



## D1.1 SUPPORTIVE, MOTIVATING AND PERSUASIVE APPROACHES, TOOLS & METRICS

*Authors* Beatriz Gallego-Nicasio Crespo (ATOS), Gianmarco Genchi (CRF), Evgeny Shindin (IBM), Wojciech Jaworski (RTC), Aris Lalos (ISI), Apostolos Fournaris (ISI), Evangelos Haleplidis (ISI), Christos Didachos (ISI), Elena Theodoropoulou (ISI), Stavros Nousias (ISI), Konstantinos Berberidis (ISI), Francesco Regazzoni (USI), Harold Ship (IBM), Neofytos Gerosavva (8BELLS), Petros Kapsalas (PASEU), Pekka Jääskeläinen (TAU), Eva Zacharaki (UPAT), Gerasimos Arvanitis (UPAT), Javier Fernández (I2CAT), Marisa Catalán Cid (I2CAT), UoP, Ruben Trapero (ATOS)

*Work Package* WP1 CPSoS Requirements, use cases, Specifications and Architecture

### Abstract

This report constitutes the output of task T1.1 “SoA analysis, technological selection and benchmarking of best practices” and provides a review of the state-of-the-art methodologies and best practices, techniques and mechanisms, technologies and solutions for capturing requirements, developing and maintaining dependable Cyber-Physical System of Systems (CPSoS). The document identifies and analyses existing methodologies, techniques and solutions for each of the five phases of the CPSoS Aware Lifecycle, namely elicitation of requirements, model-based design, simulation, operation and monitoring of the CPSoS; with a specific focus on addressing the three CPSoS Aware pillars: Artificial Intelligence, Model-based Design/Computing and Cybersecurity. Furthermore, the review of the state-of-the-art also takes into account the requirements and specific characteristics of the two project use cases, i.e. Connected and Autonomous Vehicles (CAVs) and Human-robot interaction in manufacturing, to guarantee applicability of the CPSoS Aware solution to these domains. The ultimate aim of this document is to provide a list of appropriate candidates to support the implementation of the CPSoS Aware architecture, and which will be further extended, adapted, integrated, deployed and tested in the context of the corresponding project technical work-packages.



Funded by the Horizon 2020 Framework Programme  
of the European Union

## Deliverable Information

<i>Work Package</i>	WP1 CPSoS Requirements, use cases, Specifications and Architecture
<i>Task</i>	SoA analysis, technological selection and benchmarking of best practices.
<i>Deliverable title</i>	D1.1 Supportive, Motivating and Persuasive Approaches, Tools and Metrics
<i>Dissemination Level</i>	PU
<i>Status</i>	F: Final
<i>Version Number</i>	1.0
<i>Due date</i>	30/06/2020

## Project Information

---

<i>Project start and duration</i>	01/01/2020 – 31/12/2022, 36 months
<i>Project Coordinator</i>	Industrial Systems Institute, ATHENA Research and Innovation Center 26504, Rio-Patras, Greece
<i>Partners</i>	<ol style="list-style-type: none"><li>1. ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS (ISI) the Coordinator</li><li>2. FUNDACIO PRIVADA I2CAT, INTERNET I INNOVACIO DIGITAL A CATALUNYA (I2CAT),</li><li>3. IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD (IBM ISRAEL)</li><li>4. ATOS SPAIN SA (ATOS),</li><li>5. PANASONIC AUTOMOTIVE SYSTEMS EUROPE GMBH (PASEU)</li><li>6. EIGHT BELLS LTD (8BELLS)</li><li>7. UNIVERSITA DELLA SVIZZERA ITALIANA (USI),</li><li>8. TAMPEREEN KORKEAKOULUSAATIO SR (TAU)</li><li>9. UNIVERSITY OF PELOPONNESE (UoP)</li><li>10. CATALINK LIMITED (CATALINK)</li><li>11. ROBOTEC.AI SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA (RTC)</li><li>12. CENTRO RICERCHE FIAT SCPA (CRF)</li><li>13. PANEPISTIMIO PATRON (UPAT)</li></ol>
<i>Website</i>	<a href="http://www.CPSoSAware.eu">www.CPSoSAware.eu</a>

<i>VERSION</i>	<i>DATE</i>	<i>SUMMARY OF CHANGES</i>	<i>AUTHOR</i>
0.1	26/03/2020	Initial Draft circulated to the Consortium	Ruben Trapero (ATOS)
0.2	26/03/2020	Revised ToC	Beatriz Gallego-Nicasio (ATOS)
0.3	14/04/2020	ToC revised and extended by section leaders (CRF, RTC, UOP, ISI)	Beatriz Gallego-Nicasio (ATOS)
0.4	16/04/2020	Updated contributions by partners (UoP) sect. 3 and 6	Beatriz Gallego-Nicasio (ATOS)
0.6	05/05/2020	Integrating contributions from partners to sections 3-7	Beatriz Gallego-Nicasio (ATOS)
0.8	03/06/2020	Integrating second round of contributions from partners to sections 3-7	Beatriz Gallego-Nicasio (ATOS)
0.9	09/06/2020	Integrating third round of contributions in sections 3, 5 and 7	Beatriz Gallego-Nicasio (ATOS)
0.91	10/06/2020	Integrating third round of contributions in section 4	Beatriz Gallego-Nicasio (ATOS)
0.96	12/06/2020	Complete version ready for internal review	Beatriz Gallego-Nicasio (ATOS)
0.97	16/06/2020	Updated section 5 – missing contribution from UoP (section 5.4)	Beatriz Gallego-Nicasio (ATOS)
0.99	03/07/2020	Updated document addressing comments from CTL and ISI. Version ready for approval.	Beatriz Gallego-Nicasio (ATOS)

	<i>NAME</i>
<i>Prepared by</i>	ATOS
<i>Reviewed by</i>	Catalink, ISI
<i>Authorised by</i>	ISI
<i>DATE</i>	<i>RECIPIENT</i>
25/03/2020	Project Consortium
30/06/2020	European Commission

## Table of contents

Executive summary .....	15
1 Introduction .....	17
1.1 Document structure .....	17
1.2 Definitions and Acronyms .....	17
2 Research Methodology .....	22
3 Techniques and tools for definition of CPSoS requirements and KPIs.....	25
3.1 Determination of techniques for the identification of use cases Requirements .....	25
3.1.1 Volere Method – Manufacturing use case .....	25
3.1.2 Techniques and tools for definition of requirements and KPIs for the Automotive Use-Case	26
3.2 KPI methodology (CERBERO outcomes) .....	29
3.3 Customization to world CPSoS Aware use cases .....	30
3.4 Use of the methodology developed in the Autonomous Cars use case .....	32
3.5 Use of the methodology developed in the Manufacturing use case .....	34
4 Techniques and tools for model-based design of CPSs: the MODD approach .....	38
4.1. Modelling techniques and tools .....	38
4.1.1 Modelling Techniques .....	38
4.1.2 Modelling Tools .....	39
4.2 Software Profiling Tools.....	43
4.3 Evaluation of task scheduling problems .....	44
4.3.1 Scheduling of preventive maintenance in a CPSoS.....	44
4.3.2 Optimization of task scheduling of under fatigue [39] .....	44
4.3.3 CPSoS scheduling framework for quality prediction and manufacturing control .....	46
4.4 Model optimization .....	47

4.4.1	Commercial Software .....	47
4.4.2	A&D Domain Modelling and Analysis Software .....	55
4.4.3	Optimization Packages for Architecture .....	57
4.4.4	Usage for Architecture Optimization .....	63
4.5	Techniques and tools for the design of CP(H)SoSs .....	63
4.5.1	Hardware Acceleration Using Vitis .....	63
4.5.2	Profiling the Application .....	67
4.5.3	Real-time monitoring .....	71
5	Techniques and tools enabling the simulation of CPSs .....	82
5.1	Autonomous Driving scenario .....	82
5.1.1	Introduction.....	82
5.1.2	Requirements and description of components .....	82
5.1.3	AV/ADAS Simulators .....	84
5.1.4	Drivers behaviour modelling .....	91
5.1.5	V2X Communication modelling .....	93
5.1.6	Cybersecurity scenarios simulation. ....	95
5.1.7	Conclusions. Selection of tools for prototyping.....	98
5.2	Human-Robot interaction scenario .....	99
5.2.1	Introduction.....	99
5.2.2	Description and Requirements.....	99
5.2.3	Industrial Robot simulation .....	100
5.2.4	Human simulation .....	104
5.2.5	Conclusions. Suggested approach .....	104
5.3	Architecture level Simulators .....	104
5.3.1	Architecture-level CPU Simulators .....	105

5.4	Intra-communication Simulation Frameworks and Models .....	110
5.4.1	Critical characteristics of the selected Simulation Environments.....	110
5.4.2	CPSoS-Aware Intra-communication Network Simulators .....	111
6	Techniques, tools and best practices for the operation of CPSs .....	117
6.1	Individual CPSs AI solutions .....	117
6.1.1	Benefits of distributed machine learning in coalition environments.....	117
6.1.2	Coalition in CPSoS.....	118
6.1.3	Coalition in a cooperative road infrastructure system .....	118
6.1.4	Coalition Clustering Communication in Vehicular CPSoS .....	119
6.2	Deep-priors-driven scene understanding.....	120
6.2.1	3D object detection from 2D images.....	121
6.2.2	Multimodal fusion for object detection .....	124
6.2.3	Point cloud-based scene understanding .....	129
6.3	Model Compression and acceleration approaches .....	134
6.3.1	Parameter pruning and sharing.....	136
6.3.2	Low rank factorization .....	136
6.3.3	Transferred/compact convolutional filters.....	137
6.3.4	Knowledge distillation .....	138
6.4	Multi-modal Localization for Connected and Autonomous Vehicles .....	138
6.4.1	Multi-modal cooperative localization approaches .....	140
6.4.2	Related issues.....	142
6.5	Distributed framework with cyber-physical modelling.....	144
6.5.1	Overview of distributed computation models.....	144
6.5.2	Connected and autonomous vehicles as distributed CPS.....	150
6.5.3	Connected and collaborative industrial robots .....	156

6.6	CPS Commissioning and Inter and Intra CPS Communication .....	160
6.6.1	Inter CPS Communication.....	160
6.6.2	Intra CPS Communication.....	164
6.7	Extended reality tools for improving safety and situational awareness of the human in the loop	171
6.7.1	Situational awareness of the human in the loop.....	171
6.7.2	Related works and use cases using AR to increase situational awareness .....	173
7	Techniques and tools for monitoring CPSs infrastructures .....	176
7.1	Algorithms and techniques for monitoring KPIs in CPSs.....	176
7.1.1	Monitoring and measurements.....	176
7.1.2	Online and offline monitoring .....	177
7.1.3	Algorithms for the selection of an appropriate KPI .....	177
7.1.4	Quality criteria regarding KPIs [388].....	177
7.1.5	KPI Visualization .....	178
7.1.6	AR application for KPI visualization on a full productive line .....	179
7.1.7	Software and tools for KPI monitoring .....	180
7.2	Cybersecurity primitives, monitoring techniques and tools.....	182
7.2.1	CPSoSAAware ecosystem model .....	183
7.2.2	Cybersecurity landscape.....	190
7.2.3	Security primitives and mechanisms for protection.....	205
7.2.4	Monitoring security: tools and techniques.....	207
7.2.5	Assessment of the Security of a System .....	216
8	Conclusions .....	218
	References.....	219





## List of tables

Table 1 Template for defining a system requirement. ....	31
Table 2 Tools and Techniques matrix .....	42
Table 3 Head and eye-based metrics .....	75
Table 4 General comparison of simulation tools for autonomous driving .....	88
Table 5 Comparison of available sensors in simulation tools .....	89
Table 6 Comparison of simulators in terms of control and communication features .....	90
Table 7. Comparison of simulators in terms of machine learning support.....	91
Table 8 Potential cyber-attacks in V2X communications [167][168][169][170][171][172][173][174][175].	95
Table 9 Comparison of features available in simulators analysed for Human-Robot Interaction scenario	103
Table 10 algorithmic compression and model acceleration and their characteristics .....	135
Table 11 Summary of the comparison of various techniques based on transferred convolutional filters	138
Table 12. Survey papers related to WSN and Robotics .....	141
Table 13 Summary of multi-modal CL in VANET.....	142
Table 14 Comparison between the DSCR and C-V2X.....	162
Table 15. <i>Wireless Communication Technologies</i> Ranking .....	170
Table 16 Quality criteria and the corresponding motivation regarding KPIs .....	178
Table 17 Cyber physical system Assets: classes and categories .....	185
Table 18 Threat agents and attacks in the V2X domain .....	199
Table 19 Threats and Attacks on the Device Perception Layer (Sensors and Actuation) .....	202
Table 20 Threats and Attacks on the Device Application Layer (Sensors and Actuation).....	204

## List of figures

Figure 1 CPSoSAAware lifecycle used in the research methodology .....	22
Figure 2: The CPSoSAAware Pillars .....	23
Figure 3 Volere Requirements shell as a guide to writing each requirement.....	25
Figure 4 Overview of the requirement development.....	27
Figure 5 Traceability of requirements .....	28
Figure 6 Example for Analysing Customer Requirements and Deriving a Linked System Specification.....	29
Figure 7 Example for Analysing System Requirements and Deriving Linked Software, ECU, Mechanics Requirements.....	29
Figure 8 Interactions between the different elements of the proposed validation pipeline. Dashed lines represents future developments to connect the decision and perception .....	33
Figure 9 General overview of the Vitis framework.....	63
Figure 10 Data flow between the host and kernel. The FPGA hardware platform, on the right-hand side, contains the hardware accelerated kernels, global memory along with the DMA for memory transfers. Kernels can have one or more global memory interfaces and are pro .....	65
Figure 11 Vitis compile flow .....	66
Figure 12 Vitis hardware generation flow .....	67
Figure 13 Vitis optimization flow .....	69
Figure 14 Key components of the Vitis unified software platform .....	70
Figure 15 Vitis AI inference library .....	70
Figure 16 Vitis AI optimizer .....	71
Figure 17 Example of model pruning.....	71
Figure 18 Description of the needed physiological function for the Driver state diagnostic [134]. .....	73
Figure 19 Different types of factors that can be monitored for identifying the driver's status.....	74
Figure 20 Different driving situations and corresponding messages from a real-time monitoring system. 74	
Figure 21 Pipeline for fatigue estimation based on face analysis [136].....	76

Figure 22 Flow chart of our approach to classify driver distraction level. An example from the Distracted Driver dataset for "Drinking" action. ....	77
Figure 23 System architecture of real time car driver's condition monitoring system.....	78
Figure 24 Pipeline of the real time safety method. ....	80
Figure 25 Definition of safety states associated with a crane. ....	80
Figure 26 Definition of monitor states with safety supervisor. ....	81
Figure 27 Robotec Simulation -view with LIDAR rays' projection.....	84
Figure 28 LIDAR point cloud in Robotic Simulation .....	85
Figure 29 Screenshot from AirSim simulator.....	85
Figure 30 Screenshot from Carla simulator .....	86
Figure 31 Screenshot from Apollo Auto Simulator .....	86
Figure 32 Screenshot from Deepdrive simulator.....	87
Figure 33 Screenshot from LGSVL Simulator .....	87
Figure 34 Screenshot from Udacity Self-Driving Car Simulator .....	88
Figure 35 Screenshot from MADRaS simulator .....	88
Figure 36 Diagram of parametric model for driving behaviour modelling [5] .....	92
Figure 37 Screenshot from Gazebo simulator .....	100
Figure 38 Screenshot from NVIDIA Isaac simulator .....	101
Figure 39 Screenshot from Webots simulator.....	101
Figure 40 Screenshot from Coppelisim simulator.....	102
Figure 41 Screenshot from Process Simulate simulator .....	103
Figure 42 Configuration trade-off between speed and accuracy (based on [207]).....	107
Figure 43 Overview of <i>gem5-gpu</i> architecture with an example configuration. ....	109
Figure 44 OMNeT++ Basic Simulation Network.....	111
Figure 45 NS-3 Basic Simulation Model.....	114

Figure 46 A diagram of the cooperative overtaking assistance system with the critical zones .....	119
Figure 47. Coalition into Vehicular CPS [218] .....	120
Figure 48. Object detection using multi-modal signals for perception.....	121
Figure 49. Parameters vs execution time (ms) for different traffic sign detectors .....	123
Figure 50. Real-time performance among several state-of-the-art semantic segmentation architectures comparing the operations (GFLOPs.....	123
Figure 51. Multi-View 3D object detection network (MV3D) .....	127
Figure 52. AVOD Architecture .....	127
Figure 53. Architectures of the four multi-modal objection detection approaches [319] .....	128
Figure 54. Different fusion schemes [319] .....	129
Figure 55. VoxelNet [327] architecture .....	130
Figure 56. PointPillar[331] Architecture .....	131
Figure 57. PointNet Architecture [325] .....	131
Figure 58. Frustum PointNets [312] for 3D object detection .....	132
Figure 59. An overview of the dense PointFusion[329] architecture .....	132
Figure 60. F-ConvNet architecture .....	133
Figure 61. The overall framework of our part-aware and aggregation neural network for 3D object detection .....	134
Figure 62 Pruning a filter results in removal of its corresponding feature map and related kernels in the next layer.....	136
Figure 63. Low rank factorization .....	137
Figure 64. Example of VANET .....	140
Figure 65. NLOS .....	143
Figure 66. A graphical representation of a time-varying network with N=6.....	144
Figure 67. Cost-coupled setup where each agent $i$ only knows $f_i$ and $X$ [105].....	145
Figure 68. Cost-coupled setup where each agent $i$ only knows $f_i$ and $X$ [105].....	146

Figure 69. Constraint-coupled optimization [105].....	147
Figure 70. Set of data points that can be fit using a polynomial model (i.e., linear in the parameters). The coefficients of the polynomial are obtained with a regression approach. ....	147
Figure 71. Regression problem over a network of 4 agents. ....	148
Figure 72. Concept of DCPS for autonomous and connected vehicles.....	152
Figure 73. Simple formation .....	153
Figure 74. 2-D Lattice graphs with different boundary conditions .....	154
Figure 75 Fully connected multilayer neural network .....	156
Figure 76 Pipeline overview .....	158
Figure 77 Regressed trajectory taken from two distinct views [119]. ....	158
Figure 78. initial path of the manipulator (red) and the corresponding proactive one (blue) [119]. ....	159
Figure 79. Proactive paths computed for one of the experiments of the first group [119]. ....	159
Figure 80. Specific case of a two-joints robot and presents the set of processing allowing a real time and explicit computation of the configuration freespace. ....	160
Figure 81. Network hierarchy layers for the connected autonomous cars use case.....	164
Figure 82. The ZigBee protocol is defined by layer 3 and above. It works with the 802.15.4 layers 1 and 2. ....	165
Figure 83 In a mesh network, each node communicates with its closest neighbour as conditions permit. Note that there are alternate paths between any two nodes.....	166
Figure 84. Model of situation awareness proposed by Endsley. ....	173
Figure 85. Comprehension problems in tele-maintenance scenarios. ....	174
Figure 86 Support of Driver Assistance System .....	175
Figure 87 Example of different visualization graphs.....	179
Figure 88 Conceptualization of using AR application for KPIs visualization on a full productive line [389]. ....	180
Figure 89 ENISA Threat Taxonomy .....	192
Figure 90 Threat Modelling Methods and main features.....	194

Figure 91 WSN attacks classification- H2020 project Anastacia [229]..... 198

## Executive summary

This report constitutes the output of task T1.1 “SoA analysis, technological selection and benchmarking of best practices” and provides a review of the state-of-the-art methodologies and best practices, techniques and mechanisms, technologies and solutions for capturing requirements, developing and maintaining dependable Cyber-Physical System of Systems (CPSoS).

The document identifies and analyses existing methodologies, techniques and solutions for each of the five phases of the CPSoS-Aware Lifecycle, namely elicitation of requirements, model-based design, simulation, operation and monitoring of the CPSoS; with a specific focus on addressing the four CPSoS-Aware pillars: Artificial Intelligence, Model-based Design/Computing, Cybersecurity and Extended Reality Interfaces. Furthermore, the review of the state-of-the-art also takes into account the requirements and specific characteristics of the two project use cases, i.e. Connected and Autonomous Vehicles (CAVs) and Human-robot interaction in manufacturing, to guarantee applicability of the CPSoS-Aware solution to these domains. The ultimate aim of this document is to provide a list of appropriate candidates to support the implementation of the CPSoS-Aware architecture, and which will be further extended, adapted, integrated, deployed and tested in the context of the corresponding project technical work-packages.

More specifically, the document firstly reviews techniques and methodologies for the definition of requirements of CPSoS. The document presents the Volere method, used in the manufacturing use case, and some techniques used in the Automotive domain for the formalization of Key Performance Indicators (KPIs), which ensure the validation and traceability of the CPSoS requirements.

Secondly, this report reviews techniques and tools for the modelling of the architecture and design of CPSs. In particular, the document identifies general-purpose modelling techniques and tools (e.g. SysML, MATLAB), software analysis tools based on state-of-the-art FPGA platform, modelling optimization software that allow black box input/output data flow integration between different models/tools, A&D domain modelling and analysis and other packages that can be used for architecture optimization.

Thirdly, the document describes tools and techniques enabling the simulation of CPSs, specifically in the context of the autonomous driving scenario and in the human-robot interaction scenario. But also, the document provides a review of architectural level simulators that permit simulating various systems with various configurations and architectural level enhancements, which permit verifying their functional correctness as well as their timing characteristics.

Fourthly, the document revises best practices related to the operation of CPSs in the node level. Specifically, the document investigates AI solutions for individual CPSs, deep priors driven approaches for multimodal scene understanding, localization and path planning, compression and acceleration approaches for deep architectures, distributed frameworks for cyber-physical modelling, communication protocols and AR tools to facilitate situational awareness of the human in the loop.

Lastly, and focusing on the monitoring of the CPSoS, the document presents algorithms, techniques and tools for the definition of monitoring KPIs, which shall serve two purposes. On the one hand, KPIs give users of the CPSoS visibility on the quality and efficiency offered by the system at operational level, create an analytical basis to help for better decision making, offer a comparison that measures the degree of performance change over time, and help to give attention on what really matters most. On the other hand, monitoring system operation is part of the continuous improvement process of CPSoS-Aware that uses the

values obtained from KPI monitoring as input for the Simulation and Design phases and thus, learning, reconfiguring and adapting the CPSoS to changes in the context. Moreover, the document presents an in-depth analysis of the most relevant security aspects that should be considered in CPSoS monitoring, including a review of the cybersecurity landscape, a survey of security primitives and mechanisms for protecting CPSoS assets and a report of widely-adopted tools and techniques for monitoring and assessing security properties.



# 1 Introduction

This deliverable is the output of task T1.1 “SoA analysis, technological selection and benchmarking of best practices” and provides a review of existing methodologies and best practices, techniques and mechanisms, technologies and solutions that can support the complete lifecycle of CPSoSAware and specifically, for the elicitation of requirements, model-based design, simulation, operation and monitoring of Cyber-Physical System of Systems (CPSoS). The aim of this document is to identify a wide range of candidates available in the market and to provide a preliminary analysis of first, their suitability for addressing the project innovation objectives as well as functional and non-functional technical requirements; and second, their applicability to the specific domains of the two project use cases: connected and autonomous vehicles and human-robot interaction in a manufacturing environment.

This document shall serve as input for the other tasks in WP1, namely T1.2 “use cases Specifications and Use-Case Scenarios for dependable CP(H)Ss” and T1.3 “CPSoSAware System Specifications and Architecture”, and for the other technical work-packages in the project, mainly WP2 “Virtual User/Physical Environment Models, CPS Models and orchestration support tools”, WP3 “Model based CP(H)S Layer Design and Development supporting Distributed Assisted, Augmented and Autonomous Intelligence”, WP4 “CPSoSAware System Layer Design and adaptation of dependable CP(H)SoS”; which based on the candidates and analysis presented here will select the most appropriate methodologies, techniques and solutions to implement the different layers of the CPSoSAware architecture.

## 1.1 Document structure

This document is structured into eight major sections:

- **Section 1** introduces the document, outlining its structure, and identifying terms and acronyms used across the document.
- **Section 2** describes the research methodology adopted in the document to review the state of the art of all different aspects of interest for the CPSoSAware project, and which is developed in the following sections 3 up to 7.
- **Section 3** presents relevant methodologies and tools for the definition of requirements of the two use cases of the project.
- **Section 4** revises techniques and tools for model-based design of Cyber-Physical Systems.
- **Section 5** surveys most relevant techniques that enable simulation of the Autonomous driving scenario and the Human-Robot scenario.
- **Section 6** compiles techniques, tools and best practices for the operation of Cyber-Physical Systems.
- **Section 7** describes different methodologies, techniques and tools for monitoring Cyber-Physical Systems, including mechanisms to protect and monitor the CPS from a security perspective.
- **Section 8** concludes the document.

## 1.2 Definitions and Acronyms

Below are listed the most relevant acronyms used in the document and recurring definitions:

Acronym / Term	Definition
1P1-ES	Single-objective Evolution Strategy

ACAP	Adaptive Compute Acceleration Platform
ADAS	Advanced driver-assistance systems
ADMM	Alternating Direction Method of Multipliers
AEE	Architectural Enumeration & Evaluation
AES	Advanced Encryption Algorithm
AGPS	Assisted Global Positioning System
AMOSAS	Simulated Annealing-Based Multi-objective Optimization Algorithm
API	Application Programming Interface
APSoC	All Programmable System on Chip
ARM	Adaptive Region Method
ARMOGA	Adaptive Range Multi-objective Genetic Algorithm
ASM	Abstract State Machines
BEC	Business Email Compromise
BEV	Bird's Eye View
BLE	Bluetooth Low Energy
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAN	Vehicle data
CAV	Connected and Autonomous Vehicle
CIF	Covariance Intersection Filter
CKF	Cubature Kalman Filter
CNN	Convolutional Neural Network
COBYLA	Constrained Optimization by Linear Approximation
CONMIN	CONstrained MINimization
CPM	Camera Plane Map
CPS	Cyber-Physical System
CPSoS	Cyber-Physical System of Systems
CPU	Central Processing Unit
CSMA	Carrier-Sense Multiple Access
CU	Compose Units
DBM	Deep Boltzmann Machines
DCPS	Distributed Cyber-Physical System
DE	Differential Evolution
DES	Derandomized Evolution Strategy
DGPS	Differential Global Positioning System
DIRECT	Dividing RECTangles
DMA	Diversity Maximization Approach
DNN	Deep Neural Network
DoA	Grant Agreement No. 871738 – CPSoSaware. Annex 1 Description of the Action.

DOE	Design of Experiments
DoS	Denial of Service
DRL	Deep Reinforcement Learning
DSE	Design space exploration
DSRC	Dedicated Short-Range Communications
ECS	Environmental Control Systems
Edrx	extended discontinuous reception
EGO	Efficient Global Optimizatoin
EKF	Extended Kalman Filter
FC	Fully Connected
FCN	Fully Convolutional Network
FCS	Flight Control System
FDI	False data injection
FEM	Finite Element Analysis
fmGA	fast messy Genetic Algorithm
FPGA	Field Programmable Gate Array
FPR	False Positive Rate
FTA	Fault Tree Analysis
GA	Genetic Algorithm
GAMP	Generalized Approximated Message Passing
GLONASS	Global Navigation Satellite System
GMP	Gaussian Message Passing
GNSS	Global Navigation Satellite Systems
GPGPU	general-purpose GPU
GPS	Global Positioning System
GRG	Generalized Reduced Gradient
HDL	Highlighted Deep Learning
HWMP	Hybrid Wireless Mesh Protocol
IDS	Intrusion detection systems
ILP	Instruction Level Parallelism
ILP	instruction-level-parallelism
IMU	Inertial Measurement Unit
IoT	Internet of Things
ISA	Instruction Set Architecture
ISM	Industrial, Scientific, and Medical application
ITS	Intelligent Transportation Systems
KBE	Knowledge Based Engineering
KF	Kalman Filter
LAN	Local Area Network
LIDAR	Light Detection and Ranging

LIN	Local Interconnect Network
LOS	Line of Sight
LSGRG2	Large Scale Generalized Reduced Gradient Algorithm
MC	Memetic Computing
MGE	Multi-Gradient Explorer
MGP	Multi-Gradient Pathfinder
MISQP	Mixed-integer Sequential Quadratic Programming
MMES	Multi-Member Multi-objective Evolution Strategy
MMFD	Modified Method of Feasible Directions
MODD	Model – Optimize – Desing - Deploy
MODE	Multi-Objective Differential Evolution Algorithm
MOES	Multi-objective Evolution Strategy Algorithm
MOGA	Multi-objective Genetic Algorithm
MOGT	Multi-objective Game Theory Algorithm
MOPSO	Multi-Objective Particle Swarm Optimization
MOSA	Multi-objective Simulated Annealing
MPC	Model Predictive Control
NED	Network Description
NFV	Network Function Virtualization
NLOS	Non-Line of Sight
NLPQL	Non-linear Programming by Quadratic Lagrangian
NSGA-II	Non-dominated Sorting Genetic Algorithm
OBU	Onboard Units
ONF	Open Networking Foundation
OTA	over-the-air
PF	Problem Frames approach
PPS	Pedestrian Protection System
QoS	Quality of Service
RADAR	Radio Detection and Ranging
RBAC	role-based access control
ROI	Region of Interest
RPN	Region Proposal Network
RSS	Received Signal Strength
RTK	Real Time Kinematic
SA	Simulated Annealing
SDN	Software Defined Networking
SE	Self-adaptive Evolution
SIEM	Security Information and Event Management
SIG	Special Interest Group
SoC	System on a Chip
SOGA	Single-objective Genetic Algorithm
SPEA-II	Strength. Pareto Evolutionary Algorithm

SQP	Sequential Quadratic Programming
SR	strategic rationale
SSD	Single Shot MultiBox Detector
SUMT	Sequential Unconstrained Minimization Technique
TDOA	Time Difference of Arrival
TLU	Table Lookup
TOA	Time of Arrival
TPR	True Positive Rate
TTC	Time to Collision
V2I	vehicle-to-infrastructure
V2N	vehicle to network
V2P	vehicle to pedestrian
V2V	vehicle-to-vehicle communication
VANET	Vehicula-ad hoc-Network
VDK	Virtual Development Kit
VFE	Voxel Feature Encoding
WSN	Wireless Sensor Networks
XSS	Cross-site Scripting

## 2 Research Methodology

The research methodology used in this document is structured along with the different phases of the CPSoSaware lifecycle defined in the DoA, and depicted in Figure 1, namely: Requirements, Design, Simulation, Operation and Monitoring.

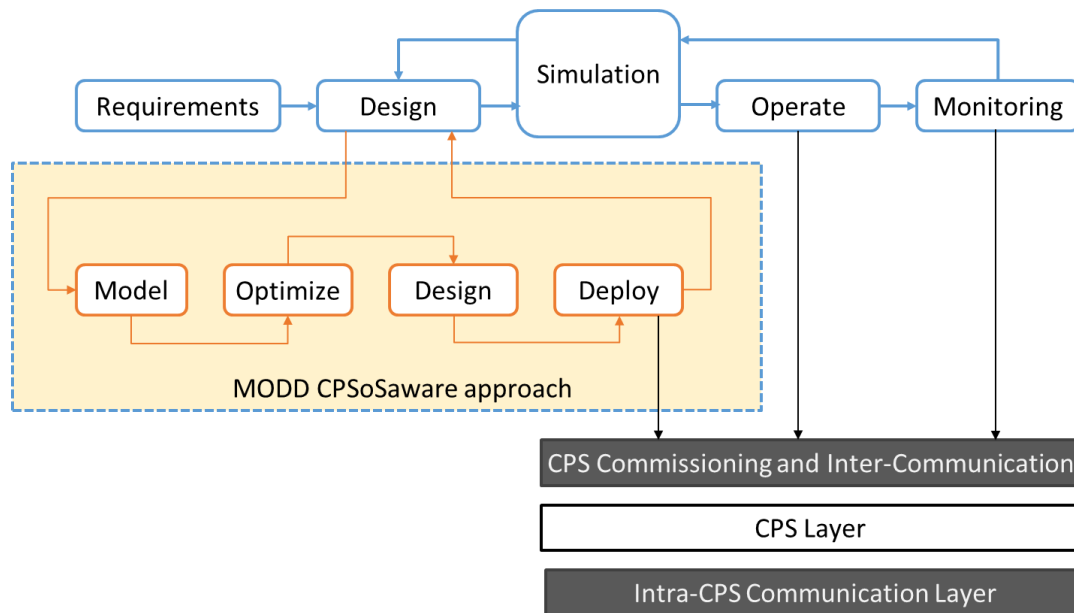


Figure 1 CPSoSaware lifecycle used in the research methodology

For the Requirements phase, Section 3 presents first the methodologies and tools commonly used in the domains of the two project use cases. These allow identifying, formalizing and documenting requirements and metrics in a way that permit their traceability and verification with the support of the CPSoSaware platform.

The Design phase is supported by the MODD concept, which stands for Model – Optimize – Design – Deploy, also represented in Figure 1 with an orange rectangle. The MODD concept unifies various modelling approaches (e.g. UML, SysML, etc.) under the same framework to capture the System level functionalities of a CPSoS and to introduce specialized components to handle cognitive behaviour at CPS level and CPSoS level. Section 4 elaborates on the most relevant techniques and tools to support modelling the CPSoS, as well as analysing and optimizing these models before deployment in the CPSs.

The Simulation phase takes the optimized models produced in the previous phase and uses simulators to further refine the models, in a context with similar characteristics to the real environment of the autonomous driving use case or the human-robot interaction use case. A feedback loop is established to re-design the models if needed, based on the experience with the simulators. Section 5 presents simulators commonly used in the domains of two use case scenarios, but also specifically used for the simulation of the system architecture (CPU and GPU simulators).

The Operation phase refers to all the aspects that support the execution of the CPSoS in the terms established by the requirement KPIs. Decentralized CPSoS processes, with dynamically changing network

topologies, while at the same time ensuring collaboration between CPSs to achieve overall CPSoS resiliency, safety and efficiency goals. Several approaches and solutions to support these requirements are presented and evaluated in Section 6.

At the Monitoring phase, the overall status of the CPSoS is assessed, using the system-wide requirement KPIs. Individual CPSs are monitored by sensors at different levels: device, network, data, system; and the collected information is aggregated and processed at the CPSoS level to take the necessary decisions, in case of deviations from the expected behaviour, for instance, a redesign of the CPSoS following the MODD approach. Section 7 surveys methodologies, techniques and tools for KPI monitoring, including specific aspects related to monitoring security properties of the CPSoS.

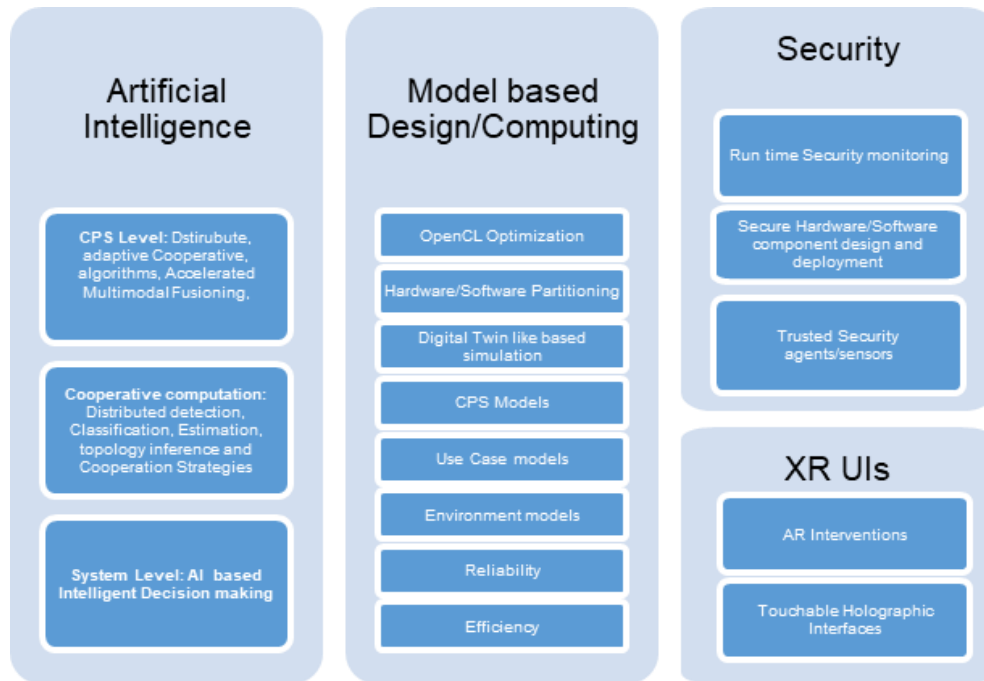


Figure 2 The CPSoS Aware Pillars

Furthermore, the CPSoS Aware Lifecycle is supported by four key innovative pillars, represented in Figure 2, which have been also reviewed as part of the state-of-the-art analysis of the different Lifecycle phases.

- **Artificial Intelligence pillar:** is considered mainly in Section 6, when evaluating individual CPSs AI solutions to support the autonomic, decentralized operation of CPSs; but also in Section 7, when reviewing monitoring techniques for anomaly detection at the system level.
- **Model-based Design/Computing pillar:** is exploited mainly at the Design phase, which relies on most advanced modelling techniques, software optimization, and hardware acceleration to implement MODD approach, as it is described in Section 4.
- **Security pillar:** relies on the application of the security-by-design paradigm at the Design phase, leveraging on existing methodologies and languages for modelling secure systems and for the analysis of security requirements, as well as security primitives and mechanisms for the protection of the CPSoS assets (Section 7). At the Operation and Monitoring phase, the monitoring of security KPIs at the system level is done through the use of sensors that observe the system at runtime, and

tools that detect attacks and threats that put at risk the overall system security and trust (Section 7). Attacks and threats can be simulated as well to determine the impact of these incidents in the overall system requirements, and this is also tackled in Section 5.

- **Extended Reality User Interfaces (XR UIs) pillar** – These aspects are considered when evaluating tools and techniques enabling the simulation of CPSs with human assistance, in Section 5; and in Section 6 to introduce human in the loop situational awareness.



### 3 Techniques and tools for definition of CPSoS requirements and KPIs

This chapter describes the techniques and methodologies for the definition of requirements of CPSoS and for the formalization of KPIs that ensure the validation and traceability of these requirements, highlighting the tools that can be used for defining the requirements of the project use cases. Everything will be adapted and customized to the world of CPSoS Aware, and Sections 3.4 and 3.5 briefly describe how these tools and methodologies will be used specifically in the two use cases of the project.

#### 3.1 Determination of techniques for the identification of use cases Requirements

##### 3.1.1 Volere Method – Manufacturing use case

The first version of the Volere Requirements Specifications template was released in 1995 and focused on a highly detailed structure that tries to integrate the widest possible spectrum of requirement categories. The Volere template covers the drivers, constraints and the dynamically arising issues of a project. In addition, based on the Volere template the system requirements are separated into two fundamental categories, functional and non-functional. The Functional Requirements describe the desired functionalities that the project should have and how they should be connected in a completely useful final product. The Non-Functional Requirements on the other hand describe the desired properties of all the components of the system such as their performance, efficiency, and usability.

The main advantage and quality that separates the Volere methodology over its alternatives is the detail in which the functional and non-functional requirements are identified. In this way, the Volere template facilitates the organization of the requirements thorough understanding with regards to the project. Besides, Volere offers a formal template for the collection of the requirements in tabular format through its “requirements shell” (also called a “snow card”). The suggested template is illustrated in Figure 3.

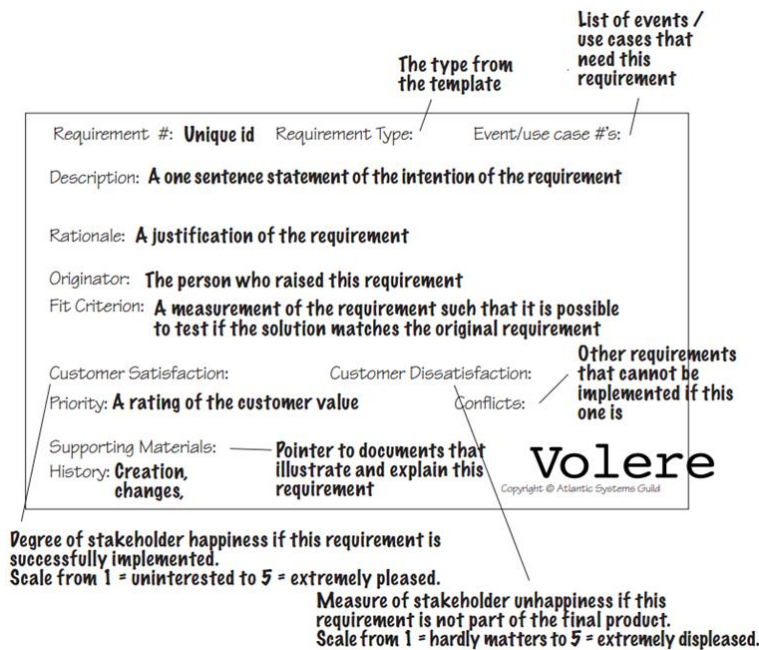


Figure 3 Volere Requirements shell as a guide to writing each requirement

### 3.1.2 Techniques and tools for definition of requirements and KPIs for the Automotive Use-Case

Nowadays, automotive systems are safety critical and their development process must comply with modern certification/safety standards such as the ISO-26262. During safety engineering analysis, performed at the early stages of the design, safety engineers establish the possible failures of the system and their origins and needs on design architecture. During the requirement development stages, a requirement phase elicits multiple requirements potentially of different types (functional, non-functional –safety, timing, hardware, performance ...), including those expressed by the safety engineers. A model-based development phase follows, characterized by *specification*, *design* and *implementation* steps which take into account the overall requirements over different disciplines (system, software, hardware). The verification and validation phase checks whether or not the developed models and the final product complies with the initial requirements. In this context, requirement modelling, traceability and analysis is a key issue in a design flow for electronic embedded systems. Safety engineering analysis is a mandatory stage in the design of critical embedded automotive systems. The derivation of safety requirements and their verification requires establishing traceability links between requirements and the different artifacts involved in the design flow. In this section, we will present the different steps of a method for expressing functional and non-functional requirements (safety, timing, hardware, performance) and ensuring their validation and their traceability over a design flow for automotive system A specific meta-model for requirements modelling and traceability is used. The methodology is illustrated on an industrial knock-control system characterized by strict safety and temporal constraints. Gotel and Finkelstein [2] defined the requirement traceability as the ability to describe and follow the life of a requirement through its development, specification, validation and verification. Industrials from IT have proposed and developed standards and engineering tools which partially cover these needs as they focus on functional requirements. Moreover, the relationship between the initial expression of requirements and their impact on solution models is not fully established. In particular, the traceability of non-functional requirements raises some others automotive specific issues dedicated to reusing and maintenance, such as the way to express and distinguish a safety or a timing requirement in a set of requirements; the ability for models to express temporal and safety properties, the link with validation tools for the test or the analysis of models and finally, the feedback of the analysis results on the design flow.

#### 3.1.2.1 Traceability and Safety Critical Systems

Despite a lot of efforts, requirement management and traceability still remain a challenging problem in the automotive industry. Automotive applications design process should comply with safety standards (ISO 26262) and customer expectations which impose vertical and horizontal bi-directional traceability of requirements:

- Vertical traceability identifies the origin of items through the work breakdown structure.
- Horizontal traceability identifies requirements relationships across workgroups or components

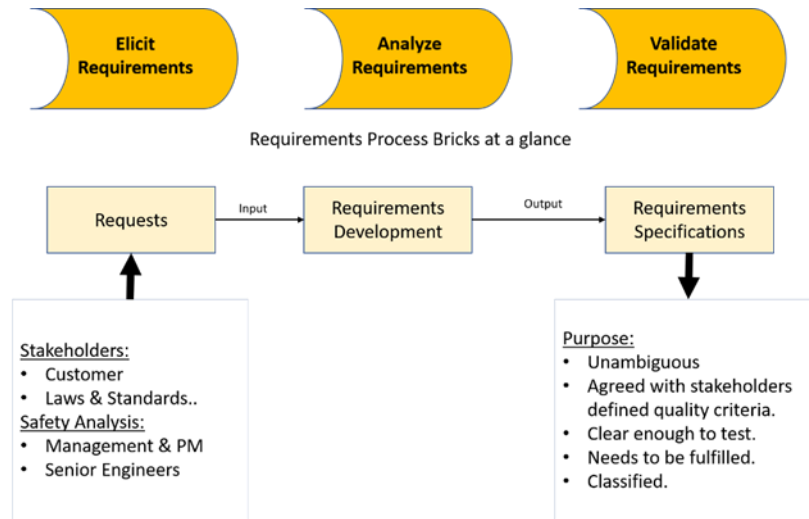


Figure 4 Overview of the requirement development.

The requirements development process, shown in Figure 4, illustrates the strong link between stakeholder's requests, safety analysis and the requirements specification document. Currently, safety requirements are often not classified and generally not formally traced to safety analysis results. In the automotive domain, safety analysis is decomposed in a Failure Mode and Effect Analysis (FMEA) and a Fault Tree Analysis (FTA). A FMEA is a procedure for analysing and classifying by severity degree the potential failure modes within a system, their effects, and their causes. A failure mode can be any errors or defects especially those that affect the customer. Effects analysis refers to FTA in which an undesired state of a system is analysed using Boolean logic to combine a series of lower-level events. The FMEA and the FTA are under the responsibility of safety engineers which express in a safety document, needs, and expectations on the system. This documentation is collected as stakeholder requests during the requirement development process and the extracted requirements are identified and elicited by the requirement engineer and then formalized in the requirement specification document. The requirement engineer can trace the requirement from the request by using inter-document traceability tools or within an integrated framework.

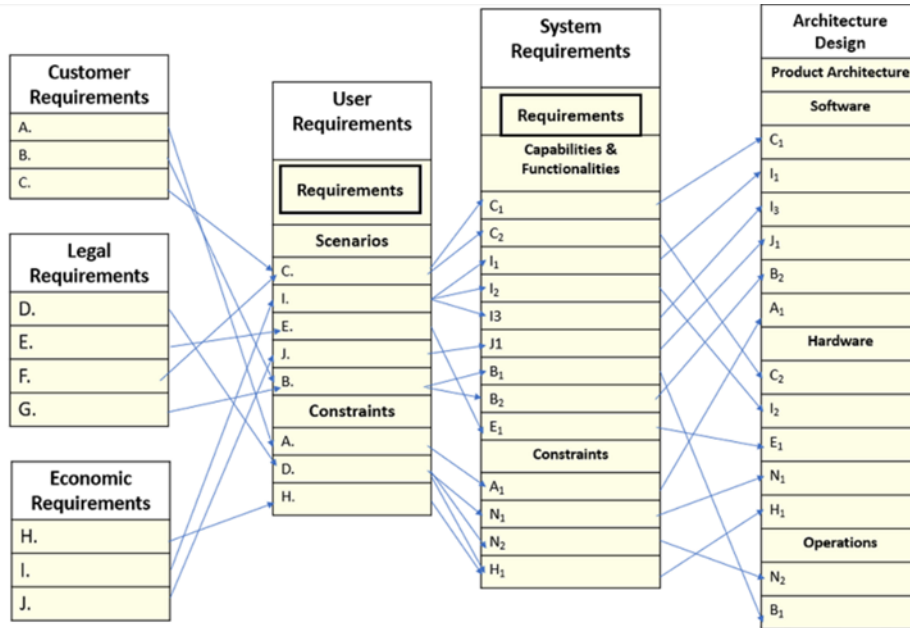


Figure 5 Traceability of requirements

Requirements derived from a safety analysis process can impact the functional and/or the hardware architecture of the system to be designed. Multiple functional and non-functional requirements (timing, consumption, hardware, dependability) derive from such analysis. These requirements are tackled at various levels (analysis, design, implementation) of a design flow that merges different tools dedicated to architecture modelling, code generation and validation and verification. Provision must therefore be made to connect initial requirements to all system components in the design flow and this, in both backward and forward directions. Indeed, the design flow includes heterogeneous models and tools. The validation and verification flow integrates dedicated tools for model and product testing, model verification and validation by temporal analysis and simulation. In this context, performing a full and bi-directional traceability of requirements and system solutions requires developing a new model for traceability that considers all the characteristics of multi-levels modelling, requirements classification, and model heterogeneity.

### 3.1.2.2 Requirements development phase

Requirements should be defined on different levels and linked. This creates a functional / requirements tree. In the automotive industry, car requirements are linked to systems, system requirements are linked to component requirements (mechanical component, hydraulic component, etc.), and in case of a SW component it is expected that more detailed SW module requirements will be derived and traced as well. The idea underlying this requirement tree is to create a structure which allows to do an impact analysis. You have an impact network and if the manufacturer sees a malfunctioning of one of the car functions they can directly address the different components which are affected and influence the fault situation.

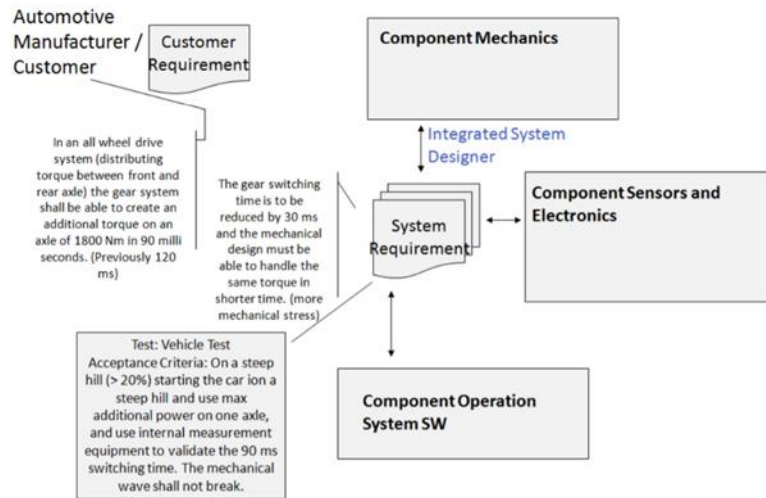


Figure 6 Example for Analysing Customer Requirements and Deriving a Linked System Specification

Figure 7 shows an example of how the system requirements impact the functionality of the mechanical, ECU, and Software subsystems leading to changes in more than one subsystem to satisfy this one system requirement. In practice this derivation is needed to provide a clear understanding of the functional decomposition and dependencies in the system. Usually in Automotive industry this is supported by a DOORS [3] or MKS RM[4] tool set which allows to store these documents, link content, and create reports about requirements coverage.

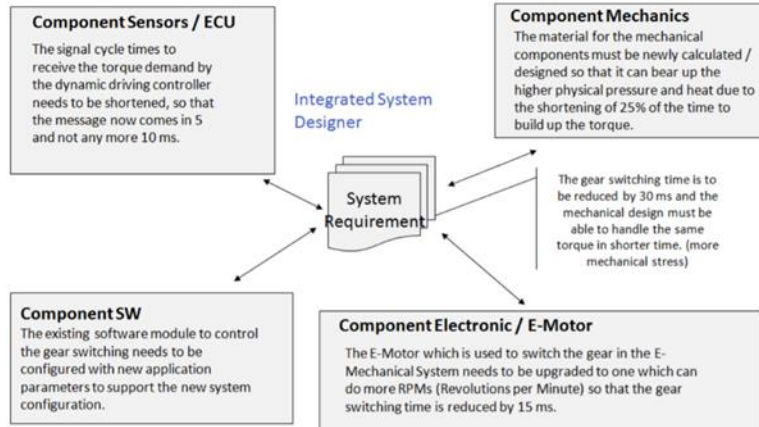


Figure 7 Example for Analysing System Requirements and Deriving Linked Software, ECU, Mechanics Requirements

### 3.2 KPI methodology (CERBERO outcomes)

One of the crucial results of the CERBERO project [1] was the definition of a complete methodology KPI based methodology for the design of cyber-physical systems [7]. IBM and USI are planning to put forward these efforts and to adapt to the needs of CPSoSaware. In this section we summarize the key achievement of the CERBERO project that will be used as a base for CPSoSaware. The fundamental points of the KPI based design methodology are the following:

- Each KPI should be defined with a metric associated to it. It is in fact of little use to have a generic KPI such as “power consumption” if it is not clearly defined how the power consumption should be calculated
- KPIs should be customized to the system that they have to evaluate. A system is measured by its specific KPIs (defined, as mentioned, together with a metric to measure them). Using the same KPIs in multiple systems is certainly not the best approach, since it would not allow to have a clear and complete picture of the specificity of the system that we are evaluating.
- Even if KPIs are specific to the systems, each KPI belongs to a family, that is identified by the way in which the KPIs are calculated (for instance, KPIs that are calculated with additions, are additive KPIs). The families, and the properties associated to them are generic and reusable.
- KPIs should follow a system from the design phase to the decommissioning. This implies that the same KPIs used at the design time, should be monitored (with appropriated monitors) during the whole live of the system
- KPI should follow a system from the design phase to the decommissioning. This implies that the same KPIs used at the design time, should be monitored (with appropriated monitors) during the whole life of the system
- KPIs should be expressed in a form that is understandable by current design tools. In fact, design tool chains are quite mature and suitable for their purpose. It would not make sense to rebuild well established and working tools to support the KPIs, it should be the opposite. Naturally, designers can choose, among the available tools, the most relevant and suitable to use in combination with the needed KPIs

### 3.3 Customization to world CPSoSAAware use cases

For the description of each specific requirement that belongs to each one of the categories listed above (3.1), a tabular template was created based mainly on the Volere requirements shell, after applying the following modifications:

- Name, this field has been added in order to provide in addition to the ID field, a short name that describes the specific requirement in human readable format.
- Constraints, which describes potential constraints / conditions for the requirement to be executed.
- Difficulty, which indicates the level of difficulty for the implementation of this requirement (estimated from a technical point of view). Difficulty ranges on a scale from 1 (=low difficulty) to 5 (=extreme difficulty).
- Actors, indicates either those persons or things that interact externally with the system or one of its components.

Removal/replacement of fields:

- Supporting materials: This field has been also removed because the majority of the documents that are related to requirements will be subjected to IPR.
- Originator (the person who raised this requirement), this field has been replaced by the Author field (the owner of each recorded requirement).
- History this field has been replaced by Revision (indicates versioning).

The final template followed for the definition of a system requirement for CPSoSAAware is presented in Table 1.

**Table 1 Template for defining a system requirement.**

<b>ID</b>	A unique identifier.
<b>Name</b>	Title of the requirement.
<b>Requirement Type</b>	Functional / Non-functional
<b>Description</b>	A requirement must be described with as much detail as possible.
<b>Rationale</b>	A justification of the requirement.
<b>Fit Criterion (Measurable)</b>	The term measurable refers to the ability to identify if the requirement has been met at the final stages of the project, and after the system has been constructed. In other words this means the tests which must be performed in order to verify whether the requirement has been addressed.
<b>User satisfaction</b>	Degree of stakeholder satisfaction depending on the successful implementation of the current requirement (Scale from 1=uninterested to 5=extremely pleased). Definition for every category of involved stakeholders (worker, production manager, system technician, researcher).
<b>User dissatisfaction</b>	Degree of stakeholder dissatisfaction if this requirement is not implemented (Scale from 1=hardly matters to 5=extremely displeased). Definition for every category of involved stakeholders (worker, production manager, system technician, researcher)
<b>Priority</b>	The requirement is ranked according to the value that distinct categories of users attach to it (worker, production manager, system technician, and researcher). (Scale from 1=low priority to 5=highest priority).
<b>Conflicts</b>	Description of any relation of the current requirement with previously described ones. Special attention to conflict with other requirements whose implementation is blocked by this one.
<b>Constraints (Attainable)</b>	An attainable requirement will usually answer the question:

	<p>“How can the requirement be accomplished?”</p> <p>Hence, here we explain any constraints / conditions for the requirement to be executed.</p>
<b>Difficulty</b>	Level of difficulty for requirement implementation (estimation). (Scale from 1=low difficulty to 5=extreme difficulty).
<b>Actors</b>	An actor is someone or something outside the system that interacts with it or with one of its components (primary actor). If the actor is interacted by the system or one of its components is a secondary actor.
<b>Author</b>	The owner of each requirement that was recorded.
<b>Revision</b>	This section lists when a version of the requirement was created.

### 3.4 Use of the methodology developed in the Autonomous Cars use case

In the automotive industry, the development and testing of human centric systems must follow the guidelines of the ISO26262. This kind of testing can be divided into two main classes:

- Vehicle-in-the-loop to test interactions between a human and the system in dangerous situations.
- Hardware-in-the-loop to test interactions between an embedded system, such as the Active Brake Control Systems, and the physics of a vehicle.

For autonomous functionality higher than level 3, as defined by the SAE, drivers will not be responsible of most driving decisions. A thorough validation of the concerned algorithms and subsystems is thus of fundamental importance in these contexts. However, as these systems will rely more and more on machine learning and probabilistic methods, conventional validation methods are not suitable and new solutions need to be adopted. Furthermore, the vehicles will eventually operate in a wide range of scenarios, including dangerous situations. As a result, a validation and verification process performed in simulations is preferable, since it allows increasing the coverage of system testing, while also reducing costs. Two main challenges can be identified in the validation of algorithms for autonomous driving. First, the complexity and variety of scenarios that the vehicles can face is larger than in Advanced Driver Assistance Systems (ADAS). Second, the necessity of considering the constant possibility of interaction between multiple systems. In this study we focus on a use-case that highlights these two difficulties: road intersection crossing. Road intersections are among the most dangerous part of road networks with more than 8% of the total road fatalities in Europe. From a perception point of view, this scenario is particularly challenging because of the limitations in the visibility field, resulting in only partially observed vehicles. In terms of decision-making, the possibility of a wrong, unexpected behaviour of other drivers makes the road intersections particularly complex to consider. The global architecture for validation and verification is illustrated in Figure 8.



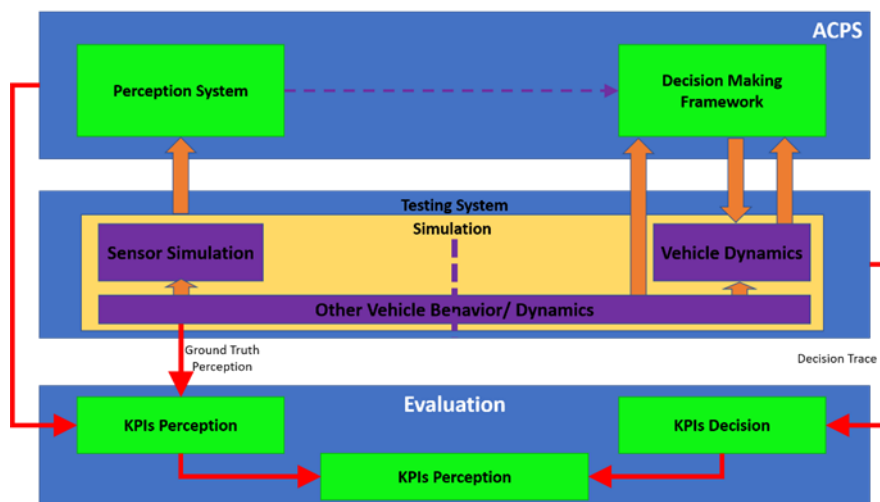


Figure 8 Interactions between the different elements of the proposed validation pipeline. Dashed lines represents future developments to connect the decision and perception

Watzenig [5] states that new validation methodologies, procedures, and laws are needed in order to successfully incorporate emerging technologies into traffic and thus improve safety and provide traffic flow optimization and enhanced mobility. Some steps towards this goal have already been taken. The EU made legal obligations on new passenger cars to include certain safety related ADAS systems (EPS, EBA) and the level of automation will increase in the following years. However, demonstrating the reliability, safety, and robustness of the technology in all conceivable situations, e.g. in all possible traffic situations under all potential road and weather conditions, has been identified as the main roadblock for product homologation, certification and thus commercialization.

The test scenarios are usually taken from collections generated by engineers, which include the complete scenario description together with the expected response of the ADAS system. However, even when utilizing these collections of tests, one cannot prove that the ADAS system will not fail in a test scenario that was not previously covered. In addition, proving ground and real-world testing is associated with high costs, low reproducibility and long validation times. Especially reproducibility in a real-world setup is challenging because of the difficulty to reach correct initialization, exact traffic behaviour, similar environmental influences, and so on. Furthermore, safety is a very important aspect and further limitations arise because some test cases could be dangerous or even impossible to be carried out by human drivers. All these limitations add up and influence the overall time needed to successfully validate an ADAS function.

The KPIs considered in the autonomous case (SAE level 3-4) can be subdivided into 3 main categories. The first two of which are directly linked to the safety requirements set by EU NCAP (European New Car Assessment Programme) and US NCAP (US New Car Assessment Programme)[6]:

- Accuracy Based:  
The KPIs falling in this category are Application Aware metrics quantifying the robustness of the scene understanding engine to detect the structural components of the scene at the correct spatio-temporal coordinates.

- Performance-Based Metrics are used to measure the runtime efficiency of the functional modules involved in the architecture and all of their sub-modules in association with the runtime of the full-architecture.
- Driver Comfort Metrics: Comprise a set of metrics quantifying the enhancement of driver comfort experience stemming from using the Autonomous functions developed in CPSoSAware. This set of metrics assesses the efficiency of the Scene Understanding engine and the HMI in delivering higher comfort of driving.

### 3.5 Use of the methodology developed in the Manufacturing use case

The current industry (besides HRC applications) is designed for a traditional approach which implies a physical separation between the machines and the operators. This forma mentis is kept and used even in many collaborative applications, where a kind of separation among the operator and the robot is maintained and it is leading to widespread use of applications with distance separation (pure SMS approach).

The co-presence of operators and robots (space sharing and collaboration between humans and robots that, up to now, was not allowed in industrial applications) deeply modifies the approach to the workspace and application design.

The identification and choice of workplaces and applications that can take proper advantage of the HRC technology are not trivial and requires the comparison with the currently used technology (based on the traditional approach) where the operator and the robot cannot share the same space and cannot cooperate on the same application.

It is important to underline that every industrial manufacturer decides whether to implement HRC according to a cost/benefit analysis to fit the initiative into its innovation plans and costs. This fact implies that:

- The evaluation is based on that specific industry's main KPIs (e.g. Automotive  $\neq$  Health  $\neq$  Avionics  $\neq$  Industrial craftsmanship  $\neq$  Naval)
- The result of the evaluation can be different according to the existing grade of development in some industries (for example in the automotive sector the ergonomics factors are already deeply considered, thus it is rare that some work cell is not optimized in terms of ergonomics)
- Similar operations can be performed manually in a fixed workstation for a long time, or in a continuous moving in a semi-autonomous operation in a short time.
- Extension and scope of the operations, as well as ergonomic characteristics and requirements, can be extremely different (e.g. Consumer electronics vs Naval Industry)
- The same operation can be performed in line (in a high or low JPH -Job Per Hours-configuration, thus resulting in very low TAKT Time, e.g. 30 s, or high TAKT Time, e.g. >10 min in line) or with mixed manual/autonomous operations in a fixed workspace (single assembly workstation, e.g. niche high target automobiles) or in fixed large workstations (e.g. Train construction, Naval, etc.)

Because of the previous considerations a unique approach to the identification of proper use-cases is not applicable. Nevertheless, HRC has a clear impact on some of the most important and common high level KPIs in industry. Independently from the type of industry, HRC impacts all the KPIs that are usually affected

by human operators; thus, HRC impacts on ergonomics/safety, productivity, inherent quality. Collaborative robots can be and are often used as “cheap, flexible robots” even in non-collaborative applications thanks to their programming simplicity; this kind of applications are not considered as reference in the current deliverable which is mainly concentrated to the description of the collaborative phases.

The reference KPIs are many; in this section we concentrate on manufacturing KPIs. The main KPIs that are related to manufacturing and production of goods (and can thus be influenced by the application of COBOTs) can be grouped into three large groups:

- **Productivity**  
All KPIs describing parameters, behaviour, performances that affect the total number of products produced and check if the effective productivity is in-line with the expected one. We even group here KPIs describing economic wastes and losses, production timing respect and so on.
- **Quality**  
Quality’s KPIs are for example those related to the amount of scraps, the number of returned parts, and the defectiveness of the final products due to Man, Machines or Manufacturing processes (Materials affect the quality as well, but it is not related to the human/robotic processes).
- **Environment, Health and Safety (EHS)**  
In relation to the manufacturing processes, other important KPIs are a factory’s energy consumption (not affected strongly by the use of robotics rather than human operators), safety and finally health; Many factors impact on operators’ health and can cause both direct (quality defects due to tiredness, stress, dis-attention), indirect (absence of operators from the production due to accidents, illness or Musculo Skeletal Disorders(MSD) and cumulative issues (social costs related to serious illnesses or MSD). To prevent many health problems, in particular MSD, the ergonomics approach is fundamental.

Each of the three groups can be split into more detailed KPIs that:

- are always dependent on the specific process
- describe in a quantifiable way the process in detail
- support the analysis of the production trend

KPIs are strongly process related, thus, it is practically impossible to list all the important KPIs upon which the application of an HRC solution should be defined.

The comprehension of the criteria upon which the HRC applications can be defined requires both the knowledge of the industrial methodologies in the specific process upon which current workplaces are based, and HRC technology both in terms of technical possibilities and state-of-the-art, and regulatory framework. Finally, there is the need to consider the operator’s involvement in conjunction with the automation planning. The operator involvement means a proper design of methods, operations and interactions with the robot as well as considerations on reachability, work repetitions, forces, body postures (ergonomics) and so on.

It is important to highlight that the set of reference KPIs can vary significantly among use cases.

This is due to the difference in the methods used in production, and in criteria used to select the work cell (that can't be unique but depend on the target and scope of the analysis). For instance, the identification of the best HRC cell in a new Assembly shop in order to reduce costs related to ergonomics is different from the identification of the best HRC cell in an existing paint shop with the aim to improve quality. For these reasons a fixed set of KPIs could be counterproductive.

The set of Parameters is also different when considering the "Brown Field" work cells (i.e. adaptation and renovation of existing work cells) or "Green Field" work cells (i.e. design of HRC work cells starting from the application without strong layout constraints).

- KPIs are identified and evaluated according to how the benefit will be evaluated/validated;
- Requested data have to answer to the following questions:
  - a) Which cell/application (identification data);
  - b) How can the application be described;
  - c) PROs: which characteristics of the application or workspace can be improved by the use of HRC;
  - d) CONs: which characteristics of the application or workspace can affect negatively the application of HRC;
  - e) Benefits: which quantifiable benefit the HRC application can give.

The most significant group of data that need to be taken into account in order to identify most suitable work cells (in brown field case) for HRC are:

- Ergonomics and related tools: description of the current ergonomics and tools which are used in production to improve it. These data are used in order to better rank the applications in which ergonomics for the operators is currently critical or achieved by expensive tools and supports. HRC can improve in these cases the current situation;
- Operator's position and room availability: description of the position of the operator, for example, in respect to the vehicle and room availability (for hands and for body) during the operations in order to consider difficulties and risks arising from the co-working in restricted areas; space in the cell/area in order to evaluate if the robot can be placed without hindering operative and safety areas for the operators;
- Conveyor description: description of the conveyor type and its use (stop station, continuous moving...). These factors can affect both the technical complexity and the difficulties for the operator in relation to the application;
- Operating time: Takt time, duration of the reference operation, NVAA (Non-Value-Added Activities) percentage. These data are originated from the MTM-UAS (Methods-Time Measurement - Universal Analysis System) analysis of the cells;
- Type of logistics: information about logistics for the specific cell in order to consider impact on the new cell's layout coming from the presence of the robot.

All above parameters are suitable for "green field" as well, but parameters like, for example, logistics, in the brown field case are a constraint, while in the green field logistics has to be designed according to the work cell and it is thus a weaker constraint.

For further design of the work cells it is necessary to define:

- A draft hypothesis of the main phases made for each potential application. The description level gets recursively more and more detailed;
- A concept design of the cells defined in terms of required functionalities and thus in terms of systems and hardware.

The following main aspects should be considered:

- Technological complexity
- HRC need and use
- Benefits/Costs indicator
- Ergonomics & Safety
- Logistic Interface

## 4 Techniques and tools for model-based design of CPSs: the MODD approach

In this chapter we review techniques and tools for the modelling of the architecture and design of CPSs. In Section 4.1 we give an overview of modelling techniques and tools. In Section 4.2 we describe the role of software profiling tools for CPSs. In Section 4.3 we describe different approaches for the evaluation of task scheduling problems. In Section 4.4, we cover modelling optimization. Finally, in Section 4.5 we describe techniques and tools for the design of CPSs.

### 4.1. Modelling techniques and tools

In this section we review existing software that can be used for modelling Cyber-Physical Systems (CPS). In Section 4.1.1 we review modelling techniques, and in section 4.1.2 we review several software modelling tools.

#### 4.1.1 Modelling Techniques

CPS models generally have both physical and computational components. There are three levels of modelling associated with these components. The first is modelling the components themselves; the second is modelling the interfaces between the components; and the third is how to model the integration.

##### 4.1.1.1 Modelling languages

<b>SysML</b>	<p>An important modelling language is Systems Modelling Language (SysML). UML-based SysML [8] has been established by the International Council of Systems Engineering (INCOSE) [9] and the Object Management Group (OMG).[10] SysML can be used for, in order of increasing rigour:</p> <ul style="list-style-type: none"><li>• Creating diagrams to describe a system</li><li>• Model simulation</li><li>• System architecture blueprint</li><li>• Executable system model</li></ul>
<b>Modelica</b>	<p>The Modelica [11] language can be used to model complex cyber-physical systems containing mechanical, electrical, electronic, hydraulic, thermal, control, electric or process-oriented subcomponents. The language itself is non-proprietary, object-oriented and based on equations.</p>
<b>MoML</b>	<p>Modelling Markup Language (MoML) [12] is an abstract, XML based language for modelling the interconnections of hierarchical components. MoML was designed for the Ptolemy II tool described below.</p>
<b>VHDL and Verilog</b>	<p>IEEE standard 1364, known as Verilog [13], is a hardware description language used to model electronic systems. VHDL [14] is a hardware description language used to model digital systems.</p>

#### 4.1.1.2 Simulation

Due to the expense and difficulty of creating complex cyber-physical systems, it is useful to be able to create simulations of both physical and computational components. This allows for quick, inexpensive verification and analysis of model designs and facilitates rapid development.

#### 4.1.1.3 Verification

An important part of modelling is the verification that the target CPS will meet all the functional and, in particular, the non-functional specifications. The components, the interfaces, and the system itself must be proven to function under a wide range of conditions and scenarios. It is highly beneficial to be able to do this as part of the modelling. Specifically, interface testing, fault testing, and stress testing should be done at the modelling phase.

#### 4.1.1.4 Analysis

Cyber-physical systems are inherently complex. The components are diverse, the interfaces are non-trivial, and the operating conditions can be severe. Analysis of the model's components, interfaces, and dynamic behaviour are critical in order to arrive at a solid design in a cost-effective manner.

### 4.1.2 Modelling Tools

There are several modelling tools available for CPSs. This includes both commercially available as well as open source tools. Most of the tools support one of the modelling languages mentioned above, and each one has a different set of features.

#### 4.1.2.1 MATLAB® and Simulink®

MATLAB and Simulink [15] are a mathematical and scientific programming environment and a modelling and simulation tool which are suitable for modelling CPS's. They are developed and sold by Mathworks [16] for mathematicians, scientists, engineers and designers for modelling the information domain of cyber-physical systems. The related products Simscape [17] and SimEvents [18] model the physics and performance behaviour respectively. MATLAB supports continuous-time, discrete-time, discrete-event, and finite state modelling.

Some examples of MATLAB and Simulink support for cyber-physical systems design include [19]:

- Computer vision and signal processing tools for designing automated situational awareness
- Concurrency modelling of the computing platform to identify architectural pitfalls, such as timing and synchronization
- Graphical state machine modelling for exploring designs and analyzing model logic for autonomous functionality
- System-level physics modelling that enables system design studies at various levels of detail and across digital and analog modalities
- Integrated communication and computation modelling to analyse channel, protocol, and operation logic interaction for the design of robust and resilient operation
- In-the-loop technologies modelled with target hardware and software to help minimize risk in bringing your design online in a physical world

- Qualification and certification toolkits, critical to deploying high-integrity cyber-physical systems in a space shared with humans

#### 4.1.2.2 *Synopsys*

Synopsys [20] has several commercial products for embedded and cyber-physical systems. For the automotive sector, Synopsys Saber [21] is used to design and verify the interaction of heterogeneous components based on differing technologies such as electrical, mechanical, and hydraulic. Saber includes the ability to create virtual prototypes and test different design variations according to a “Robust Design” methodology. This methodology encompasses the following CPS modelling techniques:

- Simulation of many component types including electronics, magnetics, hydraulics, DSPs, ECUs, D/A & A/D
- Modelling tools including components, state diagrams, multi-dimensional Table Lookup (TLU)
- Modelling languages including MAST, VHDL-AMS [22]
- Model analysis and results management including sensitivity analysis, statistical analysis, stress, fault and worst-case analyses.

For software development, Synopsys has a tool called Virtualizer [23]. The Eclipse-based Virtualizer Studio is aimed at two roles: the virtual prototype developer has tools for creating a Virtual Development Kit (VDK) which consists building blocks for creating virtual prototypes; and software developers who develop software for the virtual embedded target.

Some features of Virtualizer Studio:

- Simulation of the virtual embedded target
- Debugging on the simulator
- Integration with third-party debuggers and embedded development environments

Some examples of Synopsis Saber and Virtualizer support for modelling cyber-physical systems:

- Modelling and Simulation of Hybrid Electric Vehicle Power Systems
- Wire Harness / Electric System Design
- Simulation for system-level and hardware-level verification (ISO 26262 Road Vehicle Functional Safety Standard compliant)
- In-vehicle networking

#### 4.1.2.3 *The gem5 simulator*

The gem5 [24] simulator is “a modular discrete event driven computer system simulator platform”. It is focused on simulating computer systems. It models time as “a series of discrete events”.

The gem5 simulation infrastructure is based on earlier simulators known as the M5 [25] and GEMS [26] simulators. From the M5 simulator, gem5 inherits a highly configurable simulation framework, multiple ISAs, and diverse CPU models. GEMS add a memory system, including support for multiple cache coherence protocols and interconnect models. Currently, gem5 supports most commercial ISAs (ARM, ALPHA, MIPS, Power, SPARC, and x86 [27]), including booting Linux on three of them (ARM, ALPHA, and x86).



The project is the result of the combined efforts of many academic and industrial institutions, including AMD, ARM, HP, MIPS, Princeton, MIT, and the Universities of Michigan, Texas, and Wisconsin. Over the past ten years, M5 and GEMS have been used in hundreds of publications and have been downloaded tens of thousands of times. The high level of collaboration on the gem5 project, combined with the previous success of the component parts and a liberal BSD-like license, make gem5 a valuable full-system simulation tool. [24]

#### 4.1.2.4 Xilinx Vitis

Xilinx Vitis [28] is a system for the development of embedded software and accelerated applications on heterogeneous Xilinx platforms including FPGAs, SoCs, and Versal ACAPs. The Vitis unified software platform includes:

- Development environment
- Open-source accelerated libraries
- Plug-in domain-specific development environments enabling development directly in familiar, higher-level frameworks
- A growing ecosystem of hardware-accelerated partner libraries and pre-built applications.

The Vitis AI development environment is a specialized development environment for accelerating AI inference on Xilinx embedded platforms, Alveo accelerator cards[29], or on the FPGA-instances in the cloud. Vitis AI development environment supports the industry's leading deep learning frameworks like Tensorflow [30] and Caffe [31], and offers comprehensive APIs to prune, quantize, optimize, and compile your trained networks to achieve the highest AI inference performance for your deployed application.

The Vitis Accelerated Libraries are open-source, performance-optimized libraries that offer out-of-the-box acceleration with minimal to zero-code changes to your existing applications, written in C, C++ or Python. Leverage the domain-specific accelerated libraries as-is, modify to suit your requirements or use as algorithmic building blocks in your custom accelerators.

The Vitis Core Development Platform contains a complete set of graphical and command-line developer tools that include the Vitis compilers, analysers and debuggers to build, analyse performance bottlenecks and debug accelerated algorithms, developed in C, C++ or OpenCL. Leverage these features within your own IDEs or use the standalone Vitis IDE.

The Xilinx Runtime Library Xilinx Runtime library (XRT) facilitates communication between your application code (running on an embedded ARM or x86 Host) and the accelerators deployed on the reconfigurable portion of PCIe based Xilinx accelerator cards, MPSoC based embedded platforms or ACAPs. It includes user-space libraries and APIs, kernel drivers, board utilities, and firmware.

- The Vitis target platform defines base hardware and software architecture and application context for Xilinx platforms, including external memory interfaces, custom input/output interfaces and software runtime.
- For Xilinx accelerator cards on-premise or in the cloud, the Vitis target platform automatically configures the PCIe interfaces that connect and manage communication between your FPGA accelerators and x86 Application code – you don't need to implement any connection details!

- For Xilinx embedded devices, the Vitis target platform also includes the operating system for the processor on the platform, boot loader and drivers for platform peripherals, and root file system. You can use predefined Vitis target platforms for Xilinx evaluation boards or define your own in Vivado® Design Suite.

#### 4.1.2.5 Ptolemy II

The Ptolemy [32] Project is an ongoing project aimed at modelling, simulating, and designing concurrent, real-time, embedded systems. The focus of the Ptolemy Project is on assembling concurrent components. The principal product of the project is the Ptolemy II model-based design and simulation tool. The Ptolemy Project is conducted in the Industrial Cyber-Physical Systems Center (iCyPhy) in the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley, and is directed by Prof. Edward A. Lee.

“Ptolemy II is an open-source software framework supporting experimentation with actor-oriented design. Actors are software components that execute concurrently and communicate through messages sent via interconnected ports. A model is a hierarchical interconnection of actors. In Ptolemy II, the semantics of a model is not determined by the framework, but rather by a software component in the model called a director, which implements a model of computation.” [32]

Ptolemy II supports combinations of both state machine and continuous-time models. Ptolemy supports the MoML (XML) modelling language. There are many third-party tools, both open-source and commercial, have been developed for Ptolemy II.

Some examples of third-party tools built on Ptolemy:

- CyPhySim [33] --- An open-source Cyber-Physical Systems Simulator that provides a graphical editor, an XML file syntax for models, and an open API for programmatic construction of models.

#### 4.1.2.6 Summary of tools

Table 2 summarizes the tools and techniques being considered for this project.

**Table 2 Tools and Techniques matrix**

	License	Language	Simulation	Verification	Analysis
<b>gem5 [24]</b>	Open source	no	computers	no	no
<b>ptolemy [32]</b>	Open source	MoML	yes	yes	yes
<b>simulink [15]</b>	Commercial	VHDL, Verilog	yes	yes	yes

<b>synopsys [20]</b>	Commercial	MAST (proprietary) and VHDL	yes	yes	yes
<b>xylinx [28]</b>	Commercial	OpenCL	computers	yes	yes

## 4.2 Software Profiling Tools

The CPSoSaware computational nodes will be based on a state-of-the-art FPGA platform. The FPGA fabric will mainly undertake the most intensive computational tasks i.e., the multimedia and AI workloads. The FPGA platform will be a Xilinx FPGA equipped with a Zynq all programmable System on Chip (APSoC) [34]. The Zynq FPGA includes an ARM Cortex A9 multicore processor (equipped with a Neon co-processor) interfaced with programmable logic allowing high flexibility and performance. In other words, and as it will be further analysed below, the FPGA platform combines a processing element that executes algorithms at a software environment (the ARM processor) and a processing element that can efficiently accelerate demanding tasks in reconfigurable logic (the FPGA itself). In this way, the high demanding tasks (e.g., multimedia data streams) can be continuously monitored by the processing elements of the FPGA platform. Depending on the specific combination of algorithms that get triggered, some or all computational tasks may be executed in the processor (ARM cores) or accelerated with fixed logic or reconfigurable hardware components inserted in the FPGA reprogrammable logic. Most importantly, typically the multimedia workloads must be processed in real-time further increasing the computational burden posed to FPGA platform.

To be able to deal with the above requirements, we must take advantage of all processing capabilities of the Zynq-based FPGA platform. In particular, our view is not to consider the Zynq platform as a typical FPGA acceleration mean, but as an effective heterogeneous multicore platform. Under this direction, three different types of processing nodes are co-allocated in a Zynq platform. Those are the multicore core ARM processors, the Neon co-processor, and the FPGA programmable logic itself. Operating this apparent heterogeneous system in tandem formulates, inter alia, a mapping-allocation problem. As a first step, we must extract specific and dedicated characteristics of each workload and second to dispatch each workload (or part of a workload) to the appropriate computational node for processing.

Of course, the problem is more complex since the programmable FPGA logic is a “morphable” computation resource without predefined computational capabilities. In any case, the first step in managing efficiently the underlying heterogeneous system is to formulate a toolchain of profiling, visualization and software analysis tools. The target will be to extract specific characteristics from the executing applications as a whole (even from the kernels and/or program phases of the applications) that will allow us to i) make safe mapping-resource allocation decisions and ii) guide to the extent possible the upcoming FPGA implementation phase. Interestingly, there is a plethora of available profiling tools both open-source and proprietary. After carefully reviewing the characteristics of the majority of those tools, fortunately, we end up with a set of open-source tools that exhibit all the required aspects. Of course, there are intrinsic drawbacks and limitations in each tool, e.g., if a tool is based on instrumentation or sampling.

As a result, a group of profiling tools is needed and the process of selecting what tool must be used to extract a particular application characteristic (e.g., computational vs. memory bound phases of an application) is challenging. Based on our analysis, the combination of valgrind [35], oprofile [36], and

vampire [37] offer all the necessary characteristics to assist us through the envisioned direction (the latter tool will be used for visualization purposes). The combination of those tools allows effectively extracting the ILP (instruction-level-parallelism) properties, the instruction dependencies, cache and memory requirements both at function and instruction level of target applications and most importantly in a nice graphical representation. Exercising and operating those tools in an integrated manner will be one of our activities in the CPSoS Aware project.

### 4.3 Evaluation of task scheduling problems

Task scheduling in CPS is complex. There is a wide range of tasks of different scales that need to be scheduled for heterogeneous hardware platforms with different sets of non-functional requirements.

#### 4.3.1 Scheduling of preventive maintenance in a CPSoS

Preventive maintenance is a time-based or interval-based planned service to detect and prevent potential failures and extend the life of the equipment. It is scheduled maintenance of a plant and its equipment that is designed to improve equipment life and avoid any unplanned maintenance activity. Preventive maintenance scheduling is used:

- To minimize the number of failures of critical equipment.
- To reduce the loss of production from equipment failures.
- To acquire meaningful data from the equipment history so we can make more intelligent decisions on repair, overhaul, and replacement to maximize the return on capital employed.
- To provide tasks for planning and scheduling for minimal production disruption.
- To promote better safety, health, and environmental conditions for our workforce.
- To reduce overtime costs and provide more economical use of maintenance mechanics due to working on a scheduled basis instead of an emergency basis to repair breakdowns.
- To use timely, routine repairs to bring about fewer large-scale repairs.
- To reduce product rejects, rework, and scrap through better overall equipment condition.
- To identify equipment with high maintenance costs, indicating the need for corrective maintenance, operator training, or replacement of obsolete equipment.
- To better care for assets and increase the life span of assets, thereby eliminating premature replacement of machinery and equipment.
- To increase the life span of the equipment.

The key to a successful preventive maintenance program is scheduling and execution. Scheduling should be automated to the maximum extent possible, which may mean having to update your systems with meter information. Priority should be given to preventive maintenance and a very aggressive program to monitor work and ensure it is completed according to a schedule that should be in place. The frequency of inspections is determined by the type of equipment, its age, its condition, and the consequences of failure [38].

#### 4.3.2 Optimization of task scheduling of under fatigue [39]

CPSoS and other manufacturing systems involve interdependent machines and components. The maintenance of these complex systems plays a critical role in their efficient usage in terms of cost, availability, and safety [40]. Mechanical components may deteriorate under cyclic loadings. Cracks or

defects may initiate and propagate through the structure, leading to eventual structural failure of the component, or a loss of serviceability. The fatigue life is characterized by three different stages: fatigue crack initiation, stable crack growth, and unstable crack growth. Similarly, human fatigue during long distance driving or boredom in autonomous vehicles progresses through several stages and depending on patterns of work and the driver's tolerance to fatigue.

The effects of uncertainties cannot be neglected for the scheduling of maintenance activities. Uncertainties are considered in the non-destructive inspection, as well as in the crack initiation and growth processes. Hence, the costs associated with repair and fracture are not fixed, but they are influenced by the uncertain parameters. In the case of professional drivers, uncertainties due to irregularities in rest and duty hours are not covered by current regulations that place limits on consecutive hours of work and rest irrespective of time of day without taking into account the contribution of human circadian rhythms to alertness or of sleep physiology [41].

The optimum of a function, including uncertainties, can be found in the framework of reliability-based optimization. Several definitions of reliability-based optimization have been proposed in the literature. The outcomes from reliability analysis can be considered in the performance function, or the constraints, or both. Herein, a function whose expression includes a linear combination of outcomes from reliability analysis is minimized. The problem of reliability-based optimization is formally stated as:

$$\min_{x=(q,t_I)^T} C_T(x) \quad \text{subject to } h_i(x) \leq 0, \quad i = 1, \dots, N_c$$

where  $C_T$  denotes the total lifetime costs of the structure, which have to be minimized,  $h_i(x)$  denote the constraint functions, which are fulfilled as long as their value is less than (or equal to) zero and  $N_c$  is the total number of constraints. The time of inspection  $t_I$  and quality of inspection  $q$  are introduced as the design variables of the optimization procedure (i.e. the objective of a study like this is finding the values of these parameters leading to minimized total costs).

In the case of manufacturing, the total costs can be expressed as the summation of the costs of inspection, repair and failure:

$$C_T(x) = C_I(x) + C_R(x) + C_F(x)$$

where  $C_I$ ,  $C_R$  and  $C_F$  denote the cost functions associated with inspection, repair and failure respectively. No additional information about the relative costs is considered and it is assumed that there is a linear relation between the costs associated with the uncertain events (fracture, repair) and their respective probability of occurrence. Similarly, the costs associated with inspection are assumed to be proportional to the parameter  $q$ . The costs associated with inspection are assumed to be proportional to the quality of inspection:

$$C_I(x) = C_i q$$

where  $C_i$  is a coefficient weighting the contribution of the inspection to the total costs. The costs associated with repair and failure are expressed as:

$$C_R(x) = C_r p_R(x)$$

$$C_F(x) = C_r p_R(x) + C_f p_F(x)$$

where  $p_R$  and  $p_F$  are the probability of repair and the probability of fracture during the service life respectively,  $C_r$  and  $C_f$  are coefficients weighting the contribution of the repair and of the failure of the structure within the total costs respectively.

In the case of autonomous freeway driving, task scheduling problems involve from more simple adaptive cruise control, to lane-keeping, intelligent route planning, off-road navigation, to more challenging interaction problems between autonomous vehicles and human-driven vehicles, pedestrians or infrastructure.

For autonomous freeway driving, the main modules to be considered include the distance keeper controlling the distance from the leading vehicle, lane selector which outputs the intended lane, i.e., the lane the autonomous vehicle wants to merge into, and the merge planner which is triggered whenever the intended lane is different from the current lane and performs an adjustment on the vehicle's position and speed to find the best merging opportunity. A prediction engine is commonly used [42] to simplify the modelling complexity for producing agile reactions to complicated and hard-to predict traffic, and to emulate human-like prediction ability, which helps select better control strategies. After predicting a number of scenario sequences corresponding to these strategies, a cost can be computed for each strategy  $C_{str}$ :

$$C_{str} = \sum_{t=0}^T C_{scenario(t)}$$

As the surrounding vehicles' actual behaviours might be different from the prediction engine estimate, constraints on the safety conditions should be imposed for the selection of the optimal strategy.

There are different types of costs to evaluate the strategies, i.e. the progress cost, the comfort cost and the safety cost [42]. The progress cost represents how well a strategy does in finishing a given task by penalizing those strategies which take longer to finish the task. The comfort cost is built to represent the experienced human drivers' behaviour, who for example avoid large accelerations for greater comfort while driving. The safety cost of a scenario consists of the clear distance cost and the braking distance cost.

### 4.3.3 CPSoS scheduling framework for quality prediction and manufacturing control

When a system fails, then a repair or restoration process must be followed, as presented below [44]:

- When a failure or a defect has been detected, then it must be confirmed, otherwise it could result in a waste of time at a high cost.
- Next, the failure report is completed when the system or the component is prepared for the maintenance.
- The next step in corrective maintenance is the localization and isolation of a failed system or component in the assembly.
- The failed system or component is removed for disposal or repair.
- If it is repaired, it is checked before being put back to use.

A schedule depends upon other influences that cannot be controlled by the scheduling system, like the sudden failure of the CPSoS component. Therefore, the final schedule is only available after the execution of all jobs. Next, the scheduling system is divided into three parts: a scheduling policy, an objective function and a scheduling algorithm.

## 4.4 Model optimization

In this section we review existing optimization software that can be used for design architecture optimization in early development stages of Model Based Systems Engineering. In Section 4.4.1 we review commercial software enabling black box input/output data flow integration between different models/tools including simulation models or proprietary code if needed. ModeFrontier, modelCenter and Isight take most of the market share. In Section 4.4.2 we review A&D domain modelling and analysis software developed by Pacelab. In Section 4.4.3 we review some optimization packages that, by their claim, could be used for architecture optimization. In Section 4.4.4 we discuss how these can be used for architecture optimization.

### 4.4.1 Commercial Software

In this section we review commercial software that enables black box input/output data flow integration between different models/tools. There are several well-known products and we review them here.

#### 4.4.1.1 ModeFRONTIER

ModeFRONTIER [45] is a multi-objective optimization and design environment, written to couple computer aided design (CAD)/computer aided engineering (CAE) tools, finite element structural analysis and computational fluid dynamics (CFD) software. It is developed by ESTECO SpA and provides an environment for product engineers and designers. modeFRONTIER is a GUI driven software written in Java that wraps around the CAE tool, performing the optimization by modifying the value assigned to the input variables, and analysing the outputs as they can be defined as objectives and/or constraints of the design problem.

Key features:

- Process integration: The logic of the optimization loop can be set up in a graphical way, building up a "workflow" structure by means of interconnected nodes. Serial and parallel connections and conditional switches are available. modeFRONTIER builds automatic chains and steers many different external application programs using scripting and direct integrations nodes. This allows building model from various components, where each component is a "black-box" exposed input and output parameters. Data flows from component to component where each component can be exposed by different tool. Since output parameters of one component typically are input parameters of another component, corresponding tools running sequentially, but parallel tool running is also available where it is suitable. The model can be used for simulation and optimization purpose.
- Pre-processing: modeFRONTIER includes design of experiments (DOE) tools, that can be combined and blended to build up the most efficient strategy to solve complex multi-disciplinary problems. DOE includes various statistical techniques such as random generator sequences, Factorial DOEs [46], Orthogonal and Iterative Techniques, as like as D-Optimal or Cross Validation [47]. DOE techniques can be used to extract as much information as possible from a limited number of simulation runs or to provide the initial data points for optimization algorithms.

- Multidiscipline design optimization: Optimization techniques can be characterized as multi-objective simulation-based optimization over set of "black-boxes". Various heuristic algorithms can be used for optimization purpose. These include following families of algorithms:
  - *Genetic Algorithms* represented by improved Multi-objective Genetic Algorithm (MOGA) [48], Non-dominated Sorting Genetic Algorithm (NSGA-II) [49], Adaptive Range Multi-objective Genetic Algorithm (ARMOGA) [50] and Multi-Objective Particle Swarm Optimization (MOPSO) [51]. Genetic Algorithms use the analogy of natural evolution of species: natural selection and reproduction guide the evolution towards individuals that are better adapted to the environment. MOGA uses a smart multi-search elitism for robustness and directional crossover for fast convergence. The efficiency of MOGA is ruled by its reproduction operators: classical crossover, directional crossover, mutation, and selection. The main features of NSGA-II are a parameter-less diversity preservation mechanism and the capability of dealing directly with continuous variables. ARMOGA can adjust the search region according to the statistics of the former data and help to reduce the number of function calls. MOPSO is a meta-heuristic algorithm which takes inspiration by the simulation of social behaviour of bird flocking and fish schooling. This algorithm allows both continuous and discrete variables, the constraint handling method does not make use of penalty parameters; a clustering method is used to prune non-dominated set and to maintain diversity between the solutions.
  - *Simulated Annealing* methods work on the basis of a thermo-dynamical analogy: the slow cooling of a heated system let it settle down in a configuration of minimum energy. Out of the metaphor it means that the optimum is reached. These are represented by Multi-objective Simulated Annealing (MOSA) [52]. It works with the concept of a population of points moving towards the set of solutions. The thrust is given by thermal perturbation, driven by a genuine multi-objective search. The hot phase and cold one balance between robustness and convergence.
  - *Evolution Strategies* are optimization techniques based on the ideas of adaptation and evolution. In this sense they are similar to Genetic Algorithms, but here the main search procedure is a smart mutation operator. These are represented by single-objective Evolution Strategy (1P1-ES) [53], Derandomized Evolution Strategy (DES) [54] and Multi-member Multi-objective Evolution Strategy (MMES) [55]. 1P1-ES performing robust local optimization with accelerated step-size control. It is a very fast evolutionary algorithm, and it is robust even with slightly noisy functions and in presence of discontinuities. DES is a robust local search procedure that works with a fast deterministic (derandomized) adaptation of the multivariate mutation distribution. It is a fast single-objective optimizer in continuous search space. MMES is a global optimization method. It supports multi-objective Pareto optimization, and can deal with discrete and continuous variables.
  - Other algorithms include Simplex (Nelder–Mead) heuristic [56], Multi-objective Game Theory Algorithm (MOGT) [58]. The Nelder–Mead method is a commonly used nonlinear optimization technique, which is a well-defined numerical method for twice differentiable problems. The method approximates a local optimum of a problem when the objective function varies smoothly and is unimodal. However, the Nelder–Mead technique is a heuristic search method that can converge to non-stationary points on problems that can be solved by alternative methods. The MOGT algorithm implemented in modeFRONTIER follows different steps: first, the variable space is initially decomposed randomly, and then each player, starting from a common original design, launches a mono-objective optimization algorithm (Simplex heuristic), to improve the objective assigned to it. After a



certain number of Simplex iterations, each player finds the best configuration (and set of variables) for its objective, and then the search continues with a new step, for which each player starts a new Simplex sharing the optimal variables found by the other players.

After a candidate solution has been found it can be refined by using sequential quadratic programming, gradient-based or trusted region methods. Optimization heuristic techniques are able to manage continuous, discrete and mixed variable problems.

- Post-processing: Post-optimal analysis available via so-called robust design tool to verify the system's sensitivity to manufacturing tolerances or small changes in operating conditions. The software offers wide-ranging toolbox, allowing the user to perform sophisticated statistical analysis and data visualization. This set of tools enables the user to explore, filter and rank the set of optimal solutions of a multi-objective problem (the so-called Pareto frontier), to perform sensitivity analyses, robustness verifications and also to produce standard and customizable reports of the optimization project.

Usage in Architecture Optimization:

ModeFRONTIER can be used for optimizing parts of complex systems or optimizing a whole system over a small set of predefined architectures. Many complex systems cannot be optimized by this software for the following reasons:

1. Simulation-based techniques typically require long running times in order to support multiple possible topologies or parts. In fact, for each possible part and for each possible topology there is a need to run a new simulation. Statistical methods such as those found in this software are not able to reduce the number of possible solutions because of combinatorial nature of problem. This can lead to unrealistically long running-times even on very powerful hardware.
2. Using heuristic algorithms effects the quality of the obtained solution. In case of a large number of possible candidate solutions this can lead to a trade-off between run-time and quality. Reducing the runtime (which is also affected by simulation nature of software) can result in a solution which is very far from optimum.
3. This software is not suitable for creating an optimal system topology. It is possible to manually define a set of optional system topologies and run the software sequentially over them all in order to obtain best one. This can cause several types of problems (for example, optimal system topology might not be included in set of testing topologies). This approach can be further augmented by automatic generation of optional system topologies, but that would require a special program to create the alternatives, and a set of topological rules (in an arbitrary suitable language) that will serve as input to the program. Such generation is not a simple task by itself.
4. Constraints are checked sequentially. Could be hard to find a feasible solution. The first feasible solution could be far from optimum.
5. Usage of "white-box" parts of the model is not possible and therefore there is no usage of problem structure. Standard MILP and MIQCP solvers not used, and its usage are not possible under these settings.

#### **4.4.1.2 ModelCenter**

ModelCenter [79], developed by Phoenix Integration, Inc., is a software package that aids the design and optimization of systems. It enables users to conduct trade studies, as well as optimize designs. ModelCenter

is used in a variety of applications, primarily system design and optimization in the aerospace and defense industry. It is also used for process design and optimization in the manufacturing industry.

Key features:

- Process integration: Users can utilize a wide variety of methods and tools to encapsulate or “wrap” a software application so that it can be re-used and integrated with other tools. Applications can be wrapped in a number of ways including: File I/O, Scripting or Plug-Ins. Almost any software application can be wrapped: user-generated tools, legacy engineering codes, CAD and CAE tools. ModelCenter allows users to graphically create simulation workflows by dragging and dropping wrapped applications from a simulation library and combining them using if-then branches, loops, and other flowchart-like constructs. As in the modeFRONTIER data flows from component to component where each component can be exposed by another tool.
- Multidiscipline design optimization: As in the modeFrontier, optimization techniques can be characterized as multi-objective simulation-based optimization over set of "black-boxes". A wide range of heuristic algorithms are available. Built-in optimization algorithms and libraries includes:
  - Nelder-Mead simplex heuristic [56], Hooke-Jeeves direct search algorithm [57]
  - SwarmOps heuristic optimization library [59] which includes Differential Evolution method, Differential Evolution method with dithered parameters, Self-adaptive Differential Evolution method, Local Unimodal Sampling method, Many Optimizing Liaisons method, Pattern Search method, Particle Swarm Optimization method and Random sampling method.
  - DOT [60] general-purpose gradient-based optimization software library which includes Broydon-Fletcher-Goldfarb-Shanno (BFGS) variable metric method [61], Fletcher-Reeves (F.R.) conjugate gradient method [62], Modified Method of Feasible Directions (MMFD) [63], Sequential Linear Programming (SLP) [64] and Sequential Quadratic Programming (SQP) [65].
  - BIGDOT optimization library [66] intended to solve very large scale optimization problems includes Sequential Unconstrained Minimization Technique (SUMT) [67].
  - The DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit [68] provides following optimization algorithms and methods: Asynchronous Parallel Pattern Search [69], Coliny Constrained Optimization by Linear Approximation (COBYLA, Coliny DIRECT, Coliny Evolutionary Algorithm, Coliny Pattern Search, Coliny Solis-Wets ) [70], CONMIN methods [72], Multi-objective Genetic Algorithm (MOGA) [48], North Carolina State University (NCSU) DIRECT [73], OPT++ Polak-Ribiere conjugant gradient, OPT++ Finite differences Newton,, OPT++ Full Newton, OPT++ Parallel direct search, OPT++ Quasi Newton [74], Single-objective Genetic Algorithm (SOGA) [75].
  - Boeing Design Explorer and Boeing SQP Gradient Optimizer [76].
  - Darwin optimization framework [77] which is based on fast messy Genetic Algorithm (fmGA) solver [78].
  - Non-dominated Sorting Genetic Algorithm (NSGA-II) [49]

Additional optimization algorithms and libraries can be added through plug- in architecture.

- Post-processing. A collection of multi-dimensional visualization plots available to graphically interpret results.

Usage in Architecture Optimization:

ModelCenter [80] is simulation-oriented software. Usage of ModelCenter in architecture optimization is limited for the same reasons as usage of modeFrontier for this purpose. Absence of powerful pre-processing techniques additionally limits the usage of modelCenter in architecture optimization.

#### 4.4.1.3 Isight

Isight [81] is an open desktop solution for creating flexible simulation process flows, consisting of a variety of applications, to automate the exploration of design alternatives, identify optimal performance parameters and integrate added-value systems. Isight is a product of Dassault Systèmes Simulia Corp. that is known as engineering simulation software (CAE) vendor.

Key features:

- Process integration: Isight includes a number of valuable application components that are included with all package levels. These components are most commonly used for building simulation process flows and exchanging data with external sources. Additional application components are available as add-on packages or can be created in component development environment based on Eclipse. The platform provides drag-and-drop process flow creation, parameter mapping, and problem formulation. This feature-rich process editor supports powerful file parameters that can represent simulation models as variables, as well as dynamically sizable arrays for both inputs and outputs. It also provides branching, looping, conditional statements of arbitrary complexity with any parameter, and other execution logic. As in the modeFrontier and ModelCenter, data flows from component to component where each component can be exposed by another application.
- Pre-processing: Isight includes a full suite of DOE methods including Central Composite [82] and Data File [83], Full Factorial and Fractional-Factorial [46], Box-Behnken [84], Latin Hypercube and Optimal Latin Hypercube [85], Orthogonal Array [86], Dependent Variable Sampling and Attribute Sampling [87]. Custom DOE techniques can be created using component development environment.
- Multidiscipline design optimization: As in the modeFrontier and ModelCenter, optimization techniques can be characterized as multi-objective simulation-based optimization over set of "black-boxes". A wide range of heuristic algorithms are available. Built-in optimization algorithms include:
  - *Pattern search methods*: Nelder-Mead simplex heuristic [56], Hooke-Jeeves direct search algorithm [57], Adaptive Simulated Annealing [88].
  - *Genetic Algorithms*: Multi-Island GA, ARMOGA, NSGA-II [49], Neighborhood Cultivating GA (NCGA), Particle Swarm [89].
  - Evolution algorithm [90].
  - Mixed-integer sequential quadratic programming (MISQP) [91].
  - *Gradient-based algorithms*: Non-linear Programming by Quadratic Lagrangian (NLPQL) [92], Modified Method of Feasible Directions (MMFD) [93], Large Scale Generalized Reduced Gradient Algorithm (LSGRG2) [94].
  - *Other algorithms*: Stress-Ratio Method [95], Multi-objective Approximation Loop [96].

Additional optimization algorithms can be added through component development environment.

- Post-processing. The user interface supports the creation of visual tools for real-time post-processing of data such as tables, 2D and 3D plots, correlation maps, self-organizing maps, and

statistical analysis. Run data can be filtered and graded with a flexible set of criteria. All scatter plots allow easy one-click visualization of the virtual prototype by dedicated simulation results viewers.

Usage in Architecture Optimization:

Isight is also simulation-oriented software. Usage of Isight in architecture optimization is limited for the same reasons as usage of modeFrontier for this purpose.

#### 4.4.1.4 *OptiY*

OptiY [97] is an open and multidisciplinary design environment providing most modern optimization strategies and state of the art probabilistic algorithms for uncertainty, reliability, robustness, sensitivity analysis, fatigue life prediction, data-mining and meta-modelling. The simulation model can be considered as a black box with inputs and outputs. Within, it is an open platform for different kind of model classes. The adaptation to a special simulation environment takes place by a suitable interface. Collaborating different simulation systems is possible as networks, finite-element-method, rigid body dynamics, also material test bench as control optimization for drives.

Key features:

- Process integration. OptiY is an open and multidisciplinary design environment, which provides generic and direct interfaces to many commercial CAD/CAE-systems or in-house codes. The integration of any system into an arbitrary process chain is very easy with the graphical workflow editor. Collaborating different simulation model classes is possible as networks, finite-element-method, multi-body-system and material test bench as control optimization for drives etc. The multidisciplinary simulation and optimization process is presented as a graphical workflow by the Workflow-Editor. Predefined nodes can be inserted and edited using drag & drop with the mouse. In a project, a customized multi-experiment is possible using the Script-Editor. The parameters and results of different experiments can be coupled and exchanged using its COM-interface. Within, conditional loops of experiments are programmable.
- Pre-Processing. OptiY provide various DOE techniques including: Full Factorial Design [46], Center Composite Design [82] Monte-Carlo-Sampling [98], Latin-Hypercube-Sampling [85], Sobol-Sampling [98], Response Surface and Adaptive Gaussian Process [99], First Order Moment Method, Second Order Moment Method and Subset Simulation [100]. User-defined DOE also available.
- Optimization. OptiY provides simulation-based optimization. Single and multi-objective optimization heuristics can be used. Optimization techniques support continuous, discrete and binary decision variables.
- Post-processing. A mathematical decision-making support tool is available. The choice of a best suitable solution is done automatically by input of an ideal and normalized design point by user.

Usage in Architecture Optimization:

Usage of OptiY in architecture optimization is limited for the same reasons as usage of modelCenter for this purpose.

#### 4.4.1.5 Nexus

Nexus [101] is process integration and optimization software designed to solve multi-disciplinary and multi-objective optimization problems via a flowchart representation validated on the fly. Nexus is developed by iChrome Ltd., a British engineering and software company that specialize mainly in mathematical optimization and finite element structural analysis.

Key features:

- Process integration. The kernel of Nexus is designed to provide a scalable framework, with the possibility to link dynamic modules at runtime. Commonly used modules are packed within standard libraries and provided as part of the Nexus standard release. Additional libraries can be added to extend the software functionalities. The Nexus flowchart module offers the user a graphical representation of the design process and of the optimization procedures. Different evaluation nodes can be used to link the software with external procedures and executables to define multi-disciplinary procedures. Expressions, Conditional Expressions, C/C++ external user code, Java external user code, Matlab [16], Python, Microsoft Excel, Fluent [102], Abaqus [103], Nastran [104], CATIA, Radioss [105] and other evaluation nodes are available [106]. External databases can be used to store performed evaluations and to get already performed results across concurrent or subsequent runs. The latest release supports SQLite, Firebird, PostgreSQL and Microsoft Access databases.
- Pre-Processing. Within Response Surfaces module it is possible to create static and dynamic Design of experiments, i.e. a set of support points for a response surface or surrogate model. The set of points can be imported from external files and databases or generated on the fly with one of the supported allocation algorithms. Response Surfaces can be created directly in the optimization flowchart or can be built independently using the Metamodel module. In the latter case a new response surface can be built using existing tables as DOE and following a graphical wizard. Surfaces created with this tool can be exported and later imported back in a new flowchart to be used within the flowchart module. The Response Surface Library provides following DOE methods: Random point allocation, Full factorial allocation [46], Latin and optimal Latin Hypercubes [85], Latin square and optimal Latin Square [107], Cubic Face Centred, Box-Behnken [84] and Plackett-Burman designs [108], Taguchi orthogonal matrix design [86], D-optimal design [47].
- Optimization. Nexus is released with 3 main optimization libraries: Standard Library, Gradient-based Optimization Library and Genetic Algorithm Library. They include the following algorithms: Simplex (Nelder-Mead) heuristic, Bounded BFGS optimization [61], Levenberg-Marquardt least-square minimization method [109], Sequential linear programming, SQP (nlConst) optimization, Generalized Moving Asymptotes method, Adaptive Simulated Annealing [88], Pattern Search Algorithm (Mesh Adaptive Direct Search), Multi Criteria Decision Making procedure, Feasibility Region Search Algorithm, Single and multi-objective Genetic Algorithms, Single and multi-objective Particle Swarm Optimization [110]. As other related software of this type Nexus provides simulation-based optimization over set of black-boxes.
- Post-processing. The Visualization and plotting module allows the user to create two or three-dimensional plots and charts to monitor the optimization procedure and to display results.

Usage in Architecture Optimization:

Usage of Nexus in architecture optimization is limited for the same reasons as usage of other simulation-based optimization software for this purpose.

#### 4.4.1.6 *Kimeme*

Kimeme [114] is an open platform for multi-objective optimization and multidisciplinary design optimization. It is intended to be coupled with external numerical software such as CAD, Finite Element Analysis (FEM), Structural analysis and Computational Fluid Dynamics tools. It has been developed by Cyber Dyne Srl and provides both a design environment for problem definition and analysis and a software network infrastructure to distribute the computational load.

Key features:

- Process integration. The problem definition workflow is based on the data flow paradigm. Several icons can be interconnected in order to describe the flow of data from the design variables to the desired objectives and constraints. Input/output nodes can be used to calculate any part of the objective computation, using internal or external numerical procedures. Any of these procedures can be distributed over the LAN, exploiting all the available computational resources. Integration of external software available via DOS/Bash scripts or by importing Matlab or Java code directly to Kimeme.
- Pre-Processing. Different DOE strategies are available, including random generator sequences, Factorial DOEs [46], Orthogonal and Iterative Techniques, as like as D-Optimal or Cross Validation [47]. Monte Carlo and Latin hypercube [85] are available for robustness analysis. Set of DOE strategies can be easily extended by user-defined strategies.
- Optimization. The optimization core is open, and using the Memetic Computing (MC) approach [115], which is an extension of the concept of Memetic algorithm, the user can define its own optimization algorithm as a set of independent pieces of code called "operators". Operators can be implemented either in Java or Python. Optimization type can be characterized as simulation-based optimization. Built-in optimization heuristics includes: Simulated Annealing-Based Multi-objective Optimization Algorithm (AMOS) [52], Multi-Objective Differential Evolution Algorithm (MODE), Multi-objective Evolution Strategy Algorithm (MOES), Multi-objective Particle Swarm Optimization (MOPSO) [116], NSGA-II [49] and Strength Pareto Evolutionary Algorithm (SPEA-II) [117].
- Post-Processing. Kimeme offers a wide range of visualization tools including Scatter 2D, Scatter 3D, Matrix, Line 2D, Bubble 3D, Generation based, Parallel, PDF, CDF, Scatter 2D/PDF, PCA and Boxplot as well as sensitivity analysis tools. Local Sensitivity as correlation coefficients and partial derivatives can be used only if the correlation between input and output is linear. If the correlation is nonlinear, the global sensitivity analysis has to be used based on the variance-relationship between input and output distribution as Sobol index.

Usage in Architecture Optimization:

Usage of Kimeme in architecture optimization is limited for the same reasons as usage of other simulation-based optimization software for this purpose.

## 4.4.2 A&D Domain Modelling and Analysis Software

In this section we review A&D domain modelling and analysis software developed by Pacelab. Systems designed and developed in the A&D industry often have the need to combine electrical, structural, hydraulic, thermal and communications. Software platforms exist that support modelling the different components and their integration through design, development, simulation, validation and verification of these systems.

### 4.4.2.1 *Pacelab Suite*

Pacelab [118] Suite is platform for knowledge based engineering (KBE), which supplies the functional and procedural infrastructure for early-stage product design: the ability to parametrically model, analyse and size complex technical systems combining all relevant disciplines such as geometry, structure performance and cost.

Key features:

- **Process integration:** Through its .NET based architecture, Pacelab Suite provides a variety of integration and interfacing options to fully leverage proprietary codes, commercial analysis tools and existing MDO and federated software integration/execution environments. Legacy codes (Fortran, C, C++, VB) and commercial tools (Matlab [16], MS Excel, FEM and CAD tools) can be seamlessly integrated into the data models and methods created with Pacelab Suite. This ensures that product data and methods derived from legacy tools are represented and used in exactly the same way as natively defined ones. An open application programming interface permits users to create a customized environment for a specific set of tasks by adding highly specialized or even proprietary functionality and graphical interfaces.
- **Problem Formulation:** Pacelab Suite build mathematical model in process of composition of the engineering model. Specifying the output (the unknowns) of the later solving process is the step following upon composition of the engineering model. The input / output status determines which parameters are known to the system (input) and which parameters are to be calculated in the solving process (output). By setting the output parameters for a given project, engineers determine the overall objective of the engineering task and hence the solving direction. In the next step Pacelab conducts a combinatorial analysis of the mathematical system's dependency structure. This analysis is aimed at identifying inconsistencies and over or under-constrained sections of the mathematical system. Offering a variety of synchronized views and solving strategies, the Resolution System helps users to quickly remedy such faults to achieve the principle of solvability of the system. Once the mathematical system is well-defined and free of inconsistencies, Pacelab computes the optimal Resolution Plan for the given engineering task. The Resolution Plan defines the strategy that will be used by the Resolution System's solver to compute each unknown output parameter in the system. In the Resolution Plan, the mathematical system is broken down into a sequence of solving steps that allow each output variable to be calculated. These solving steps can include simple function evaluations or numerical solving blocks, when input statuses are reversed or when iterative solving cycles are to be performed.
- **Pre-processing:** The mathematical system of Pacelab suite provides automatic identification of output parameters impacted by free parameters and selects constraints to be respected by the optimization process prior to calculation, thus improving performance and user guidance.

- Optimization: Pacelab suite offers single-objective as well as multi-objective optimization capabilities. For this purpose, Pacelab use a library of non-linear, constrained optimization algorithms, which can be supplemented by third party optimization routines. External optimization routines can be integrated using the Pacelab plug-in mechanism. As other software described above optimization techniques offered by Pacelab suite can be characterized as simulation-based optimization. The optimization engine uses fully incremental update mechanism which saves computation resources by minimizing the number of recalculations.
- Post-processing: Multi-objective optimization provided by Pacelab suite through so-called Trade Studies. Trade studies compute a whole set of parameter variations at once and store the complete results sets of each calculation run. The optimum solution can be identified by the user, based on expressive 2D and 3D representations visualizing the effects of parameter variations.

Usage in Architecture Optimization:

Usage of Pacelab suite in architecture optimization is limited for the same reasons as usage of modeFrontier or Isight for this purpose.

#### 4.4.2.2 *Pacelab SysArc*

Placelab SysArc [119], another tool of Pace, represents a technical approach to the aircraft-level design and analysis of aircraft system architectures, which unites the logical definition of systems architectures with the physical layout of system components and their connections in the aircraft geometry. The tight, yet runtime-efficient integration of systems architecture configuration within the conceptual aircraft model allows an instantaneous investigation of the impact of system architecture modifications on the aircraft characteristics and overall performance.

Key features:

- Design capabilities: These include auxiliary geometry and system component library. In order to facilitate the layout of systems architectures, the aircraft model can be equipped with auxiliary geometries including a flexible compartment model and pathway model for connections between systems components such as cables, ducts, or pipes. Compartments provide comprehensive geometric data, including volumes and adjacent surfaces, which can be directly linked to internal thermal analysis models or external analysis tools. Pathways allow the definition of permissible connection routes between components and are prerequisite to automatic routing. The component library contains basic parametric component models covering systems from Flight Control (FCS) and Environmental Control Systems (ECS) to Electrical Power Generation and Hydraulic Systems. Each component can be accompanied by an in-depth hypertext documentation detailing design intent, parameter descriptions and suggested usage to help systems engineers select the appropriate building blocks for their investigation. Part of the parametric description of system components are connector points, or Ports. Ports constitute a specific implementation of Pacelab Suite's Smart Linking technology, which allow the graphical definition of physical and abstract connections between components. If the user draws a line between two electrical components, Pacelab SysArc will create the required power summation formulas or propagate voltages and automatically take into account the voltage drops induced by the resistance of the physical cable connection.



- Optimization: Pacelab SysArc use optimization algorithms (Dijkstra & Steiner tree algorithms) for the so-called automatic routing. This allows finding shortest route between components along previously defined pathways. Automatic selection of suitably sized cables, pipes, ducts, etc. from catalog of standard parts done after optimal pathways computed as well as automatic calculation of distribution elements' key properties such as length, mass, voltage or pressure losses.

Usage in Architecture Optimization:

Pacelab SysArc can be used as complementary tool for architecture optimization. Optimization future is not enough developed to optimize system architecture (only routing subsystem can be optimized).

### 4.4.3 Optimization Packages for Architecture

In this section we review some optimization packages that, by their claim, could be used for architecture optimization.

#### 4.4.3.1 HEEDS MDO

HEEDS (Hierarchical Evolutionary Engineering Design System) MDO [120] is a software package that interfaces with commercial CAE tools in order to automate and improve the search for better product and/or process designs. HEEDS is based on spin-out technology from Michigan State University and utilized across many industries, including Aerospace, Automotive, Biomedical, and Manufacturing. HEEDS MDO is a product of Red Cedar Technology that is known as software development and engineering services company.

Key features:

- Process integration. HEEDS MDO integrates well with all popular CAE applications to automate and expedite design optimization. It can work with multiple software tools to handle pre- and post-processing, simulation, and multidisciplinary optimization. Within HEEDS, data flows automatically among CAD, meshing tools, simulation tools, in-house proprietary codes and cost models, eliminating tedious manual data transfer and costly errors. After a process is captured and validated, it can be used over and over. HEEDS MDO's process automation features include direct portals to common CAE tools for data extraction, automated execution of multiple simulation and analysis tools within a design evaluation process, integration and sharing of data among separate simulations and support for parallel processing on networks, clusters, and multiprocessors. HEEDS MDO features direct input and output portals for the following tools: Abaqus [103], MATLAB, Adams, Nastran [104], ANSYS WB, NX (Input only), Excel, SolidWorks [111], LS-DYNA, SolidWorks Simulation. Additionally, HEEDS MDO offers a powerful generic interface that allows it to link to any commercial or proprietary CAE tool that creates input and/or output files in ASCII format.
- Pre-processing. HEEDS MDO offers a broad range of DOE sampling methods. HEEDS' unique DOE wizard can guide through the definition of the problem, ensuring that the necessary information can be obtained. DOE methods include: Full factorial designs (2-level and 3-level), Fractional factorial designs (2-level and 3-level) [46], Taguchi orthogonal arrays [86], Plackett-Burman designs [108], Latin hypercube designs [85], Central composite designs [82], D-optimal designs [47], Taguchi robust design arrays, User-defined arrays. With HEEDS MDO, stochastic variations can be assigned to all design variables, as well as fixed system parameters, so that a stochastic simulation

can be performed, helping to achieve designs that meet the highest quality standards for robustness and reliability.

- Optimization. HEEDS MDO provides single- and multi-objective simulation-based optimization, by using proprietary optimization search strategies, SHERPA and MO-SHERPA. SHERPA employs multiple search strategies at once and adapts to the problem as it “learns” about the design space. SHERPA requires significantly fewer model evaluations than other leading methods do to identify optimized designs, and often finds a solution the first time. During a single parametric optimization study, SHERPA uses the elements of multiple search methods simultaneously (not sequentially) in a unique blended manner. This approach attempts to take advantage of the best attributes of each method. Attributes from a combination of global and local search methods are used, and each participating approach contains internal tuning parameters that are modified automatically during the search according to knowledge gained about the nature of the design space. MO-SHERPA (Multi-Objective SHERPA) is a modified version of the algorithm SHERPA for multi-objective Pareto search. It uses a non-dominated sorting scheme to rank designs but is quite different from NSGA-II and NCGA in other aspects. MO-SHERPA is designed to be used with projects with multiple objectives when those objectives are in conflict with one another. It works fundamentally like SHERPA but has the advantage of handling multiple objectives independently of each other to provide a set of solutions, each of which is optimal in some sense for one of the objectives. HEEDS COMPOSE module for HEEDS MDO helps optimize sub-systems using smaller models, while maintaining the coupling between the system and sub-system.

Usage in Architecture Optimization:

Usage of HEEDS MDO in architectural optimization is limited because of its simulation-based nature for the same reasons as modeFrontier and other related software.

#### 4.4.3.2 IOSO

IOSO (Indirect Optimization on the basis of Self-Organization) [121] is a multi-objective, multidimensional nonlinear optimization technology. IOSO is based on the technology being developed for more than 20 years by Sigma Technology which grew out of IOSO Technology Center in 2001. IOSO is the name of the group of multidisciplinary design optimization software that runs on Microsoft Windows as well as on Unix/Linux OS and was developed by Sigma Technology. It is used to improve the performance of complex systems and technological processes and to develop new materials based on a search for their optimal parameters.

Key features:

- Process integration. IOSO software can be easily integrated with different applications for engineering analysis both in-house and commercial, such as NASTRAN [104], ANSYS, StarCD, FineDesign, Fluent Concepts NREC Turboopt [102], ANSYS WB2, FLOW-3D, SolidWorks [111], FlowVision, etc.
- Optimization. IOSO group of software consists of:
  - IOSO NM: Multi-objective optimization;
  - IOSO PM: Parallel multi-objective optimization;

- IOSO LM: Multilevel multi-objective optimization with adaptive change of the object model fidelity (low-, middle-, high fidelity models);
- IOSO RM: Robust design optimization and robust optimal control software;

All these software uses efficient simulation-based optimization technologies. IOSO Technology is based on the response surface methodology approach. At the each IOSO iteration the internally constructed response surface model for the objective is being optimized within the current search region. This step is followed by a direct call to the actual mathematical model of the system for the candidate optimal point obtained from optimizing internal response surface model. During IOSO operation, the information about the system behaviour is stored for the points in the neighbourhood of the extremum, so that the response surface model becomes more accurate for this search area. The following steps are internally taken while proceeding from the one IOSO iteration to another:

- the modification of the experiment plan;
- the adaptive adjustment of the current search area;
- the function type choice (global or middle-range) for the response surface model;
- the adjustment of the response surface model;
- the modification of both parameters and structure of the optimization algorithms; if necessary, the selection of the new promising points within the search area.

Multi-objective optimization allows solution of all classes of optimization problems including stochastic, multi-extreme and having non-differential peculiarities. IOSO can solve large-dimensionality problems (up to 100 independent design variables and up to 100 constraints) and stochastic problems, having complex topology of objective and the large number of constraints. However, the IOSO algorithm is not initially designed for usage of enumeration-type design variables. Support of this type of variables was added only in the last version of IOSO.

Usage in Architecture Optimization:

Usage of IOSO in architecture optimization is limited for the same reasons as usage of other simulation-based optimization software for this purpose. Additionally, efficiency of IOSO techniques is questionable under intensive usage of enumeration-type design variables, because these techniques are not initially designed for usage of this type of variables. This may cause additional problem because such type of variables is most common type in architecture optimization (choosing of different part from the catalogue, choosing different option, etc.)

#### 4.4.3.3 *Optimus*

Optimus [122] is a Process Integration and Design Optimization platform developed by Noesis Solutions. Optimus allows the integration of multiple engineering software tools into a single and automated workflow. Once a simulation process is captured in a workflow, Optimus will direct the simulations to explore the design space and to optimize product designs for improved functional performance and lower cost, while also minimizing the time required for the overall design process.

Key features:

- Process integration. The Optimus GUI enables the creation of a graphical simulation workflow. A set of functions supports the integration of both commercial and in-house software. A simple

workflow can cover a single simulation program, whereas more advanced workflows can include multiple simulation programs. These workflows may contain multiple branches, each with one or more simulation programs, and may include special statements that define looping and conditional branching. Optimus' workflow execution mechanism can range from a step-by-step review of the simulation process up to deployment on a large (and non-heterogeneous) computation cluster. Optimus is integrated with several resource management systems to support parallel execution on a computational cluster. Optimus support integration with Abaqus [103], ANSYS, DELMIA, GT Power, LMS Imagine.Lab AMESim, LS-DYNA, MapleSim, Matlab/Simulink [15], MS Excel, MSC Marc, MSC Nastran [104], PAM-CRASH, PAM-STAMP 2G, Ricardo WAVE, STAR-CD and other software.

- Pre-Processing. Optimus supports following DOE methods: Full Factorial (2-level and 3-level), Adjustable Full Factorial, Fractional Factorial [46], Plackett-Burman [108], Central composite [82], Random, Latin-Hypercube, Starpoints, Diagonal, Minimax and Maximin, Optimal design (I-, D- and A-optimal). It is also allow creating various user-defined DOE methods.
- Optimization. The Optimus numerical optimization software offers a robust set of algorithms for single and multi-objective simulation-based optimization. The local optimization algorithms, includes NLPQL, SQP [65], Generalized Reduced Gradient (GRG) and Adaptive Region Method (ARM Order 1 and 2) [123]. Optimus also contains state-of-the-art global optimization algorithms – including Differential Evolution (DE) and Self-adaptive Evolution (SE) [124], Simulated Annealing (SA) and Efficient Global Optimization (EGO) [125]. Additionally, Multi-Gradient Explorer (MGE) algorithm and Multi-Gradient Pathfinder (MGP) algorithm [126] as well as hybrid version of these algorithms available through eArtius Optimization Plugin. Optimus also incorporates into its standard graphical interface any in-house trade study algorithms for single and multiple objectives optimization along with custom algorithm options.

Usage in Architecture Optimization:

Usage of Optimus in architecture optimization is limited for the same reasons as usage of other simulation-based optimization software for this purpose.

#### 4.4.3.4 ACSOM

ACSOM[127], known as Armored Combat System Optimization Modeler or Advanced Collaborative System Optimization Modeler, is the multi-criteria/multi-domain optimization methodology implemented at General Dynamics Land Systems (GDLS).. ACSOM was developed to provide a structured, analytical framework for conducting system level trade studies. ACSOM quickly generates a Pareto set of design solutions that span the entire design space to consider the full spectrum of subsystem options. The vision of ACSOM is to provide a unique set of balanced vehicle concepts to aid design engineers and decision makers in selecting the preferred balance among vehicle performance and program burdens.

Key features:

- Process integration. ACSOM requires definition of a system architecture comprised of a hierarchy of system elements, subsystems, and subsystem options. The options are the decision variables of the model. Data provided for each option include an assessment of the level of performance and burdens achievements, and subsystem option interactions vis-à-vis combinatorial constraints. Programmatic input requirements include system level performance and burden targets/bounds derived from customer requirements documents and interaction through the normal systems

engineering requirements analysis process. Excel workbook is used for identification of the system architecture, options, and associated data.

- Optimization. Optimization technique is not specified. It seems like ACSOM go throw all set of feasible solutions to find Pareto Frontier.
- Post-Processing. The postprocessor draws on results from the model stored in the main database to provide multiple outputs to assist design engineers and decision makers in interpreting the results of a model run. The postprocessor outputs are relayed to the design engineers via an Excel workbook. The workbook contains various tables and graphs related to Pareto optimal solutions. Up to 50 optimal solutions can be displayed. A “What-If Analysis Worksheet” gives the design engineers a “build your own concept” capability. This is accomplished by allowing selection of subsystem options directly and comparing the “user-built concept” with any desired Pareto solution

Usage in Architecture Optimization:

ACSOM can be used for optimization subsystem of complex systems or optimization system with small set of possible parts. Complex systems with large number of parts and/or big numbers of possible topologies can't be optimized by this software by following reasons:

1. Input method is not suitable for architectural purpose. Building architectural template via the data table may cause to various human-related errors even in relatively small systems.
2. It seems like optimizer use combinatorial optimization only. Usage of continuous metrics or design variables is impossible under these settings. This greatly limited scope of possible optimization scenarios where ACSOM can be used. Black-box optimization techniques also not available.
3. Efficiency of optimization techniques used by ASCOM is questionable. This may tend to very long run-time in case of large complex system.

#### **4.4.3.5 Architectural Enumeration & Evaluation toolset**

AEE [128] is a method for rapid, efficient and thorough consideration of enormous architectural design spaces to find the best low-complexity solutions. AEE assembles promising system architectures from any number of candidate technologies and evaluates them relative to a set of customer-value metrics, which can include a complexity metric as a proxy for cost. One of the key ideas of AEE is that complexity metrics can roll up the complexity within each hierarchical module based on the complexity of its internal elements and interconnections. This enables a multi-level approach that enhances the efficiency of AEE and improves the quality of assessment because, at each successive level, the trade space is not only successively refined, but it can also be adaptively refined according two user-defined complexity metrics. The process going from higher level of abstraction two lower. Issues absent from the previous higher level of abstraction can be introduced to the system model, based on which architectures are brought forward from the previous level as promising.

Usage in Architecture Optimization:

AEE is a not production tool, but some research toolset used for testing and evaluation of quality and performance of underlining methods. It can't be used immediately in production environment. Moreover, usage of AEE method is questionable when target metrics is not depending on system complexity. Another issue is that number of feasible solution growth exponentially with number of possible options and number elements in the model. Usage of AEE for complex systems with large number of options can take a very

long time or even be computationally intractable. All these, limited usage of AEE in architecture optimization.

#### 4.4.3.6 *Architecture Optimization Workbench*

AOW [129] is a set of tools created by IBM Research for architecture optimization purpose.

Key features:

- Process integration. At the current state AOW can be integrated with IBM Rational Rhapsody, Excel and Pacelab Engineering Workbench.
- Optimization. AOW use CPLEX solver [130] for optimization purpose. Linear, Mixed-Integer and quadratic programming techniques are available. Multi-objective optimization performed by using Diversity Maximization Approach (DMA) algorithm [131]. The DMA algorithm developed to find number of most

**Post-Processing.** The postprocessor draws on results from the model stored in the main database to provide multiple outputs to assist design engineers and decision makers in interpreting

#### 4.4.3.7 *Summary*

Existing software suitable for Architecture Optimization includes following products: modeFRONTIER, modelCenter, Isight, Pacelab Suite, Pacelab SysArc, HEEDS MDO, OptiY, IOSO, Nexus, Kimeme, and Optimus.

Key features:

- Process integration. Existing software can be integrated with large number of tools including: CAD/CAE tools, FEM tools, CFD tools, scripts, databases and other software. Various components typically integrated via GUI in the single model structure. Data flows from component to component where each component can be exposed by different tool. Since output parameters of one component typically are input parameters of another component, corresponding tools running sequentially, but parallel tool running is also available where it is suitable. The model can be used for simulation and optimization purpose.
- Pre-Processing. Most of the tools allow pre-processing via various DOE techniques. DOE techniques used to extract as much information as possible from a limited number of simulation runs or to provide the initial data points for optimization algorithms.
- Optimization. Optimization techniques can be characterized as multi-objective simulation-based optimization over set of "black-boxes". Various single and multi-objective heuristic algorithms can be used for optimization purpose. These includes: Pattern search methods, Gradient-based methods, Genetic algorithms, Evolution strategies and others. Some of the tools allow including additionally user-defined optimization algorithms.
- Post-Processing. Most of the tools offer various visualization components which enable the user to explore, to filter and to rank the set of optimal solutions of a multi-objective problem. Some of the tools allow running sensitivity analysis to verify the system's sensitivity to manufacturing tolerances or small changes in operating conditions.

#### 4.4.4 Usage for Architecture Optimization

Existing software can be used for optimization of parts of complex systems or optimization of a whole system over a small set of predefined architectures. This software may not be sufficient for solving complex CPSoSs for the same reasons highlighted in Section 4.3.1.

To summarize, the existing black box / simulation based optimization (BB/SBO) software is good for sizing of design parameters for a given architecture since (1) there is no an easy way to define potential architectures; (2) BB/SBO is not scalable and/or could be far from the optimal heuristic. The enumeration-based method is suitable for relatively small design problems and not scalable.

### 4.5 Techniques and tools for the design of CP(H)SoSs

In this section we review techniques and tools for the design of CP(H)SoSs, with humans in the loop. In Section 4.5.1 we consider hardware acceleration. In Section 4.5.2 we look at application profiling. In Section 4.5.3 we review real-time system monitoring.

#### 4.5.1 Hardware Acceleration Using Vitis

The purpose of the section is to provide a summary of the capabilities of the recently announced Vitis tool and identify specific ways of how to use these tools in the context of the CPSoSaware project.

The Vitis unified software platform is a new tool that combines all aspects of Xilinx software development into one unified environment. The Vitis software platform supports both the Vitis embedded software development flow, for Xilinx Software Development Kit (SDK) users and the Vitis application acceleration development flow, for software developers looking to use the latest in Xilinx FPGA-based software acceleration.

In general, the Vitis application acceleration development flow provides a framework for developing and delivering FPGA accelerated applications using standard programming languages for both software and hardware components (Figure 9). The software component, or host program, is developed using C/C++ to run on x86 or embedded processors, with OpenCL API calls to manage runtime interactions with the accelerator. The hardware component, called kernel, can be developed using C/C++, OpenCL C, or RTL. The Vitis software platform accommodates various methodologies, letting you start by developing either the application or the kernel.

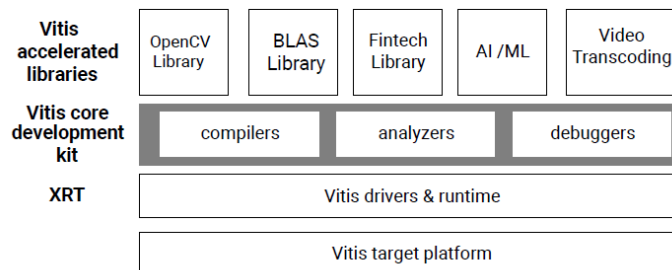


Figure 9 General overview of the Vitis framework

It is important to note that the Vitis core development kit also supports running the software application on an embedded processor platform running Linux, such as on Zynq UltraScale+ MPSoC devices. It is expected that the CPSoSAAware project will rely on a Zynq FPGA for performing edge-level computations. For the embedded processor platform, the Vitis core development kit execution model also uses the OpenCL API and the Linux-based Xilinx Runtime (XRT) to schedule the HW kernels and control data movement. The Vitis core development kit includes the v++ compiler for the hardware kernel on all platforms, the g++ compiler for compiling the application to run on an x86 host, and an Arm compiler for cross-compiling the application to run on the embedded processor of a Xilinx device. For the CPSoSAAware project, the edge-level node will be consisted of a multicore ARM processor along with a FPGA fabric (tightly coupled to the ARM cores).

The API calls, managed by XRT (Xilinx runtime library), are used to process transactions between the host program and the hardware accelerators. Communication between the host and the kernel, including control and data transfers, occurs across the PCIe® bus or an AXI bus for embedded platforms. In the CPSoSAAware program the latter case will be utilized. While control information is transferred between specific memory locations in the hardware, global memory is used to transfer data between the host program and the kernels. Global memory is accessible by both the host processor and hardware accelerators, while host memory is only accessible by the host application. Data can be passed directly from the software program to the specified kernels, or it can be placed in global memory which is shared memory space accessible by both the software and the kernels. For instance, in a typical application, the host first transfers data to be operated on by the kernel from host memory into global memory. The kernel subsequently operates on the data, storing results back to the global memory. Upon kernel completion, the host transfers the results back into the host memory. Data transfers between the host and global memory introduce latency, which can be costly to the overall application. To achieve acceleration in a real system, the benefits achieved by the hardware acceleration kernels must outweigh the added latency of the data transfers. This will be one of the main challenges in CPSoSAAware project, the edge-level computations are expected to be dominated by data intensive applications.

The general structure of this acceleration target platform is shown in the Figure 10.



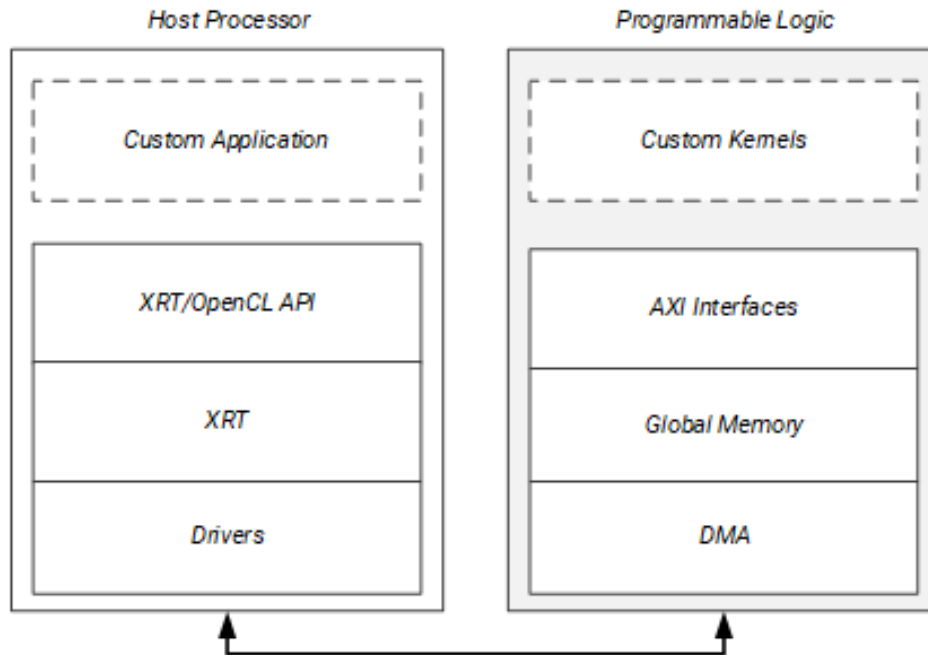


Figure 10 Data flow between the host and kernel. The FPGA hardware platform, on the right-hand side, contains the hardware accelerated kernels, global memory along with the DMA for memory transfers. Kernels can have one or more global memory interfaces and are pro

The execution model can be broken down into the following steps:

- The host program writes the data needed by a kernel into the global memory of the attached device through the AXI bus
- The host program sets up the kernel with its input parameters
- The host program triggers the execution of the kernel function on the FPGA
- The kernel performs the required computation while reading data from global memory, as necessary
- The kernel writes data back to global memory and notifies the host that it has completed its task
- The host program can transfer the data from global memory to host memory or can transfer the data to another kernel for processing

The Vitis core development kit offers all of the features of a standard software development environment:

- Compiler or cross-compiler for host applications running on x86 or Arm® processors
- Cross-compilers for building the FPGA binary
- Debugging environment to help identify and resolve issues in the code
- Performance profilers to identify bottlenecks and help you optimize the application

The build process follows a standard compilation and linking process for both the host program and the kernel code. As shown in Figure 11, the host program is built using the GNU C++ compiler (g++) or the GNU C++ Arm cross-compiler for MPSoC-based devices. The FPGA binary is built using the Vitis compiler.

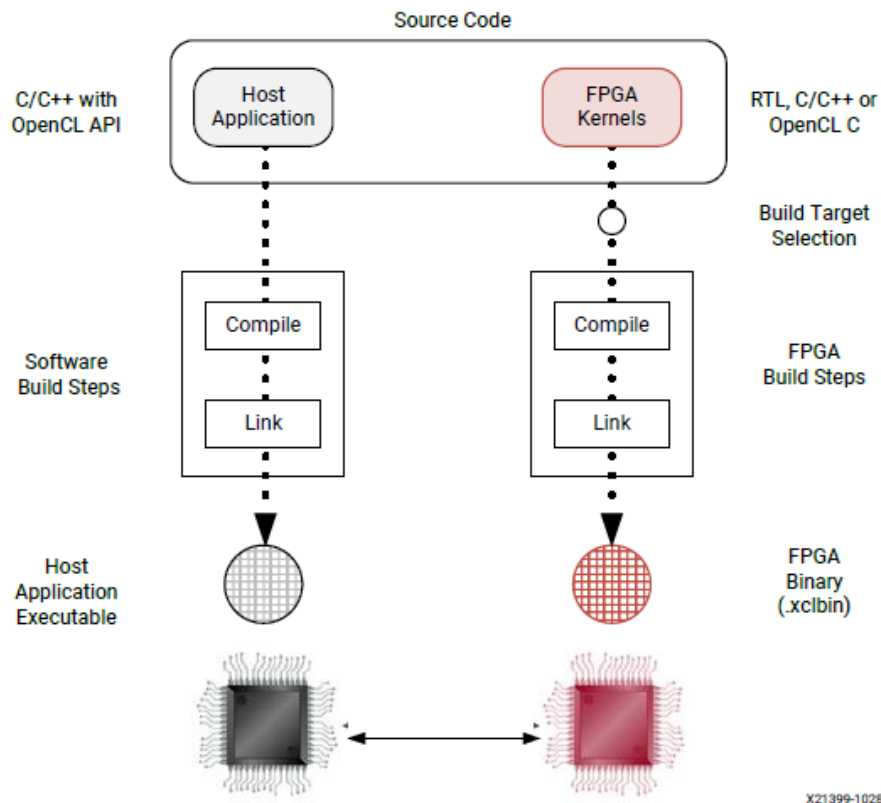


Figure 11 Vitis compile flow

Kernels can be described in C/C++, or OpenCL C code, or can be created from packaged RTL designs. As shown in Figure 12 below, each hardware kernel is independently compiled to a Xilinx object file (.xo). Xilinx object files are linked with the hardware platform to create an FPGA binary file (.xclbin) that is loaded into the Xilinx device on the target platform.

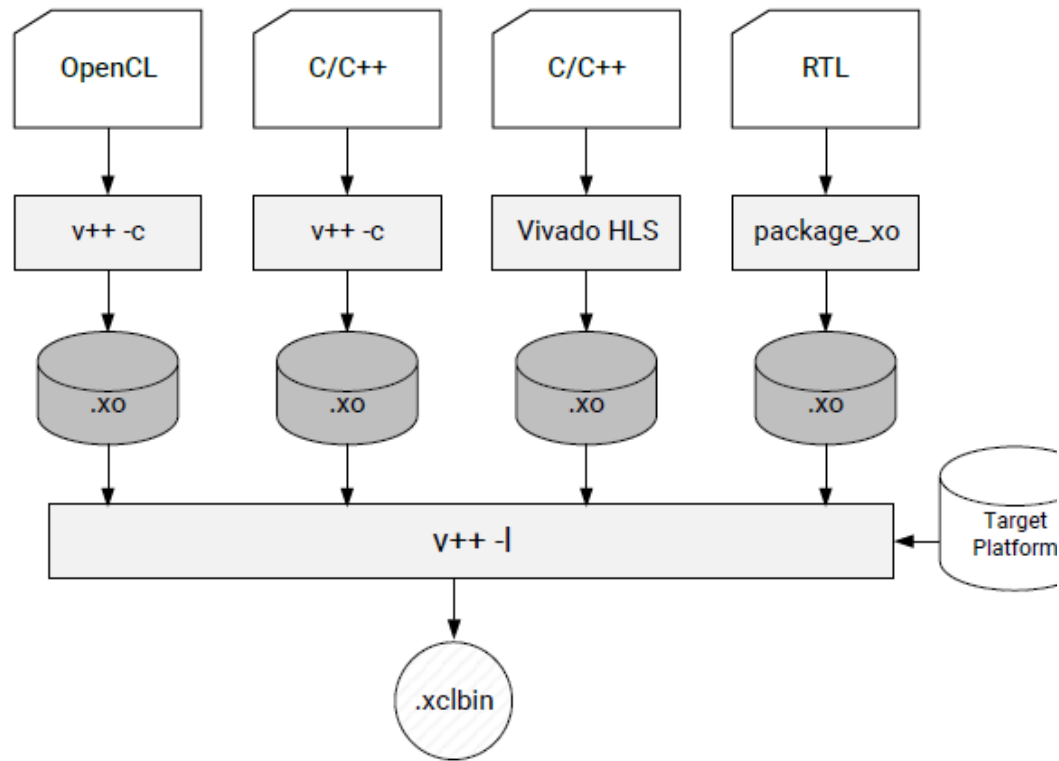


Figure 12 Vitis hardware generation flow

The Vitis compiler provides three different build targets, two emulation targets used for debug and validation purposes, and the default hardware target used to generate the actual FPGA binary:

- Software Emulation (sw\_emu): Both the host application code and the kernel code are compiled to run on the host processor. This allows iterative algorithm refinement through fast build-and-run loops. This target is useful for identifying syntax errors, performing source-level debugging of the kernel code running together with application, and verifying the behaviour of the system.
- Hardware Emulation (hw\_emu): The kernel code is compiled into a hardware model (RTL), which is run in a dedicated simulator. This build-and-run loop takes longer but provides a detailed, cycle-accurate view of kernel activity. This target is useful for testing the functionality of the logic that will go in the FPGA and getting initial performance estimates.
- System (hw): The kernel code is compiled into a hardware model (RTL) and then implemented on the FPGA, resulting in a binary that will run on the actual FPGA.

#### 4.5.2 Profiling the Application

The Vitis core development kit generates various system and kernel resource performance reports during compilation. These reports help you establish a baseline of performance on your application, identify bottlenecks, and help to identify target functions that can be accelerated in hardware kernels. The Xilinx Runtime (XRT) also collects profiling data during application execution in both emulation and system mode configurations. Examples of the reported data includes:

- Host and device timeline events
- OpenCL API call sequence
- Kernel execution sequence
- FPGA trace data including AXI transactions
- Kernel start and stop signals

Together the reports and profiling data can be used to isolate performance bottlenecks in the application and optimize the design to improve performance. Optimizing an application requires optimizing both the application host code and any hardware accelerated kernels. The host code must be optimized to facilitate data transfers and kernel execution, while the kernel should be optimized for performance and resource usage. There are four distinct areas to be considered when performing algorithm optimization in Vitis: System resource usage and performance, kernel optimization, host optimization, and data transfer optimization. More details for each of these levels are depicted in Figure 13 below:

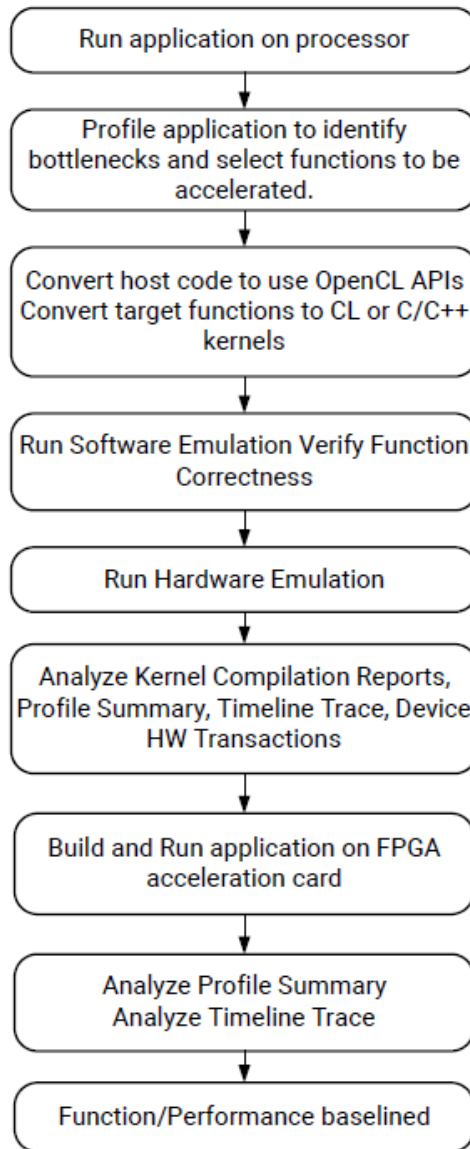
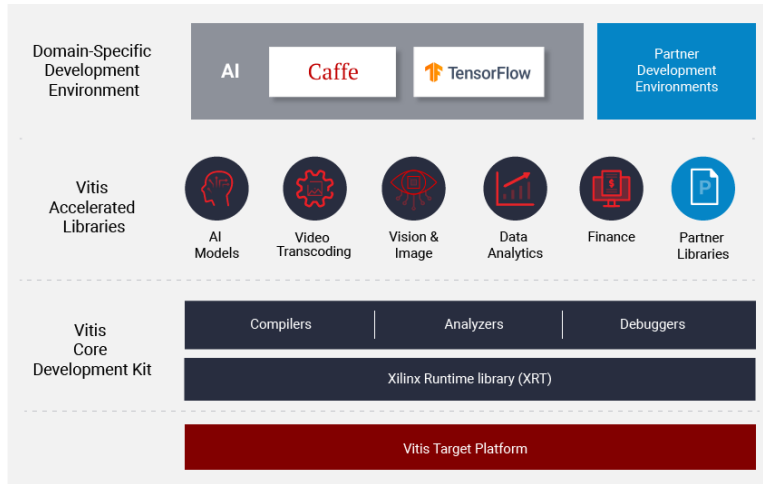


Figure 13 Vitis optimization flow

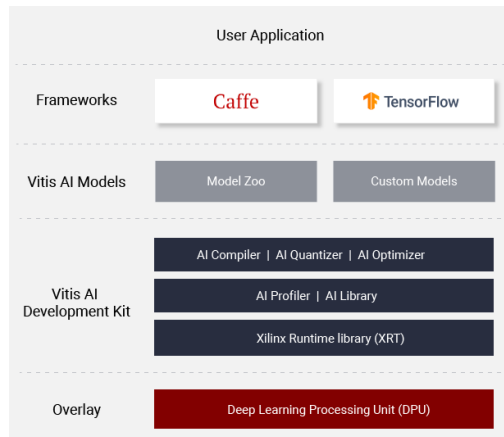
#### 4.5.2.1 Hardware Acceleration of AI/ML/DNN Applications

It is important to mention that Vitis is not only an IDE to design customized hardware IPs, but a complete ecosystem. As such, Vitis contains a rich set of hardware-accelerated open-source libraries optimized for Xilinx hardware platforms and also it contains support for various higher-level frameworks. As Figure 14 depicts, Vitis includes hardware-accelerated partner libraries and zero-code changes, pre-built applications for AI models, vision/image applications and data analytics and it also supports two commonly used AI frameworks (caffe and tensorflow). The just mentioned features are very important in the context of the CPSoSaware project since the project relies heavily in AI algorithms and models. These are the main reasons that drive our consideration of this framework to be used as the main hardware acceleration framework in the project.



**Figure 14** Key components of the Vitis unified software platform

In addition, Vitis includes accelerated-libraries for Math, Statistics, Linear Algebra, and DSP offering a set of core functionality for a wide range of diverse applications.



**Figure 15** Vitis AI inference library

Apart from the above, the Vitis AI inference library includes various other tools that would help to further optimized the inference part of a given AI application (Figure 15). More specifically, Vitis provides a comprehensive set of pre-optimized models that are ready to be deployed on Xilinx devices. In addition, the said library includes a quantizer that supports model quantization, calibration, and fine tuning. This library contains also the AI profiler that provides layer by layer analysis to identify specific bottlenecks. A particularly interesting tool is the AI optimizer that is described below and also illustrated in the following figure (Figure 16).



Figure 16 Vitis AI optimizer

The AI optimizer can reduce model complexity by 5x to 50x with minimal accuracy impact. In essence, the AI optimizer relies on model pruning to perform the compression (as depicted in the following Figure 17).

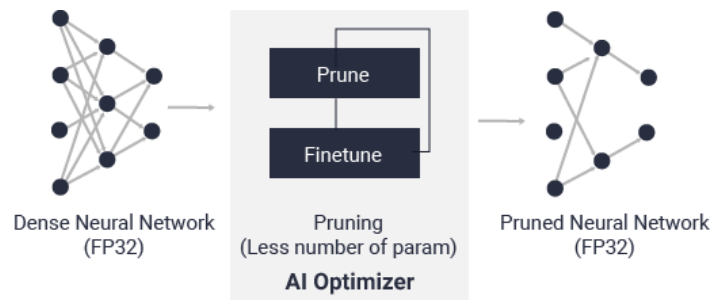


Figure 17 Example of model pruning

### 4.5.3 Real-time monitoring

For the use case of semi-autonomous driving, the planned CP(H)SoS is anticipated to require real-time monitoring of the driver’s attention. Real-time monitoring of driver attention by computer vision techniques is a key issue in the development of advanced driver assistance systems. Therefore, we look at this example in order to develop a methodology for CP(H)SoS design. Therefore, we need to monitor the human driver’s state from CPS part of the design process. We use the example of a human driver because it is connected to use case, but we are looking to extend the design techniques to other CP(H)SoS.

The objective of a user state monitoring system has to satisfy the next points:

- **Affordability:** The price is one of the main factors that kept on mind during the design phase.
- **Portability:** Additionally, it can be easily installed in different vehicle models.
- **Safety:** The safety of the system is achieved by choosing the appropriate location for each component.
- **Speed:** The response and processing time to react in case of a driver’s emergency is one of the key factors since the accident happens in a few seconds.
- **Otherwise,** other algorithms have been chosen.

A primary factor of driving on roads is the driver's attention, and once this attention is lost, significant accidents could happen. The attention of the driver can be diverted by many factors:

- using mobile phones
- changing radio stations
- eating and drinking
- daydreaming

In addition to that, sleepiness due to stress or fatigue; when the driver is sleepy or tired, his reaction is slower than the typical driver, which leads to accidents. Many symptoms can help detect sleepiness or distraction of the driver. The main symptom is the eyes of the driver. One of the ways that assists the driver in paying attention while driving is to add an eye-tracking system using a camera to detect sleepiness due to stress, fatigue, or any distraction. The system alerts the driver when his attention is distracted. This system is to be added to the wearable bracelet, which is used to monitor the physiological parameters of the driver.

Real-time eye-tracking system is used to track the driver's eye. When the driver is drowsy or distracted his response time to react in different driving situations is slow. Therefore, there are higher possibilities of accidents.

There are three ways of detecting driver's drowsiness.

- The first one is the physiological changes in the body like pulse rate, brain signals, and heart activity which can be detected by a wearable bracelet system.
- The second way is behavioural measures, for example, sudden head nods, eye closure, blinking, and yawning which is achieved by the proposed eye-tracking system.
- The third way is vehicle-based like lane position and steering wheel movements.

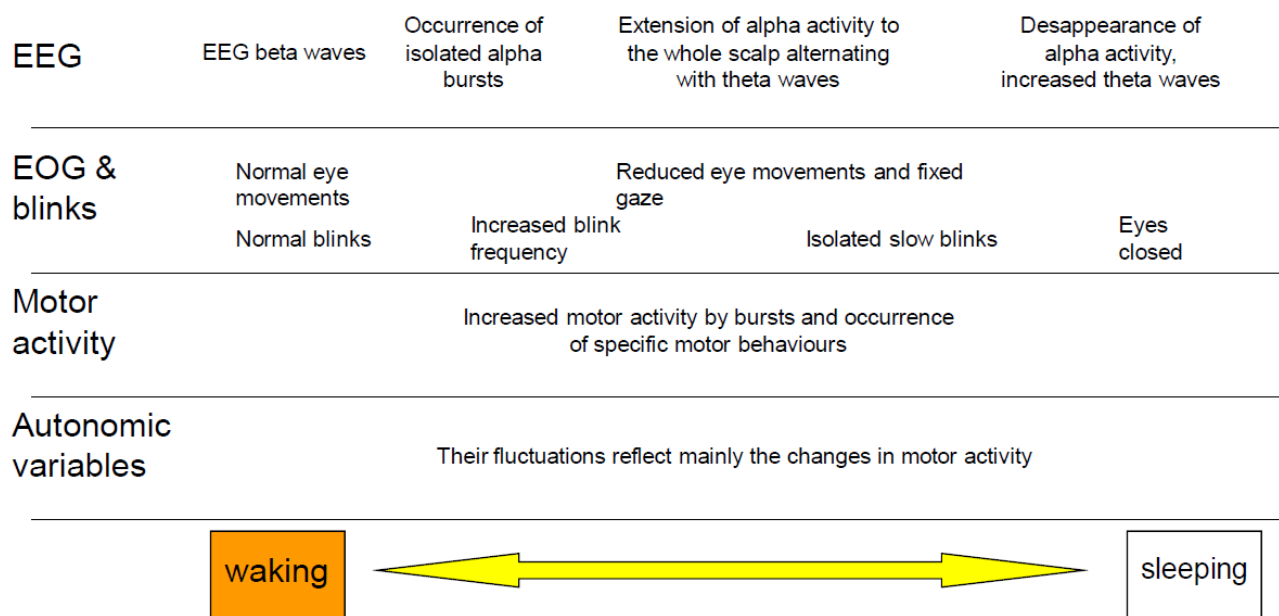
Based on the literature, the eye-tracking system is the most accurate and precise way to detect drowsiness and fatigue. In addition to that, it is used to detect the driver's attention on the road which might happen due to texting on the mobile phone, changing the radio station, or chatting with passengers [132], [133].

#### *4.5.3.1 Monitoring factors through the driving*

Figure 18 presents an overview of the needed physiological functions for the monitoring of the wakefulness and the transition to sleep. It summarizes the evolution of various observations that characterize the involuntary transition from waking to sleeping states.



## Starting unexpectedly in an active state with eyes open



**Figure 18** Description of the needed physiological function for the Driver state diagnostic [134].

For drowsiness, the reported results can be clustered according to activity measures (direct inputs on the steering wheel, pedals etc.) and performance measures (refer to the external criteria to evaluate driving quality), both in longitudinal and lateral control.

Longitudinal control:

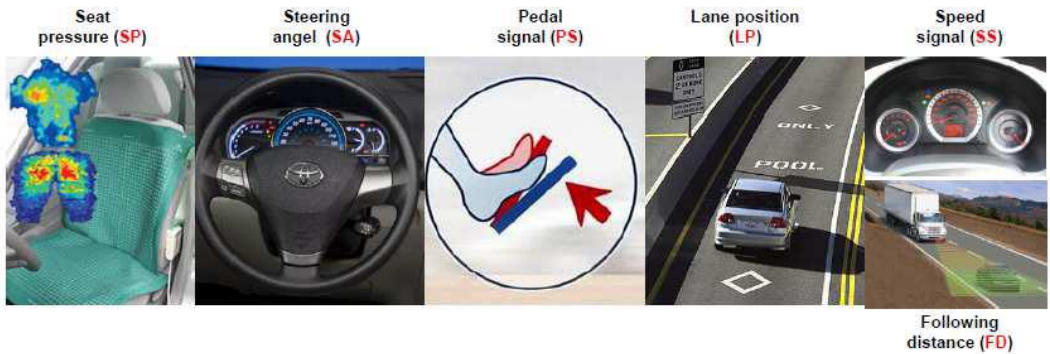
- Driving performance measure: speed (mean and variability) and distance control (distance to lead vehicle)
- Driver activity measure: pedal activity (driver's use of pedals)

Lateral control:

- Driving performance measure: lane keeping performance (standard deviation of the lateral position, time to line crossing, number of lane crossings, mean lateral position, mean yaw rate)
- Driver activity measure: steering behaviour (magnitude and frequency of steering activity), steering variability, slow and fast steering corrections

Another group is the assessment of driver's reaction to specific events (e.g. braking reaction to suddenly braking lead vehicle). Also, the performance in a secondary task or vigilance task (reaction times and error rates) is measured.

Figure 19 shows different factors that can be monitored during driving, including driver, vehicle, and environmental factors. Figure 20 shows different driving situations that may occur while driving and relative alertness level of the driver. Some possible corresponding messages are shown.



Following distance (FD)



Figure 19 Different types of factors that can be monitored for identifying the driver's status.

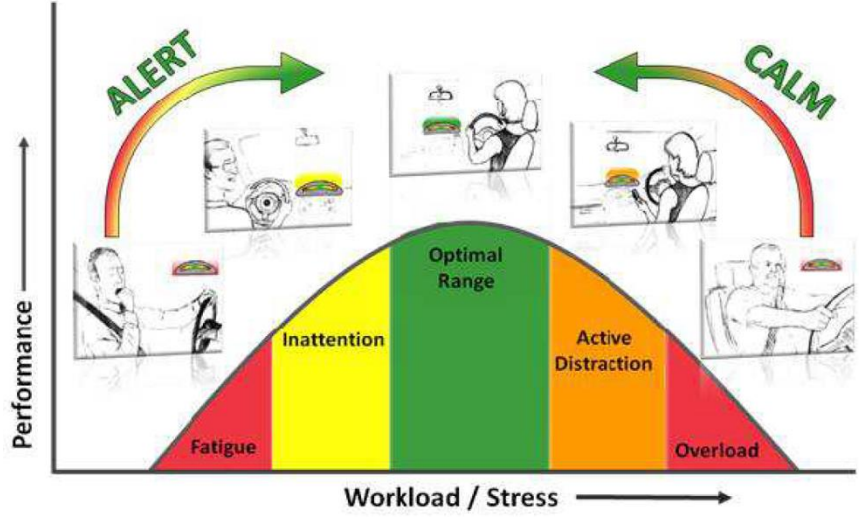


Figure 20 Different driving situations and corresponding messages from a real-time monitoring system.

Table 3 presents the most relevant head and eye-based metrics obtained from eye-tracking systems. Visual distraction is usually associated with looking away from the road scene. Regarding the eyes, the gaze is the dominant subprocess used to detect distraction. Eyes-Off-Road (EOF) duration is perhaps the most used metric to detect distracted drivers, the higher the time, the lower the driver’s awareness due to its simplicity and effectiveness to check if the driver was looking to the road. In studies where researchers create virtual areas-of-interest, then glance’s space and time dimensions allow a more detailed analysis than the binary approach of eyes on/off the road. Glance Pattern refers to a sequence of areas-of-interest fixated by the driver. Areas-of-interest sequence allows predicting the driver’s intentions, for instance, mirrors checking before overtaking another vehicle. Visual task engagement is also possible to detect, by comparing with the normative pattern. Mean Glance Duration highlights the time spent on each area-of-interest, which similar to the previous Glance Patter utility enables the detection of disproportional gaze time allocation as an indicator of task engagement/distraction [135].

**Table 3 Head and eye-based metrics**

Type Distraction	Reference Metrics
Visual	Glance Pattern
	Mean Glance Duration
	Eyes-Off-Road Duration
Auditory	Pupil Diameter
	Blink Frequency
Mechanical	Head direction[/
Cognitive	Pupil Diameter

Pupil Diameter has been reported to be sensitive to Cognitive (i.e. mind wandering) and for Auditory (i.e. reacting to cellphone ring). The pupil reacted consistently by allowing distinguishing between different task, and also the level of difficulty. Mechanical distraction is related to the driver’s body posture during the driving task. For instance, a driver facing the passenger seat reduces his vision of the road centre and also is in a non-ideal position for resuming control of the car in the case of a sudden event. Head direction has been used as a variable to assess the driver distraction. A typical processing scheme for face-based driver monitoring includes the following steps (Figure 21)[133]:

- face localization
- localization of facial features (e.g. eyes or mouth)
- estimation of specific cues related to fatigue or distraction

- the fusion of cues in order to determine the global attention level

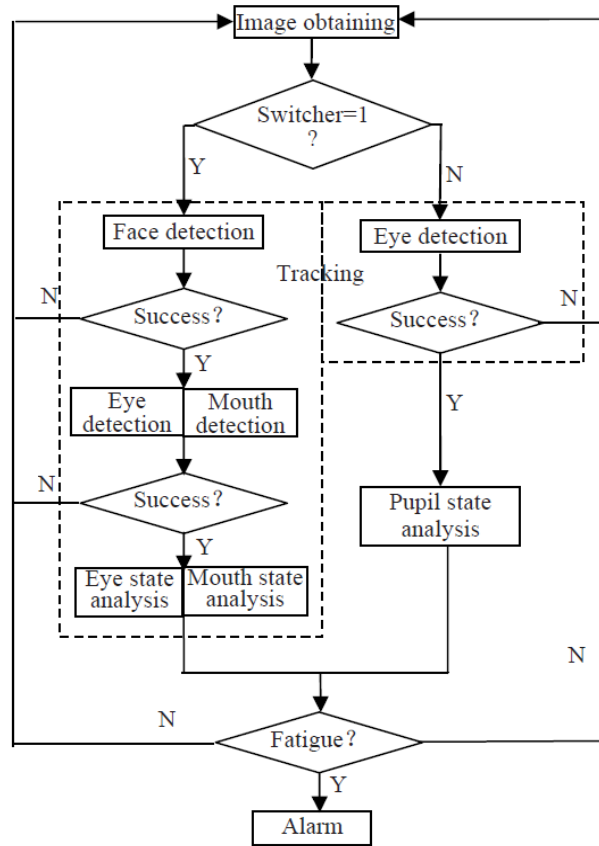


Figure 21 Pipeline for fatigue estimation based on face analysis [136]

#### Real-Time Driver State Monitoring Using a CNN Based Spatio-Temporal Approach [135]

The flow chart of our approach is shown in Figure 22. Initially,  $N$  frames are selected from each action video ( $N = 4$  in this example) based on a sparse selection of frames. Next, A CNN, which is pre-trained on a large-scale image dataset, is applied to extract features from each selected frame. The extracted features are then concatenated and applied as input for the classification, which is achieved with a softmax layer to predict class conditional driver action probabilities.

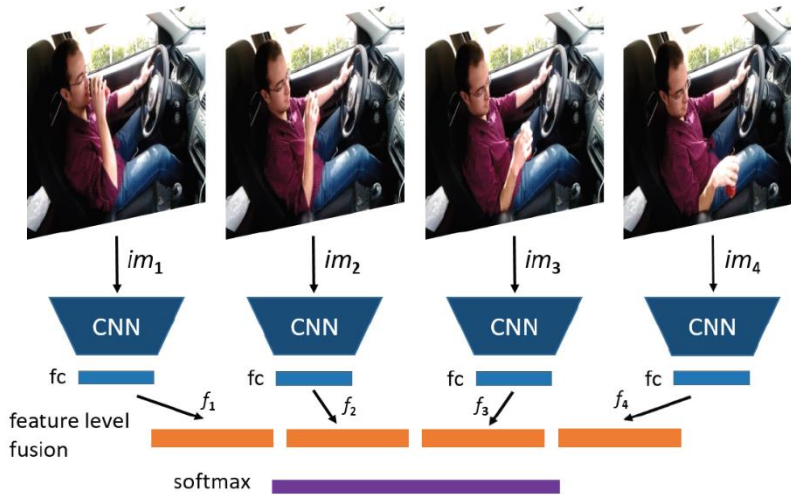


Figure 22 Flow chart of our approach to classify driver distraction level. An example from the Distracted Driver dataset for "Drinking" action.

#### Real Time Car Driver's Condition Monitoring System using ECG signal [137]

The design of ECG (Electrocardiogram) sensor with conductive fabric electrodes and PPG (Photoplethysmogram) sensor to obtain physiological signals for the car driver's health condition monitoring. ECG and PPG signals are transmitted to a base station connected to the server PC via a personal area network. An intelligent health condition monitoring system is designed at the server to analyse the PPG and ECG signals. The purpose for an intelligent health condition monitoring system is managed to process Heart Rate Variability HRV signals analysis derived from the physiological signals in time and frequency domain and to evaluate the driver's drowsiness status.

Figure 23 shows one possible example of an overall system architecture for a car driver's condition monitoring system, which represents one part of the semi-autonomous vehicle use case. The proposed system consists of three parts: sensor, personal area network, and server. The sensor part includes ECG sensor, PPG sensor, and wireless sensor node, which measure physiological signals from the user's hands and send data to a base station via IEEE 802.15.4. The transmitted physiological signals from wireless sensor nodes are saved, analysed, and displayed at the server through the personal area network environment for the practical test properly.

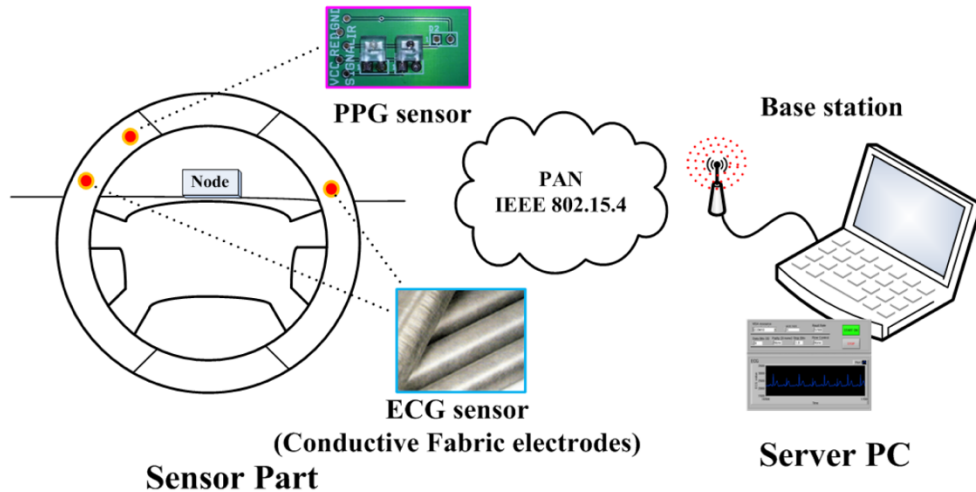


Figure 23 System architecture of real time car driver's condition monitoring system.

For analysis of the measured physiological signals, HRV signals are defined as the constant change of the interval between heart rate. In general, HRV signals are easily obtained and used as an indicator of the autonomic nervous system over the stress and drowsiness related factor since the autonomic nervous system is influenced by the sympathetic nervous system and parasympathetic nervous system. HRV signals are usually calculated by analysing a time series of beat to beat intervals from the ECG or derived from a pulse wave signal measured by means of the PPG waveform.

Safety risk assessment based on a real-time location of the user in an industrial environment [135]

Recent studies related to the safety management have asserted that most accidents on sites could have been reduced and some even eliminated, if there existed an effective and consistent safety management process of identification, planning, education/training, and inspection. Safety risk assessment is very important for developing a safety management system.

Figure 24 shows a logic diagram that briefly describes the safety assessment method associated with our example. On-site assets affecting the safety risk are catalogued into the three groups: workers, hazards including dynamic and static hazards, and safety supervisors. Site assets location data are collected. Relative position relationships, in the form of a time series, between workers and hazards as well as for safety supervisors are obtained by processing and analysing the location data. The HMM is used to find the most likely probability distribution of each monitored worker's states, followed by obtaining the real-time safety risk for each worker.

On-site workers carry devices implementing the provision of monitoring and communication. Thus, the monitoring application not only can provide worker real-time safety risks information for managers but also can send "warnings" or "alerts" to workers and site supervisors. The application also enables computing to be bidirectional and ubiquitous by utilizing state-of-the-art computer and communication technology. Based on the relative positions between workers and hazards, different safety states are defined. Figure 25 gives an example of safety states definition associated with another example of a CP(H)SoS: a crane and operator. The HMM calculates the probabilities of different safety states for a worker.

This example CP(H)SoS consists of an integrated system that tracked worker and asset locations using GPS and Wi-Fi locating technology. The application consists of a series of human-machine interfaces (HMIs) for end-user. Those compatible display terminals include multimedia dispatchers, personal computers, and portable devices such as smartphones and tablet computers.

The main function of the application is to provide engineers and project managers with visual vision including the real-time trajectories of workers on site, information on work context awareness, and real-time safety risk assessment. The users of the system can be divided into two groups of on-site users and off-site users. On-site users include workers, site supervisors, and machinery. The software is an Android App that controls the GPS module and the Wi-Fi module of smartphones. GPS is used when users are outdoors and Wi-Fi when users are in indoor places such as tunnels. Off-site users include engineers, project managers, and researchers. The main application for off-site users is a web-based software system allowing access to any authenticated user with internet access, no matter where they are.

The web page provides users with a view of the realtime locations of the on-site assets. The historical trajectories are also available for ad hoc queries. These functions assist engineers and project managers to be aware of the situation on-site very conveniently. For safety management purposes, the identification and classification of hazards and real-time worker risk assessments are implemented. Thus, a knowledge-based system is created in which system parameters, threshold values, and rules are stored and organized.

Unsafe worker activities are direct causes of accidents. Different unsafe activities are usually associated with specific hazards. Most dynamic hazards on site are caused by vehicles, machinery, and risky tools. A static hazard and its affected area remain unchanged for a relatively long period. These latter hazards consist of elevation changes, hazardous installations, and other danger zones. For both dynamic and static hazards, the critical measurable factor is the distance between the workers and hazards. Accidents are more likely to happen if workers are close to hazards. Dissimilarity is that the position of a static hazard is fixed while a dynamic hazard is always moving. In consequence, in addition to the tracking of workers, dynamic hazards must also be monitored in order to make valid safety risk assessments.

Another critical factor is the presence of safety supervisors. They are responsible for safety management and possess specialized safety knowledge. Effective safety supervision improves the safety performance of workers. The safety supervisor must be in such a position that he can have a positive impact on the workers' behaviour. The distance between worker and safety supervisor must be pre-defined and hence is another measurable factor. Monitor states are compared with safety states in assessing the effectiveness of supervision, as shown in Figure 24, Figure 25 and Figure 26.

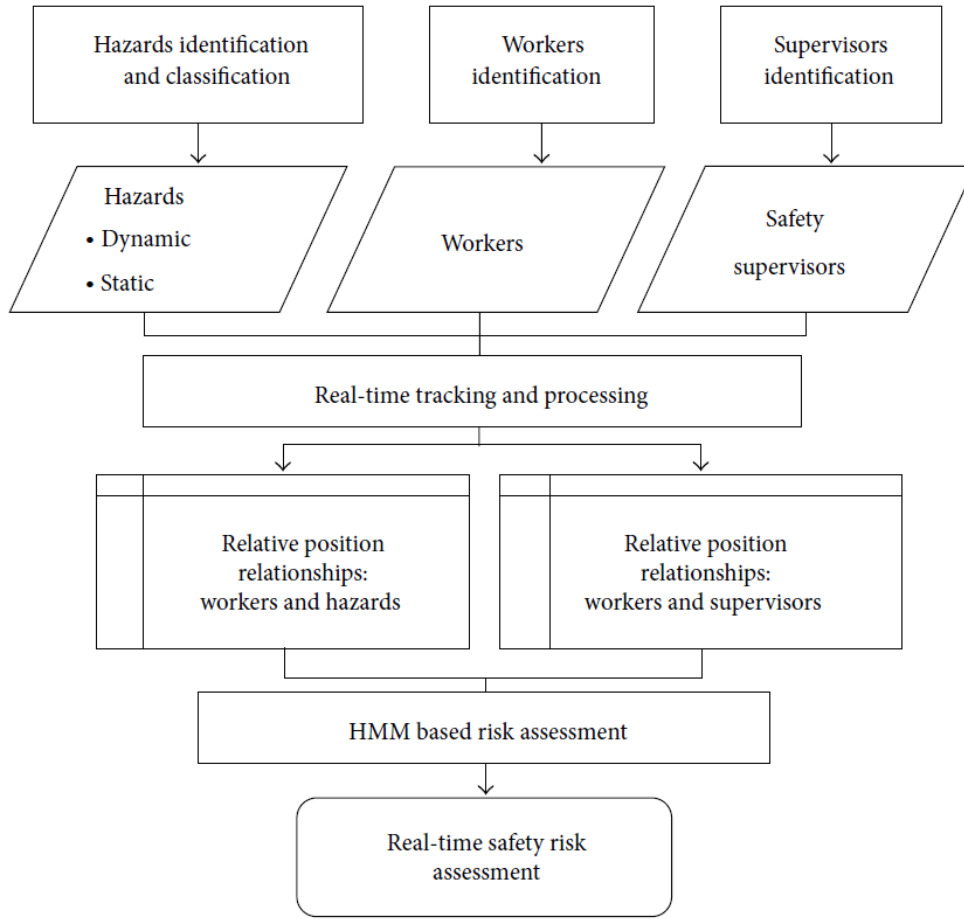


Figure 24 Pipeline of the real time safety method.

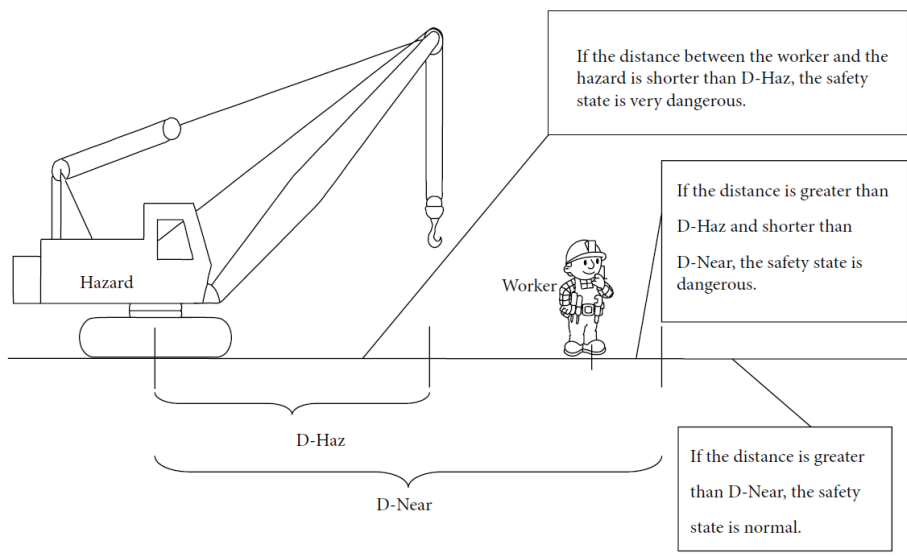


Figure 25 Definition of safety states associated with a crane.



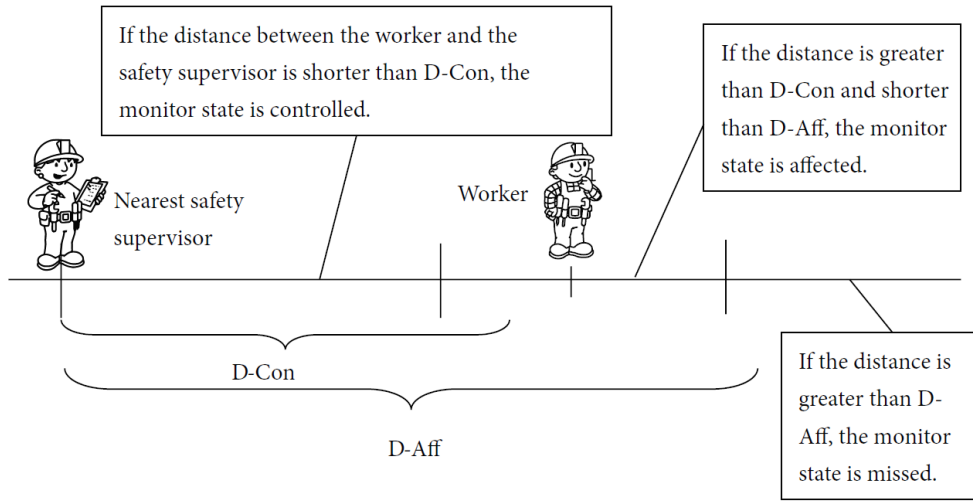


Figure 26 Definition of monitor states with safety supervisor.

## 5 Techniques and tools enabling the simulation of CPSs

In this section, techniques and tools enabling simulation of CPSs are described and analysed. The analysis of simulators is divided into tools directly covering scenarios of interest (automotive and human-robot interaction) and supportive tools for architecture level simulation and communication simulation. In each section, available tools are described and compared against CPSoSAAware requirements and most suitable simulators are proposed for each the applications considered.

### 5.1 Autonomous Driving scenario

#### 5.1.1 Introduction

The first use case is focused on connected semi- autonomous vehicles where we will perform trails focused on Human-in-the-loop scenarios, like nonpredictable failures that may involve the human driver and how they affect the design operation continuum support of the CPSoSAAware solution, as well as human situational awareness enhancement when using the CPSoSAAware architecture. We also use this use case to access the cybersecurity mitigation strategies using the CPSoSAAware architecture and its response to cyberattacks.

#### 5.1.2 Requirements and description of components

##### 5.1.2.1 3D simulation for AV/ADAS with multiple sensors

Research in autonomous driving requires huge amount of data to be collected that cover multitude of different cases which must be considered both for training and validation of the algorithms. Training and validation of the algorithms in simulation is an alternative to extremely expensive and time-consuming data collection in the physical world. Simulation platforms can significantly reduce cost and time of data collection and can be used for creating scenarios that are too dangerous to arrange in real world. Obviously, all the algorithms training and validation should not be reduced to the simulated data exclusively. The most common approach in the industry is to use both simulation and real-world data collection to create robust and reliable solutions at the most optimal cost.

High level requirements:

- Realistic representation of the environment;
- Available sensors (LIDAR, GNSS, IMU, RADAR, Camera, Vehicle data (CAN));
- Machine learning support;
- Communication interfaces (ROS/ROS2 [138]);
- Possible simulation of multiple agents;
- Possible extension with additional modules: Drivers behaviour, V2X communication, cooperative collision warning system.

##### 5.1.2.2 Drivers behaviour modelling

Realistic driver behaviour models are critical for accurate simulation of driving scenarios and extend possibilities of simulation-based research in autonomous driving. Human driver modelling can be achieved with two different approaches: Rule based modelling (advanced parametric models) and data-driven modelling (machine learning based, GAN networks, behavioural cloning etc.).

High level requirements:

- Realistic simulation of driver behaviour;
- Multiple traffic agents simulated at the same time with different driving style.

### 5.1.2.3 V2X communication modelling

V2X (Vehicle-to-Everything) enable automobiles to interact and exchange information with the road infrastructure (through Roadside Units) and other road users (e.g., vehicles, motorbikes) and provide several benefits. For instance, through the early recognition of the intentions of other traffic participants, improved situational awareness and optimized driving behaviour for automated systems and human drivers can be achieved. Furthermore, 5G-V2X may enable high-added-value services, such as “Vehicle Platooning”, meaning group of cars travelling very closely together, that coordinate in terms of speed and driving behaviour, effectively reducing fuel consumption and polluting emissions, while simultaneously increasing traffic safety. V2X includes several subsets, such as vehicle-to-vehicle communication (V2V), vehicle-to-infrastructure (V2I), vehicle to network (V2N), and vehicle to pedestrian (V2P). V2V allows vehicles to communicate with one another. Vehicle to infrastructure (V2I) allows vehicles to communicate with external systems such as street lights, buildings, and even cyclists or pedestrians [139]. V2P (Vehicle-to-Pedestrian) establishes communication between a vehicle and a pedestrian or multiple pedestrian in close proximity.

V2V involves the exchange of wireless data transmissions between nearby vehicles with OBUs along with information such as vehicle speed, coordinated position, etc. Having received this information, the driver will have a clearer view about the surrounding environment and recognize in advance potential threats that may not even be in their line of sight [140]. In this way, vehicles can improve their awareness especially for unforeseen external events that can negatively impact their driving course and act accordingly, making the appropriate adjustments.

V2I communication, on the other hand, occurs between vehicles and RSUs [141]. [141]. Through V2I, RSUs transmit warnings related to red light and stop sign violations or even upcoming changes in speed limits [140]. Common use-cases for V2X applications include (but are not limited to): road safety (collision warning and collision avoidance), cooperative automated driving, infotainments services (e.g., traffic information services), green driving etc.[142]

Despite provided benefits (collaborating and coordinating driving and increased safety, eco-friendly driving through reducing emissions and fuel consumption, improved situational awareness etc.), V2X applications face great challenges concerning security breaches and privacy issues.

High Level Requirements:

- Realistic multi-modal traffic flow simulation at least at the microscopic level;
- Network Simulator;
- Communication interfaces;
- Modular architecture.

#### 5.1.2.4 Cybersecurity scenarios simulation

Another important component for automotive scenarios simulation in CPSoSAware project is realistic simulation of cyber-attacks both in network layer and directly on sensors. These attacks have to be simulated in 3D simulator directly or in V2X communication component.

High level requirements:

- Realistic simulation of cyber-attacks;
- Easily applicable to simulation ecosystem (as component);
- Configurable scenarios with variety of attacks

### 5.1.3 AV/ADAS Simulators

#### 5.1.3.1 Simulation tools

##### 5.1.3.1.1 Robotec Simulation [143]

Robotec Simulation is autonomous driving simulator developed by Robotec company. It is based on Unity engine and provides highly realistic visualization of the scene, high performance integration with ROS2 and simulation of multiple sensors.

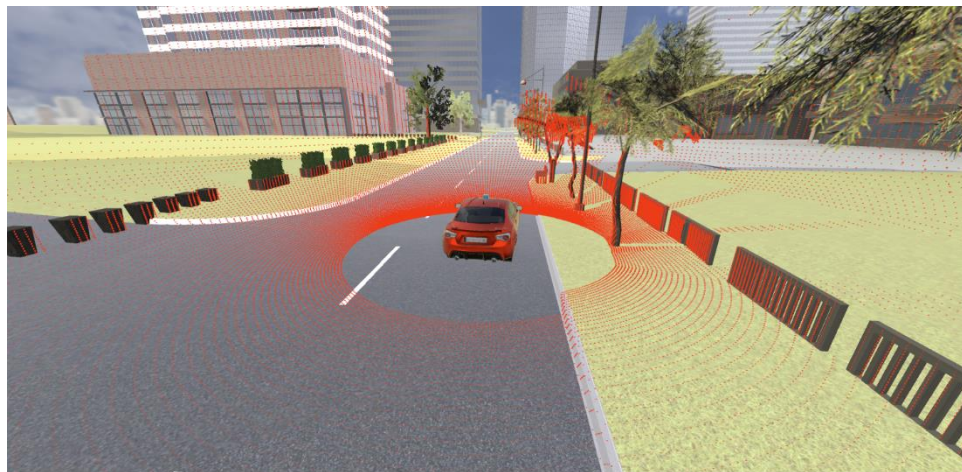


Figure 27 Robotec Simulation -view with LIDAR rays' projection

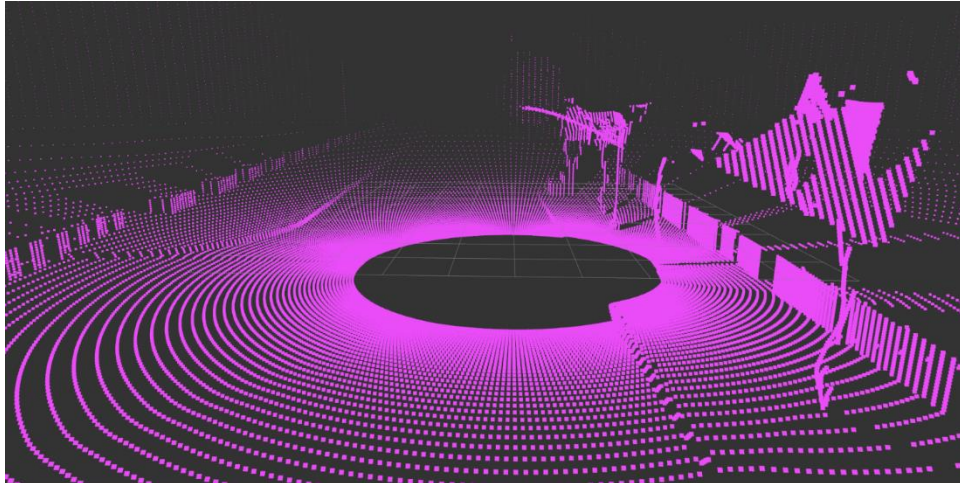


Figure 28 LIDAR point cloud in Robotic Simulation

#### 5.1.3.1.2 AirSim (Microsoft) [144]

AirSim is the open-source simulator based on Unreal Engine, supporting multiple vehicles (drones, cars etc.). It provides visually realistic simulation of the environment, simulation of variety of sensors, and hardware in the loop integration. AirSim was developed as a platform for AI research enabling experiments for deep learning, computer vision, and reinforcement learning in the field of autonomous vehicles.



Figure 29 Screenshot from AirSim simulator

#### 5.1.3.1.3 Carla [145]

Carla is one of the most popular simulators for autonomous driving research. It has a flexible API for all aspects of simulations (traffic generation, pedestrians' behaviour, sensors, etc.). Carla provides simulation of multiple sensors, hardware in the loop integration, and ROS interface. Of big advantages of CARLA simulator is the availability of high number of digital assets (urban layouts, vehicles, buildings) that can be used freely.



Figure 30 Screenshot from Carla simulator

#### 5.1.3.1.4 Apollo Auto [146]

Apollo Auto is an open-source platform that creates an entire ecosystem for autonomous driving. It consists of vehicle, hardware, software, cloud, and simulation. The simulation module provides all required features: sensors, photorealistic visualization, and a ROS interface for flexible control of the vehicle using external modules.



Figure 31 Screenshot from Apollo Auto Simulator

#### 5.1.3.1.5 Deepdrive [147]

Deepdrive is a simulator for self-driving car research based on Unreal Engine. The simulator provides sensors needed for training perception and localization algorithms, but its functionalities are more limited than in simulators described above, especially in the field of communication interfaces and extensibility.

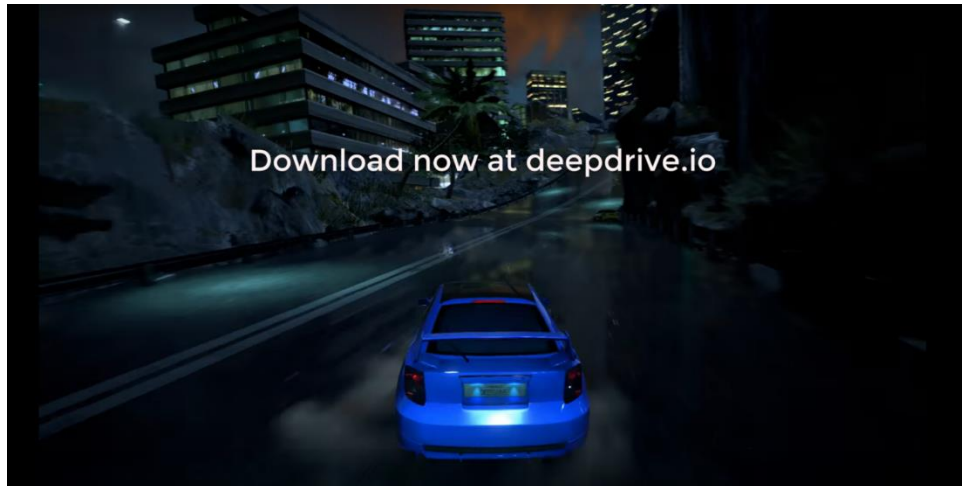


Figure 32 Screenshot from Deepdrive simulator

#### 5.1.3.1.6 LGSVL Simulator (LG) [148]

LGSVL is one of the most mature simulators in the comparison. It provides integration with the Apollo Auto platform and the Autoware platform, as well as communication interfaces with ROS/ROS2. The simulation is based on the Unity Engine and provides implementation of all sensors required in the research on autonomous driving.

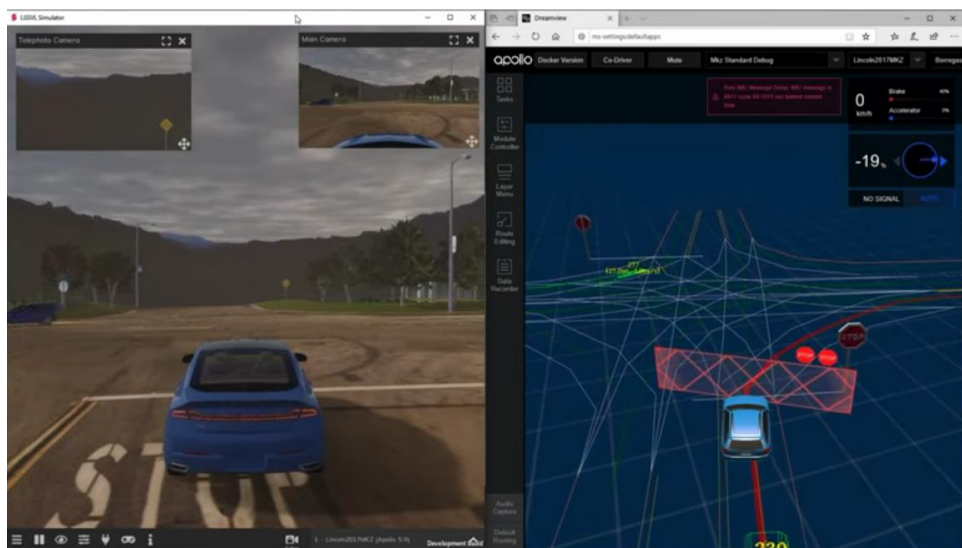


Figure 33 Screenshot from LGSVL Simulator

#### 5.1.3.1.7 Self-Driving Car Simulator (Udacity) [149]

This simulator was created for Udacity Self-Driving Car Nanodegree to teach students how to train models for autonomous driving. Its visualizations are much less realistic than in the other simulators, and the number of available sensors is limited and insufficient for complex projects like CPSoSAAware.



Figure 34 Screenshot from Udacity Self-Driving Car Simulator

#### 5.1.3.1.8 MADRaS [150]

MADRaS is a multi-agent simulator for autonomous driving built on top of TORCS. It enables parallel control over multiple vehicles, but the quality of visualization is lower than in most of the analysed simulators. Availability of other features required in the project is also limited.



Figure 35 Screenshot from MADRaS simulator

#### 5.1.3.2 General Comparison

Table 4 General comparison of simulation tools for autonomous driving

Name	Repo Link	Active	Stars	Demo Link	Licence
Robotec Simulation	Not open source	Y	-	-	-



<b>AirSim (Microsoft)</b>	<a href="https://github.com/Microsoft/AirSim">https://github.com/Microsoft/AirSim</a>	Y	~9.7k	<a href="https://www.youtube.com/watch?v=gz1X3UNM5Y&amp;feature=youtu.be">https://www.youtube.com/watch?v=gz1X3UNM5Y&amp;feature=youtu.be</a>	MIT
<b>Carla</b>	<a href="https://github.com/carla-simulator/carla">https://github.com/carla-simulator/carla</a>	Y	~4.2k	<a href="https://www.youtube.com/watch?v=TOojcfcRBA">https://www.youtube.com/watch?v=TOojcfcRBA</a>	MIT
<b>Apollo Auto</b>	<a href="https://github.com/ApolloAuto/apollo">https://github.com/ApolloAuto/apollo</a>	Y	~16k	<a href="https://www.youtube.com/watch?v=2Os-4TRCwFo">https://www.youtube.com/watch?v=2Os-4TRCwFo</a>	Apache 2.0
<b>Deepdrive</b>	<a href="https://github.com/deepdrive/deepdrive">https://github.com/deepdrive/deepdrive</a>	Y	~0.5k	<a href="https://www.youtube.com/watch?v=p4DbNFkQU78">https://www.youtube.com/watch?v=p4DbNFkQU78</a>	MIT
<b>LGSVL Simulator (LG)</b>	<a href="https://github.com/lgsvl/simulator">https://github.com/lgsvl/simulator</a>	Y	~0.8k	<a href="https://www.youtube.com/watch?v=YOCCh10Mlvw">https://www.youtube.com/watch?v=YOCCh10Mlvw</a>	-
<b>Self-Driving Car Simulator (Udacity)</b>	<a href="https://github.com/udacity/self-driving-car-sim">https://github.com/udacity/self-driving-car-sim</a>	Y	~3.1k	<a href="https://youtu.be/hTPADovdyfA">https://youtu.be/hTPADovdyfA</a>	MIT
<b>MADRaS</b>	<a href="https://github.com/madras-simulator/MADRaS">https://github.com/madras-simulator/MADRaS</a>	Y	~0.03k	<a href="https://www.youtube.com/watch?v=ZKzExvth3UE">https://www.youtube.com/watch?v=ZKzExvth3UE</a>	AGPL-3.0

### 5.1.3.3 Sensors

Table 5 Comparison of available sensors in simulation tools

Name	LIDAR	GNSS	Camera	RADAR	IMU	Vehicle data (CAN)
Robotec Simulation	X	X	X	-	X	X
AirSim (Microsoft)	X	X	X	-	X	X
Carla	X	X	X	X	X	X
Apollo Auto	X	X	X	X	X	X

Deepdrive	X	X	X	X	-	X
LGSVL Simulator (LG)	X	X	X	X	X	X
Self-Driving Car Simulator (Udacity)	-	-	X	-	-	X
MADRaS	-	-	X	X	X	X

#### 5.1.3.4 Control & Communication

Table 6 Comparison of simulators in terms of control and communication features

Name	ROS Interface	Multiple Agents	Hardware in the loop
Robotec Simulation	X	X	-
AirSim (Microsoft)	X	X	X
Carla	X	X	X
Apollo Auto	X	-	X
Deepdrive	-	-	-
LGSVL Simulator (LG)	X	X	X
Self-Driving Car Simulator (Udacity)	-	-	-
MADRaS	-	X	-

#### 5.1.3.5 Machine Learning support

Table 7. Comparison of simulators in terms of machine learning support

Name	Photorealistic visualization	Automatic Evaluation	Parallelization	Headless mode
Robotec Simulation	X	-	X	X
AirSim (Microsoft)	X	-	X	X
Carla	X	-	X	X
Apollo Auto	X	-	X	-
Deepdrive	X	-	-	-
LGSVL Simulator (LG)	X	-	X	X
Self-Driving Car Simulator (Udacity)	-	-	-	-
MADRaS	-	-	-	-

#### 5.1.4 Drivers behaviour modelling

##### 5.1.4.1 Parametric models

One of the approaches to model characteristics of driver behaviour is creating a parametric model covering multiple driving scenarios [151][152] such as:

- Free cruising model
- Car following model
- Lane changing behaviour
- Collision avoidance model

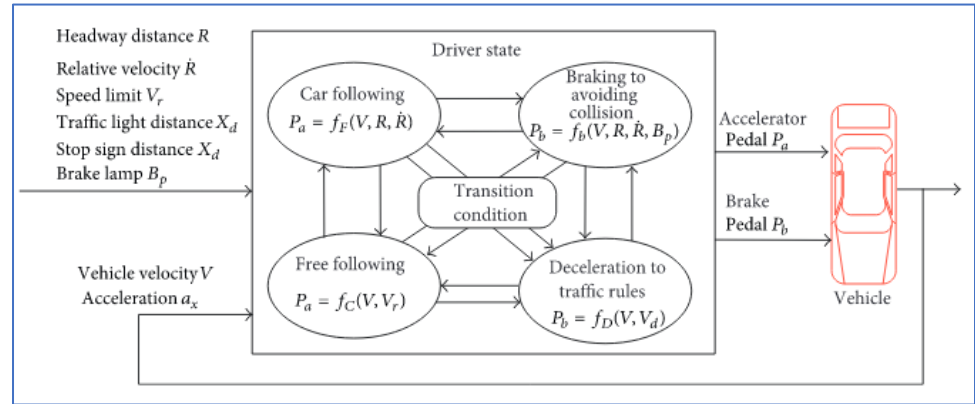


Figure 36 Diagram of parametric model for driving behaviour modelling [5]

Modelling of driver’s actions in various driving states relies on multiple parameters related to the ego vehicle and the surrounding traffic agents. To properly imitate complex traffic behaviour, transition conditions between all the states must be defined as well (Figure 36). Such models are complex to build, but thanks to many configurable parameters can be easily adapted to imitate different styles of driving.

#### 5.1.4.2 Machine Learning models

Another way for realistic imitation of driving behaviour is the machine-learning-based approach. Most popular machine learning methods for driver behaviour simulations are listed and briefly described below.

##### 5.1.4.2.1 Reinforcement learning [153]

Reinforcement learning approach assumes that driver’s actions on the road follows the policy which aims at maximizing global reward function. Such an approach, after many iterations of training, provides robust algorithm that can realistically imitate behaviour of the driver.

##### 5.1.4.2.2 Behavioural cloning

Behavioural cloning solves the regression problem in which the likelihood of performing actions occurring in the training set is maximized. This method has been successfully used for learning policy for simple scenarios (e.g., free following on the highway) but fails to generate general predictions of behaviour in more complex scenarios [154][155].

#### 5.1.4.3 Datasets

In the field of Driver Behaviour modelling, strong emphasis should be put on simulating human driver behaviour as closely and accurately as possible. For this purpose, data collected during real drives, containing sensor data crucial for driving style replication, should be analysed. Suitable datasets for Driver Behaviour Modelling are listed and described below.

##### 5.1.4.3.1 DBNet [156]

DBNet is a large-scale driving behaviour dataset recorded during 1000 km real-world driving. It includes all sensor data required for driving behaviour research:

- Camera video;
- LIDAR Point Cloud;
- GPS;
- Vehicle speed;
- Steering angle.

#### 5.1.4.3.2 Comma2k19 [157]

Comma2k19 is a dataset of over 33 hours of driving on California’s highway, created by comma.ai. It contains the following data:

- Road-facing camera video;
- GPS;
- 9-axis IMU;
- CAN data (GNSS, speed, pedals, steering angle, etc.).

#### 5.1.4.3.3 Honda Research Institute Driving Dataset [158]

HRI Driving Dataset is a dataset created to enable research on learning driver behaviour in real-life environment. Dataset includes 104 hours of real human driving in San Francisco Bay Area. It contains the following data:

- Camera video;
- LIDAR Point Cloud;
- CAN data (GNSS, speed, pedals, steering angle, etc.).

### 5.1.5 V2X Communication modelling

Simulating V2X scenarios involves incorporating various tools. In particular, a traffic simulator must be used to simulate the vehicular movements, create accurate urban mobility models and investigate real-world traffic problems. Additionally, a network simulator can be used for simulating the wireless communication among the vehicles involved in the scenario and build dynamic topologies between moving nodes. Furthermore, an application simulator provides the environment for simulating a V2X application [159][160] listed and briefly described below.

#### 5.1.5.1.1 SUMO



SUMO (Simulation of Urban Mobility) [161] an open-source traffic simulator tool. SUMO allows for modelling of intermodal traffic systems including road vehicles, public transport, and pedestrians offering a variety of features such as: microscopic simulation, online interaction, automatic generation of time schedules of traffic lights, etc. SUMO includes several supporting functionalities, which handle tasks such as route finding, visualisation, network import, and emission calculation. SUMO can be enhanced with custom models and provides various APIs to remotely control the simulation.

SUMO is widely used in V2X scenarios for generating mobility traces in different environments and can be integrated with other tools such as network simulators (NS-3, OMNeT++). Generated vehicular mobility traces from SUMO can include time stamped vehicles' coordinates, 2D speed and absolute heading information in a given scenario (for instance specifying the traffic network, surrounding environment, and vehicles density). These traces can be subsequently used by other simulation tools of the overall chain. The latter models (e.g., MATLAB) generate coherent sensor data traces accordingly (e.g., time stamped GNSS or LiDAR data, etc.), in coherence with the mobility traces generated by SUMO. Additionally, a Network simulator can provide communication related KPIs for instance latency for messages, packet reception ratios, etc.

#### 5.1.5.1.2 MATLAB Simulink



Simulink is integrated with MATLAB and is an environment suitable for multi-domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink [162] offers functionalities such as graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems.

#### 5.1.5.1.3 Vanetza

Omnet++ is a general-purpose discrete event simulator that is used to model telecommunication protocols. The research community has developed detailed V2X communication models for Omnet++ including IEEE 802.11p[163] and LTE [164].



Vanetza [165] is an open source ETSI G5 stack that can be used to generate V2X messaging including CAM and DENM messages.

Vanetza and Omnet++ can be used together to provide detailed packet level simulations of V2X scenarios. In addition, when connected with a mobility simulator like SUMO, they can create a full simulation stack for detailed V2X simulations including realistic mobility models, and protocol level simulations.

Currently, a variety of solutions are widely used for performing the aforementioned tasks. Nonetheless, it is possible to integrate (bidirectional coupling) a number of different simulation tools that can run in parallel and exchange real time data for implementing specific V2X scenarios. Such approaches listed and briefly described below.

#### 5.1.5.1.4 Veins

Veins [166][167] is an open source framework based on the simulators OMNeT++ and SUMO. OMNeT++ is a network simulator which includes a wide range of models for IVC, such as IEEE 802.11p and IEEE 1609.4 DSRC/WAVE, while SUMO is a widely used microscopic traffic simulator which implements realistic vehicle-following models and allows importing real-life map data.

#### 5.1.5.1.5 iTETRIS

iTETRIS [168][167] is an open-source modular architecture that integrates traffic and network simulator, with SUMO and NS-3 as an example. A complete model of the IVC networking stack (according to the ETSI standards) and the wireless channel are implemented. iTETRIS supports the implementation of IVC-based applications in a language agnostic fashion.

### 5.1.6 Cybersecurity scenarios simulation.

Attacks to V2X systems can be active or passive. In the former case the intruder actively interacts with the system. Examples of active attacks include false code/data injection, denial-of-service (DoS), alteration of transmitted data (e.g., GPS spoofing, broadcast/transaction tampering [169]), etc. On the other hand, passive attackers do not directly interact with the system. Eavesdropping is an example of passive attack.

Thus, securing V2X communicating platforms and ensuring the integrity and authenticity of the exchanged information is essential. Moreover, efficient attack detection is equally important.

Attacks to V2X systems can cause data loss, component failure, and severe damaging of the environment/infrastructures. Security is considered in various related standards, the most important currently being SAE J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems [170]. SAE J3061 describes a process framework that can support organizations towards the development of an internal process for designing and addressing cybersecurity into vehicle systems.

Moreover, towards the emergence of attack scenarios, a new standard, named ISO/SAE 21434, is currently being development [171]. This standard describes how security engineering should work in the automotive environment. It offers a relatively holistic view of the development and the life cycle of vehicles. Among other things, the standard focuses on the risk analysis and the development of secure concepts.

The table below provides an overview of the various types of potential cyber-attacks in V2X communications.

**Table 8 Potential cyber-attacks in V2X communications [172]-[180].**

ATTACK	EASE OF ATTACK	ACTIVE/PASSIVE	Detection probability	Description
--------	----------------	----------------	-----------------------	-------------

<b>Eavesdropping/Interception</b>	High	PASSIVE	Low	These attacks occur when intruders are able to gain access to vehicular messages.
<b>GPS spoofing</b>	High	ACTIVE	Low	An attacker can use GPS spoofing in order broadcast false signals (falsified location information) of higher strength than usual. Then the targeted vehicles may end up accepting these generated, fake, signals. Incorrect data reception could decrease message delivery efficiency by up to approximately 90% [181][182].
<b>Alteration/ Replay</b>	High	ACTIVE	Low	In this kind of attack , intruders continually broadcasts messages in order to impede the targeted vehicle's real-time functioning and to exploit the conditions at the time when the original message was sent [183][184]
<b>Electromagnetic pulse</b>	High	ACTIVE	Low-Driver, High-System	EMP [185] attack aiming to damage vehicle's electronic devices such as onboard sensors and processors (ECU).
<b>Location and identity tracking</b>	High	PASSIVE	Low-at High Traffic Density	The location at a specific moment or the route followed along a period of time can be used to trace the vehicle and gain information for the driver.[186]
<b>Sybil</b>	High	ACTIVE	Moderate	During this kind of attack, a Sybil attacker uses multiple false identities simultaneously to maliciously send erroneous information.
<b>Denial of Service</b>	High	ACTIVE	High	During a denial of service attack, an attacker can overwhelm the cluster head with excessive requests so that other vehicles cannot communicate with the cluster head [187]



<b>Timing</b>	High	ACTIVE	High	In this type of attack a malicious vehicle receives a message, alters it by adding some timeslots to the original message to create delay, and then transmits the message to other vehicles, thus leading to improper timing in-formation
<b>Bogus Information –flooding attacks</b>	Moderate	ACTIVE	Low-Driver, Moderate-System	Attackers generate bogus traffic information and make other vehicles to select different routes in order to free up the road for themselves [188]. Flooding attacks[182][184] can eventually make the network resources unavailable to legitimate users.
<b>Black hole</b>	Moderate	ACTIVE	Moderate	A malicious node uses its routing protocol in order to publicize itself for having the shortest route to the destination node. [189]
<b>Man in the middle (MITM)</b>	Moderate	PASSIVE / ACTIVE	Moderate	Man-in-the-middle is a type of eavesdropping attack that occurs when a malicious actor inserts himself as a relay/proxy into a communication session between people or systems. In a passive MITM attack attackers access the information in transit without trying to modify it. If attackers attempt to modify the information itself, they are committing an active MITM attack. [190]
<b>False data Injection</b>	Moderate	ACTIVE	Moderate-Driver, High-System	A malicious vehicle broadcasts fake traffic/safety messages or incorrect traffic estimation information to the traffic network with in an aim create disruption or trigger collision[182][184][191]
<b>Saturating/ Blinding attack on lidar</b>	Moderate	ACTIVE	High	Saturating renders the victim sensor unable to reflect the input

				signal changes. The victim systems can easily detect the attack but cannot prevent the sensor from saturating. [192]
<b>Illusion</b>	Low	ACTIVE	Low-Driver/System	During an illusion attack, attackers create false traffic events by altering vehicle sensor readings to trigger the sending of false traffic information messages
<b>Impersonation/Masquarading</b>	Low	ACTIVE	High	Impersonate attack attacker assumes the identity and privileges of an authorised node, either to make use of network resources that normally may not be available to it, or to disrupt the normal functioning of the network [193]

**5.1.7 Conclusions. Selection of tools for prototyping**

For the proper selection of the 3D simulation tool, all the requirements need to be considered. Simulators that can be eliminated from further research because of low quality of visualization and limited functionalities are MADRaS and Self Driving Car Simulator by Udacity. Analysis of remaining simulators shows that the most suitable tools in terms of available sensors, control and communication and machine learning support are:

- Carla;
- LGSVL Simulator;
- AirSim;
- Robotec Simulation.

In the field of driving behaviour modelling, both rule-based and machine learning models can be used in CPSoSAAware project. The biggest advantage of parametric models is the fact that different driving styles of traffic agents can be easily controlled by setting parameters. On the other hand, machine learning based methods are better in imitating real behaviour in general. Both solutions should be developed as components controlling the vehicle.

For V2X communication and cybersecurity network threats modelling there is another set of simulation tools available. OMNET++ based simulation seems to be good choice in CPSoSAAware autonomous vehicle scenario, as it can be integrated with traffic simulators like SUMO. To enable full simulation of vehicles with perception, control, and V2X communication, the integration of V2X/Cybersecurity simulator with photorealistic 3D simulator (CARLA, LGSVL etc.) has to be developed. In early phase of prototyping, V2X communication can be incorporated in 3D simulator in the offline way by playing the same scenario in both simulators. V2X result file can be shared to photorealistic simulator and used as another abstract sensor adding information about other, non-visible vehicles.

The overall tool for autonomous vehicle scenario will consist of following components:

- 3D simulator - realistic simulation tool with modular architecture. This component is the core of simulation environment for autonomous driving use cases. It is used for sensor data simulation and collection, scenarios definition and validation, integration of all other components into one ecosystem.
- Control component - component for controlling multiple vehicles independently with driver behaviour models included for more realistic behaviour of traffic agents. This component will be connected to 3D simulator through ROS/ROS2 interface.
- V2X simulator - simulation module for V2X communication. This module is used to pass the information between the vehicle and all other entities that may affect the vehicle (other vehicles, infrastructure, pedestrians, etc.).

Communication component between V2X and 3D simulator - component connecting V2X and 3D simulator is required to provide real time simulation for cooperative scenarios. Data collected in 3D simulator need to be shared between traffic agents to extend the perception of the vehicle and data received through V2X influence path planning process of the vehicle.

## 5.2 Human-Robot interaction scenario

### 5.2.1 Introduction

The second use case will be focused on HRC in the manufacturing environment and will involve trails that challenge the CPSoSAAware MODD concept and trails on accidents/failures, as well as cybersecurity attacks that challenge the collaborative control mechanism and the autonomic decentralized operation of the CPSoSAAware solution as well as the Design operation continuum support in the presence of cybersecurity attacks.

### 5.2.2 Description and Requirements

Simulators play an important role in industrial robotics, enabling faster design and verification of the manufacturing process and higher production quality. Simulation tools are used in robotics research to test the efficiency, safety and robustness of new algorithms [194]. Extensive testing of each implemented algorithm is crucial especially in scenarios that require close interaction of robots with humans.

High level Requirements:

- Photo-realistic representation of the environment
- Machine learning support
- Communication interfaces (ROS)
- Robotic arm model
- Possibility of modelling additional elements of the use case scenario: human, light curtain, safety eye, windshield carrier etc.
- Availability of sensors simulation: light curtain, safety eye, camera
- API for human behaviour customization in case of application of external model user

## 5.2.3 Industrial Robot simulation

### 5.2.3.1 Gazebo [195]

Gazebo is a 3D simulator for accurate and effective simulation of robots in complex indoor and outdoor environments. It provides realistic visualization with multiple available sensors and flexible API for simple development of custom plugins. Gazebo is the most popular simulator in the ROS community.

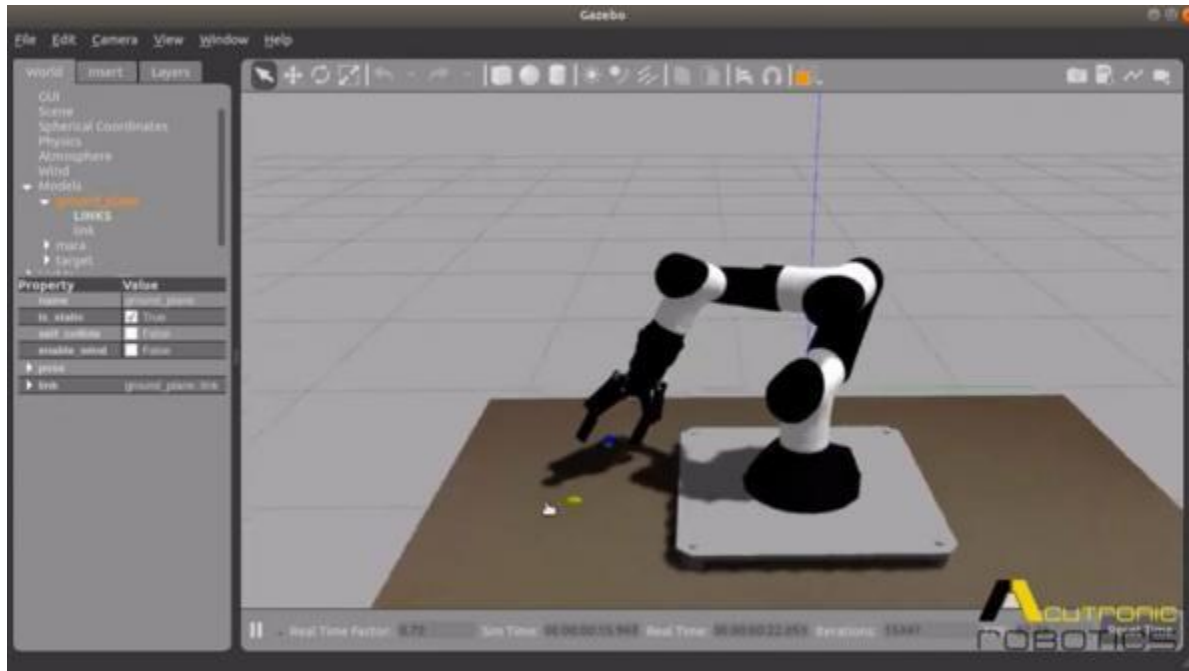


Figure 37 Screenshot from Gazebo simulator

### 5.2.3.2 NVIDIA Isaac [196]

NVIDIA Isaac is a high-performance robotics simulation for data generation as well as testing of perception and control algorithms. Thanks to the integration with NVIDIA Jetson, tested applications can be easily applied to physical robots. Simulation provides all required sensors and possibility of simulation of human agents, what is crucial for CPSoSAAware Human-Robot Interaction scenario.

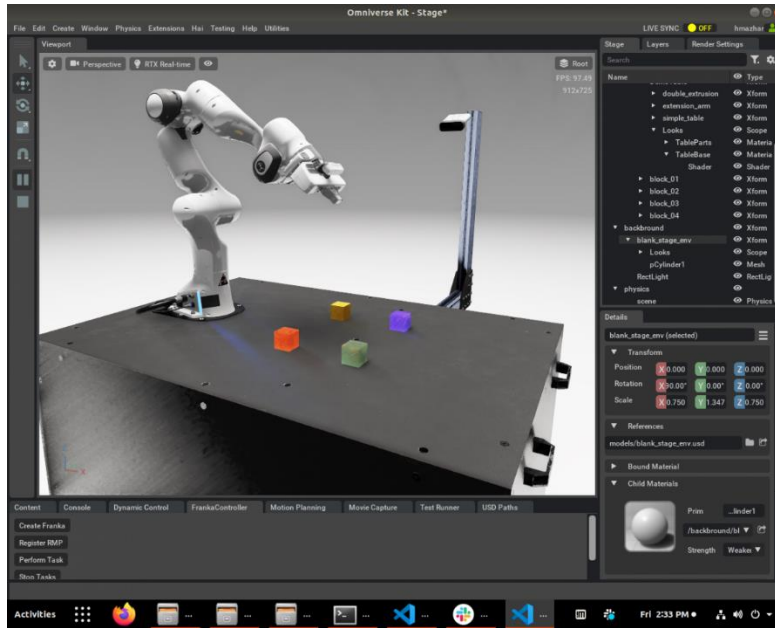


Figure 38 Screenshot from NVIDIA Isaac simulator

### 5.2.3.3 Webots [197]

Webots is an open-source, multi-platform robotic simulation application. It provides a complete development environment to model, program and simulate robots and is widely used in industry, education, and research. In addition to the source code, large Webots asset library is available which includes multiple robots, sensors, actuators, objects, and materials.

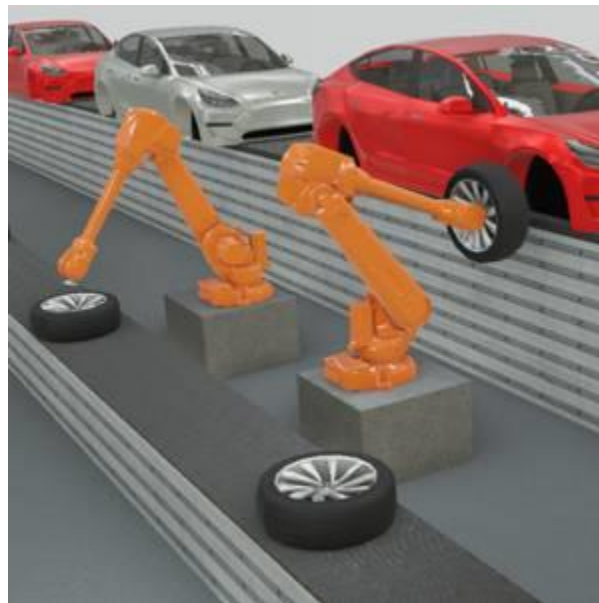


Figure 39 Screenshot from Webots simulator

#### 5.2.3.4 Coppeliasim (V-REP)

Coppeliasim is the robotic simulator widely used for fast algorithm development, factory automation simulations, fast prototyping and verification and robotic related education. It contains multiple features in fields of sensors, path/motion planning and advanced physics. Coppeliasim has also distributed control architecture – each object/model can be individually controlled with embedded script, plugin, ROS/BlueZero node or remote API client.

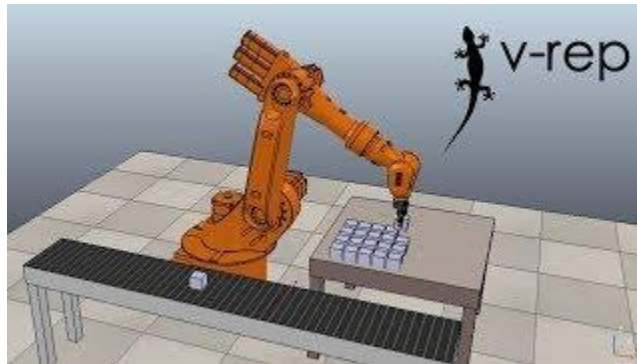


Figure 40 Screenshot from Coppeliasim simulator

#### 5.2.3.5 Process Simulate (Siemens PLM Tecnomatix) [199]

Process Simulate is a digital manufacturing solution for manufacturing process verification and simulation in 3D environment. It is a mature solution, widely used in industry for virtual validation, optimization, and commissioning of complex manufacturing processes, resulting in faster launch and higher production quality. It provides realistic simulation of manufacturing environment, simulation of all sensors required in CPSoSAAware scenarios and very realistic, easily controllable simulation of human movement.

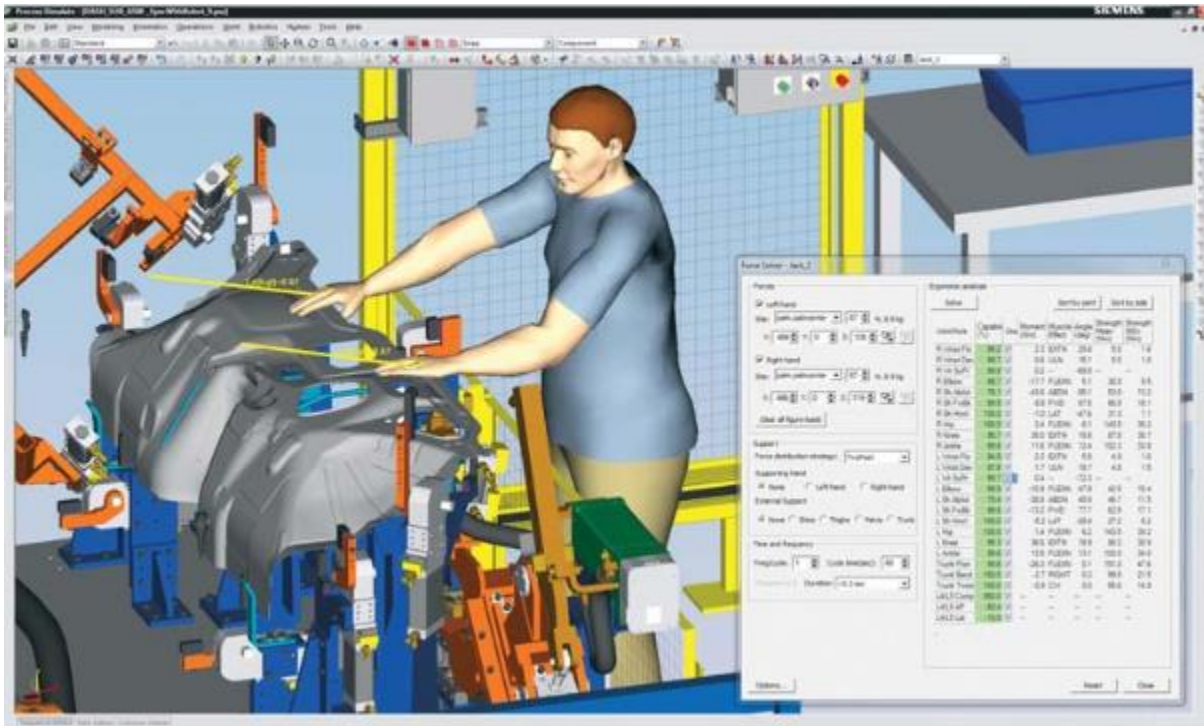


Figure 41 Screenshot from Process Simulate simulator

### 5.2.3.6 Comparison

Table 9 Comparison of features available in simulators analysed for Human-Robot Interaction scenario

	Gazebo	NVIDIA Isaac	Webots	CoppeliaSim	Process Simulate
Photorealistic visualization	X	X	X	X	X
Camera sensors	X	X	X	X	X
ROS Interface	X	X	X	X	-
Robotic model arm	X	X	X	X	X
Additional elements modelling	X	X	X	X	X

<b>Headless mode</b>	X	X	-	X	X
<b>API for human behaviour control</b>	X	X	-	X	X

#### 5.2.4 Human simulation

In Human-Robot Interaction scenario, an important part is the simulation of factory worker movement. Simulators like Gazebo, NVIDIA Isaac or Process Simulate, provide realistic modelling of pedestrian that can be used in analysed use cases. Simulation of human biomechanics and movement of workers in factory environment is crucial for development of reliable safety algorithms for Industrial Robot scenario. Different scenarios can be manually defined as set of paths and behaviours of worker and these scenarios can be used both for data collection and validation of algorithms.

Another possible use of realistic modelling of human movement is training models for gesture-based interface for Human-Robot Interaction [200]. Such solution with properly implemented interface can create more natural way of integration of human worker with the robot. Additionally, thanks to being contactless, it might have additional benefits in the field of safety of manufacturing systems.

#### 5.2.5 Conclusions. Suggested approach

Four of the analysed simulators (Gazebo, NVIDIA Isaac, CoppeliaSim, and Process Simulate) meet all the requirements for Industrial Robot simulation in the scenario considered in CPSoSAAware project. However, NVIDIA Isaac seems to have the most realistic visualization, what might be very important for vision-based algorithms training that have to be applicable in the real world. On the other hand, Process Simulate provides all required features and is already used by CRF in the factory in which CPSoSAAware trials will be conducted, what makes setup process much easier for this simulator compared to other tools analysed in this document. The overall steps to simulate Human-Robot Interaction scenario should consist of the following parts:

- Final selection of tool for simulation;
- Test scene design and creation;
- Test scenarios definition;
- Data collection for scene understanding models;
- Algorithms training;
- Full scenarios validation.

### 5.3 Architecture level Simulators

In the CPSoSAAware project, we also rely on open source architectural level simulators. Architectural level simulators allow us to simulate various systems with various configurations (i.e., core and memory system configurations). Most importantly, architecture level simulators offer the possibility to design and simulate various architectural level enhancements and verify their functional correctness as well as their timing characteristics. For example, the reliability aware techniques that are going to be developed as a part of WP3 will be first developed and studied in the architectural level simulators. In addition, architectural level



simulators typically include power and thermal models that allows to estimate the overall benefits of the proposed techniques. As a second step, the most promising techniques will be selected and ported to our FPGA-based platforms.

In general, the architecture-based simulators will be part of the Simulation and Training (SAT) Block of the CPSoSAware module. The SAT block constitutes the basic testing and training data extraction environment for the (re)design (reconfiguration) procedures of the project. We are planning to include both CPU-based and GPU-based architectural level simulations as part of the SAT block development activities of the project.

### 5.3.1 Architecture-level CPU Simulators

While various architecture level simulators are available, we are going to rely on gem5 simulator (briefly described in 4.1.2.3). The gem5 simulator is the most popular and widely used architecture simulator in academia and industry. It is object oriented and based on discrete-event model of computation. This feature allows us to connect the simulator with other simulators that will be part of the SAT block of the CPSoSAware project. It also provides modular and interchangeable computer architecture components such as CPUs, memories, buses, and interconnects. An important characteristic of gem5 that is critical in the context of CPSoSAware project is that gem5 is also flexible in terms of accuracy and simulation time providing levels of accuracy, such as more accurate but slower simulation models and faster but less accurate simulation models.

Simulators offer the possibility to perform Design space exploration (DSE) of complex embedded systems that combine a number of CPUs, dedicated hardware and software is a tedious task for which a broad range of approaches exists, from the use of high-level models to hardware prototyping. Each of these entails different simulation speed/accuracy trade-offs, and thereby enables exploring a certain subset of the design space in a given time. Some simulation frameworks devoted to CPU-centric systems have been developed over the past decade, that either feature near real-time simulation speed or moderate to high speed with quasi-cycle level accuracy, often by means of instruction-set simulators or binary translation techniques.

The basic criteria to characterize a full system simulator are:

- Accuracy
- Supported processor architectures
- Licensing
- Development activity.

#### 5.3.1.1 Simics

Simics is a functionally-accurate full-system simulator that enables unmodified target software (e.g. operating system, applications) to run on the virtual platform similar to the physical [201]. Simics supports a wide range of processor architectures (e.g. Alpha, ARM, MIPS, PowerPC, SPARC, x86), as well as operating systems (e.g. Linux, VxWorks, Solaris, FreeBSD, QNX, RTEMS). Simics is composed of an instruction-set simulator, memory management units' models, as well as all memories and devices found in the memory map of the processors. Simics has two main disadvantages, it is not claimed to be cycle-accurate and a commercial license is required.

### 5.3.1.2 PTLsim

Another simulator is PTLsim. PTLsim is a cycle accurate full system x86 microprocessor simulator that has an out of order pipelined model. PTLsim also supports modelling of multi-processor or simultaneous multithreading (SMT) machines[201]. PTLsim presents two main drawbacks, only x86 architectures are supported and the tool suite is not actively maintained anymore.

### 5.3.1.3 SimpleScalar

SimpleScalar is another open source infrastructure for simulation and architectural modelling. It supports several processor architectures including Alpha, ARM, PowerPC and x86. Moreover, it features a large range of CPU models, which varies from simple unpipelined processors to detailed dynamically scheduled microarchitectures with multiple-level memory hierarchies [203]. SimpleScalar features were widely improved in the past, but it seems that both development and support have slowed down significantly.

### 5.3.1.4 OVPSim

OVPSim is a dynamic linked library marketed by Imperas, which simulates complex multiprocessor platforms containing arbitrary local and shared-memory topologies [204]. An important feature of this simulator is the dynamic binary translation that improves simulation speed. OVPSim advantages are extensive documentation and excellent support for different processor architectures. However, OVPSim does not models cycle-accurate processors but rather instruction accurate processors.

As noted, in CPSoS Aware project, we are going to rely on the gem5 simulator. gem5 is a modular discrete event driven full-system simulator, under BSD license. This simulator supports different instruction set architectures, such as Alpha, ARM, x86, SPARC, PowerPC and MIPS [205]. Moreover, this simulator has an active development and support team.

### 5.3.1.5 Further investigation of the gem5 Simulator

Two different system modes are supported in gem5:

- System emulation (SE)
- Full system (FS) mode.

The SE emulates most operating system-level services through stubs on the simulation workstation, which include the Operating System services and devices, resulting in a significant simulation speedup at the cost of limited support for some functionalities such as multithreading. On the other hand, the FS mode performs complete system simulation, including the OS, thread scheduler and devices that runs on both user-level and kernel-level instructions, making the simulation accuracy, penalizing the simulation time.

Gem5 supports four different CPU models:

- AtomicSimple
- TimingSimple
- In-Order
- Out-Of-Order (O3)

These differ in speed/accuracy trade-offs. AtomicSimple is the simplest scalar one cycle-per instruction/ideal memory model, while TimingSimple is also a non-pipelined CPU model, but uses the reference memory timing for modelling memory access latencies. In-order and O3 are pipelined models. O3 models inter-instruction dependencies for out-of-order/superscalar, simultaneous multithreading (SMT) CPUs.

Gem5 supports a variety of platforms in a transparent manner, which makes it very modular and easy to switch CPU model. As part of the CPSoSAAware project we are going to setup gem5 for specific ARM based models and in particular for the Cortex-A9 core (ARMv7 A-profile ISA) including support for Thumb, Thumb-2, VFPv3 and NEON instruction set extensions.

Moreover, gem5 provides two different memory system modes. The simplest is the Classic model which provides a fast and easily configurable memory system, whereas Ruby model focuses on accuracy and support for various cache coherence protocols. The figure below (Figure 42) shows the configuration trade-offs between speed and accuracy. In addition, an overview of each architecture and the current supported features can be found on the official website [206].

Processor		Memory System		
CPU Model	System Mode	Classic	Ruby	
			Simple	Garnet
Atomic Simple	SE	Speed		
	FS			
Timing Simple	SE			
	FS			
InOrder	SE			
	FS			
O3	SE			
	FS			Accuracy

Figure 42 Configuration trade-off between speed and accuracy (based on [207])

### 5.3.1.6 GPU-based Simulation

One additional feature of gem5 simulator is that it also includes support for simulating GPUs. The GPU model of gem5 is based on the popular GPGPU-Sim simulator [208]. GPGPU -sim is a detailed general-purpose GPU (GPGPU) simulator which models GPGPU compute units (CUs) — called streaming multiprocessors by NVIDIA — and the GPU memory system. Gem5-gpu builds on ideas used in related CPU-GPU simulators but makes different design choices. It captures interactions with execution-driven simulation rather than well-partitioned trace-driven simulation, e.g., MacSim [209]. It captures interactions with execution-driven simulation rather than well-partitioned trace-driven simulation, e.g., MacSim. It uses a more detailed — therefore slower — GPU component than MV5 [210] and does not rely on the deprecated m5 simulator. It supports more flexible memory hierarchy and coherence protocols than Multi2Sim [211] or FusionSim [212] at a possible increase in simulation time.

Gem5-gpu is the only simulator with all the following advantages:

- Detailed cache coherence model
- Full-system simulation
- Checkpointing
- Tightly integrated with the latest gem5 simulator
- Increased extensibility of GPGPU programming model and entire system architecture.

By integrating GPGPU-Sim's CU model into gem5, gem5-gpu can capture interactions between a CPU and a GPU in a heterogeneous processor. In particular, GPGPU-Sim CU memory accesses flow through gem5's Ruby memory system, which enables a wide array of heterogeneous cache hierarchies and coherence protocols.

In general, GPU (GPGPU) computing is the practice of offloading computation to run on programmable GPUs. Applications commonly targeted to GPGPU computing include data-parallel image processing, scientific, and numerical algorithms, though there is a trend toward more irregularly parallel workloads, such as graph analysis. Work units offloaded to the GPU are called kernels. Kernels can be structured to execute thousands of threads on the GPU in a single-instruction, multiple-thread (SIMT) fashion. In systems with separate CPU and GPU address spaces, such as discrete GPUs, data is explicitly copied between the GPU address space and the CPU address space.

Writing an application to take advantage of a GPU requires user-level calls to a GPGPU application programming interface (runtime), and this runtime interfaces with a kernel-level driver that controls the GPU device. Currently, the most popular GPGPU runtimes are CUDA and OpenCL.

GPGPU-Sim is a detailed GPGPU simulator ([gpgpu-sim.org](http://gpgpu-sim.org)). It models the compute architecture of modern NVIDIA graphics cards. GPGPU -sim executes applications compiled to PTX (NVIDIA's intermediate instruction set) or disassembled native GPU machine code. GPGPU-Sim models the functional and timing portions of the compute pipeline including the thread scheduling logic, highly banked register file, special function units, and memory system. GPGPU-Sim includes models for all types of GPU memory as well as caches and DRAM.

GPGPU-Sim applications can access a multitude of memory types. Global memory is the main data store where most data resides, similar to the heap in CPU applications. It is accessed with virtual addresses and is cached on chip. Other GPU-specific memory types include constant, used to handle GPU read-only data; scratchpad, a software-managed, explicitly addressed and low-latency in-core cache; local, mostly used for spilling registers; parameter, used to store compute kernel parameters; instruction, used to store the kernel's instructions; and texture, a graphics-specific, explicitly addressed cache. GPGPU-Sim consumes mostly unmodified GPGPU source code that is linked to GPGPU-Sim custom GPGPU runtime library. The modified runtime library intercepts all GPGPU-specific function calls and emulates their effects. When a compute kernel is launched, the GPGPU-Sim runtime library initializes the simulator and executes the kernel in timing simulation. The main simulation loop continues executing until the kernel has completed before returning control from the runtime library call.

GPGPU-Sim is a functional-first simulator; it first functionally executes all instructions, then feeds them into the timing simulator. GPGPU-Sim has some limitations when modelling heterogeneous systems:

- No host CPU timing model,
- No timing model for host-device copies,

- Rigid cache model,
- No way to model host-device interactions.

Because of these limitations, researchers interested in exploring a hybrid CPU-GPU chip as a heterogeneous compute platform cannot rely on GPGPU-Sim alone.

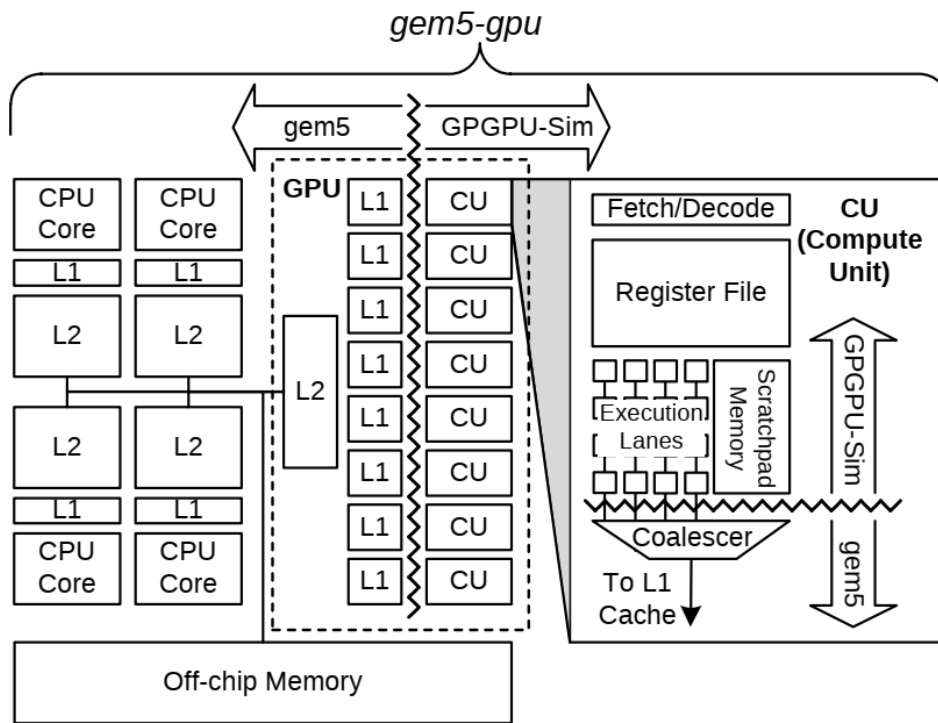


Figure 43 Overview of *gem5-gpu* architecture with an example configuration.

The figure above (Figure 43) shows one example architecture *gem5-gpu* can simulate: a four core CPU and an eight CU GPU integrated on the same chip. The number of CPUs, CUs, and topology connecting them is fully configurable. Two on-chip topologies that *gem5-gpu* provides out of the box are a shared and a split memory hierarchy (i.e., integrated and discrete GPUs, respectively).

Many CUs make up the GPU, each of which has fetch/decode logic, a large register file, and many (usually 32 or 64) execution lanes. When accessing global memory, each lane sends its address to the coalescer, which merges memory accesses to the same cache block. The GPU may also contain a cache hierarchy that stores data from recent global memory accesses.

In order to have a clean interface between *gem5* and GPGPU-Sim, the *gem5-gpu* includes a single pseudo-instruction to *gem5* to facilitate calls into the simulator for DMA engine and GPU functionality. Then, *gem5-gpu* routes general-purpose memory instructions—accesses to the global address space— from GPGPU-Sim to Ruby through *gem5*'s port interface.

Finally, it should be noted that in order to be able to run OpenCL-based kernels on GPGPU-Sim, *gem5-gpu* offers an emulated CL runtime system [213] and an OpenCL compiler [214].

## 5.4 Intra-communication Simulation Frameworks and Models

In the context of CPSoSAAware intra-communication simulation scenarios the main objective is to utilize Simulation Environments that are well-known, as well as widely utilized and accepted both by the research and industrial community. Additionally they are required to offer open, flexible and extensible frameworks (simulation environments) as well as models for multiple communication technologies/protocols that, on one hand, are suitable for the CPSoSAAware intra-communication requirements and, on the other hand, facilitate the integration collaboration with the rest of the CPSoSAAware components in general but more specifically the CPSoSAAware Simulation and Training architectural component. Therefore, critical characteristics of the selected Simulation Environments are mentioned below.

### 5.4.1 Critical characteristics of the selected Simulation Environments

#### 5.4.1.1 *Open Source Approach Support*

As CPSoSAAware is primarily a research project, the open-source approach is highly appreciated in all aspects of it. Particularly with respect to the intra-communication simulation environment, offering accessible standards/protocols, well-defined APIs, and even access to the internal functionality of respective communication protocols and mechanisms also comprise a valuable requirement. Without a doubt the degree by which aforementioned features are offered by a specific simulation framework can greatly influence the configurability and extensibility of the respective solution offered in the context of CPSoSAAware. Also, availability of adequate development/debugging and performance measuring capabilities play an important role, since they can help reduce the time required from designing a solution to actually developing it and testing in the environment drastically.

#### 5.4.1.2 *Energy Model Support*

In intra-communication scenarios the ability of a wireless communication to promote power consumption minimization comprises a characteristic of paramount importance. Meeting such objective is a multifaceted task involving many critical aspects of a wireless communication platform. Starting with a physical medium, the respective technology must support low power transmission, receive, idle, and sleep operation modes (e.g., typically transmission power in nowadays WSN technologies do not surpass 0dBm but can go as low as -30dBm). Consequently, supporting multiple operation state power consumption modelling and respective network wide energy consumption measuring capabilities is critical for intra-communication scenarios.

#### 5.4.1.3 *Network topology flexibility*

Unpredictable, rapid, and dynamic topology changes are an inherent characteristic of WSN networks. Even more, simulation support of such conditions is an important advantage of simulation frameworks. Therefore, it is a critical requirement for a respective simulation environment to support or/and give the tools to develop flexible network and mobility patterns. This capability is critical since, on the one hand, unpredictable topology changes lead to increased/decreased of network congestion in particular areas of the network and in general changes in the communication conditions that the technology must take into consideration and adjust adequately. On the other hand, unpredictable topology supports cases where a node can't communicate directly with the intended receiver and thus intermediate nodes act as relays to forwarding data packets towards the final receiver. Therefore, the simulation framework must allow for dynamically changing node topology and routes.

#### 5.4.1.4 Support of adequate Communication technologies

Since CPSoSAAware aims to offer a platform of an extended and evolving life cycle, it is beneficial to rely on communication technologies that have a significant footprint in both industry and academia. Having a substantial impact and acceptance in industry through offering models for respective commercial solutions or/and upcoming research-based solutions comprises a critical advantage for any simulation environment.

#### 5.4.2 CPSoSAAware Intra-communication Network Simulators

Based on the main requirements as indicated in the previous section, a presentation of adequate Simulation Environments that will be further explored and utilized in the context of intra-communication scenarios as well as integration with inter-communication simulation environment is offered.

##### 5.4.2.1 Omnet++

OMNeT++ is an object-oriented, modular, discrete event network simulation framework. It has a generic architecture, so it can be used in various domains for modelling, communication, and queuing networks with various protocols. OMNeT++ itself is not a simulator of anything concrete, but rather provides infrastructure and tools for writing simulations. One of the fundamental ingredients of this infrastructure is the component architecture for simulation models. OMNeT++ users describe the structure of the simulation models in the NED (Network Description) language, which lets the users to declare simple modules, connect them and assemble them into compound modules. OMNeT++ models consist of modules, written in C++ programming language and communicate with message passing. Messages can be sent either via connections that span modules or directly to other modules.

##### 5.4.2.1.1 Conceptual Overview

Figure 44 below, represents a basic simulation network on OMNeT++. A network consists from simple modules (grey background) and compound modules, which connected through connections and gates.

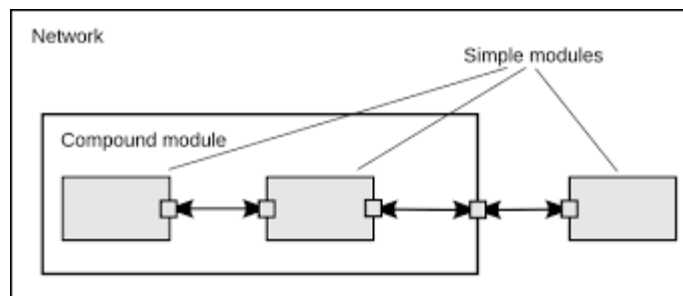


Figure 44 OMNeT++ Basic Simulation Network.

Both simple and compound modules are instances of module types, where simple modules are combined in order to initiate a compound module. Modules communicate by exchanging messages, which can represent frames or packets in a computer network and can also contain arbitrarily complex data structures. Simple modules can send messages either directly to their destination or along a predefined path, through gates and connections. Gates are the input and output interfaces of modules, where messages sent out through output gates and arrive through input gates. Each connection (also called link)

is created within a single level of the module hierarchy: within a compound module, one can connect the corresponding gates of two submodules, or a gate of one submodule and a gate of the compound module.

#### 5.4.2.1.2 INET Framework

INET Framework is an open-source model library for the OMNeT++ simulation environment. It provides protocols, agents and other models to work with communication networks. INET supports a wide class of communication networks, including wired, wireless, mobile, ad-hoc and sensor networks. It contains models for the Internet stack (TCP, UDP, IPv4, IPv6, etc.), link layer protocols (Ethernet, PPP, IEEE 802.11, various sensor MAC protocols, etc.), refined support for the wireless physical layer, MANET routing protocols, DiffServ, several application models, and many other protocols and components.

#### 5.4.2.1.3 Building Network Topologies

In OMNeT++ a communication network can be defined using the NED language. A network consists of nodes, where on each node there is an application running which generates packets at random intervals. A sample network definition can be shown below:

```
network Network
{
submodules:
node1: Node;
node2: Node;
node3: Node;
...
connections:
node1.port++ <--> {datarate=100Mbps;} <--> node2.port++;
node2.port++ <--> {datarate=100Mbps;} <--> node4.port++;
node4.port++ <--> {datarate=100Mbps;} <--> node6.port++;
...
}
```

#### 5.4.2.1.4 Models

##### 5.4.2.1.4.1 IEEE 802.15.4

IEEE 802.15.4 is a technical standard which defines the operation of low-rate wireless personal area networks (LR-WPANS). IEEE 802.15.4 was designed for data rates of 250 kbit/s or lower, in order to achieve



long battery life (months or even years) and very low complexity. IEEE 802.15.4 is the basis for the ZigBee, ISA100.11a, WirelessHART, MiWi, SNAP, and the Thread specifications, each of which further extends the standard by developing the upper layers which are not defined in IEEE 802.15.4. The INET Framework contains a basic implementation of IEEE 802.15.4 protocol.

#### **5.4.2.1.4.2 Wifi**

IEEE 802.11 a.k.a. WiFi is the most widely used and universal wireless networking standard. In INET, nodes become “WiFi-enabled” by adding IEEE80211Interface to them. APs are represented with the AccessPoint node type. WiFi networks require a matching transmission medium module to be present in the network.

#### **5.4.2.1.4.3 Mobility**

In order to simulate ad-hoc wireless networks, it is important to model the motion of mobile network nodes. Received signal strength, signal interference, and channel occupancy depend on the distances between nodes. The selected mobility models can significantly influence the results of the simulation (e.g., via packet loss rates). Mobility models provided by OMNeT++ can be single or group mobility models. Single mobility models describe the motion of entities independent of each other. Group mobility models provide such a motion where group members are dependent on each other. Mobility models can also be categorized as trace-based, deterministic, stochastic, and combining models.

#### **5.4.2.1.4.4 Energy**

Modeling power consumption becomes more and more important with the increasing number of embedded devices and the upcoming Internet of Things. High-fidelity simulation of power consumption allows designing power-sensitive routing protocols, MAC protocols with power management features, etc., which in turn results in more energy efficient devices. OMNeT++ provides power models, though INET Framework, as a separated module, with extensibility in mind. INET power model consists of the following components:

- energy consumption models,
- energy generation models,
- temporary energy storage models.

The power model elements fall into two categories, abbreviated with Ep and Cc as part of their names:

- Ep models are simpler, and deal with energy and power quantities;
- Cc models are more realistic, and deal with charge, current, and voltage quantities.

### **5.4.2.2 NS-3 Network Simulator**

NS-3 is a discrete event network simulator, developed to provide an open, extensible network simulation platform for networking research and education. The NS-3 software infrastructure encourages the development of simulation models which are realistic, allowing NS-3 to be used as a real time network simulator, with real world interconnection. One of the key features of NS-3 is that designed as a set of libraries that can be combined with external software libraries, such as Linux OS libraries. Finally, the development of models in NS-3 made with C++ and Python scripts.

#### 5.4.2.2.1 Conceptual Overview

Figure 45, below, shows a basic simulation model in NS-3.

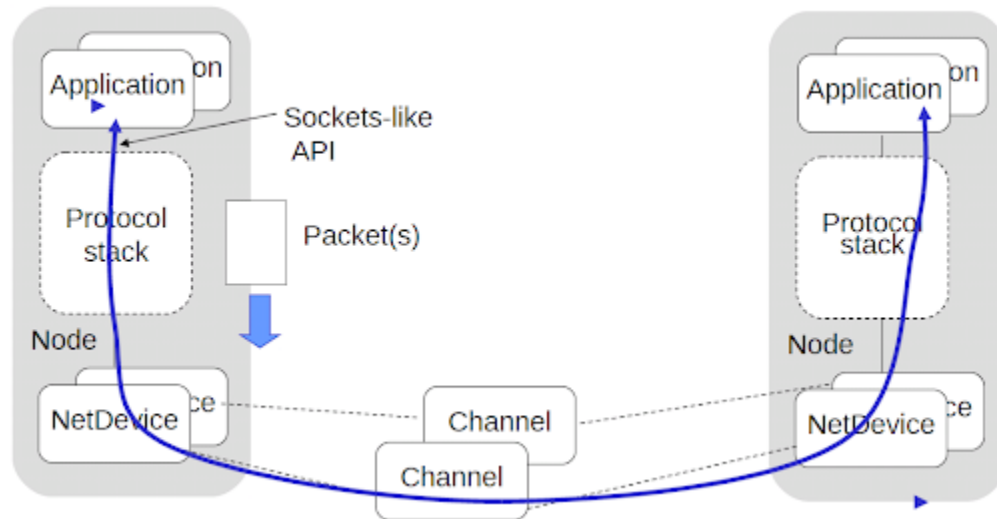


Figure 45 NS-3 Basic Simulation Model.

#### 5.4.2.2.2 Node

In NS-3 the basic computing device abstraction is called the **Node**. Node class provides methods for managing the representations of computing devices in simulations, with added functionality. NS-3 node is like applications, protocol stacks and peripheral cards with their associated drivers to enable the computer to do useful work.

#### 5.4.2.2.3 Application

In NS-3 the basic abstraction for a user program that generates some activity to be simulated is the application, represented by the Application class. The Application class provides methods for managing the representations of user-level applications in simulations.

#### 5.4.2.2.4 Net Device

In NS-3, the net-device abstraction, represents a peripheral hardware device. Net-Devices can be attached in a node, allowing node to communicate with each other in the simulation via Channels. The net-device abstraction covers both software driver and the simulated hardware. Finally, a node may be connected to more than one Channel via multiple NetDevices.

#### 5.4.2.2.5 Channel

Channels in NS-3 represents basic communication. The Channel class provides methods for managing communication subnetwork objects and connecting nodes. A Channel may model something simple wire network or more complicated networks like large Ethernet switches, or three-dimensional space full of obstructions in the case of wireless networks.

#### 5.4.2.2.6 Building Network Topologies

Network topologies in NS-3 can be built by reading traces from topology mapping engines or a user can create a network generator from NS-3 simulator. Known topology traces supported by NS-3 are:

- Orbis 0.7 traces,
- Inet 3.0 traces,
- Rocketfuel traces.

#### 5.4.2.2.7 Models

##### 5.4.2.2.7.1 LR-WPAN (802.15.4)

NS-3 models for the low-rate, wireless personal area network (LR-WPAN) as specified by IEEE standard 802.15.4. The model design closely follows the standard from an architectural standpoint.

##### 5.4.2.2.7.2 Wifi

NS-3 provides models for Wifi, using the WifiNetDevic, allowing to create models of 802.11-based infrastructure and ad hoc networks. The set of 802.11 models provided in NS-3 attempts to provide an accurate MAC-level implementation of the 802.11 specification and to provide a packet-level abstraction of the PHY-level for different PHYs, corresponding to 802.11a/b/e/g/n/ac/ax specifications.

The implementation provides three sublayers of models:

- the PHY layer models;
- the so-called MAC low models: which models functions such as medium access (DCF and EDCA), RTS/CTS and ACK;
- the so-called MAC high models: which implements non-time-critical processes in WIFI such as the MAC-level beacon generation, probing, and association state machines, and a set of Rate control algorithms.

##### 5.4.2.2.7.3 Mobility

NS-3 support mobility models, and includes:

- set of mobility models which are used to track and maintain the current Cartesian position and speed of an object. NS-3 uses only the Cartesian coordinate system;
- a “course change notifier” trace source which can be used to register listeners to the course changes of a mobility model;
- a number of helper classes which are used to place nodes and setup mobility models (including parsers for some mobility definition formats).

##### 5.4.2.2.7.4 Energy

Energy consumption is a key issue for wireless devices, and wireless network researchers often need to investigate the energy consumption at a node or in the overall network while running network simulations

in NS-3. The NS-3 Energy Framework provides the basis for energy consumption, energy source and energy harvesting modelling. The NS-3 Energy Framework is composed of three parts, listed below:

**Energy Source:** The Energy Source represents the power supply on each node. A node can have one or more energy sources, and each energy source can be connected to multiple device energy models.

**Device Energy model:** The Device Energy Model is the energy consumption model of a device installed on the node. It is designed to be a state-based model where each device is assumed to have a number of states, and each state is associated with a power consumption value.

**Energy Harvester:** The energy harvester represents the elements that harvest energy from the environment and recharge the Energy Source to which it is connected. The energy harvester includes the complete implementation of the actual energy harvesting device (e.g., a solar panel) and the environment (e.g., the solar radiation).

## 6 Techniques, tools and best practices for the operation of CPSs

The CPS domain has transitioned from individual systems to a collection of collaborative systems. Centralized CPSoS have demonstrated the need for elaborate modelling of the relations between the subsystems, highlighting the need for an approach beyond traditional control and management center. Specifically, having a centralized authority that handles all CPSoS processes, subsystems and control loops leads to extremely complex control and management routines. Decentralization of CPSoS processes and overall functionality by appointing tasks to individual CPSs within the System of Systems can be a reasonable solution. However, the collaborative mechanism between CPSs is an active research area since appropriate tools and methodologies are needed in order to assess whether the CPSoS operates as it should be and that the CPSoS remains resilient, safe and efficient. CPSoS consist of various, autonomous CPSs. CPSs, in general, are self-organized and, in several occasions, they may have conflicting goals, thus competing to get access to shared resources (i.e. autonomous driving vehicles utilize driving space in a highway, collaborative robots access the same restricted amount of materials). From a CPSoS perspective, all actors must collaborate to achieve overall CPSoS efficiency goals.

The rest of Section 6 is organized as follows. Subsection 6.1 focuses on AI solutions for individual CPSs. Subsection 6.2 analyses deep priors driven approaches for scene understanding. Subsection 6.3 presents compression and acceleration approaches for deep architectures. Subsection 6.4 is dedicated to multi-modal localization methods for connected and autonomous vehicles. Subsection 6.5 investigates distributed frameworks for cyber-physical modelling. Subsection 6.6 presents communication protocols, while Subsection 6.7 focuses on AR tools to facilitate situational awareness of the human in the loop.

### 6.1 Individual CPSs AI solutions

Fault detection and handling of errors or abnormal behaviours is a key issue in CPSoS design and operation. Due to the large scale and the complexity of CPSoS, failures occur all the time. The average system performance, as well as the degree of satisfaction of the users, is strongly affected by the impact of unforeseen events and outer influences that require non-continuous actions and cannot be compensated on the lower system levels. There is a massive need for detecting such situations quickly and, if possible, preventing them, and for fail-soft mechanisms and resiliency and fault tolerance at the systems level. The handling of faults and abnormal behaviour is challenging from a systems design point of view. In many cases, it cannot be done optimally by a design based on a separation of concerns but requires a trans-layer design of the reaction to such events.

CPSoS are operated and continuously improved over long periods of time. New functionalities or improved performances have to be realized with limited changes in many parts of the overall system. Components are modified and added, the scope of the system may be extended, or its specifications may be changed. Thus, engineering to a large extent has to be performed at runtime. Additions and modifications of system components are much facilitated by plug-and-play capabilities of components that are equipped with their own management and control systems (decentralized intelligence).

#### 6.1.1 Benefits of distributed machine learning in coalition environments

Distributed machine learning techniques do not require the transmission of all local data to a central location. Thereby, it has the benefits of reducing communication bandwidth consumption. Additionally, it reduces the risk of unwillingly disclosing sensitive information to other entities. Distributed machine learning is also able to make use of the computation and storage capability of multiple nodes, improving

the performance compared to the centralized setting in resource-scarce environments, such as in emerging systems of mobile edge computing, and Internet-of-Things [215].

### 6.1.2 Coalition in CPSoS

A CPSoS contains multiple autonomous systems (multi-agent systems) that may have different preferences, goals, beliefs, and capabilities. One of the main objectives that the multi-agent systems have is to build agents that can take joint, coordinated actions in order to improve their performance or to achieve goals that are beyond the capabilities of individual agents.

This type of interaction is useful both in cases where the agents are cooperative or self-isolated. In the former case, the goal of these agents is to maximize some overarching system-wide objective. In contrast, in the latter, each agent acts in its own best interests, regardless of the consequences on other agents.

The formation of coalitions is a structure where groups of agents typically exhibit the following characteristics:

- They are goal-directed.
- Coordination can occur between members of the same coalition, but not among members of different coalitions.
- Regarding the organization of the structure, the coalition is usually flat (non-hierarchical).

The activities that the coalition-formation process can involve are:

- To form the coalition structure. In this step, each agent joins a coalition. Typically, we are interested in coalition structures that maximize a global function or minimize the agents' incentive to deviate from their coalitions.
- To solve the optimization problem of each coalition. In this step, we estimate the activities of the members of a coalition that maximize the performance of the coalition.
- To divide the reward of each coalition among its members. Typically, the goal is to be satisfied with specific desirable criteria, such as fairness, or stability [217].

### 6.1.3 Coalition in a cooperative road infrastructure system

In the next figure, we present an example of a cooperative road infrastructure system that aims at supporting drivers to avoid overtake-related accidents on undivided rural roads, where road-surface-based units, vehicles, and other road infrastructure cooperate to solve this problem.

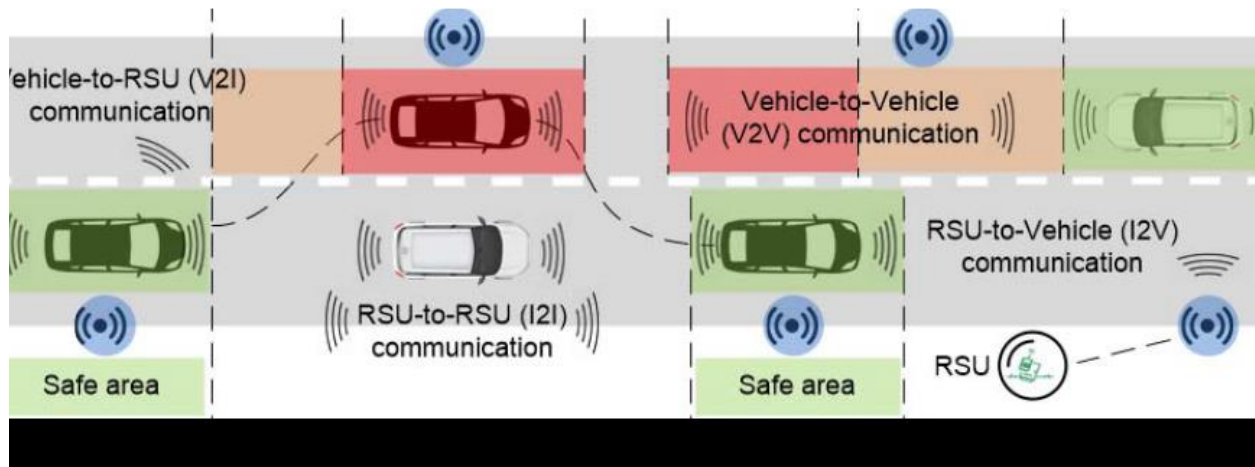


Figure 46 A diagram of the cooperative overtaking assistance system with the critical zones

The driver has the ability to be aware of the operational environment (road situation) if there are no obstacles limiting his/her visual capabilities (e.g., heavy fog, a sharp curve). In the case of heavy fog or a sharp curve, a driver may need to depend on RMUs for such knowledge since he/she does not have the self-capability for acquiring it. As the CPS may be dependent on such knowledge, we cannot say that the CPS is fully autonomous for performing its activity, thus, its autonomy should be adjusted (e.g., limited, restricted).

In order to coordinate their activities, CPSs that operate in the same environment need to be aware of one another. In other words, awareness is essential for any coordination since it is about understanding the activities, locations, and situations of other CPSs, which provides the required knowledge for each CPS to safely perform its own activities[216]

#### 6.1.4 Coalition Clustering Communication in Vehicular CPSoS

In a CPSoS, various sensors are used to collect and transmit the sensing data to achieve intelligent control. However, the communication models, schemes, and strategies, which are generally applied in CPSoS systems, cannot be directly used in Vehicular CPS, due to:

- The high speed of vehicles may lead to frequent and rapid changes of the network topology.
- The communication quality is degraded.
- The channel quality is vulnerable to the complex and changeable environment and various factors (e.g., various topologies of roadside entities, time-varying road situations, and the relative velocity among different vehicles).
- The high traffic density can result in huge overhead in the network.
- A large number of vehicles cause a high bandwidth demand.

In the next figure, an example of a coalition into Vehicular CPS is presented. The coalition, as a cluster with high transmission efficiency, is formed with nodes that have similar mobility. Through this scheme, each vehicle makes its own decision on which coalitions to join. Furthermore, the strategy is concerned with both the stability and efficiency of clusters in crossroads and reaching a stable coalition structure. There are also some vehicles, called malicious nodes, that can tell a lie to benefit from the clusters. They may report a false provided bandwidth that they do not have to occupy more resources. Those fraud behaviours result in a serious degradation in the aspect of communication efficiency or even communication interruption [218]

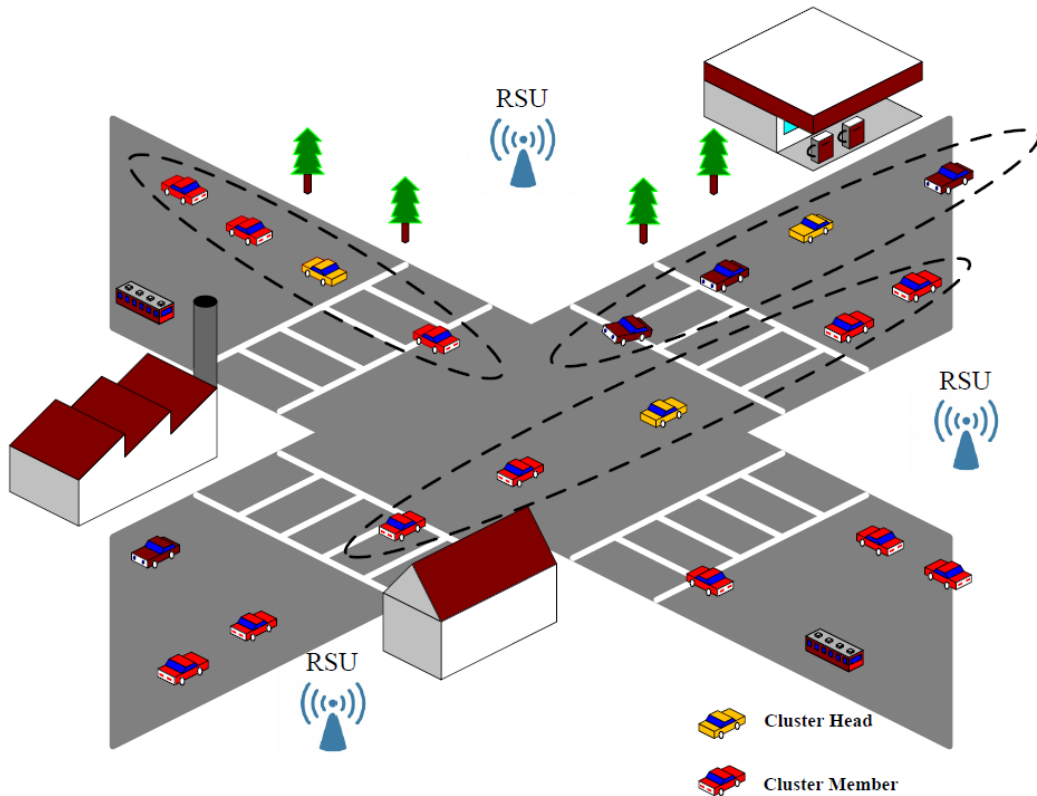


Figure 47 Coalition into Vehicular CPS [218]

## 6.2 Deep-priors-driven scene understanding

Autonomous driving is an important research field as it has a great impact in driving safety, reducing the traffic congestion and carbon emissions. Although a lot of effort has been made to develop autonomous driving vehicles, this is still a challenging task because an autonomous vehicle's system should accurately detect each object of the driving scenario. This detection must also be accurate, and it should occur in real time. A lot of different sensors are used to offer important information that could be exploited to detect different objects. Images from cameras are a key agent in the object detection task, so computer vision and machine/deep learning play a crucial role in object detection. In autonomous driving, usually an object is detected by estimating a classification probability and drawing a bounding box to map each object in the image, as observed in Figure 48.

In the case of manufacturing, the advent of a new age is characterized by notions like Industry 4.0 and Industrial Internet of Things, defined by technologies that affect vertically and horizontally the manufacturing environment. Even though manufacturing systems have always interacted with the physical environment via Operational Technologies, the ongoing convergence with Informational Technologies leads to a novel world of Cyber Physical Systems and smart manufacturing comprised of intelligent high-accuracy networked systems which usually have real-time requirements. Just like in the automotive sector, deep scene understanding can play a crucial role in production systems in a wide range of applications like object detection, defect detection, visual inspection etc.



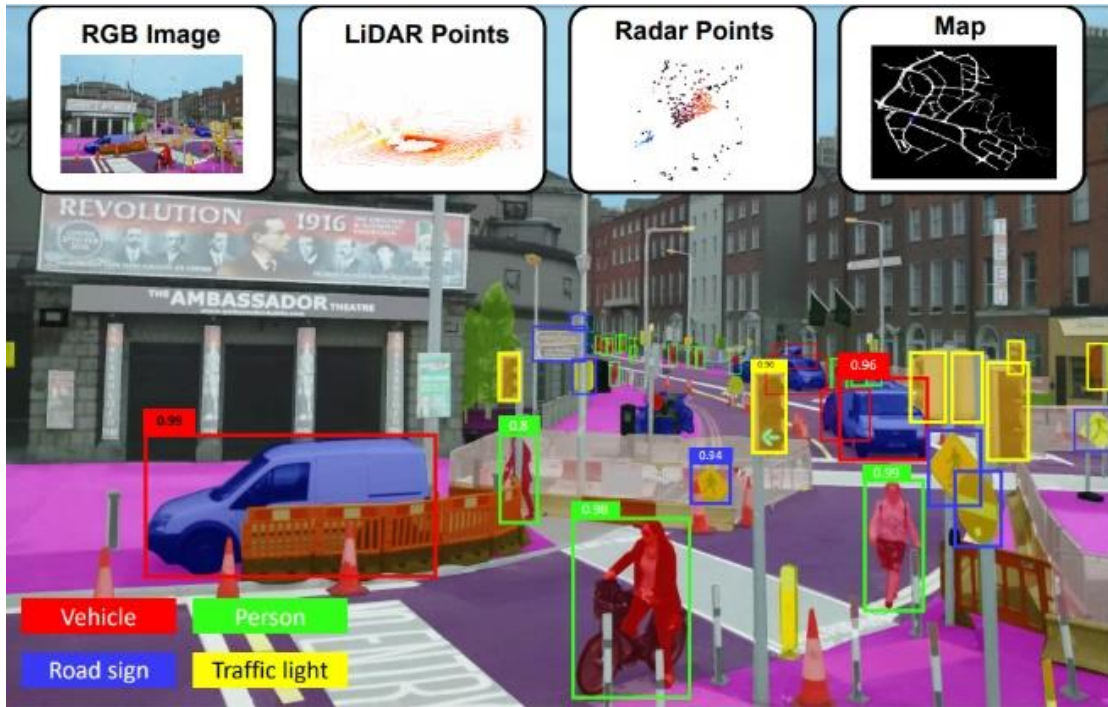


Figure 48 Object detection using multi-modal signals for perception.

### 6.2.1 3D object detection from 2D images

Deep learning algorithms are one of the most important approaches in object detection datasets [240][241] and autonomous driving dataset [251]-[265]. There are two main different approaches in deep object detection:

- **Two-stage object detection:** This approach is consisted of two stages. Initially, a lot of different regions of an image that could be an object are extracted. These regions are called regions of interest (ROI). Secondly these potential objects are classified calculating the probability to belong to each class. OverFeat[267] and R-CNN[268] are some of the state-of-the-art studies that use the two-stage object detection approach. Regions of interest are extracted using the sliding window approach OverFeat and selective search R-CNN. Secondly these regions lead to CNN for feature extraction and classification. Usually for each detected object a bounding box is produced. SPPnet[269] and Fast-RCNN[270] extract ROI using larger CNN (e.g VGG[271], ResNet[272], GoogleNet[273]) on the whole image.
- **One stage object detection:** This method uses a CNN model to map the feature maps to bounding boxes and classification scores. YOLO network [274] estimates the bounding boxes directly from the CNN model. Two stage object detectors tend to have a better prediction performance but are more computationally complex. This complexity usually leads to higher inference time and need more time to be trained[238].

### 6.2.1.1 Person detection

Person detection algorithms could be characterized as a subcategory of the general object detection task and are important in fields such as autonomous driving. These algorithms are valuable for Pedestrian Protection Systems (PPS). The most modern PPS use CNN and learn the extracted features in an end-to-end fashion[283]-[285]. Person detection algorithms can be applied to estimate the pose and a gaze of a person. This information could be valuable for an autonomous vehicle. Initial approaches try to detect body parts and, in second stage, try to estimate the pose of a person. As these methods have a lot of limitations (especially when one person is close to another) Pishchulin et al.[286] developed the Deepcut model which simultaneously calculates the poses of all persons in a single image[282].

### 6.2.1.2 Traffic sign detection

Deep traffic detection models could also be characterized as a subcategory of the general object detection task and are important in fields such as autonomous driving. Artificial neural networks are used to extract feature maps from images and sequentially deep models are used to classify these features and correctly detect traffic signs. Usually these models consist of many layers and are complex. Different architectures and optimizations lead to different computational complexity, so there is a variety of execution times, as it can be observed in Figure 49 [287].

### 6.2.1.3 Deep Semantic Segmentation

Semantic segmentation is a method which tries to associate each pixel of an image with a class label. In more detail, semantic segmentation divides an image into several meaningful parts. Each part is associated with a class-label. The segmentation and labeling of the image can occur in different levels. The most common levels are at:

- Pixel level semantic segmentation (labeling each pixel in the image with semantics)
- Instance level semantic segmentation (detecting objects and at the same time applying per-instance and per-pixel segmentation)
- Panoptic segmentation, which is a combination of instance segmentation (detect and segment each object instance) and semantic segmentation (assign a class label to each pixel)[288]

Although each segmentation method has a great impact on the accuracy of the prediction, in this study (Siam, M. *et al.*)[289] researchers measured the real-time performance among several state-of-the-art semantic segmentation architectures comparing the operations (GFLOPs) presented in Figure 50.

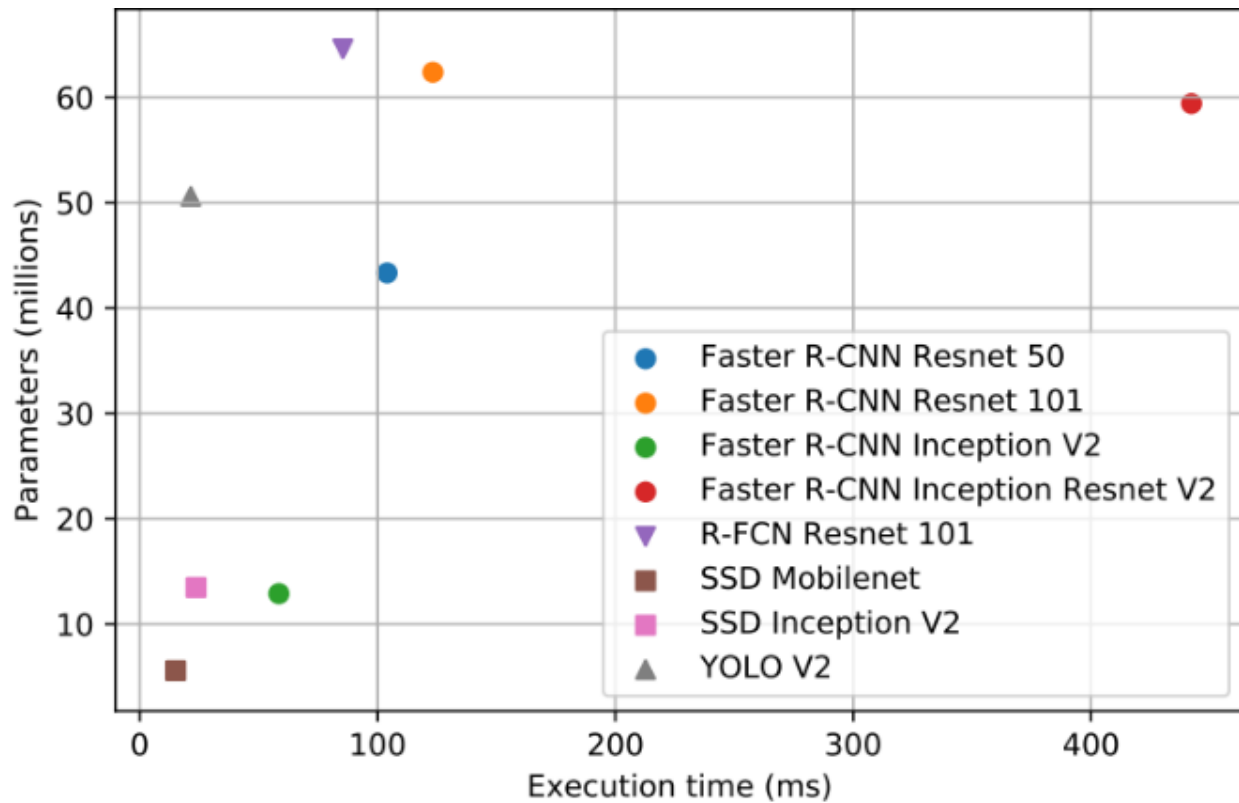


Figure 49 Parameters vs execution time (ms) for different traffic sign detectors

Model	GFLOPs	Class IoU	Class iIoU	Category IoU	Category iIoU
SegNet[1]	286.03	56.1	34.2	79.8	66.4
ENet[39]	3.83	58.3	24.4	80.4	64.0
SkipNet-VGG16[35]	445.9	<b>65.3</b>	<b>41.7</b>	<b>85.7</b>	<b>70.1</b>
SkipNet-ShuffleNet	<b>2.0</b>	58.3	32.4	80.2	62.2
SkipNet-MobileNet	6.2	61.5	35.2	82.0	63.0

Figure 50 Real-time performance among several state-of-the-art semantic segmentation architectures comparing the operations (GFLOPs)

The KITTI dataset [275] is the most used dataset for object detection in autonomous driving. Figure 50 shows the results of the comparison between the state-of-the-art methods that are used to detect objects such as cars, pedestrians and cyclists. This comparison is calculated in the KITTI dataset. The minimum bounding box height is 40 px, 25 px and 25 px for the easy, moderate and hard examples respectively. The objects that are considered as easy examples are fully visible and the objects that are considered moderate examples include partial occlusion. The objects that are characterized as hard examples include the maximum level of occlusion [282].

#### 6.2.1.4 Defect detection and production line inspection in manufacturing

Defect detection is a very popular area for employing deep learning technologies. For example, in [276] they deal with the visual investigation of vehicles produced by an assembly line in an automotive factory. The authors use a series of setups including GoogleNet and Alexnet over several platforms like Caffe, Torch, and Tensorflow. The system takes as input images of vehicles in the assembly line along with corresponding labels. The setup which includes Tensorflow outperformed the rest in terms of reaching the best accuracy in the shortest time. In [277] they propose a defect detection algorithm for tiny parts which is based on Single Shot MultiBox Detector (SSD) network and deep learning. The defects were categorized in four different class types and their size was in range of 0.8 cm. In addition, they compared their proposed algorithms with classical object detection algorithms (e.g. YOLO v3, Faster-RCNN etc.) and in different operational conditions (field of view and conveyor speed). Continuing with the defect detection field, Nakazawa *et al.*[278] proposed deep convolutional encoder-decoder neural network architectures for anomaly detection and segmentation in wafers in the semiconductor industry.

The paper in [279] deals with the identification of Pharmaceutical Blister Packages at dispensing stations. This Highlighted Deep Learning (HDL) approach automatically detects and segments the packages making them ideal classification candidates for a ResNet classifier. The application of deep neural networks in the identification of assembly operations is proposed by Chen *et al.* [279]. They present an architecture which is based on YOLO v3 for identifying the operations and a convolutional pose machine scheme for estimating the joint coordinates of the operators.

The authors of [280] present a multi-modal training and assisting system for smart manufacturing. The data from a series of ambient and wearable sensor are fed to CNNs for understanding the worker's behaviour and infer the possible guiding requirements. A system similar to ADAS is presented in [281] but adapted to the special needs of a manufacturing environment where certain areas should be inaccessible to automated vehicles. These areas can be delineated by beacons and a robust framework which fuses LIDAR and camera data in order to detect those beacons is proposed.

### 6.2.2 Multimodal fusion for object detection

The scene understanding can be further enhanced by using more than one sources of information. However, the increase of available data sources and corresponding datasets increases the degrees of freedom of the system and poses new questions concerning not only the processing of each dataset separately but what data can be combined, how this will happen and when it will take place.

#### 6.2.2.1 Data

The majority of these methods is based on supervised learning. Therefore, there is a need for multi-modal datasets providing both multi-modal data and the labelled ground-truth. Almost every one of them uses RGB camera images and in addition LiDAR point clouds [296][297][298], thermal images[299][300], a combination of the previous two[301], and radar data[302][303][304]. The size of the reviewed datasets ranges between 1,500 frames and 1.4 million frames. Even though the datasets used by the computer vision community are larger, the increase of volume of the automotive datasets is constant.

These datasets lack a variety in driving scenarios, weather conditions and sensor setups. One possible solution is the augmentation of the datasets with data generated by simulation engines or the use of datasets solely comprised by images produced by simulation environments. For instance, the import of

artificial blank areas, illumination invariances, occlusions etc. into KITTI dataset, has been reported as beneficiary to the overall robustness and accuracy of the networks, which use augmented KITTI as input for training. In addition, there are several works introducing virtual datasets or virtual simulators for producing datasets, but the question whether these artificial products can fully represent real-world objects is still open. Another issue of open datasets under consideration is the labelling of the data. Driving a car and acquiring data via multiple sensors mounted on the platform is one thing but labelling them is something different and much more difficult, especially when LiDAR point clouds are included.

Special care must be taken to ensure the precision and accuracy of the generated data, meaning that the sensors used have to be well-calibrated and positioned. In practice, the sensors are many times misaligned producing erroneous training datasets, especially when the algorithm must be aware of the exact sensor position. Several methods have been proposed for addressing this issue like the work presented in [318]. They propose a method which takes as input multiple channels of heterogeneous data and uses Deep Neural Networks (DNNs) for detecting the misalignments of the LiDAR and video inputs. More specifically, they discretize the spatial misalignments of the LiDAR-video input into nine classes and use a DNN for classifying these misalignments.

### 6.2.2.2 Process

When we have to define the pipeline for multi-modal object detection the main three topics that have to be tackled are the following:

- What kind of data should be fused? How will they be represented and processed?
- How will the fusion be done?
- When is the optimum point for the fusion to take place?

In Deep multi-modal object detection, the usual norm is the combination of RGB images and LiDAR point [305]-[313]. However, there are certain works where fusion includes different combinations of modalities, like RGB and thermal images [314], stereo and RGB images [315], radar and images [316], or RGB and depth images [316]. We will emphasize on the fusing of LiDAR points and camera images since this is by far the most popular method.

The point clouds provided by LiDAR provide depth and reflectance information of the scene. The useful depth information can be encoded in many ways, like in Cartesian coordinates, in HHA features, in density etc. They can be processed mainly in three ways: by producing 3D voxels from the 3D space and assigning the points to the voxels, by learning directly over 3D point clouds without the discretization of the 3D space, and by projecting the 3D point onto a 2D feature maps (e.g. spherical map, camera-plane map, Bird's Eye View (BEV) map etc.) and importing it afterwards into a 2D CNN. The spherical map is created by projecting the LiDAR points onto a spherical plane. The representation of the 3D points is very rich, which makes it suitable for scene segmentation. The different size of the images can result into different representation sizes, which makes the fusion at an early stage difficult. The Camera Plane Map (CPM) is the outcome of the projection of the point cloud onto the camera coordination system. Its main advantage is that it can be fused directly with camera images since they have the same size. On the other hand, many pixels are empty because of the projection and thus resulting in sparse feature maps which have to be upsampled. The BEV map is widely adopted in the area of scene understating since it avoids occlusion problems and makes the localization easier by providing the objects' lengths, widths and positions on the ground. Concerning the images acquired by camera sensors, it is certain that they provide rich information about the environment.

However, the restricted field of view produces occlusion and scaling problems. Therefore, the BEV images are considered more suitable for representing the scene data.

In most multi-modal object detection models the 3D point cloud supplied by the LiDAR is transformed into a 2D image through a type of a projection (e.g. plane, spherical, cylindrical etc.). Afterwards, conventional 2D object detection methods are employed in combination with the data from other modalities. The method proposed in [306], which is called MV3D, depicted Figure 51, combines data from a front facing RGB camera with bird-eye and front view projections of a point cloud. Each view constitutes an input and the features are extracted by VGG based feature extractors. The bird's eye input yields 3D proposals from a set of 3D prior boxes which are projected on every feature map. After that, a ROIpooling layer extracts feature vectors from each of the three different branches and feeds a deep fusion multi-view network that combines the features hierarchically. The generated fusion features are used to regress 3D boxes from 3D proposals. Moreover, the paper of [188] presents an evaluation study between different fusion schemes and it is concluded that the deep fusion approach outmatches the other methods in terms of flexibility concerning the aggregation of features from different sensing modalities.

In the work described in [305], AVOD is presented ( Figure 52), an object detection network for autonomous driving scenarios. The model uses input from RGB images and LiDAR point clouds for generating feature vectors that feed two different networks, a region proposal network (RPN) and a second stage object-detector network. The RPN network regresses the differences between a set of already defined 3D boxes, which are called anchors, and the ground truth. The areas having the greatest score are sampled and afterwards are projected onto the relevant view feature map. The distinct proposal from each branch are merged and a Fully Connected (FC) layer makes the classification and the 3D box proposals. However, due to detail loss after the convolutional layer, small objects cannot be detected. The upsampling of the feature maps is proposed utilizing Feature Pyramid Networks[318], something which leads to the increase of the detection sensitivity of the model concerning the size of the objects.

The main drawback of the aforementioned strategy is that the data representation and transformation may conceal or change 3D patterns or data invariances. The direct process of the 3D point clouds may be the solution to this. For example, another fusion strategy, which follows the opposite route than the one mentioned before, is to firstly define 2D object proposals from the monocular image and then extrapolate these proposals to the 3D space. The main issue is the efficient object localization in large point clouds. In the model presented in [307] and called Frustum Point-Net, monocular images are used for the generation of region proposals on the image plane and, afterwards, the point cloud provided by

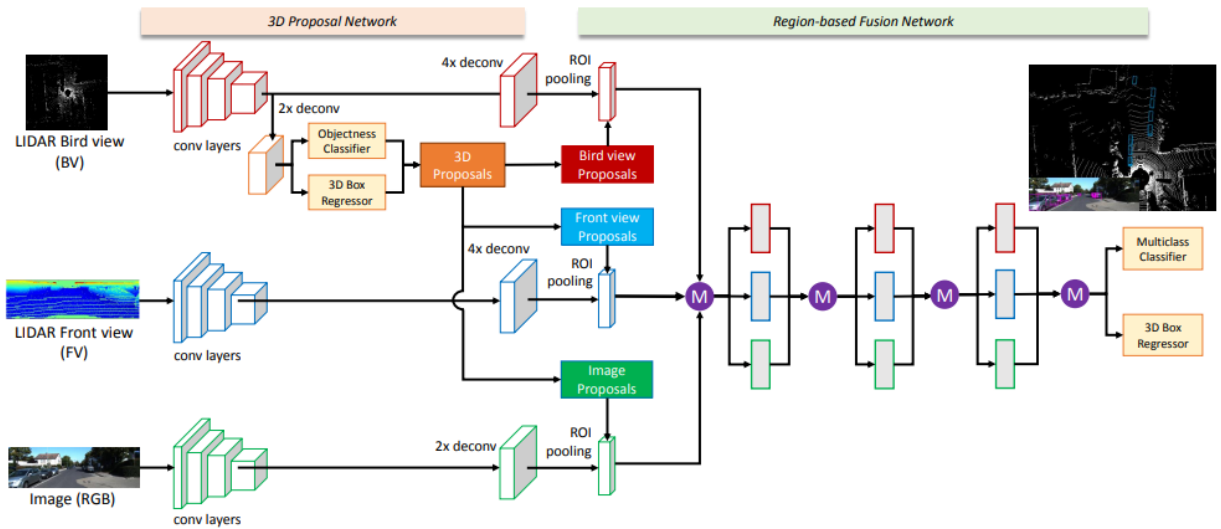


Figure 51 Multi-View 3D object detection network (MV3D)

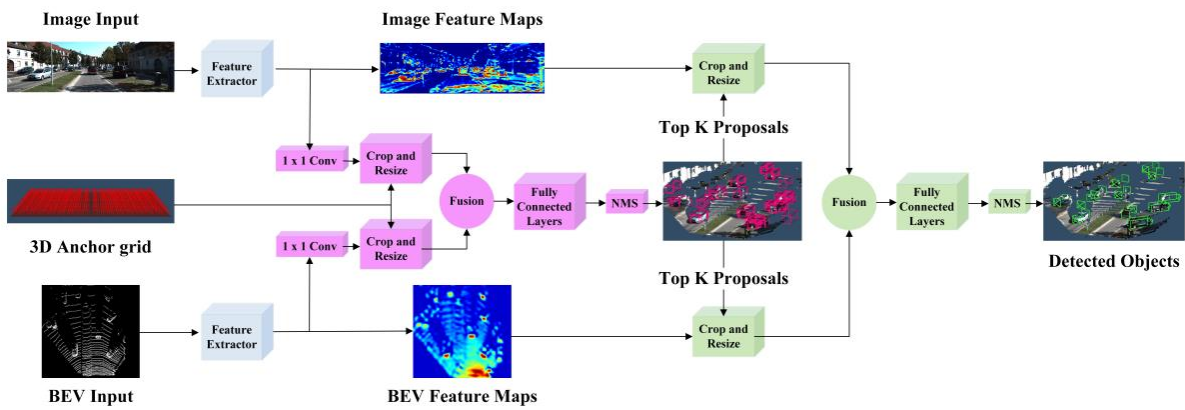


Figure 52 AVOD Architecture

LiDAR is used for performing the classification and regression of the bounding boxes. The extrapolation of the 2D boxes defined on the image plane to 3D space is accomplished by using the camera calibration values, something which outputs frustum region proposals. Within the 3D space trimmed by the frustums, consecutive 3D object segmentation and 3D bounding box regression are performed, employing two different instances of PointNet. The segmentation part produces a 3D mask of the object of interest and the nest instance provides an estimation of a 3D bounding box. In a similar manner, in [308] the points that lie in the bounding box projected in the image plane are selected and are used for model fitting and producing initial 3D proposals. These preliminary proposals are fed to a two-stage CCN that performs refinement and outputs the classification scores and the final 3D bounding boxes.

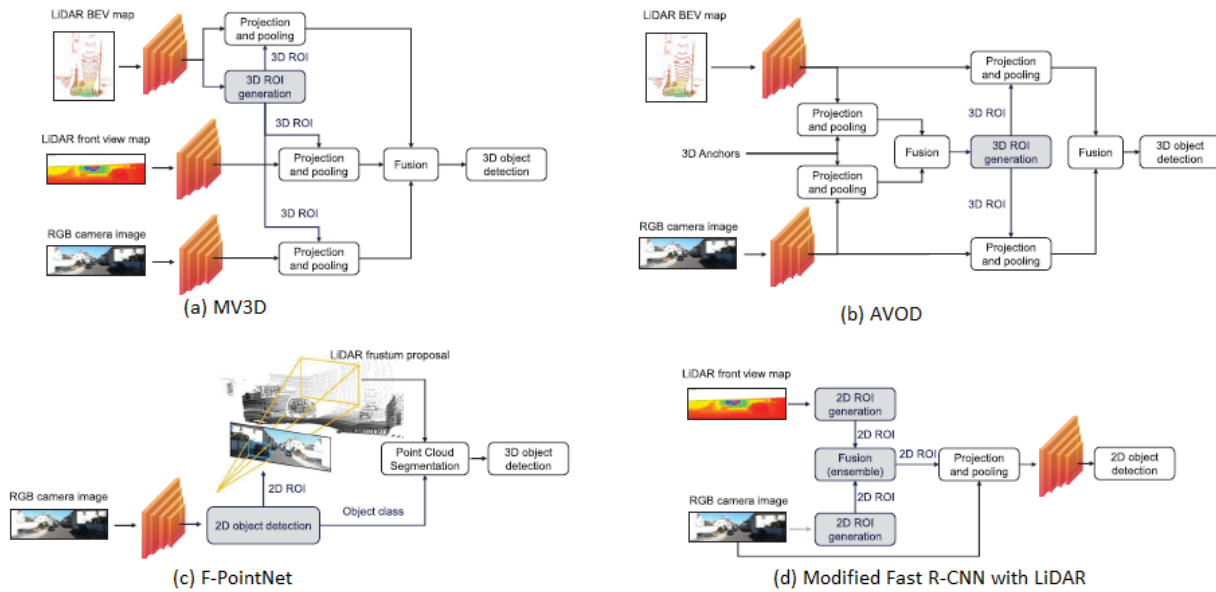


Figure 53 Architectures of the four multi-modal objection detection approaches [319]

In [309], they propose a modified Fast R-CNN with the incorporation of LiDAR. Initially, ROIs are extracted from RGB images using the fast mode of Selective Search [320]. The 3D point cloud is projected on the image plane resulting in a sparse LiDAR image. With the assistance of an inpainting module, which is provided by OpenCV, the depth image is interpolated. Using the Selective Search’s intensity mode for handling the gray-scale images, ROIs are extracted. The regions produced both from the 2D image and 3D LiDAR point cloud are imported into a CNN for extracting the feature vectors inside the proposed ROIs. Finally, a deep neural network provides class estimation from SoftMax classifier and object position form bounding box regression.

We assume that we have only two modalities, but the operations described can be extended to include a larger number of input data streams. With reference to the way the fuse is executed in a deep multi-modal object detection scenario we can distinguish the following categories [319]. Figure 53 visualizes four different multi-modal object detection approaches.

- Addition of Average Mean: the feature maps are added elementwise
- Ensemble: The feature maps from different modalities are ensembled
- Concatenation: the features are usually stacked along their depth and after the proceed to a convolutional layer.
- Mixture of Experts: this approach provides weights from each modality

Regarding the time when the fusion takes place, there are roughly tree approaches in the bibliography [319][320], which are listed below and depicted in Figure 54 :

- The **early** fusion approach, where the modalities are combined at an early stage of the process and thus the generated data stream is depended on all the modalities. This method uses raw pre-processed sensor data.



- The **late** fusion scheme where the modalities follow separate paths until they are combined at the final stage of the process. The absence of a modality is not significant since the object detection can rely on a prediction of the absent input data stream.
- The **middle** fusion scheme which is a comprise of the aforementioned schemes and the modalities are fused in intermediate levels (e.g. in [11] the fusion takes place hierarchically in neural network layers).

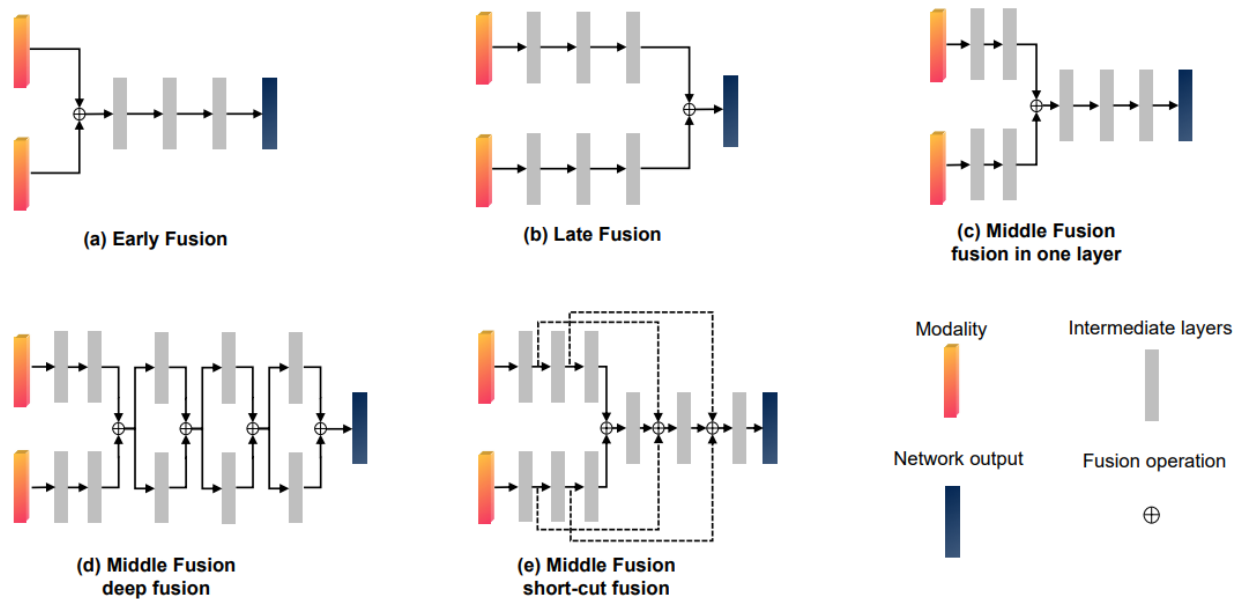


Figure 54 Different fusion schemes [319]

In the work presented by Schlosser *et al.* [322] an evaluation of the results of different fusion strategies was executed for a pedestrian detection application. Each fusion took place at a different stage and the authors tried to conclude about the optimal fusion stage in connection with the generated output. The input data were of two different types, monocular images and depth frames, and at the end it was found that late fusion schemes were more efficient, even though early fusion could yield comparable results with a minor performance drop.

### 6.2.3 Point cloud-based scene understanding

The 3D point cloud [324], as a primitive representation for objects, has become increasingly prevalent in many research fields, such as object recognition [320] and reconstruction [354], due to its simplicity, flexibility and powerful representation capability. In contrast to triangle meshes, the point cloud does not require to store or maintain the polygonal-mesh connectivity [324] or topological consistency. Processing and manipulating point cloud therefore can demonstrate better performance and lower overhead. These prominent advantages make the research on processing point cloud a hot topic. This section presents four categories

### 6.2.3.1 Volumetric methods

Voxelization of point cloud refers to transforming unstructured point clouds and making it into the regular volumetric occupancy grid, then learning its features by using neural networks to achieve the semantic segmentation of point cloud. Such solutions generate an occupancy grid and extend convolutional and pooling function to the 3D space.

VoxelNet, visualized in Figure 55 is a generic 3D detection framework that simultaneously learns a discriminative feature representation from point clouds and predicts accurate 3D bounding boxes in an end-to-end fashion. A voxel feature encoding (VFE) layer combines point-wise with a locally aggregated feature. Stacking multiple VFE layers allows learning complex features for characterizing local 3D shape information. Specifically, VoxelNet [327] divides the point cloud into equally spaced 3D voxels and encodes each voxel via stacked VFE layers, and then 3D convolution further aggregates local voxel features, transforming the point cloud into a high-dimensional volumetric representation. Subsequently a region proposal network [314] consumes the volumetric representation and yields the detection result. This efficient algorithm benefits both from the sparse point structure and from the efficient parallel processing on the voxel grid.

PointPillars [331], depicted in Figure 56 detects 3D objects using only 2D convolutional layers. PointPillars employs an encoder that learns from vertical columns of the point cloud to predict 3D oriented boxes for objects. First, by learning features instead of relying on fixed encoders, PointPillars operates on pillars instead of voxels avoiding the need to tune object orientation. Pillars are highly efficient since all key operations can be formulated as 2D convolutions which are extremely efficient to compute on a GPU.

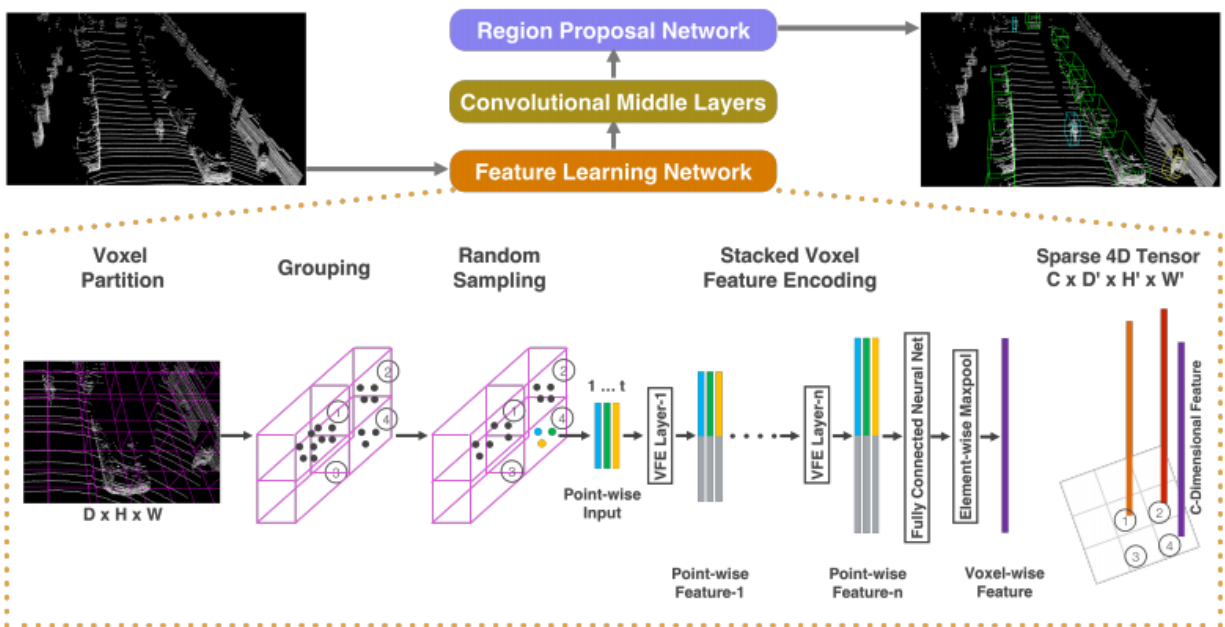


Figure 55 VoxelNet [327] architecture

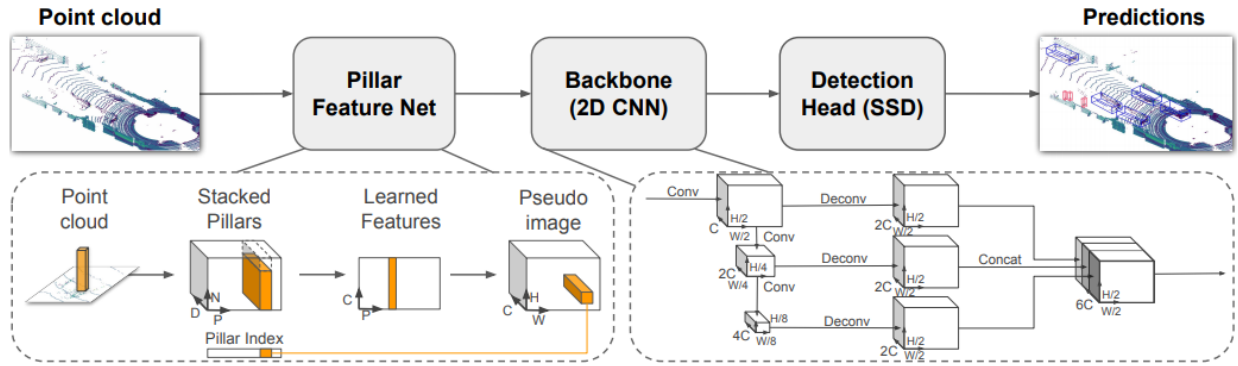


Figure 56 PointPillar[331] Architecture

### 6.2.3.2 Learning from raw point cloud data

To improve computational efficiency and memory requirements several approaches operate directly on the point cloud space. In contrast to fine grids, point cloud is a non-uniform sampling of a three-dimensional manifold.

PointNet [325] presents an approach receiving as input point clouds, with each point represented by just its three coordinates  $(x, y, z)$ , and generating labels for the entire input or in a per segment basis since the input can be a single object for part region segmentation, or a sub-volume from a 3D scene for object region segmentation. Initially, each point is processed identically and independently. Additional dimensions can be taken into account using normal vectors, color, or other local or global features. PointNet is depicted in Figure 57.

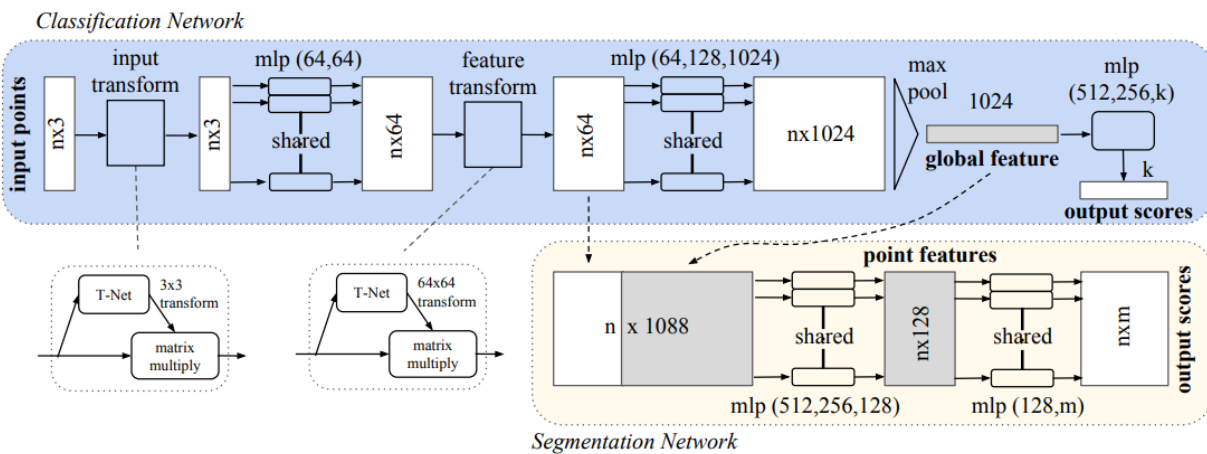


Figure 57 PointNet Architecture [325]

### 6.2.3.3 Fusion of 2D image processing and point cloud data

Approaches that detect 3D objects from RGB-D data [307][315] efficiently propose possible locations of 3D objects in a 3D space. Imitating the practice in image detection, it is straightforward to enumerate candidate 3D boxes by sliding windows. To reduce the search space mature 2D object detectors are employed. First, extract 3D bounding frustum are extracted by extruding 2D bounding boxes from image

detectors. Then, 3D object instance segmentation is consecutively performed. Amodal 3D bounding box regression is implemented using two variants of PointNet (Figure 58 ). The network predicts the 3D mask of the object of interest and the regression network estimates the amodal 3D bounding box of the entire object even if only part of it is visible. A variant of the aforementioned approach is PointFusion [329], presented in Figure 59 , using a different variant of PointNet. Frustum ConvNet [312], depicted in Figure 60, processes 2D region proposals in RGB images. They are extracted from object detectors already available in the literature. The network identifies the sequence of (possibly overlapped) frustums by sliding along the frustum axis and aggregates point-wise features as a frustum-level feature vector. A fully convolutional network (FCN) is afterwards employed to extract multi-resolution frustum features.

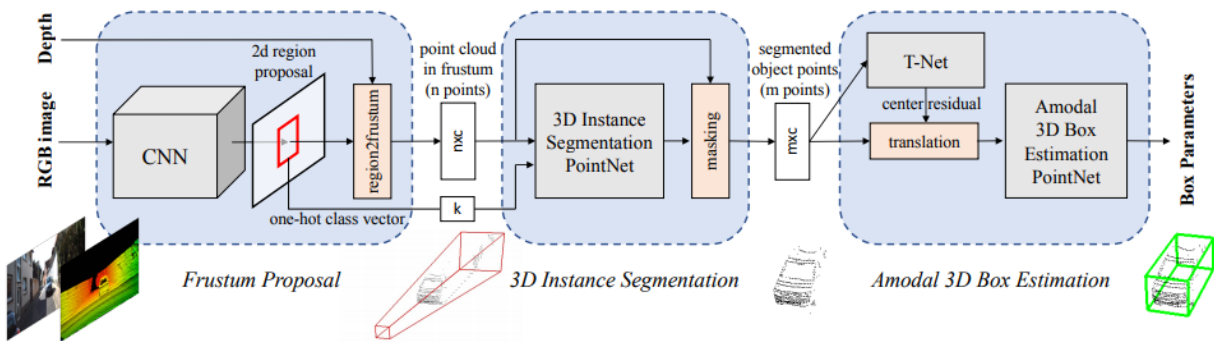


Figure 58 Frustum PointNets [312] for 3D object detection

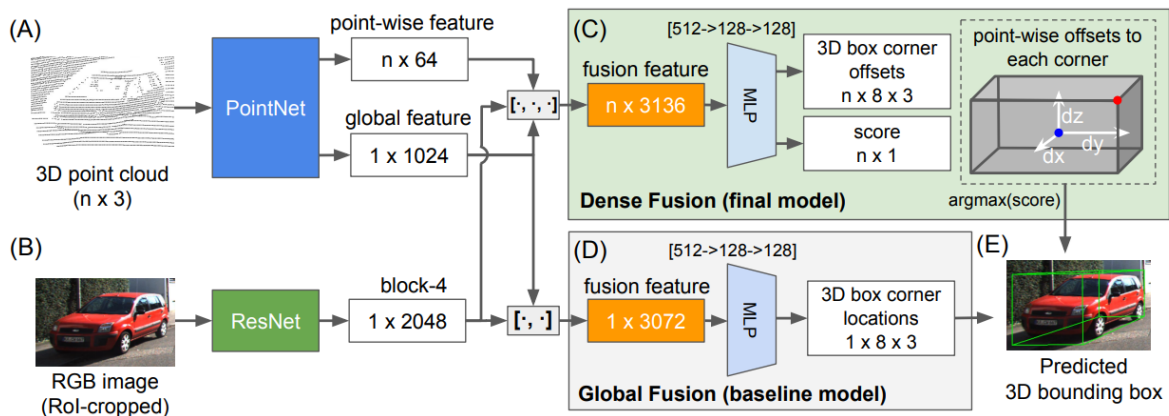


Figure 59 An overview of the dense PointFusion [329] architecture

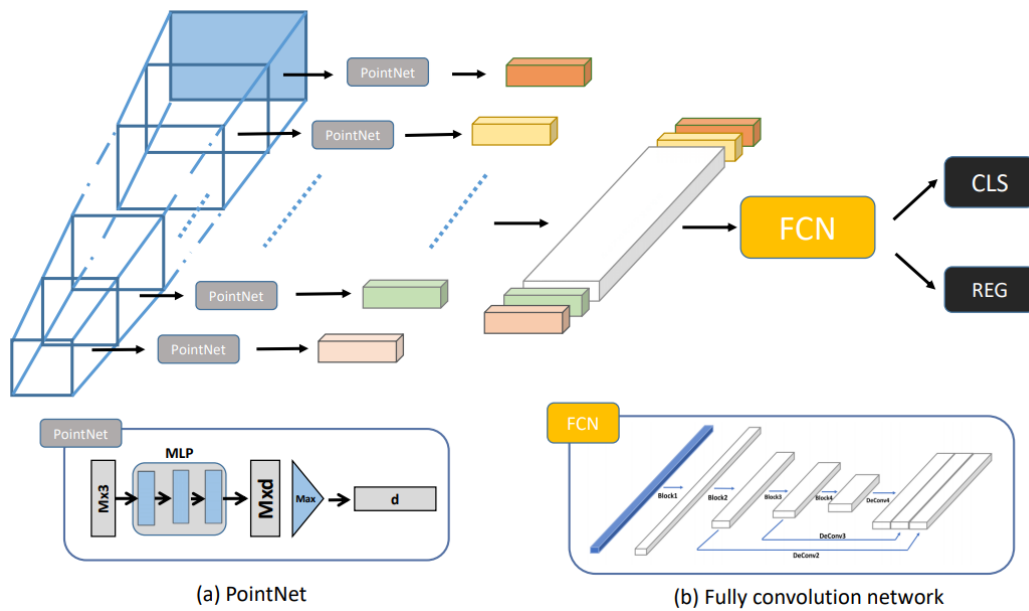


Figure 60 F-ConvNet architecture

#### 6.2.3.4 Two step part aware part aggregation

A novel detection framework [334], visualized in Figure 63, presents a two-stage 3D detection framework to detect 3D objects from raw point clouds. The first stage predicts intra-object part locations. The second stage aggregates the part information to improve the quality of predicted boxes. 3D bounding boxes are produced parameterized with  $(x, y, z, h, w, l, \theta)$ , where  $(x, y, z)$  are the box center coordinates,  $(h, w, l)$  are the height, width and length of each box respectively, and  $\theta$  is the orientation angle of each box from the bird's eye view. The network learns to segment the foreground points and estimate the intra-object part locations the segmentation masks and ground-truth part location annotations are directly generated from the ground-truth 3D box annotations 3D proposals from the raw point cloud simultaneously with foreground segmentation and part estimation.

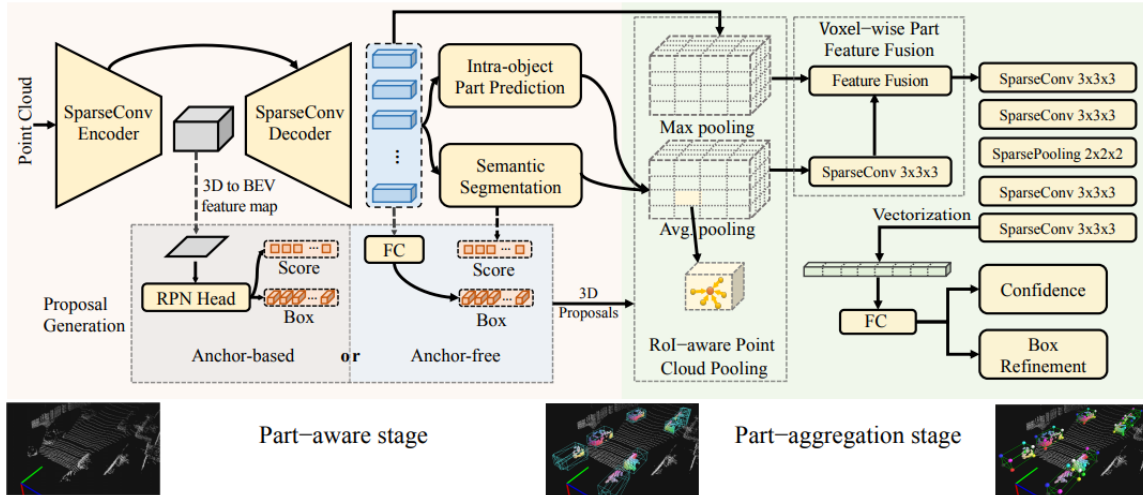


Figure 61 The overall framework of our part-aware and aggregation neural network for 3D object detection

### 6.3 Model Compression and acceleration approaches

Reducing the storage and computer power requirements of deep networks with many layers and nodes is an important factor, especially in real time applications (such as online learning and incremental learning). Additionally, in recent years we have witnessed significant progress in virtual and augmented reality applications, as well as in the development of a plethora of smart mobile devices (such as smart watches). These portable devices have limited memory, computing power and energy resources, therefore deep network compression and acceleration techniques are a challenge for the scientific community.

Applying these techniques in deep convolutional networks may have an important impact. The ResNet-501 is a very deep network as it is consisted of 50 convolutional layers, needs more than 95M of memory to be saved and a large amount of mathematical calculations for each image. However, applying the method of weight pruning has proven that the need of memory storage can be reduced by 75% and the computational complexity to be decreased by 50% without any notable change of network's performance

In general, these techniques could be categorized as follows:

- **Parameter pruning and sharing:** This technique aims in detecting and avoiding using network's parameters which are not especially important for the overall performance
- **Low rank factorization:** Low rank techniques exploit the decomposition of matrix/tensor in order to estimate which of the parameters are important (informative)
- **Transferred/compact convolutional filters:** The aim of this technique is to design convolutional filters in such a way as to reduce the computational complexity and storage needs.
- **Knowledge distillation:** These methods learn from a distillate model and train one more compressed network (with less layers) for the compressed network to reproduce the output of a deeper network.

Table 10: CNN model compression and acceleration techniques

Category	Description	Applications	Details
<b>Parameter pruning and sharing</b>	Detection and reduction of parameters which are not especially important	Convolutional and fully connected layers	Robust, incredibly good performance, it can be applied in pre-trained models or in models trained from scratch
<b>Low rank factorization</b>	Usage of matrix/tensor decomposition in order to detect the informative parameters	Convolutional and fully connected layers	Easily operated models, it can be applied in pre-trained models or in models trained from scratch
<b>Transferred/compact convolutional filters</b>	Designing of convolutional filters in such a way as to reduce the computational complexity and storage needs	Only for convolutional layers	These algorithms usually show good performance, support only training from scratch
<b>Knowledge distillation</b>	learning a distillate model and train one more compressed network in order for the compressed network to reproduce the output of a deeper network	Convolutional and fully connected layers	The performance of the model depends on the application and the structure of the network, supports only training from scratch

### 6.3.1 Parameter pruning and sharing

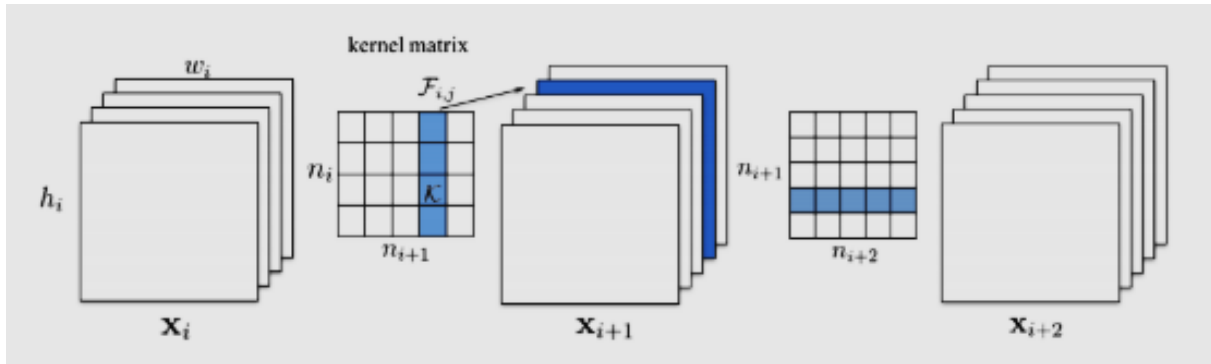


Figure 62 Pruning a filter results in removal of its corresponding feature map and related kernels in the next layer [558].

Parameter pruning and sharing techniques can be categorized as follows:

1. **Quantization of networks:** The process of quantization compresses an artificial neural network by reducing the number of binary digits which are used to represent the involved parameter (weights)
2. **Pruning and sharing:** Another approach is to prune redundant parameters or even entire neurons [357] of pretrained models.
3. **Designing of "structured matrix":** When network architectures consists of only fully connected layers, the number of parameters can increase rapidly, approaching even billions. Therefore, it is crucial to study the redundancy of these parameters, which has a great impact on the required memory and the complexity of the calculations. At each level of these networks, nonlinear transformations of the form  $f(\mathbf{x}, \mathbf{M}) = \sigma(\mathbf{M}\mathbf{x})$  are used, where  $\sigma(\cdot)$  is a non-linear function,  $\mathbf{x}$  is the input vector and  $\mathbf{M}$  is a matrix of parameters with dimension  $m \times n$  ( $m, n$  are related to the dimensions of the layer to which matrix  $\mathbf{M}$  is applying). When matrix  $\mathbf{M}$  has large dimension and is dense (it is consisted of many non-zero elements), the cost of storing  $mn$  parameters and the required operations for calculating products of Matrix-Vector are  $O(mn)$ . So, an intuitively correct way to prune parameters is to transform the matrix  $\mathbf{M}$  to a new "structured table" associated with far fewer parameters than  $mn$ .

### 6.3.2 Low rank factorization

As convolution is the main mathematical operation in CNNs, simplifying the convolutional layer has a direct and significant impact on total computational cost and could lead to network acceleration. The convolution kernel in CNNs is a four-dimensional tensor. An important observation is the fact that these tensors usually contain redundant information. Approaches based on tensor decomposition have been shown to be highly effective in removing this extra information, thus leading to the network's effective compression and acceleration. Low rank factorization is a prominent example of tensor decomposition approach.



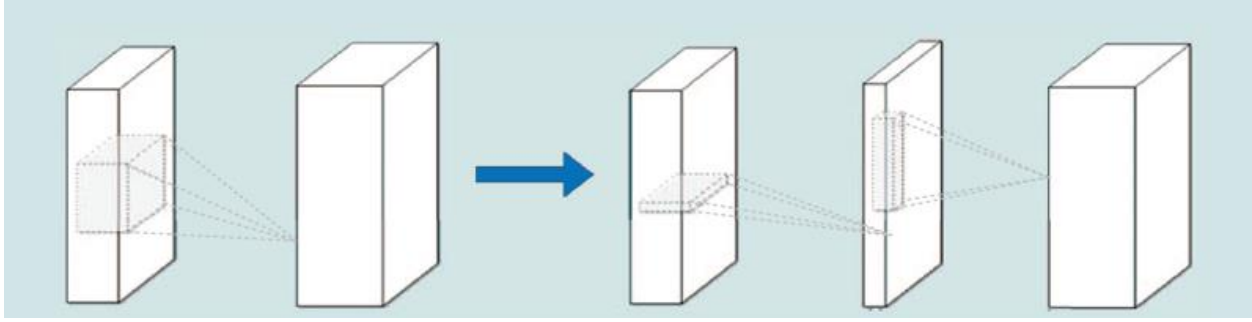


Figure 63 Low rank factorization

It should be noted that low rank decomposition can be used not only to the convolution but also to the fully connected layers (which can be represented as two-dimensional matrix). There are several studies aiming to exploit the low rank property of fully connected layers. For example, Denil et. al [359] uses the low-rank method to reduce the number of dynamic parameters in DL models, while Sainath et. al[360] developed a low rank decomposition method of the last layer of the CNN they created, aiming at acoustic modelling.

### 6.3.3 Transferred/compact convolutional filters

The translation invariant property, which is depicted by the representations of the input image, is a key factor of training very deep models without the risk of overfitting. Due to this property, CNNs are highly effective in terms of the number of exploitable parameters. Many empirical observations suggest that the translation invariant property combined with convolutional weights sharing, plays an important role in achieving good performance. Transferred convolutional filters can take advantage of this concept. The idea of using these filters for CNN compression is based on recent studies, introducing the so called equivariant group theory.

Specifically, if  $\mathbf{x}$  is the input of the network,  $\Phi(\cdot)$  the network or one of its layers and  $\mathcal{T}(\cdot)$  the matrix of the applied transformation, then the concept of equivariant can be defined as follows:

$$\mathcal{T}' \Phi(\mathbf{x}) = \Phi(\mathcal{T}\mathbf{x}) \quad (1)$$

This means that by transforming the input through  $\mathcal{T}(\cdot)$  and then passing it through the network or the layer  $\Phi(\cdot)$ , it should be equal with initially passing the input  $\mathbf{x}$  through the network and then transforming the resulting representation. Notice that the transformations  $\mathcal{T}, \mathcal{T}'$  in equation (1) are not necessarily identical since they act on different objects.

Based on that, the desired  $\mathcal{T}(\cdot)$  transformation can be applied directly to the layers of the network  $\Phi(\cdot)$  in order to achieve the compression of the network.

**Table 11** briefly compares the performance of different methods with transferred convolutional filters, using VGGNet (16 layers) as the baseline model. The results are reported on the CIFAR-10 and CIFAR-100 data sets with top-five error rates. It is observed that they can achieve reduction in parameters with little or no drop in classification accuracy.

Table 11 summarizes the comparison of various techniques based on transferred convolutional filters, using the VGGNet (16 layers), the CIFAR-10 and the CIFAR-100 data sets. It is easily observed that these techniques achieve a reduction in the number of parameters with little or no loss of accuracy.

**Table 11** briefly compares the performance of different methods with transferred convolutional filters, using VGGNet (16 layers) as the baseline model. The results are reported on the CIFAR-10 and CIFAR-100 data sets with top-five error rates. It is observed that they can achieve reduction in parameters with little or no drop in classification accuracy.

**Table 11 Compression efficiency of transferred convolutional filters**

Model	CIFAR-10	CIFAR-100	Compression factor
VGG-16 [553]	34.26%	9.85%	1
MBA [554]	33.66%	9.76%	2
CRELU [555]	34.57%	9.92%	2
CIRC [556]	35.15%	10.23%	4
DCNN [557]	33.57%	9.65%	1.62

#### 6.3.4 Knowledge distillation

One of the first works exploiting knowledge transfer in order to achieve network compression, was Bucilua et al. [366]. Initially, a compressed model was trained with pseudo-data, classified by a batch of strong classifiers. Based on this initialization, authors achieved to reproduce the error of the original uncompressed model. However, this work was limited to shallow networks. The main idea of Bucilua et al. has recently taken the form of knowledge distillation by Ba and Caruana et al. [367], where compression is achieved by creating a shallow network that mimics the training cost function of the deep network. Knowledge distillation focuses on to transfer knowledge from a large network-teacher to a simpler network-student, where the second learns the distribution of the first network's output classes, using the SoftMax function. Despite its simplicity, techniques that follow knowledge distillation based compression strategy, show promising results in various image classification tasks.

### 6.4 Multi-modal Localization for Connected and Autonomous Vehicles

Intelligent Transportation Systems (ITS) are expected to become one of the main pillars of the modern society. It is estimated that road transportation will be safer (less traffic accidents), environmentally friendly (reduced gas emissions), and smarter (vehicles will be informed which travel path to follow according to their needs). As such, ITS's social, environmental and economic impact must be taken into serious account by the industry and the scientific society, in order to develop efficient and advanced services and applications.

Autonomous vehicles are an integral part of ITS. Their functionality relies on five Systems[335]: Localization, Perception, Planning, Control and System Management. Localization provides absolute position information, allowing the vehicle to identify its position in a global coordinate system. Perception System is responsible for risk assessment. The vehicle perceives the surrounding environment, e.g. neighbouring vehicles, vulnerable road users, traffic signs, the road segment, etc., and the objects of the traffic scene are categorized according to the risk they pose. Planning System uses an input, the output of Localization and Perception, in order to determine the best possible actions to the travel path, such as braking, accelerating, steering, etc. Control System (also known as Adaptive Cruise Control), manipulates the vehicle according to the recommended actions of the Planning System. Finally, System Management coordinates the other four Systems and provides the human-machine interface, between the driver and the vehicle's operational system.

Apparently, Planning and Control require accurate localization information in order to operate accurately. Some meters (m) error can cause serious damage. Consider that the vehicle may be localized on the wrong side of the road and the estimated travel actions can lead to a traffic accident with other vehicles or pedestrians. Therefore, it is of high need to provide a Localization System for the autonomous vehicles, even with a centimetre (cm) level of accuracy. In [336] the positioning accuracy is distinguished to three levels: 1) which-road ( $\sim 5.0$  m), 2) which-lane ( $\sim 1.5$  m), and 3) where-in-lane (below 1.0 m). In autonomous driving, the first level is considered to be the worst-case localization error, while the third is the most desired.

Global Navigation Satellite Systems (GNSS), e.g. Global Positioning System (GPS), BeiDou, Galileo and Global Navigation Satellite System (GLONASS), provide absolute position information. GPS is the most common and cheap device currently employed for vehicle applications. However, its accuracy is between 5-10 m with clear sky or line-of-sight conditions between receivers and satellites, while in harsh and challenging environments, such as urban and dense areas, tunnels etc., localization error may reach 30-50 m [337], [338]. It is obvious that GPS alone, is prohibitive for autonomous driving.

In order to develop robust and accurate solutions to the localization challenge, GPS measurements need to be fused with measurements extracting from advanced sensors such as LiDAR, RADAR, Inertial Measurement Unit (IMU), Camera, etc. In recent years there is a growing interest and focus on Cooperative Localization (CL) as a means to improve GPS accuracy. The emergence of Internet of Things, 5G and Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) wireless communications technology (see Section 5.1.2.3) will allow the autonomous vehicles to connect and communicate with other vehicles, pedestrians and road-side-unit. As such, connected and autonomous vehicles (CAVs) can form a Vehicular-Adhoc-NETwork (VANET) (shown in Figure 66), exchange their measurements from different modalities and fuse them (cooperative multi-modal fusion), in order to reduce the localization error. Typical measurement models usually deployed for CL, are, apart from absolute position, relative distance using LiDAR, RADAR or Camera, relative angle or azimuth angle using LiDAR or RADAR, and radio ranging techniques [339], using V2X, such as Received Signal Strength (RSS), Time-Of-Arrival (TOA) Time-Difference-Of-Arrival (TDOA)), etc.

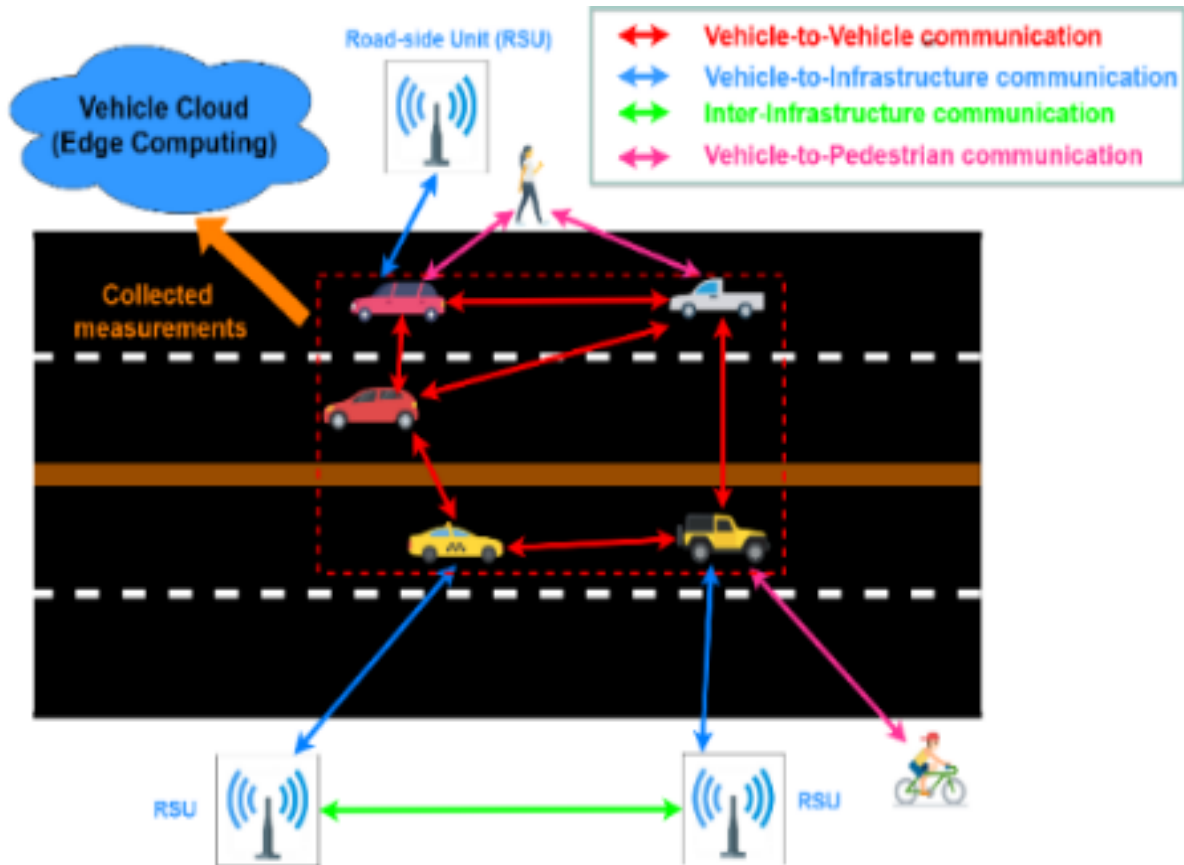


Figure 64 Example of VANET

#### 6.4.1 Multi-modal cooperative localization approaches

Multi-modal CL in VANETs is a research area closely related to CL in Wireless Sensor Networks (WSN) and Robotics. The works of [340], [341], provide an overview of multi-modal CL in WSN. CL algorithms can be categorized as:

- **Centralized vs Distributed:** In centralized approaches, all the measurements are sent to a central processor, which compute and estimate the true locations of nodes. In distributed approaches, the computation procedure is spread all over the network and only local information between connected neighbours is needed.
- **Bayesian vs non-Bayesian:** Bayesian approaches treat the position of each node as a random variable with a prior distribution, which depends on the node's measurements. Common Bayesian cooperative methods involve belief propagation (BP), particle filtering (PF), etc., that rely on minimum mean square error (MMSE) estimator and the maximum a posteriori (MAP) estimator. Non-Bayesian approaches treat the position of node as an unknown deterministic parameter. Common estimators are Least Squares and Maximum Likelihood (MLE). The former assumes that the measurements of node are depending on node's location.
- **One-shot vs Tracking:** One-shot approaches are generally non-Bayesian, which do not exploit motion models. On the contrary, tracking methods (Bayesian) integrate the mobility model of nodes into the CL.

A summary of centralized and distributed CL methods in Robotics, based on Extended Kalman Filter (EKF) is presented in [342]. In [343], four connected vehicles obtain absolute positions, relative distances and relative angles measurements using GPS and TDOA. Afterwards an objective cost function, formulated by the measurements and the MLE, is minimized by the optimization method of Alternating Direction Method of Multipliers (ADMM). In [344], vehicles share absolute position (using GPS), relative position (using laser scanners) and motion state (using odometers, accelerometers, gyroscopes) measurements and CL is performed by a covariance intersection filter (CIF), integrating those measurements. The VANET of [345], fuses absolute position (from GPS) and range measurements (from LiDAR), using Extended Kalman Filter (EKF) and CIF. In [338], a CL method in tunnels, that fuses: 1) V2X measurements between vehicles and road-side-units, 2) absolute positions of vehicles using GPS and 3) the lane constraints detected by a camera, using PF, is presented. In [337], a CL method in urban canyons, that fuses absolute position from GPS, odometers and gyroscopes and range measurements from V2V communications using EKF is proposed. In [346], a technique is proposed to improve the GPS-based vehicle positioning by sharing information and enabling cooperation amongst vehicles through V2V communication links, while making use of on-board sensing devices rather than active wireless technologies enabling explicit V2V measurements. In particular, a processing framework is proposed where a set of non-cooperative features (e.g., people, traffic lights, trees, etc.) are used as common noisy reference points that are cooperatively localized by the vehicles and implicitly used to enhance the vehicle location accuracy. A distributed Gaussian Message Passing (GMP) algorithm is designed to solve the positioning problem, integrating a Kalman filter (KF) to track the vehicle dynamics based on the on-board GPS measurements. Vehicles gather noisy observations such as relative locations, distances, and angles by their on-board RADAR. In [347], a Bayesian approach combined with KF that fuses absolute positions from GPS and inter-vehicle distance measurements using V2X, is employed in order to perform CL. In [348], a robust cubature Kalman filter (CKF) is proposed using GPS and V2V measurements in order to improve the performance of the data fusion under uncertain sensor measurements. In the proposed solution, the structure of the standard CKF is enhanced using the Huber M-estimation technique, in which the original measurement update in the CKF is modified considering the probable anomalies in state estimation. In [349], a hybrid-CL method is proposed, based on generalized approximate message passing (GAMP) and Kalman filter for vehicular network applications, which can make full use of navigation measurements from an onboard IMS, GPS receiver, Signals of Opportunity sources (e.g. TDOA), ground stations and inter-vehicle range measurements using V2V and V2I. The method adopts the framework of GAMP which exploits the central limit theorem and Taylor expansion to simplify the classical sampling mechanism in BP to a numerical computation process. For initialization, a KF to provide the initial states and parameters to the GAMP is used.

**Table 12. Survey papers related to WSN and Robotics**

	<b>Date of publication</b>	<b>Scientific Area</b>
H. Wymeersch et. al [340]	2009	WSN
R. M. Buehrer et. al [341]	2018	WSN
Anusna Chakraborty et. al [342]	2019	Robotics

Table 13 Summary of multi-modal CL in VANET

	Date of publication	Type of measurements	Estimation algorithm
H. Kim et. al [343]	2018	absolute position, relative distances and angles	ADMM
H. Li et. al [344]	2013	absolute and relative position, motion measurements	CIF
F. Bounini et. al [345]	2016	absolute position, relative distances, and angles	CIF, EKF
G. Hoang et. al [338]	2017	V2X measurements, absolute position, lane constraints	PF
Mariam Elazab et. al [336]	2017	absolute position, V2V measurements	EKF
G. Soatti et. al [345]	2018	absolute positions, relative locations, distances, angles	GMP, KF
M. Rohani et. al [346]	2015	absolute position, V2X distance measurements	Bayesian approach, KF
J. Liu et. al [347]	2017	absolute position, V2V measurements	CKF, Huber M-estimation
J. Xiong et. al [348]	2020	IMS, GPS, SOOP, V2V, V2I	GAMP, KF

## 6.4.2 Related issues

### 6.4.2.1 Conventional CL

The previously discussed CL methods rely on V2X and the fusion of different measurement modalities of a VANET. According to [336], they can be classified as modern CL techniques, whilst conventional CL methods are based on the sharing of data between a user and a reference base station which knows its position with high accuracy. Common approaches are Differential-GPS (DGPS), Real Time Kinematic (RTK) GPS, Assisted-GPS (AGPS), etc. Although their positioning precision may be lower than 1m, they cannot be considered a viable option for ITS applications. The main drawbacks include GPS signal coverage and blockage and multipath errors, caused by the urban areas.

### 6.4.2.2 GPS spoofing

The security and intact operation of GPS is a crucial factor of the feasibility of ITS. However, as shown in [350],[351],[352], GPS is susceptible to attacks, such as spoofing. GPS spoofing is related to the deception of the user by the transmission of signals with the same characteristics of legitimate GPS satellite signals. GPS vulnerability to spoofing can be seen on three levels:

1. **Vulnerability in signal processing:** The type of GPS signals (e.g. modulation type, transmit frequency, etc.) is publicly known. An attacker can generate signals with similar characteristics to the true ones and deceive the user.
2. **Vulnerability in data bit level:** Framing structure like satellite ephemeris and clock is also known and does not change rapidly. As such, the attacker can also exploit that information to generate deceiving signals.
3. **Vulnerability in Navigation and Position Solution:** The attacker can inject counterfeit pseudo range measurement and lead to wrong position, velocity, and time solution for the legitimate GPS receiver.

GPS spoofing techniques include Lift-off-delay, Lift-off-aligned, Meaconing or Replay, Jam and Spoof, Trajectory spoofing, etc. Techniques that try to defend against spoofing are based on Signal Power Monitoring, Signal Arrival Characteristics, Signal Correlation Peak, Antenna Array and Multi-Sensor Fusion. The latter defence approach fits with the rationale of CL in VANETs. The nodes of the network collaborate between themselves, exchange measurements and aim not only to improve location accuracy but also to detect and mitigate possible location attacks.

#### 6.4.2.3 Non-Line-of-Sight range measurements

The noise in range measurements is assumed to be Gaussian under the hypothesis of Line-of-Sight (LOS) between the vehicles. However, in a highly complex environment, it is probable that between two vehicles, an occluding object (e.g. building, vehicle etc.) also exists (Figure 65) and therefore, LiDAR, RADAR or V2X cannot provide an accurate range estimation e.g. for distances and angles. This effect is known as Non-LOS (NLOS) and it is a serious challenge of autonomous driving. The work in [353], tackles with detecting hidden objects (e.g. hidden around corners), using LiDAR confocal scanning. Under these assumptions, the noise in range measurements can be modelled as [354]:  $w \sim (1 - \epsilon) \cdot N(0, \sigma_{LOS}^2) + \epsilon \cdot N(\mu, \sigma_{NLOS}^2)$ , where  $\sigma_{LOS}^2$  is the variance of noise of range measurements in LOS conditions,  $\mu$  and  $\sigma_{NLOS}^2$  are the mean and the variance of noise of range measurements in NLOS conditions. Typically,  $\mu \gg 0$  and  $\sigma_{NLOS}^2 \gg \sigma_{LOS}^2$ . Moreover,  $\epsilon \in [0,1]$  is in fact the probability that the current measurement is NLOS affected. CL methods in VANETs could be useful also, in mitigating and fusing severely noisy range measurement, caused by NLOS.

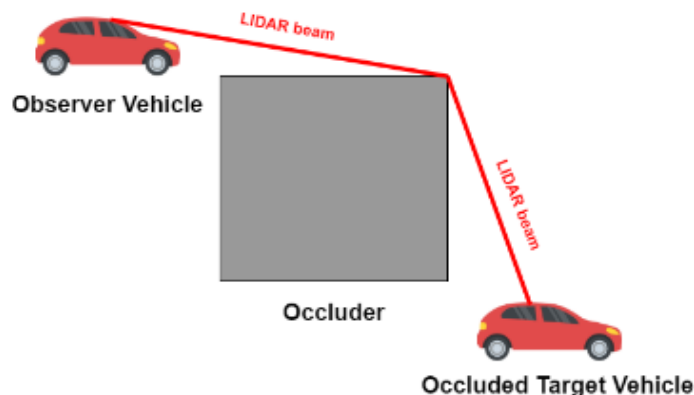


Figure 65 NLOS

## 6.5 Distributed framework with cyber-physical modelling

Several challenges arising in cyber-physical networks can be stated as optimization problems. Examples are estimation, decision, learning and control applications. To solve optimization problems over cyber-physical networks it is not possible to apply the centralized optimization algorithms, which require the data to be managed by a single entity. In fact, the problem data are spread over the network, and it is undesirable or even impossible to collect them at a unique node. In distributed computation the communication topology cannot be thought of as a design parameter, but as something that changes dynamically and has to be tracked. Thus, in cyber-physical networks, the goal is to design algorithms based on the exchange of information among the processors that take advantage of the aggregated computational power. All the agents must be treated as peers and each of them must perform the same tasks and no “master” node must be present. Moreover, information privacy is often a requirement (i.e., private problem data at each node must not be shared with the other nodes). These challenges call for tailored strategies and have given rise to a novel, growing research branch termed distributed optimization.

In a distributed scenario, we consider  $N$  units, called agents or processors, that have both communication and computation capabilities. Communication among agents is modelled by means of graph theory. Informally, given a graph  $G$  with  $N$  nodes, one for each agent, an agent  $i$  can send (receive) data to (from) another agent  $j$ , when the graph  $G$  contains an edge connecting  $i$  to  $j$  ( $j$  to  $i$ ). In a distributed algorithm, agents initialize their local states and then start an iterative procedure in which communication and computation steps are iteratively performed, with all the nodes performing the same actions. In particular, local states are updated by using only information received by in-neighbours. We consider a distributed framework in which agents cooperatively solve an optimization problem. The basic assumption we make is that each agent  $i$  has only a partial knowledge of the entire problem, e.g., only a portion of the cost and/or a portion of the constraints is locally available.

### 6.5.1 Overview of distributed computation models

In this section we formally define the communication model for a distributed algorithm. Given a network topology, agents can run distributed algorithms according to several communication protocols. Due to the dynamic environment, the communication links are time dependent, as shown in Figure 66. Links can be formulated/vanished using several criteria, such as distance between agents or correlation between their data.

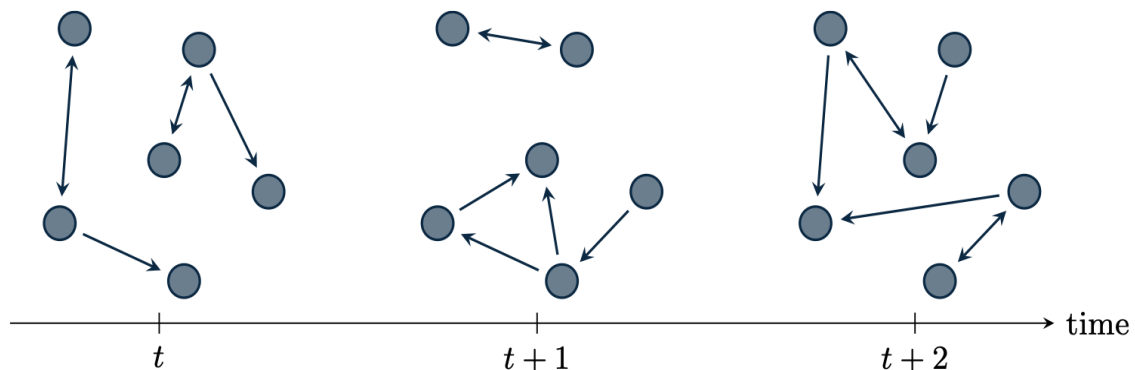


Figure 66 A graphical representation of a time-varying network with  $N=6$ .



When the network topology changes, what follows is that the steps of a distributed algorithm explicitly depend on each time instance, denoted by the value of  $t$ . In this case, we say that the algorithm is synchronous, i.e., agents must be aware of the current value of  $t$ , and, thus, their local operations must be synchronized to a global clock. If a distributed algorithm is designed to run over a jointly strongly connected graph, and the local computation steps do not depend on  $t$ , then the algorithm can be also implemented in an asynchronous network. Thus, agents are not aware of any global time information, i.e. their updates do not depend on  $t$ , and we term these algorithms asynchronous.

Next, we describe three general optimization setups that comprise several estimation, learning, decision and control application scenarios. A distributed optimization algorithm for such classes of problems consists of an iterative procedure based on the distributed computation model introduced previously. For an optimization algorithm the aim is to minimize a scalar objective function (or cost function) subject to the constraints, i.e.

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ subj. to } \mathbf{x} \in X.$$

where the generic set  $X$  can also be expressed by means of equalities or inequalities. In cost-coupled optimization problems, the cost function is expressed as the sum of local contributions of the cost function and all of them depend on a common optimization variable. The global constraint set is assumed to be known at each agent while the local cost function is known only by each agent. The Figure 67 provides a graphical representation of how problem information is spread over the network. The goal is to design a distributed algorithm where each agent updates a local estimate that converges (asymptotically or in finite time) to the global solution, by means of local computation and neighbouring communication only.

$$\min_{\mathbf{x} \in \mathbb{R}^d} \sum_{i=1}^N f_i(\mathbf{x})$$

subj. to  $\mathbf{x} \in X$ ,

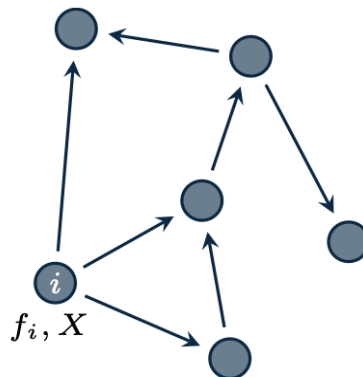


Figure 67 Cost-coupled setup where each agent  $i$  only knows  $f_i$  and  $X$  [105].

In the common-cost optimization problems, the cost function is shared, and the coupling among the agents is due to the fact that the optimization variable must belong to all the local constraint sets. Figure 68 provides a graphical representation of how information is spread over the network. In this case, the cost function is shared, and the coupling among the agents is due to the fact that the optimization variable must belong to all the local constraint sets. Again, the goal is to design a distributed algorithm where each agent updates a local estimate that converges (asymptotically or in finite time) to the global solution, by means of local computation and neighbouring communication only.

$$\begin{aligned} & \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) \\ & \text{subj. to } \mathbf{x} \in \bigcap_{i=1}^N X_i, \end{aligned}$$

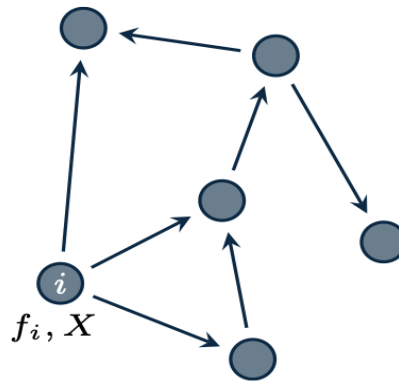


Figure 68 Cost-coupled setup where each agent  $i$  only knows  $f_i$  and  $X$  [105].

A different set-up, constraint-coupled optimization problems, agents in a network want to minimize the sum of local cost functions, each one depending only on a local vector satisfying local constraints. The decision vectors are then coupled to each other by means of separable coupling constraints. This feature leads easily to the so-called big-data problems having a very highly dimensional decision variable that grows with the network size. However, since agents are typically interested in computing only their (small) portion of an optimal solution, novel tailored methods need to be developed to address these challenges.

$$\begin{aligned} & \min_{\mathbf{x}_1, \dots, \mathbf{x}_N} \sum_{i=1}^N f_i(\mathbf{x}_i) \\ & \text{subj. to } \mathbf{x}_i \in X_i, \quad i \in \{1, \dots, N\} \\ & \sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}, \end{aligned}$$

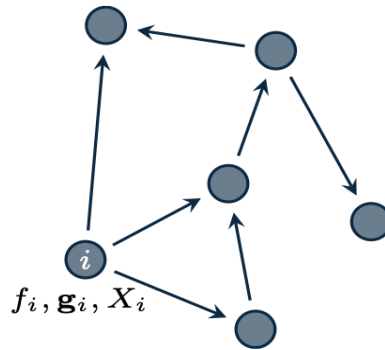


Figure 69 Constraint-coupled optimization [105].

Notice that this problem is challenging because of the coupling constraints. If there were no coupling constraints, the optimization would trivially split into  $N$  independent problems. The goal is to design a distributed algorithm where each agent updates a local estimate that converges (asymptotically or in finite time) to the local solution, by means of local computation and neighboring communication only. Next, we describe two basic optimization problems that intended to be used in the cases of interest, i.e., connected and autonomous cars and connected collaborative industrial robots.

**Linear regression** is an important problem that occurs in several applications. In linear regression, we assume that a set of points in a training dataset is used to estimate the parameters of a model (assumed to be linear in the parameters). The model can be exploited, e.g., to predict new generated samples. Figure 69 proposes a pictorial representation of a simple scenario. A natural scenario is to assume that the training data are not (or cannot be) gathered at a main collection center. Rather, it is reasonable to assume that the samples are (spatially) distributed in a network.

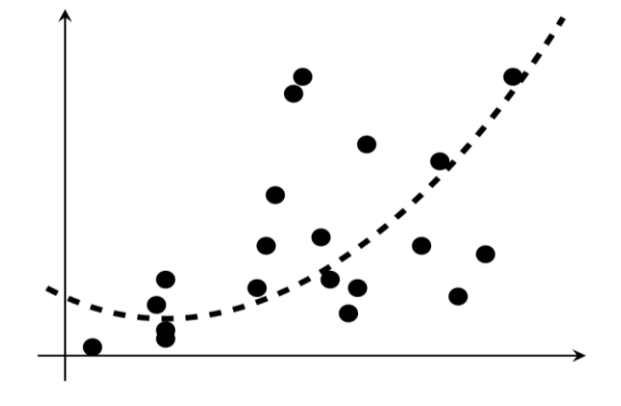


Figure 70 Set of data points that can be fit using a polynomial model (i.e., linear in the parameters). The coefficients of the polynomial are obtained with a regression approach.

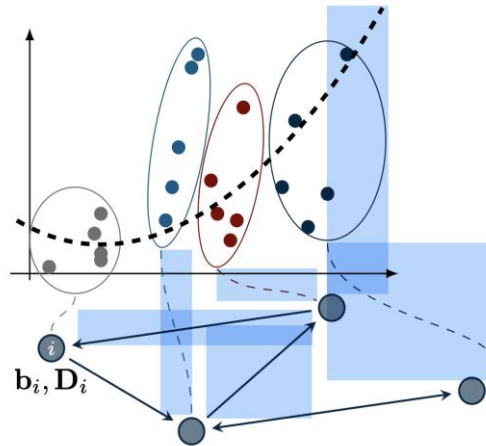


Figure 71 Regression problem over a network of 4 agents.

A popular regression approach is Least Squares (LS). If  $N$  processors in a network want to solve a regression problem, where  $\mathbf{x}$  denotes the parameter vector that has to be estimated, and each agent has  $n_i$  observations, the (unweighted) LS problem can be formulated as

$$\min_{\mathbf{x}} \sum_{i=1}^N \|\mathbf{D}_i \mathbf{x} - \mathbf{b}_i\|^2$$

A typical challenge arising in regression problems is due to the fact that the LS problem may be ill-posed and can easily lead to over-fitting phenomena. A viable technique to prevent over-fitting consists in adding a suitable regularization term  $r(\mathbf{x})$  in the cost function, leading to

$$\min_{\mathbf{x}} \sum_{i=1}^N \|\mathbf{D}_i \mathbf{x} - \mathbf{b}_i\|^2 + r(\mathbf{x})$$

where  $r$  is assumed to be known by all the agents in the network. Several possibilities for the regularizer  $r(\mathbf{x})$  can be chosen. For instance, by using l1-norm, we obtain the so-called LASSO (Least Absolute Shrinkage and Selection Operator) problem, i.e.,

$$\min_{\mathbf{x}} \sum_{i=1}^N \|\mathbf{D}_i \mathbf{x} - \mathbf{b}_i\|^2 + \rho \|\mathbf{x}\|_1$$

where  $\rho$  is a positive scalar used to strengthen or weaken the effects of the regularizer.

**Model Predictive Control (MPC)** is a widely studied technique in the control community and is also used in distributed contexts. The goal is to design an optimization-based feedback control law for a (spatially distributed) network of dynamical systems. The leading idea is the principle of receding horizon control, which informally speaking consists of solving at each time step an optimization problem (usually termed optimal control problem), in which the system model is used to predict the system trajectory over a fixed

time window. After an optimal solution of the optimal control problem is found, the input associated to the current time instant is applied and the process is repeated.

$$\begin{aligned} & \min_{\substack{\mathbf{z}_1, \dots, \mathbf{z}_N \\ \mathbf{u}_1, \dots, \mathbf{u}_N}} \sum_{i=1}^N \left( \sum_{s=0}^{S-1} \ell_i(\mathbf{z}_i(s), \mathbf{u}_i(s)) + V_i(\mathbf{z}_i(S)) \right) \\ \text{subj. to } & \mathbf{z}_i(s+1) = A_i \mathbf{z}_i(s) + B_i \mathbf{u}_i(s), \quad s \in \{0, \dots, S-1\}, \forall i, \\ & \mathbf{z}_i(s) \in Z_i, \mathbf{u}_i(s-1) \in U_i \quad s \in \{1, \dots, S\}, \quad \forall i, \\ & \mathbf{z}_i(0) = \mathbf{z}_i^0, \quad \forall i, \\ & \sum_{i=1}^N H_i \mathbf{z}_i(s) \leq h, \quad s \in \{1, \dots, S\}, \end{aligned}$$

where:

$$\sum_{s=0}^{S-1} \ell_i(\mathbf{z}_i(s), \mathbf{u}_i(s)) + V_i(\mathbf{z}_i(S))$$

represents the sum of the stage and terminal costs over the prediction horizon  $S$ . While,

$$\mathbf{z}_i(s+1) = A_i \mathbf{z}_i(s) + B_i \mathbf{u}_i(s).$$

is the state space equation that captures the dynamics of the system. With,

$$\begin{aligned} & \mathbf{z}_i(s) \in Z_i, \mathbf{u}_i(s-1) \in U_i \\ & \mathbf{z}_i(0) = \mathbf{z}_i^0, \\ & \sum_{i=1}^N H_i \mathbf{z}_i(s) \leq h, \end{aligned}$$

representing the constraints on the local optimization variables. This problem can be fit into the constraint-coupled set-up and solved accordingly over the network.

Alternating direction multiplier method (ADMM) is an iterative algorithm originally used for solving convex minimization problems by means of parallelization. ADMM assures convergence for convex problems, but also with non-convex problems under very light assumptions. In [106] a formulation of ADMM is introduced to make it scalable when applied to scenarios (different from the classical average consensus one). Every node in the network is interested in estimating a common but local vector of parameters and eventually reach consensus only with neighbours.

Privacy preservation is also an important aspect of distributed computation. In [107] the problem of privacy preservation is addressed at the physical layer, by tracking the correlation of multivariate streams recorded in a network of devices. To improve communication efficiency between connected devices, the inherent properties of the correlation matrices is exploited, and track the essential correlations from a small subset of correlation values.

## 6.5.2 Connected and autonomous vehicles as distributed CPS

Navigation or guidance is of paramount importance in an autonomous car because its primary function is to enable the car to travel on the desired path. When the autonomous car is aware of its environment then it needs to plan its path based on the destination. With the help of navigation hardware such as the well-known GPS module, the car generates a path between the current position and destination as a function of time. GPS is the primary source of navigation for the car because of its accuracy, optimized and compact hardware, on-chip design, low cost, and wide range use. Furthermore, the path is dynamically re-calculated in case of certain events such as roadblock, diversion, and so forth. The car's navigation system must be robust to handle sudden and subsequent changes in the path by adjusting the already pre-computed route. Road networks are physically predefined and the autonomous car's guidance system regularly checks the car's movement against the calculated path.

It is worth pointing out that although a GPS-based solution provides a rich set of functionalities in guidance and navigation, in certain scenarios, GPS on its own is not sufficient. Since GPS is based on signals from in-orbit satellites, the signals may sometimes get blocked or deteriorated due to natural or artificial phenomena, such as underground roads and tunnels. In such cases, other means of inertial guidance and navigation are needed. To address this issue, the autonomous car must be equipped with gyroscopes and accelerometers.

The inertial method of positioning (i.e. a gyroscope-based solution) does not provide information about the position of the vehicle, therefore the initial position for the gyroscope must be either provided through GPS or entered manually. In the case of autonomous cars, both the gyroscope and GPS can work well together if the context of movement is known. For autonomous cars, GPS information is frequently used as an input to a special map-generation algorithm that uses data acquisition and sensory information acquired from the vehicle. Several research efforts have been conducted and tested on real-world data to generate a map for autonomous cars [224]. The results are promising and will help in the initial commercial deployment phase of autonomous cars.

The heart of an autonomous car is its computing unit that implements the logic of the autonomous car in a holistic way. Sensors and actuators play a pivotal role in the realization of an autonomous car system. The autonomy of an autonomous car means handling of both known and unknown environments without any human intervention and needs machine learning, deep learning, and artificial intelligence algorithm techniques as discussed in the previous subsections. These algorithms are data-intensive, and the data is acquired through arrays of different sensors, which collectively form a massive sensor network within the car. Therefore, data acquisition, collection, storage, processing, communication among different entities within the car and with the environment, and the control of autonomous car are key aspects that need proper mechanisms.

On the other hand, with the removal of human involvement, autonomous cars have to make autonomous decisions based on what is best in a particular circumstance. This characteristic also requires the autonomous car to be more connected to the surrounding environment and draw as much data as it can from neighbours, infrastructure, and the Internet to make the best decision. Therefore, communication is of pivotal importance for the autonomous car. In this context, this subsection considers communication

within the autonomous car among different modules, communication between the autonomous car and the environment (including pedestrians and infrastructure) and in-car sensor data analytics.

In the field of intelligent transportation system (ITS), many different formulations based on IoT and CPS are proposed [223]-[229]. However, most studies only focus on one special aspect of ITS, such as architecture design, control algorithm, communication technology, or application in a certain scenario. As a comprehensive system, an ITS consists of a cyber part (such as information collection, communication, control mode, collaborative algorithm, and so on) and a physical part (such as connected and automated vehicles (CAVs), basic infrastructures, different kinds of sensors, on-board computers and controllers, and so on). Hence, CPS are very suitable for ITS to improve safety and mobility. Through CPS, people can directly pass instructions to physical objects in ITS to control them or to collect information while making decisions.

A three-layer distributed cyber-physical system (DCPS) is proposed in [223] to describe and analyse the performance of CAVs in ITS. All the physical parts of the DCPS are in the upper portion of each layer and the cyber parts are in the lower portion in Figure 72 [223]. In the information layer, the infrastructures will collect information from multiple sources. The collected information will be passed only to leading vehicles through V2I communication to help plan the trajectory. When the leading vehicles start to move, the following vehicles are expected to follow the desired trajectory. The following vehicles do not need to receive real-time information feedback from infrastructures, and thus the communication delay caused by exchanging and processing information is eliminated. In this way, the interaction of information between the information layer and the cooperation layer will be compressed to a lower level to protect the privacy

of CAVs compared with the centralized control mode, in which every CAV needs to communicate to the infrastructures.

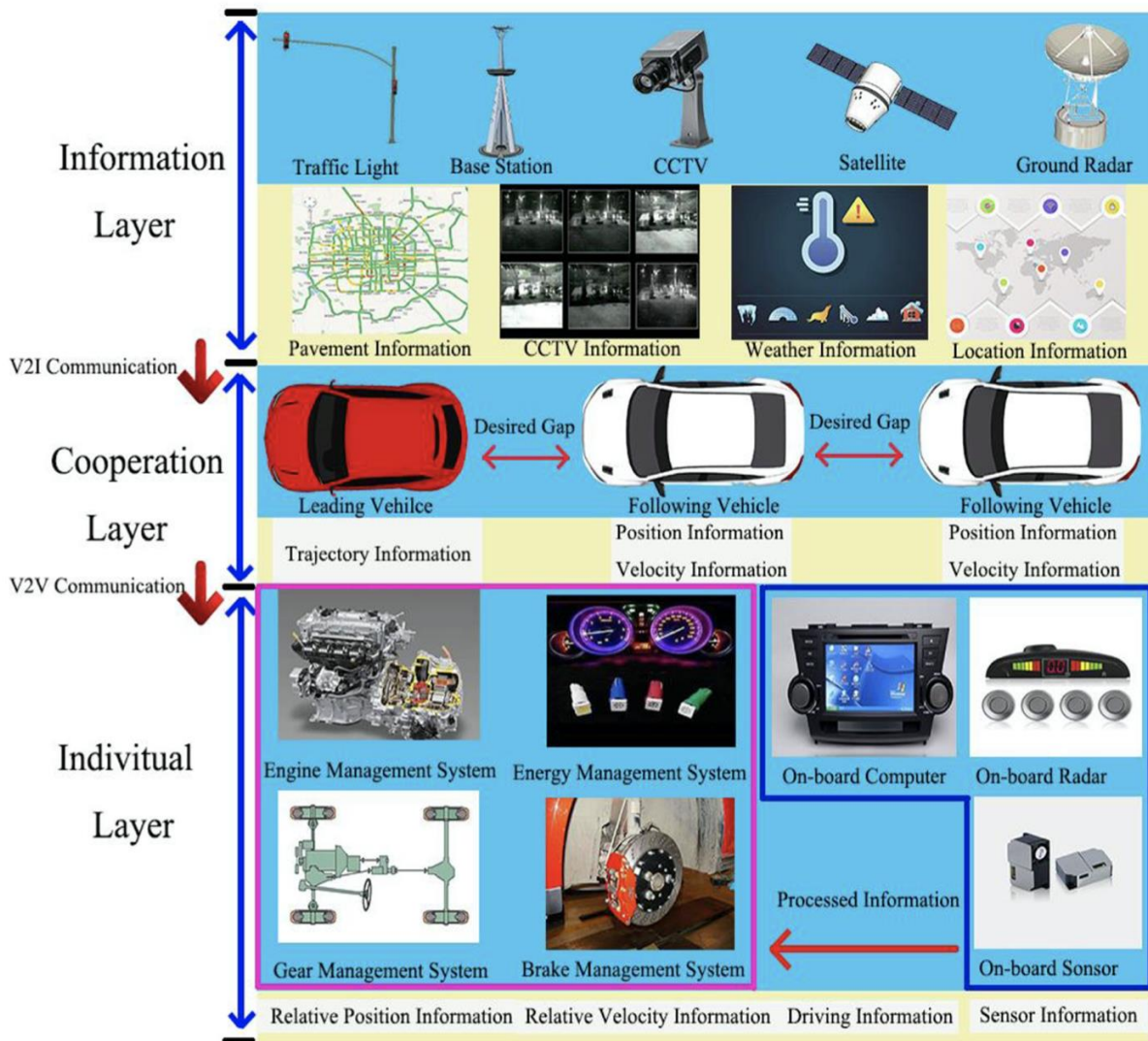


Figure 72 Concept of DCPS for autonomous and connected vehicles

In the individual layer, each CAV consists of a physical part and a cyber part. In the physical part, the on-board radar will collect velocity and position information from the nearest neighbours and the on-board sensors will collect sensor information such as the surrounding information, the lane information, and the driving information from various systems of the vehicle. In the meantime, the on-board computer will calculate the relative velocity information and the relative position information according to its own velocity and position information. The processed information will be passed to the engine management system, the energy management system, the gear management system, and the brake management system to control the speed for maintaining the desired gap.



For instance, let us consider the example where the vehicles need to keep a constant space with their neighbours while traveling along a desired trajectory. The trajectory is planned by the leading vehicles and is specified by a constant space, i.e., the desired gaps. In the cooperation layer, leading vehicles are assumed to travel in the boundary of the CAV team and perfectly track the desired trajectory. Since the following vehicles will adjust their speed according to the velocity and position information of their nearest neighbours, the real-time information exchange, processing and feedback are not necessary. The velocity and position information can be obtained through an on-board radar. That is, the trajectory information planned by leading vehicles is not passed through direct instructions, but through velocity and position that can be detected. If we consider the following distributed linear control law, where the control action  $u_i$  of an agent only depends on the relative position  $p_i, p_j$ , and relative velocity information from its neighbours, i.e.,

$$u_i = - \sum_{j \in \mathcal{N}_i} (k(p_i - p_j + \Delta_{(j,i)}) + b(\dot{p}_i - \dot{p}_j))$$

where  $k$  and  $b$  are positive constants  $i=1,2,\dots,N$ . The relative velocity and relative position information can be obtained by the on-board radar. The closed-loop dynamics of the network can now be expressed by the following coupled ordinary differential equation:

$$\ddot{\tilde{p}}_i = - \sum_{j \in \mathcal{N}_i} (k(\tilde{p}_i - \tilde{p}_j) + b(\dot{\tilde{p}}_i - \dot{\tilde{p}}_j)) + \omega_i$$

where

$$\tilde{p}_i := p_i - p_i^*$$

is the position tracking error. Information graphs can be used to model the interaction topology between agents. In [223], the analysis is restricted to a specific class of information graph, namely a finite rectangular lattice. Reference nodes are only placed on the boundaries because leading vehicles typically are the outermost vehicles in a formation. The next figure shows a simple formation and its information graph of a vehicular platoon, and several 2-D lattice graphs with different boundary conditions.

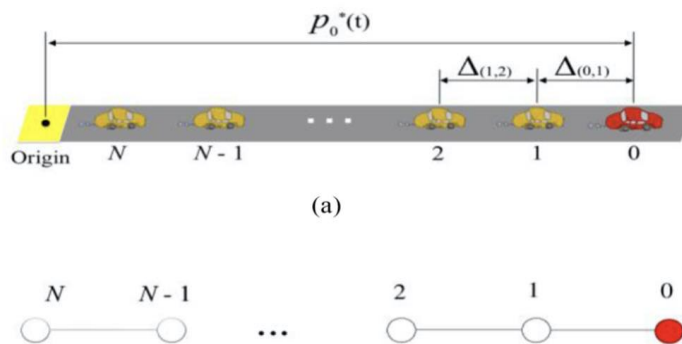


Figure 73 Simple formation

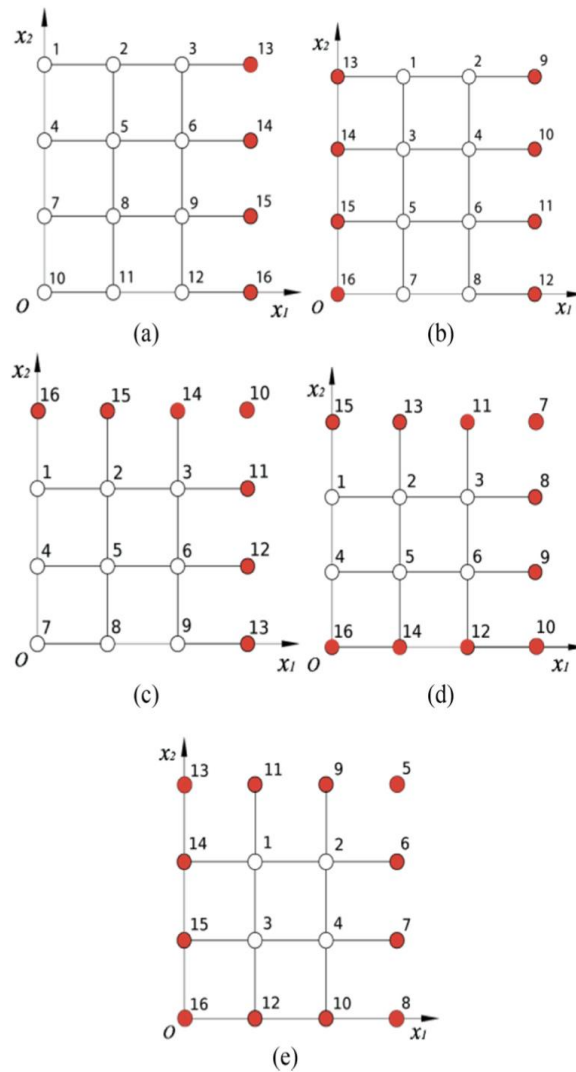


Figure 74 2-D Lattice graphs with different boundary conditions

For a network of DCPS where the interaction topology of CAVs is described by information graphs, it is very challenging to obtain an analytic formula for the eigenvalues of the Laplacian of an arbitrary information graph. However, if simple lattice topology is adopted, it is possible to obtain the exact formulae for the smallest eigenvalue of its graph Laplacian [223]. Therefore, performance metrics, such as network convergence rate and sensitivity, can be studied under different disturbances scenarios.

Machine learning, deep learning, and artificial intelligence- based techniques are indispensable for autonomous cars. The main reason for the significance of these technologies is the unpredictable environment and behaviour of the surrounding objects. As we have mentioned before, most of the computer vision related algorithms and mechanisms such as object detection, perception, scene identification, reconstruction, and estimation use both machine learning and deep learning mechanisms. In addition to the previous contributions of machine learning, software testing is also aided by machine learning techniques in autonomous cars. In traditional software, the operational logic is written manually

and tested over a series of test cases whereas in Deep Neural Network (DNN)- based software, the software learns and adapts with the help of large data sets.

One of the crucial aspects of autonomous cars is perception and it is a good candidate for applying deep learning models. The actuation of the autonomous car heavily depends on perception and therefore, it is important for autonomous cars to mimic the human-like perception capability. Deep learning models also contribute to the processing of massive sensory data in order to make informed decisions. In addition to perception, other functional requirements of the autonomous car that are supported by deep learning include, but not limited to, scene recognition, object (obstacle, car, pedestrian, and vegetation) detection and recognition, human activity recognition, environment recognition, road signs detection, traffic lights detection, and blind spot detection. The popular deep learning models used in autonomous car technology to achieve the aforementioned goals include end-to-end learning, CNN, deep CNN, Fully Convolutional Network (FCN), DNN, belief networks, Deep Reinforcement Learning (DRL), Deep Boltzmann Machines (DBM), and deep autoencoders.

The combination of the DNN and the model predictive control (MPC) makes it possible to simplify the objective function which reduces the computational load of the optimization. At the same time, the MPC formulation makes it possible to include criteria such as control, state-limitations, and comfort. The loss function in the MPC is a combination of collision-risk together with a function of vehicle and obstacles velocities at the optimization time instant. This formulation allows the method to be used in unknown environments in the presence of static and dynamic obstacles.

In [225], the MPC method is introduced for autonomous driving. It uses a deep learning technique for velocity-dependent collision avoidance in unknown environments. At each optimization iteration, by using the data collected by LiDAR sensor, the autonomous vehicle obtains the position, size and velocity information of key obstacles. Then, an ensemble of deep neural networks are used in order to estimate the probability of collision that is used by classical controllers to reduce the velocity in high collision-risk areas and concentrate on the task when probability of collision is low. The collision cost is a product of the probability of collision and vehicle velocity in the directions with high collision-risk. These directions are found using the relative velocity between autonomous vehicle (AV) and obstacles which form collision cones for each key obstacle. Then, according to minimum distance between vehicle and final destination, maximum comfort and minimized velocity in high-collision risk regions, an MPC problem is solved which outputs the optimal acceleration and steering rate of the vehicle.

The dynamic obstacle avoidance optimization method minimizes the velocity in the obstacle cones where the probability of collision is high or in unfamiliar environments and increases the velocity when probability and variation in predicted values of the ensemble are low. The control input is obtained by solving the finite-horizon optimal control problem at each sampling time instant. Each optimization produces an open-loop optimal control trajectory and the first portion of this control trajectory is applied to the system until the next sampling time.

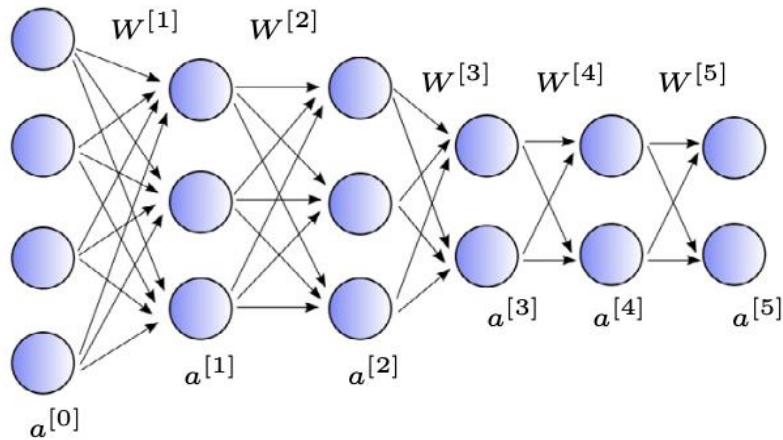


Figure 75 Fully connected multilayer neural network

A deep neural network, as shown in Figure 75, could be constructed by a fully connected multilayer neural network. The input layer would be the union of LiDAR sensor measurements, vehicle's state and control inputs for each point from the predicted optimal trajectory.

However, learning the optimal trajectory for an autonomous vehicle navigation problem by using only supervised learning is a hard task that usually does not lead to a good performance. The environment in which the autonomous vehicle operates is dynamic and, if the vehicle's trajectory is planned from the start position to the destination, small inaccuracies in the prediction from the learning model can cause vehicles collision. However, if the prediction of the vehicle movement is performed for a short horizon, as it is done in MPC, and information about probable collision is added to the controller, this can lead to reliable and safe vehicle movement.

In [226], a model-predictive fuel-optimal control scheme is proposed. The control policy is more precise and, thereby, induces higher car- following energy efficiency, in contrast with commonly myopic decisions in the case of V2V/V2I communication loss. State trajectories of the leading vehicle are harnessed by V2V/V2I communication, and the dependency of continuously variable transmission efficiency on its operating conditions is incorporated. To minimize the fuel consumption per unit distance over each prediction horizon, the objective function of the optimal control problem is solved by a sequential quadratic programming algorithm.

### 6.5.3 Connected and collaborative industrial robots

Robots are increasingly linked to the network and, thus, not limited by onboard resources for computation, memory, or software. Internet of Things (IoT) applications and the volume of sensory data continue to increase, leading to a higher latency, variable timing, limited bandwidth access than deemed feasible for modern robotics applications [230][231]. Moreover, stability issues arise in handling environmental uncertainty with any loss in network connectivity. Another important factor is the security of the data sent and received from heterogeneous sources over the Internet. The correctness and reliability of information has direct impact on the performance of robots. Robots often collect sensitive information (e.g., images of home, proprietary warehouse and manufacturing data) that needs to be protected.

In the past years, robots were mainly conceived to work alone in highly structured environments, replacing humans in carrying out activities that were repetitive, dangerous or requiring high precision. In such contexts, robots were physically segregated from humans by means of barriers. Only recently the topic of human-robot collaboration has attracted much attention. Such interest is motivated by the Industry 4.0 paradigm, which considers as a fundamental pillar the massive presence of robots in production plants, cooperating with humans. The factories of the future will adapt their behaviours, reacting to rapidly changing production plants. In this scenario, robots can no longer be adopted to simply accomplish repetitive tasks. Instead, humans and robots will both adapt and synchronize in many ways, collaborating to accomplish common tasks.

Clearly, one fundamental requirement for cooperation is the safe coexistence in a shared space. To this purpose, a new generation of robots, called cobots, were specifically designed to be deployed in collaborative workplaces, and many works started addressing the problem of safe motion planning. However, the majority of the developed approaches is classifiable as reactive: the robot is slowed down along its nominal path or it is forced to undertake local dodging manoeuvres, in case imminent collisions are detected [235][236]. A more sophisticated approach is to compute proactive trajectories, i.e. trajectories designed to reduce in advance the risk of collision with humans, without waiting for the situation to become critical [237]. Here, we adopt the paradigm of cobots in the concept of fog robotics, presenting the best practices and state-of-the-art on this topic [238].

The computation of proactive trajectories requires an estimation of which human activities are likely to be executed simultaneously with a specific robotic one. This kind of modelling is not required by reactive approaches, which compute paths that are only locally optimal. On the opposite, proactive paths aim to be globally optimal, since the whole path is optimized. The reactive and the proactive approaches can be also combined: a proactive path can be computed minimizing the probability of collisions with the human, but then its execution can be managed by a reactive motion controller, enforcing some additional safety constraints.

The typical scenario is a collaborative assembly for which a certain number of activities have to be accomplished to compose some products. Some of them are assigned to the robot and others are undertaken by the human. Suppose also that actions are preassigned to agents, according to their capabilities.

In Figure 76, the entire pipeline of the approach is shown. A depth camera records the motion of the human, whose trajectories are then segmented and stored in different databases, one for each human action. Such samples are considered by Gaussian Process Dynamical Model to compute regressed trajectories for the human motion, which in turn are exploited to compute the corresponding probability clouds and the proactive paths. An adaptive approach is adopted, periodically recomputing proactive paths, according to the most recent data describing the motion of the operator.

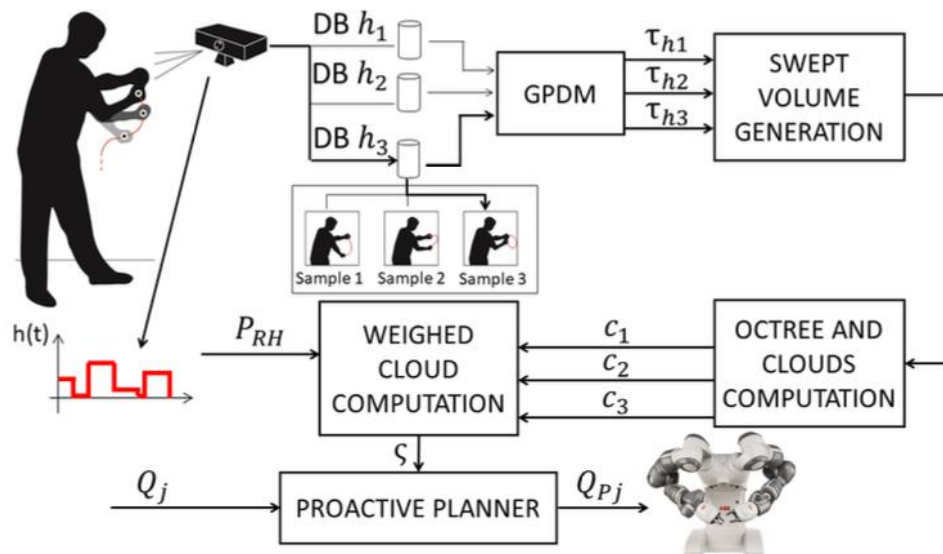


Figure 76 Pipeline overview

In Figure 77, pictures on the left depict an example of regressed trajectory taken from two distinct views. The red box bounding the trajectory is considered for defining a grid of  $g$  points. The volume swept by the arm of the operator, is approximated by the light green convex set in the middle, which in turn is approximated by the dark green OctTree (an OctTree is a tree data structure in which each internal node has exactly eight children).

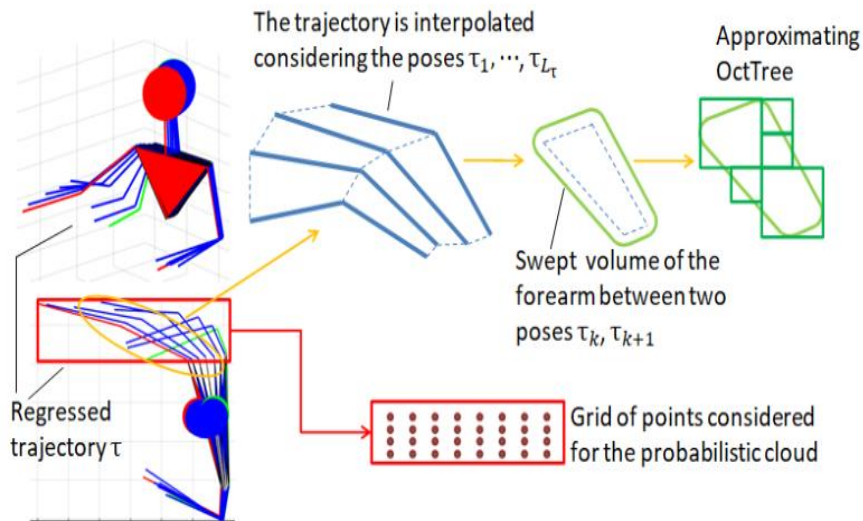


Figure 77 Regressed trajectory taken from two distinct views [119].

In Figure 78, the initial path of the manipulator (red) and the corresponding proactive one (blue). The probability cloud induces the repulsive field depicted with green arrows. The black shape on the right upper corner is a fixed known obstacle.

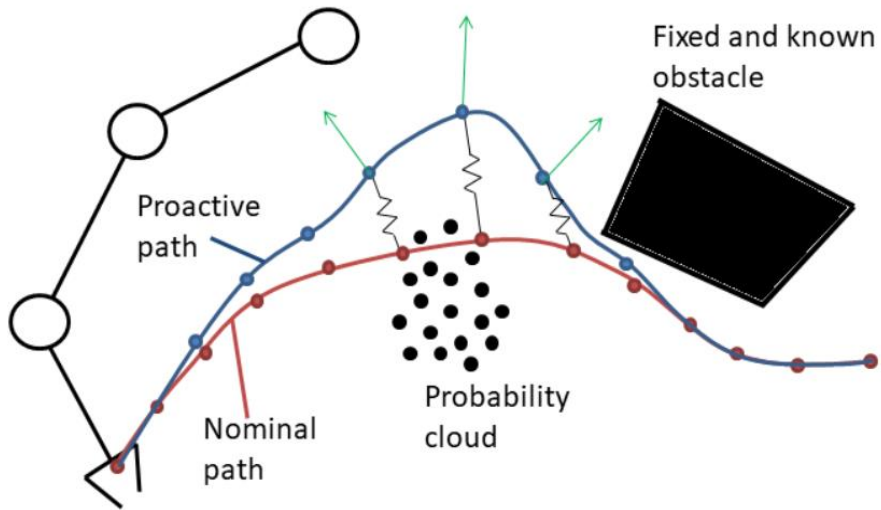


Figure 78 initial path of the manipulator (red) and the corresponding proactive one (blue) [119].

In Figure 79, examples of proactive paths obtained via stochastic trajectory optimization for motion planning (red shapes are the fixed obstacles populating the scene) during the experiments. For every column: on the top, the probabilities considered for the computation of the probabilistic cloud; in the bottom, two distinct views of the proactive paths computed (cyan), compared with the initial nominal one (black). The probability clouds considered for planning are depicted as a series of blue points whose intensity is proportional to the probabilities contained in  $\gamma$ . The minimum for  $J$  is found by applying the STOMP algorithm, which is a sample-based algorithm that iteratively deforms an initial robotic path, for optimizing cost  $J$ . This results in a kind of gradient descend applied to path planning.

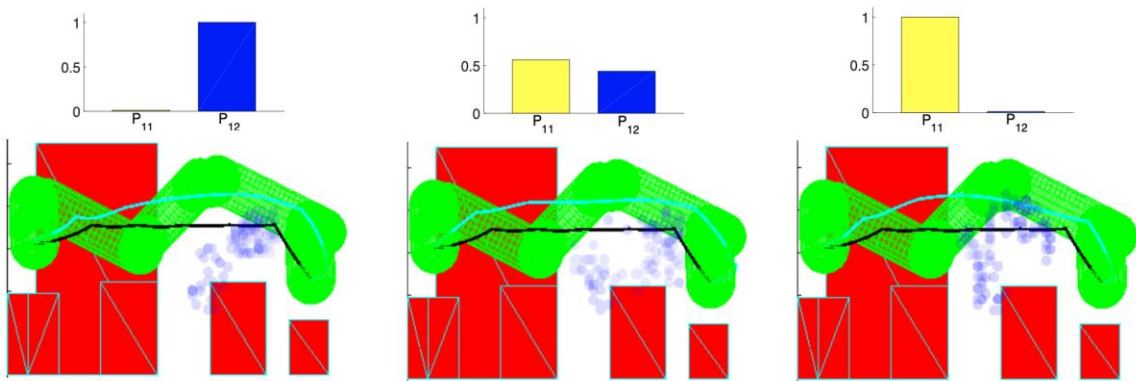


Figure 79. Proactive paths computed for one of the experiments of the first group [119].

In the previous figure, some significant proactive paths computed for one of the experiments of the first group is shown. In that experiment, the operator took all the parts from buffer 2 for the initial cycles, while switched to buffer 1 for the final ones, which explains the values for the probabilities  $P_{11}$  and  $P_{12}$ . Note that probability  $P_{j_i}$  is used to indicate the conditional probability that the human is doing a certain action  $h_i$  while the robot simultaneously executes a particular action  $r_j$ .

When  $P_{11}$  is greater than  $P_{12}$ , the path is more deformed in its initial part (the one for which the robot passes close to the human when this latter is executing action  $h_1$ ). On the opposite, when  $P_{11}$  is lower than  $P_{12}$ , the path is more deformed toward the end.

In [233], a real-time and explicit construction of the configuration freespace is introduced, based on a probabilistic representation of the workspace. It allows to interact with moving obstacles, while dealing with sensor noise. A complete path planner in dynamic environments is constructed, based on the fusion of the skeleton extract from the current configuration freespace and current robot trajectory from previous iteration. It guarantees to produce a safe trajectory, maximizes the joint distance along a specified trajectory, and allows specifying joint safety margin, which is crucial in a collaborative robotic scenario.

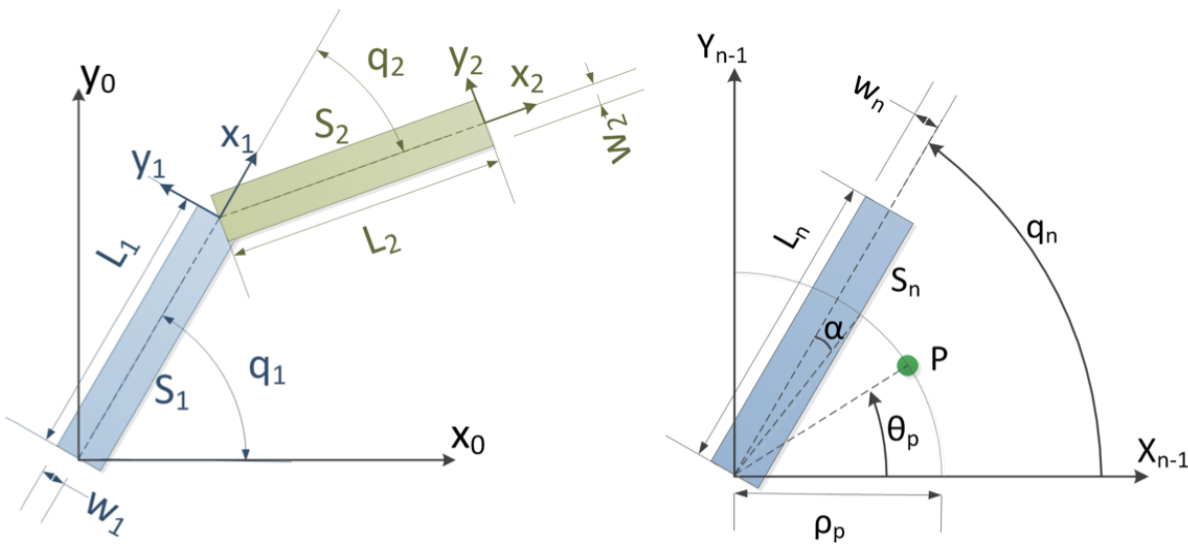


Figure 80 Specific case of a two-joints robot and presents the set of processing allowing a real time and explicit computation of the configuration freespace.

## 6.6 CPS Commissioning and Inter and Intra CPS Communication

### 6.6.1 Inter CPS Communication

Building and operating CPSoS require high degrees of accuracy on the responses to events, and that can only be achieved when the system can react in an efficient manner. These decisions are based on the analysis of the data that the system receives. This kind of bidirectional flow requires enhanced inter CPS communications, low latency and high-reliability networking. The network topology in which this flow of information circulates is dynamic and continuously shifts, due to the movement of the different CPSs (vehicles, robots in a factory), interacting in this environment.

To cope with the fact that CPSoS are made of heterogeneous systems, these systems must follow standardized protocols. There are several ways in which to implement the network needed for the communication between these systems; in other words, to build the inter-communication layers. The industry follows the standards for the inter-communication layers which tend to evolve with the technology, first based on LTE, but evolving later to 5G when the technology has matured.



At the time of writing this document, the two main solutions for communications mechanisms between CPS (or inter CPS communications) are Dedicated Short-Range Communications (DSRC) or Cellular Network Technologies (mainly C-V2X). The evolution of these technologies is mainly driven by the needs of the autonomous and connected cars industry, but similar technologies can be found for cases of Industry 4.0.

DSRC generally refers to wireless technologies used by applications and to exchange of information among DSRC devices in short-range. Examples of these are: a) onboard units (OBUs) located inside the vehicles, b) units placed on the side of the road (RSUs), or c) hand-held devices carried by pedestrians or factory operators. Different spectrum management organizations have allocated radio spectrum bands to be exclusively used for DSRC-based applications. However, studies show that DSRC present poor performance, especially in high vehicle density scenarios. Furthermore, with the allocated DSRC radio spectrum it will not be possible to meet the high data traffic demand for all the new data-hungry applications.

To compensate for these limitations, researchers have been looking into cellular network technologies for cellular-based inter CPS communications. With cellular networks, inter-CPS communications can make use of available infrastructures which provide high capacity, large cell coverage, and wide deployment.

However, the centralized nature of the cellular networks may limit the support for low-latency communications, which can be an issue when thinking about the effectiveness of some of the applications running on the different CSPs. Furthermore, adding all the inter-CPS communications in the same infrastructure can affect and tax the capacity of the cellular networks, which still need to support the growing demand for cellular services.

Consequently, the optimal approach and where the research community has recently been looking into is to use hybrid DSRC-cellular architectures, to support reliable and efficient communications. The following subsections will describe in more detail the technologies used in both options.

#### **6.6.1.1 DCSR - IEEE 802.11x**

DSRC is a standard for vehicle-based communication networks. The scenarios in which this technology can be very beneficial are the ones using applications such as toll collection, services for vehicle safety, or commerce transactions. The idea here is to have a nationwide network enabling communications between vehicles and roadside access points or other vehicles.

Communication via 802.11p goes beyond line-of-sight-limited sensors such as cameras, radar, and LiDAR. It covers V2V and V2I use cases such as collision warnings, speed limit alerts, and electronic parking and toll payments.

Regarding the functional characteristics of 802.11p, it includes short range (under 1km), low latency (~2ms) and high reliability. 802.11p extends vehicle's ability to sense the environment around, even in cases when weather is far from optimal. In other words, makes the vehicle immune to extreme weather conditions (e.g. rain, fog, snow etc.).

Meanwhile, Europe is using 802.11p as a basis for the ITS-G5 standard, supporting the GeoNetworking protocol for vehicle-to-vehicle and vehicle-to-infrastructure communication. The European Telecommunications Standards Institute (ETSI) group for Intelligent Transport Systems is pushing to get ITS G5 and GeoNetworking standardized.

Note that IEEE 802.11p is not dependent on the presence of cellular network coverage, and solutions for onboard units (OBUs) and road-side units (RSUs) are available from different vendors.

### 6.6.1.2 Cellular V2X

As an alternative to IEEE 802.11p appeared the Cellular V2X (C-V2X). The main proponents are the 5G Automotive Association and chipmaker Qualcomm.

An advantage that C-V2X has, when compared with DSRC, is that it has two operational modes, which cover most of the scenarios. First mode is the low-latency C-V2X Direct Communications over the PC5 interface on the unlicensed 5.9GHz band. It is meant for active safety messages such as immediate road hazard warnings or other short-range V2V, V2I, and V2P cases. This mode is the one that competes closely with what is offered by the 802.11p technology, which also uses the 5.9GHz band.

The second mode is used when communicating over the interface, on the regular licensed-band cellular network. This mode is used to handle V2N scenarios like infotainment and latency-tolerant safety notifications for long-range road hazards or traffic conditions. These kinds of scenarios favour C-V2X, because DSRC can only solve this problem by making adhoc connections to roadside base stations, when available.

As a summary of the last two subsections, Table 14 depicts some of the main characteristics of both solutions. As it is evident, they are very similar and, in a way, complementary (to cover all the scenarios).

Table 14 Comparison between the DSRC and C-V2X

	DSRC	C-V2X
<b>Support for low latency communications</b>	✓	✓
<b>Support for network communications</b>	Only via Aps	✓
<b>Can operate without network assistance</b>	✓	✓
<b>Transmission range</b>	Up to approx. 225m	Over 450 m

### 6.6.1.3 Hybrid Architecture

After reviewing both approaches DSRC and C-V2X, we can conclude that is not that simple to pick the best technology. The available infrastructure must be considered and when comparing to IEEE 802.11p, C-V2X is behind in terms of deployment in the V2X market. Testing of current C-V2X technology is underway by many manufacturers. C-V2X has many features in its roadmap, including 5G NR (New Radio) features such as high throughput, wideband carrier support, and high reliability. Using both DSRC and cellular technologies is the most sensible approach, as hybrid solutions exploit the benefits of both DSRC and cellular technologies.

A cellular network can act as:

- a) a backup for vehicular data when V2V multi-hop connections are shattered in a sparse network,
- b) an access network to the Internet,

- c) a backbone network for control messages.

On the other hand, the part of the DSRC spectrum can be utilized by cellular networks in cases that cellular data traffic peaks.

In terms of nodes, there are two types of network nodes in a hybrid DSRC-cellular network: static (i.e., cellular BSs and RSUs) and mobile (i.e., vehicles). These nodes can be organized into a hierarchical or a flat architecture.

In a hierarchical architecture, the cellular/DSRC technology for V2X communications can only be used between network nodes belonging to specific hierarchical levels. For example, public vehicles, such as transit buses and taxis, may belong to a certain hierarchical level, while the rest of private vehicles may be assigned to another level (maybe lower) in the hierarchy. In this hierarchical hybrid architecture, the vehicles are equipped with two interfaces: one for the cellular network and one for communication with other private vehicles via DSRC. For the DSRC part, the antenna can be dedicated (fixed hierarchy level) or generic (can be dynamically reconfigured according to the different needs).

The other type of the hybrid architecture is the so called Flat DSRC-cellular network. In that model the technology to be used for the V2X communications will be determined by the type of data to be transmitted or by certain performance metrics (the quality of service (QoS) provisioning, network data-traffic load, or network coverage). The choice of the type of communication is made based on the service needs. For example, the transmission of control packets may be restricted to the cellular network while the forwarding of data traffic is achieved using DSRC.

#### *6.6.1.4 Characteristics of the required network*

To get beyond the state of the art for the inter-communication layers implementations, SDN and NFV enhanced CPS inter-networking are to be used when possible.

The use of SDN, leveraged by NFV techniques, will allow network service chaining or the creation of slices, where tenant isolation is guaranteed. Linked with capability for the system to create slices across all these heterogeneous CPSs, the system will be able to interact with an orchestrator, which in turn will manage the lifecycle of the different virtual resources, which represent networking resources or computing resources.

Figure 81 depicts a possible example of vehicular network for the connected autonomous cars use case. In Figure 81 the different network hierarchy layers can be seen and how slices can be created to establish communication between different areas in the topology. A similar diagram can be produced for the Manufacturing processes with the Robotics use case.

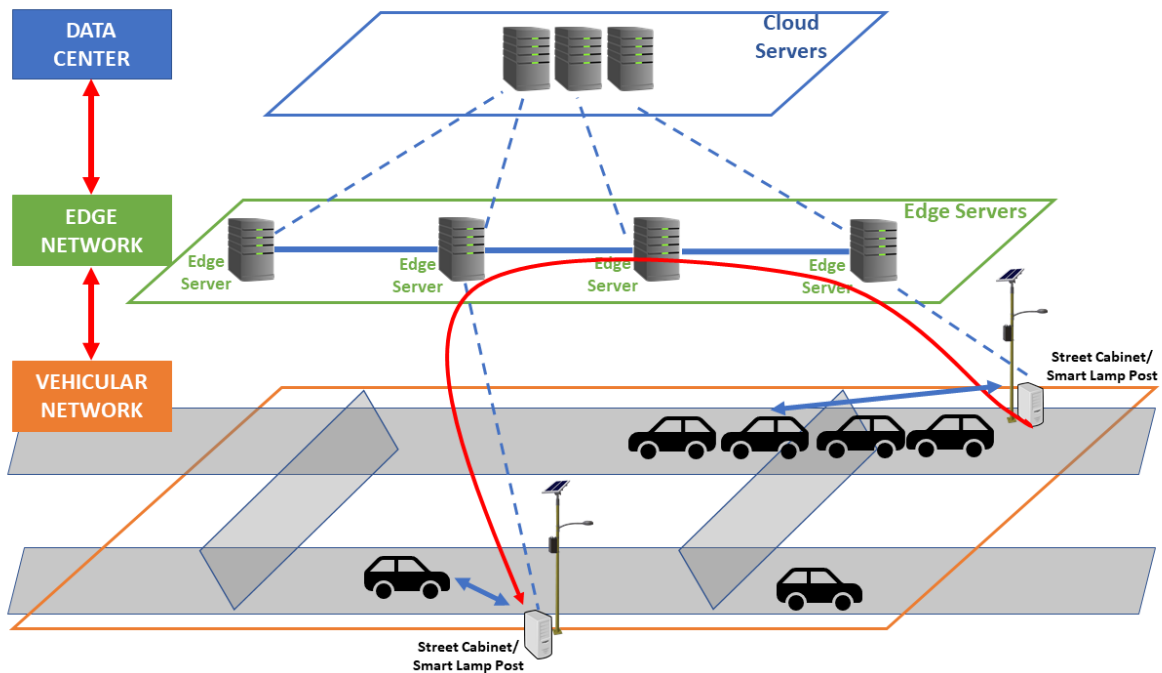


Figure 81 Network hierarchy layers for the connected autonomous cars use case

Note that CPS systems work in an isolated manner and monitoring them is crucial. Establishing a control loop requires the management and configuration of the different CPSoS resources. Frameworks like OpenCL will allow the administrators of the systems to write programs that execute across heterogeneous platforms. Thanks to frameworks like OpenCL, the system can be reconfigured by hot deploying network elements when doing the deployments of the CPS software.

### 6.6.2 Intra CPS Communication

Nowadays, a plethora of short range, ultra-low power wireless communication technologies is available, all aiming to meet the requirements posed mostly by industrial automation monitoring and control, event detection and even data streaming of relatively low bit rates. Therefore, in the context of the CPSoS Aware use cases, a basic investigation has been undertaken aiming to evaluate the prominent examples of such technologies so as to extract useful and practical conclusions with respect to the real requirements anticipated in the CPSoS Aware use cases. The set of technologies under evaluation comprise mature solutions integrated in a wide range of commercial solutions targeting the industrial and automotive application domains. The goal of this effort is mainly to extract and highlight respective pros and cons, enabling the optimum selection of respective technology with respect to specific application scenario requirements.

Since the evolution of wireless technologies has presented significant progress over the last years, our investigation has been expanded not only to the set of the so called “low power – short range” technologies, but also to popular technologies that achieved significant improvements in terms of power consumption which is vital for embedded applications. In this category, the WiFi protocol will be presented.

### 6.6.2.1 IEEE 802.15.4

Respective solutions comprise prominent candidates as they are utilized in several experimental and commercial scenarios. As the title implies, the communication capabilities are based on the IEEE 802.15.4 [242] standard finalized by October 2003 [Part15.4]. Their popularity, gained throughout the years, is based on significant advantages when aiming towards very low power, low complexity, low price and low application demands characteristics.

At the physical layer, IEEE 802.15.4 offers three possible frequency ranges, although the most popular is the 2.4GHz ISM band where 16 channels can be utilized, offering the highest bit rates equal to 250Kbps [Part15.4]. However, it is noted that at each particular time only one channel can be used, thus not being a multi-channel protocol.

Concerning the data transfer approaches, although IEEE 802.15.4 defines approaches for both contention-less and contention-based access schemes, the respective platforms implement and utilize only simple-contention CSMA-based approaches. Following such an approach, all nodes are peers (i.e. there is no coordinator) and sense the transmission medium for two reasons. On one hand, if a node wants to transmit a packet, it senses the medium until it is identified as idle and then transmits the packet. On the other hand, from the receiver perspective, a node senses the transmission medium in order to identify a packet transmission towards itself.

### 6.6.2.2 ZigBee

The most widely deployed enhancement to the 802.15.4 standard is ZigBee[243], which is a standard of the ZigBee Alliance. The organization maintains, supports, and develops more sophisticated protocols for advanced applications. It uses layers 3 and 4 to define additional communications features (Figure 82). These enhancements include authentication with valid nodes, encryption for security, and a data routing and forwarding capability that enables mesh networking. The most popular use of ZigBee is wireless sensor networks using the mesh topology.

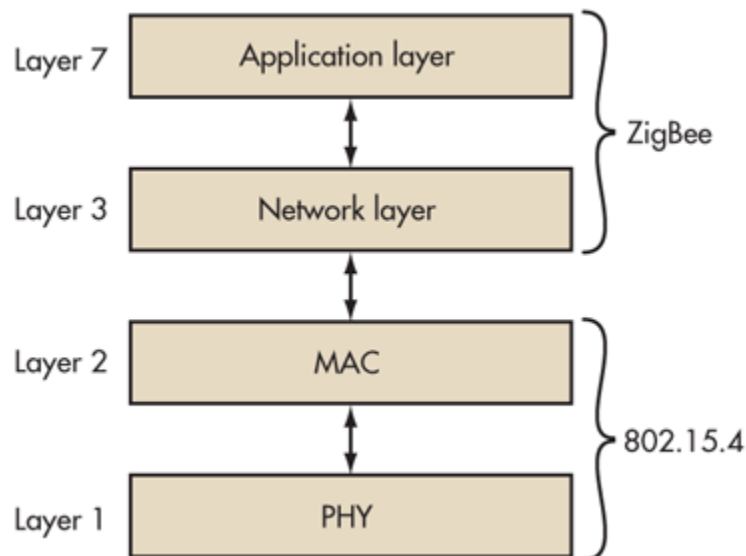
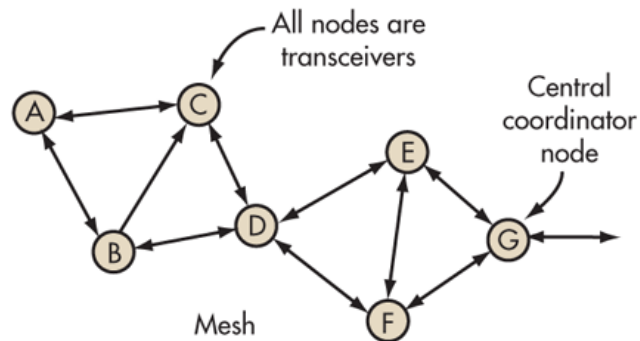


Figure 82 The ZigBee protocol is defined by layer 3 and above. It works with the 802.15.4 layers 1 and 2.

The main benefit of the mesh topology is that any node can communicate with any other node. In case where the destination node is not directly within range relaying the transmission through multiple additional nodes is performed (Figure 83). The network, then, can spread out over a larger area. Furthermore, it increases network reliability as it still functions even if one node is disabled. There are usually alternate paths through the network to sustain a connection. For example, if node A wishes to communicate with node G, it can relay data through nodes C and E. If node C fails, another path is via nodes B, D, and F. ZigBee mesh networks are self-configuring and self-healing.



**Figure 83** In a mesh network, each node communicates with its closest neighbour as conditions permit. Note that there are alternate paths between any two nodes.

ZigBee is also available in a version that supports energy harvesting where no battery or AC mains power is available. And, one of the key benefits of ZigBee is the availability of pre-developed applications. These upper-layer software additions implement specialized uses for ZigBee. Some of these applications include:

- Building automation for commercial monitoring and control of facilities
- Remote control (RF4CE or RF for consumer electronics)
- Smart energy for home energy monitoring
- Health care for medical and fitness monitoring
- Home automation for control of smart homes
- Input devices for keyboards, mice, touch pads, wands, etc.
- Light Link for control of LED lighting
- Retail services for shopping related uses
- Telecom services
- Network services related to large mesh networks

The ZigBee Alliance also offers full testing and certification of ZigBee-enabled products to ensure interoperability. ZigBee has been around for many years now and is widely used. It is a great option for

many applications. For some simpler communication projects, it may be an overkill with its extra complexity and cost. Plain old 802.15.4 may be a better choice in such cases.

### *6.6.2.3 Bluetooth*

Bluetooth[244] is a wireless specification designed to replace cables as the medium for data and voice signals between electronic devices. The specification is defined by the Bluetooth Special Interest Group (SIG) which is made up of over 1000 electronics manufacturers. Intended primarily for mobile devices, Bluetooth's design places a high priority on small size, low power consumption and low costs. The Bluetooth specification seeks to simplify communication between electronic devices by automating the connection process.

Bluetooth radio operates in the unlicensed 2.4GHz Industrial, Scientific, and Medical application (ISM) frequency range. This frequency is already widely used by devices such as microwave ovens, baby monitors, cordless telephones, and 802.11b/g wireless networking devices. In order to avoid interference from these devices, Bluetooth uses a technology called spread spectrum frequency hopping. Spread spectrum frequency hopping changes the transmission frequency up to 1600 times per second across 79 different frequencies. As a result, interference on any one of those frequencies will only last a fraction of a second. This fact coupled with the limited range of Bluetooth radio transmitters, results in a robust signal that is highly tolerant of other devices sharing the same frequency.

Bluetooth devices automatically attempt to communicate whenever one device comes within range of another. Bluetooth devices discover each other and initiate communication via inquiry and page transmissions.

Bluetooth devices have the ability to form adhoc networks. The topology of these networks is both temporary and random. An adhoc network of two or more Bluetooth devices is called a piconet. When two Bluetooth devices initiate a connection, they automatically determine if one device needs to control the other. Generally, the device that initiates the communication assumes the role of master and exercises certain controls over the other members of the piconet which are known as slaves. Upon establishing a piconet, the slave devices synchronize their frequency hopping sequence and system clock with that of the master in order to maintain their connection. A master device can have up to seven slaves. A slave in one piconet can also be the master in another, thus allowing piconets to overlap and interact forming what is known as a scatternet.

Contrary to IEEE 802.15.4-based solutions, where all relative platforms are characterized by analogous capabilities, the platforms in Bluetooth-based solutions can vary significantly depending both on the version of the protocol supported and even more on the specific implementation's characteristics. Therefore, concerning data rates, solutions covering a wide range from 300Kbps up to 1.5Mbps can be found.

### *6.6.2.4 Bluetooth Low Energy*

Bluetooth Low Energy (BLE) [240] represents a different technology from Classic Bluetooth (and in fact incompatible technology). It is being promoted by the Bluetooth Special Interest Group (SIG) and benefitting of the huge success of Classic Bluetooth. It shows significant dynamics compared to analogous technologies being incorporated for example in most mobile devices such as smart phones and tablets, in high percentage. BLE offers high degree of flexibility both concerning implementation

approaches and communication approaches supporting different ways for nodes to communicate through different data structure profiles so as to best fit the application requirements. Both these aspects are critical for the CPSoSAAware objectives, highlighting relative solutions as good candidates for CPSoSAAware purposes.

Bluetooth Low Energy hit the market in 2011 as Bluetooth 4.0. When talking about Bluetooth Low Energy vs. Bluetooth, the key difference is in Bluetooth 4.0's low power consumption. With Bluetooth LE's power consumption, applications can run on a small battery for four to five years. Although this is not ideal for talking on the phone, it is vital for applications that only need to exchange small amounts of data periodically. Just like Bluetooth, BLE operates in the 2.4 GHz ISM band. Unlike classic Bluetooth, however, BLE remains in sleep mode constantly except for when a connection is initiated. The actual connection times are only a few ms, unlike Bluetooth which would take ~100ms. The reason the connections are so short is that the data rates are as high as 1 Mb/s.

BLE's M2M/IoT Applications:

- Blood pressure monitors
- wellbeing devices, smartwatches, etc.
- Industrial monitoring sensors
- Geography-based, targeted promotions (iBeacon)
- Public transportation apps

In summary, Bluetooth and Bluetooth Low Energy are used for very different purposes. Bluetooth can handle a lot of data but consumes battery life quickly and costs a lot more. BLE is used for applications that do not need to exchange large amounts of data and can, therefore, run on battery power for years at a lower cost.

#### 6.6.2.5 WiFi

WiFi[246], by definition, cannot be categorized as a short-range, low-power, wireless technology. However, since it presents significant improvements in power consumption and the “infrastructure less” operation (WiFi-direct delivers peer-to-peer communication without the need of an access point) we include respective solutions in the present investigation.

WiFi is a local area network (LAN) designed to provide internet access within a limited range. It is widely used in home networks and most public places (like coffee shops and airports). WiFi is a star network where there is one central hub and all nodes or devices connect to it. This star topology makes it easy to add or remove devices without affecting the rest of the network. Bandwidth is high—up to 2MHz—which is why it is perfect for most of the application domains. However, the downfall is that it only works if the signal is strong and the client device is close to the access point. Its average range is between 30 to 100 meters. Another drawback is that it is not as power-efficient as the aforementioned low power solutions. For that reason, until recently, WiFi was not applicable if the M2M application includes sensors or other power constrained embedded devices in remote places. However, since the first versions of WiFi, significant improvements have been achieved in that sector with posterior WiFi variations.



Although much faster than the other low power wireless solutions, WiFi shows a variation in speed when it comes to data transfer. Networks defined under 802.11b standard (legacy but popular WiFi version) have a maximum transfer speed of 11mbps, while other versions (like a and g) have a max speed of 54mbps. WiFi variations with specifications that match the use cases requirements (industrial & automotive) follow.

#### 6.6.2.5.1 802.11ac

In 2013, WiFi AC was introduced. AC was the first step in what is considered “Gigabit WiFi,” meaning it offers speeds of nearly 1 Gbps, which is equivalent to 8000 Mbps. That’s This is roughly 20 times more powerful than 802.11n, making this an important and widely used protocol. AC runs on a 5 GHz band, which uses higher frequency and modulation rate, which results into limited range. In 2016, amendments were made to AC [247] to improve its performance.

#### 6.6.2.5.2 802.11ah (HaLow)

To increase the relatively short range of WiFi—specifically for IoT sensors that do not require high data rates—802.11ah[248] was introduced. HaLow is 900 MHz WiFi meant for long-range data transmission. HaLow also theoretically addresses low power consumption. For example, HaLow uses target wake time to reduce the amount of energy a device needs to stay connected to the network. It does this by having devices wake up for very short times at defined intervals—for milliseconds every 15 seconds—to accept messages. This is similar in concept to how extended discontinuous reception (eDRX) works to help LTE-M[249] save power.

Benefits:

- It penetrates through walls and obstructions better than high frequency networks like 802.11ad, which will be discussed below.
- It is ideal for short, bursty data that do not consume a large amount of power and need to travel long distances such as smart building applications, like smart lighting, smart HVAC, and smart security systems. It would also work for smart city applications, like parking garages and parking meters.

Downfalls:

- There is no global standard for 900 MHz. Right now, 80% of the world uses 2.4 GHz WiFi, which means you can connect on these global standard bands anywhere in the world. But because there is not a global standard for 900 MHz, HaLow is very U.S.-centric.
- 802.11ah is available but unused. HaLow was released in 2016 but, presently, there is no product on the market that uses this standard. Partially, this may be due to the lack of a global standard, but it is also likely due to the fact that there are competing technologies on the market that better address the needs of IoT. For example, Symphony Link has an even lower data rate, which increases the link budget. HaLow needs to support IP traffic, but Symphony can address TCP/IP traffic over the air.

### 6.6.2.5.3 802.11s

IEEE 802.11s[250] is Wireless LAN standard and an IEEE 802.11 amendment for mesh networking, defining how wireless devices can interconnect to create a WLAN mesh network, which may be used for relatively fixed (not mobile) topologies and wireless adhoc networks. 802.11s extends the IEEE 802.11 MAC standard by defining an architecture and protocol that supports both broadcast/multicast and unicast delivery using radio-aware metrics over self-configuring multi-hop topologies. 802.11s inherently depends on one of 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, or 802.11ax carrying the actual traffic. One or more routing protocols suitable to the actual network physical topology are required. 802.11s requires Hybrid Wireless Mesh Protocol, or HWMP, be supported as a default. However, other mesh, ad-hoc (Associativity-Based Routing, Zone Routing Protocol, and location based routing) or dynamically link-state routed (OLSR, B.A.T.M.A.N. [252]) may be supported. A mesh often consists of many small nodes. When mobile users or heavy loads are concerned, there will often be a handoff from one base station to another, and not only from 802.11 but also from other networks (GSM, Bluetooth, PCS and other cordless phone). Accordingly, IEEE 802.21, which specifies this handoff between nodes both obeying 802.11s and otherwise, may be required.

### 6.6.2.6 Summarization

Nowadays - and most probably in the future - there is a variety of different and diverse communication technologies claiming a part of the short range, ultra-low power, and wireless communication market. In this section a basic evaluation research was presented aiming to evaluate suitability with respect to the CPSoSAAware communication requirements and to reveal specific characteristics as well as pros and cons. What is clear from studying the considered technologies is that each offers unique characteristics distinguishing each solution from the other and making each technology a better fit compared to the others considering different communication requirements both functional as well as non-functional. Therefore, we present a table of characteristics and an attempt to rank considered technologies.

**Table 15. Wireless Communication Technologies Ranking**

	IEEE 802.15.4	Bluetooth	BLE	WIFI
<b>Low Power</b>	Very good	Good	Excellent	Medium
<b>Link Capacity</b>	Medium	Very Good	Very Good	Very Good
<b>Affected by communication competition</b>	Medium	Very Good	Very Good	Very Good
<b>Multi-hop communication support</b>	Excellent	Medium	Very Good	Medium
<b>Dynamic Topology</b>	Excellent	Medium	Very Good	Medium

<b>Time constrained performance</b>	Yes	Yes	Yes	Yes
<b>Low Cost</b>	Very Good	Good	Excellent	Excellent
<b>Market Widespread</b>	Medium	Excellent	Excellent	Excellent

## 6.7 Extended reality tools for improving safety and situational awareness of the human in the loop

### 6.7.1 Situational awareness of the human in the loop

Situation Awareness is used to describe the level of awareness that operators/drivers/users have of the situation in order to perform tasks successfully [368]. Based on the definition in [369], situational awareness needs to include four specific requirements:

1. to easily receive information from the environment.
2. to integrate this information with relevant internal knowledge, creating a mental model of the current situation.
3. to use this model to direct further perceptual exploration in a continual perceptual cycle.
4. to anticipate future events.

Taking these four requirements into account, situational awareness is defined as the continuous extraction of environmental information, the integration of this information with previous knowledge to form a coherent mental picture, and the use of that picture in directing further perception and anticipating future events.

Nowadays, CPSoS leverage of augmented, virtual, or mixed reality to facilitate many processes like training, planning/analysis, and situational awareness simulations. Through a model-driven contextual interface, trainees can experience a virtual representation of a real-world facility and participate in realistic training. AR is being used to explore and visualize new security concepts. Simulations also often incorporate a mixture of these live and simulated assets.

Real-time data channels that feed real sensory data to the virtual representation and vice-versa are also very important in CPSoS. Autonomous systems functionality, real-time data of sensors, previously stored information, and humans with AR technology work all together simultaneously to improve overall situational awareness. AR presents a compelling opportunity to improve user's situational awareness by displaying elements of the virtual world's model, including entities that are not in the responder's line-of-sight, and predictive analytics based on a simulation's ability to run faster than real-time. Model-based situational awareness is required for improvements in analysis, rehearsal, and training.

An implementation of AR tools and technologies into a situation awareness application can be utilized:

- in large distributed systems with decentralized management and control.
- to handle large amounts of data in real-time.
- to monitor the system performance.

- to detect faults and degradation.
- to learn operation patterns from past examples, auto-reconfiguration, and adaptation.
- to analyse user behaviour and detect needs or anomalies.
- to improve the situational awareness of the Operators when they have to accomplish a specific mission.

Extended reality application for situational awareness requires an additionally research in several other areas in order to achieve a successful integration. More specifically:

- sensor level
- supporting algorithms for information extraction
- decision support
- automated and self-learning control
- dynamic reconfiguration features

To operate a system of systems efficiently and robustly there is a need to detect changes in demands and operational conditions (both equipment and other external factors) and to deal with anomalies and failures within the system. This can be achieved by introducing much higher levels of data acquisition throughout the CPSoS and the use of this data for optimization, decision support, and control.

Endsley's model is the most common theoretical framework representing the situation awareness, as presented in the next figure. According to Endsley's model, situational awareness involves three phases that have an increasing trend of awareness as the information is processed at a higher level [374], [378].

- Level 1 – Perception of the elements in the environment
- Level 2 – comprehension of the current situation
- Level 3 – Projection of future status

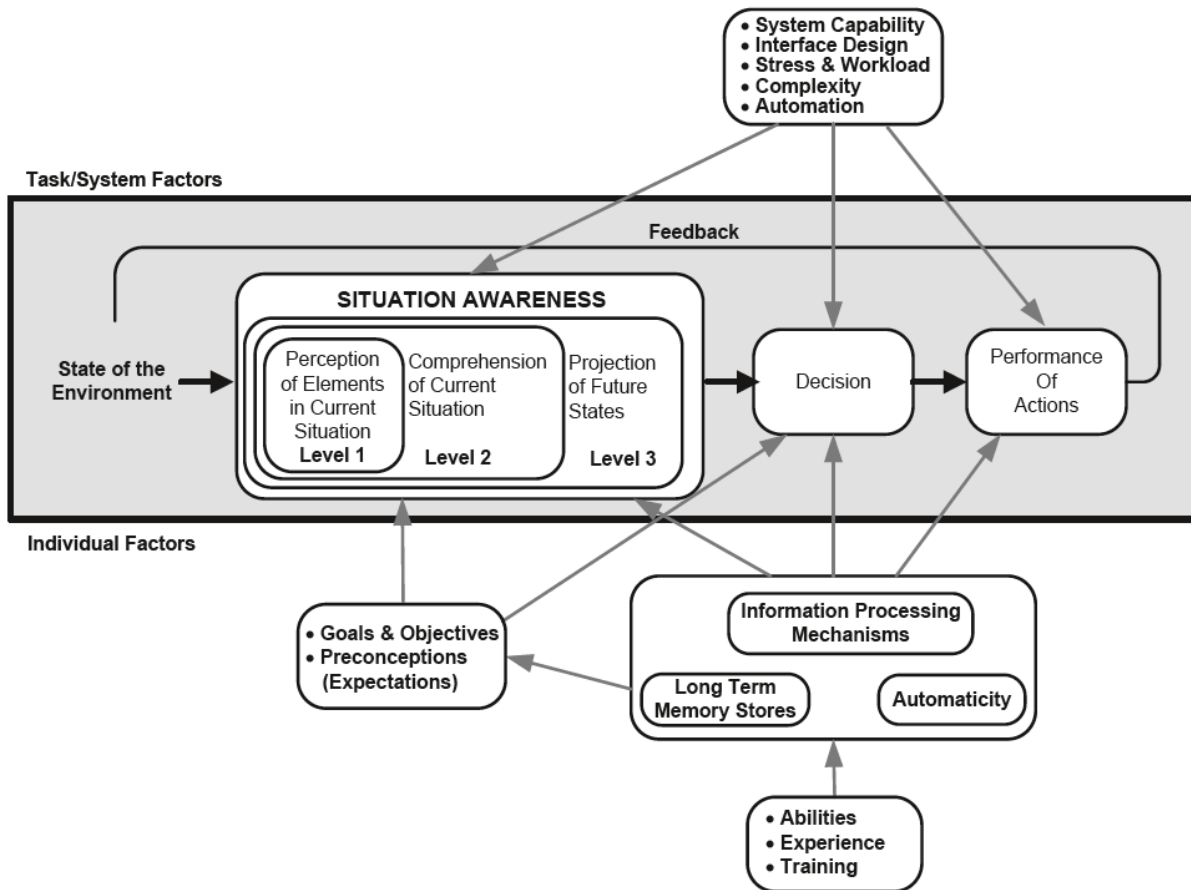


Figure 84 Model of situation awareness proposed by Endsley.

## 6.7.2 Related works and use cases using AR to increase situational awareness

### 6.7.2.1 Situational awareness in a distributed collaborative AR environment

In this example, a service technician needs external support for a specific problem or process optimization. This can be realized with a tele-maintenance infrastructure where the expert provides a remote solution to the AR device of the technician allowing to have his/her hands free. AR technology can be applied in distributed collaborative augmented reality environment for the situation awareness of all worker members in industrial environments that is based on a centralized architecture for data communication, to support virtual co-location of users, consisting of four major components [372]:

- Local user AR support
- Remote user AR support
- Localization and mapping
- Shared memory space

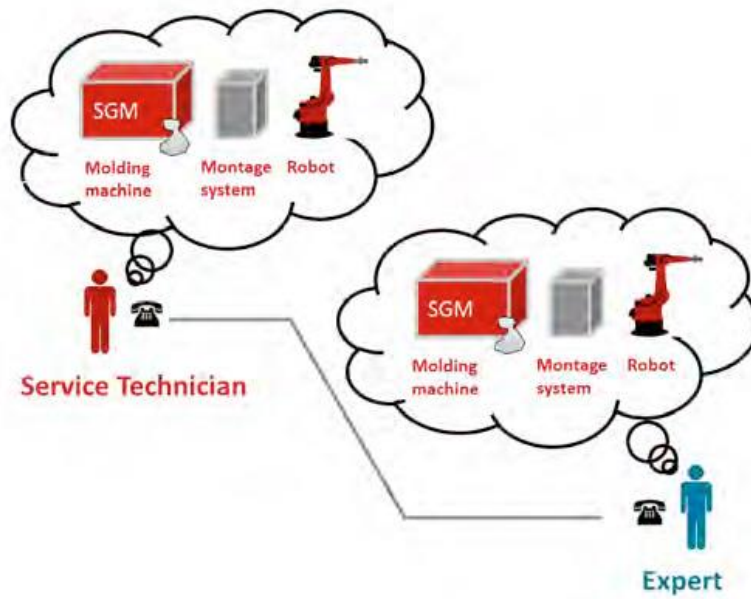


Figure 85 Comprehension problems in tele-maintenance scenarios.

### 6.7.2.2 Driving Situation Awareness in vehicles

Applications related to situation awareness for drivers consist of dynamic information that needs to support different tasks simultaneously. For example, drivers must perceive and comprehend their current situation, the position and speed of their own and other vehicles, road conditions, and the condition of their vehicles. Additionally, drivers have to estimate how these variables might change through time in order to make good decisions about navigation, manoeuvring, and other driving subtasks [374].

AR-based display systems are helpful in reducing driver distractions, thereby increasing driver safety, and provide intelligent interactions for enhancing driver convenience. Such types of systems offer information about the driving situation and warn the driver through AR devices. They consist of several sub-modules like sensors, vehicle/pedestrian recognition, vehicle state information, driving information, time to collision (TTC), threat assessment, warning strategy, display modules, etc. [375].

With the rapid development of driverless cars, Vehicle-to-Vehicle systems using AR technologies have the potential to communicate with other vehicles to improve tracking performance and make better prediction or detection of critical events. [376].

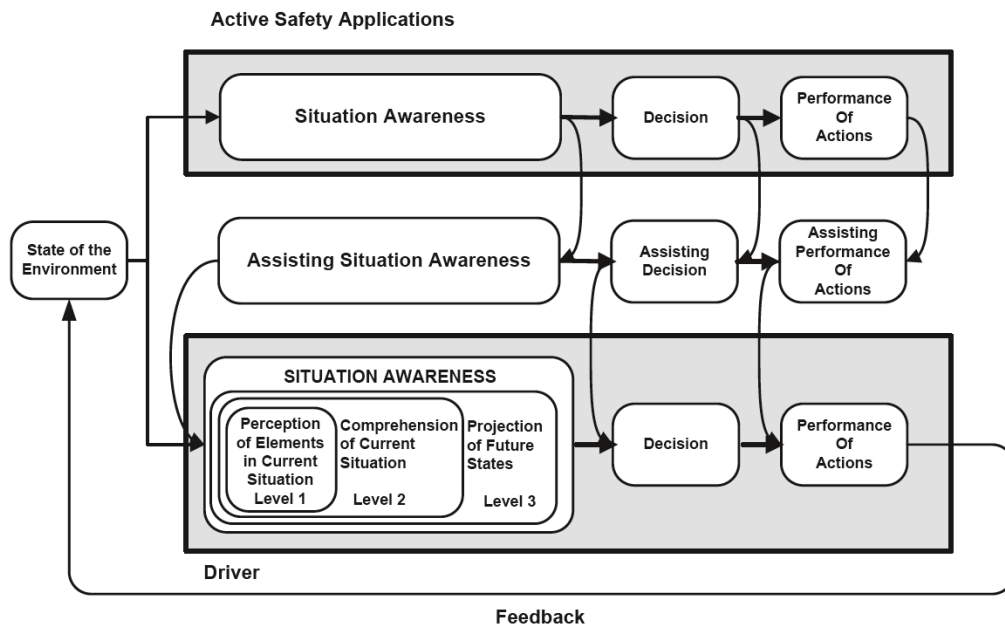


Figure 86 Support of Driver Assistance System

## 7 Techniques and tools for monitoring CPSs infrastructures

As introduced in Section 2, the fifth phase of the CPSoS Aware Lifecycle is devoted to monitoring the CPSoS, according to some predefined Key Performance Indicators (KPIs). This serves two purposes: on the one hand, monitoring KPIs gives users of the CPSoS visibility on the quality and efficiency offered by the system at operational level; on the other hand, monitoring system operation is part of the continuous improvement process of CPSoS Aware that uses the values obtained from KPI monitoring as input for the Simulation and Design phases and, thus, learning, reconfiguring and adapting the CPSoS to changes in the context. This section firstly presents different algorithms and techniques for monitoring KPIs from a generic perspective, and outlines software solutions that can be used in the context of CPSoS Aware. Secondly, the section focuses on the cybersecurity aspects of CPSoS, describing tools and techniques for protecting CPSoS at different levels and supporting the monitoring of security related KPIs.

### 7.1 Algorithms and techniques for monitoring KPIs in CPSs

KPIs are used as metrics to indicate the progress toward an expected result. In CPSoS, KPIs provide objective evidence for operational improvement to achieve the desired results, create an analytical basis to help for better decision making, offer a comparison that measures the degree of performance change over time, and help to give attention on what really matters most. Additionally, the use of KPIs includes setting targets (the desired level of performance) and tracking progress against that target. KPIs help to measure and compare the progress for the final goals and can indicate if the goals have been realized or not. Finally, KPIs create a continuous improvement scenario as they are automated in nature.

There are two ways of reaching KPIs. The first is the direct way, meaning the goals can be directly associated with a measurable entity, such as the number of finished products. The second approach is the indirect way, where the KPI needs calculations before showing meaningful information. An example is the cycle time in a production line, where a calculation between the starting time and end time for a product is done. That difference will give the cycle-time for a product to go through the production. KPIs require data from several processes and machines. However, acquiring these data in a CPS is more straightforward than traditional manufacturing sites due to the connected nature of CPSoS. In the following paragraphs we will present algorithms, tools, and techniques for monitoring KPIs in CPSoS.

#### 7.1.1 Monitoring and measurements

Monitoring is an extremely important task in a CPSoS, since it is used to ensure that the entire system, as well as the individual subsystems, operate at an acceptable level of quality and efficiency. However, it is crucial to use appropriate measurements that can satisfy some predefined quality and functional criteria. It is important to mention here that the used measurement has to be well planned and its usage well defined in order to fulfill the initial objectives.

The main concerns, which need to be decided and designed based on the industrial partners' needs during KPI monitoring, are:

- i. when the KPIs should be measured
- ii. the frequency of monitoring
- iii. the distribution of the measurements

To deal with this issue, a first approach for managing the monitoring in the proposed framework has been to add the functionality to "start" and "pause" monitoring in order to deal with data management.



However, the frequency or the distribution of the measurement is an issue that should be further investigated until the final implementation of the proposed framework.

### 7.1.2 Online and offline monitoring

The measurement of KPIs could be achieved using different types of sensors. It is possible to have online monitoring in which real-time information is required, and offline monitoring. The online monitoring supports on-demand monitoring of the CPSoS statement while the offline monitoring supports reports and evaluation. At this point, it should be mentioned that online monitoring expresses the ad-hoc retrieval of information using web-based communication technologies. An illustrated example could be the online monitoring of energy consumption via wire sensor networks. In contrast to that, offline monitoring could be the monitoring using human sensors where, via data mining or questionnaires, it is possible to extract “static” results at the end of each procedure/step.

### 7.1.3 Algorithms for the selection of an appropriate KPI

Another aspect that needs to be addressed and it is related to measuring KPIs is the software tools. Through open-source or existing software tools based on simulation environments, multiple KPIs can be measured and monitored. As a result, the different sources for measuring KPIs are the following:

- Hardware
- Software
- Human-related factors

The majority of projects dealing with monitoring and diagnosis of CPSs relies on models created by human experts. Nevertheless, these models are rarely available, hard to verify and maintain, and often incomplete, so the need for data-driven algorithms seems to be a necessary and inevitable solution. Some conventional algorithm that are used, are:

- Support vector machine [379]
- Linear regression [380]
- Lasso regression [381]
- Ridge regression [382]
- Elastic net [383]
- K nearest neighbour [384]
- Robbins-Monro control [385]
- Kiefer-Wolfowitz algorithms [386]

The traditional process of experts selecting the appropriate KPI requires considerable experience in the domain. A solution is to detect KPIs based on historical data using data mining algorithms, and analyse the relations between factors that affect the performance. The main advantage of these approaches is the selection of the right KPIs without prior experience in the domain (data-driven)[387].

### 7.1.4 Quality criteria regarding KPIs [388]

In the following Table 16, some quality criteria related to KPIs are presented [260].

**Table 16 Quality criteria and the corresponding motivation regarding KPIs**

<b>Criteria</b>	<b>Motivation</b>
<b>Frequent Measures</b>	Since data needs to be up to date and timely, frequent measures is a requirement for quick actions.
<b>Acted upon</b>	It is useless to have a KPI that is not acted upon when changes occur, then the motivation behind the measurement has to be re-evaluated.
<b>Clearly actionable</b>	A KPI must be actionable, as actions needs to be taken when changes occur.
<b>Clear responsible</b>	There must be a stakeholder responsible for the KPI and the required action.
<b>Significant impact</b>	If a KPI does not have a significant impact, it is not a KPI, it is a PI.
<b>Not having too many</b>	Having too many KPI can be confusing for stakeholders as the key focus areas is then blurred out.

### 7.1.5 KPI Visualization

The final stage during the building of a KPIs evaluation is the visualization of the results. Thus, visualization is a part that needs to be further analysed in order to have a deep impact contribution on network design and reconfiguration. In Figure 87, we present examples of different visualization graphs that can be used for the effective representation of the information.

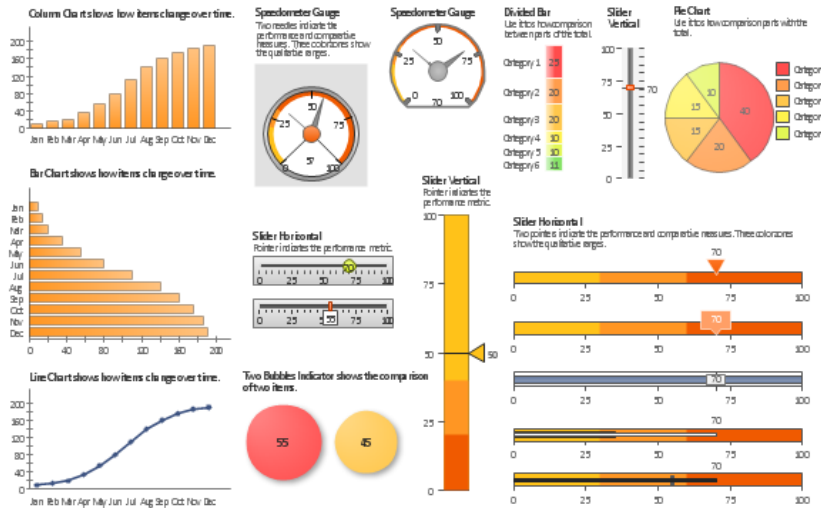


Figure 87 Example of different visualization graphs

Following and enhancing the aforementioned visualization graphs and methods, various methods and graphs can be used:

- Dashboards
- bullet graph
- 3D Scatter plot
- Slider KPI charts
- Speedometer KPI charts
- Traffic light KPI charts
- Charts use case

### 7.1.6 AR application for KPI visualization on a full productive line


AR technologies, as an information visualization tool, can be used in manufacturing industries where real-time reports are essential for the decision-making process. This type of industry must guarantee that its operations are being monitored continuously in order to avoid failures or other critical situations. More specifically, AR technologies could be used to display, for instance, KPIs of each workstation inside an industrial plant, gathered from measuring devices and other sensors, as presented in Figure 88. The implementation of this system results in a dynamic tool that allows reducing inspection times.





Figure 88 Conceptualization of using AR application for KPIs visualization on a full productive line [389].

### 7.1.7 Software and tools for KPI monitoring

A KPI tool is a reporting solution that is used to track, monitor, and generate actionable insights from KPIs in order to achieve sustainable goals. In a CPSoS environment, it is crucial to utilize every possible resource that can be obtained, and KPI reporting tools are on top of the list. By using a KPI dashboard, the selected performance indicators will always stay up to date, ensuring the health and good operation of the CPSoS. Additionally, these tools provide numerous features and benefits that are invaluable for the CPSoS. There is a plethora of software and tools that can be used for KPI monitoring.

Klipfolio	<a href="https://www.klipfolio.com/">https://www.klipfolio.com/</a>
	<p>Klipfolio is a cloud-based application for building and sharing real-time dashboards on web browsers, TV monitors, and mobile devices. It provides immediate visibility of the most important data and metrics. It gives to users the freedom to access all data wherever they are. Some top features of Klipfolio are:</p> <ul style="list-style-type: none"> <li>• Pre-built data visualization</li> <li>• Unlimited users</li> <li>• Private reports</li> <li>• Access your data from anywhere</li> <li>• Private links</li> <li>• A powerful data visualization editor</li> <li>• Provides the ability to create math functions</li> </ul>
SimpleKPI	<a href="https://www.simplekpi.com/">https://www.simplekpi.com/</a>

	<p>SimpleKPI is designed to be incredibly simple and easy to use. It is a complete package, suitable for industry, that is capable of tracking a wide range of metrics including financial, marketing, operational, and service metrics. Some top features of SimpleKPI are:</p> <ul style="list-style-type: none"> <li>• Full-screen option to display KPIs in real-time</li> <li>• Powerful and flexible KPI dashboards</li> <li>• Standard and customizable KPI Reports</li> <li>• Shareable reports, dashboards, and analytics</li> <li>• Streamlined KPI analytics</li> </ul>
<p><b>Bilbeo</b></p>	<p><a href="https://www.bilbeo.com/">https://www.bilbeo.com/</a></p>
	<p>Bilbeo is a web-based KPI tool that helps for improving the performance by automatically transforming simple metrics into an intelligent dashboard. Its auto-populated dashboard is ready to use in minutes and provides actionable recommendations based on data mining techniques, easily to be understood by anyone. Some top features of Bilbeo are:</p> <ul style="list-style-type: none"> <li>• Smart alerts</li> <li>• Leading indicators algorithm</li> <li>• Custom reports</li> <li>• Unlimited dashboards</li> <li>• Collaboration</li> <li>• No setup, coding, or design is required from the user</li> </ul>
<p><b>Geckoboard</b></p>	<p><a href="https://www.geckoboard.com/">https://www.geckoboard.com/</a></p>
	<p>Geckoboard provides an excellent live KPI tracking dashboard. It features a drag-and-drop interface, allowing users to design their own visual dashboards across a range of metrics. It is one of the simplest to set up and use KPI tool. Some top features of Geckoboard are:</p> <ul style="list-style-type: none"> <li>• Pull live metrics from popular business tools without requiring any technical know-how</li> <li>• Visualize metrics from databases, in-house systems, and third-party software</li> <li>• Allows you to display full-screen dashboards</li> <li>• Quickly share links in an email</li> <li>• Use a drag-and-drop interface</li> </ul>
<p><b>Salesforce</b></p>	<p><a href="https://www.salesforce.com/eu/?ir=1">https://www.salesforce.com/eu/?ir=1</a></p>
	<p>Salesforce KPI is a measurable performance metric used to monitor, analyse, and optimize customer relationship management (CRM). It is one of the most popular solutions for KPI tracking. It allows users to build personal dashboards and it offers different options to find the perfect mix.</p>

Tableau	<a href="https://www.tableau.com/">https://www.tableau.com/</a>
	<p>Tableau is an excellent KPI tracking tool with great pre-built KPI templates for the users. It is a great tool for data analytics and customer engagement KPIs. It has an intuitive user interface and two features really set it apart: real-time reporting and data blending. It connects directly to databases. Additionally, it can blend data from a range of different sources. The system combines data to create informative and actionable insights. Some top features of Tableau are:</p> <ul style="list-style-type: none"> <li>• Get up and running in minutes and seamlessly add users as your needs grow.</li> <li>• Real-time data blending</li> <li>• Give external teams, partners, and clients simple and secure access to analytics.</li> <li>• Empower site admins to easily manage authentication and permissions for users, content, and data.</li> </ul>
Asana	<a href="https://asana.com/">https://asana.com/</a>
	<p>Asana is a great tool for tracking KPIs as it is accessible to everyone. It also allows users to track individual subtasks that are necessary to achieve a specific KPI. Its dashboard provides users with a range of tactical overviews that anyone can easily track the progress across each of individual projects.</p>

### 7.2 Cybersecurity primitives, monitoring techniques and tools

This section revises the state of the art of methodologies, primitives, tools and techniques for cybersecurity protection and security monitoring of CPSoS. The following subsections are structured in order to address the following questions:

- **What to protect:** the CPSoS, including the elements that compose the system and the technologies that allow the interconnection of these elements in the system, are analysed and modelled from a security perspective. These aspects are addressed in Section 7.2.1.
- **Protect against what:** a review of existing models and taxonomies of threats and attacks relevant to CPSoS. This is addressed in Section 7.2.2.
- **How to protect a CPSoS:** what are the available mechanisms to put in place in the system, in order to fulfil the desired security properties and to protect against relevant threats and attacks. This is addressed in Section 7.2.3.
- **How to monitor security in a CPSoS:** once the security mechanisms are deployed and configured, how to monitor their correct operation to achieve the desired security goals. This topic is covered in Section 7.2.4.
- **How to assess security (KPIs):** proposals to measure how effective are the security mechanisms implemented in achieving the desired security goals. This is outlined in Section 7.2.5.

## 7.2.1 CPSoSAware ecosystem model

The CPSoSAware ecosystem is composed of all the assets of the system that need to be protected both at the system level and the individual CPS level. We can distinguish different classes or domains to group assets in a CPSoS:

- **System:** physical security, control systems, utilities
- **Application:** cloud-based applications, critical applications, business-specific applications, and the data managed by these applications.
- **Communication:** entails network, connection between devices, sensors, gateways. We can distinguish between assets involved in the inter-communication between the CPSs and the system layer (e.g. SDN, 5G), and assets involved in the communication within an individual CPS (intra-communication) such as message queue telemetry transport (MQTT), WiFi, BLE, ZigBee, LoraWAN etc.
- **Device:** assets of smart devices and Cyber Physical System devices that can be considered edge nodes of a system that match the IoT/Industrial IoT paradigm.

### 7.2.1.1 System domain

The CPSoSAware architecture defined in Section 1.3 of the Description of Action (DoA), describes the system layer as that *“in charge of handling the system of system functionality of a CPSoS and provide system level orchestration, control and monitoring of the various CPS”*. Therefore, the critical assets to protect from a system domain perspective would be:

- Controller
- Virtual / Physical infrastructure
- CPS Commissioning component
- Data resources

### 7.2.1.2 Communication domain

Section 6.6 reviewed communication technologies, both for inter-communication between CPS and the system, and intra-communications within the individual CPS. These wireless technologies are more susceptible to security attacks than wired networks because of its inherent higher accessibility characteristics.

The Open Networking Foundation (ONF) identifies four SDN-specific security challenges, related to the controllers and the communications related to the Controller plane, in addition to the traditional attack vectors [390]:

- **Centralized control** is a high value asset to attackers who may try to tamper with common network services or even compromising the controller.
- **Programmability** poses new types of threats related to the programmatic access SDN, presenting security requirements that do not exist within closed administrative domains, such as integrity, protection of third-party data and open interfaces. Within this challenge, the ONF identifies the following security issues:
  - **Traffic and resource isolation:** operators must ensure full isolation of the information exchanged between the business management and real-time control and all others.

- **Trust between third-party applications and the controller:** authentication and different authorization levels should be enforced at the point of application registration to the controller in order to limit the controller exposure
- **Interface security protection on Application-controller plane interface (A-CPI) and Intermediate-controller plane interface (I-CPI):** lack of protection across these interfaces may lead to malicious attacks on the SDN.
- **Challenge of Integrating Legacy Protocols:** it is critical that compatibility be checked before implementing legacy protocols (e.g., Network Address Translation (NAT), BGP) into SDN. It is also important that weaknesses previously addressed by legacy architectures not be repeated or even inflated when building the SDN framework.
- **Cross domain connection:** this is related to the connection of controllers of different providers via the I-CPI, which requires to put in place mechanisms to establish trust relationships and determine appropriate authorization levels to prevent abuse in the secure channel.

### 7.2.1.3 CPS device domain

In this subsection we capture assets of smart devices and Cyber Physical System devices that can be considered edge nodes of a system that match the IoT/Industrial IoT paradigm. Major sources of information for this study are “[391], [392], and” [393] as well as the Asset specifications that is reported in the deliverable of CONCORDIA EU project [394].

According to [393], IoT security, IT security and cyber physical system security share a lot of fundamental principles. However, given the unique nature of the CPS ecosystem it is not possible to apply the same methodologies and principles with the traditional IT security. In fact, many security measures (e.g., TLS/SSL) cannot be adopted due to the resource restrictions in CPS devices. Such devices can run for a very long period of time without supervision and in a hostile environment susceptible to hacking. Patching is almost impossible due to restrictions in terms of interfaces and they can be difficult detect and force upgrades. At the same time, powerful CPS devices, like those in cars, could afford over-the-air (OTA) updates but still many car manufacturers do not use it. CPS devices have a greater impact in case of attacks since they are embedded into physical systems and can cause physical damage. In addition, traditional security is primarily focused on fortifying the perimeter. With the advent of Cloud, mobile devices, and IoT/CPSs, this perimeter is becoming more articulated and almost impossible to define and protect in a traditional way [395]. This requires to re-think current security practices and guidelines.

In addition to [390], a major source of information for this section is the work jointly commissioned by the Cyber Security Agency of Singapore and the Ministry of Economic Affairs and Climate Policy of the Netherlands and their 2019 IoT security landscape white paper [396]. Other sources of information are [398] [399] and [400].

Cyber physical system Assets can be clustered in the following classes:

- **Data Class:** CPS devices are collecting, processing and extract actuation decisions based on data. However, traditional CPSs do not store data for future use but rather pre-process and forward them to some data aggregation point following a streaming approach. In that sense, this data streaming itself can be an asset to be protected. However, in the latest CPS (as part of CPSoS) data aggregation is partially done on the edge of the system (i.e. the CPS devices). Data become an



important asset for this class also considering the OWASP principle of IoT security related to “Data aggregation” [397], which can reveal sensitive patterns.

- **Infrastructure class:** It comprises of all the CPS components that support the intra and inter communication infrastructure of the CPS. It is defined by appropriate network protocols (e.g. MQTT, ZigBee, Modbus, CAN etc.), but also power supply units and batteries.
- **Devices class:** It is the essence of this category and refers to sensors, actuators, as well as firmware driving them. It also includes devices that serve the purposes of aggregating data (e.g., in edge systems) and managing sensors/actuators, as well as embedded systems in general.
- **Decision support class:** CPSs, after acquiring data, are able to transform them and extract metadata information that will be translated through some control logic into actions on the actuators or models. This transformation should be protected and constitutes an asset of the CPS.
- **Management class:** CPS include device management services that provide information on the CPS status, as well as set configurations on CPS level. The management information and mechanisms may include device usage, battery status, and the like, as well as update management, network setup and statistics, and applications and diagnostics.
- **Security and Privacy Mechanism class:** An important asset of the CPS system is the cybersecurity measures that are infused in the CPS Hardware and Software structure. This may include security establishment techniques like access control mechanisms, user authentication, integrity etc. They may also include all the security primitives that support any security service that exists on the CPS.

Each asset class can be further refined in different sub-categories as presented in the following Table 17[390].

**Table 17 Cyber physical system Assets: classes and categories**

Class	Category	Description
<b>Data</b>	In transit	Assets focusing on encapsulating data while they need to be sent to another component/layer.
	At rest	It is mostly associated with the data that temporarily or permanently reside on the CPS, gateways that are streamed as batches (data blocks).
	Aggregated	It is mostly associated with the data that temporarily or permanently reside on the CPS but are collected from various different sensor sources (or other CPSs).
	Credentials	Files including important credentials like certificates tokens.
<b>Infrastructure</b>	Network/ protocols	Network protocol specialized for CPS like MQTT, and Zigbee, CAN bus, Modbus etc.
	Routers/gateways	Networking devices used to provide connectivity via packet forwarding and bridging between different protocols.
	Power supply	External (and wired) or internal via batteries
<b>Devices</b>	Hardware	It includes the physical part of CPS devices like the physical memory, sensors, and physical interfaces.

	Edge CPS/ embedded systems	Computing services on the devices or at the edge, offering interfaces, aggregations, management services.
	Firmware/software	Software installed on the device including low-level software for operating system-level functionalities.
	Sensors/actuators	The subsystems to detect and measure events, and to take a decision based on previously processed information.
<b>Platform</b>	Device interface/services	It includes APIs and services. It is a major target for a number of impacting attacks.
<b>Decision making</b>	Device/edge processing	It refers to data aggregation, an important trend in CPSs, open to a number of issues related to the possibility of revealing sensitive patterns. It may include components that are related to machine learning (increased intelligence) taking place at the CPS level.
<b>Management</b>	Device and network	Management components of CPS devices (e.g., updates). It also considers configurations at any level including networking.
	Device status	Status level monitoring including batteries, usage patterns etc.
<b>Security and privacy mechanisms</b>	Device	It includes access controls, authentication, identification security mechanisms adopted by the device itself. It also includes physical security tokens that provide security services and may contains important security related information
	Infrastructure	It includes security mechanisms that are in place in the CPS and provide security at the level of infrastructures, like firewalling, channel encryption, incident detection etc.

#### 7.2.1.4 Security modelling approaches

Security breaches in software systems can occur from a single line of program code, the level of power consumption by the computer, to simple human mistakes. Model-based techniques for assessing cybersecurity have been at the forefront of recent research in CPS. According to [401] these techniques traditionally stem from dependability and safety analysis. Nicol et al. [402] have stated the need for model-based methods for assessing security that comes from the general area of dependability. Further, Chen et al. [403] have proposed a model-based graph oriented analysis technique for assessing a system for acceptable safety based on a workflow. Kopetz [404] presents the notion of categories of interfaces to model real-time systems. Davis et al. [405] present a framework that extends the notion of dependability to include a possible security violation for the power grid. This utilizes state estimation and is evaluated in a simulated model of the power grid. Security engineers, therefore, need an array of tools at their disposal for dealing with diverse security problems. One of the cornerstones of a security engineer's toolkit is the ability to access transferable design knowledge, often conveniently documented as a pattern. A pattern is a description of a recurring problem and its corresponding successful solution [406]. A pattern can be

described in many different languages. The description of a pattern using a specific modelling language is called representation.

Security patterns are well-understood solutions to recurring security problems [407]. They enable engineers to recognize known vulnerabilities in their design and potential applied solutions. The following is a state-of-the-art discussion on security pattern modelling approaches [408][409]. The discussion is held around three categories because these approaches are repeatedly referred as representative ones that address security in models [410][411]: object-oriented design (UML, SecureUML, UMLsec, Misuse Cases), goal-oriented (KAOS, Secure Tropos, i\*), and problem-oriented (problem frames, abuse frames).

#### *7.2.1.4.1 Design Approaches*

Design approaches are based on the notion that models help requirement analysts in understanding complex software problems and identifying potential solutions through abstraction [412].

- UML

Unified Modelling Language (UML)[413] is a widely used model notation method for mainly software and systems. Although UML does not originally cover non-functional characteristics including security in an explicit way, it is possible to analyse and represent vulnerabilities in the target system.

UML provides a special diagram notation, which can be used for modelling structure and behaviour of any pattern. UML also provides a built-in generic extension mechanism called UML Profile to customize UML models for particular domains by using additional stereotypes, tagged values and constraints for specific model elements. There are several UML profiles for patterns such as the UML Profile for Patterns as a part of the UML Profile for Enterprise Distributed Object Computing (EDOC) specification [414]. However, these existing diagrams and profiles are not specific to security patterns, so they are incapable of representing security concerns with precise semantics explicitly.

- SecureUML

SecureUML is a modelling language based on UML, presented by Lodderstedt et al.[415] SecureUML focuses on modelling access control policies and how these (policies) can be integrated into a model-driven software development process. SecureUML is based on an extended model of role-based access control (RBAC) and uses RBAC as a meta-model for specifying and enforcing security. RBAC lacks support for expressing access control conditions that refer to the state of a system, such as the state of a protected resource. In addressing this limitation, SecureUML introduces the concept of authorization constraints. Authorization constraints are preconditions for granting access to an operation.

The combination of the graphical capability of UML, access control properties of RBAC, and authorization constraints makes it possible to base access decision on dynamically changing data such as time. SecureUML focuses on the design phase of software development.

- UMLsec

UMLsec [416] is an extension of UML that allows an application developer to embed security-related functionality into a system design and perform security analysis on a model of the system to verify that it satisfies particular security requirements. Security requirements are expressed as constraints on the

behaviour of the system and the design of the system may be specified either in a UML specification or annotated in source code.

UMLsec allows the use of automated theorem proving or model checking to establish whether security requirements hold in the design. A Prolog-based tool can be applied to generate a scenario in the form of attack sequences, in case of a design violating a security requirement, of how security requirements may be violated by the design as well as countermeasures to remove the vulnerability. In essence, UMLsec assumes that requirements are already elicited and there exists some system design to satisfy them. Its objective is to establish whether the system design satisfies security properties. The design is then progressively refined to ensure that it satisfies security requirements.

- **Misuse Cases**

Similar to anti-goals [417], misuse cases are a negative form of use cases and, thus, are use cases from the point of view of an actor hostile to the system[418][419]. They are used for documenting and analysing scenarios in which a system may be attacked. Once the attack scenarios have been identified, countermeasures are taken to remove the possibility of a successful attack. Misuse cases have become popular as a means of representing security concerns in system design, although they are not entirely design-oriented. It is worth noting that they are limited by the fact that they are based only on scenarios.

#### 7.2.1.4.2 Goal Oriented Requirements Approaches

- **Secure i\***

Distributed intentions, also known as the i\* representations[420] are used to model and explore goal-oriented requirements of stakeholders in the problem space. Tropos [421] is a process that applies i\* to analyse early requirements in order to procure a validated list of specifications for a solution.

In i\*, distributed stakeholders are split into agents, actors, roles and positions. Two kinds of relations among the goals of stakeholders are often analysed. The first concerns the of individual stakeholders with AND-OR refinement respectively through decomposition and means-ends links. The second concerns the strategic dependencies (SD) among different stakeholders.

In an SR model, four types of intentions can appear. These are goals, softgoals, tasks and resources. These can appear as nodes on the AND-OR refinement trees or appear in the SD model as the dependum of the dependencies. Goals represent the desired states of the stakeholders, whilst softgoals model quality requirements that do not have clear-cut Yes/No answers, such as security. The goals/tasks connect to softgoals through four types of contribution links (HELP + , HURT -, MAKE ++ or BREAK -).

- **Secure TROPOS**

Secure Tropos extends both the i\* modelling language and its Tropos development process[423]. The main concept introduced by secure Tropos is that of *security constraints*[424], which represent security requirements. The Secure Tropos methodology also provides a number of modelling activities for developers to analyse, delegate and decompose security constraints. Security constraints are satisfied by secure entities, which describe any goals and tasks that are related to the security of the system. Representing the strategic interests of an actor in security, *secure goals* model the high-level goals an actor

employs to satisfy any imposed security constraints. A secure goal can be achieved by an actor in various ways since alternatives can be considered [424].

- **KAOS**

KAOS is a requirement engineering framework introduced by van Lamsweerde et al., that supports patterns of goal refinement. These patterns allow high-level goals to be stated in terms of a combination of lower-level goals [426][427][428]. Goals are statements that express the intended behaviour of the system under development. It is expected that the cooperative interaction of the agents that make up the system will achieve the intended behaviour. Agents are the active components of the system - which could either be humans, hardware and software, or software that will be installed - that will play some role in satisfying the goals of the system.

In KAOS, the goals can be specified using both a formal, in temporal logic notation [429] and informal notation, in natural language. KAOS provides reusable patterns of goal refinement which can be formally proven in temporal logic expressions. The patterns ensure that each stage of the elaboration process is correct: achieving the low level goals is equivalent to achieving the higher-level goal; consistent: being possible to satisfy all the low-level goals; and minimal: such that there are no redundant goals in the refined set.

#### 7.2.1.4.3 *Problem-Oriented Approaches*

Problem-oriented approaches[430][431], a theoretical framework, allows for: identification and clarification of system requirements; the understanding and structuring of the problem world; the structuring and specification of a machine that can ensure satisfaction of the requirements in the problem world; and the construction of adequacy arguments, to convince both developers and other stakeholders that the system will provide what is needed.

- **Problem Frames**

The Problem Frames approach (PF) was introduced by Jackson [431] and provides an intellectual structure for analysing software problems in the problem space. Problem Frames introduces several principles. PF firstly emphasizes on the need for separating three descriptions: *specification (S)*, a description of a software system, *problem world domains (W)*, and the context of the problem and the *requirement (R)*, that the specification has to satisfy. Secondly, in PF, subproblems of a more complex problem are fitted into known problem patterns called problem frames. A frame captures the contextual structure of a problem and concerns associated with the frame. One of such concern is the 'proof obligation' to show  $W, S \models R$ .

The main logic behind problem frames is that some software development problems are recurring. Based on this logic, the main idea is to document classes of commonly recurring problem structures and their solutions in problem-solution patterns. As such, it becomes easier to find a solution that solves the problem if it matches a well-known structure.

- **Abuse Frames**

Abuse frames were proposed by Lin et al [433][434] as an approach to analyse security problems to determine security vulnerabilities. This approach is based on Jackson's Problem Frames approach to

structuring and analysing software development problems [435]. Abuse frames are based on the notion of an anti-requirement, the requirement of a malicious user that can subvert an existing requirement [436].

Abuse frames represent the notion of a security threat imposed by malicious users and a means for bounding the scope of security problems. Scope binding of a security problem allows describing it more explicitly and precisely. Identification and analysis of threats are facilitated by such explicit and precise descriptions, which in turn drives the elicitation and elaboration of security requirements.

### 7.2.2 Cybersecurity landscape

In the past, CPS device safety and reliability have been the foremost concern, even though their security had always been recognized as important. Despite this acknowledgement, CPS security has traditionally been treated as an “afterthought” in the engineering phases of systems [437], owing mostly to the fact that traditional CPS security dangers were limited, since these systems were operated in isolation from the rest of the world. The use of open networks, wireless technologies, the Internet, IoT, and the cloud, has terminated the isolation of CPS resulting to Internet-based attacks being the majority of attacks after 2001 [438]. Unfortunately, despite the increasing connectivity of systems and realization that the connectedness requirements of modern systems also imply an increased need for system security, the application of security-conscious design practices in industrial systems remains fragmented. With the ever-changing nature of cyber threats, it would be impossible to design a completely secure system that could be robust against any future type of attack. However, to ensure CPS reliability and security, it is imperative that these systems are designed from the outset with a thorough understanding and consideration of the threats and challenges that at least a *current* adversary may pose.

Although the precise form of a CPS security threat may differ, and even though it is impossible to predict the novel and ingenious types of attack that may emerge in the future, the fundamental nature of a threat remains unchanged. A security threat represents a set of circumstances with the potential to cause loss or harm [439]. The US National Institute of Standards & Technology defines threats more analytically as “*Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.*”[440]. In a CPS, system vulnerabilities that can cause such circumstances or events might be discerned into cyber, physical or cyber-physical. Physical vulnerabilities include physical sabotage of equipment or jamming but also fault injection and side channel attacks [441]. Cyber vulnerability types include communications and communication protocols, software, and web-based attacks. Cyber-physical vulnerabilities include interconnected devices, insecure protocols, insecure Operating Systems, software, replay and injection attacks.

Summarizing, a threat is a situation where a given system is vulnerable to one or multiple *attackers*, attempting to gain access to one or more components of the *system* in order to carry out an unauthorized *objective*. Through a review of existing literature, Lei et al.[442] propose a concise taxonomy of threats to the security (and by extension, reliability) of CPS, in which a specific attack can be categorized from three major perspectives, namely its origination, purpose and target. Furthermore, in [443], we also find a consideration about the *consequences* of a successful attack.

In this section we revise the existing work done in different projects and initiatives to model threats and provide taxonomies to categorise threats and attacks at different levels of the CPS environment.

### 7.2.2.1 System domain

From a general perspective, the CPSoS is vulnerable to any type of threat, from physical attacks, failures and malfunctions, to intentional malicious activities. There are different taxonomies and models of threats that apply to any kind of system that should be considered. ENISA proposed in 2016 a general Threat Taxonomy [444] which is updated regularly in the ENISA Threat Landscape Report (the latter is from 2018 [445]) to analyse emerging technologies and reflect new threats. Figure 89 depicts the hierarchy of threats defined by ENISA. The high-level threats are depicted in dark blue fill at the top of the figure, and for each high-level threat, the list of threats that are classified in each of the high-level groups depicted in white fill.

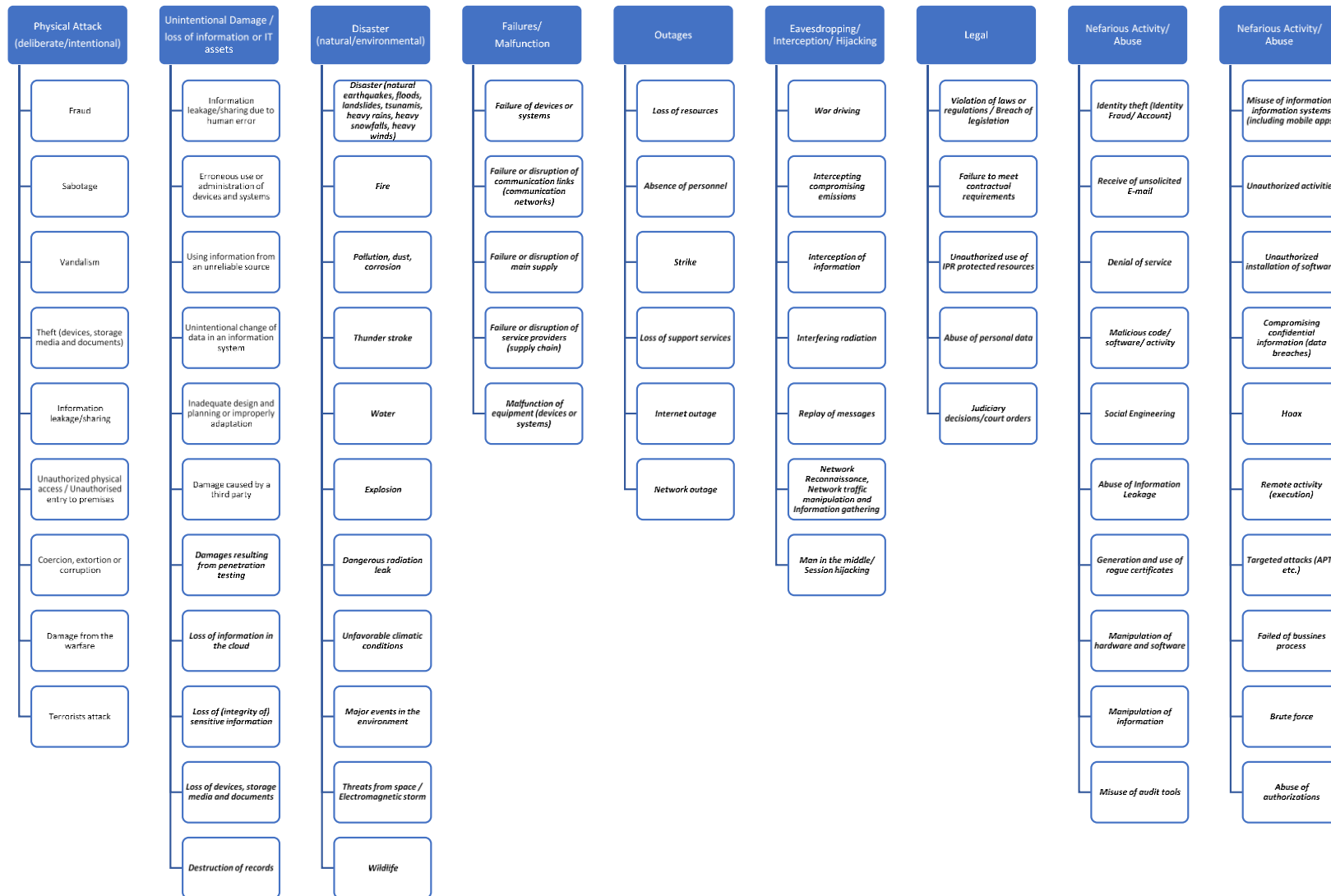


Figure 89 ENISA Threat Taxonomy



Threat modelling has been a popular field of research, Shevchenko et al. summarize all available methods in [448]. The methods considered and their main features are depicted in Figure 90. A very popular threat-modelling method developed by Microsoft is STRIDE [226], which groups threats into six categories: Spoofing, Tampering, Repudiation, Information disclosure (privacy breach or data leak), Denial of service, and Elevation of privilege.

Other authors have presented taxonomies of threats and attacks with a more specific focus on CPS aspects. The Xu et al. [446] classification includes Spoofing identity, Tampering with data, Repudiation of origin, Information disclosure, Elevation of privilege, Denial of Service (DoS). Alguliyev et. al [224] propose an attack tree that classifies attacks into five groups:

- Attacks on sensor devices, including Injecting false radar signals, dazzling cameras with light, GPS spoofing, etc.
- Attacks on actuators, two classes are considered: finite energy attack (incl. modification and loss of personal packets, finite time attack and impulse attack) and Bounded attack (suppression of control signal)
- Attacks on computing components, include trojans, viruses, worms and DoS attacks, integrity attacks (obtaining a key for secure communication or capturing some network devices)
- Attacks on communications: selective forwarding, packet spoofing, packet replaying, sybil, etc.
- Attacks on feedback: feedback integrity attack

The following sections provide a more specific review of threats and attacks relevant to the CPSoS at different levels: application, communication and CPS device.

Threat Modeling Method	Features
STRIDE	<ul style="list-style-type: none"> <li>Helps identify relevant mitigating techniques</li> <li>Is the most mature</li> <li>Is easy to use but is time consuming</li> </ul>
PASTA	<ul style="list-style-type: none"> <li>Helps identify relevant mitigating techniques</li> <li>Directly contributes to risk management</li> <li>Encourages collaboration among stakeholders</li> <li>Contains built-in prioritization of threat mitigation</li> <li>Is laborious but has rich documentation</li> </ul>
LINDDUN	<ul style="list-style-type: none"> <li>Helps identify relevant mitigation techniques</li> <li>Contains built-in prioritization of threat mitigation</li> <li>Can be labor intensive and time consuming</li> </ul>
CVSS	<ul style="list-style-type: none"> <li>Contains built-in prioritization of threat mitigation</li> <li>Has consistent results when repeated</li> <li>Has automated components</li> <li>Has score calculations that are not transparent</li> </ul>
Attack Trees	<ul style="list-style-type: none"> <li>Helps identify relevant mitigation techniques</li> <li>Has consistent results when repeated</li> <li>Is easy to use if you already have a thorough understanding of the system</li> </ul>
Persona non Grata	<ul style="list-style-type: none"> <li>Helps identify relevant mitigation techniques</li> <li>Directly contributes to risk management</li> <li>Has consistent results when repeated</li> <li>Tends to detect only some subsets of threats</li> </ul>
Security Cards	<ul style="list-style-type: none"> <li>Encourages collaboration among stakeholders</li> <li>Targets out-of-the-ordinary threats</li> <li>Leads to many false positives</li> </ul>
hTMM	<ul style="list-style-type: none"> <li>Contains built-in prioritization of threat mitigation</li> <li>Encourages collaboration among stakeholders</li> <li>Has consistent results when repeated</li> </ul>
Quantitative TMM	<ul style="list-style-type: none"> <li>Contains built-in prioritization of threat mitigation</li> <li>Has automated components</li> <li>Has consistent results when repeated</li> </ul>
Trike	<ul style="list-style-type: none"> <li>Helps identify relevant mitigation techniques</li> <li>Directly contributes to risk management</li> <li>Contains built-in prioritization of threat mitigation</li> <li>Encourages collaboration among stakeholders</li> <li>Has automated components</li> <li>Has vague, insufficient documentation</li> </ul>
VAST Modeling	<ul style="list-style-type: none"> <li>Helps identify relevant mitigation techniques</li> <li>Directly contributes to risk management</li> <li>Contains built-in prioritization of threat mitigation</li> <li>Encourages collaboration among stakeholders</li> <li>Has consistent results when repeated</li> <li>Has automated components</li> <li>Is explicitly designed to be scalable</li> <li>Has little publicly available documentation</li> </ul>
OCTAVE	<ul style="list-style-type: none"> <li>Helps identify relevant mitigation techniques</li> <li>Directly contributes to risk management</li> <li>Contains built-in prioritization of threat mitigation</li> <li>Encourages collaboration among stakeholders</li> <li>Has consistent results when repeated</li> <li>Is explicitly designed to be scalable</li> <li>Is time consuming and has vague documentation</li> </ul>

Figure 90 Threat Modelling Methods and main features

### 7.2.2.2 Application domain

With a focus on the applications and services running in the system, the following initiatives and frameworks regularly identify attacks and threats that should be taken into account when monitoring the security of the CPSoS.

- **OWASP Top 10 Web Application Security Risks**

The OWASP Top 10 [325] is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.

Category	Description
<b>Injection</b>	Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
<b>Broken Authentication</b>	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
<b>Sensitive Data Exposure</b>	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
<b>XML External Entities (XXE)</b>	Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
<b>Broken Access Control</b>	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
<b>Security Misconfiguration</b>	Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.
<b>Cross-site Scripting (XSS)</b>	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
<b>Insecure Deserialization</b>	Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be

	used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
<b>Using Components with Known Vulnerabilities</b>	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
<b>Insufficient Logging &amp; Monitoring</b>	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

- **Software Vulnerabilities and Errors**

Vulnerable software is a usual entry point for attackers to gain access to the data managed by applications and put data at risk. Therefore, these weaknesses and vulnerabilities must be regularly monitored, usually by programming regular vulnerability assessment and pen testing sessions. The MITRE - Common Vulnerabilities and Exposures (CVE) [450] database is a list of common identifiers for publicly known cybersecurity vulnerabilities. This is a widely used source of information to monitor vulnerabilities for a specific software vendors and solutions (e.g. OpenCL). The SANS Institute developed the CWE (Common Weakness Enumeration)/ SANS 25 [452][452], along with MITRE, to identify software security weaknesses and vulnerabilities that attackers usually target to exploit. The following is the list of the Top 25 most dangerous software weaknesses according to SANS/MITRE

- CWE-119 - Improper Restriction of Operations within the Bounds of a Memory Buffer
- CWE-79 - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- CWE-20 - Improper Input Validation
- CWE-200 - Information Exposure
- CWE-125 - Out-of-bounds Read
- CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
- CWE-416 - Use After Free
- CWE-190 - Integer Overflow or Wraparound
- CWE-352 - Cross-Site Request Forgery (CSRF)
- CWE-22 - Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
- CWE-78 - Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
- CWE-787 - Out-of-bounds Write
- CWE-287 - Improper Authentication

- CWE-476 - NULL Pointer Dereference
- CWE-732 - Incorrect Permission Assignment for Critical Resource
- CWE-434 - Unrestricted Upload of File with Dangerous Type
- CWE-611 - Improper Restriction of XML External Entity Reference
- CWE-94 - Improper Control of Generation of Code ('Code Injection')
- CWE-798 - Use of Hard-coded Credentials
- CWE-400 - Uncontrolled Resource Consumption
- CWE-772 - Missing Release of Resource after Effective Lifetime
- CWE-426 - Untrusted Search Path
- CWE-502 - Deserialization of Untrusted Data
- CWE-269 - Improper Privilege Management
- CWE-295 - Improper Certificate Validation

### 7.2.2.3 *Communication domain*

#### 7.2.2.3.1 *SDN networks attacks*

Software-Defined Networks (SDN) is a widely accepted paradigm for its many benefits [453], such as agility, lower operating costs, network automation, efficient and adaptable resource management, capacity for network supervision and control among others. SDN also offers benefits for increased cybersecurity, by making it easy to collect network usage information for use in detecting attacks, and to block and filter malicious traffic while allowing normal traffic flows. But the SDN modern networks are vulnerable and increasingly a popular target of attacks [454] [455], becoming a key asset to monitor and protect especially in Industrial deployments (e.g. Industrial Control systems) and other critical infrastructures. Krishnan and Najeem [456] classifies attack vectors in SDN-based networks into three: behaviour characteristics, based on resources and key functional components; and identifies six main attack techniques:

- Spoofing attacks
- Man in the middle attacks
- Tampering
- Repudiation
- Information disclosure
- Denial of Service – Flooding and Saturating attacks

The Open Networking Foundation (ONF) has suggested to follow eight SDN security principles to prevent these attacks[390]:

- Clearly define security dependencies and trust boundaries.
- Assure robust identity.
- Build security based on open standards.
- Protect the information security triad.
- Protect operational reference data.
- Make systems secure by default.
- Provide accountability and traceability.
- Follow properties of manageable security controls

ONF also proposes an attack model for the OpenFlow Switch protocol in

### 7.2.2.3.2 Wireless Sensor Network (WSN) attacks classification

In a so-called *smart scenario* context, smart objects or devices may communicate through standard networks, such as Wi-Fi or Ethernet, or build a dedicated network to communicate with other sensors, hence creating a network called Wireless Sensor Network (WSN). The H2020 Project Anastacia (G.A. N° 731558) proposed in [229] a classification of attacks, depicted in Figure 91, which is based on two main categories of threats: active and passive attacks. In active attacks, a malicious client actively injects or alters a network message in order to exploit some sort of vulnerability affecting the targeted host or network. In passive attacks, the aim of the attacker is to obtain the information without actively communicate on the network. In both cases, the aim is to introduce delays on the network or to steal sensitive information from the targeted systems.

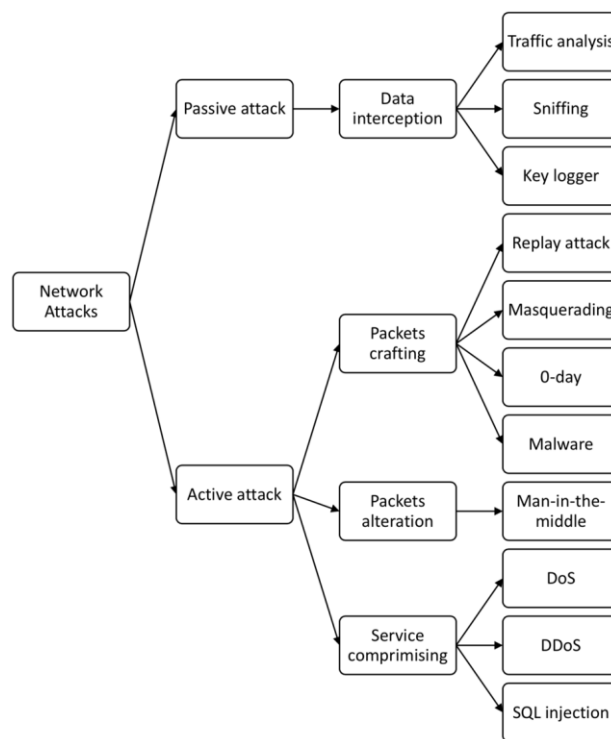


Figure 91 WSN attacks classification- H2020 project Anastacia [229]

### 7.2.2.3.3 V2X Communication Threat model

For the V2X domain, amongst the top priorities is to identify the threat landscape and especially the threat agents, to understand the potential threats and attacks that these agents can cause in the specific assets and, in this way, to be in place to ensure the security and authenticity of the exchanged information and take the appropriate mitigation actions.

The following Table 18 shows an overview of threat agents and attacks in the V2X domain, compiled by literature review ([458]-[466]).

Table 18 Threat agents and attacks in the V2X domain

TYPES OF THREAT AGENTS	CHARACTERISTICS
<p><b>Hacktivist groups</b></p>	<ul style="list-style-type: none"> <li>• DDoS attacks, doxing, website defacements</li> <li>• information theft, virtual sabotage, website parodies</li> <li>• Whistleblowing</li> <li>• Gathering information about network (reconnaissance)</li> <li>• Man in the middle (MITM_</li> <li>• Session hijacking</li> <li>• Repudiation of actions</li> </ul>
<p><b>Cybercriminal groups or individuals</b></p>	<ul style="list-style-type: none"> <li>• Use crimeware. phishing, and spear-phishing</li> <li>• Trojan</li> <li>• Smash-and-grab, social engineering, business email compromise (BEC) scams, botnets, password attacks, malware, ransomware</li> <li>• Interception of information</li> <li>• Replay of messages</li> <li>• Account hijacking</li> <li>• Network reconnaissance</li> <li>• Man in the middle</li> <li>• Session hijacking</li> <li>• Repudiation of actions</li> </ul>
<p><b>Insider threat agents</b></p>	<ul style="list-style-type: none"> <li>• Repudiation of actions</li> <li>• Data exfiltration or privilege misuse</li> <li>• Interception of information</li> <li>• Replay of messages</li> <li>• Network reconnaissance</li> <li>• Man in the middle/</li> <li>• session hijacking</li> </ul>
<p><b>State actor or state-backed actor</b></p>	<ul style="list-style-type: none"> <li>• DDoS attacks</li> <li>• Spear-phishing password attacks, social engineering, direct compromise, data exfiltration, remote access trojans, and destructive malware.</li> <li>• Interception of information</li> </ul>

	<ul style="list-style-type: none"> <li>• Interfering radiation</li> <li>• Network reconnaissance</li> <li>• cyber reconnaissance of critical infrastructure</li> <li>• Man in the middle</li> <li>• Session hijacking</li> <li>• Repudiation of actions</li> </ul>
Other possible actors: Cyber-terrorists and corporate entities	<ul style="list-style-type: none"> <li>• Defacements and claimed leaks</li> <li>• Interception of information</li> <li>• Interfering radiation</li> <li>• Replay of messages</li> <li>• Network reconnaissance</li> <li>• Man in the middle/</li> <li>• Session hijacking</li> <li>• Account hijacking</li> <li>• Repudiation of actions</li> <li>• Worm</li> <li>• Spoofing</li> </ul>

Some of the most severe security threats on the V2X domain and the potential assets that can be respectively attacked are listed next:

**Attacking a vehicle using V2X communication channels:** A potential intruder could attempt a spoofing attack through active ways, e.g. sending related messages, interfering with transmission of malicious data or code, inject malware V2X messages, replay messages initiate Denial of service attacks for overloading the requests and disrupting the vehicle system functionalities. Moreover an intruder could attempt also an attack through passive ways such as for instance eavesdropping & man in the middle hacking meaning to attempt to gain access to critical and sensitive data (that is going to be used afterwards for exploiting further vulnerabilities)

**Attacking a vehicle by exploiting automotive software update:** An attacker may attempt to damage the targeted vehicle’s systems through manipulating software updates and, this way, make misuse of the procedure or even deny normal OTA updates that would allow vehicles to resist efficiently against the latest cyber-attacks. As a result, the vehicles are exposed to risk of malfunctions. Cybercriminals have even had success manipulating cellular networks through built-in SIM cards which car companies use to extract real-time information and update firmware. For instance, Cyber-security firm IntSights released a study [239] that provides information on how hackers are managing to attack vehicles. According to the study, the problem is getting worse because of the need for constant updates, which may not take place considering the decades-long life of most cars.

**Attacks on backend server:** An attacker can attempt to compromise a vehicle’s assets through attacking the vehicle’s backend server sensitive information in order to stop the provision of service to the users. This way an attacker can cause data breaches and can furthermore use the backend server for attacking other vehicle assets or stop service provisioning.



**Attacking Controller Area Network (CAN) protocol:** The increasing amount of information exchange within the CAN bus system and the buses that interact with the outside world introduces an array of security threats fighting their way to penetrate the system [240][241][242][243]. The most popular method involves attacking a car's CAN protocol, something that can give a hacker full access to all of the vehicle's functions. A hacker can use various methods to succeed it e.g through masquerade and replay attacks in order to intrude and send illegitimate CAN messages. In addition, the CAN bus can be attacked and penetrated physically through the OBD port, charging unit etc.

**Attacking automotive networks, infotainment and telematics:** Local Interconnect Network (LIN) can be an alternative to CAN (in case that the higher bandwidth and increased adaptability of the latter are not needed) [244] is another in-vehicle network that can be exploited by malicious actors. Threats to LIN communication include Message Spoofing, Response Collision, and Header Collision attacks. The FlexRay network is often thought of as a potential successor to CAN, having a communication rate of 10 Mbps and similarly to LIN is also vulnerable to security attacks. FlexRay communication is vulnerable to Eavesdropping and Static Segment attacks. Ethernet is also considered as a next generation option in vehicle network [245]. And automotive networks that are using Ethernet can offer high bandwidths and timing guarantees. Threats to Ethernet communication include Network Access attacks, Traffic Confidentiality attacks, Traffic Integrity attacks, and Denial of Service attacks. Infotainment and telematics systems are vulnerable to control override attacks and injection attacks.

**Attacks based on vulnerabilities caused by human errors:** Attackers use different methods for exploiting human errors and this can be accomplished through the usual way of social engineering -tricking a legitimate stakeholder to download and install malicious software that will be used afterwards for implementing the attack or information interception, manipulating users into installing USB malware, clicking on unknown links or installing fake software from untrusted sources

**OBUs manipulation:** In this case a potential intruder may assess or alter the sensor readings in order to attack the system. Another example will be to transmit falsely messages to trick the targeted OBU in order to believe that it received legitimate messages without having doing so, to install malicious software, to manipulate OBU to interpret wrongly messages, to alter or overwrite OBU software, to alter configuration and sensors etc

**Interception of V2X communication packets:** Malicious attackers despite security measures may still manage to intercept the V2X packets, and obtain information such as a vehicle owner's identity, vehicle location information, driving trajectory, and so on [474].

#### **7.2.2.4 Device domain**

According to [475] CPS security can be viewed in three different levels: perception, transmission and application level. Each of these layers is defined by the devices within it and the related functions that should be implemented [476] in those devices. We base our analysis on the above categorization focusing explicitly on the CPS devices themselves.

The first layer is the perception layer, also called recognition layer (Kumar and Patel, 2014) or sensors layer [477]. This layer has multiple terminal equipment such as sensors, actuators, cameras, GPS, laser scanners, intelligent devices, RFID tags with 2-D bar code labels and readers [478][479]. Devices at this layer have the ability to collect real-time data that is needed for different purposes (e.g. monitoring and tracking),

interpret what they receive from the physical world and perform commands from the application layer. The collected data can include sound, light, mechanics, chemistry, heat, electricity, biology or location [480]. Sensors can generate real-time data with node cooperation in wide and local network domains [481], which will be aggregated and analysed in the application layer. Sensors, depending on their type, can aggregate information related to temperature, acceleration, humidity, vibration, location or air chemical changes [482]

The second layer is related to transmission and involves all the communication activities of the CPS with other CPSs and the CPSoS level (intra and intercommunication). This layer is related to the network protocol and infrastructure of the CPS, thus relevant security threats are analysed in the network threat section of this document.

The third and most interactive layer is the application layer. Its mission is to process the received information from the data transmission level and issue commands to be executed by the physical units, sensors and actuators [483]. This layer works by implementing complex decision-making algorithms on the aggregated data to generate correct decisions [484], and control commands which will be used in corrective actions. In addition, this layer receives and processes information from the perception layer and, then, determines the required automated actions to be invoked[485]. Data aggregation from different resources and intelligent processing of massive data are performed at this layer with object control and management. Cloud computing, middleware, and data mining algorithms can also be used in management implementation of connected devices at the physical layer [486].

#### 7.2.2.4.1 Perception-based CPS Security Threats

Perception layer threats and attacks are related to the two main procedures that take place in the CPS, ie. sensor and actuation. As all threat and attacks on CPSs, they are linked to the assets of the device that need to be protected. Given that devices are mostly located in external and outdoor environments, this results in physical attacks, such as tampering with the devices' components or replacing the devices. Common attacks at the perception layer include equipment failure, line failure, witch, electromagnetic interference, perceptual data corruption [487], side channel attacks analysis (e.g. differential power analysis) and microarchitectural attacks [488], information disclosure, information tracking, tampering, sensing information leakage [489], physical destruction and energy-exhaustion attacks [490]. Some common threats and attacks on the Perception layer are presented in the following Table ( Table 19).

**Table 19 Threats and Attacks on the Device Perception Layer (Sensors and Actuation)**

Threat/ Attack class	Objective	Possible CPS Device components	Security Goal Breach	Description
<b>Sensor Alteration, Data theft</b>	Collect or alter data provided by a sensor	CPS Sensors (GPS, Vision, LiDAR, motion, pressure,	Integrity, Confidentiality	Tries to inject external control inputs and false sensor measurements, wishing to disrupt the system (Fault Data injection)[491]

		temperature etc)		
<b>Sensitive information leakage</b>	Reveal Sensitive information within the CPS structure	CPS processor, memory, storage area,	Confidentiality, Data disclosure, Access control	Takes over the node and attains and leaks information that could involve encryption keys, which is then used to threaten the security of the entire system. Perform side channel attack so that the cryptographic keys can be retrieved
<b>Sensitive Information lea</b>	Corruption of the CPS hardware structure	CPS hardware components	Confidentiality, Data disclosure, Access control	Hardware Trojan can be used in order to create fake CPS devices. A Hardware Trojan is a malicious modification of an integrated circuit (IC), which enables the attacker to use the circuit or to exploit its functionality to obtain access to data or software running on the integrated circuits [492]
<b>Denial of Service</b>	Make a CPS sensor or actuation process inaccessible	CPS sensors and actuators	Availability	Send a large number of packets, flooding packets, along the routing path to a CPS, leading to CPS battery draining, sleep deprivation, and outage attacks[493] memory/processing resources. Code Injection attacks can also achieve the same goal by injecting code that will stop the system.
<b>Physical Attack on a Device</b>	Steal a device in order to extract information for future use, e.g., find the	CPS device	Confidentiality, Authentication, Access control	The attacker, with a physical access to the device, may extract valuable cryptographic information, tamper with the circuit, modify programming, or change the operating system [494]

	fixed shared key			
--	------------------	--	--	--

#### 7.2.2.4.2 Application based CPS Security Threats

Regarding the Application Layer of the CPS Device, threats that stem from applications on the device introducing vulnerabilities that will impact the CPSoS are been considered. Thus, the main security concern is the vulnerabilities that might result from the application design, which can be exploited by adversaries to attack the system. Thus, malicious code or software can be launched to affect system security. Another security concern can be a result of integrating various techniques, which might impede data processing, resulting in a bottleneck in the system. These security issues can affect the availability and reliability of the system [495].

Security at the application layer includes information accessing, user authentication, information privacy, establishment-retainment of secure data links, platform stability and management [496]. Moreover, each application has its own security requirements that need to be addressed and that may constitute the target of cybersecurity attacks. In all applications, the data processing and M2M or user access control are prime targets for attacks. Possible threats and attacks can be found in the following Table 20.

**Table 20 Threats and Attacks on the Device Application Layer (Sensors and Actuation)**

<b>Malformed Firmware/Hardware</b>	Gain Access to CPS resources, Privilege escalation	CPS processing and update mechanism	Access Control, Authentication, Availability, Integrity  Confidentiality	Trying to update the CPS software or hardware structure by deploying a Malicious firmware or hardware bitstream
<b>Integrity Attacks Against Machine Learning</b>	Extracted intelligent decisions are misaligned and false	CPS data aggregation and processing,  CPS intelligence Machine learning classifiers	Integrity  Authentication	Trying to infuse appropriate data so as launch to causative attacks, where the attacker changes the training process by manipulating the training dataset [497][498] (poisoning), or exploratory attacks, where the attacker

					exploits vulnerabilities without altering the training process
<b>Logging alteration</b>	<b>Mechanism</b>	Change values in log files or introduce on essential logging information	CPS log mechanism	Repudiation Availability Integrity	The attacker tries to alter the logging mechanism so that there is insufficient or irrelevant logging of CPS application activities.
<b>Application functionality change</b>	<b>software</b>	Malicious activity infused in CPS application software	CPS Application Software	Integrity, Access control Confidentiality	An attacker could inject a malicious input that causes the service providers to perform operations on behalf of the attacker. The attack can use software code vulnerabilities (e.g. zero day)

### 7.2.3 Security primitives and mechanisms for protection

To provide security it is generally needed to ensure the following properties (or, in some cases, a subset of them): confidentiality, integrity, non-repudiation, and availability of the data. Confidentiality means that the data should be accessed only by authorized parties. Integrity means that the data should not be altered by a non-authorized party. Non-repudiation means that none of the parties involved can deny to send or receive information. Finally, availability means that the data should be available when needed.

The first and most obvious way to ensure that the CPSs are guaranteeing these properties is to include in the CPSs the security primitives needed to implement them. The security primitives that should be included in the CPSs depend on the specific system and on the functions that it has to provide. Basic security primitives include block ciphers, stream ciphers, public key algorithms, hash functions, authenticated encryption scheme. By combining these primitives, it is possible to guarantee the security of the CPSs. The primitives, however, as it will be discussed later, need to be used properly and implemented in a way that

is robust against side channel attacks, otherwise their presence is useless. In the rest of this section we will summarize each of these basic primitives, reporting the information currently available in literature.

Block ciphers, as the name indicates, are encryption algorithms that carry out the encryption operations one block at a time. The block has fixed length that is often 64 bits or 128 bits. These algorithms are symmetric, which means that they use the same key for encrypting and decrypting. The most common block cipher is the Advanced Encryption Algorithm (AES) that is the standard block cipher [499]. The algorithm is a block cipher operating on a block of 128 bits and have key lengths of 128, 192, or 256 bits. The encryption starts with the first key addition, then the round function is repeated a specific number of times depending on the key size. In the encryption operation the round is composed of the following four transformations: SubBytes, which is the non-linear transformation (S-box) and operates on bytes, ShiftRows, which cyclically shifts to the left the bytes of the state with an offset depending on the line index, MixColumns, which is a multiplication by a matrix over  $GF(2^8)$ , and the AddRoundKey, where the round key is added to the state by means of XORs. The round keys are expanded from the secret key using a dedicated expansion routine. The decryption process is similar to the encryption one, but applies the inverse of the round transformation. The design space of AES has been largely explored these years. Implementations of the AES algorithm proposed so far covers almost completely the possible options, ranging from high throughput to extremely lightweight [500].

Several other block ciphers have been recently proposed aiming, in particular, at devices with limited resources and strict constraints. These algorithms belong to the class of lightweight algorithms, which are algorithms designed to be lightweight in the broader sense (occupying little amount of area or have an extremely limited power and energy footprint). Among the most popular ones is certainly PRESENT [501] that has been standardized by the ISO in 2012. The state size is 64 bits; the algorithm supports two key sizes of 80 and 128 bits, respectively. The encryption process consists of 31 rounds, each of which including three transformations: the addRoundKey, which adds the current state with the round key; the sBoxLayer which is the S-box and operates on blocks of 4bits; and the pLayer step which performs the permutation specified by the standard. The reference implementation of PRESENT requires 32 clock cycles to encrypt a 64 bits plain-text with an 80-bit key, occupies 1570 GE and has a simulated power consumption of  $5\mu W$ . However, designs optimized for low power can be as small as 1000 equivalent gates (GE) and consume less than  $3.3\mu W$ .

Among other lightweight block ciphers is worth mentioning Midori [502], the first algorithm designed having low energy in mind, KATAN [503], and PRINCE [504].

Public key algorithms are cryptographic algorithms that use different keys for encryption and decryption. The key is composed by a pair of keys, one public and one private. The private key should be kept in the hand of the user that generates, while the public key can be freely shared with any user. The keys used for encryption cannot be used for decrypting messages. These properties allow to build security functionalities such as signatures and solve issues related with the secure exchange of the key. The security of public key algorithms is based on the security of the underlying hard problem that should be easy to compute in one direction while computationally difficult to invert. The current hard problems on which the security of public key algorithms is based are the discrete logarithm problem or the factorization of integer numbers. Unfortunately, the future advent of quantum computers will make these problems treatable, causing the immediate obsolescence of current standards for public key cryptography. Addressing the issue, several institutions and public bodies started the process of selecting new and quantum resistant algorithms as new public key standard. Among the selection processes it is worth mentioning the one of NIST [505], that

is a public context, currently ongoing, where several algorithms, proposed by the scientific community, are under scrutiny. The most popular problems that are used to build quantum-resistant public key cryptography are lattice-based, hash-based or code-based. It is thus clear that, in the near future, devices providing public key capabilities will have to provide that feature including quantum resistant algorithms. However, in this moment where the standard is still under selection it is not possible to commit to an algorithm. It is thus fundamental to ensure the possibility to eventually update the algorithm deployed, guaranteeing the so-called crypto agility.

Hash functions convert a message into a digest of fixed length. Standards currently used are SHA-2 [506] and the recent SHA-3 [507]. SHA-256 operates in four steps: message pre-processing, where the message is divided into blocks of 512 bits, message schedule, where the message is expanded into 64 words of 32 bits, digest calculation, which is iterated 64 times to produce the data sent to the final step, and digest update, where the digest is updated by adding to the current digest the data coming from the digest calculation. The implementation of SHA-2 can be as small as 9.036 GE [508]. Keccak, the SHA-3 algorithm, uses  $r + c$  bits for the state (where  $r$  and  $c$  are configurable parameters of the algorithm named bitrate and capacity, respectively). The algorithm operates in two steps called absorbing and squeezing. In the absorbing phase  $r$  bits are updated by XORing them with the message bits and by applying the Keccak permutation (called  $f$ ). In the squeezing step,  $r$  bits are provided in output after each application of the same permutation. The function  $f$  is iterated a number of times determined by the size of the state and is composed of five basic operations. Operation Theta consists of a parity computation, a rotation of one position, and a bitwise XOR. The Rho operation implements a rotation by an offset which depends on the word position. Operation Pi is a permutation and operation Chi consists of a bitwise XOR, NOT and AND gates. Finally, operation Iota is a round constant addition. As in the case of AES, also the design space of Keccak has been widely explored, and several implementations ranging from high speed to low cost have been presented and discussed in the past.

As in the case of post quantum cryptography, also lightweight cryptography is a hot topic of research due to a standardization process currently going on [509] (even if the context is mostly dedicated to lightweight algorithms). The recently concluded CAESAR competition aimed at identifying, among several submissions, a portfolio of algorithms for the next generation of devices. Indication and lessons learned from the CAESAR competition are a fundamental building block for the development of authenticated encryption in the context of CPSs.

In the context of CPSs, it is extremely important to consider lightweight cryptography. CPSs, in fact, very often have limited resources available, thus either they cannot include full-fledged cryptographic algorithms, or they require to implement them in a way that has an extremely limited impact on the available resources. Research in the field of lightweight cryptography follows two main approaches. The first one envisages solutions aiming at minimizing the area occupation and limiting the energy consumption for standard cryptographic algorithms. The second approach addresses this issue by designing protocols, cryptographic algorithms, and physical attack countermeasures, having in mind the specific constrained devices where they will be implemented.

#### 7.2.4 Monitoring security: tools and techniques

The following subsections describe techniques and tools for monitoring security aspects at different levels of the CPSoS.

#### 7.2.4.1 Run Time Monitoring and Anomaly Detection in CPS

Considering the security threats and challenges that CPSs face, as described in the previous section, it becomes obvious that there is a considerable need to continuously monitor a CPS during its regular operation for security anomalies that can result to some security attack. Typical ICT systems have a series of well-developed tools that, by combining a wide range of technologies and methods, can detect, respond and mitigate security attacks. The generic category of run-time monitoring systems may comprise of various components like intrusion detection systems (IDS), zero-vulnerability malware detectors and anomaly detectors that are all interconnected under a security information and event management (SIEM) system. SIEM is usually responsible for the correlation between various events and logs to extract security alerts and make attack mitigation suggestions. However, a CPS runtime security monitoring system must consider the CPS specificities that, in several cases, are distinctly different than those of a typical ICT system.

According to [510], there are four basic characteristics that distinguish CPSs from typical ICT systems in terms of runtime security intrusion detection: physical process monitoring, Machine-to-Machine communications, heterogeneity and legacy system interactions. Due to their connection between the cyber and the physical world, the CPS devices measure physical phenomena and perform physical processes that are governed by the laws of physics. Thus, a CPS security monitoring system must perform physical process monitoring, using physical laws as a control mechanism to model and predict valid instructions and outcomes. Furthermore, many CPS application scenarios are highly focused on automation and time driven processes that realize closed control loops, that do not require human intervention (and its associated unpredictability). This kind of behaviour focused on Machine-to-Machine communications increases the regularity and predictability of the CPS activities. The CPS security monitoring system should be able to monitor regularly closed control loops. Thirdly, the attack surface of a CPS is considerably broader than that of an ICT system. CPSs consist of many heterogeneous subsystems and devices while they follow a broad range of different, non ICT-related, control protocols like ISA 100, Modbus, CAN etc. Some of these devices and protocols have proprietary software or standards that may constitute ICT attacks unfitting. This characteristic, along with the fact that a successful CPS attack has high impact and thus high payoff, attracts very skilled attackers that can mount very sophisticated attacks on CPSs[511]. Such attacks are usually very hard to discover and document since typical ICT intrusion detection software cannot identify them (e.g. the attacks may not be IT related but rather OT related). Attackers exploit CPS zero-day vulnerabilities which would render many ICT security monitoring toolsets useless (e.g. knowledge-based ones [510]).

Lastly, many CPSs include legacy hardware that is difficult to modify or physically access. Such components may be partially analogue, have very limited installed software resources and be dictated by physical processes. The challenge here is how to install security monitoring sensors on such devices and how to predict/model their behaviour correctly in order to detect possible anomalies. It needs also to be considered that legacy devices do not have many computational resources and it becomes hard for the monitoring system to retain its real-time responsiveness when collecting security metrics from them.

Runtime Security monitoring in the CPS domain, considering the above specificities, can take various forms. However, they all rely on two core functions, the collection of data from various CPS sources and the analysis of data in a dedicated runtime security monitoring subsystem. To achieve appropriate data collection, the security monitoring system must deploy security agent software/hardware [512] on the monitored CPS devices, or introduce virtual entities (Virtual Machines) for data collection [513] within the CPS infrastructure. Examples of collected data can be Syslog log events, system call logs, traffic recordings from network interfaces, reputation scores, processing loads, connection/communication failures etc. All



collected data are analysed in the CPS runtime security monitoring system that uses data mining, machine learning, pattern recognition or statistical data analysis to extract metrics on security issues that may take place inside the CPS at runtime. Such issues may be possible incidents detected via data that can be binarily characterized as bad/good, or continuously characterized by a specific significance grade. The performance of the security monitoring system is measured by the False Positive Rate (FPR), the False Negative Rate (FNR) and the True Positive Rate (TPR). The system is also measured in terms of incident detection latency and consumed resources number, computational overhead, excessive network traffic and power consumption [510].

To better understand the monitoring/detection approach that runtime security monitoring systems follow, we can broadly identify two approach categories, knowledge-based detection and behavioural-based approaches. In a knowledge-based security monitoring system runtime, features that are extracted from collected data are matched with a specific profile pattern or model. Alarms are raised when there is a behaviour mismatch with the existing profiles or models. This approach may lead to low FPR, but needs a very well described profile or model to be effective (e.g. an attack dictionary, a CPS device functionality pattern) since it relies on identifying a specific pattern/model.

On the other hand, behaviour-based security monitoring systems do not rely on a specific prescribed knowledge but rather look for runtime features that seem out of the ordinary and act as outlier values to the expected behaviour of a CPS. Supervised, semi-supervised or unsupervised machine learning algorithms can be employed on this approach. As expected, in supervised and semi-supervised algorithms a predefined training set must be constructed in such a way that it reflects accurately the expected CPS behaviour. Given the CPS specificities, this is a non-trivial task. It takes a lot of time and effort to structure such a dataset (e.g. using state-of-the-art feature analysis, discovery and engineering techniques) and still the behaviour-based monitoring may result in high FPR. Unsupervised behaviour-based monitoring does not need a pre-structured training set and creates the dataset using CPS live data [510]. The behaviour pattern that the above approaches evaluate can be a deviation from good behaviour or a match to bad behaviour [514]. Bad behaviour matching monitors detect attacks by building profiles of known bad system behaviour, such as statistical profiles of attacks [515][516]. Such monitors are robust since machine learning techniques tend to generalize from the presented data [514]. On the other hand, good behaviour deviation monitors build a statistical profile of normal (good) behaviour and detect deviations from this profile [517][518]. Their robustness is better than that of bad behaviour monitors since their employed machine learning techniques do not rely on historical knowledge of possible attacks [514].

There are several CPS security monitoring systems that consider some of distinguishing CPS characteristics in their design, like the works in [519][520], which are focused on closed control loop monitoring in autonomous computing systems and on traditional network traffic monitoring. Specifically, for industrial network runtime security monitors, there are solutions that take advantage of the physical process measuring taking place in an industrial site as well as the closed control loop processes [521], but they still use techniques based on traditional network traffic monitoring. For example, the ARMET [522] system can identify good behaviour deviations in a reliable way that has very low FPR and FNR. ARMET can observe an application's execution at runtime, compare it against the predicted execution behaviour and identify deviations.

When it comes to security runtime monitoring based on knowledge-based approaches using models, there is a need for some model description language that can take into account CPS characteristics like real-time responsiveness [523]. Barnett et al. in [524] propose the use of AsmL as an executable specification

language for run-time monitoring. AsmL, an extension of Abstract State Machines (ASM), is based on the formalism of a transition system whose states are first order algebras [525]. In [526] a full framework for executing specifications of real-time systems is proposed. This proposal can be used for security runtime monitoring in CPS timed systems.

There are very few CPS security runtime monitoring systems that provide efficiency metrics as are specified at the beginning of this section [519]. There exist works where such results are provided but only for CPS monitoring subsystems like IDSs [514].

What also needs to be mentioned is the fact that existing solutions on CPS security monitoring are primarily focused on detecting computational and network security incidents happening in a CPS. However, since a CPS implements closed control loops that rely on collected data for autonomic decision making, malicious attacks on the collected data can also constitute a very serious threat. Recently, effort has been invested in detecting false data injection (FDI) attacks that aim to maliciously alter the CPS control loops. Research works aiming to provide protection against FDI are focused on making efficient vulnerability analysis like the work in [514] where vulnerability to FDI is expressed as a satisfiability problem and solved using a solver that supports functions over real numbers [527] or focused on utilizing appropriately FDI fault diagnosis techniques [528][529].

It is important here to note that the full potential of a runtime security monitoring system is not unrelated with the security-by-design principles described in the previous section. A very important aspect of any monitoring tool is the mechanism that provides input to such a tool. As mentioned, in security monitoring tools inputs are provided by event data collection points (security agents or sensors) that are installed in various parts of a CPS. It is of prime importance that these security sensors are designed and realized in the CPS architecture during engineering phase (design time) and that they are fully integrated with the CPS architecture. Only then can such sensors maximize their efficiency (in terms of speed but also in terms of impact) on collecting all security related information that may trigger runtime security anomalies.

#### *7.2.4.2 Cybersecurity analytics: cross-correlation of security incidents*

Reducing the time between the start of a security incident and the reception of the first reports from sensors about that particular incident is the workhorse of every organisation's security incidents response team, in order to reduce the impact and damage this incident may cause to the organisation's tangible and non-tangible assets. Threats and attacks can be decomposed and modelled into a set of security events, in some cases happening in sequential, parallel or repetitive patterns. Security events are the result of monitoring the system at runtime, using different techniques and tools. Collecting, processing and correlating these security events in order to promptly identify threats and attacks helps security teams to analyse and handle security incidents faster and more effectively. This technique also permits reducing false positives and false negatives (security events detected in isolation, not followed or happening together with other events, which, put together in the same context, may result in a real threat), improving the overall quality of the security monitoring, incident detection and response processes.

Security Information and Event Management (SIEM) technologies implement security events collection, consolidation and correlation, with different degrees of support, flexibility, interoperability and usability. Gartner analyse SIEM software available in the market every year [530], considering both proprietary and Open Source solutions, looking at different evaluation criteria, and classifying them into market leaders, challengers, visionaries and niche players. However, despite their well consolidated role in every

organisation's security operation team nowadays, SIEM technologies have limitations, especially when dealing with complex attack patterns and lack of user or entity behaviour analytic features [530].

To overcome some of these limitations, many SIEMs offer the possibility to develop plugins to collect and process security events produced by a wide variety of sensors or probes. These can be generic sensors that can be used to monitor a multi-purpose infrastructure or developed ad-hoc to monitor specific types of devices, communication technologies or applications used in a specific domain. Moreover, SIEMs often allow security experts to extend the basic set of correlation rules provided by default, with custom rules designed to cover domain-specific threats and attack patterns, or complex scenarios that entail malicious activities observed at different layers of the system. Sections 7.2.4.3 to 7.2.4.5 describe technologies that can be used by SIEMs as a source of security information and events. These technologies focus on monitoring the infrastructure of a system at different levels: application, communication and device.

Furthermore, cross-correlation rules permit analysing security from a business-level perspective. Cross-correlation rules take as input security alarms generated by SIEMs, instead of security events collected from sensors. This way, we can have multiple SIEMs monitoring threats and attacks for a specific subsystem or context, which are additionally feeding a system-level SIEM. The system-level SIEM uses cross-correlation rules to combine alarms generated in each subsystem, in order to monitor business-level security objectives and indicators. Cross-correlation rules can be applied to the context of CPSoS to monitor system-level security indicators, such as system confidentiality, availability and integrity among others. The alarms generated by the SIEMs deployed at the individual CPS level can give information about the security status of the individual CPSs, whereas the system-level SIEM cross-correlation processes provides an holistic view of the security of the entire system.

### *7.2.4.3 Security monitoring of threats at application-level*

Section 7.2.2.2 reviewed most usual application level threats nowadays. The following is a list of technologies of different types that can be used to inspect and monitor software applications with the aim of detecting suspicious activities and threats.

- **Vulnerability scanners and security testing for applications**

These technologies scan software applications looking for vulnerabilities that may leave them open to exploitation. There are different methods to perform vulnerability scanning and security testing: white-box, which requires a deep understanding of the application internals and includes source code review; and black-box, which does not require source code review. Vulnerabilities and security flaws found can be correlated with related security events detected by other sensors (e.g. NIDS) in a SIEM to, for instance, confirm that an attack is on the way.

- **Content scanning, filtering and blocking technologies**

This type of technologies inspects the application data to look for harmful content and dispose it. Predefined file types, such as executables or code, are usually scanned in the system for automatic removal following security policy; even text files can be inspected for specific unwanted words. Content filtering can be also applied to content accessed online when browsing the web, looking for specific strings in the text of a web page, objects of images. Content filters are often part of firewalls but there is also specific software that focus on filtering and blocking content such as emails, websites or files. It is worth mentioning this

group of technologies, *Antivirus/Antimalware technologies*, which specifically focus on fighting against malicious software, preventing their execution in a computer system following two approaches: blacklisting (blocking the execution of malicious programs) or whitelisting (allowing only the execution of programs known to be benign). Many attacks start with the successful installation and execution of a malware in the system that allows the attackers to gain some privileges and take control of the system to perform additional actions. These actions may finally lead him/her to the ultimate objective, e.g. access or even leak sensitive or confidential data. Therefore, these technologies are an important source of security events for SIEMs as well.

- **Authentication and authorization frameworks**

Data privacy protection is the most important issue at the application domain in CPS. A weak authentication or a complete lack of access control in a software application or service are weaknesses usually exploited by attackers to e.g. enter and gain control of a system or to disclose CPS sensor data to an unauthorized party. Brute force attacks, dictionary attacks, elevation of privileges attempts, segregation of privilege violations, illegitimate access to confidential or sensitive data, data tampering attempts, etc. can be captured by authentication and authorization frameworks to generate security events. These can be considered together with other security events in a SIEM to detect, for example data breaches.

- **File integrity monitoring technologies**

These technologies are used to detect unauthorized changes in configuration files of critical applications or registry files of the operating system. This is a requirement included in many security standards such as PCI-DSS (Req. 10.5.5, 11.5 and 12.10.5) [532]. Attackers may try to modify a configuration file to allow for certain operations to be performed by a regular user and escalate privileges, or simply to damage the service's correct operation. Another type of files that should be monitored is log files, which attacker may try to modify in order to remove any tracks of their activity in the system.

- **Log monitoring technologies**

Application-level logs can be monitored and inspected to extract relevant information that can be used to identify anomalous behaviours that may lead to determine, for instance, that a device has been manipulated by an attacker.

#### **7.2.4.4 Security monitoring of the network**

As already introduced in Section 7.2.4.2, SIEM technologies collect, process and correlate security events generated by sensors and probes which monitor the CPS infrastructure at different levels. The SIEM can, then, generate security alarms to warn security operation teams and system administrators that a potential attack is happening, or a threat being materialized. In particular, to detect network attacks and threats, there are different technologies that can be used as source of information (i.e. security events) for a SIEM:

- **Intrusion Detection Systems (IDS)**

Kizza [533] defines *Intrusion detection* as a technique of detecting unauthorized access to a computer system or a computer network. An intrusion into a system is an attempt by an outsider to the system to illegally gain access to the system. Aurobindo Sundaram [548] divides intrusions into six types as follows:

- **Attempted break-ins**, which are detected by atypical behaviour profiles or violations of security constraints. An intrusion detection system for this type is called anomaly-based IDS.
- **Masquerade attacks**, which are detected by atypical behaviour profiles or violations of security constraints. These intrusions are also detected using anomaly-based IDS.
- **Penetrations of the security control system**, which are detected by monitoring for specific patterns of activity.
- **Leakage**, which is detected by atypical use of system resources.
- **Denial of service**, which is detected by atypical use of system resources.
- **Malicious use**, which is detected by atypical behaviour profiles, violations of security constraints, or use of special privileges.

These six can now be put into three models of intrusion detection mechanisms: anomaly-based/behavioural-based detection, signature-based/misuse-based detection, and hybrid detection. In the first two approaches, it is necessary to model first what is considered a benign or legitimate activity and what is considered unacceptable or anomalous activity. Both approaches have limitations and, therefore, a hybrid approach, although currently under research, seems to be the way to go.

Classification of IDS technologies:

- **Network-Based Intrusion Detection Systems (NIDSs)**: They monitor the traffic on the network to detect intrusions
  - **Host-Based Intrusion Detection Systems (HIDSs)**: They locally inspect the systems within an organisation to detecting malicious activities on a single computer.
  - **Hybrid Intrusion Detection System (HyIDSs)**: They combine the capabilities of the other two types in a single interface
  - **System Integrity Verifiers (SIVs)**: monitor critical files in a system, such as system files, to find whether an intruder has changed them.
  - **Log File Monitors (LFMs)**: first create a record of log files generated by network services. Then they monitor this record, just like NIDS, looking for system trends, tendencies, and patterns in the log files that would suggest that an intruder is attacking.
  - **Honeypots**: is a system designed to look like something that an intruder can hack. They attract and deceive attackers to learn about their tools and methods, without compromising the security of the network.
- **Scanning, Filtering and Blocking technologies**

Scanning is a systematic process of sweeping through a collection of data looking for a specific pattern. In a network environment, the scanning process may involve a program that sweeps through thousands of IP addresses looking for a particular IP address string or a string that represents a vulnerability or a string that represents a vulnerable port number. Filtering and blocking processes use a hardware device, a computer program or both, to limit or stop a computer or device from being able to establish a connection to another device through the network, based upon predetermined criteria such as IP or MAC addresses.

Some examples of technologies that implement scanning, filtering and blocking processes in the network are *firewalls*, used to monitor, filter and block data packets based on certain rules; *jammers*, which are signal blocking devices that transmit synchronized radio waves at the same frequency as a device, like a cell

phone or drone, in order to blur its signal. These technologies aim at protecting the system from external attackers but cannot protect from malicious insiders.

#### *7.2.4.5 Security monitoring of CPS devices*

As discussed previously, compared to pure software systems, the design of a CPS needs to be concerned with a mixture of physical, cyber and cyber-physical threats. To mitigate the effects of such attacks, a CPS may be designed so as to implement preventative measures at both the physical level (e.g. restricting physical access to certain assets, monitoring employee, contractor or asset presence, maintaining backup copies in physically remote locations) and also the cyber levels (e.g. enforcing strong cryptography, software certification and user/service authentication measures). Some measures involve a hybrid cyber-physical approach, for example, securing physical access to assets can be implemented using both physical equipment (e.g. gates, door locks) and cyber equipment (e.g. RFID employee badges, smart locks, sensor equipment). Other examples might include the physical location of networking equipment, which in turn, dictates the type of networking equipment that can be used, and thus the type of cyber security mechanisms that need to be implemented on top of the network layer. For example, at the design phase, the engineer may have to choose between wired and wireless communication technologies for certain assets, depending not just on the physical properties of the environment where these assets might be located, but also on the likelihood of potential types of attack that exploit weaknesses of the network types. In the next sections, we discuss a few core considerations during the design of secure CPSs and present common approaches that an engineer might consider during the design phase.

Monitoring of a CPS device is performed in the Perception layer by observing the behaviour of physical processes, and the issued commands that change the behaviour of the physical device. Thus, we can ensure the work environment functions both correctly and optimally. The application layer also saves past actions so that feedback of any previous action can be given for ensuring future operational improvements. The objective of this layer is to create a smart environment [477], and combine CPS and industry professional applications. This has led to extensive and intelligent applications in areas that may include private and secure data, such as: SmartPower Grid; Smart Homes and Cities; Intelligent Transportation [480] Smart Auto; environmental monitoring; industry control [478] , Smart Health; and Smart Farming. Such applications might collect users' private data, such as health information and habits. Therefore, it is important to apply mechanisms to protect the data. On the other hand, application systems are different and require appropriate security policies. Hence, it is difficult to individually address a security policy for each application system.

When attackers are able to bypass network security and gain access to a networked CPS, there should be measures that can be implemented from the design phase, which can help ensure reliable operation of the system. A frequent objective of attack after penetration is to enact interruption or fabrication types of attack, threatening the integrity of CPS data. It is possible to implement simple yet effective detectors of bad data, either using simple thresholds (which the attacker cannot know in advance) or by detecting significant deviations from the expected reported states[534]. Even with such measures, small changes effected by attackers may incrementally mount to large consequences in the operation of CPSs, and still, an attacker might adopt conservative strategies to minimize the chances of being detected. As such, it is apparent that real-time detection should ideally be paired with longitudinal monitoring of system behaviour in order to detect such cumulative effects on the system.

More recently, the rise of popularity (and accessibility) of machine learning tools has led to the recommendation for applying such techniques (especially deep learning neural networks) to detect reliability or security issues [535]. One drawback of these approaches is that although a trained classifier can work in real-time to detect threats, on the cloud, fog, or even edge level [536], the training process has to be performed typically offline, and particularly so when the training data consists a large volume. Hence such classifiers cannot be re-trained online and require multi-tier architectures [537] for their implementation (e.g. online for detection, near-line for model tuning, off-line for training).

The common goal of almost all defences against integrity attacks on machine learning methods performed in the CPS (as part of the application layer) is to reduce the influence of adding invalid data points to the result. These invalid data points are thus deemed outliers in the training set. Rubinstein et al. have designed a defence framework against poisoning attacks based on robust statistics to alleviate the effect of poisoning [538]. In addition, a bagging defence against such integrity attacks has been proposed by Biggio et al. [539]. They examine the effectiveness of using bagging, i.e., a machine learning method that generates multiple versions of a predictor and utilizes them to get an aggregated predictor by getting averages over the versions or using a plurality vote, in reducing the influence of outlying observations on training data.

Also, the introduction of trusted computing as part of the security-by-design approach can also provide a proactive countermeasure against possible attacks on CPS devices. Latest processor technologies provide trusted execution environment (TEE) generation that can be used for security sensitive software execution. Such execution environments cannot be accessed by attackers to install malicious code or alter existing software code since all activities are monitored. For example, ARM offers the ARMTrustzone TEE for all its cortex A and in some of its cortex M processor family. Dedicated hardware tokens can also be placed in non-embedded system CPS devices like Trusted Platform Modules (TPMs) in order to install security and trust on control management subsystems of a CPS [540].

Furthermore, securing intelligently selected subsets of data, or strategically introducing secure infrastructure at key areas of a CPS, like hardware security tokens [441], may act as a strong deterrent, especially in large scale systems such as power grids, since such approaches make it extremely hard and impractical for an attacker to effect more than small and inconsequential compromises to the system's integrity [541].

Additionally, side-channel signal analysis can provide an effective approach for the detection of some hardware-based security issues in Device hardware. A characteristic example is the assessment of side channel attacks through side channel analysis using techniques like t-test Leakage assessment. Apart from that, side-channel signals, including timing [542][543], power [544], and spatial temperature [545] can be used for Hardware Trojan detection (hardware trojans can interrupt trustworthy manufacturing of CPS devices). Also, the side channel information can be used for malicious firmware/software detection by revealing abnormal behaviour of the device, e.g., a significant increase in its power consumption, which are the results of a malware installed on the device.

Finally, monitoring of malicious behaviour can also be done using antimalware tools and IDSs that can detect the existence of a malicious node that tries to inject invalid information into the system or violate the policies. Several recent research efforts have proposed IDS based methods to address code injection issue [546][547] at run-time.

## 7.2.5 Assessment of the Security of a System

There are multiple methodologies for measuring the security of a system, some of them proposing specific KPIs. Here are a few of them listed:

- **Assessment of the severity of a detected security issue in the system**

Based on the DREAD methodology [549] to rate, compare and prioritize the severity of the risk presented by a security issue detected in the system. The methodology gives scores to each of the following five aspects:

- **Damage (D)**: indicates the impact against the assets of the infrastructure, which, depending on the degree of damage, might affect the correct operation of the device.
- **Reproducibility (R)**: indicates how easy the threats are to be repeated.
- **Exploitability (E)**: indicates the expertise required to be able to exploit this threat.
- **Affected user (A)**: different criteria can be chosen here. It can indicate the number of users affected for the threat or can also indicate the importance of the users affected (user admins vs restricted users).
- **Discoverability (d)**: indicates how easy or difficult is to discover the threat.

Another alternative for a specific type of security incidents (data breaches) is to use the GDPR-related Data Breach Severity Assessment Methodology of ENISA [550]. This methodology considers the impact of a data breach in the rights and freedoms of the individuals, when the security incident affects assets that process/store personal data of individuals.

- **Assessment of the impact that a detected security issue has in the system**

Based on the STRIDE methodology to assess the impact that a detected security issue has on the security properties of each asset of the system. Security properties to consider are: Confidentiality, Integrity, Availability, Authenticity, Non-repudiability, Authorization.

- **Criticality of a detected security issue**

The H2020 Project ANASTACIA proposes a way of measuring the criticality of a detected security issue as a combination of the severity of the issue and the impact it has in the requirements of the system. [551]

$$\hat{C} = S * \frac{k}{K}; k = (0, \dots, K)$$

With:

$\hat{C}$  = Normalized Criticality

S = Threat severity

K = Maximum requirement impact level

- **Risk associated to a detected security issue**

Based on the OWASP Risk Rating Methodology[552]: Risk=Likelihood \* Impact

Likelihood is measured by:



- **Threat agent factors:** skill level, motive, opportunity, size
- **Vulnerability Factors:** ease of discovery, ease of exploit, awareness, intrusion detection

Impact can be measured by:

- **Technical impact:** loss of confidentiality, loss of integrity, loss of accountability
- **Business impact:** financial damage, reputational damage, non-compliance, privacy violation.

## 8 Conclusions

The CPSoSASware project aims to provide adaptive, cognitive, decentralized reconfigurable support for the design operation continuum for the full lifecycle of CPSoS, which entails five phases: Requirements, Design, Simulation, Operation and Monitoring. This report reviewed the state-of-the-art and discussed the most relevant methodologies, techniques and solutions that can support the implementation of the CPSoSASware architecture, with special attention to those that can be applied to the specific context of the two project use cases: Autonomous vehicles and Human-robot interaction in manufacturing.

The aim of this document is to present a landscape of possibilities for the technical work-packages in the project, including the other tasks in WP1, to facilitate the identification of most prominent candidates and the selection of those techniques and tools that will be later adapted and extended to fit the objectives of CPSoSASware.

## References

- [1] <https://www.cerbero-h2020.eu/>
- [2] O. Gotel, A. Finkelstein, "An Analysis of the Requirements Traceability Problem", 1st International Conference on Requirements Engineering (ICRE'94), Colorado Springs, April 1994, pp. 94-101.
- [3] Rational DOORS.  
[https://www.ibm.com/support/knowledgecenter/SSYQBZ\\_9.5.0/com.ibm.doors.requirements.doc/topics/c\\_welcome.html](https://www.ibm.com/support/knowledgecenter/SSYQBZ_9.5.0/com.ibm.doors.requirements.doc/topics/c_welcome.html)
- [4] PTC Integrity Lifecycle Manager (formerly MKS Integrity)  
<https://www.ptc.com/en/products/plm/plm-products/windchill/rv-s>
- [5] D. Watzenig, M. Horn, "Introduction to Automated Driving", (2017), 10.1007/978-3-319-31895-0\_1
- [6] EU NCAP (European New Car Assessment Programme) and US NCAP (US New Car Assessment Programme). <http://www.globalncap.org/>
- [7] DELIVERABLES OF THE CERBERO PROJECT.
- [8] "SysML Open Source Project - What is SysML? Who created it?," [Online]. Available: <https://sysml.org>.
- [9] "International Council on Systems Engineering Website" [Online]. Available: <https://www.incose.org>.
- [10] "OMG | Object Management Group" [Online]. Available: <https://www.omg.org/>.
- [11] "Modelica Language — Modelica Association," [Online]. Available: <https://www.modelica.org/modelicalanguage>.
- [12] E. A. L. a. S. Neuendorffer, "MoML — A Modelling Markup Language in XML — Version 0.4," 14 March 2000. [Online]. Available: [https://ptolemy.berkeley.edu/publications/papers/00/moml/moml\\_erl\\_memo.pdf](https://ptolemy.berkeley.edu/publications/papers/00/moml/moml_erl_memo.pdf).
- [13] "Verilog," [Online]. Available: <https://en.wikipedia.org/wiki/Verilog>.
- [14] "VHDL," [Online]. Available: <https://en.wikipedia.org/wiki/VHDL>.
- [15] "Simulink - Simulation and Model-Based Design," [Online]. Available: <https://www.mathworks.com/products/simulink.html>.
- [16] "MathWorks - Makers of MATLAB and Simulink," [Online]. Available: <https://www.mathworks.com>.
- [17] "Simscape — MATLAB and Simulink," [Online]. Available: <https://www.mathworks.com/products/simscape.html>.
- [18] "SimEvents — MATLAB and Simulink," [Online]. Available: <https://www.mathworks.com/products/simevents.html>.
- [19] "Cyber-Physical Systems," [Online]. Available: <https://www.mathworks.com/discovery/cyber-physical-systems.html>.
- [20] "Synopsys | EDA Tools, Semiconductor IP and Application Security Solutions," [Online]. Available: <https://www.synopsys.com/>.
- [21] "Saber Power Electronics," [Online]. Available: <https://www.synopsys.com/verification/virtual-prototyping/saber.html>
- [22] "Modelling Languages," [Online]. Available: <https://www.synopsys.com/verification/virtual-prototyping/saber/capabilities/languages.html>
- [23] "Virtualizer," [Online]. Available: <https://www.synopsys.com/verification/virtual-prototyping/virtualizer.html>
- [24] "gem5: Learning gem5," [Online]. Available: [https://www.gem5.org/documentation/learning\\_gem5/introduction/](https://www.gem5.org/documentation/learning_gem5/introduction/).

- [25] Black G., Binkert N., Reinhardt S.K., Saidi A. (2010) Modular ISA-Independent Full-System Simulation. In: Leupers R., Temam O. (eds) Processor and System-on-Chip Simulation. Springer, Boston, MA
- [26] Milo M. K. Martin, Daniel J. Sorin, Bradford M. Beckmann, Michael R. Marty, Min Xu, Alaa R. Alameldeen, Kevin E. Moore, Mark D. Hill, and David A. Wood. 2005. Multifacet’s general execution-driven multiprocessor simulator (GEMS) toolset. SIGARCH Comput. Archit. News 33, 4 (November 2005), 92–99. DOI:<https://doi.org/10.1145/1105734.1105747>
- [27] “Comparison of instruction set architectures,” [Online]. Available: [https://en.wikipedia.org/wiki/Comparison\\_of\\_instruction\\_set\\_architectures](https://en.wikipedia.org/wiki/Comparison_of_instruction_set_architectures)
- [28] “Vitis Unified Software Platform,” [Online]. Available: <https://www.xilinx.com/products/design-tools/vitis.html>.
- [29] “Alveo,” [Online]. Available: <https://www.xilinx.com/products/boards-and-kits/alveo.html>.
- [30] “TensorFlow,” [Online]. Available: <https://www.tensorflow.org>.
- [31] “Caffe | Deep Learning Framework,” [Online]. Available: <http://caffe.berkeleyvision.org>.
- [32] “Ptolemy II FAQ,” [Online]. Available: <https://ptolemy.berkeley.edu/ptolemyII/ptIIfaq.htm#ptolemy%20II%20description>.
- [33] “cyphysim” [Online]. Available: <https://ptolemy.berkeley.edu/projects/chess/cyphysim/>
- [34] Xilinx Inc. „Zynq-7000 All Programmable SoC Overview”, DS190 (v1.8), 2015.
- [35] Valgrind Developers. <http://valgrind.org>, date of access: August 2015.
- [36] OpenSource project. <http://oprofile.sourceforge.net>, date of access: August 2015.
- [37] GWT-TUD GmbH. <https://www.vampir.eu>, date of access: August 2015.
- [38] “Preventive and Predictive Maintenance Concepts - Industrial Wiki,” [Online]. Available: <https://www.myodesie.com/wiki/index/returnEntry/id/2965>.
- [39] Beaurepaire, P., Valdebenito, M. A., Schuëller, G. I., & Jensen, H. A., “Reliability-based optimization of maintenance scheduling of mechanical components under fatigue,” *Computer Methods in Applied Mechanics and Engineering*, vol. 221–222, pp. 24-40, 2012.
- [40] Linard, A., & Bueno, M. L., “Towards Adaptive Scheduling of Maintenance for Cyber-Physical Systems,” Springer International Publishing, 2012, pp. 134-150.
- [41] Hartley LR. *Fatigue and driving: Driver impairment, driver fatigue, and driving simulation*. Routledge; 2018 Oct 31.
- [42] Wei J, Dolan JM, Litkouhi B. A prediction-and cost function-based algorithm for robust autonomous freeway driving. In 2010 IEEE Intelligent Vehicles Symposium 2010 Jun 21 (pp. 512-517).
- [43] F. Camci, “System Maintenance Scheduling With Prognostics Information Using Genetic Algorithm,” *IEEE Transactions on Reliability*, vol. 58, no. 3, pp. 539-552, September 2009.
- [44] Anil Mital, Anoop Desai, Anand Subramanian, Aashi Mital, *Designing for Maintenance, Product Development (Second Edition)*, Elsevier, 2014, pp. 203-268.
- [45] “www.esteco.com,” [Online]. Available: <https://www.esteco.com/modelfrontier>.
- [46] Box, G.E.; Hunter, J.S.; Hunter, W.G. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*, 2nd Edition. Wiley.
- [47] Huda, S., & Al-Shiha, A. A. (1999). On D-optimal designs for estimating slope. *Sankhyā: The Indian Journal of Statistics, Series B*, 488-495.
- [48] Murata, T., & Ishibuchi, H. (1995, November). MOGA: multi-objective genetic algorithms. In *IEEE international conference on evolutionary computation (Vol. 1, pp. 289-294)*.
- [49] Deb K., Agrawal S., Pratap A., Meyarivan T. (2000) A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In: Schoenauer M. et al. (eds) *Parallel Problem Solving from Nature PPSN VI. PPSN 2000. Lecture Notes in Computer Science*, vol 1917. Springer, Berlin, Heidelberg

- [50] Sasaki, D. (2004). Adaptive range multi-objective genetic algorithms for aerodynamic design problems (Doctoral dissertation, Phd Thesis, Department of System Information Sciences, Tohoku University, Japan).
- [51] Mostaghim, S., & Teich, J. (2003, April). Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706) (pp. 26-33). IEEE.
- [52] Smith, K. I., Everson, R. M., & Fieldsend, J. E. (2004, June). Dominance measures for multi-objective simulated annealing. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753) (Vol. 1, pp. 23-30). IEEE.
- [53] Knowles, J., & Corne, D. (1999, July). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406) (Vol. 1, pp. 98-105). IEEE.
- [54] Ostermeier, A., Gawelczyk, A., & Hansen, N. (1994). A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation*, 2(4), 369-380.
- [55] Hiremath, S. S., Ramakrishnan, R., & Singaperumal, M. (2013, September). Optimization of process parameters in series hydraulic hybrid system through multi-objective function. In 13th Scandinavian International Conference on Fluid Power; June 3-5; 2013; Linköping; Sweden (No. 092, pp. 199-205). Linköping University Electronic Press.
- [56] J. A. Nelder and R. Mead, A Simplex Method for Function Minimization, *Comput. J.*, 1965, 7(4), 308–313, DOI:10.1093/comjnl/7.4.308.
- [57] Kirgat, G. S., & Surde, A. N. (2014). Review of Hooke and Jeeves direct search solution method analysis applicable to mechanical design engineering. *Int. J. Innov. Eng. Res. Technol*, 1, 1-14.
- [58] Li, W. D., Gao, L., Li, X. Y., & Guo, Y. (2008, April). Game theory-based cooperation of process planning and scheduling. In 2008 12th International Conference on Computer Supported Cooperative Work in Design (pp. 841-845). IEEE.
- [59] <https://github.com/Hvass-Labs/swarmops>
- [60] <http://www.vrand.com/products/dot-optimization/>
- [61] Head, J. D., & Zerner, M. C. (1985). A Broyden—Fletcher—Goldfarb—Shanno optimization procedure for molecular geometries. *Chemical physics letters*, 122(3), 264-270.
- [62] Fletcher, R., & Reeves, C. M. (1964). Function minimization by conjugate gradients. *The computer journal*, 7(2), 149-154.
- [63] Hambric, S. A. (1995). Approximation techniques for broad-band acoustic radiated noise design optimization problems.
- [64] Chen, T. Y. (1993). Calculation of the move limits for the sequential linear programming method. *International Journal for Numerical Methods in Engineering*, 36(15), 2661-2679.
- [65] Bedair, Osama K. "Analysis of stiffened plates under lateral loading using sequential quadratic programming (SQP)." *Computers & Structures* 62.1 (1997): 63-80.
- [66] <http://www.vrand.com/products/bigdot-optimization/>
- [67] Hsu, F. T. (1971). Sequential Unconstrained Minimization Technique (SUMT) for Optimal Production Planning. Kansas State University, Institute for Systems Design and Optimization.
- [68] <https://dakota.sandia.gov>
- [69] Hough, P. D., Kolda, T. G., & Torczon, V. J. (2001). Asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Scientific Computing*, 23(1), 134-156.
- [70] Peckham, S. D., Kelbert, A., Hill, M. C., & Hutton, E. W. (2016). Towards uncertainty quantification and parameter estimation for Earth system models in a component-based modelling framework. *Computers & Geosciences*, 90, 152-161.

- [71] C.D. Perttunen D.R. Jones and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Application*, 79(1):157–181, October 1993.
- [72] G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design: With Applications*. McGraw-Hill, New York, 1984.
- [73] [https://ctk.math.ncsu.edu/Finkel\\_Direct/](https://ctk.math.ncsu.edu/Finkel_Direct/)
- [74] <https://software.sandia.gov/opt++/>
- [75] Gao, Y., Shi, L., & Yao, P. (2000, June). Study on multi-objective genetic algorithm. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No. 00EX393) (Vol. 1, pp. 646-650)*. IEEE.
- [76] <http://www.boeing.com>
- [77] [https://communities.bentley.com/communities/other\\_communities/bentley\\_applied\\_research/w/bentley\\_applied\\_research\\_wiki/5976/darwin-optimization-framework](https://communities.bentley.com/communities/other_communities/bentley_applied_research/w/bentley_applied_research_wiki/5976/darwin-optimization-framework)
- [78] Whitley, L. D., Beveridge, J. R., Guerra-Salcedo, C., & Graves, C. R. (1997, July). Messy Genetic Algorithms for Subset Feature Selection. In *ICGA* (pp. 568-575).
- [79] <https://www.phoenix-int.com/product/modelcenter-integrate/>
- [80] "Engineering Software Products | Model Based Engineering | Phoenix Integration," [Online]. Available: <http://www.phoenix-int.com/software/phx-modelcenter.php>.
- [81] "Isight & SIMULIA Execution Engine | Dassault Systèmes®," [Online]. Available: <http://www.3ds.com/products/simulia/portfolio/isight-simulia-execution-engine/overview/>.
- [82] Asadi, Nooshin; Zilouei, Hamid (March 2017). "Optimization of organosolv pretreatment of rice straw for enhanced biohydrogen production using *Enterobacter aerogenes*". *Bioresource Technology*. 227: 335–344.
- [83] Massart DL, Dijkstra A, Kaufman L. Evaluation and optimization of laboratory methods and analytical procedures. A survey of statistical and mathematical techniques. Amsterdam: Elsevier; 1978. p. 1–327.
- [84] George Box, Donald Behnken, "Some new three level designs for the study of quantitative variables", *Technometrics*, Volume 2, pages 455–475, 1960.
- [85] Owen, A.B. (1992). "Orthogonal arrays for computer experiments, integration and visualization". *Statistica Sinica*. 2: 439–452.
- [86] Kai Yang and Basem El-Haik. "Taguchi's Orthogonal Array Experiment," in *Design for Six Sigma: A Roadmap for Product Development*, McGraw-Hill, 2008, pp. 469-497.
- [87] Lohr, Sharon L. (1999). *Sampling: Design and analysis*. Duxbury.
- [88] Ingber, L. (1993). *Adaptive simulated annealing (ASA)*. Global optimization C-code, Caltech Alumni Association, Pasadena, CA.
- [89] Johar, F. M., Azmin, F. A., Suaidi, M. K., Shibghatullah, A. S., Ahmad, B. H., Salleh, S. N., ... & Shukor, M. M. (2013, December). A review of genetic algorithms and parallel genetic algorithms on graphics processing unit (GPU). In *2013 IEEE International Conference on Control System, Computing and Engineering* (pp. 264-269). IEEE..
- [90] Abbass, H. A. (2002, May). The self-adaptive pareto differential evolution algorithm. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) (Vol. 1, pp. 831-836)*.
- [91] Gill, P. E., & Wong, E. (2012). Sequential quadratic programming methods. In *Mixed integer nonlinear programming* (pp. 147-224). Springer, New York, NY.
- [92] Schittkowski, K. (1983). On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function. *Mathematische Operationsforschung und Statistik. Series Optimization*, 14(2), 197-216.
- [93] Venter, G. (2010). Review of optimization techniques. *Encyclopedia of aerospace engineering*.

- [94] Gould, N., Orban, D., & Toint, P. (2005). Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, 14, 299.
- [95] Rozvany, G. I. N. (2001). Stress ratio and compliance based methods in topology optimization—a critical review. *Structural and Multidisciplinary Optimization*, 21(2), 109-119.
- [96] Ravi, R., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J., & Hunt III, H. B. (1993, June). Many birds with one stone: Multi-objective approximation algorithms. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing* (pp. 438-447).
- [97] “OptiY - Leader in CAE-based Design for Reliability and Quality,” [Online]. Available: <http://www.optiy.eu/>.
- [98] Burhenne, S., Jacob, D., & Henze, G. P. (2011, November). Sampling based on Sobol’ sequences for Monte Carlo techniques applied to building simulations. In *Proc. Int. Conf. Build. Simulat* (pp. 1816-1823).
- [99] Lam, C. Q. (2008). Sequential adaptive designs in computer experiments for response surface model fit (Doctoral dissertation, The Ohio State University).
- [100] Lee, S. H., & Kwak, B. M. (2006). Response surface augmented moment method for efficient reliability analysis. *Structural safety*, 28(3), 261-272.
- [101] “Nexus - Optimization - iChrome,” [Online]. Available: <http://ichrome.eu/nexus/>.
- [102] <https://www.ansys.com/products/fluids/ansys-fluent>
- [103] <https://www.simuleon.com/simulia-abaqus/>
- [104] <https://www.mssoftware.com/product/msc-nastran>
- [105] <https://altairhyperworks.com/product/radioss>
- [106] [http://www.hpcadvisorycouncil.com/events/2014/swiss-workshop/presos/Day\\_3/8\\_NVIDIA.pdf](http://www.hpcadvisorycouncil.com/events/2014/swiss-workshop/presos/Day_3/8_NVIDIA.pdf)
- [107] Bradley, J. V. (1958). Complete counterbalancing of immediate sequential effects in a Latin square design. *Journal of the American Statistical Association*, 53(282), 525-528.
- [108] R.L. Plackett and J.P. Burman, "The Design of Optimum Multifactorial Experiments", *Biometrika* 33 (4), pp. 305–25, June 1946
- [109] Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis* (pp. 105-116). Springer, Berlin, Heidelberg.
- [110] Jorge Nocedal and Stephen J. Wright (2006). *Numerical Optimization*. Springer.
- [111] <https://www.solidworks.com>
- [112] <http://www.lstc.com>
- [113] <https://www.solidworks.com/category/simulation-solutions>
- [114] “Our solutions - Cyberdyne,” [Online]. Available: <https://cyberdyne.it/our-solutions/>.
- [115] Ozcan, E.; Basaran, C. (2009). "A Case Study of Memetic Algorithms for Constraint Optimization". *Soft Computing: A Fusion of Foundations, Methodologies and Applications*. 13 (8–9): 871–882.
- [116] Monsef, H., Naghashadegan, M., Jamali, A., & Farmani, R. (2019). Comparison of evolutionary multi objective optimization algorithms in optimum design of water distribution network. *Ain Shams Engineering Journal*, 10(1), 103-111.
- [117] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm, Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich (2001)
- [118] “PACE GmbH,” [Online]. Available: <http://www.pace.de/products/preliminary-design/pacelab-suite.html>.
- [119] “PACE GmbH,” [Online]. Available: <http://www.pace.de/?id=40>.
- [120] “HEEDS MDO,” [Online]. Available: <https://www.redcedartech.com/index.php/solutions/heeds-software>.
- [121] “Sigma Technology. Products.,” [Online]. Available: <http://iosotech.com/product.htm>.

- [122] "Software Solutions that enable Objectives Based Engineering | Noesis Solutions," [Online]. Available: <http://www.noesisolutions.com/Noesis/>.
- [123] Faco, J. L. D. (1989). A generalized reduced gradient algorithm for solving large-scale discrete-time nonlinear optimal control problems. *IFAC Proceedings Volumes*, 22(2), 45-50.
- [124] Qin, A. K., & Suganthan, P. N. (2005, September). Self-adaptive differential evolution algorithm for numerical optimization. In *2005 IEEE congress on evolutionary computation* (Vol. 2, pp. 1785-1791). IEEE.
- [125] Younis, A., & Dong, Z. (2010). Trends, features, and tests of common and recently introduced global optimization methods. *Engineering Optimization*, 42(8), 691-718.
- [126] Sevastyanov, V. (2010). Gradient-based multi-objective optimization technology. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference* (p. 9092).
- [127] D. V. Strimling, "Multi-Criteria/Multi-Domain Optimization and Target Cascading for System Level Design," *The voice of the systems*, vol. 11, pp. 22-38, 2013.
- [128] Zeidner, L., Reeve, H., Khire, R., & Becz, S. (2010). Architectural enumeration and evaluation for identification of low-complexity systems. In *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference* (p. 9264).
- [129] Broodney, H., Masin, M., Shindin, E., Shani, U., Kalawsky, R., Joannou, D., ... & Sanduka, I. (2015). Leveraging Domain Expertise in Architectural Exploration. In *Complex Systems Design & Management* (pp. 87-103). Springer, Cham.
- [130] <https://www.ibm.com/analytics/cplex-optimizer>
- [131] Masin, M., & Bukchin, Y. (2008). Diversity maximization approach for multiobjective optimization. *Operations Research*, 56(2), 411-424.
- [132] S. Said, S. AlKork, T. Beyrouthy, M. Hassan, O. Abdellatif, M.F. Abdraboo "Real Time Eye Tracking and Detection- A Driving Assistance System", *Advances in Science, Technology and Engineering Systems Journal*, vol. 3, no. 6, pp. 446-454 (2018).
- [133] G.L. Masala, E. Grosso, Real time detection of driver attention: Emerging solutions based on robust iconic classifiers and dictionary of poses, *Transportation Research Part C: Emerging Technologies*, Volume 49, 2014, 32-42.
- [134] S. Boverie, N.Rodriguez, D.Bande, A.Saccagno, "General driver monitoring module definition SoA", 2013
- [135] N. Kose, O. Kopuklu, A. Unnervik and G. Rigoll, "Real-Time Driver State Monitoring Using a CNN Based Spatio-Temporal Approach\*", 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 2019, pp. 3236-3242, doi: 10.1109/ITSC.2019.8917460.
- [136] X. Tang, P. Zhou and P. Wang, "Real-time image-based driver fatigue detection and monitoring system for monitoring driver vigilance," *2016 35th Chinese Control Conference (CCC)*, Chengdu, 2016, pp. 4188-4193, doi: 10.1109/ChiCC.2016.7554007.
- [137] H. Shin, S. Jung, J. Kim and W. Chung, "Real time car driver's condition monitoring system," *SENSORS*, 2010 IEEE, Kona, HI, 2010, pp. 951-954, doi: 10.1109/ICSENS.2010.5690904.
- [138] Robot Operating System <https://www.ros.org/>
- [139] Investopedia V2V/V2X <https://www.investopedia.com/terms/v/v2x-vehicletovehicle-or-vehicletoinfrastructure.asp>
- [140] NHTSA Cybersecurity protection methods <https://www.nhtsa.gov/technology-innovation/vehicle-cybersecurity> (2018)
- [141] Z. Lu, G. Qu, Z. Liu A survey on recent advances in vehicular network security, trust, and privacy *IEEE Trans. Intell. Transp. Syst.* (2018)
- [142] M. Hasan, S. Mohan, T. Shimizuy, H. Luy ,Securing Vehicle-to-Everything (V2X) Communication Platforms



- [143] Robotec Simulation: <https://robotec.ai/services/#robotics>
- [144] Microsoft AirSim: <https://github.com/Microsoft/AirSim>
- [145] Carla Simulator: <https://github.com/carla-simulator/carla>
- [146] Apollo Auto: <https://github.com/ApolloAuto/apollo>
- [147] Deeddrive: <https://github.com/deepdrive/deepdrive>
- [148] LGSVL Simulator: An Autonomous Vehicle Simulator: <https://github.com/lgsvl/simulator>
- [149] Udacity's Self-Driving Car Simulator: <https://github.com/udacity/self-driving-car-sim>
- [150] MADRaS: <https://github.com/madras-simulator/MADRaS>
- [151] W. Wang, J. Xi, H. Chen, Modelling and Recognizing Driver Behaviour Based on Driving Data: A Survey
- [152] A. Liu and D. Salvucci, "Modelling and prediction of humandriver behaviour," in Proceedings of the 9th International Conference on Human-Computer Interaction, New Orleans, La, USA,2001.
- [153] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 1. MIT Press, 1998.
- [154] T. A. Wheeler, P. Robbel, and M. Kochenderfer, "Analysis of microscopic behaviour models for probabilistic modelling of driver behaviour", in IEEE International Conference on Intelligent Transportation Systems (ITSC), 2016
- [155] S. Lef'evre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction" , American Control Conference (ACC), pp. 3494–3499, 2014.
- [156] DBNet dataset <http://www.dbehaviour.net/>
- [157] Comma2k19 dataset <https://github.com/commaai/comma2k19>
- [158] Honda Research Institute Driving Dataset <https://usa.honda-ri.com/hdd>
- [159] T. Queck, B. Schunemann, I. Radusch, C. Meinel, Realistic Simulation of V2X Communication Scenarios
- [160] A. Choudhury, T. Maszczyk, C. B. Math, H. Li, J. Dauwels, An integrated simulation environment for testing V2X protocols and applications, Nanyang Technological University, Singapore
- [161] SUMO <https://civitas.eu/tool-inventory/sumo-simulation-urban-mobility>
- [162] MATLAB SIMULINK <https://www.mathworks.com/help/simulink/>
- [163] D. Eckhoff, C. Sommer. "A multi-channel IEEE 1609.4 and 802.11 p EDCA model for the veins framework." Proceedings of 5th ACM/ICST international conference on simulation tools and techniques for communications, networks and systems: 5th ACM/ICST international workshop on OMNet++. (Desenzano, Italy, 19-23 March, 2012). OMNeT. 2012.
- [164] Viridis, A., Stea, G., & Nardini, G. (2014, August). SimuLTE-A modular system-level simulator for LTE/LTE-A networks based on OMNeT++. In 2014 4th International Conference On Simulation And Modelling Methodologies, Technologies And Applications (SIMULTECH) (pp. 59-70). IEEE.
- [165] Vanetza, <https://github.com/riehl/vanetza>
- [166] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," IEEE Trans. Mobile Comput., vol. 10, no. 1, pp. 3–15, 2011.
- [167] Simulation of cooperative automated driving by bidirectional coupling of vehicle and network simulators Conference Paper, June 2017, Simulation in V2X
- [168] M. Rondinone et al., "iTETRIS: A Modular Simulation Platform for the Large Scale Evaluation of Cooperative ITS Applications," Elsevier Simulation Modelling Practice and Theory, vol. 34, pp. 99–125, 2013.
- [169] C. Laurendeau and M. Barbeau, "Threats to security in DSRC/WAVE," in AdHoc-Now, 2006, pp. 266–279
- [170] SAE J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems
- [171] STANDARD ISO/SAE 21434, <https://www.security-analyst.org/iso-sae-21434-a-field-report/>

- [172] L. He, W.T. Zhu, Mitigating DoS attacks against signature-based authentication in VANETs, in: IEEE International Conference on Computer Science and Automation Engineering, 2012, pp.261–265.
- [173] R.A.R. Mahmood, A.I. Khan, A survey on detecting black hole attack in AODV-based mobile ad hoc networks, in: International Symposium on High Capacity Optical Networks and Enabling Technologies, 2007, <https://www.researchgate.net/publication/4362772>.
- [174] H. Shin, D. Kim, Y. Kwon, Y. Kim, Illusion and dazzle: adversarial optical channel exploits against lidars for automotive applications, in: Cryptographic Hardware and Embedded Systems – CHES 2017, 2017, pp.445–467.
- [175] C. Yan, W. Xu, J. Liu, Can you trust autonomous vehicles: contactless attacks against sensors of self-driving vehicle, in: DEFCON, 2016.
- [176] D. Singelee, B. Preneel, Location verification using secure distance bounding protocols, in: IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.
- [177] S. Park, B. Aslam, D. Turgut, C.C. Zou, Defense against sybil attack in the initial deployment stage of vehicular ad hoc network based on roadside unit support, Secur. Commun. Netw. 6 (2013) 523–538.
- [178] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, I. Seskar, Security and privacy vulnerabilities of in-car wireless networks: a tire pressure monitoring system case study, in: USENI
- [179] S. Narain, A. Ranganathan, G. Noubir, Security of GPS/INS based on-road location tracking systems, Computing Research Repository (2018).
- [180] S. Bittl, A.A. Gonzalez, M. Myrtus, H. Beckmann, S. Sailer, B. Eissfeller, Emerging attacks on VANET security based on GPS time spoofing, in: 2015 IEEE Conference on Communications and Network Security, 2015, pp.344–352.
- [181] T. Leinmuller and E. Schoch, “Greedy routing in highway scenarios: The impact of position faking nodes,” in Proc. of WIT, 2006
- [182] M. Hasan, S. Mohan, T. Shimizuy, H. Luy, Securing Vehicle-to-Everything (V2X) Communication Platforms
- [183] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, “VANET security challenges and solutions: A survey,” Vehicular Communications, vol. 7, pp. 7–20, 2017.
- [184] F. Sakiz and S. Sen, “A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV,” Elsevier Ad Hoc Net., vol. 61, pp. 33–50, 2017
- [185] J. Petit, S.E. Shladover, Potential cyberattacks on automated vehicles, IEEE Trans. Intell. Transp. Syst. 16 (2015) 546–556.
- [186] R.S. Raw, M. Kumar, N. Singh, Security challenges, issues and their solutions for VANET, Int. J. Netw. Secur. Appl. 5 (2013) 95–105.
- [187] Z. El-Rewini, K. Sadatsharana, D. F. Selvaraja, S. J. Plathottamb, P. Ranganathana, Cybersecurity challenges in vehicular communications
- [188] V.H. La, A. Cavalli, Security attacks and solutions in vehicular ad hoc networks: a survey, Int. J. AdHoc Netw. Syst. 4 (2014).
- [189] Black hole attack, <https://www.sciencedirect.com/topics/computer-science/black-hole-attack>
- [190] Man in the middle attack, <https://www.ssl.com/faqs/what-is-a-man-in-the-middle-attack/>
- [191] F. A. Ghaleb, A. Zainal, and M. A. Rassam, “Data verification and misbehaviour detection in vehicular ad-hoc networks,” J. Teknologi, vol. 73, no. 2, pp. 37–44, 2015.
- [192] H. Shin, D. Kim, Y. Kwon, Y. Kim, Illusion and dazzle: adversarial optical channel exploits against lidars for automotive applications, in: Cryptographic Hardware and Embedded Systems – CHES 2017, 2017, pp.445–467.

- [193] R.S. Raw, M. Kumar, N. Singh, Security challenges, issues and their solutions for VANET, *Int. J. Netw. Secur. Appl.* 5 (2013) 95–105.
- [194] A.Staranowicz, G.L. Mariottini, A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software
- [195] Gazebo: <http://gazebosim.org/>
- [196] Isaac Sim 2020: <https://developer.nvidia.com/isaac-sim>
- [197] Cyberbotics: <https://cyberbotics.com/>
- [198] Coppeliasim: <https://www.coppeliarobotics.com/>
- [199] Siemens Tecnomatix:  
<https://www.plm.automation.siemens.com/global/en/products/tecnomatix/>
- [200] S. Walker , R. Romero, S. Thrun, “A Gesture Based Interface For Human-Robot Interaction”
- [201] Simics. [www.windriver.com/products/simics](http://www.windriver.com/products/simics)
- [202] Yourst M.T. “PTLsim: A Cycle Accurate Full System x86-64 Microarchitectural Simulator”. IEEE International Symposium on Performance Analysis of Systems and Software, 2007, pp. 23–34.
- [203] Austin T., Larson E. and Ernst D.; “SimpleScalar: An Infrastructure for Computer System Modelling”. The 8th IEEE International Symposium on High-Performance Computer Architecture, 2002, pp. 59–67.
- [204] OVPsim: [www.ovpworld.org/technology\\_ovpsim.php](http://www.ovpworld.org/technology_ovpsim.php)
- [205] Binkert N.; et al. “The gem5 simulator”. ACM SIGARCH Computer Architecture News, vol.39 (2), 2011.
- [206] Status Matrix in Gem5, [www.m5sim.org/Status\\_Matrix](http://www.m5sim.org/Status_Matrix)
- [207] Binkert N.; et al. “The gem5 simulator”. ACM SIGARCH Computer Architecture News, vol.39 (2), 2011.
- [208] A. Bakhoda, G. Yuan, W. Fung, H. Wong, and T. M. Aamodt, “Analyzing CUDA workloads using a detailed GPU simulator,” in ISPASS, 2009
- [209] “Macsim,” <https://code.google.com/p/macsim/>.
- [210] J. Meng and K. Skadron, “A reconfigurable simulator for largescale heterogeneous multicore architectures,” in ISPASS 2011, 2011.
- [211] R. Ubal, B. Jang, P. Mistry, D. Schaa, and D. Kaeli, “Multi2sim: A simulation framework for CPU-GPU computing,” in PACT ’12, 2012.
- [212] V. Zakharenko, T. Aamodt, and A. Moshovos, “Characterizing the performance benefits of fused CPU/GPU systems using fusionsim,” in DATE ’13, 2013.
- [213] Emulated CL runtime system: available at <http://old.gem5.org/dist/current/gpu/cl-runtime.tar.xz.html>
- [214] OpenCL compiler. <https://github.com/HSAFoundation/CLOC>.
- [215] Tuor, T., Wang, S., Leung, K. K., & Chan, K. (2018). Distributed Machine Learning in Coalition Environments: Overview of Techniques. 21st International Conference on Information Fusion (FUSION), 2018.
- [216] M. Gharib, L. D. Da Silva, H. Kavalionak and A. Ceccarelli, "A Model-Based Approach for Analyzing the Autonomy Levels for Cyber-Physical Systems-of-Systems," 2018 Eighth Latin-American Symposium on Dependable Computing (LADC), Foz do Iguaçu, Brazil, 2018, pp. 135-144
- [217] Talal Rahwan, Tomasz P. Michalak, Michael Wooldridge, Nicholas R. Jennings, Coalition structure generation: A survey, *Artificial Intelligence*, Volume 229, 2015, 139-174
- [218] Huo Y, Dong W, Qian J, Jing T. Coalition Game-Based Secure and Effective Clustering Communication in Vehicular Cyber-Physical System (VCPS). *Sensors (Basel)*. 2017 Feb 27;17(3):475. doi: 10.3390/s17030475. PMID: 28264469; PMCID: PMC5375761.

- [219] Notarstefano, Giuseppe, Ivano Notarnicola, and Andrea Camisa. "Distributed optimization for smart cyber-physical networks." *Foundations and Trends in Systems and Control* 7.3 (2019): 253-383.
- [220] T. Erseghe, "A distributed and scalable processing method based upon ADMM," *IEEE Signal Processing Letters*, vol. 19, no. 9, pp. 563–566, 2012.
- [221] A. Lalos, E. Vlachos, K. Berberidis, A. Fournaris and C. Koulamas, "Privacy Preservation in Industrial IoT via Fast Adaptive Correlation Matrix Completion," in *IEEE Transactions on Industrial Informatics*, 2019.
- [222] S. Gopalswamy and S. Rathinam, "Infrastructure Enabled Autonomy: A Distributed Intelligence Architecture for Autonomous Vehicles," 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, 2018, pp. 986-992.
- [223] Y. Feng, B. Hu, H. Hao, Y. Gao, Z. Li and J. Tan, "Design of Distributed Cyber-Physical Systems for Connected and Automated Vehicles With Implementing Methodologies," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4200-4211, Sept. 2018.
- [224] R. Hussain and S. Zeadally, "Autonomous Cars: Research Results, Issues, and Future Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275-1313, Second quarter 2019.
- [225] F. Mohseni, S. Voronov, and E. Frisk, "Deep Learning Model Predictive Control for Autonomous Driving in Unknown Environments," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 447–452, 2018, doi: <https://doi.org/10.1016/j.ifacol.2018.11.593>.
- [226] X. Hu, H. Wang, and X. Tang, "Cyber-physical control for energy-saving vehicle following with connectivity," *IEEE Trans. Ind. Electron.*, vol. 64, no. 11, pp. 8578–8587, Nov. 2017.
- [227] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton Univ. Press, 2008, pp. 158–165.
- [228] W. Ren and Y. Cao, *Distributed Coordination of Multi-Agent Networks: Emergent Problems, Models, and Issues*. Dordrecht, Netherlands: Springer-Verlag, 2010, pp. 23–41.
- [229] Lee, Eun-Kyu & Gerla, Mario & Pau, Giovanni & Lee, Uichin & Lim, Jae-Han, "Internet of Vehicles: From intelligent grid to autonomous cars and vehicular Fogs," *International Journal of Distributed Sensor Networks*, 2016.
- [230] K. Goldberg, "Robots and the return to collaborative intelligence," *Nature Machine Intelligence*, pp. 2–4, 2019.
- [231] J. Wan, S. Tang, H. Yan, D. Li, S. Wang, and A. V. Vasilakos, "Cloud robotics: Current status and open issues," *IEEE Access*, vol. 4, pp. 2797–2807, 2016.
- [232] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, ser. Mobidata '15. ACM, 2015, pp. 37–42.
- [233] G. Fuseiller, R. Marie, G. Mourioux, E. Duno and O. Labbani-Igbida, "Reactive path planning for collaborative robot using configuration space skeletonization," 2018 IEEE International Conference on Simulation, Modelling, and Programming for Autonomous Robots (SIMPAN), Brisbane, QLD, 2018, pp. 29-34
- [234] Tanwani, Ajay & Mor, Nitesh & Kubiawicz, John & Gonzalez, Joseph & Goldberg, Kenneth. (2019). *A Fog Robotics Approach to Deep Robot Learning: Application to Object Recognition and Grasp Planning in Surface Decluttering*.
- [235] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Robotics and Automation (ICRA)*, 2012 IEEE International Conference.
- [236] M. Ragaglia, A. M. Zanchettin, and P. Rocco, "Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements," *Mechatronics*, 2018.

- [237] Casalino, Andrea & Bazzi, Davide & Zanchettin, Andrea Maria & Rocco, Paolo. (2019). Optimal Proactive Path Planning for Collaborative Robots in Industrial Contexts. 10.1109/ICRA.2019.8793847.
- [238] Chand GUDI, Siva Leela Krishna & Ojha, Suman & Johnston, Benjamin & Clark, Jesse & Williams, Mary-Anne. (2018). Fog Robotics for Efficient, Fluent and Robust Human-Robot Interaction.
- [239] Feng, D. *et al.* Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Trans. Intell. Transp. Syst.* 1–20 (2020) doi:10.1109/tits.2020.2972974.
- [240] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [241] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Computer Vision. Springer, 2014, pp. 740–755.
- [242] Molisch, A. F., Balakrishnan, K., Chong, C. C., Emami, S., Fort, A., Karedal, J., ... & Siwiak, K. (2004). IEEE 802.15. 4a channel model-final report. IEEE P802, 15(04), 0662.
- [243] Alliance, Z. (2010). Zigbee alliance. WPAN industry group, <http://www.zigbee.org/>. The industry group responsible for the ZigBee standard and certification.
- [244] Haartsen, J. C. (2000). The Bluetooth radio system. *IEEE personal communications*, 7(1), 28-36.
- [245] Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9), 11734-11753
- [246] Perahia, E., & Stacey, R. (2013). Next generation wireless LANs: 802.11 n and 802.11 ac. Cambridge university press
- [247] Ong, E. H., Kneckt, J., Alanen, O., Chang, Z., Huovinen, T., & Nihtilä, T. (2011, September). IEEE 802.11 ac: Enhancements for very high throughput WLANs. In 2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications (pp. 849-853). IEEE.
- [248] Adame, T., Bel, A., Bellalta, B., Barcelo, J., & Oliver, M. (2014). IEEE 802.11 ah: the WiFi approach for M2M communications. *IEEE Wireless Communications*, 21(6), 144-152.
- [249] Ratasuk, R., Mangalvedhe, N., Ghosh, A., & Vejlgaard, B. (2014, September). Narrowband LTE-M system for M2M communication. In 2014 IEEE 80th vehicular technology conference (VTC2014-Fall) (pp. 1-5). IEEE.
- [250] Hiertz, G. R., Denteneer, D., Max, S., Taori, R., Cardona, J., Berlemann, L., & Walke, B. (2010). IEEE 802.11 s: the WLAN mesh standard. *IEEE Wireless Communications*, 17(1), 104-111.
- [251] M. Weber, P. Wolf, and J. M. Zollner, “DeepTLR: A single deep convolutional network for detection and classification of traffic lights,” in *IEEE Intelligent Vehicles Symp.*, 2016, pp. 342–348.
- [252] M. Abolhasan, B. Hagelstein, and J.-P. Wang, “Real-world performance of current proactive multi-hop mesh protocols,” in *2009 15th Asia-Pacific Conference on Communications*, 2009, pp. 44–47.
- [253] J. Müller and K. Dietmayer, “Detecting traffic lights by single shot detection,” in 21st Int. Conf. Intelligent Transportation Systems. IEEE, 2016, pp. 342–348.
- [254] M. Bach, S. Reuter, and K. Dietmayer, “Multi-camera traffic light recognition using a classifying labeled multi-bernoulli filter,” in *IEEE Intelligent Vehicles Symp.*, 2017, pp. 1045–1051.
- [255] K. Behrendt, L. Novak, and R. Botros, “A deep learning approach to traffic lights: Detection, tracking, and classification,” in *IEEE Int. Conf. Robotics and Automation*, 2017, pp. 1370–1377.
- [256] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-sign detection and classification in the wild,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2110–2118.
- [257] H. S. Lee and K. Kim, “Simultaneous traffic sign detection and boundary estimation using convolutional neural network,” *IEEE Trans. Intell. Transp. Syst.*, 2018.

- [258] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, "Traffic sign recognition using a multi-task convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1100–1111, 2018.
- [259] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards reaching human performance in pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 973–986, 2018.
- [260] L. Zhang, L. Lin, X. Liang, and K. He, "Is Faster [309] doing well for pedestrian detection?" in *Proc. Eur. Conf. Computer Vision*. Springer, 2016, pp. 443–457.
- [261] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals using stereo imagery for accurate object class detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1259–1272, 2018.
- [262] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2017, pp. 1513–1518.
- [263] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," in *Proc. Robotics: Science and Systems*, Jun. 2016.
- [264] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.
- [265] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1782–1792, 2017.
- [266] Mousavian, D. Anguelov, J. Flynn, and J. Kořecká, "3d bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 5632–5640.
- [267] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Int. Conf. Learning Representations*, 2013.
- [268] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
- [269] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [270] R. Girshick, "Fast R-CNN," in *Proc. IEEE Conf. Computer Vision*, 2015, pp. 1440–1448.
- [271] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556 [cs.CV]*, 2014.
- [272] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [273] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
- [274] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
- [275] Geiger, A., Lenz, P., Stiller, C. & Urtasun, R, "Vision meets robotics: The KITTI dataset" in *International Journal of Robotics Research (IJRR)*, 2013, pp. 1231–1237.
- [276] A. Luckow, M. Cook, N. Ashcraft, E. Weill, E. Djerekarov, and B. Vorster, "Deep learning in the automotive industry: Applications and tools," in *2016 IEEE International Conference on Big Data (Big Data)*, Dec. 2016, pp. 3759–3768, doi: 10.1109/BigData.2016.7841045.

- [277] J. Yang, S. Li, Z. Wang, and G. Yang, "Real-Time Tiny Part Defect Detection System in Manufacturing Using Deep Learning," *IEEE Access*, vol. 7, pp. 89278–89291, 2019, doi: 10.1109/ACCESS.2019.2925561.
- [278] T. Nakazawa and D. V. Kulkarni, "Anomaly Detection and Segmentation for Wafer Defect Patterns Using Deep Convolutional Encoder–Decoder Neural Network Architectures in Semiconductor Manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 2, pp. 250–256, May 2019, doi: 10.1109/TSM.2019.2897690.
- [279] J.-S. Wang, A. Ambikapathi, Y. Han, S.-L. Chung, H.-W. Ting, and C.-F. Chen, "Highlighted Deep Learning based Identification of Pharmaceutical Blister Packages," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2018, vol. 1, pp. 638–645, doi: 10.1109/ETFA.2018.8502488.
- [280] W. Tao, Z.-H. Lai, M. C. Leu, Z. Yin, and R. Qin, "A self-aware and active-guiding training & assistant system for worker-centered intelligent manufacturing," *Manufacturing Letters*, vol. 21, pp. 45–49, Aug. 2019, doi: 10.1016/j.mfglet.2019.08.003.
- [281] P. Wei, L. Cagle, T. Reza, J. Ball, and J. Gafford, "LiDAR and Camera Detection Fusion in a Real-Time Industrial Multi-Sensor Collision Avoidance System," *Electronics*, vol. 7, no. 6, p. 84, Jun. 2018, doi: 10.3390/electronics7060084.
- [282] Janai, J., Güney, F., Behl, A. & Geiger, A. *Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art.*, 2017.
- [283] Cai, Z., Fan, Q., Feris, R. S. & Vasconcelos, N. "A unified multi-scale deep convolutional neural network for fast object detection" in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016
- [284] Xiang, Y., Choi, W., Lin, Y. & Savarese, S. "Subcategory-aware convolutional neural networks for object proposals and detection" *arXiv.org*, 1604.04693, 2016
- [285] Yang, F., Choi, W. & Lin, Y. "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers" in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016
- [286] Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V. & Schiele, B. "Deepcut: Joint subset partition and labelling for multi person pose estimation" in *Proc. IEEE Conf. on Computer Vision and Pattern*, 2016
- [287] Alvaro Arcos-Garcia, Juan A. Alvarez-Garcia, Luis M. Soria-Morillo, "Evaluation of Deep Neural Networks for traffic sign detection systems", *Neurocomputing*, doi: <https://doi.org/10.1016/j.neucom.2018.08.009>, 2018
- [288] Kirillov, A., He, K., Girshick, R., Rother, C. & Dollar, P. "Panoptic segmentation" *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **2019-June**, 9396–9405 (2019).
- [289] M. Siam, S. Valipour, M. Jagersand, and N. Ray. "Convolutional gated recurrent networks for video segmentation" *arXiv preprint arXiv:1611.05435*, 2016.
- [290] Siam, M. et al. A comparative study of real-time semantic segmentation for autonomous driving. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.* **2018-June**, 700–710 (2018).
- [291] Behl, A. *et al.* Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios? *Proc. IEEE Int. Conf. Comput. Vis.* **2017-October**, 2593–2602 (2017).
- [292] Wu, B., Iandola, F., Jin, P. H. & Keutzer, K. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.* **2017-July**, 446–454 (2017).
- [293] Tomè, D. *et al.* Deep Convolutional Neural Networks for pedestrian detection. *Signal Process. Image Commun.* **47**, 482–489 (2016).

- [294] Lee, S. *et al.* VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition. *Proc. IEEE Int. Conf. Comput. Vis.* **2017-Octob**, 1965–1973 (2017).
- [295] Kim, D. U., Park, S. H., Ban, J. H., Lee, T. M. & Do, Y. Vision-based autonomous detection of lane and pedestrians. *2016 IEEE Int. Conf. Signal Image Process. ICSIP 2016* **2**, 680–683 (2017).
- [296] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 3354–3361, doi: 10.1109/CVPR.2012.6248074.
- [297] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, “The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario:,” *The International Journal of Robotics Research*, Oct. 2013, doi: 10.1177/0278364913507326.
- [298] X. Huang *et al.*, “The ApolloScape Dataset for Autonomous Driving,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2018, pp. 1067–10676, doi: 10.1109/CVPRW.2018.00141.
- [299] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, “Multispectral pedestrian detection: Benchmark dataset and baseline,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1037–1045, doi: 10.1109/CVPR.2015.7298706.
- [300] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada, “MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 5108–5115, doi: 10.1109/IROS.2017.8206396.
- [301] Y. Choi *et al.*, “KAIST Multi-Spectral Day/Night Data Set for Autonomous and Assisted Driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, Mar. 2018, doi: 10.1109/TITS.2018.2791533.
- [302] M. Meyer and G. Kusch, “Automotive Radar Dataset for Deep Learning Based 3D Object Detection,” in *2019 16th European Radar Conference (EuRAD)*, Oct. 2019, pp. 129–132.
- [303] M. Gadd, “Real-time Kinematic Ground Truth for the Oxford RobotCar Dataset,” *Oxford Robotics Institute*. <https://ori.ox.ac.uk/kidnapped-radar-topological-radar-localisation-using-rotationally-invariant-metric-learning-6/> (accessed May 15, 2020).
- [304] H. Caesar *et al.*, “nuScenes: A multimodal dataset for autonomous driving,” *arXiv:1903.11027 [cs, stat]*, May 2020, Accessed: May 15, 2020. [Online]. Available: <http://arxiv.org/abs/1903.11027>.
- [305] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, “Joint 3D Proposal Generation and Object Detection from View Aggregation,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1–8, doi: 10.1109/IROS.2018.8594049.
- [306] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D Object Detection Network for Autonomous Driving,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6526–6534, doi: 10.1109/CVPR.2017.691.



- [307] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 918–927, doi: 10.1109/CVPR.2018.00102.
- [308] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A General Pipeline for 3D Detection of Vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 3194–3200, doi: 10.1109/ICRA.2018.8461232.
- [309] Taewan Kim and J. Ghosh, "Robust detection of non-motorized road users using deep learning on optical and LIDAR data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 271–276, doi: 10.1109/ITSC.2016.7795566.
- [310] M. Bijelic *et al.*, "Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather," *arXiv:1902.08913 [cs]*, Feb. 2020, Accessed: May 12, 2020. [Online]. Available: <http://arxiv.org/abs/1902.08913>.
- [311] J. Dou, J. Xue, and J. Fang, "SEG-VoxelNet for 3D Vehicle Detection from RGB and LiDAR Data," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 4362–4368, doi: 10.1109/ICRA.2019.8793492.
- [312] Z. Wang and K. Jia, "Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection," *arXiv:1903.01864 [cs]*, Aug. 2019, Accessed: May 12, 2020. [Online]. Available: <http://arxiv.org/abs/1903.01864>.
- [313] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-Task Multi-Sensor Fusion for 3D Object Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 7337–7345, doi: 10.1109/CVPR.2019.00752.
- [314] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada, "Multispectral Object Detection for Autonomous Vehicles," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, Mountain View, California, USA, Oct. 2017, pp. 35–43, doi: 10.1145/3126686.3126727.
- [315] L. Schneider *et al.*, "Multimodal Neural Networks: RGB-D for Semantic Segmentation and Object Detection," in *Image Analysis*, vol. 10269, P. Sharma and F. M. Bianchi, Eds. Cham: Springer International Publishing, 2017, pp. 98–109.
- [316] S. Chadwick, W. Maddern, and P. Newman, "Distant Vehicle Detection Using Radar and Vision," *arXiv:1901.10951 [cs]*, May 2019, Accessed: May 12, 2020. [Online]. Available: <http://arxiv.org/abs/1901.10951>.
- [317] O. Mees, A. Eitel, and W. Burgard, "Choosing smartly: Adaptive multimodal fusion for object detection in changing environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 151–156, doi: 10.1109/IROS.2016.7759048.
- [318] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

- [319] D. Feng *et al.*, “Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–20, 2020, doi: 10.1109/TITS.2020.2972974.
- [320] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders, “Selective Search for Object Recognition,” *Int. J. Comput. Vision*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.
- [321] [E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, “A Survey on 3D Object Detection Methods for Autonomous Driving Applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019, doi: 10.1109/TITS.2019.2892405.
- [322] J. Schlosser, C. K. Chow, and Z. Kira, “Fusing LIDAR and images for pedestrian detection using convolutional neural networks,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2198–2205, doi: 10.1109/ICRA.2016.7487370.
- [323] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6526–6534, 2017.
- [324] X. F. Han, J. S. Jin, M. J. Wang, W. Jiang, L. Gao, and L. Xiao, “A review of algorithms for filtering the 3D point cloud,” *Signal Process. Image Commun.*, vol. 57, no. November 2016, pp. 103–112, 2017.
- [325] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 77–85, 2017.
- [326] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, “Joint 3D Proposal Generation and Object Detection from View Aggregation,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 5750–5757, 2018.
- [327] Y. Zhou and O. Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4490–4499, 2018.
- [328] H. You, R. Ji, Y. Feng, and Y. Gao, “PVNet: A joint convolutional network of point cloud and multi-view for 3D shape recognition,” *MM 2018 - Proc. 2018 ACM Multimed. Conf.*, pp. 1310–1318, 2018.
- [329] D. Xu, D. Anguelov, and A. Jain, “PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 244–253, 2018.
- [330] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, “A Review of Deep Learning-Based Semantic Segmentation for Point Cloud,” *IEEE Access*, vol. 7, pp. 179118–179133, 2019.
- [331] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 12689–12697, 2019.
- [332] D. Griffiths and J. Boehm, “A Review on deep learning techniques for 3D sensed data classification,” *Remote Sens.*, vol. 11, no. 12, 2019.

- [333] Y. Xie, J. TIAN, and X. X. Zhu, "Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation," *IEEE Geosci. Remote Sens. Mag.*, pp. 1–51, 2020.
- [334] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2020. S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough and A. Mouzakitis, "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications," in *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829-846, April 2018.
- [335] João Pinto Neto, Lucas Gomes, Fernando Ortiz, Thales Almeida, Miguel Elias M. Campista, et al. An Accurate Cooperative Positioning System for Vehicular Safety Applications. *Computers and Electrical Engineering*, Elsevier, 2019. hal-02364355.
- [336] Mariam Elazab, Aboelmagd Noureldin, Hossam S. Hassanein, "Integrated cooperative localization for Vehicular networks with partial GPS access in Urban Canyons, *Vehicular Communications*", Volume 9, 2017, Pages 242-253, ISSN 2214-2096, <https://doi.org/10.1016/j.vehcom.2016.11.011>.
- [337] G. Hoang, B. Denis, J. Härrri and D. T. M. Slock, "Robust data fusion for cooperative vehicular localization in tunnels," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 1372-1377.
- [338] N. Alam and A. G. Dempster, "Cooperative Positioning for Vehicular Networks: Facts and Future," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1708-1717, Dec. 2013.
- [339] H. Wymeersch, J. Lien and M. Z. Win, "Cooperative Localization in Wireless Networks," in *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427-450, Feb. 2009.
- [340] R. M. Buehrer, H. Wymeersch and R. M. Vaghefi, "Collaborative Sensor Network Localization: Algorithms and Practical Issues," in *Proceedings of the IEEE*, vol. 106, no. 6, pp. 1089-1114, June 2018.
- [341] Anusna Chakraborty, Sohumi Misra, Rajnikant Sharma, Kevin Brink, and Clark Taylor. (2019), "Cooperative Localization: Challenges and Future Directions". 10.1201/9780429507229-24.
- [342] H. Kim, S. H. Lee and S. Kim, "Cooperative localization with distributed ADMM over 5G-based VANETs," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1-5.
- [343] H. Li and F. Nashashibi, "Cooperative Multi-Vehicle Localization Using Split Covariance Intersection Filter," in *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 33-44, Summer 2013.
- [344] F. Bounini, D. Gingras, H. Pollart and D. Gruyer, "Real time cooperative localization for autonomous vehicles," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, 2016, pp. 1186-1191.
- [345] G. Soatti, M. Nicoli, N. Garcia, B. Denis, R. Raulefs and H. Wymeersch, "Implicit Cooperative Positioning in Vehicular Networks," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3964-3980, Dec. 2018.

- [346] M. Rohani, D. Gingras, V. Vigneron and D. Gruyer, "A New Decentralized Bayesian Approach for Cooperative Vehicle Localization Based on Fusion of GPS and VANET Based Inter-Vehicle Distance Measurement," in IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 2, pp. 85-95, Summer 2015.
- [347] J. Liu, B. Cai and J. Wang, "Cooperative Localization of Connected Vehicles: Integrating GNSS With DSRC Using a Robust Cubature Kalman Filter," in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 8, pp. 2111-2125, Aug. 2017.
- [348] J. Xiong, J. W. Cheong, Z. Xiong, A. G. Dempster, S. Tian and R. Wang, "Hybrid Cooperative Positioning for Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 69, no. 1, pp. 714-727, Jan. 2020.
- [349] M. L. Psiaki and T. E. Humphreys, "GNSS Spoofing and Detection," in Proceedings of the IEEE, vol. 104, no. 6, pp. 1258-1270, June 2016.
- [350] R. T. Ioannides, T. Pany and G. Gibbons, "Known Vulnerabilities of Global Navigation Satellite Systems, Status, and Potential Mitigation Techniques," in Proceedings of the IEEE, vol. 104, no. 6, pp. 1174-1194, June 2016.
- [351] Ali Jafarnia-Jahromi, Ali Broumandan, J. Nielsen, and Gérard Lachapelle. (2012). "GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques," International Journal of Navigation and Observation. 2012. 10.1155/2012/127072.
- [352] Matthew O'Toole, David Lindell, and Gordon Wetzstein, "Confocal non-line-of-sight imaging based on the light-cone transform", Nature. 555. 10.1038/nature25489, 2018.
- [353] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," in IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 41-53, July 2005.
- [354] M. Berger et al., "State of the Art in Surface Reconstruction from Point Clouds," Proc. Eurographics 2014, Eurographics Stars, pp. 161–185, 2014. 1 K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Computing Res. Repository, vol. abs/1512.03385, 2015. [Online]. Available: <https://arxiv.org/pdf/1512.03385.pdf>
- [355] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Computing Res. Repository, vol. abs/1512.03385, 2015. [Online]. Available: <https://arxiv.org/pdf/1512.03385.pdf>
- [356] Y. Gong, L. Liu, M. Yang, and L. D. Bourdev, "Compressing deep convolutional networks using vector quantization," Computing Res. Repository, vol. abs/1412.6115, 2014. [Online]. Available: <https://arxiv.org/pdf/1412.6115.pdf>
- [357] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," in Proc. British Machine Vision Conf., 2015, pp. 31.1–31.12.

- [358] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large scale distributed deep networks," in Proc. Conf. Neural Information Processing Systems, 2012, pp. 1223–1231.
- [359] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. D. Freitas. "Predicting parameters in deep learning." *Advances in Neural Information Processing Systems*, 26, 2148–2156. [Online]. Available: [http://media.nips.cc/nipsbooks/nipspapers/paper\\_files/nips26/1053.pdf](http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/1053.pdf), 2013.
- [360] T. N. Sainath, B. Kingsbury, V. Sindhvani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, pp. 6655–6659, 2013.
- [361] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition", in Proc. 33rd Int. Conf. on Learning Representations, <https://arxiv.org/pdf/1409.1556.pdf>, 2015.
- [362] H. Li, W. Ouyang, and X. Wang, "Multi-bias non-linear activation in deep neural networks," arXiv Preprint, arXiv:1604.00676, 2016.
- [363] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," arXiv Preprint, arXiv:1603.05201, 2016.
- [364] T. S. Cohen and M. Welling, "Group equivariant convolutional networks," arXiv Preprint, arXiv:1602.07576, 2016.
- [365] S. Zhai, Y. Cheng, and Z. M. Zhang, "Doubly convolutional neural networks," in Proc. Advances Neural Information Processing Systems, pp. 1082–1090, 2016
- [366] C. Bucilua, R. Caruana, and A. Niculescu-Mizil. Model compression. Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, pp. 535–541. [Online]. Available: <http://doi.acm.org/10.1145/1150402.1150464>, 2006.
- [367] J. Ba and R. Caruana, "Do deep nets really need to be deep?" *Adv. Neural Inform. Process. Syst.*, vol. 27, pp. 2654–2662, 2014.
- [368] Endsley, M.R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1),32–64.
- [369] Dominguez, C., Vidulich, M., Vogel, E. & McMillan, G. (1994). Situation awareness: Papers and annotated bibliography. Armstrong Laboratory, Human System Center, ref. AL/CF-TR-1994-0085.
- [370] Doris Aschenbrenner, Nicolas Maltry, Johannes Kimmel, Michael Albert, Julian Scharnagl, Klaus Schilling, ARTab - using Virtual and Augmented Reality Methods for an improved Situation Awareness for Telemaintenance\*, *IFAC-PapersOnLine*, Volume 49, Issue 30, 2016, Pages 204-209, ISSN 2405-8963

- [371] Ruano S, Cuevas C, Gallego G, García N. Augmented Reality Tool for the Situational Awareness Improvement of UAV Operators. *Sensors* (Basel). 2017;17(2):297. Published 2017 Feb 6. doi:10.3390/s17020297
- [372] Stephan Lukosch, Heide Lukosch, Dragoş Datcu, and Marina Cidota. 2015. Providing Information on the Spot: Using Augmented Reality for Situational Awareness in the Security Domain. *Comput. Supported Coop. Work* 24, 6 (December 2015), 613–664. DOI:<https://doi.org/10.1007/s10606-015-9235-4>
- [373] Elaine M. Raybourn, Ray Trechter, *Applying Model-based Situational Awareness and Augmented Reality to Next-Generation Physical Security Systems*, 2018 , *Cyber-Physical Systems security*, Springer
- [374] Chenxi Ding, Yan Mao, Wuhong Wang, Martin Baumann, *Driving Situation Awareness in Transport Operations*, 2013, Atlantis Press, Paris, 37—56
- [375] B. Park, C. Yoon, J. Lee and K. Kim, "Augmented reality based on driving situation awareness in vehicle," 2015 17th International Conference on Advanced Communication Technology (ICACT), Seoul, 2015, pp. 593-595, doi: 10.1109/ICACT.2015.7224865.
- [376] Lotfi Abdi, Faten Ben Abdallah, Aref Meddeb, In-Vehicle Augmented Reality Traffic Information System: A New Type of Communication Between Driver and Vehicle, *Procedia Computer Science*, Volume 73, 2015, 242-249
- [377] <https://www.byteant.com/blog/how-to-pick-the-best-tool-for-vr-development/>
- [378] <https://www.infoq.com/articles/augmented-reality-best-skds/>
- [379] Marti A. Hearst. 1998. Support Vector Machines. *IEEE Intelligent Systems* 13, 4 (July 1998), 18–28. DOI:<https://doi.org/10.1109/5254.708428>
- [380] Xin Yan and Xiao Gang Su. 2009. *Linear Regression Analysis: Theory and Computing*. World Scientific Publishing Co., Inc., USA.
- [381] Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society (Series B)*, 58, 267-288.
- [382] Arthur E. Hoerl and Robert W. Kennard. 2000. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 42, 1 (Feb. 2000), 80–86. DOI:<https://doi.org/10.2307/1271436>
- [383] Zou, Hui and Hastie, Trevor, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2005, doi:10.1111/j.1467-9868.2005.00503.x
- [384] Mucherino, Antonio and Papajorgji, Petraq J. and Pardalos, Panos M., *k-Nearest Neighbor Classification*, 2009, Springer New York, 10.1007/978-0-387-88615-2\_4
- [385] Robbins, Herbert; Monro, Sutton. A Stochastic Approximation Method. *Ann. Math. Statist.* 22 (1951), no. 3, 400--407. doi:10.1214/aoms/1177729586. <https://projecteuclid.org/euclid.aoms/1177729586>

- [386] Marie Chau, Huashuai Qu, Michael C. Fu, and Ilya O. Ryzhov. 2013. An empirical sensitivity analysis of the Kiefer-Wolfowitz algorithm and its variants. In Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World (WSC '13). IEEE Press, 945–956.
- [387] Nehaya Sultan, Ayman Khedr, Amira Idrees and Sherif Kholeif, 2017. Data Mining Approach for Detecting Key Performance Indicators. Journal of Artificial Intelligence, 10: 59-65.
- [388] Marcus Thorstrom 2017. Applying machine learning to key performance indicators. Master's thesis in Software Engineering. University of Gothenburg, Sweden 2017.
- [389] Daniel Segovia, Miguel Mendoza, Eloy Mendoza, Eduardo González, Augmented Reality as a Tool for Production and Quality Monitoring, Procedia Computer Science, Volume 75, 2015, Pages 291-300, ISSN 1877-0509.
- [390] ONF. "Principles and Practices for Securing Software-Defined Networks", 2015.
- [391] Baseline Security Recommendations for IoT, <https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot>
- [392] RFC 8576 - Internet of Things (IoT) Security: State of the Art and Challenges, <https://tools.ietf.org/html/rfc8576>
- [393] Y.4806 - Security capabilities supporting safety of the Internet of things, <https://www.itu.int/rec/T-REC-Y.4806/en>
- [394] CONCORDIA EU project, "D4.1 1st year report on cybersecurity threats", <https://www.concordia-h2020.eu/deliverables/>
- [395] <https://www.networkworld.com/article/3223952/internet-of-things/5-reasons-why-device-makers-cannot-secure-the-iot-platform.html>
- [396] TNO report <http://publications.tno.nl/publication/34634724/5RwNLq/TNO-2019-iot.pdf>
- [397] Principles of IoT Security [https://www.owasp.org/index.php/Principles\\_of\\_IoT\\_Security](https://www.owasp.org/index.php/Principles_of_IoT_Security)
- [398] Internet of Things (IoT) Security: State of the Art and Challenges <https://datatracker.ietf.org/doc/rfc8576/>
- [399] CSA IoT Security Controls Framework <https://cloudsecurityalliance.org/artifacts/iot-security-controls-framework/>
- [400] Future-proofing the Connected World:13 Steps to Developing Secure IoT Products <https://downloads.cloudsecurityalliance.org/assets/research/internet-of-things/future-proofing-the-connected-world.pdf>
- [401] Bakirtzis, Georgios, Bryan T. Carter, Carl R. Elks, and Cody H. Fleming. "A model-based approach to security analysis for cyber-physical systems." In *2018 Annual IEEE International Systems conference (SysCon)*, pp. 1-8. IEEE, 2018.
- [402] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-based evaluation:from dependability to security," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 48–65, 2004
- [403] B. Chen, Z. Kalbarczyk, D. M. Nicol, W. H. Sanders, R. Tan, W. G. Temple, N. O. Tippenhauer, A. H. Vu, and D. K. Y. Yau, "Go with the flow: Toward workflow-oriented security assessment," in *Proceedings of New Security Paradigm Workshop (NSPW)*, 2013.
- [404] H. Kopetz, *Real-time systems: design principles for distributed embedded applications*. Springer Science & Business Media, 2011.

- [405] K. R. Davis, C. M. Davis, S. A. Zonouz, R. B. Bobba, R. Berthier, L. Garcia, and P. W. Sauer, "A cyber-physical modelling and assessment framework for power grid infrastructures," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2464–2475, 2015.
- [406] Gamma, E., R. Helm, R. Jonson and J. Vlissides (1996). *Design Patterns : Elements of Reusable Object Oriented Software*. Buam, Holland, Addison Wesley.
- [407] Schumacher, M., E. Fernandez-Buglioni, D. Hybertson, F. Buschmann and P. Sommerlad (2005). *Security Patterns: Integrating Security and Systems Engineering*. West Sussex, England, John Wiley & Sons.
- [408] Nhlabatsi, Armstrong, Arosha Bandara, Shinpei Hayashi, Charles Haley, Jan Jurjens, Haruhiko Kaiya, Atsuto Kubo et al. "Security patterns: Comparing modelling approaches." In *Software engineering for secure systems: Industrial and research perspectives*, pp. 75-111. IGI Global, 2011
- [409] Yoshioka, N., H. Washizaki and K. Maruyama (2008). "A survey on security patterns." *Progress in Informatics*(5): 35-47.
- [410] Mayer, N., P. Heymans and R. Matulevicius (2007). *Design of a Modelling Language for Information System Security Risk Management*. 1st International Conference on Research Challenges in Information Science (RCIS), Ouarzazate, Morocco.
- [411] Cabot, J. and N. Zannone (2008). *Towards an Integrated Framework for Model-driven Security Engineering*. Proceedings of the Workshop on Modelling Security (MODSEC08), Toalose, France.
- [412] Fernández-Medina, E., J. Jurjens, J. Trujillo and S. Jajodia (2009). "Model-Driven Development for secure information systems." *Information and Software Technology* 51(5): 809-814.
- [413] OMG (2009). *Unified Modelling Language (UML)*. 2009.
- [414] OMG (2004). *UML Profile for Patterns Specification*. UML Profile for Enterprise Distributed Object Computing (EDOC) specification. 2009.
- [415] Lodderstedt, T., D. Basin and J. Doser (2002). *SecureUML: A UML-Based Modelling Language for Model-Driven Security*. «UML» 2002: The Unified Modelling Language: 426-441.
- [416] Jurjens, J. (2004). *Secure Systems Development with UML*. Heidelberg, German, Springer-Verlag.
- [417] van Lamsweerde, A. (2004). *Elaborating security requirements by construction of intentional anti-models*. Proceedings of the 26th International Conference on Software Engineering (ICSE).
- [418] Alexander, I. (2002). *Initial industrial experience of misuse cases in trade-off analysis*. Proceedings of IEEE Joint International Conference on Requirements Engineering.
- [419] Alexander, I. (2003). "Misuse cases: use cases with hostile intent." *IEEE Software* 20(1): 58-66.
- [420] Yu, E. S. K. (1997). *Towards modelling and reasoning support for early-phase requirements engineering*. Proceedings of the 3rd IEEE International Symposium on Requirements Engineering
- [421] Castroa, J., M. Kolp and J. Mylopoulos (2002). "Towards requirements-driven information systems engineering: the Tropos project." *Information Systems* 27(6): 365-389.
- [422] Yu, E. S. K. (1997). *Towards modelling and reasoning support for early-phase requirements engineering*. Proceedings of the 3rd IEEE International Symposium on Requirements Engineering.
- [423] Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia and J. Mylopoulos (2004). "Tropos: An Agent-Oriented Software Development Methodology." *Autonomous Agents and Multi-Agent Systems* 8(3): 203-236.



- [424] Mouratidis, H. (2004). A Security Oriented Approach in the Development of Multiagent Systems: Applied to the Management of the Health and Social Care Needs of Older People In England (PhD Thesis), Department of Computer Science, University of Sheffield, Sheffield, UK.
- [425] Mouratidis, H. (2004). A Security Oriented Approach in the Development of Multiagent Systems: Applied to the Management of the Health and Social Care Needs of Older People In England (PhD Thesis), Department of Computer Science, University of Sheffield, Sheffield, UK.
- [426] van Lamsweerde, A. (2004). Elaborating security requirements by construction of intentional anti-models. Proceedings of the 26th International Conference on Software Engineering (ICSE).
- [427] van Lamsweerde, A., R. Darimont and P. Massonet (1995). Goal-directed elaboration of requirements for a meeting scheduler: problems and lessons learnt. Proceedings of the 2nd IEEE International Symposium on Requirements Engineering.
- [428] Darimont, R. and A. v. Lamsweerde (1996). "Formal refinement patterns for goal-driven requirements elaboration." ACM SIGSOFT Software Engineering Notes 21(6): 179-190.
- [429] Manna, Z. and A. Pnueli (1992). The Temporal Logic of Reactive and Concurrent Systems, Springer Verlag.
- [430] Jackson, M. (1995). Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. London, United Kingdom, Addison-Wesley.
- [431] Jackson, M. (2001). Problem frames : analysing and structuring software development problems. Harlow, Addison-Wesley, 2001.
- [432] Jackson, M. (2001). Problem frames : analysing and structuring software development problems. Harlow, Addison-Wesley, 2001.
- [433] Lin, L., B. Nuseibeh, D. Ince, M. Jackson and J. Moffett (2003). Introducing abuse frames for analysing security requirements. Proceedings of 11th IEEE International Requirements Engineering Conference.
- [434] Liu, L., E. Yu and J. Mylopoulos (2003). Security and privacy requirements analysis within a social setting. 11th IEEE International Requirements Engineering Conference.
- [435] Jackson, M. (1995). Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. London, United Kingdom, Addison-Wesley.
- [436] van Lamsweerde, A. (2004). Elaborating security requirements by construction of intentional anti-models. Proceedings of the 26th International Conference on Software Engineering (ICSE).
- [437] H. Mouratidis, P. Giorgini, and G. Manson, "Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems," in *Advanced Information Systems Engineering*, Berlin, Heidelberg, 2003, pp. 63–78.
- [438] E. Byres and J. Lowe, "The myths and facts behind cyber security risks for industrial control systems," in *Proceedings of the VDE Kongress*, 2004, vol. 116, pp. 213–218
- [439] C. P. Pfleeger and S. L. Pfleeger, "Security in Computing," 2006.
- [440] R. S. Ross, S. W. Katzke, and L. A. Johnson, "Minimum security requirements for federal information and information systems," 2006.
- [441] A. P. Fournaris, L. Pocero Fraile, and O. Koufopavlou, "Exploiting hardware vulnerabilities to attack embedded system devices: a survey of potent microarchitectural attacks," *Electronics*, vol. 6, no. 3, p. 52, 2017.

- [442] H. Lei, B. Chen, K. L. Butler-Purry, and C. Singh, "Security and Reliability Perspectives in Cyber-Physical Smart Grids," in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, May 2018, pp. 42–47, doi: 10.1109/ISGT-Asia.2018.8467794.
- [443] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-Physical Systems Security—A Survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017, doi: 10.1109/JIOT.2017.2703172.
- [444] ENISA, "Threat Taxonomy", 2016. <https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy>
- [445] ENISA, "Threat Landscape Report 2018". <https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2018>
- [446] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, W. Xu, Automated security test generation with formal threat models, *IEEE Trans. Dependable Secure Comput.* 9 (4) (2012) 525–539.
- [447] Rasim Alguliyev, Yadigar Imamverdiyev, Lyudmila Sukhostat, "Cyber-physical systems and their security issues" *Computers in Industry*, Volume 100, 2018, Pages 212-223, ISSN 0166-3615, <https://doi.org/10.1016/j.compind.2018.04.017>.
- [448] Nataliya Shevchenko, Timothy A. Chick, Paige O’Riordan, Thomas Patrick Scanlon, PhD, & Carol Woody, PhD July 2018
- [449] Microsoft, "The STRIDE Threat Model", 2009. [https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878\(v=cs.20\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN)
- [450] <https://cve.mitre.org/>
- [451] OWASP Top 10 Web Application Security Risks. <https://owasp.org/www-project-top-ten/>
- [452] CWE/SANS Top 25 Most Dangerous Software Errors. [https://cwe.mitre.org/top25/archive/2019/2019\\_cwe\\_top25.html](https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html)  
<https://www.sans.org/top25-software-errors>
- [453] <https://www.ibm.com/services/network/sdn-versus-traditional-networking>
- [454] Scott-Hayward, S.; Natarajan, S.; Sezer, S., "A Survey of Security in Software Defined Networks", *IEEE Communications Surveys and Tutorials*, 2016.
- [455] ONF. "Security Foundation Requirements for SDN Controllers", v1.0. 2016
- [456] Krishnan, Prabhakar & Najeem, J.S.. (2019). A review of security, threats and mitigation approaches for SDN architecture. *International Journal of Innovative Technology and Exploring Engineering*. 8. 389-393.
- [457] ANASTACIA Project, "D2.2 Attacks Threats Analysis and Contingency Actions", 2018.
- [458] European Telecommunications Standards Institute (ETSI). (2013). Machine-to-Machine communications (M2M); Functional architecture. Technical Specification, ETSI TS 102 690 V1.1.1 (2011-10). [online] Available at: [http://www.etsi.org/deliver/etsi\\_ts/102600\\_102699/102690/01.01.01\\_60/ts\\_102690v010101p.pdf](http://www.etsi.org/deliver/etsi_ts/102600_102699/102690/01.01.01_60/ts_102690v010101p.pdf) [Accessed 18 Nov. 2016].
- [459] ENISA, "Ad-hoc & sensor networking for M2M Communications – Threat Landscape and Good Practice Guide", February 2017.
- [460] ENISA, "Recommendations for a methodology of the assessment of severity of personal data breaches", 2013. Available at: <https://www.enisa.europa.eu/publications/dbn-severity>

- [461] ENISA THREAT LANDSCAPE FOR 5G NETWORKS, Threat assessment for the fifth generation of mobile telecommunications networks (5G), NOVEMBER 2019
- [462] Cyber-Security Incidents: A Review Cases in Cyber-Physical Systems Mohammed Nasser Al-Mhiqani, Rabiah Ahmad, Warusia Yassin, Aslinda Hassan, Zaheera Zainal Abidin, Nabeel Salih Ali, Karrar Hameed Abdulkareem (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 1, 2018
- [463] Cybersecurity Spotlight – Cyber Threat Actors <https://www.cisecurity.org/spotlight/cybersecurity-spotlight-cyber-threat-actors/>
- [464] An Analysis of Malicious Threat Agents for the Smart Connected Home, Joseph Bugeja, Andreas Jacobsson, and Paul Davidsson, Internet of Things and People Research Center and Department of Computer Science, Malmö University, Malmö Sweden, {joseph.bugeja, andreas.jacobsson, paul.davidsson}@mah.se
- [465] Mouna Rekik, Zied Chtourou, Christophe Gransart. A Cyber-Physical Threat Analysis for Microgrids. SSD 2018, 15th International Multi-Conference on Systems, Signals and Devices, Mar 2018, Hammamet, Tunisia. 6p. hal-01852096
- [466] An Overview of Data Breaches Kevvie Fowler, in Data Breach Preparation and Response, 2016 <https://www.sciencedirect.com/topics/computer-science/hacktivists>
- [467] Under The Hood: Cybercriminals Exploit Automotive Industry's Software Features," study <https://www.sciencedirect.com/science/article/pii/S221420961930261X>
- [468] H. Onishi, Paradigm change of vehicle cyber security, in: 4th International Conference on Cyber Conflict, 2012, pp.381–391.
- [469] H. Onishi, Guidelines for vehicle cybersecurity, <https://docplayer.net/7458872-For-vehicle-cyber-security.html>, 2013.
- [470] M.H. Eiza, Q. Ni, Driving with sharks: rethinking connected vehicles with vehicle cybersecurity, IEEE Veh. Technol. Mag. (2017).
- [471] P. Carsten, M. Yampolskiy, T.R. Andel, J.T. McDonald, In-vehicle networks: attacks, vulnerabilities, and proposed solutions, in: Proceedings of the 10th Annual Cyber and Information Security Research Conference, 2015.
- [472] J. Takahashi, Y. Aragane, T. Miyazawa, H. Fuji, H. Yamashita, K. Hayakawa, S. Ukai, H. Hayakawa, Automotive attacks and countermeasures on LIN-Bus, J. Inf. Process. 25 (2017) 220–228.
- [473] C.-W. Lin, H. Yu, INVITED: cooperation or competition? Coexistence of safety and security in next-generation ethernet-based automotive networks, in: 53rd ACM/EDAC/IEEE Design Automation Conference, 2016.
- [474] Dolev, S.; Krzywiecki, Ł.; Panwar, N.; Segal, M. Dynamic attribute based vehicle authentication. Wirel. Netw. 2017, 23, 1045–1062. [CrossRef]

- [475] Yosef Ashibani, Qusay H. Mahmoud, Cyber physical systems security: Analysis, challenges and solutions, *Computers & Security*, Volume 68, 2017, Pages 81-97, ISSN 0167-4048, Elsevier
- [476] A. Ashok, A. Hahn and M. Govindarasu, A cyber-physical security testbed for smart grid: system architecture and studies, *Proc. Seventh Annu. Work. Cyber Secur. Inf. Intell. Res. ACM*, (2011)
- [477] R. Mahmoud, T. Yousuf, F. Aloul, I. Zualkernan Internet of Things (IoT) security: current status, challenges and prospective measures 10th Int. Conf. Internet Technol. Secur. Trans. IEEE, pp. 336–341 (2015)
- [478] Lu T., Lin J., Zhao L., Li Y., Peng Y. A security architecture in cyber-physical systems: security theories, analysis, simulation and application fields *Int J Secur Appl*, 9 (7) (2015), pp. 1-16
- [479] Zhang B., Ma X., Qin Z. Security architecture on the trusting internet of things *J Electron Sci Technol*, 9 (4) (2011), pp. 364-367
- [480] Peng Y., Lu T., Liu J., Gao Y., Guo X., Xie F. Cyber-physical system risk assessment Ninth Int. Conf. Intell. Inf. Hiding Multimed. Signal Process., pp. 442–447 (2013)
- [481] R. Mahmoud, T. Yousuf, F. Aloul, I. Zualkernan Internet of Things (IoT) security: current status, challenges and prospective measures 10th Int. Conf. Internet Technol. Secur. Trans. IEEE, pp. 336–341 (2015)
- [482] R. Khan, S.U. Khan, R. Zaheer, S. Khan FUTURE Internet: the Internet of Things architecture, possible applications and key challenges, 10th Int. Conf. Front. Inf. Technol., pp. 257–260 (2012)
- [483] Peng Y., Lu T., Liu J., Gao Y., Guo X., Xie F. Cyber-physical system risk assessment Ninth Int. Conf. Intell. Inf. Hiding Multimed. Signal Process., pp. 442–447 (2013)
- [484] S. Ali, R.W. Anwar, O.K. Hussain Cyber security for cyber physical systems: a trust-based approach *J Theor Appl Inf Technol*, 71 (2) (2015), pp. 144-152
- [485] R. Khan, S.U. Khan, R. Zaheer, S. Khan FUTURE Internet: the Internet of Things architecture, possible applications and key challenges 10th Int. Conf. Front. Inf. Technol., pp. 257–260 (2012)
- [486] Zhang B., Ma X., Qin Z. Security architecture on the trusting internet of things *J Electron Sci Technol*, 9 (4) (2011), pp. 364-367
- [487] Peng Y., Lu T., Liu J., Gao Y., Guo X., Xie F. Cyber-physical system risk assessment Ninth Int. Conf. Intell. Inf. Hiding Multimed. Signal Process., pp. 442–447 (2013)
- [488] Fournaris, A.P.; Pocero Fraile, L.; Koufopavlou, O. Exploiting Hardware Vulnerabilities to Attack Embedded System Devices: a Survey of Potent Microarchitectural Attacks. *Electronics* 2017, 6, 52.
- [489] Gou Q., Yan L., Liu Y., Li Y. Construction and strategies in IoT security system *Proc. - IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Soc. Comput. GreenCom-iThings-CPSCom.*, pp. 1129–1132 (2013)
- [490] R. Bhattacharya A comparative study of physical attacks on wireless sensor networks *Int J Res Eng Technol* (2013), pp. 72-74
- [491] S. Ali, R.W. Anwar, O.K. Hussain Cyber security for cyber physical systems: a trust-based approach *J Theor Appl Inf Technol*, 71 (2) (2015), pp. 144-152
- [492] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection", *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10-25, Jan./Feb. 2010
- [493] A. Mosenia and N. K. Jha, "A Comprehensive Study of Security of Internet-of-Things," in *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586-602, 1 Oct.-Dec. 2017, doi: 10.1109/TETC.2016.2606384.
- [494] X. Wang, S. Chellappan, W. Gu, W. Yu and D. Xuan, "Search-based physical attacks in sensor networks", *Proc. IEEE 14th Int. Conf. Comput. Commun. Netw.*, pp. 489-496, 2005

- [495] M.A. Bhabad, P.G. Scholar Internet of things: architecture, security issues and countermeasures *Int J Comput Appl*, 125 (14) (2015)
- [496] Jing Q., A.V. Vasilakos, Wan J. Security of the internet of things: perspectives and challenges *Wirel Netw*, 20 (8) (2014), pp. 2481-2501
- [497] B. Biggio, B. Nelson and P. Laskov, "Poisoning attacks against support vector machines", *Proc. 29th Int. Conf. Machine Learning*, pp. 1807-1814, 2012.
- [498] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-H. Lau, S. Rao, et al., "Stealthy poisoning attacks on PCA-based anomaly detectors", *ACM SIGMETRICS Performance Eval. Rev.*, vol. 37, no. 2, pp. 73-74, 2009.
- [499] Pub, NIST FIPS. "197: Advanced encryption standard (AES)." *Federal information processing standards publication 197*, no. 441 (2001): 0311.
- [500] Banik, Subhadeep, Andrey Bogdanov, and Francesco Regazzoni. "Atomic-AES v 2.0." *IACR Cryptology ePrint Archive 2016* (2016): 1005.
- [501] Bogdanov, Andrey, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. "PRESENT: An ultra-lightweight block cipher." In *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 450-466. Springer, Berlin, Heidelberg, 2007.
- [502] Banik, Subhadeep, Andrey Bogdanov, Takanori Isebe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. "Midori: A block cipher for low energy." In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 411-436. Springer, Berlin, Heidelberg, 2015.
- [503] De Canniere, Christophe, Orr Dunkelman, and Miroslav Knežević. "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers." In *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 272-288. Springer, Berlin, Heidelberg, 2009.
- [504] Borghoff, Julia, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander et al. "PRINCE—a low-latency block cipher for pervasive computing applications." In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 208-225. Springer, Berlin, Heidelberg, 2012.
- [505] <https://csrc.nist.gov/projects/post-quantum-cryptography>
- [506] National Institute of Standards and Technology (NIST), "Secure hash standard (shs)," *Federal Information Processing Standards Publication 180-4*, March 2012.
- [507] Dworkin, Morris J. *SHA-3 standard: Permutation-based hash and extendable-output functions*. No. Federal Inf. Process. Stds.(NIST FIPS)-202. 2015.
- [508] X. Cao, L. Lu, and M. O'Neill, "A compact sha-256 architecture for rfid tags," in *Proceedings of the 22nd IET Irish signals and systems conference, ISSC, Trinity College Dublin*, 2011.
- [509] <https://csrc.nist.gov/Projects/lightweight-cryptography/>
- [510] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 55, 2014.
- [511] A. P. Fournaris, L. Pocero Fraile, and O. Koufopavlou, "Exploiting hardware vulnerabilities to attack embedded system devices: a survey of potent microarchitectural attacks," *Electronics*, vol. 6, no. 3, p. 52, 2017.
- [512] A. P. Fournaris, K. Lampropoulos, and O. Koufopavlou, "Trusted hardware sensors for anomaly detection in critical infrastructure systems," in *Modern Circuits and Systems Technologies (MOCAST), 2018 7th International Conference on*, 2018, pp. 1–4.
- [513] M. Eckhart and A. Ekelhart, "Towards Security-Aware Virtual Environments for Digital Twins," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, New York, NY, USA, 2018, pp. 61–72, doi: 10.1145/3198458.3198464.

- [514] M. T. Khan, D. Serpanos, and H. Shrobe, "A rigorous and efficient run-time security monitor for real-time critical embedded system applications," in Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, 2016, pp. 100–105.
- [515] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.
- [516] P. V. Bro, "A system for detecting network intruders in real-time," in Proc. 7th USENIX Security Symposium, 1998.
- [517] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in ACM SIGCOMM Computer Communication Review, 2005, vol. 35, pp. 217–228.
- [518] C. Watterson and D. Heffernan, "Runtime verification and monitoring of embedded systems," *IET software*, vol. 1, no. 5, pp. 172–179, 2007.
- [519] A. Kane, "Runtime monitoring for safety-critical embedded systems," 2015.
- [520] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.
- [521] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual reviews in control*, vol. 36, no. 2, pp. 220–234, 2012.
- [522] M. T. Khan, D. Serpanos, and H. Shrobe, "ARMET: Behaviour-based secure and resilient industrial control systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 129–143, 2018.
- [523] M. Blum and H. Wasserman, "Software reliability via run-time result-checking," in *Journal of the ACM*, 1994.
- [524] M. Barnett and W. Schulte, "Runtime verification of. net contracts," *Journal of Systems and Software*, vol. 65, no. 3, pp. 199–208, 2003.
- [525] E. Börger and R. Stärk, *Abstract state machines: a method for high-level system design and analysis*. Springer Science & Business Media, 2012.
- [526] M. Chupilko and A. Kamkin, "Runtime verification based on executable models: On-the-fly matching of timed traces," *arXiv preprint arXiv:1303.1010*, 2013.
- [527] S. Gao, S. Kong, and E. M. Clarke, "dReal: An SMT solver for nonlinear theories over the reals," in *International Conference on Automated Deduction*, 2013, pp. 208–214.
- [528] G. Rigatos, *Intelligent renewable energy systems: modelling and control*. Springer, 2016.
- [529] G. Rigatos, *Differential Flatness Approaches to Nonlinear Filtering and Control: Applications to Electromechanical Systems*. Springer, New York, 2015
- [530] Gartner, <https://www.gartner.com/reviews/market/security-information-event-management>
- [531] Gustavo Gonzalez-Granadillo, Susana Gonzalez-Zarzosa and Mario Faiella, "Towards an Enhanced Security Data Analytic Platform", 15th International Joint Conference on Security and Cryptography, July 2018. DOI: 10.5220/0006831104530458
- [532] PCI Security Standards Council, «Requirements and Security Assessment Procedures (v3.2.1),» 2018.
- [533] Joseph Migga Kizza, "Guide to Computer Network Security", 4<sup>th</sup> Edition, 2015, Springer. ISBN 978-3-319-55605-5
- [534] Mo, Y., Garone, E., Casavola, A., Sinopoli, B., 2010. False data injection attacks against state estimation in wireless sensor networks, in: 49th IEEE Conference on Decision and Control (CDC). Presented at the 49th IEEE Conference on Decision and Control (CDC), pp. 5967–5972. <https://doi.org/10.1109/CDC.2010.5718158>
- [535] Kriebel, F., Rehman, S., Hanif, M.A., Khalid, F., Shafique, M., 2018. Robustness for Smart Cyber Physical Systems and Internet-of-Things: From Adaptive Robustness Methods to Reliability and Security for Machine Learning, in: 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI).

- Presented at the 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 581–586. <https://doi.org/10.1109/ISVLSI.2018.00111>
- [536] Mamdouh, M., Elrukhsi, M.A.I., Khattab, A., 2018. Securing the Internet of Things and Wireless Sensor Networks via Machine Learning: A Survey, in: 2018 International Conference on Computer and Applications (ICCA). Presented at the 2018 International Conference on Computer and Applications (ICCA), pp. 215–218. <https://doi.org/10.1109/COMAPP.2018.8460440>
- [537] Khorshed, M.T., Sharma, N.A., Kumar, K., Prasad, M., Ali, A.B.M.S., Xiang, Y., 2015. Integrating Internet-of-Things with the power of Cloud Computing and the intelligence of Big Data analytics — A three layered approach, in: 2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE). Presented at the 2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), pp. 1–8. <https://doi.org/10.1109/APWCCSE.2015.7476124>
- [538] B. I. Rubinstein et al., "ANTIDOTE: Understanding and defending against poisoning of anomaly detectors", Proc. ACM 9th SIGCOMM Conf. Internet Meas., pp. 1-14, 2009.
- [539] B. Biggio, I. Corona, G. Fumera, G. Giacinto and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks" in Multiple Classifier Systems, Berlin, Germany:Springer, pp. 350-359, 2011
- [540] Fournaris, A.P., Sklavos, N., 2014. Secure embedded system hardware design—a flexible security and trust enhanced approach. *Comput. Electr. Eng.* 40, 121–133.
- [541] Kim, T.T., Poor, H.V., 2011. Strategic Protection Against Data Injection Attacks on Power Grids. *IEEE Trans. Smart Grid* 2, 326–333. <https://doi.org/10.1109/TSG.2011.2119336>
- [542] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection", *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10-25, Jan./Feb. 2010.
- [543] A. Nejat, S. M. H. Shekarian and M. S. Zamani, "A study on the efficiency of hardware Trojan detection based on path-delay fingerprinting", *Microprocessors Microsyst.*, vol. 38, no. 3, pp. 246-252, 2014.
- [544] Fournaris, Apostolos P., Lampros Pyrgas, and Paris Kitsos. "An efficient multi-parameter approach for FPGA hardware Trojan detection." *Microprocessors and Microsystems* 71 (2019): 102863.
- [545] K. Hu, A. N. Nowroz, S. Reda and F. Koushanfar, "High-sensitivity hardware Trojan detection using multimodal characterization", Proc. IEEE Des. Autom. Test Eur. Conf. Exhib., pp. 1271-1276, 2013
- [546] W. G. Halfond and A. Orso, "AMNESIA: Analysis and monitoring for NEutralizing SQL-injection attacks", Proc. 20th IEEE/ACM Int. Conf. Automated Softw. Eng., pp. 174-183, 2005.
- [547] S. Son, K. S. McKinley and V. Shmatikov, "Diglossia: Detecting code injection attacks with precision and efficiency", Proc. ACM SIGSAC Conf. Comput. Commun. Secur., pp. 1181-1192, 2013.
- [548] Sundaram A. "An introduction to intrusion detection", ACM Digital Library. <http://dl.acm.org/citation.cfm?id%332161>
- [549] Shostack, A. (2008). Experiences Threat Modelling at Microsoft. MODSEC@ MoDELS, 2008.
- [550] Data Protection Authorities of Greece and Germany, Clara Galan Manso, ENISA, Sławomir Górnaiak, ENISA, "Recommendations for a methodology of the assessment of severity of personal data breaches", 2013.

- [551] ANASTACIA Project, "D2.4 Secure Software Development Guidelines Initial Report", 2018.
- [552] J. Williams, "OWASP Risk Rating Methodology", [https://owasp.org/www-community/OWASP\\_Risk\\_Rating\\_Methodology](https://owasp.org/www-community/OWASP_Risk_Rating_Methodology)
- [553] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition", in Proc. 33rd Int. Conf. on Learning Representations, <https://arxiv.org/pdf/1409.1556.pdf>, 2015. ykgj
- [554] H. Li, W. Ouyang, and X. Wang, "Multi-bias non-linear activation in deep neural networks," *arXiv Preprint, arXiv:1604.00676*, 2016.
- [555] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," *arXiv Preprint, arXiv:1603.05201*, 2016.
- [556] T. S. Cohen and M. Welling, "Group equivariant convolutional networks," *arXiv Preprint, arXiv:1602.07576*, 2016.
- [557] S. Zhai, Y. Cheng, and Z. M. Zhang, "Doubly convolutional neural networks," in *Proc. Advances Neural Information Processing Systems*, pp. 1082–1090, 2016.
- [558] Y. Cheng, D. Wang, P. Zhou and T. Zhang, "Model Compression and Acceleration for Deep Neural Networks: The Principles, Progress, and Challenges," in *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126-136, Jan. 2018, doi: 10.1109/MSP.2017.2765695.