF-CA	Extended Kalman Filter for Cooperative Awareness
ERP	Enterprise Resource Planning
FAS	Fatigue Assessment Scale
FCA	Fiat Chrysler Automobiles
FPGA	Field-programmable Gate Array
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HMI	Human-Machine Interface
HRC	Human-Robot Collaboration
HST	Hardware Security Token
HW	Hardware
IMU	Inertial Measurement Unit
IOU	Intersection Over Union
IVR	Interactive Voice Response
KPI	Key Performance Indicator
KSS	Karolinska Sleepiness Scale
LiDAR	Light Detection and Ranging
MES	Manufacturing Execution System
MR	Mixed Reality

MSIN	Mobile Subscription Identification Number
NIR	Near-infrared
NN	Neural Network
OBD	On-Board Diagnostics
OBU	On-Board Unit
OCRA	Occupational Repetitive Actions
OEM	Original Equipment Manufacturer
OGM	Occupancy Grid Map
OS	Operating System
OWAS	OVAKO Working posture Analysing System
PLC	Programmable Logic Controller
POV	Point Of View
RMS	Root- main-square
ROS	Robot Operating System
RTK	Real Time Kinematic
RULA	Rapid Upper Limb Assessment
SAE	Society of Automotive Engineers
SMS	Short Message Service
SRMM	Security Runtime Monitoring Mechanism
SSH	Secure Shell
SSP	Static Strength Prediction
SSQ	Simulation Sickness Questionnaire
SW	Software
TCP/IP	Transmission Control Protocol/Internet Protocol
TCU	Telematic Control Unit
TPMS	Tyre Pressure Monitoring Systems
UI	User Interface
VANET	Vehicular Ad Hoc Network
VR	Virtual Reality
VTA	Virtual Task Analysis
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
XR	Extended Reality
XRT	Xilinx Runtime library
XVR	Extended View Representation

Executive Summary

The following deliverable, D6.4 - Definition and planning of evaluation trials, describes the way in which certain CPSoSaware components will be tested and validated. Five use cases: a design operation continuum evaluation, resilience and safety, human in the loop control, cybersecurity issues, and cooperative awareness are considered. For every use case the following elements of the trials are discussed:

- Testing environments,
- Testing procedures,
- Data acquisition,
- Research instruments,
- Outcome measures.

These elements explain precisely in what environments the components will be tested, in what way, what data will be acquired in the process, what methods and instruments will be used, and what measures will be gained in the effect. The descriptions of how the evaluation phase for every use case is planned and conducted, and eventually conducting the trials, allow to create a robust and effective system.

1 Introduction

As the outcome of D6.1, which proposed testing methods based on the research literature, industrial standards, and state-of-the-art methods, the following report introduces selected methods and detailed description of definition and planning of pilot studies. The evaluation trials use cases are concentrated around two areas: human-robot interaction in manufacturing environment, with the following use cases: a design operation continuum evaluation, and resilience and safety; and connected and autonomous L3-L4 vehicles, with the use cases: human in the loop control, cybersecurity issues, and cooperative awareness. Comprehensive descriptions of how the evaluation phase is planned and conducted are provided for every use case, describing evaluated CPSoSaware components, testing requirements, testing environments, procedures, data acquisition, and the outcome data. The objective of the evaluation trials is to test the robustness and effectiveness of the CPSoSaware system components in dedicated environments to ensure the best system's performance.

1.1 Document structure

Section 1 introduced the document and explains its structure.

Section 2 describes the evaluation trials divided in two subsections: *Human-robot Interaction in Manufacturing Environment* and *Connected and Autonomous L3-L4 Vehicles*.

Subsection *Human-robot Interaction in Manufacturing Environment* describes two use cases, each presented in separate subsection:

- A Design Operation Continuum Evaluation,
- Resilience and Safety.

Subsection *Connected and Autonomous L3-L4 Vehicles* describes two use cases, each presented in separate subsection:

- Human in the Loop Control,
- Cybersecurity Issues,
- Cooperative Awareness.

Section 3 concludes the document.

2 Evaluation trials

This section includes the descriptions of every of five use cases, providing the information about the evaluated components, environments in which they are tested, the validation procedures, and data acquisition methods and outcomes.

2.1 Human-Robot Interaction in Manufacturing Environment

The manufacturing scenarios are using different CPSoS technologies mainly distributed between two main use cases:

- The first use case is the design operation continuum evaluation. In this specific use case, the Human-Robot Collaboration system is considered, together with all the information which are necessary for its execution and that arrive from all the software systems above the field level in the use case (e.g. MES = manufacturing execution system or line controllers). In the Design Operation Continuum Evaluation, the system will emulate the need of a change of part numbers. After this event occurs, the system will need to define which modification in the workplace or in the operator's activities will become necessary. The operator will be then invited to work wearing a set of Augmented Reality goggles in order to be guided in a session of training on the job.
- The second use case, namely Resilience and Safety, will consider devices defined in the CPS system which are monitoring and defining set of postures and behaviors that might generate dangerous or non-ergonomic situations to the operator.

For both use cases, the implementation will take place in the pilot in CRF premises. Parts of the developments will be executed and tested at the premises of the collaborating partners as well. Differently to the approach declared in D6.1 for the validation of the use cases, the use of surveys, with involvement of operators in the manufacturing plants is not feasible mainly due to COVID restrictions. Because of this, the tests will be carried out internally by researchers of the involved partners while testing and validating the functionalities of the system in the two use cases. Surveys will be defined to evaluate acceptance and perceived performance, but the operators will be identified mainly among internal CRF researchers. Surveys defined in D2.1 will be here used.

In the following paragraphs the related CPS technologies, testing environment, and testing procedures that will be implemented in the pilot are described.

Laboratories involved in CRF are the Robotics Laboratory and the Virtual Reality Lab. In the Virtual Reality Lab the cell is reproduced virtually and it will be used to make preliminary development of the SW modules, test the ergonomics, and acquire the reference safe positions for the robot.

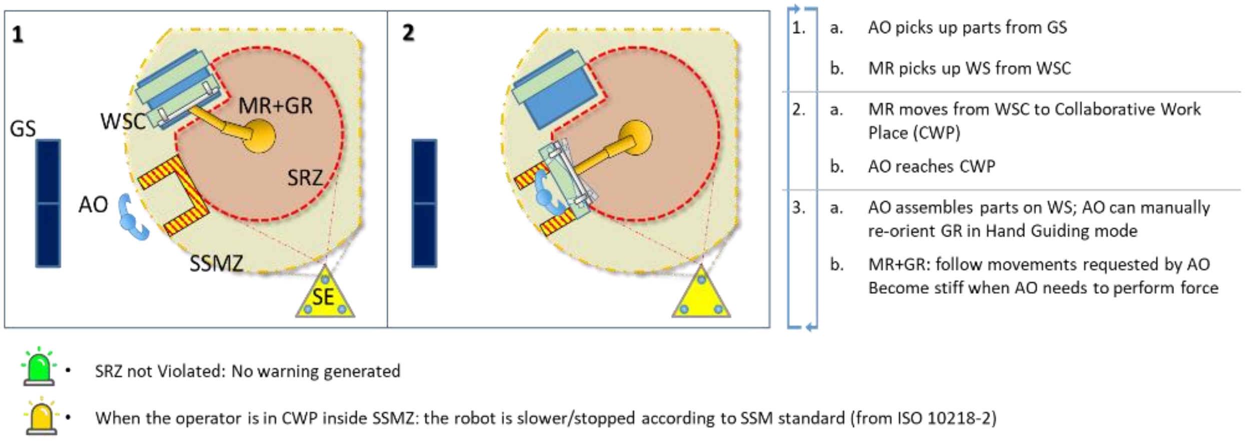
2.1.1 A design operation continuum evaluation

The use case description presented in D6.1 is still valid, however certain updates and modifications have been made concerning the details of the definition. These updates are included in this deliverable. In this use case, we consider and simulate the substitution of a Part Number in the assembly line of an automotive manufacturing factory with a new model in which a specific vehicle model is developed through an HRC approach. The workcell, described in details in D1.2 - Requirements and Use Cases (Zanella et al., 2021), is equipped to assemble the rear-view mirror, two type of sensors, and the harness on top of the windshield in a collaborative application.

In the pilot the installation of a different model of windshield will be performed, simulating a new set of supplies with different characteristics. There will be two assembly sets characterized by different conditions and assembly procedure. This new event will represent a perturbation of the design operation and trigger action in order to provide continuity to the application.

A new set of instructions will be thus required for the operator, and the change in the requested procedures will start the adaptation procedure, which is as follows:

- Update of the set of instructions.
- Transmission to the AR/XR wearable device on the updated instruction.
- Execution of the new procedures while monitoring the operator's actions.
- Continuous update of the state with evaluation of the training performance (cycle time).
- Stop of the training on the job when the result is considered satisfactory (by the operator, system, input limitation, and others).



1. a. AO picks up parts from GS
   b. MR picks up WS from WSC
2. a. MR moves from WSC to Collaborative Work Place (CWP)
   b. AO reaches CWP
3. a. AO assembles parts on WS; AO can manually re-orient GR in Hand Guiding mode
   b. MR+GR: follow movements requested by AO Become stiff when AO needs to perform force

- SRZ not Violated: No warning generated
- When the operator is in CWP inside SSMZ: the robot is slower/stopped according to SSM standard (from ISO 10218-2)

Figure 1: Normal operation cycle

The Training Scenario thus introduced is a modification, executed above the standard scenario layer.

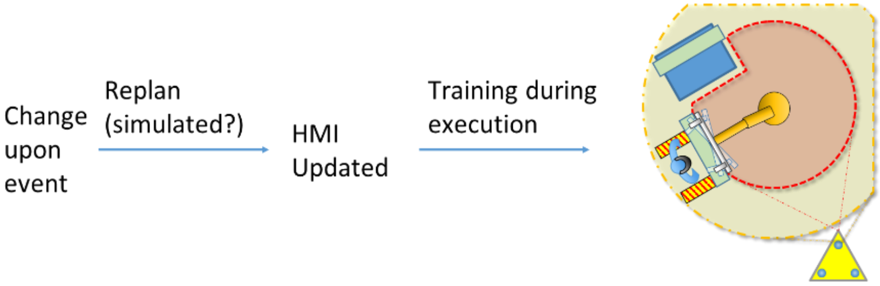


Figure 2: Training on the job operation cycle

2.1.1.1 Evaluated CPSoSaware components, related requirements and evaluation concept of the targeted components and use case

The VR Training Tool (TC5.3.1) is a knowledge exchange tool that allows the transfer of context-aware training from the physical space into a virtual reality environment. A key feature of this tool is the modelling of robot's behavior for Human Robot Collaboration training scenarios in virtual reality environments. The main requirement of this tool is the 3D representation of the workspace environment, acquired from 3D scanning devices or 3D modelling software. Semantic data is labelled on the 3D models, indicating information about the functionality, kinematics, and human interaction with the machines and tools of the workspace. A designer can then create a digital twin of the environment simulating the real machines and tools behavior in which a tutorial session will be performed. VR Training Tool comprises two distinct modules, the "Build" module and the "Train" module.

The Build module is initiated by first loading a previously designed and labelled 3D workplace of interest. A skilled user can use it for the creation of a training tutorial in order to train an inexperienced user on the use of machines or industrial control panels in simple or more complex scenarios, avoiding dangers and risks inherent during the real (physical) job assignments. Once the tutorial creation session starts, the user interacts with the different machines and tools of the workspace, while every task and process performed is tracked through a recording mechanism. After they finish the desired tutorial steps, the tutorial is encoded and saved.

The Train module is initiated by loading a 3D workplace of interest, as before, and loading a tutorial file containing all the previously recorded steps of the training process. An inexperienced user gets trained on a recorded scenario by interacting with the components of the virtual environment to perform certain tasks and processes. They are guided through visual indications and animations of the interactions performed to achieve their goal.

The evaluation of the VR Training Tool will be done by the users through Quality of Presence questionnaires or performance comparison between classic and virtual methods.

Training in manufacturing applications starts from a complete VR environment, where user can learn the processes, procedures, cycle times, and workplace from the scratch. Every element has to be digitized and proposed to the user so that he can manipulate objects and follow the virtual procedures in order to execute the work as activity, as if he were in the real workplace. Evaluation procedures will monitor user acquired awareness and will give feedback on his preparation to the next step of the training in a mixed reality environment.

When this first step is completed, the training continues in the mixed reality (MR) environment, which is a combination of real physical and virtual objects. In this step the user is guided by a virtual UI but the working tools and equipment are real, so the user can perform the activity with little help. The system will recognize the controlling UI touch events trends, time spent on UI interaction, and when the UI is still needed. The evaluation procedure will maintain some common features with the VR evaluation procedure to assure continuum of evaluation.

Within the MR experience the continuous training is assured by the implementation of Edit feature, which makes it possible to modify some elements of the training applications. A new set of instructions will be thus required for the operator, and the change in the requested procedures will start the adaptation procedure. The procedure is as follows:

- Update of the set of instructions.
- Transmission to the AR/XR wearable device on the updated instruction.
- Execution of the new procedures while monitoring the operator's actions.
- Continuous update of the state with evaluation of the training performance (cycle time).
- Stop of the training on the job when the result is considered satisfactory (by the operator, system, input limitation, and others).

The logical cycle of the training that will be integrated and developed is briefly described in the Figure 3 below.

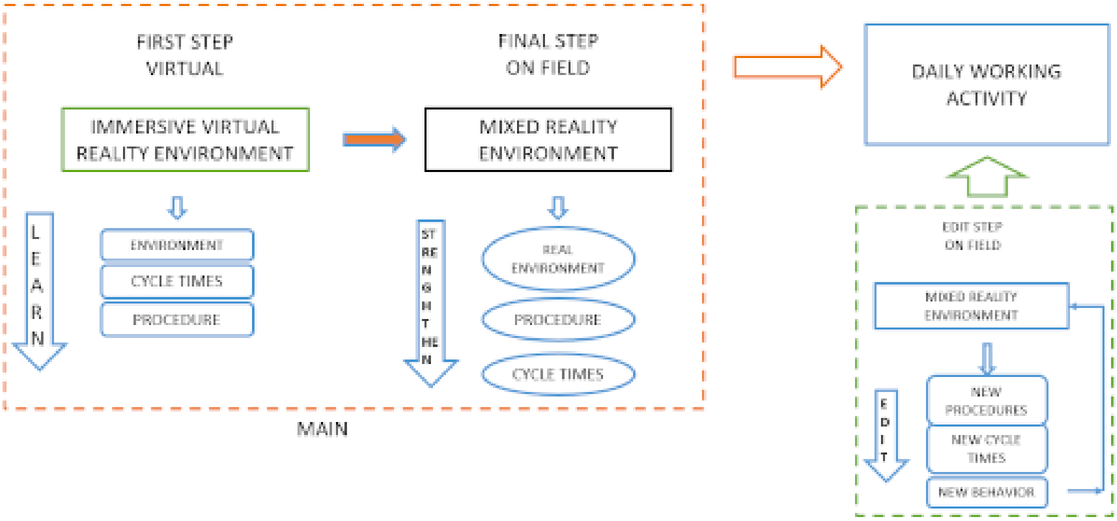


Figure 3: Training process

In this use case, the whole part related to augmented or mixed reality will be applied for the training and instruction phase of the collaborative robot. This part is still in the development and definition phase.

The use of Hololens 2[1], a state-of-the-art tool in the field of mixed reality, will help the operator in performing the tasks. Various processes are linked together, according to the project's targeted integration of different intelligent and collaborative systems.

This technology employs holograms (3D images with correct prospective and POV) projected on glasses in order to see and interact with augmented elements within a real environment.

[1] https://www.microsoft.com/en-us/hololens

The methodology must include high level of interaction between real and virtual environments assuring:

- Picking objects and their virtually assembly to the engine.
- Spatial mapping.
- Synchronized moving of real parts and holograms.

The features that could be implemented are:

- Picking objects gestures hololens/clicker/joystick/gloves.
- Animation Synchronism holograms vs real environment/Collisions.
- Collision Feedback visive/acoustic/physic.
- Physic: gravity and rigid body dynamic.
- Additional info by markers/tags (typical of ARenv.).

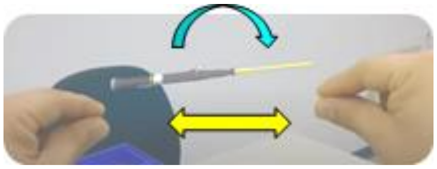


Figure 5: 3D object mapping

Figure 4: Spatial mapping

2.1.1.2 Testing environments

The pilot workcell is described in details in D1.2 - Requirements and Use Cases (Zanella et al., 2021) and in particular in D6.3 - Preliminary Evaluation and Assessment of CPSoSaware Platform (Genchi et al., 2020; Chapter 3.1 - Use Case and Pilot description). In D6.3 the pilot is detailed and different scenarios are described. The base scenario is listed in Figure 1, while Figure 2 describes the executive scenario for the XR training application in the use case Design Operation Continuum Evaluation. When the system enables a Training Assistant Mode, the training assistant is superimposed to standard and additional scenarios with AR/MR goggles for assistance to operator's procedures. It is enabled in case of:

- Product/process modification along the information path.
- New operators (initial training).
- Other cases.

The testing environment will be mainly based on two different types of environments: Physical Workcell and VR rooms (and Virtual scene).

2.1.1.3 Validation procedure

As previously stated, differently to the approach declared in D6.1 for the validation of the use cases, the use of surveys with involvement of operators in the manufacturing plants will be limited to internal personnel. Tests will be conducted internally by researchers of the involved partners while testing and validating the functionalities of the system. Surveys will be defined to evaluate acceptance and perceived performance, but the operators will be identified mainly among internal CRF researchers. Surveys defined in D2.1 - Human Factors and Metrics Analysis (Gerosavva et al., 2021) will be here used upon written informed consent. As for the rest of the validating environment and procedures, proper hardware tools will be defined in the workcell for the validation of the main KPIs that will be defined in the project. As for the Manufacturing use cases most of the KPIs will be derived by analysis of the execution time, and eventually performance (in terms of successful recognition over total analysis) of the installed devices.

2.1.1.4 Data acquisition

Data for this use case evaluation will be collected during actual use of the cell with specific functional tests performed by multiple operators in the cell equipped with the different sensors. Data will be collected by logging of connected devices and cell's safety components, as listed below:

- SafetyEYE system: Safety Zone Violation (TCP/IP strings).
- Hololens.
- PLC.
- Cell status.
- Anthropomorphic/postures detection Camera.
- Logistics requests (from Gravity Shelf or dedicated HMIs).
- Tracking of Robot Controller and Robot's Safe system.

2.1.2 Resilience and safety

The following section describes how the resilience and safety components will be validated in the pilot tests and how the tests will be conducted. Subsection 2.1.2.1 introduces the requirements and concepts of the tests, subsection 2.1.2.2 provides the information on testing environments, and subsection 2.1.2.3 explains the testing procedures and data acquisition.

2.1.2.1 Evaluated CPSoSaware components, related requirements and evaluation concept of the targeted components and use case

In an industrial manufacturing environment, cybersecurity, safety and robustness/resilience are critically important.

Normally, cybersecurity at device and workcell level is achieved by access protection and guaranteed by the use of secured technologies. At higher levels, from edge computing to local FOG, and higher to external clouds and systems (ERP, MES or other SW), the cybersecurity is guaranteed in diverse modes. The CPSoSaware project deals with cybersecurity at CPS device level, while in manufacturing the approach is more centralized, even though any device and connection is requested to be inherently secure. In the manufacturing use case of the project, the aspects which are more relevant to the system's resilience are usually related to unexpected human behavior and unexpected process behavior. While unexpected process behavior might be related to lack of supplies, presence of scraps, quality errors management and so on, the unexpected human behavior is often related to Safety.

Usually, unexpected behavior is approached by the use of operative procedures that avoid the possibility of these events to occur, but this is not always possible. For this reason, proper risk assessment procedures are usually defined.

In CPSoSaware, the low level HW and SW safety related to the robotic cell is managed according to current standard safety rules, and information from sensors is only eventually monitored and copied as additional input information. On the other end, additional tools and devices are included to support the operator safety and ergonomics.

Use case: Anthropometrics-based robot height adaptation for ergonomics optimization based on posture recognition

In industrial manufacturing environments, robustness, resilience, and safety are very important both for the good functionality of operators and for robots. The human worker’s health and the robotic equipment must be protected against situations or errors/failures that may unexpectedly appear, causing an industrial accident or stopping the production line. In this use case scenario, we focus on the aspect of operators' safety trying to reduce their body’s strain by performing biophysics assessment for the ergonomic optimization. The use case involves the utilization of all CPSoSaware components that are included in the provided workplace, like the surveillance cameras, the robot, and the operator HMI device.

The continuous monitoring of the operators while they are working is a critical element for their protection from injuries and muscle strain in this scenario. The response of the proposed implementation must focus on two directions:

- Adjusting the position of the windshield based on the operator's ergonomics
- Providing personalized suggestions and warnings to the operator based on their postures and the way that they use their body to perform an operation

We assume that a successful implementation will be able to change and adapt the robot's configuration (i.e. height) based on the monitored operator’s state that is received in a real-time mode. The main goal of this use case is to perform the biophysics assessment for ergonomic optimization. More specifically, the system has to identify the anthropometric dimensions of the operator in order to adapt the optimal position of the windshield for the operations to take place. It can also provide relative warnings and suggestions. The main output in this use case is the biophysics assessment for ergonomic optimization.

Vision capturing systems will be used for the detection of the operator’s anthropometrics parameter in order to perform a proactive ergonomics optimization of the equipment. Additionally, with the purpose to establish a multimodal approach, the performance of other technologies for motion capture (e.g. the SmartsuitPro and Xsens) will be investigated to examine if their calculations are consistent and transferable across systems. More specifically, the use of a whole-body tracking system will be utilized as a "ground truth modality" for offline training, along with a few IMUs that will act as surrogate sensors in online (real case) monitoring. The number and placement location of the IMUs will be determined based on the prediction performance on experimental data. The body joints’ estimation based on the IMUs sensors will be coupled with camera-based estimations from computer vision algorithms, developed as part of the Operator State Monitoring system. Offline testing of the algorithms will rely on a Simulator that will model the 3D workplace environment in Unity. Finally, ergonomics dedicated standards, like the EAWS method or RULA and energy expenditure scores will be examined.

To summarize, the key activities of this use case are:

- Recognition of the operator's anthropometrics.
- The development of a multimodal fusion approach taking into account the measurements of different sensors to perform robust training and testing scenarios.
- Optimization of the workplace design in respect to the selected number and location and viewing direction of required cameras, and the identification of cameras' blind spots.
- The selection of the recorded video of the camera providing the most confident results in respect to the estimation of the pose landmarks (assuming that there are more than one static cameras in the workplace).
- The video system integration.

The direct benefits expected to be shown after the implementation of the aforementioned use case are:

- Improvement of workers' wellbeing at work.
- Mitigation of risks and accidents.
- Flexibility of workplace management.
- Continuous compliance to ergonomics standards and innovation.

2.1.2.2 Testing environment

The described use case scenario will be tested and evaluated, for the accuracy of its performance, both in a virtual (i.e. simulator) and in a real industrial environment.

The validation in the virtual environment includes the extraction of pose landmarks from a digital human model (DHM), while it makes some movements into the virtual world similar to those of the real human operator. The extracted landmarks will be compared with the corresponding ground truth 3D landmarks that have been manually set into the Unity software when the simulation was designed and performed.

Let's note here that for the extraction of the pose landmarks of the virtual operator, the same pipeline process that will also be implemented in the real industrial environment using as input video captured by cameras from different views is used. In the virtual case, the input that feeds the algorithm is the prerecorded video from the virtual camera as it is set into the Unity environment.

On the other hand, for the validation of the results in a real industrial environment, a whole-body tracking system will be used as the ground truth modality and its results will be compared with those of the IMUs sensors and the output of the computer vision (e.g. OpenPose) algorithm. This procedure will be implemented only during the training/validation phase, while in real working conditions only the video capturing system will be used so that the operator will be undistracted by wearable devices and will be able to work naturally.

In collaboration with UPAT, the workstation cell is also created in an immersive virtual environment in order to analyze and evaluate in a completely safe way the various steps, movements and interactions between the human and the robot.

Within the scene in the virtual environment, the operator can interact with the robot and the virtual objects without running real risks, but reproducing exactly the same operations he would do in reality.

CRF will use body motion tracking systems to define safety and interaction zones with the robot, so that the robot's movement can be automatically reprogrammed and adjusted before the final test with the real robot.

Regarding the virtual environment, this method will be used to perform the test:

- Reproduction of the real cell layout.
- Reproduction of the actual movement of the robot and its safety volumes.
- Reproduction of the task cycle steps with virtual object manipulation.
- Test with users of the scene in virtual environment.

CRF will use the VR XVR system currently in use at CRF laboratories. This system integrates the Xsens motion capture systems for body movements and the Cyberglove for fine hand movements. This way the user can move freely over the VR scene and interact with his own hands/body virtual representation and the virtual objects. The mocap systems described above increase the presence of the user in the virtual environment. In specific, the object grasping can be fully simulated with the Cyberglove system.

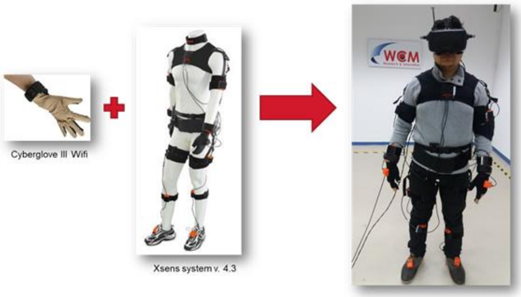


Figure 6: Integrated systems in CRF VR room

At first step of analysis, it is possible to navigate the VR environment reproduced at scale 1:1 and to understand spaces and equipment disposition.

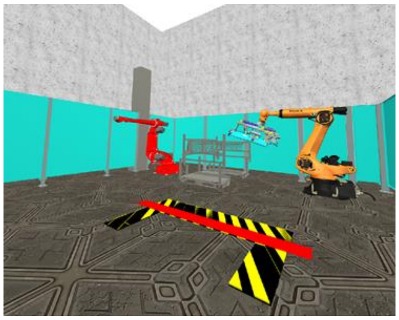


Figure 7: Navigable VR environment

Then the operator can control spaces and layout, and navigates the scene where he actually is.

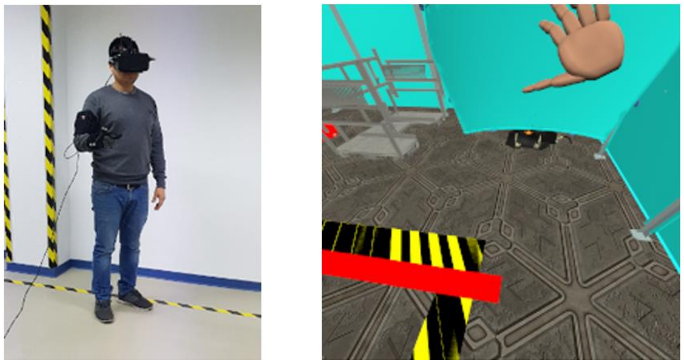


Figure 8: VR navigation

When the user interacts with the virtual environment there is both visual and audio feedback. The collision is recognized between the virtual manikin or hands and the virtual objects.

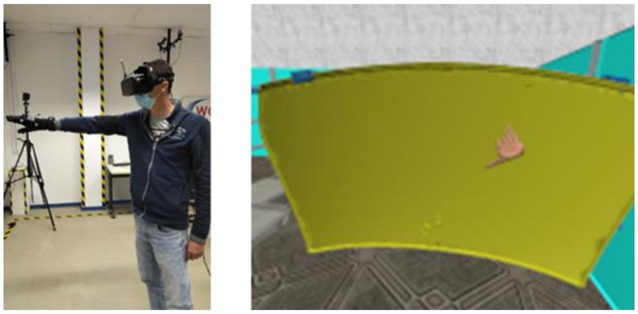


Figure 9: VR interaction

2.1.2.3 Testing procedure and data acquisition

The collected pose landmarks, representing specific key points of the human body, are acquired by the OpenPose algorithm. For the evaluation of the landmarks detection algorithm accuracy, the positions of the corresponding ground truth landmarks are used and the root-mean-square (RMS) error will be used as evaluation metric.

Data for this use case evaluation will be collected in real-time, through the multimodal approach, while the worker performs typical operations in the workplace.

The data that need to be collected are listed below.

Components and sensors:

- Videos from cameras.
- IMUs.
- Whole-body tracking system.
- Robot Controller.

Ergonomics is related to safety as it refers to long term health problems. Ergonomics is the approach to study and optimize the body physical parameters of the operators during his repetitive tasks. It analyses all the operator's tasks in search of wrong postures, positions, and loads in order to avoid them and give support to the operator.

Factors that affect the operator's ergonomics are:

- Weight of lifted objects (usually supported by mechanical lifters called partners or manipulators).
- Repetition rate of movements.
- Wrong postures introducing weird awkward body angles (e.g. too much bending, operations with elbow above shoulders for a long time).

All the above situations require design workplace optimization or the use of additional supporting tools or procedures. The use of Collaborative Robotics itself is often approached thanks to its capability to support operators in heavy load lifting and in performing repetitive tasks.

The CRF IVR system can provide all information regarding anthropometric data like the coordinates of human segment and angular time histories for ergonomic and reachability analysis. The ergonomic analyses that CRF can provide are EAWS, OCRA, OWAS, MURI using commercial software such as Process Simulate or Jack by Siemens, and proprietary software such as CRF VTA® (Virtual Task Analysis).

Process Simulate© per EAWS[2]

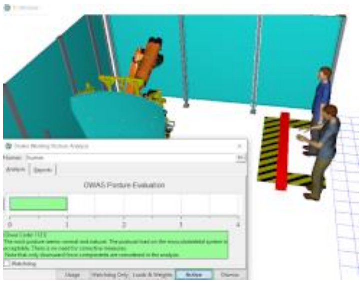
- EAWS (ERGONOMIC ASSESSMENT WORKSHEET) is an innovative approach to improve the ergonomic design of a workplace. The main objective is a good work planning. EAWS is adopted globally by leading multi-national groups in the industrial manufacturing, automotive (including FCA), and Aerospace & Defense sectors. It is an ergonomic screening system for biomechanical overload risk, designed to address a holistic risk assessment, i.e. including all types of biomechanical risk (static and dynamic postural loading, application of forces, load carrying and repetitive upper limb movements) to which an operator is exposed (especially regarding spine, neck, upper limbs, and lower limbs) when performing a work task.

Figure 10: EAWS

[2] https://www.plm.automation.siemens.com/global/en/products/tecnomatix/

Jack© for OWAS and SSP

- OWAS, The Ovako Working Posture Assessment System was formulated in Finland. The OWAS method was intended to identify the frequency and time spent in the postures adopted in a given task, to study and evaluate the situation, and thus, recommends corrective actions. The OWAS identifies the most habitual back postures in workers (4 postures), arms (3 postures), legs (7 postures) and weight of the load handled (3 categories). All this implies up to 252 possible combinations. Therefore, each posture assumed by a worker was assigned a 4-digit code that depended on the classification within the previous postures for each part of the body and the load.



OWAS Code: 1111
The work posture seems normal and natural. The postural load on the musculoskeletal system is acceptable. There is no need for corrective measures.
Note that only downward force components are considered in the analysis.

Figure 11: OWAS

- SSP, Static Strength Prediction, evaluates the percentage of a worker population that has the strength to perform a task based on posture, exertion requirements and anthropometry, including wrist strength calculations.

VTA CRF tools for OCRA and MURI

The VTA software had been developed to assist the ergonomic analysis and is focused on a detailed and objective evaluation of the working activities carried out on a work station, conducted according to the requirements of international and corporate standards.

The general aim is the implementation and development of methods for the ergonomic evaluation process, with the aim of applying the data and the information obtained also in the design stage of new work stations or on redesign/modification of existing workstations.

The VTA software is also focused on providing an easy interface to collect ergonomic input from a work activity, coming from videos, which is:

- Body motion capture system.
- Hand motion capture system.
- Hand pressure sensor.

Main benefits are to:

- Identify the most critical tasks of the work activity.
- Identify the main ergonomic criticalities.
- Provide the ergonomic evaluation in accordance with the international and corporate standards.

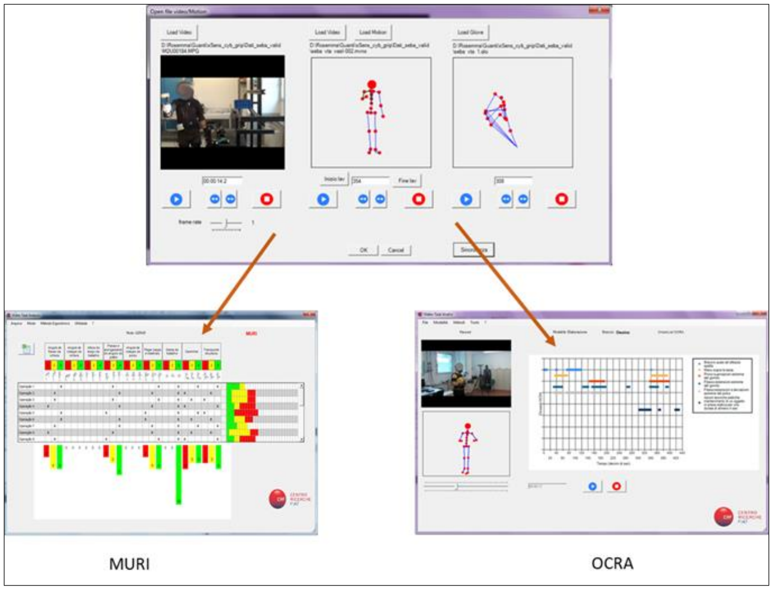


Figure 12: VTA CRF tools for OCRA and MURI

Operator State Monitoring - evaluation of the manufacturing application of the DSM system

The analysis and tracking of the operator's state can support safety and resilience in the workcell. Indeed distraction, tiredness, and other states of the operator causing discontinuity and lack of attention in the operator's tasks can generate risks for the same operator and his colleagues.

During the small-scale trials there is the intention to evaluate the performances of the DSM (Driver State Monitoring) solutions developed in the context of the automotive use case, in the manufacturing Safety and resilience use case. The DSM's concept can be used to evaluate states of distraction of the operator that may generate dangerous operations and other health conditions. The DSM developed for the automotive use case runs on android platform and uses the frontal camera to recognize the operator. The solution, after some alteration, may be suitable for application in the manufacturing use case.

In the case of manufacturing the boundary conditions of the "state monitoring" application are different. For suitability, in the manufacturing use case, the camera can be placed on the gripper, but likely not behind the windshield (though transparent, it may generate reflections and distortions).

The table below represent the main differences in the two use cases:

Table 1: State monitoring in manufacturing use case vs automotive

	Automotive	Manufacturing
Object	Driver	Professional operator
Position in respect to sensor	Almost fixed, driver seated in an almost fixed position	Standing few seconds in front of the windshield, walking in the workcell
Distance from the camera	~600 mm	> 1000 mm
Likely behavior to monitor	Sleepiness (eye blink, yawning)	Distraction, eye movement, looking in wrong direction with respect to expected ones
Obstacle on the face	Occasionally eyeglasses/sunglasses	Occasionally eyeglasses; Hololens during training
Exposure time	Continuous	Few seconds
Demand for real time	Might be useful	Video elaboration required within cycle time (300 s): the video can be transmitted and elaborated offline

As can be noticed in the above table, there are differences in applying this technology to the manufacturing use case:

- User is always in motion.
- Few seconds of facial framing.
- More focus on user distraction or lose of concentration and focus than falling asleep or yawning.

Nevertheless, there is the intention to evaluate how the software works in the manufacturing use case. The plan is to evaluate the feasibility of application of the system developed for the automotive use case.

Apart from the above differences, in the automotive use case the pictures and videos are taken from the front camera of a smartphone or tablet device. To be suitable for application in the manufacturing environment, these technologies' placement methods need to be adjusted to fit the safety requirements. The devices have to be small, rigidly fixed on the robot not to get ejected during movement, resistant to dust, moisture, oils and other factors.

For all the above reasons the plan is to evaluate the direct applicability of the developed system (based on smartphone technology) to perform a preliminary evaluation:

- Field of view of the camera.
- Capability to identify features of the operator's face in the available timeframe.

If the first phase is successful the camera can be substituted with a remote industrial camera to improve the field of view of the camera itself. Finally, in case all the previous phases are successful, the developed algorithms will be adapted to improve the recognition of the distraction in the operator.

The applicability of the system is an interesting perspective which can add to the safety of the user in the manufacturing environment. The analysis of the systems and related parameters will be carried. A deeper analysis of the application, with technical equipment changes, can be considered. For example, the use of a different optics (mobile phone cannot be used in an industrial environment even if mobile phone definition is fine) and the connection of the new optics to a device where the image analysis can be executed. The evaluation will also be in terms of an offline/online analysis and on the customization of the parameters of analysis such as distraction and drowsiness.

2.2 Connected and Autonomous L3-L4 Vehicles

Second group of use cases relate to the autonomous L3-L4 vehicles. There are three use cases described in this section: Human in the loop control in single vehicle scenario, Cybersecurity issues in connected cars, and Cooperative awareness scenario.

2.2.1 Human in the loop control use case in single vehicle scenario

Despite the growing automation of cars it is still the human user that is the main actor in the interactions with the vehicle. For this reason the following use case concentrates on safety-critical driver's states: drowsiness and distraction that can be monitored during driving to provide the car users' safety. The validation of the Driver State Monitoring system, described in the following subchapters, is an important part of the system's development.

2.2.1.1 Evaluated CPSoSaware components, related requirements and evaluation concept of the targeted components and use case

2.2.1.1.1 Detailed use case concept

The goal of this human in the loop control scenario is to facilitate the cooperation between the vehicle systems and the human driver (e.g. during the request to intervene) and improve subjective attitudes of the drivers towards the instrumented L3-L4 vehicles.

The vehicle autonomization is an ongoing process, with several automation stages. On each of these stages the transition from automated to manual control and vice versa might take place, initiated by the driver or by the system, depending on the mode used when the transition started.

Autonomous Driving Systems (ADS) are able to work unattended only under mild conditions, however they require a human driver to take control in situations that cannot be handled in an automatic way. In such cases the so-called Request to Intervene (i.e. SAE level 3 vehicles) is issued by the system, if it is the system that controls the vehicle and it requests to change driving state. ADS require the cooperation of a Driver State Monitoring System (DSM) that assesses the state of the human driver and a sub-system that performs an analysis of the scene outside the vehicle and controls the vehicle to move autonomously on a predefined path (e.g. recognizing vehicles ahead, estimating their velocity/trajectory, forecasting future vehicle locations). The human driver state should be assessed constantly in order to monitor the driver's availability.

To evaluate the scenario, it is important to provide the framework for safety-critical states that can lead to road accidents, with driver inattention being one of the most significant factors. It can be classified into two basic and distinctive categories – misdirected attention and fatigue/drowsiness. These categories will be the focus of the following CPSoSaware use case.

2.2.1.2 Testing environments

The human in the loop scenario can be validated in two types of procedures: laboratory testing (including non-driving tasks or driving simulators) and on-road studies (e.g. test tracks or naturalistic driving studies). Laboratory studies allow to manipulate the driver state while providing safe test conditions, the use of test track allows for moderate level of driver state manipulation with the safety provided by the presence of safety driver, and naturalistic driving studies give the possibility to observe natural behavior of the drivers in real-road conditions.

Several measures can be recorded in the test scenarios, depending on the chosen method of testing, and these can be data from external sensors, video-based measurements, and physiological signals. In both simulator and naturalistic driving physiological measures of human state (e.g. heart activity), eye movements and blink characteristics, and driving performance based on data from the simulator or car's Control Area Network can be obtained.

Most popular drowsiness testing procedures refer to long, monotonous tasks (e.g., Anund, 2018), circadian rhythm (National Sleep Foundation, 2007 after: Bowman et al., 2012), and subjective feelings of the subjects. Consequently, drowsiness related studies are usually conducted during the night (i.e. between 11 PM and 5 AM) with the use of driving simulator with dedicated long, monotonous scenario allowing the drowsiness and microsleep events to occur in safe conditions. Certain preparation of subjects is also mentioned in the literature. Ahlström and his team for example (Ahlström et al., 2010) instructed the drivers not to drink alcohol for 72 hours before the test day, not to drink coffee/tea or eat for four hours before coming to the laboratory, and have 7-8 hours of sleep for three subsequent nights before the test day, as such factors can influence the drowsiness study's outcome.

The level of sleepiness is usually controlled with physiological measures and subjective evaluation mostly with the use of Karolinska Sleepiness Scale (KSS, Åkerstedt & Gillberg, 1990) - a subjective scale, in which the drivers assess their own sleepiness on a 9-point Likert-type scale. European Commission (2020) recommends the use of KSS to measure the self-described level of drowsiness in drivers in the context of Driver Drowsiness and Attention Warning (DDAW) systems validation. The scale is considered a good predictor of possible occurrence of road events, as the self-rated sleepiness is related to certain changes in physiological states and driving behavior (Ingre et al., 2006), especially on levels 7 and higher. Due to that, level 7 has been chosen by the European Commission (2020) as a required threshold of DDAW systems informing the driver of sleepiness. KSS was positively validated against the EEG and driving performance measures, and as such it is a useful tool for assessing sleepiness (Kaida et al., 2006).

The use of car simulators in drowsiness-related studies serves to provide the safety of the participants, who often drive in the state of severe sleepiness. This is especially important when considering the validation of the driver monitoring systems against high levels of KSS, when the risk of road accident occurrence grows. Although there are various constructions of driving simulators, for the purpose of such research it is advised to use a high-class one, for the best possible fidelity of driving experience and thereby most natural participants' behavior. Real-driving studies rarely include high levels of drowsiness, mostly due to safety reasons, however they allow to gather data about the surrounding environment and explore the driver's state in the wider context. For the purpose of the described use case both simulator laboratory scenario and naturalistic driving conditions are proposed.

2.2.1.3 Testing procedure and data acquisition

Two testing environments mentioned above, driving simulator and real-road conditions, allow to gather similar sets of data, but with different procedures applied and for slightly different purposes. This section provides the information about the procedures of conducting both types of studies, the instruments that can be used in both setups and the outcome measures delivered.

2.2.1.3.1 Simulator laboratory test procedures

To provide the most reliable data the testing with the use of simulator may be conducted in two sessions: a day and a night one. The day session serves as a baseline and the night session allows to observe the natural process of decreasing alertness and growing drowsiness of the subjects. Day session starts at the morning or day hour. Night session does not start before 11 PM, yet the participants are invited to the lab earlier, so the use of caffeinated drinks, food, electronic devices and taking naps can be controlled.

Research instruments applied in the simulator scenario include: tested DSM application, KSS keypad, eye tracking with Smart Eye Pro system, heart activity monitored with wearable device, context camera, demographic survey, Fatigue Assessment Scale, and Simulation Sickness Questionnaire.

2.2.1.3.2 Naturalistic driving test procedures

The procedure with the repeated day and night driving sessions is also applied to naturalistic driving study. The participants start their drive in a set localization in Warsaw, at morning or day hour for the day test drive, and in the night hour for the night test drive. Before taking part in the study, they undergo a short training in the use of the test car, to get accustomed with the driving and the equipment inside the vehicle. During every drive a safety backup driver is present, to take over driving in case the participant is too drowsy to continue driving or cannot finish the ride due to other reasons. Both day and night test drives are approximately 2 hours long, and are conducted on a pre-defined route, with sections of urban and express roads.

Research instruments applied in the naturalistic driving scenario include: tested DSM application, KSS keypad, eye tracking with Smart Eye Pro system, heart activity monitored with wearable device, two context cameras, demographic survey, and Fatigue Assessment Scale.

Ethical statement

The study will be conducted in compliance with the Declaration of Helsinki from 1964 and its successive revisions, as well as with the Code of Ethics of Polish Psychological Association. Every participant fills in a written informed consent before taking part in the study, which provides the information on the data processing, in compliance with GDPR.

Adequate precautions should be undertaken to ensure the safety of participants and researchers in relation to COVID-19 pandemic, in all conducted studies.

2.2.1.3.3 Applied research instruments

Participants' sample and preparation

The sample is 10 participants, with equal gender distribution. Three ethnic groups should be represented: Caucasian, African (at least 30% of sample), and Asian (at least 30% of sample), to cover different anatomical facial and eye features. Only active drivers can take part in the study. Three age groups are planned, 20-29 years, 30-39 years, and 40-49 years, with at least three participants in every group. The participants' sample is controlled with regard to gender and ethnicity distribution for the two turns of night driving. Professional drivers and shift workers should be excluded from taking part in the study. Similar sample should be chosen for the naturalistic driving scenario.

Tested DSM solution

Provided Driver State Monitoring system is based on the Android application installed on the smartphone. The application uses the front camera of the device to constantly monitor the driver and extracts facial and upper body landmarks, e.g. eyes, mouth, shoulders, or wrists. Blinking, yawn and pose detection algorithms are applied to monitor the occurrence of drowsiness or distraction signs. The application rises an alert if the estimated drowsiness or distraction level surpasses the given threshold.

Driving simulator

The simulator is built on the Opel Astra cabin positioned on a moving platform with six degrees of freedom (angular movements: shift ±22°, speed ±30°/s, acceleration ±500°/s^2; linear movements: shift ±0.25 m, speed ±0.5 m/s, acceleration ±0.6 g). The display is a 60 Hz multi-image display system covering the whole visual field in the driver's view in the range of 200° horizontal and 40° vertical. The rear and side mirrors are replaced with LCD screens and show the view corresponding with the surrounding simulation. The AutoSim environment allows to simulate and control vast variability of road conditions, like day and night, rain, fog, presence of other vehicles or road hazards e.g. pedestrians or wild animals, different elements of the exterior, including buildings and trees. These components create a very natural experience, yet provide safe environment for driving. The collision effects in the study's scenarios can be minimalized to limit participant's stress connected with such events, especially as they could occur in the state of severe drowsiness. Several driving parameters can be recorded during the drive, including speed, position in lane, steering wheel angle, and use of acceleration/brake pedals. During every drive the following data are recorded: context video recording, driving performance (simulator) log, KSS keypad log, DSM application and Smart Eye Pro system.

Scenarios

Two scenarios are used in the study, a day and a night one. They are designed to meet the study's purpose and requirements, and to be used in a repeated measures design, which means – to be similar enough to obtain reliable data comparison from the two trials, but also to vary enough not to allow for learning the design and not paying attention to the road in the second drive. The simulated roads are equipped with adequate road signs (horizontal and vertical).

An adaptive drive of 7 to 10 minutes is used to train the participants in the use of the simulator, in-cabin test equipment, and to exclude the drivers vulnerable to the simulator sickness.

Day test drive is a rainless daytime driving scenario with moderate traffic. Participants drive on a huge eight-shaped loop, on two different road types: single carriageway with four lanes (two in each direction) and single carriageway with two lanes (one in each direction). The surroundings vary from highway to urban, industrial, and rural areas. There is one 90° turn to the left (at an intersection) and mild arcs both to the left and right in the scope of the scenario. Participants drive for around 45 minutes (single loop takes approximately 40 minutes) and are instructed to obey the speed limits.

Night test drive is a rainless nighttime driving clove-shaped scenario with moderate traffic. Participants drive on a huge loop with two different road types: single carriageway with four lanes (two in each direction) and single carriageway with two lanes (one in each direction). The surroundings vary from highway to urban, industrial, and rural areas. There are also barriers, trees, traffic posts, and streetlights on the roadsides. There is one 90° turn to the left (at an intersection) and mild arcs both to the left and right in the scope of the scenario. Participants drive for around 90 minutes (single loop takes approximately 60 – 70 minutes) and are instructed to obey the speed limits. In case of the participant falling asleep, the simulation is stopped.

Testing vehicle for naturalistic driving

The car used for test drives is Škoda Octavia, equipped with the Smart Eye Pro eye tracking system with three cameras. The testing devices, including the smartphone with DSM application and the KSS keypad will be installed beforehand.

Karolinska Sleepiness Scale

In this study easy to operate keypad-based version of KSS with thorough training is used. The keypad has the numbers 1 to 9 placed horizontally and has an adjustable backlight that can be lowered for night driving, not to provide too offensive stimulus in the dark environment. Keypad log is generated for every drive, recording timestamp of the request to answer, timestamp of the answer (time of pressing any number of the scale), and the number chosen.

The training consists of two parts, both presented to the participants in the form of digital questionnaires. The first one concentrates on familiarizing the participant with the scale, ensuring understanding of each of the scale levels, and learning to describe internal latent states with the scale levels. The second one is oriented towards making direct associations between each numerical label of the scale and the appropriate linguistic anchor to achieve automatized answers which do not drive participants' attention from the driving task while providing his/her answers.

The answer to KSS is requested every 5 minutes. The 5-minute time interval between giving answers to the scale is widely described in the literature as a satisfactory balance between a good sampling rate of drowsiness with minimal arousing effect (Ahlström et al., 2018; Åkerstedt et al., 2005; Anund et al., 2017; Van Loon et al., 2015). Moreover, it is compliant with the European Commission General Safety Regulations (2020), which recommends the use of 5-minute testing interval with KSS. The European Commission (2020) as well emphasizes the importance of using the fully labelled version of the scale, as used in the following study.

The KSS answering procedure is also included in the adaptive drive both in the simulator and in real-world conditions to introduce the device and to train the participants in its use, yet due to short length of the adaptive drive the time interval between the answers is shortened.

Eye tracking

For the purpose of eye tracking the Smart Eye Pro system is used to monitor head pose, gaze directions, blinks, and eyelid and eye movements. The system is built on 3 cameras that constantly monitor parameters mentioned above and works in NIR (850 nm wavelength) and 60 Hz sampling rate. According to the specification, it provides gaze and head tracking accuracy of up to 0.5 degrees. Gaze directions are represented by two separate gaze vectors, one for each of the eyes. The system allows to detect blinks and their full characteristics, including eyelid closing and opening amplitude and speed, and blink duration. Saccades and fixations of the eye are also detected.

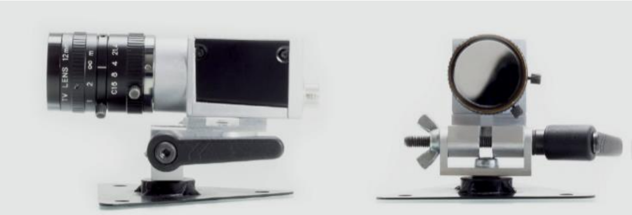


Figure 13: Smart Eye camera[3]

Heart activity

For the purpose of monitoring heart activity, a wearable device is used. The device can be placed on the wrist or the arm of the participant, allowing for long-time, non-invasive monitoring. The available data provide information on the driver’s heart rate and heart rate variability, and can support the monitoring of drowsiness.

Context cameras

An RGB camera is used to record the course of the drives, in both laboratory and real-driving conditions. This serves as an additional source of information about the scope of the test drive. For naturalistic driving scenario an additional external camera is used for lateral behavior monitoring, i.e. vehicle position and lane keeping monitoring, allowing for indirect monitoring of driver state.

Questionnaires used

Demographic survey

Basic demographic data are acquired from every participant. The survey has a form of computer questionnaire and also contains questions about visual impairments and potential correction, and driving experience of the participants.

[3] Source: Smart Eye

FAS

Fatigue Assessment Scale is a short questionnaire based on Thomas (2009). It is used to control the fatigue/arousal level of each participant before they start the drive.

SSQ

Simulation Sickness Questionnaire (Kennedy et al., 1993) is used to control the occurrence of simulator sickness symptoms. It contains a list of 27 symptoms, with four levels of intensity possible to select. The participants fill in SSQ before the adaptive drive, after the adaptive drive, and after the test drive to actively check for any changes of wellbeing and physical state.

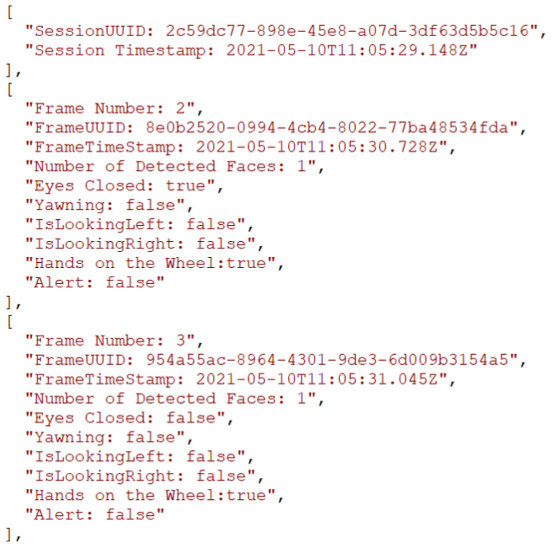
Electroencephalography can serve as an optional measure of driver drowsiness and can be applied in the study if needed.

2.2.1.3.4 Outcome measures

For every scenario integration and synchronization of measuring tools is performed, including context camera, DSM system, Smart Eye Pro system, and KSS keypad. Quality check of logged data is also performed.

DSM application

The data from the application are stored as JSON files. Each file contains the data about the Unique Session ID, Session Timestamp, Unique Frame ID, Frame Timestamp, the frame number, the number of the detected faces in the frame, if the driver is yawning or has his/her eyes closed in this frame, if he/she is looking left/right or has his/her hands off the wheel, and finally if the alert was fired. The JSON file content is presented in Figure 14.



```
[
  "SessionUUID: 2c59dc77-898e-45e8-a07d-3df63d5b5c16",
  "Session Timestamp: 2021-05-10T11:05:29.148Z"
],
[
  "Frame Number: 2",
  "FrameUUID: 8e0b2520-0994-4cb4-8022-77ba48534fda",
  "FrameTimeStamp: 2021-05-10T11:05:30.728Z",
  "Number of Detected Faces: 1",
  "Eyes Closed: true",
  "Yawning: false",
  "IsLookingLeft: false",
  "IsLookingRight: false",
  "Hands on the Wheel:true",
  "Alert: false"
],
[
  "Frame Number: 3",
  "FrameUUID: 954a55ac-8964-4301-9de3-6d009b3154a5",
  "FrameTimeStamp: 2021-05-10T11:05:31.045Z",
  "Number of Detected Faces: 1",
  "Eyes Closed: false",
  "Yawning: false",
  "IsLookingLeft: false",
  "IsLookingRight: false",
  "Hands on the Wheel:true",
  "Alert: false"
],
```

Figure 14: DSM Application's JSON file content

Driving simulator

The output file from the simulator is a TXT file containing data values describing driver's performance and other parameters from the scenario. It is software defined and can be adjusted to suit the study's needs (for example *time headway to next vehicle* or other variables). They can also be raw or calculated values. Some basic data values are presented in the Table 2 below.

Table 2: Basic simulator output data values[4]

Data item	Description
position_x	vehicle's position according to the simulated world coordinates
position_y	
position_z	
speed_km_h	the vehicle's speed
steering_wheel	steering wheel rotation angle
accelerator_percent	the use of acceleration pedal
brake_percent	the use of brake pedal
direction_left	describes if the left blinker is used
direction_right	describes if the right blinker is used
distance_to_lane_center	vehicle's position in the lane, in reference to the lane center
collisions	describes if the collision occurred

Similar parameters can be recorded from the CAN bus and Inertial Measurement Unit (IMU) in real-world driving scenario, if the proper recording device is provided.

Eye tracking

Smart Eye Pro system used for eye tracking generates TXT files with the data items are presented in the Table 3 below.

Table 3: Data items from Smart Eye Pro output file[5]

Data item	Description
FrameNumber	sequential frame number, with count starting from 0
TimeStamp	based on the PC hardware clock
HeadPosition	x, y, z values in the defined World Coordinate System
HeadHeading	rotation of the head towards left or right
HeadPitch	rotation of the head downwards or upwards

[4] Source: Motor Transport Institute

[5] Source: Smart Eye

HeadRoll	tilt angle of head rotation
LeftGazeOrigin	starting point of left eye gaze vector
RightGazeOrigin	starting point of right eye gaze vector
GazeDirection	vector describing the direction of gaze
LeftEyePosition	position of the center of the left eyeball, in the defined World Coordinate System
RightEyePosition	position of the center of the right eyeball, in the defined World Coordinate System
LeftGazeDirection	vector describing the direction of gaze deriving from the left eye
RightGazeDirection	vector describing the direction of gaze deriving from the right eye
Saccade	zero value, if saccade is not detected; non-zero value, if the saccade is detected; non-zero value is the subsequent number of the saccade
Fixation	zero value, if fixation is not detected; non-zero value, if the fixation is detected; non-zero value is the subsequent number of the fixation
Blink	zero value, if blink is not detected; non-zero value, if the blink is detected; non-zero value is the subsequent number of the blink
LeftEyelidOpening	describes how widely is the left eye opened
RightEyelidOpening	describes how widely is the right eye opened
EyelidOpening	the average value calculated on the basis of left and right eyes opening values
LeftBlinkClosingSpeed	the speed with which the left eyelid closes during the blink, in m/s
LeftBlinkOpeningSpeed	the speed with which the left eyelid opens during the blink, in m/s
RightBlinkClosingSpeed	the speed with which the right eyelid closes during the blink, in m/s
RightBlinkOpeningSpeed	the speed with which the right eyelid opens during the blink, in m/s

Karolinska Sleepiness Scale

The keypad used by participants has the output file in CSV format, with timestamp of the request to answer, timestamp of the given answer and the key pressed.

Heart activity

The data collected by the wearable device are exported in the CSV files and contain the information of heart rate and heart rate variability and the timestamps of measurements. The data structure is provided by the device's manufacturer and may vary, depending of the model chosen for the purpose of study.

Other data recorded in the study

For every drive the RGB context camera video is recorded in the AVI format. Questionnaires used throughout the study are exported to XLSX files. In the NDS the route of the drive is recorded and retrieved from the GNSS device as GPX file.

2.2.2 Cybersecurity issues in connected cars scenario

Automated driving systems were developed to automate, adapt and enhance vehicle systems for safety and improved driving. Most road accidents occur due to human error, and automated systems use input from sensors like video cameras to reduce human error by issuing driver alerts or controlling the vehicle. Such systems have become common in modern cars, with automobile manufacturers integrating these systems in their cars. There are six levels of automation as shown in Figure 15. When it comes to Advanced Driver Assistance Systems (ADAS), the highest level (5) corresponds to full automation where the automated functions control all aspects of the car, and the lowest level (0) where the driver controls all aspects of the car.

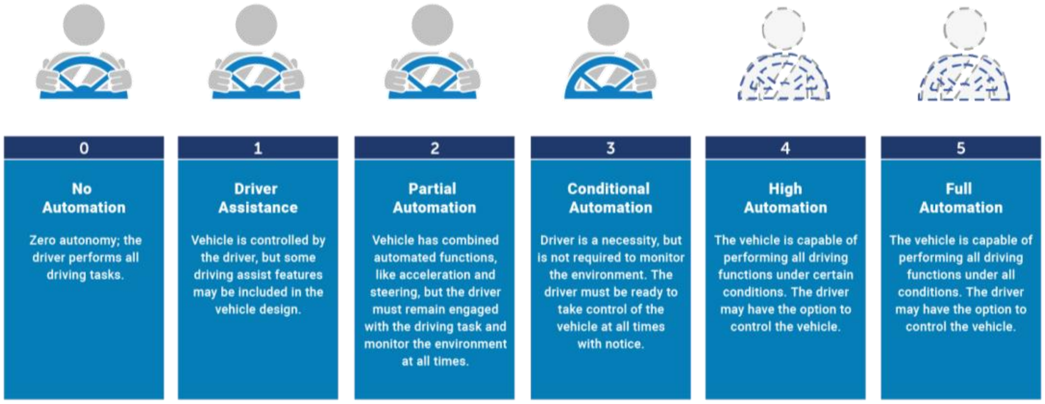

0
No Automation
Zero autonomy; the driver performs all driving tasks.
1
Driver Assistance
Vehicle is controlled by the driver, but some driving assist features may be included in the vehicle design.
2
Partial Automation
Vehicle has combined automated functions, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times.
3
Conditional Automation
Driver is a necessity, but is not required to monitor the environment. The driver must be ready to take control of the vehicle at all times with notice.
4
High Automation
The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.
5
Full Automation
The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle.

Figure 15: Society of Automotive Engineers (SAE) Automation Levels[6]

As vehicles become more connected to their external environment, the number of attack surfaces and risk of vulnerabilities being exploited escalates. A growing research literature has identified Connected Autonomous Cars (CAVs) vulnerabilities and analyzed the potential impact of successful vulnerability exploitation while suggesting some mitigation measures (Parkinson et al., 2017), documenting successful

[6] Source: SAE

cyber-attacks on security keys used by the ECUs, on Tyre Pressure Monitoring Systems (TPMSs), as well as remote attack against a Jeep Cherokee by introducing malicious data into the Controller Area Network (CAN) bus control parts of the car (including the braking system) by an adversary.

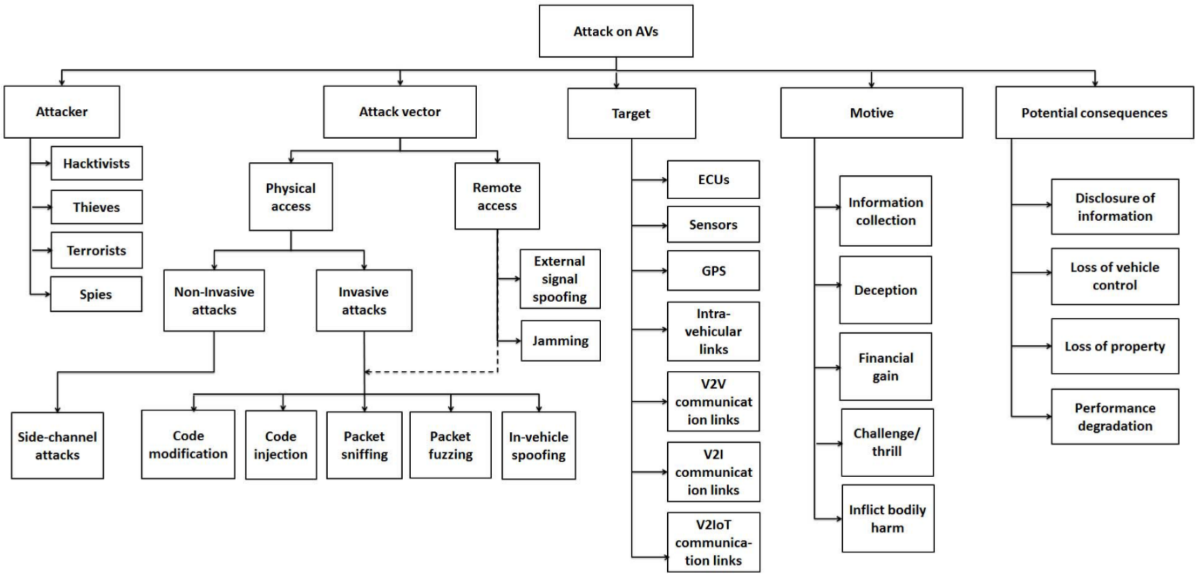


Figure 16: CAV attack taxonomy and assets

As seen in the above figure, the attack topology on Connected and Autonomous Vehicles is very broad. Practically, the attackers will target one of more of the following assets of the CAV, that is the EDU/OBU, the sensor data collection and distribution mechanism including the GPS and the Vehicle to Vehicle, Vehicle to Infrastructure and Vehicle to IoT communication topology.

GPS location spoofing attack attempts have been broadly researched for example, focusing on deceiving a GNSS/RTK receiver by broadcasting incorrect satellite signals, structured to resemble a set of normal satellite signals (e.g., GPS, GLONASS, GALILEO, etc.). One common form of a location spoofing attack begins by broadcasting signals synchronized with the genuine signals observed by the target receiver. The power of the counterfeit signals is then gradually increased and drawn away from the genuine signals.

Regarding V2V/V2X communication, in the research literature there is a broad range of security exploits that are related to Adhoc and peer to peer network security, which focus on especially wireless access technologies. Based on the threat model and taxonomy described by Boddupalli & Ray (2019) diverse types of attacks are identified (e.g. masquerade, worhmhole, man-in-the-middle) and three main vectors of attacks for wireless communication have been identified: frequency of malicious communication, the effect of the attack on V2X (e.g. injection of fabricated message, message mutation or even preventing delivery of the message) and effect on the vehicle (e.g. compromising safety or loss of efficiency of the targeted cooperative application). In addition to those, the connectivity of the vehicles with the Internet can introduce many traditional IT network security attacks in the CAVs domain including Denial of Service (DoS) and Distributed Denial of Service (DDoS) as well as other related exploits.

Furthermore, intravehicular communication attacks are also reported in CAVs. This mainly include the electronic control units (ECUs) interconnection mechanism via the controller area network (CAN) bus. The CAN protocol has several inherent weaknesses that are common to any implementation. Key among these:

- Broadcast Nature: Since CAN packets are both physically and logically broadcast to all nodes, a malicious component on the network can easily snoop on all communications or send packets to any other node on the network.
- Fragility to DoS: The CAN protocol is extremely vulnerable to denial-of-service attacks. In addition to simple packet flooding attacks, CAN's priority-based arbitration scheme allows a node to assert a "dominant" state on the bus indefinitely and cause all other CAN nodes to back off. While most controllers have logic to avoid accidentally breaking the network this way, adversarially-controlled hardware would not need to exercise such precautions.
- No Authenticator Fields: CAN packets contain no authenticator fields — or even any source identifier fields— meaning that any component can indistinguishably send a packet to any other component. This means that any single compromised component can be used to control all of the other components on that bus, provided those components themselves do not implement defenses.
- Weak Access Control: The protocol standards for our car specify a challenge-response sequence to protect ECUs against certain actions without authorization. A given ECU may participate in zero, one, or two challenge-response pairs.
- Tester capabilities: Modern automobiles are complex and thus diagnosing their problems requires significant support. Thus, a major use of the CAN bus is in providing diagnostic access to service technicians. In particular, external test equipment (the "tester") must be able to interrogate the internal state of the car's components and, at times, manipulate this state as well.

A broad range of attacks is also related specifically to the Vehicle control modules (eg. the ECU) within the Vehicle or in general any onboard unit (OBU). Changing ECU firmware has large implications as it can completely reprogram the vehicle's behavior, resulting in it becoming a potential threat to public safety. The firmware could be modified or replaced by performing a physical and valid update via the On-Board Diagnostics (OBD) port or by performing a remote update using the V2X communication channel. The source of the firmware/software update should be verifiable and in case there is a security mismatch then the vehicle should respond accordingly.

Cryptography can be used to provide a prevention mechanism against such attacks, however, ECUs within the vehicle might not be able to handle effectively the overhead of the Public Key Cryptographic schemes in order to match the automotive constrains. Also, any cryptography scheme needs to be implemented in such a way that it can be considered trusted. This means that it should be protected against various microarchitectural and side channel attacks that may be mounted through the OBD in order to obtain secret information (like the private keys).

To address some of the above security issues and related challenges, in CPSoSaware we use dedicated security sensors at the CPS level that can prevent, detect, respond and mitigate potential attacks and also report them in a security monitoring system of systems (the CPSoSaware SRMM). This mechanism at the CPS level is encapsulated by the CPSoSaware Hardware Security Token as well as the PASEU perception engine dedicated to adversarial attack detection.

More specifically, on this last topic, recent studies (Parkinson et al., 2017; Lu et al., 2017; Petit et al., 2015) showed that ADAS alerts and notifications can be spoofed by applying adversarial machine learning techniques to scene structural elements (e.g. traffic signs, objects, etc.).

Adversarial attacks seek small perturbations of the input causing large errors in the estimation by the perception modality. Attacking perception functions using adversarial examples is a popular way to examine the reliability of learning approaches for data classification (Nassi et al., 2019). The key to all such attacks is that the change to the image should be minor yet have a large influence on the output. Adversarial examples typically involve small perturbations to the image that are not noticeable by the human eye. The adversaries are shown to work even when a single pixel is perturbed in the image (Parkinson et al., 2017). Although these attacks reveal limitations of deep networks, they are hardly replicated in real-world settings. For instance, it is rather difficult to change a scene such that one pixel captured by a camera is perturbed in a specific way to fool the network. However, recent work demonstrates that adversarial examples can also work when printed out and shown to the network under different illumination conditions. Athalye et al. (2018) show that adversarial examples can be 3D printed and are misclassified by networks at different scales and orientations. Sharif et al. (2017) construct adversarial glasses to fool facial recognition systems. Nassi et al. (2019) show that stop signs can be misclassified by placing various stickers on top of them.

Apart from the adversarial attacks, which involve scene modifications on the physical layer, within the Autonomous driving, the vulnerability of the Perception Engine is also an important issue to address. CPSoSaware focuses on this point thanks to the perception engine contributed by Panasonic Automotive Europe. The Perception engine must be secured against a variety of cyber-attacks at the sensors layer with the help of proper approaches for detecting the attacks and mitigating them.

In line with what is described above, CPSoSaware in the framework of Automotive Pillar believes the following scenarios presented below are the most important cases to be addressed. Note that the presented scenarios are selected based on the CPSoSaware consortium knowledge, available resources, and showcasing capacity.

2.2.2.1 Detailed Use Case Description

2.2.2.1.1 Cybersecurity threats and Defense in Depth employed methodology

Based on the above analysis we are aiming to test the CPSoSaware security pillar solutions in detecting various attacks that are associated with CAVs. For this reason, the testing procedure to be followed will be based on a defense in depth strategy that tries to capture cyber-attacks in layers. If the attacker manages to penetrate one layer (the attack is not detected) then the CPSoSaware solution should provide additional detection mechanisms on a different layer of the CPSoSaware framework. Our goal is to test the system layer as well as the CPS layer of the CPSoSaware framework in the automotive pilot by adapting our solution to the pilot and providing prevention, detection, response and mitigation (when possible) in a cyberattack. This attack can be targeting a single CPS (one specific car) or it can be systemic (targeting connected car clusters). Thus, the cyber-attack testing procedure consist of a multi-stage attack scenario that can target various layers of CAVs. (i.e. Several attack entry points). The attacker exploits vulnerabilities in each stage by focusing on a specific attack target.

- The first stage has to do with attacks on the intercommunication medium (V2V and V2X communication).
- The second stage of the attack is focused on the intracommunication network within the vehicle.
- The third stage of the attack is to install a malware that can provide the attacker eavesdropping, modification and forging of in-car ECU processing as well as make the ECU non-operational (loss of availability).
- The fourth stage of the attack is the deployment of a malicious software that alters the sensory data of a vehicle.

Stage 1. Remote InterCommunication attack: Isolating a Vehicle from the Network

General Description

This type of attack exploits vulnerabilities that may appear at the communication mechanism between connected vehicles or between the vehicle and the infrastructure. This may include the cellular network or a Wi-Fi protocol and the IP networks that can be created on top of them including the Internet (if access to it is provided by the vehicle specifications) (Khan et al., 2020).

Possible Attacks

- Man-in-the-middle attacks/passive attacks/relay attacks can be mounted when an adversary sends the original message to the CAV, updates it and sends a new message to the vehicle that causes an incorrect message switch between the CAVs communication. This attack can lead to eavesdropping, disclosure of information, and traffic-data analysis (Khan et al., 2020).
- Replay attack can be mounted that enable the perpetrators to continually forward a valid frame to prevent the CAV from working in real-time or to spoof another node's identity.
- Masquerade attack is an intrusion that uses a false identity, such as a network identity to obtain unauthorized access to the CAVs without valid access authentication.
- In a tunneling attack, two parts of the network are linked by an attacker using an external communication channel that may result from eavesdrop on V2C communication to halting of CAVs real-time operation (Khan et al., 2020).

Stage 2. Remote attack: exploiting an external interface vulnerability

General Description

This type of attack exploits vulnerabilities that are present in external functional interfaces which are usually related to telematics or infotainment. The entry points for such attack can be connected ECUs/OBU operating for a variety of functional uses.

Possible attacks

- In an Internet/Cellular Carrier environment, the attacker tries to identify a vulnerable vehicle.
- If the direct IP connectivity of the car is unprotected, the attacker discovers the Telematic Control Unit (TCU) through the Shodan platform.
- If the direct IP connectivity of the car is protected, the attacker can move to another entry point such as SMS.
- Knowing that a given vehicle model has a vulnerable SMS link and its MSIN, the attacker enumerated MSINs hoping that all numbers have been sequentially assigned, thus discovering other vulnerable vehicles.
- The attacker uploads an altered firmware to the ECU/OBU, either by exploiting the lack of authentication for SMS update or alternatively through the direct communication with the ECU/OBU due to non-diversified SSH credentials.
- Depending on the level of access to the ECU/OBU the attacker has gained, the crafted firmware is able to communicate legitimately on the CAN bus with the driving systems.

Stage 3. Local attack: unauthorized flashing of malicious code

General Description

This local attack can be perpetrated by obtaining a legitimate or illegitimate access to diagnostic equipment and subsequently exploiting a vulnerability to persistently alter the behavior of an OBU/ECU. The actor of this attack could be either a legitimate vehicle user wanting to alter some vehicle characteristics with compromised equipment or a garage/Third party OEM car producer employee with various motives (business intelligence, organized crime, ransomware).

Possible attacks

- At any time and independently of the access availability to the vehicle under attack, the attacker downloads an ECU update from the manufacturer in order to gain information about the target vehicle, its functionality, and its possible weak points.
- The attacker monitors the update process to get information about the update routine.
- Based on the information gathered on the previous steps, the update algorithm is reverse engineered and modified appropriately.
- The attacker flashes the maliciously altered code on the engine ECU over CAN by using the diagnostic reprogramming routine through the OBD-II port found in all vehicles.

Stage 4. Local attack: unauthorized Fabrication of Sensor Data

General Description

This stage involves a cyber-attack based on activating some malicious software which got installed during the software/firmware update process (in the previous stage of the attack). We assume that this malware has not been detected in the previous stage and gets installed and executed. Since adversarial attacks are very popular in the Automotive sector, the target of this stage is the Connected car's sensor data i.e. the camera or the LiDAR sensor.

Possible attacks

1. Attack on the Camera Sensor Layer: Throughout this use-case the camera sensor could be attacked in several different ways, which could vary between adding noise lying on specific bands of the frequency spectrum/ introducing morphological overall or parts of the image.
2. Attack on the Camera Sensor Layer by de-synchronizing the data: Throughout this scenario, the cyber-attack will be geared towards disturbing the association between the captured frames and the timestamp assigned to them. This will cause the failure of the perception engine, as all the architectural modules performing stochastic filtering on the scene observations will be affected by error. This use case should study the potential and the limitations of the cyber-attack detection and mitigation engine in assessing and recovering the failures.
3. Attack on the Camera Sensor by a remote agent: In addition to the scenario, the cyber-attack detection and mitigation engine will be used to detect and mitigate the camera signal distortion in the case that a malicious remote agent interferes with the test vehicle by knowing the IP of the processing unit and sharing some erroneous data. More specifically, this use case will assume that the remote agent sends via V2X communication: time zone/daylight related data in order some sensor parameters (e.g.: gain/exposure time) to be tuned accordingly.
4. Attack on the LiDAR sensor: Apart from the camera, cyber-attacks on the LiDAR sensor is another important issue. As in use case 2, the attack will involve triggering malicious software through either a remote agent or some date-related software update process. The malicious software could distort multiple attributes of the LiDAR signal which could vary by either adding noise to the measured data or changing arbitrarily some of the sensor configuration parameters (e.g.: scanning frequency).

2.2.2.2 Evaluated CPSoSaware components, related requirements and evaluation concept of the targeted components and use case

The testing approach, the testing environment and various multistage attack scenarios are evaluating the CPSoSaware architecture components of:

- Security Accelerators for CPS Security Agents/Sensors.
- Xilinx XRT KPI Monitoring.
- CPS Layer Security Sensors/Agents.
- Security Runtime Monitoring.
- Deep Multimodal Scene Understanding.

In the following subsections we describe the testing environment for the evaluation of the CPSoSaware architecture for the Cybersecurity pillar in the Automotive pilot. In short, we evaluate the requirements of the CPSoSaware architecture of efficiency, scalability, responsiveness, detectability and the level of achievable confidentiality, integrity, accountability. Also, the secure logging mechanism between CPS and system level on cybersecurity events. In the following subsection we provide more insight on the evaluation framework and the metrics employed to quantify the efficiency of the cyberattack detection and mitigation engines.

2.2.2.3 Testing environments

The described use case scenarios will be tested and evaluated, for the accuracy of its performance, in a virtual environment (i.e. simulator) and in the real vehicle when this evaluation is realistic and practical (ideally the testing/evaluation will be done in both simulation and real vehicle).

The validation in the virtual environment complements the multi-stage cyber-attack lifecycle described and consists of a remote server (preliminary a TCP server) connected through the Internet to a client software residing within an ECU or OBU of a vehicle under test. Following various different OBU/ECU testing environment we can emulate such an environment in a MultiCore System-on-Chip heterogeneous embedded Linux based platform that includes FPGA fabric and includes the CPSoSaware Hardware Security Token developed mainly in T3.5 of the project, or an AI Neural Network (NN) supporting embedded platform that includes CPUs and GPUs (e.g. Nvidia Jetson) as well as a dedicated Personal Computer deployed within the vehicle.

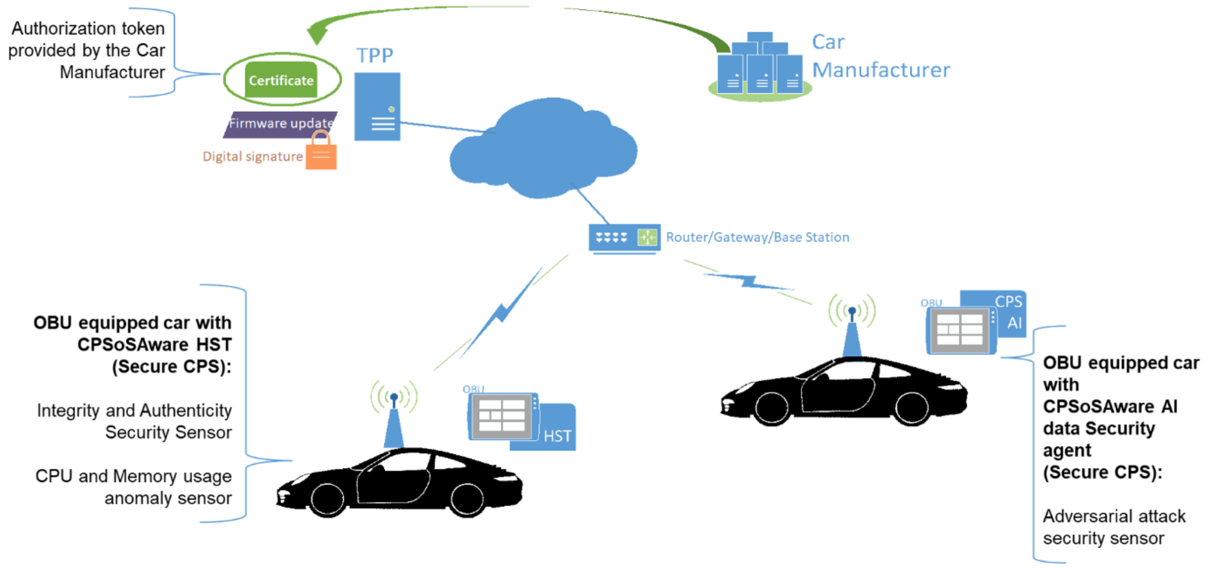

Authorization token provided by the Car Manufacturer
Certificate
Firmware update
Digital signature
TPP
Car Manufacturer
Router/Gateway/Base Station
OBU
CPS
AI
OBU
HST
OBU equipped car with CPSoSAware HST (Secure CPS):
Integrity and Authenticity Security Sensor
CPU and Memory usage anomaly sensor
OBU equipped car with CPSoSAware AI data Security agent (Secure CPS):
Adversarial attack security sensor

Figure 17: Testing environment approach

A preliminary use case scenario (Figure 17) that can be linked to the multi-stage cyber-attack lifecycle using the above testing environment would be the following:

1. We assume that the car manufacturer has given permission (by providing credentials) to some third-party (OEM) developer to provide firmware/software updates on a car OEM subsystem. The credential is an appropriate certificate that includes an asymmetric cryptography public key assigned to the OEM developer (corresponding to its asymmetric cryptography private key). The certificate is digitally signed by the car manufacturer. The car manufacturer asymmetric

cryptography public key is publicly known and is stored securely inside the CPS device (for example in the CPS Hardware security token).

2. The third-party OEM provider creates an update-firmware that needs to be sent to the car. The firmware/software update includes the submission of 3 information:

    - Firmware/software update
    - A digital signature of the firmware (using the third-party Asymmetric cryptography private key)
    - The certificate of the third-party OEM provider Asymmetric cryptography public key signed by the car manufacturer. This certificate has been provided by the car manufacturer and cannot be forged

3. The firmware/software is sent to the CPS and its security needs to be validated. An integrity and authenticity check needs to be made by the CPS security mechanisms (provided by the CPSoSaware Hardware security token or software security equivalent).

5. If the firmware passes the integrity/authenticity check it can get installed and executed.

The testing environment is assuming that the firmware/software is malicious and has some malware that gets activated while the update happens or at a later time during the car operation.

The malware’s presence may be detected by its anomalous behavior. It leaves traces of its behavior in the OS execution flow that can be monitored by appropriate detectors. It may attempt a Denial of Service to the car’s OBU/ECU, attempt information leakage and/or eventually perform adversarial attacks on the car’s sensors as those are described in the stage 4 of the multi-stage attack lifecycle.

The testing environment preliminary assumes 2 types of attackers:

1. Man in the Middle attacker: The attacker is able to eavesdrop the communication channel (inter or intra communication) and alter the traffic that passes through the attacker node. This means that the attacker can spoof the attacker identity acting as any of the communication entities (client or server, OEM third party provider, cat manufacturer or car CPS-OBU device). Thus, the attacker is able to alter the information send during a software/firmware update.

2. Man at the end attacker: The attacker has access to one of the end nodes involved in a communication. This node can be any client or server node and in the testing environment it is assumed that this type of attacker can have access to the OEM third party provided node. Thus, this attacker can modify the firmware/software and introduce malware.

The two attackers’ profiles in the preliminary tests using the above described test environment are able to perform attacks on the software integrity and authenticity by maliciously manipulating the software/firmware, attacks on the availability of the OBU/ECU (Denial of Service) by excessively increasing the CPU and memory usage as well as attacks on the car’s sensor signal using noise models of specific type and parameters as described in following subsections. The goal of the evaluation process is that the attacks get detected by the CPSoSaware hardware security token detection mechanism or the CPSoSaware cyberattack detection and mitigation software engine developed under the security pillar of the project. Regarding the second engine, the mitigated output will then be compared towards the noiseless input both at the viewing at the perception layer.

The evaluation protocol quantifying the accuracy of the cyberattack and mitigation engine is summarized in the following subsections.

2.2.2.4 Testing procedure, data acquisition protocols and preliminary results

The experiments that are performed as part of the preliminary tests on the cybersecurity in connected autonomous cars will follow two procedures. The first procedure is focused on the stages 1 to 3 of the multistage cyberattack lifecycle and includes testing of cybersecurity detection capabilities of the CPSoSaware architecture components on data integrity/authenticity attacks, on network security attacks and on availability (denial of service) attacks targeting a CPS device. The second testing procedure is focused on the detection of adversarial attacks (stage 4 of the multistage cyberattack lifecycle) on the camera sensors of a single vehicle using the CPSoSaware cyberattack detection and mitigation software engine provided by PASEU.

2.2.2.4.1 Testing procedure 1

In this testing procedure we follow the full testing environment use case scenario that includes server and client nodes (the client device periodically checks on a remote server for software to be updated in the cps). Following the two attacker profiles we test for two aspects of the CPSoSaware detection mechanism.

Test 1: A Malicious firmware/software that is digitally signed by an attacker's Asymmetric cryptography private key that is created by a man in the middle attacker.

In this procedure the CPSoSaware Hardware Security Token (HST) deployed at CPS level is tested in terms of its ability to detect data integrity and authentication failure. The HST is equipped with several different cryptography primitive implementations (both hardware and software based) and can be used in order to verify digital signatures and store securely keys. The verification outcome is producing log entries that follow the syslog protocol and can be stored locally in the OS (on which the HST is deployed) as well as sent remotely (securely) in a syslog server. In the CPSoSaware project the consumer of the log entries is the SRMM component provided by ATOS which is also involved in this test procedure.

The goal of the test is to evaluate the functional and non-functional requirements of the HST in detecting (under the given scenario): a) digital signature verification, b) extraction from the secure storage the appropriate certificate Asymmetric keys under the presence of attackers, c) the efficiency of the digital signature verification mechanism based on the expected pilot KPIs, d) the ability of the HST to correctly capture, describe, and transmit an appropriate event to the SRMM.

Test 2: A Malicious firmware/software that is using the digital signature of a legit software that is created by a man in the middle attacker.

In this procedure the CPSoSaware Hardware Security Token deployed at CPS level is again tested in terms of its ability to detect data integrity and authentication failure. However, the secure storage is not needed in this procedure but rather the public key to be used is extracted from the provided car manufacturer certificate. Similarly, the verification outcome is producing log entries that follow the syslog protocol and can be stored locally in the OS (on which the HST is deployed) as well as sent remotely (securely) in a syslog server. In the CPSoSaware project the consumer of the log entries is the SRMM component provided by ATOS which is also involved in this test procedure.

The goal of the test is to evaluate the functional and non-functional requirements of the HST in detecting (under the given scenario): a) faulty digital signature verification, b) extraction from a given certificate of the correct Asymmetric keys under the presence of attackers, c) the efficiency of the digital signature verification mechanism and extraction mechanism based on the expected pilot KPIs, d) the ability of the HST to correctly capture, describe, and transmit an appropriate event to the SRMM.

Test 3: A Malicious firmware/software with valid using the digital signature and certificate that is created by man at the end attacker.

In this procedure the CPSoSaware Hardware Security Token deployed at CPS level is again tested in terms of its ability to detect some unknown malicious activity based on the abnormalities that such a malware will have in the ECU/OBU processing mechanism. A typical attack that can be linked in the automotive domain that can be detected in such a way is a Denial of Service attack that is aiming at making an ECU of the car non-responsive. The HST is monitoring the CPU and memory usage of the CPS and if a pattern is detected that seems abnormal (due to a possible malware activity, a possible DoS) then this is detected and a Syslog message is generated (following a Json format) which is propagated to the SRMM tool provided by ATOS.

The aim of this test is to evaluate the responsiveness of the detection mechanism as well as to identify the number of false positive and false negatives that the sensor will produce. This will contribute to the evaluation of the accuracy of the detection mechanism but also in the eventually fine-tuning of the mechanism in order to determine the appropriate threshold after which a set of CPU and memory measurements are considered malicious. In the current version of the detector the parameters that determine the threshold are the distance of mean, standard deviation as well as the time window size for each measurement block to be estimated.

In the following figure, an overview of the preliminary tests that are currently performed or going to be performed the relevant project task can be seen.

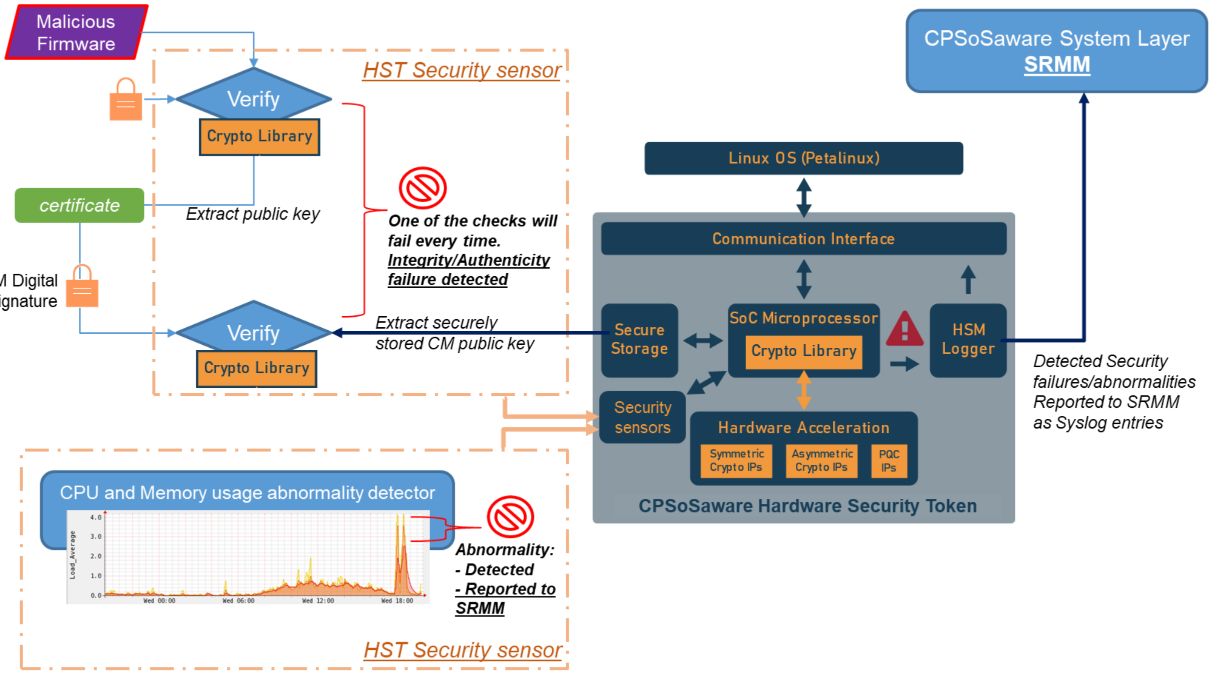


Figure 18: Overview of testing procedure

The extracted output from the testing procedure, apart from efficiency, accuracy and scalability metrics will be log entries that follow the syslog protocol and are stored in the syslog log of the OS (Linux based OS) as well as in the dedicated log of the HST. The structure of this collected data can be seen in the following figure (JSON format entries).

```
{
   "HostIP":<integer>,
   "HostID":<integer>,
   "HostState":<string>,
   "HSMid":<integer>,
   "timestamp":<integer>,
   "event":{
       "type":<integer>,
       "failure":<integer>
       "severity":<integer>
   }
   "auth_token": <string>

   "comments": <string>
}
```

Figure 19: Log entry JSON structure

2.2.2.4.2 Testing procedure 2

Our experiment takes into account images extracted from 100 recorded videos, 1 minute each, in different parking areas, including supermarkets, office with angle, parallel and perpendicular parking scenarios. The vehicle platform used for data collection is illustrated in Figure 20.

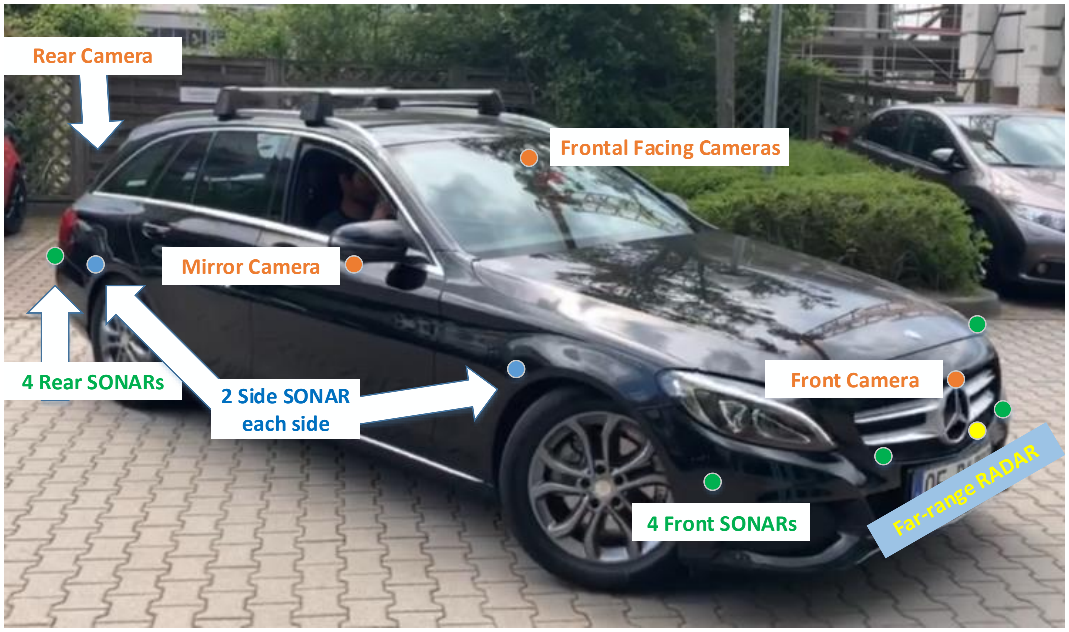


Figure 20: Picture of our ego-vehicle with sensors mounted

Each raw image (width 1280 and height 960) is attacked by noise, which are then used as inputs for our denoised models. The parameters for denoising models are tuned to generate a noise free image. Secondly, this denoised image is being fed to our computer vision algorithm to analyze the variation in output.

Figure 21 up to Figure 23 illustrate the architecture of the cyberattack and some instances of the mitigation output.

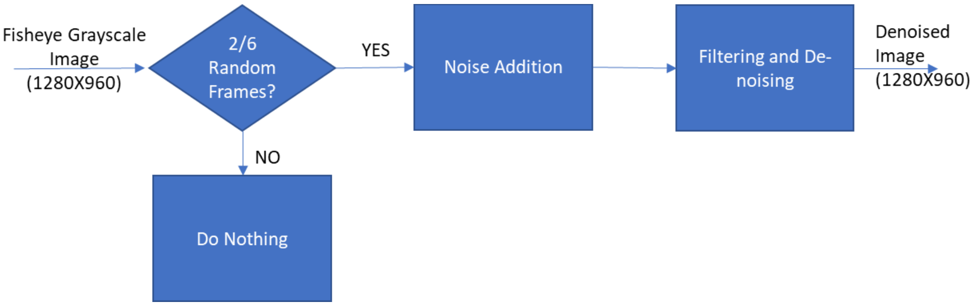


Figure 21: First Experimental Setup

Taken the input image, examples of noisy images and their denoised versions are show in Figure 22.

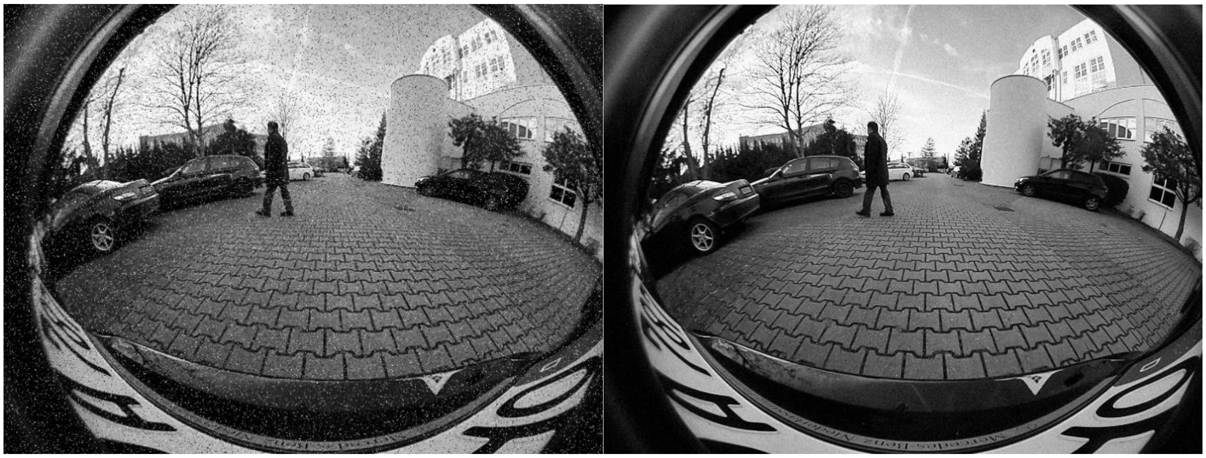


Figure 22: Examples of noise added image
Left: Impulse noise added randomly in 110000 pixels. Right: Gaussian noise (zero mean, standard deviation of 10.0) added for Cyber Attack.

Autonomous navigation is the main task of autonomous vehicle, which relies on a reliable occupancy grid map (OGM) for obstacle avoidance and path planning. An example of occupancy is illustrated in Figure 23. The interest of this research is to understand the impact of noise on the performance of autonomous vehicle in its autonomous navigation. Concretely, we would like to understand how big the degradation of the OGM is when noise is added to the input images. And thus, will such degradation degrade the performance or decision making of the autonomous vehicle? In fact, there are several generic approaches (Thrun et al., 2005) for evaluating OGM in robotics they but can be treated mainly as references. Since there are many different functions using OGM for different purpose during autonomous navigation, such as automated emergency braking, collision avoidance, autonomous parking, etc., it is not obvious to justify the impact without concrete evaluation per specific purpose.

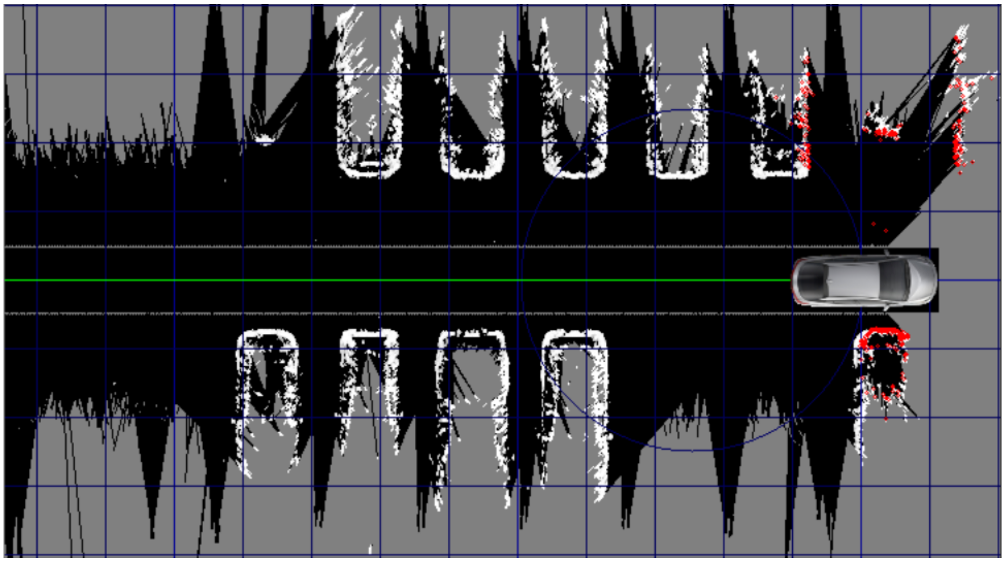


Figure 23: Example of Occupancy Grid Map
The Grid Map contains obstacle cells (white), free-space cells (black), and unknown cells (grey). Red cell are reflecting the current update of cells of obstacle points.

Based on different purposes, we investigate how to analyze OGM in order to extract useful information. Reliable and efficient approaches are derived depending on types of functions. For instance, for obstacle avoidance in navigation, it is practically efficient and much more meaningful to compare the outcome of obstacle polygons extracted between OGMs. This is because the polygons are fundamentally used for calculating time-to-collision and planning the local path to avoid possible collision. On the other hand, for parking applications it is much more meaningful to measure the free-space and obstacle's boundaries within surrounding environment to examine whether or not a possible parking slot is detected and an optimal trajectory can be planned accordingly. To make this clear the examples of Figure 25 and Figure 26 below come in handy. In Figure 24 point cloud created by a camera-based 3D reconstruction solution is fed into an OGM. Obstacle polygons are boundaries of high objects (>15 cm height), which are extracted and marked in red by a pre-determined solution. Meanwhile free-space is detected and defined in orange quadrilateral by a pre-determined solution.

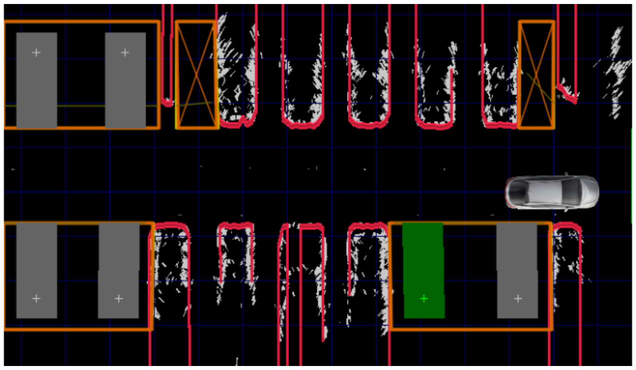


Figure 24: Occupancy Grid Map

Obstacle polygons (red), free space analysis (orange), parking-slot candidates (grey), optimal parking slot (green)

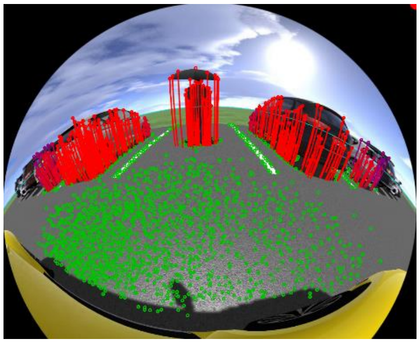


Figure 25: 3D reconstruction on the left camera

Including ground points (green) and obstacle points (red). To have an intuitive understanding of 3D perspective, a red line connecting the obstacle point to its foot-point is drawn.

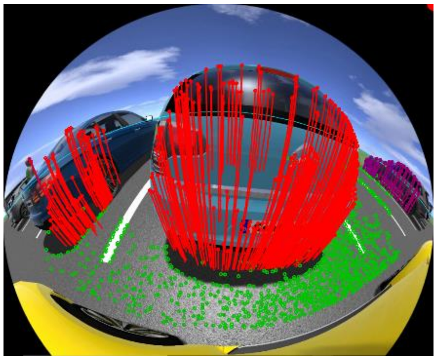


Figure 26: 3D reconstruction on the right camera

Including ground points (green) and obstacle points (red). To have an intuitive understanding of 3D perspective, a red line connecting the obstacle point to its foot-point is drawn.

It is trivial to observe that if having the same obstacle polygons and free-space quadrilateral areas, the performance of the autonomous vehicle will be identical in its autonomous driving and autonomous parking. In other words, if almost having the identical obstacle polygons and free-space quadrilateral areas by both OGMs resulted from the noise added input images and the raw (no noise) input images, it is obvious to conclude that the impact of noise after the de-noising is minimal or negligible. In contrast, a big difference between the two will lead to totally different behaviors of the autonomous vehicle in its autonomous navigation. Therefore, we propose to examine two KPIs (Key Performance Indexes) in this evaluation: 1) Distance transform between obstacle-polygons, 2) Intersection over union (IOU) between free-space quadrilateral areas.

Distance transform between obstacle-polygons

The distance transform D of a binary image B is a real-valued image of the same size, where each pixel is assigned the distance between itself and the closest non-zero pixel of the binary image:

$$D(\boldsymbol{p}) = \min_{\boldsymbol{q} \in B} ||\boldsymbol{p} - \boldsymbol{q}|| \qquad (1)$$

It is a classic tool in morphological image processing and can be computed very efficiently by the algorithm described by Felzenszwalb & Huttenlocher (2012). We have implemented a highly performant, dependency-free version of this algorithm on our own data structures.

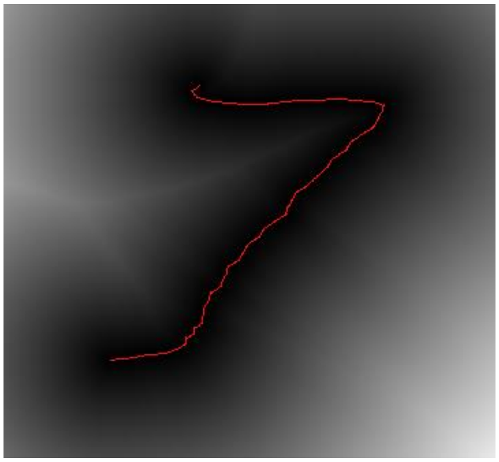


Figure 27: Distance transform of a sample trajectory (red)
The distance of each pixel to the trajectory is encoded in its brightness

In order to use the distance transform as a KPI for the difference between two obstacle polygons p_1, p_2 we can treat the outlines of the polygons as binary images B_1, B_2 and obtain the distance transform D_1 of B_1. A robust measure for the difference between the two polygons can be devised by sampling D_1 along the outline of p_2, e.g. by sampling D_1 at all locations of non-zero pixels in B_2 and obtaining the average distance per pixel by dividing by the number of pixels sampled:

$$E = \frac{1}{|\Omega_{\mathrm{B}_2}|} \sum_{\boldsymbol{p} \in \Omega_{\mathrm{B}_2}} D_1(\boldsymbol{p}) \qquad (2)$$

Where Ω_{B_2} is the set of non-zero pixels in B_2.

The value E gives a good measure of similarity between the shapes of the two polygons. However, it is agnostic of other features of the trajectories like car speed.

Intersection over union (IOU) between free-space quadrilateral areas

Intersection over Union (IOU) is one of the most common KPIs in the field of object detection. It is a specialization of the Jaccard Index met in set theory and later reformulated by Tanimoto (1958), for measuring the similarity between two bounding boxes, usually the ground truth and the detection. Its main advantage is its scale invariance. This attribute along with the easy calculation of its value made it a preferred measure of object detection accuracy, especially after it was included in the Pascal visual object classes benchmark (Everingham et al., 2010). The IOU measure is calculated by dividing the intersection of the two bounding boxes (sample sets) by their union, hence the name. In mathematical notation, this is given by:

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \qquad (3)$$

Where A and B are the two sample sets (free-space Ground Truth and detected free-space in our case). From its definition, IOU is bounded between 0 and 1 and this makes it an appropriate metric for measuring similarity (perfectly matched shapes have IOU = 1 while non-overlapping shapes have IOU = 0). However, it does not convey well the rotation error between the two bounding boxes. Additionally, as most generic metrics, it doesn't capture the severity of the error. In Figure 28 we can see some examples of IOU values on wrong detections of the same free-space area.

In Figure 28, the two safe detections that are depicted in examples (a) and (f) have lower IOU values than the rotated example of picture (e). Similarly, example detection (f) has a lower IOU value compared to that of example (b), which clearly reports a dangerous free-space area partially covering an obstacle. Nevertheless, IOU values of standalone affine transformations of the ground truth quadrilateral are smaller than the values of mixed transformations, therefore IOU is still an appropriate similarity measure.

From the examples above, IOU proves to be useful as an indicative measure of detection accuracy, however it must never be examined standalone, but rather in conjunction with other metrics that can provide misdetection severity assessment. Examples of IOU usage from the Automotive domain are plenty, usually by incorporating the Mean Average Precision, i.e. the mean value of the average precision per object in the scene, when acceptable detections are the ones with IOU value over a threshold. A common threshold used is 50%, therefore mAP_{50} is the usual metric met in relevant papers, like in the one from Li et al. (2020a) targeting parking slot detection.

More recently, researchers have proposed the use of more customized metrics to fit the ADAS use cases. Staying in the application domain of parking slot detection, which is a similar concept to free-space detection, Li et al. (2020a, 2020b) proposed the use of more descriptive features of slots, like their position, orientation and length for the assessment of True Positives. The disadvantage of this approach is the need of several heuristic thresholds for comparisons, which is acceptable in the case of parking slot detection, but less so in the free-space detection domain. A more suitable set of metrics for free-space detection is met in the paper from Do and Choi (2020), who propose the use of a combined score metric from two individual ones - an area score and a location score. This approach has only one disadvantage, which is the failure to capture the orientation mismatch between detection and ground truth in a solid way. The two scores used by Do and Choi (2020) are given by the following equations:

$$S_{\text{Area}} = \frac{min\{\text{Area}(G),\text{Area}(P)\}}{max\{\text{Area}(G),\text{Area}(P)\}},$$

$$S_{\text{Loc}} = \sqrt{\text{Area}(P')/\text{Area}(P)},$$

where G and P are the Ground Truth and Predicted parking slots respectively and P' is the scaled down location of P so that $P\,' \subset G$.

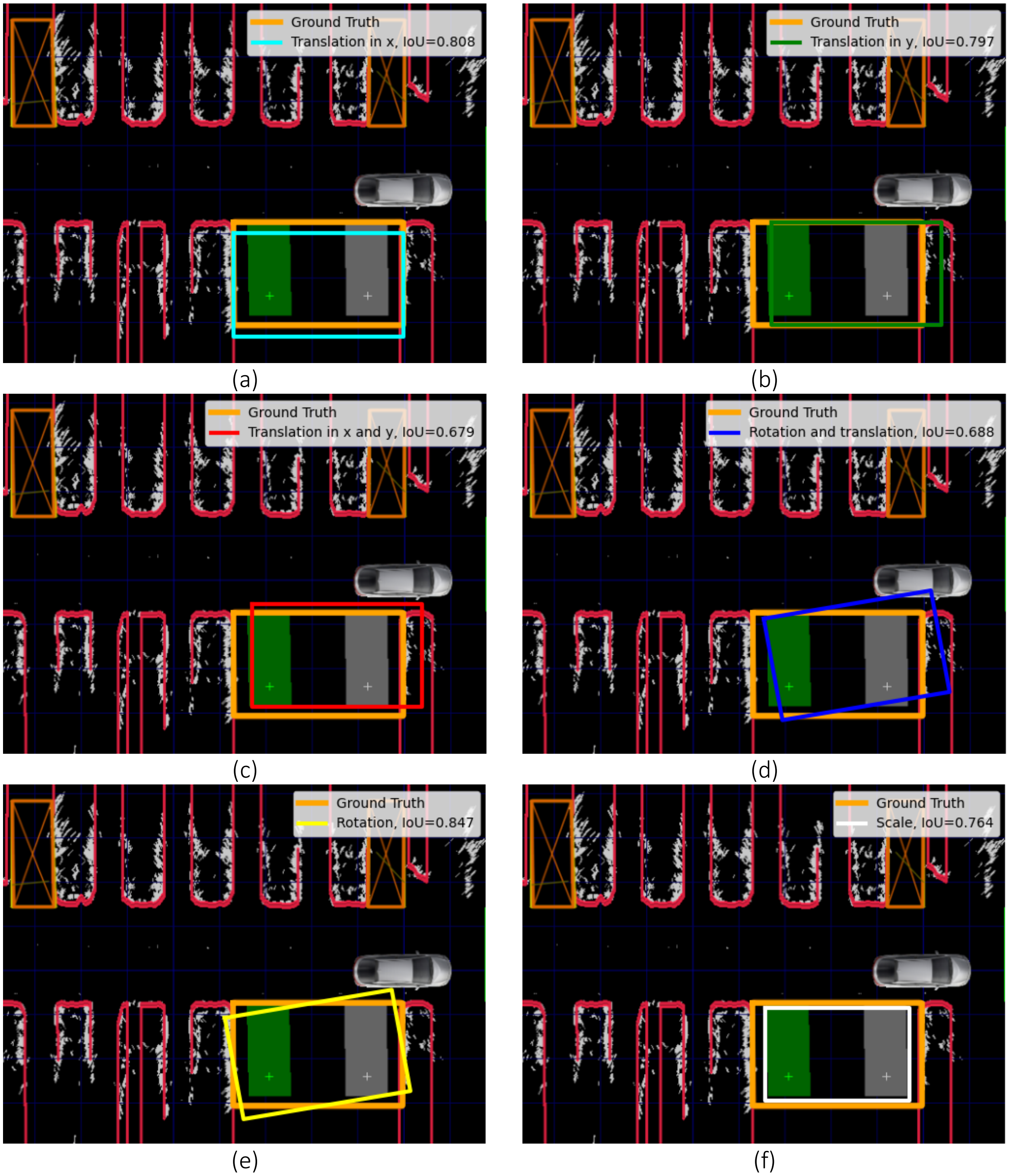


Figure 28: Examples of IOU values

Several examples of IOU values of partially erroneous detections, artificially generated from combined affine transformations of the ground truth. The transformation and IOU value are captured in the legend of each example.

2.2.2.4.3 Evaluation protocol

The evaluation protocol includes the following steps:

Step 1: Feed original images into the environmental reconstruction to result in occupancy grid map.

Step 2: Extract and store:

a) Referenced obstacle polygons,
b) Referenced free-space analysis in form of quadrilateral area as a *reference data*.

Step 3: Feed denoised images into the environmental reconstruction to result in occupancy grid map.

Step 4: Extract:

a) "Evaluating" obstacle polygons,
b) "Evaluating" free-space analysis in form of quadrilateral area,

as *evaluating data*.

Step 5: Compute differences between evaluating and referenced data:

a) Intersection over union (IOU) between the evaluating and referenced free-space area,
b) Distance transformation between the evaluating and referenced obstacle-polygons.

2.2.3 Cooperative awareness scenario

IEEE.802.11p (ETSI TC ITS, 2011) proposes cooperative awareness as an interactive framework of networking among vehicles, where every vehicle is transformed into a moving sensor platform that is capable of sharing information collected using its on-board sensors. This helps extending the operational range of autonomous vehicles, which otherwise suffer from blind spots and occlusions. Co-operative situational awareness promotes safe driving over a short range and improves traffic flow efficiency over a long range. CPSoSaware proposes a framework for cooperative perception and localization. Cooperative localization is achieved using both passive and active sensors and V2X for incorporating the position of other traffic agents into the scene awareness.

2.2.3.1 *Evaluated CPSoSaware components, related requirements and evaluation concept of the targeted components and use case*

Subsections 2.2.3.1.1 through 2.2.3.2 summarize the use cases, evaluation framework and data characteristics which will be considered through the co-operative awareness pillar of the automotive pilot.

2.2.3.1.1 Detailed use case concept

Cooperative awareness of autonomous vehicles from a localization perspective is directly linked to the output of *T3.3: Distributed and Coalitional AI supporting autonomic intelligence*. CAVs have the potential to enhance road safety and the overall performance of the transportation sector, through the strict control of their positions and motion planning actions. Furthermore, they can achieve increased scene analysis ability through the exchange of information by using wireless Vehicle-to-Vehicle or Vehicular Ad Hoc Network (V2V and V2I) communication technologies. For a better perception of the surrounding environment the members of a 5G Vehicular Ad Hoc Network (VANET) are required to have exact and timely knowledge of their position, but also of neighboring vehicles' positions. Cooperative awareness is critical for the efficient path and motion planning of CAVs.

For the evaluation of the cooperative awareness use case, a scenario is assumed with two vehicles and two pedestrians. Vehicle A is driving straight with a speed of less than 30 kph, reducing speed to turn right, while vehicle B is driving straight with a speed less than 30 kph. Vehicle A *detects* moving Pedestrian P1, *estimates motion of pedestrian* P1, informs neighboring vehicle regarding the driving event including *its ego position* and motion of detected pedestrian P1. Vehicle B detects the moving Pedestrian P2, estimates the motion of pedestrian P2, informs neighboring vehicle regarding the driving event including its ego position and motion of detected pedestrian P1. The evaluation scenario is depicted in Figure 29.

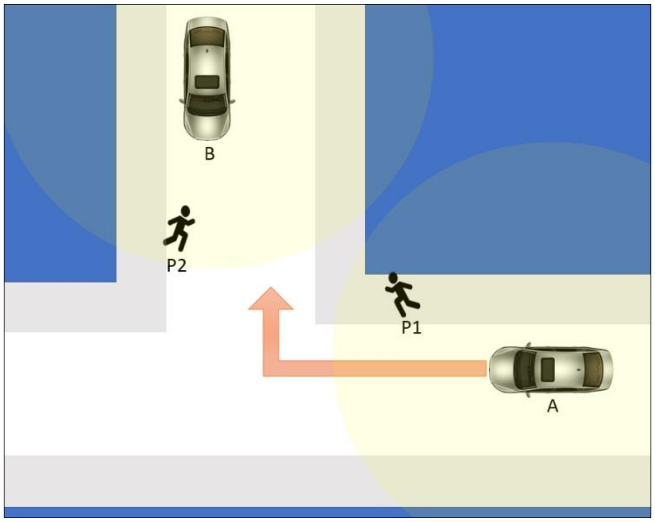


Figure 29: Cooperative awareness scenario

The scenario will be validated using Robotec V2X Simulator and Robotec Real World Simulator integrated with Cooperative Localization and Multimodal Scene Understanding. Figure 30 depicts the simulation of the abovementioned scenario with two vehicles and two pedestrians.

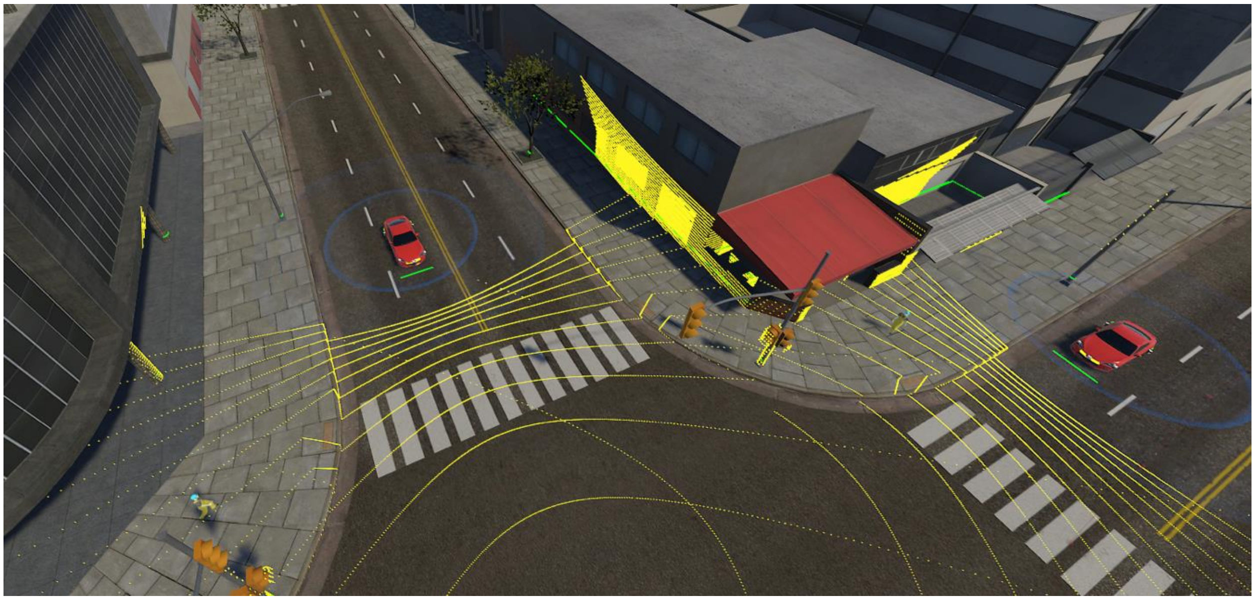


Figure 30: Simulation of two vehicles and two pedestrians scenario

Cooperative awareness use case scenario includes the following steps:

1. Scene analysis module of *T3.1: CP(H)S medium: Enabling Multimodal Sensing and Embedded Assisted and Augment Intelligence* provides relative range measurements for pedestrians P1 and P2.

2. Ego vehicle receives from its neighbors' information (i.e. scene analysis data, GPS, velocity, acceleration) exploiting V2V communication.

3. Fusion algorithm combines the different sources of information. The fusion algorithm is named *Extended Kalman Filter for Cooperative Awareness (EKF-CA)*.

4. Position and egomotion are calculated for each vehicle.

5. Position and motion are calculated for each visible pedestrian by each corresponding vehicle.

6. V2V communication layer publishes the acknowledged positions and motion vectors to all corresponding parties.

Multimodal scene analysis module will be performed using Pointpillar (Lang et al., 2019) and PVRCNN (Shi et al., 2020a) LiDAR based 3D object detection and will be fused with SqueezeDet 2D object detection (Wu et al., 2017) and DeepLab semantic segmentation (Wu et al., 2017). The scene analysis module will be evaluated with the average precision (AP) score evaluating three classes of detected objects: cars, pedestrians, and cyclists, and the overall mean average precision (mAP).

For the cooperative awareness part, three test cases are taken into account:

- Inter-vehicular distances can be extracted by scene analysis module of T3.1 using LIDAR and/or Camera.
- Inter-vehicular distances GPS positions.
- Inter-vehicular distances by the estimated positions by the cooperative awareness algorithm.

CARLA simulator generates GPS positions and EFK-CA errors. A visualization of the CARLA testing environment is depicted in Figure 31.

Metrics and key performance indicators

For the multimodal scene analysis, part Intersection over Union (IOU) of bounding boxes is computed to subsequently compute average precision (AP) and mean average precision (MAP). The following definitions apply:

$$IOU = \frac{P \cap G}{P \cup G}$$

P is the predicted bounding box and G the groundtruth bounding box. The general definition for the Average Precision (AP) is finding the area under the precision-recall curve and defined as:

$$\mathrm{AP@}n = \frac{1}{\mathrm{GTP}} \sum_{k}^{n} \mathrm{P@}k \times \mathrm{rel@}k$$

GTP refers to the total number of ground truth positives, n refers to the total number of documents you are interested in, P@k refers to the precision@k and rel@k is a relevance function. The relevance function is an indicator function which equals *1* if the document at rank *k* is relevant and equals *0* otherwise. Finally, mean average precision is simply the mean of all the queries that the use made:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP$$

The key performance indicator would be to achieve a state of the art mAP on the CARLA simulator according to KITTI benchmarks (Shi et al., 2020b).

For the localization module usually the GPS sensor is responsible for providing absolute position information. However, GPS is less reliable in challenging urban environments. To obtain highly accurate positioning solutions vehicles exploit V2V and integrated sensors like LIDAR, Cameras, IMU, etc., to generate, exchange and fuse heterogeneous measurements. This promising technique is known as Cooperative Localization (CL) and it has received increasing interest during the past few years. Therefore, evaluation of cooperative awareness could be derived by measuring the localization error attained by the different fusion algorithm, GPS, etc. An important evaluation metric is how accurate ego vehicle can measure inter-vehicular relative distances, to design more efficiently its motion actions. Ego vehicle should estimate its position, following the previous three steps, with an error of 1.24 m, instead of 5.91 m with GPS. Additionally, estimates neighbors' positions much more accurate than the received GPS.

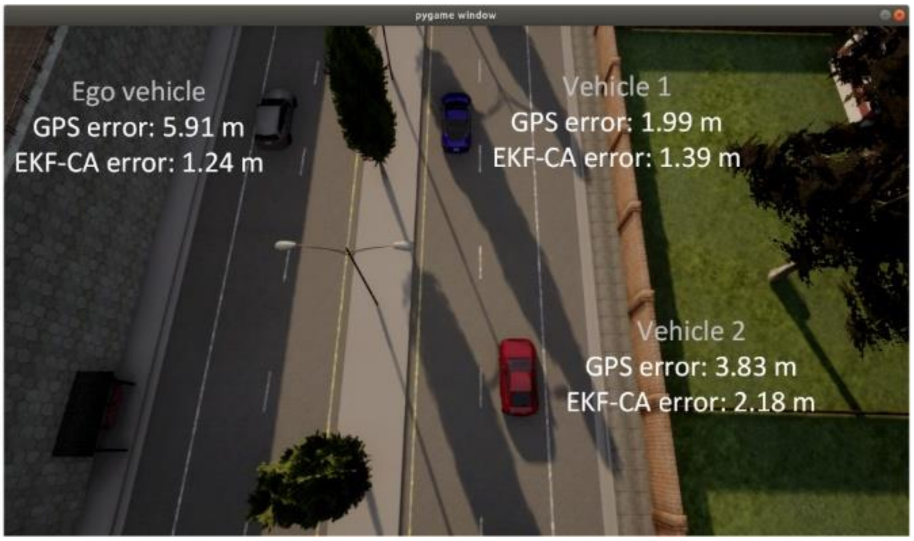
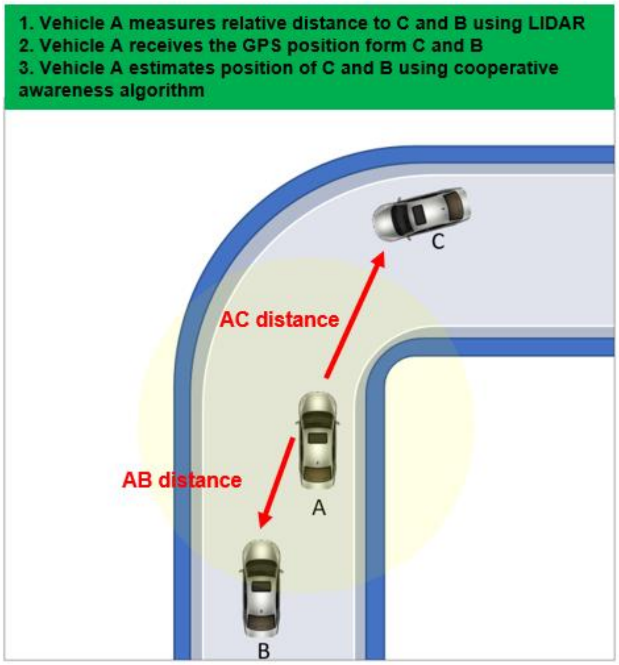


Figure 31: Cooperative awareness output



1. Vehicle A measures relative distance to C and B using LIDAR
2. Vehicle A receives the GPS position form C and B
3. Vehicle A estimates position of C and B using cooperative awareness algorithm

Figure 32: Inter-vehicular distances as a cooperative awareness evaluation metric

For the overall use case, *Awareness Quality Level* (AQL) is employed metric that uses the intersection of an actual number of neighbors and the number of neighbors discovered. AQL is defined as:

$$Awareness_k^T(i) = \frac{\left|N_k^T(i) \cap V_k^T(i)\right|}{V_k^T}$$

Where $\boldsymbol{V_k^T(i)}$ is the number of neighbours of vehicle i and $N_k^T(i)$ is the number of neighbours received by vehicle *i* within an area *k* at a certain time *T.* Figure 33 demonstrates a more generic use case for the calculation of Awareness quality level.

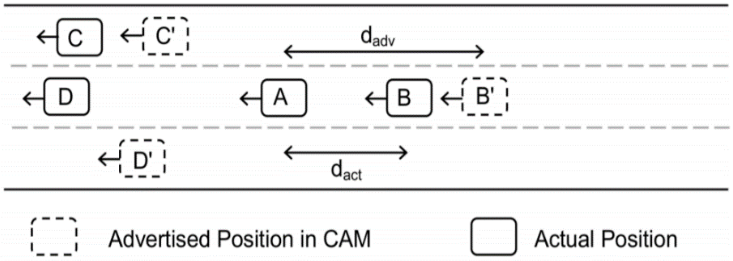


Figure 33: Awareness metric in a scenario with multiple vehicles

2.2.3.2 Data acquisition

Due to the required safety of maneuvers of autonomous vehicles, these vehicles require a high precision perception of their surrounding environment. All the possible information ranging from the details of motions of used ego vehicle to the motion of surrounding pedestrians can lead the vehicle to make an optimal decision at the right time to avoid any collision. This level of safety confidence can be just archived using an optimal fusion between the collected information of various mounted sensors on the vehicle.

In this section a short description of mounted sensors on Panasonic Automotive Systems Europe (PASEU) automated vehicle is introduced. The proposed system is a vision-based autonomous system including several components as explain below.

Cameras

Four wide angle ("fish-eye", 30 frame/sec with a 190° field of view) cameras which are mounted on the side mirrors, the front, and rear bumper of the test vehicle to detect the free spaces and obstacle around the vehicle simultaneously are used. The detailed specifications of the cameras are shown in Table 4.

Table 4: Technical specifications of PASEU cameras

Parameter	Information-Size
Sensor Model	Sony ISX016
Image Format	Parallel Output YUV422
Serializer Model	FPD Link-Ⅲ
Effective Area	H：1296×V：976
Output Resolution	H：1280×V：960 (Cropping)
Frame rate	30 (Frame/S)
Data logging	54 (MB/S)

Sonars

Sonar sensors (10 Hz low range up to 6 m) in front, rear, left, and right of the vehicle to collect additional data about the close object-obstacle to the vehicle.

Velodyne LiDAR Sensor

For the further validation of the performance of the vision-based parking system a 360-degree laser scanner is used. In recent modern “level five” of autonomous vehicles (e.g. Google car), LiDAR sensors are mainly used to collect the information of the surrounding area to feed the perception sections. LiDAR technology is still in its infancy, and cost-effective sensors are not yet readily available on the market. In the Panasonic driving platform, scanning LiDARs are therefore only used for validation. The technical specification of the used sensor is shown in Table 5.

Table 5: Technical specifications of the used laser scanner in PASEU

Parameter	Information-Size
Sensor Model	HDL-32E
Number of Channels	32Up to 100(m)
Range Accuracy	Up to ±2(cm)
Field of View (Vertical)	+10.67 to -30.67 (41.33) (Deg)
Angular Resolution (Vertical):	1.33 (Deg)
Field of View (Horizontal)	360 (Deg)
Angular Resolution (Horizontal/Azimuth)	0.1 – 0.4 (Deg)
Rotation Rate	5 – 20 (Hz)

Differential GPS and its internal Inertial Measurement Unit (IMU)

The position of the vehicle can be measure precisely at each position using an Applanix Differential GPS (DGPS). The precise position of the vehicle with the captured information of the surrounding of the vehicle helps the autonomous system to ensure the performance of other mounted sensors due to the accurate collected ground truth. The accuracy of the used DGPS sensor is listed in Table 6.

Table 6: Technical specifications of the mounted DGPS sensor on the test vehicle of PASEU

Parameter	DGPS	With Post Processing
Position (m)	0.3-0.5	0.02-0.05
Roll/Pitch (Deg)	0.015	0.015
Heading (Deg)	0.02	0.02

Vehicle network

The related information and data collected from the vehicle are transferred to each component of the system via a local CAN and FlexRay system of the vehicle. This internal information regarding each vehicle motion is updated as presented in Table 7.

Table 7: FlexRay messages and their update frequencies in PASEU test vehicle

Parameter	Update frequency (ms)
Steering wheel position	20
Gear position	5
Rack position	20
Wheels rotation counter	20
Wheels speed	20
Blinker information	10

Data logger

All the captured data of used sensors are stored on the onboard car-pc of the vehicle. Regarding synchronizing the data together, currently the live capture engine (Win7 64-bit with Intel(R) Xeon(R) CPU 3.50 GHz processor) which is responsible for data logging considers the time stamp of receiving the data (e.g. from each Camera) on the PC and stores them accordingly. The logger does not consider the capture time stamp of the data itself.

When the data logger receives a new data, it assigns a timestamp to it from a global time stamp service that is starting from 0.00000 second and is always incrementing.

Ego vehicle

In the automotive industry ego vehicle term is mainly used to refer to the test vehicle which is manipulating the requested tasks. In our system, the whole APS system is mounted on a test vehicle: Mercedes Benz C-Class 2014 (S204).

Figure 34 introduces a simplified version of the autonomous driving system which has been used in PASEU.

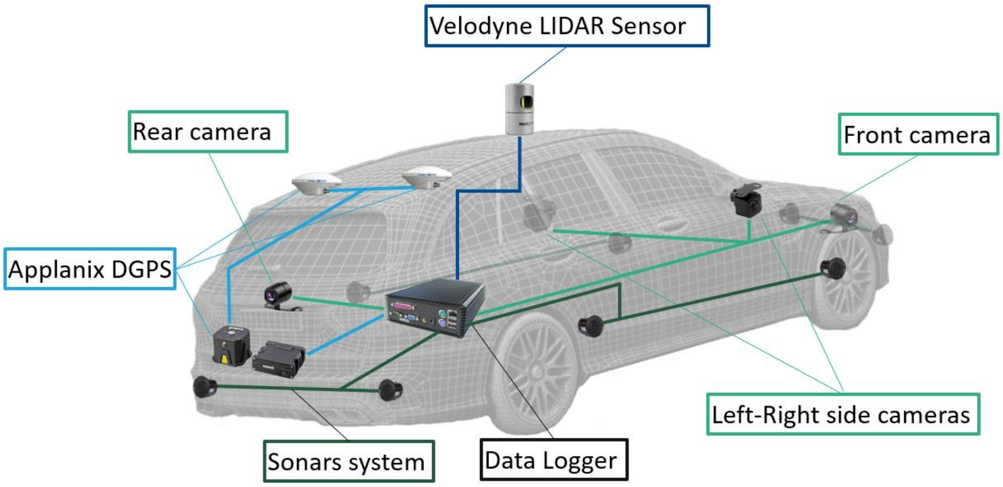


Figure 34: Panasonic autonomous driving system setup on one of the test vehicles

2.2.3.3 *Testing environments, testing procedure*

The testing environment can be considered as an end-to-end testing framework that is composed of a data-producing end (simulator) and data consumers (ROS nodes) on the other end. The simulator is used for producing data from sensors attached to vehicles. The ROS Bridge translates the sensor data to ROS compliant messages which are afterwards utilized by the ROS nodes.

More specifically, the simulator in the data producing end is CARLA[7], which is an open-source autonomous driving simulator. It is based on Unreal Engine for conducting the simulation and utilizes the OpenDrive standard for defining targets and urban settings. Simulation parameters can be controlled programmatically via a C++ or Python API. Moreover, it is built upon a scalable client-server architecture, in which the server is responsible for anything concerning the simulation, like scene rendering, updating physics, actors' state etc. A brief list of the simulator's distinct features includes:

- The traffic manager which is a built-in system used for enforcing realistic behaviors upon the vehicles.
- Various sensors for publishing information (e.g. LiDAR, RGB, depth, radar, IMU, GNSS etc.).
- A recorder for re-enacting the simulation step by step.
- ROS bridge for integrating CARLA to ROS.
- Easy creation and customization of assets.
- The scenario runner for describing routes and traffic scenarios.

In addition, the ROS-Bridge facilitates the bidirectional communication between the simulation environment and ROS runtime. The messages from CARLA are translated into ROS compliant messages

[7] https://carla.org/

under relevant ROS topics and at the same time messages originated from ROS get translated to CARLA commands. Finally, the Robot Operating System (ROS), which is a set of software libraries, tools and conventions, consists of a flexible framework for writing robot software and integrating heterogeneous modules in the form of ROS nodes written in C++ or Python. Therefore, this pluggable middleware constitutes the perfect facilitator for evaluating the algorithms, which are fed with synthetic data in real-time.

Figure 35 depicts an indicative setup of the environment, in which data produced by CARLA are passed to ROS and consumed initially by three ROS nodes which implement different pose estimation algorithms based on three different modalities (RGB, LiDAR, and GNSS). The poses produced are collected by another node and are stored into a CSV file or sent to a remote RabbitMQ server.

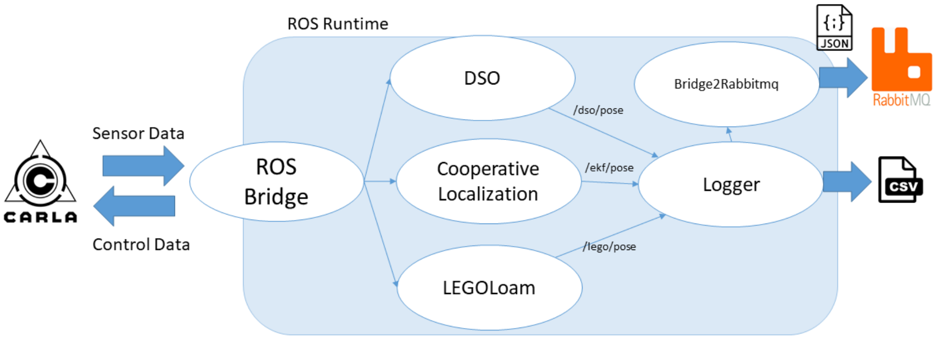


Figure 35: Indicative setup of the environment

3 Conclusion

This deliverable introduces the detailed description of CPSoSaware components’ evaluation trials. Testing environments, procedures, and acquired data are thoroughly discussed for all the use cases to provide the most reliable testing methods. In the Human-Robot Interaction in Manufacturing Environment section two use cases are described: a design operation continuum evaluation, and resilience and safety. In the Connected and Autonomous L3-L4 Vehicles section three use cases are described: human in the loop control, cybersecurity issues, and cooperative awareness. Testing and validation of the CPSoSaware system components allow to detect possible errors and provide necessary improvements, aiming to achieve best performance of the system and its elements. The next step of the system’s development is to provide further testing in final trials.

References

Ahlström, C., Jan, A., & Anund, A. (2010). *Detecting sleepiness by Optalert Final report. Virtual Prototyping and Assessment by Simulation.*

Ahlström, C., Anund, A., Fors, C., & Akerstedt, T. (2018). The effect of daylight versus darkness on driver sleepiness: a driving simulator study. *Journal of Sleep Research, 27*(3).

Åkerstedt, T., & Gillberg, M. (1990). Subjective and objective sleepiness in the active individual. *International Journal of Neuroscience*, *52*(1-2), 29-37.

Akerstedt, T., Peters, B., Anund, A., & Kecklund, G. (2005). Impaired alertness and performance driving home from the night shift: A driving simulator study. *Journal of Sleep Research, 14*(1), 17–20.

Anund, A. (2018). *Intra-individual difference in sleepiness and the effect on driving performance – a three-times repeated driving simulator study*. Paper presented at The 6th International Conference on Driver Distraction and Inattention, DDI2018. Gothenburg, Sweden.

Anund, A., Fors, C., & Ahlstrom, C. (2017). The severity of driver fatigue in terms of line crossing: a pilot study comparing day- and night time driving in simulator. *European Transport Research Review, 9*(2).

Athalye, A., Carlini, N., & Wagner, D. (2018, July). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning* (pp. 274-283). PMLR.

Boddupalli, S., & Ray, S. (2019, October). REDEM: Real-Time Detection and Mitigation of Communication Attacks in Connected Autonomous Vehicle Applications. In *IFIP International Internet of Things Conference* (pp. 105-122). Springer, Cham.

Bowman, D., Hanowski, R. J., Alden, A., Gupta, S., Wiegaud, D., Baker, S., Wierwille, W. 2012. *Development and Assessment of a Driver Drowsiness Monitoring System* (Raport no: FMCSA-RRR-12-008). Washington, DC: Federal Motor Carrier Safety Administration, 2012.

Do, H., & Choi, J. Y. (2020). Context-Based Parking Slot Detection with a Realistic Dataset. *IEEE Access, 8*, 171551-171559.

European Commission. (2020). *General Safety Regulation – Technical study to assess and develop performance requirements and test protocols for various measures implementing the new General Safety Regulation, for accident avoidance and vehicle occupant, pedestrian and cyclist protection.* Luxembourg.

ETSI TC ITS. (2011) Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Technical Report TS 102 637-2 V 1.2.1.

Everingham, M., Gool, L. V., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision, 88(2),* 303-338.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of computing*, *8*(1), 415-428.

Genchi, G., Zanella, A., Cacciatore, A., Kapsalas, P. (2020). Preliminary Evaluation and Assessment of CPSoSaware Platform. CPSoSaware project report (D6.3).

Gerosavva, N., Arvanitis, G., Stagakis N., Zanella A., Cacciatore, A., Genchi, G., Kapsalas, P., Ferrer Riera, J., Tzifas, M., Fytilis, I., Markou, P. (2021). Human factors and metrics analysis. CPSoSaware project report (D2.1).

Ingre, M., Akerstedt, T., Peters, B., Anund, A., Kecklund, G., & Pickles, A. (2006). Subjective sleepiness and accident risk avoiding the ecological fallacy. *Journal of Sleep Research, 15(2),* 142–148.

Kaida, K., Takahashi, M., Akerstedt, T., Nakata, A., Otsuka, Y., Haratani, T., & Fukasawa, K. (2006). Validation of the Karolinska sleepiness scale against performance and EEG variables. *Clinical Neurophysiology, 117*(7), 1574–1581.

Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology, 3(*3), 203–220.

Khan, S. K., Shiwakoti, N., Stasinopoulos, P., & Chen, Y. (2020). Cyber-attacks in the next-generation cars, mitigation techniques, anticipated readiness and future directions. *Accident Analysis & Prevention*, *148*, 105837.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12697-12705).

Li, W., Cao, H., Liao, J., Xia, J., Cao, L., & Knoll, A. (2020a). Parking Slot Detection on Around-View Images Using DCNN. *Frontiers in Neurorobotics, 14,* 46.

Li, W., Cao, L., Yan, L., Li, C., Feng, X., & Zhao, P. (2020b). Vacant Parking Slot Detection in the Around View Image Based on Deep Learning. *Sensors, 20(7),* 2138.

Lu, J., Sibai, H., Fabry, E., & Forsyth, D. (2017). Standard detectors aren't (currently) fooled by physical adversarial stop signs. *arXiv preprint arXiv:1710.03337*.

Nassi, D., Ben-Netanel, R., Elovici, Y., & Nassi, B. (2019). Mobilbye: Attacking ADAS with camera spoofing. *arXiv preprint arXiv:1906.09765*.

Parkinson, S., Ward, P., Wilson, K., & Miller, J. (2017). Cyber threats facing autonomous and connected vehicles: Future challenges. *IEEE transactions on intelligent transportation systems*, *18*(11), 2898-2915.

Petit, J., Stottelaar, B., Feiri, M., & Kargl, F. (2015). Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe*, *11*(2015), 995.

Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2019). A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS)*, *22*(3), 1-30.

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., & Li, H. (2020a). PV-RCNN: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10529-10538).

Shi, S., Wang, Z., Shi, J., Wang, X., & Li, H. (2020b). From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*.

Tanimoto, T. (1958). An Elementary Mathematical theory of Classification and Prediction. Internal IBM Technical Report.

Thomas, M. J. W. (2009). *Human factors: Fatigue Risk Management System (FRMS)*.

Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics, vol. 1.

Van Loon, R. J., Brouwer, R. F. T., & Martens, M. H. (2015). Drowsy drivers' under-performance in lateral control: How much is too much? Using an integrated measure of lateral control to quantify safe lateral driving. *Accident Analysis and Prevention, 84*, 134–143.

Wu, B., Iandola, F., Jin, P. H., & Keutzer, K. (2017). Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 129-137).

Zanella, A., Cacciatore A., Genchi, G., Kapsalas, P., Sengupta S. (2021). Requirements and Use Cases. CPSoSaware project report (D1.2).