MAVEN

Managing Automated Vehicles Enhances Network

WP3 Vehicle automation

Deliverable nº: 3.1

Detailed concepts for cooperative manoeuvre and trajectory planning and for in-vehicle cooperative environment perception

Dissemination Level: PU

Version	Date	Release	Approval
05	21-12-2018	Reza Dariani (DLR)	Ondrej Pribyl (CTU) Julian Schindler (DLR)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Document Log

Version	Date	Comments	Name and Organisation
01	31-01-2018	Initial version	Reza Dariani
02	16-02-2018	Input from partners	Dominik Matheis (HMETC) Andreas Leich (DLR)
03	21-03-2018	First draft version	DLR and HMETC
04	18-12-2018	Review comments added	Michele Rondinone (HMETC) Thomas Walter(HMETC) Andreas Leich(DLR)
05	21-12-2018	Finalization	Reza Dariani(DLR)

Authors

Beneficiary nº (1 to 5)	Name and Organisation	Country
1	Reza Dariani	Germany
2	Julian Schindler	Germany
3	Andreas Leich	Germany
4	Dominik Matheis	Germany
5	Michele Rondinone	Germany

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Executive summary

This deliverable is the textual description of the cooperative manoeuvre and trajectory planning and in-vehicle cooperative environment perception concepts, which have been designed by DLR and HMETC for MAVEN.

An overview of the vehicle automation for DLR and HMETC on module level is given which represents briefly the main building blocks of the vehicle automation which will be used in MAVEN for simulation as well as for tests with vehicles.

Based on the analysis of the existing current approaches and the state-of-the-art review, an advanced concept for cooperative trajectory planning and cooperative environment perception on modular level is presented for DLR and HMETC. Despite the differences between vehicle automation of DLR and HMETC, both must be able to cooperate with each other and build a platoon in complex urban area. In that event a unique platoon logic concept is presented by DLR which will be used by DLR and HMETC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Table of Contents

1 Introduction 9
2 Vehicle architecture 10
  2.1 DLR 10
  2.2 HMETC 12
3 Vehicle automation function 19
  3.1 DLR 19
  3.2 HMETC 22
4 Platoon Logic Library 24
  4.1 General assumptions 24
  4.2 Message definition 24
  4.3 Platoon member state machines 25
  4.4 Use Case description 30
5 Cooperative perception 51
  5.1 Introduction 51
  5.2 Fusion paradigms 54
  5.3 Fusion System Architecture 55
  5.4 Notes on the Cooperative Perception Message definition 55
6 Conclusion and outlook 60
7 References 61

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

List of Figures

Figure 1: DLR vehicle architecture ... 11
Figure 2: HMETC vehicle architecture ... 12
Figure 3: HMETC data processing approach “sense, act & control” ... 13
Figure 4: HMETC AD_SW building blocks (Perception and Guidance, Navigation and Control) ... 15
Figure 5: HMETC framework including V2X emulation approach ... 17
Figure 6: Guidance, Navigation & Control (GNC) and Perception simulation ... 18
Figure 7: DLR vehicle automation ... 19
Figure 8: Simple kinematic vehicle model used to find initial solution ... 20
Figure 9: Moving horizon ... 21
Figure 10: Calculated trajectory following lane center line ... 22
Figure 11: Vehicle low level controller (PID) ... 23
Figure 12: The four different state machines used for platooning ... 26
Figure 13: Ranges around vehicle 1 in which another vehicle is considered as candidate for a lane change to V1’s lane. As the range is in relation to the tail of the other vehicle, V2 is not a candidate. The truck 3 is a candidate, because its front is in the orange area. ... 31
Figure 14: UC 1, Platoon initialization. Figure shows the initial situation. ... 31
Figure 15: UC 2, joining a platoon. Figure shows the initial situation. ... 34
Figure 16: UC 2, joining two platoons. Figure shows the initial situation. ... 37
Figure 17: UC 2, joining front vehicle to a platoon. Figure shows the initial situation. ... 39
Figure 18: UC 3, travelling in a platoon. Figure shows the situation where a lane change is about to start ... 41
Figure 19: UC 4, leaving a platoon. ... 42
Figure 20: UC 4, leaving a platoon, when vehicle 1b has left ... 42
Figure 21: UC 4, leaving a platoon with obstacle on adjacent lane ... 44
Figure 22: UC 5, platoon break-up. Figure shows the situation where the breaking-up has just started ... 46
Figure 23: UC 5, platoon break-up with a trailer. Figure shows the situation where the breaking-up has just started. ... 47
Figure 24: UC 5, platoon break-up with a trailer. Figure shows the situation when V1c and V1d make room. ... 48
Figure 25: UC 5, platoon break-up with a trailer. Figure shows the situation when V1c and V1d restored their platoon. ... 48
Figure 26: UC 6, platoon termination ... 49
Figure 27: Sensor equipment of automated vehicles ... 51
Figure 28: Laser scan of the DLR vehicle FASCAR and corresponding video image ... 52
Figure 29: Occupancy grid map generated from the laser scan ... 53
Figure 30: LIDAR Scan of the ViewCar2 vehicle in a complex urban scene. The scene contains cars, many static objects (markes as “Unknown Small”) and a Truck. The field of view covers 100m in each direction (Grid size is 25m) ... 54
Figure 31: Low level fusion building blocks ... 56
Figure 32: Object level fusion building blocks ... 57
Figure 33: Track level fusion building blocks ... 58
Figure 34: Object level fusion results of fusing CAM with LIDAR recorded on the HMETC test track in Griesheim. There are two vehicles in front of the ego-vehicle: a non-connected in the left lane and a connected vehicle sending CAM-Messages in the same lane. The red box denotes association between CAM message and LIDAR object in front of ego-vehicle. ... 59

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

List of Tables

Table 1: Interface description of the DLR vehicle 11
Table 2: Interface description of the HMETC vehicle 13
Table 3: Platooning-related content of extended CAM on SCH0 (ECAM) 25
Table 4: Platooning-related content of extended CAM on SCHx (PCAM). Remark: PlannedPath and PlannedLane can be either in HighFreq. or in LowFreq., depending on the requirements of the situation. 25

Abbreviations and definitions

Abbreviation	Definition
ACC	Adaptive Cruise Control
AD_SW	Automated Driving Software
ADAS	Advanced Driver Assistance Systems
AGLOSA	Adaptive Green Light Optimized Speed Advisory
C2X	Car to X (communication)
CAM	Cooperative Awareness Message
CPM	Cooperative Perception Message
DGPS	Differential Global Positioning System
DLR	German Aerospace Center (german)
DMM	Decision Making Module
ECU	Electronic Control Unit
FASCar	Name of DLR automated vehicle
GLOSA	Green Light Optimized Speed Advisory
GNC	Guidance, Navigation, Control
GPS	Global Positioning System
HAD	Highly Automated Driving
HF	High Frequency
HMETC	Hyundai Motor Europe Technical Center
I2V	Infrastructure to Vehicle (communication)
IMU	Inertial Measurement Unit
LAM	Lane Advice Message
LF	Low Frequency
MP	Motion Planner
OCP	Optima Control Problem
ODE	Ordinary Differential Equation
OEM	Original Equipment Manufacturer
PCAM	Platoon-adapted CAM
PM	Platoon Message
PP	Path Planner

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

ROS	Robot Operating System
RTK	Real-Time Kinematic
SOA	Service Oriented Architecture
SPAT	Signal Phase and Time
SQP	Sequential Quadratic Programming
UC	Use Case
UDP	User Datagram Protocol
V2X	Vehicle to X (communication)
VC	Vehicle Control
VSE	Vehicle State Estimator
WP	Work Package

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

1 Introduction

This document contains the conceptual results of the WP3, vehicle automation. It describes detailed concepts of cooperative manoeuvre and trajectory planning designed in this work package, as well as detailed concepts of cooperative environment perception. As part of concepts, detailed system architecture on module level including the definition of internal and external interfaces is presented and explained separately for both work package partners namely DLR and HMETC.
This document consists of the following chapters:

- Chapter 2 describes the general vehicle architecture on module level, for the vehicles used in MAVEN.
- Chapter 3 describes the vehicle automation and the sub-modules of the vehicle automation. The main focus of this chapter is on trajectory planning and vehicle control.
- Chapter 4 presents the concept of platoon logic module which manages the platooning states between MAVEN vehicles.
- Chapter 5 describes the cooperative perception concept of DLR vehicles.
- Chapter 6 concludes the deliverable.

As the concepts design and implementation for DLR and HMETC are done mostly separated, they are described separated as well in this document.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

2 Vehicle architecture

In this chapter, the vehicle architecture on module level is described for DLR and HMETC vehicles.

2.1 DLR

Figure 1 illustrates the architecture of the DLR automated vehicles named FASCarE and ViewCar2. The architecture contains different modules such as sensors, software and interfaces. Some of the main modules are explained briefly.

- Dominion: Dominion is a service-oriented (SOA) middleware which is being used in DLR vehicles and simulators.

- Sensors: Different sensors are used in DLR vehicles to sense the vehicle sorrounding and deliver information about vehicle states. The sensor information is sent via CAN to its interface. The following sensors are used in the FASCar:
    - GPS receiver to provide the vehicle position in global coordinates based on the Real Time Kinematic (RTK).
    - Laser, radar and camera to deliver the perception information.
    - IMU and vehicle basic sensors to deliver information about vehicle's ego motion.

- Virtual environment: This module uses a virtual map and includes obstacle data from communication such as CAM and CPM to generate a virtual environment. This module enables testing the vehicle automation in simulation and augmented reality testing. Information about any virtual object in this module can be also sent to the other road uses via communication.

- Sensor data fusion / perception: This module receives information from different sensors, information from digital map and cooperative sensing/perception information transferred via communication. Then it delivers the fused information to the vehicle automation as well as platoon logic. The vehicle perception information then is filtered and also sent to the other road users via communication.

- C2X communication: This module transfers and receives the required information needed for cooperation such as CAM messages, platoon messages etc.

- Vehicle automation: This module consists of three main sub-modules, tactical decision, trajectory planner and vehicle control.
    - Tactical decision: Tactical decision analyses the road geometry, other road users and information received via communication and define a driving task for trajectory planner.
    - Trajectory planner: based on the driving task defined by tactical decision, it plans an optimal trajectory and delivers the vehicle actuators input.
    - Vehicle control: Vehicle control consists of several feedback and feedforward controllers to guarantee that vehicle follows the planned trajectory.

- Platoon logic: To have a robust and stable platooning, it is necessary to have the same logic in all the platoon members, independent of their vehicle automation approach. Platoon logic consists of four different state machines which are explained more detailed in chapter 4. The Platoon logic receives the ego sorrounding information from "Sensor Data fusion / perception" modules and delivers ego platooning states via the communication module to other road users.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

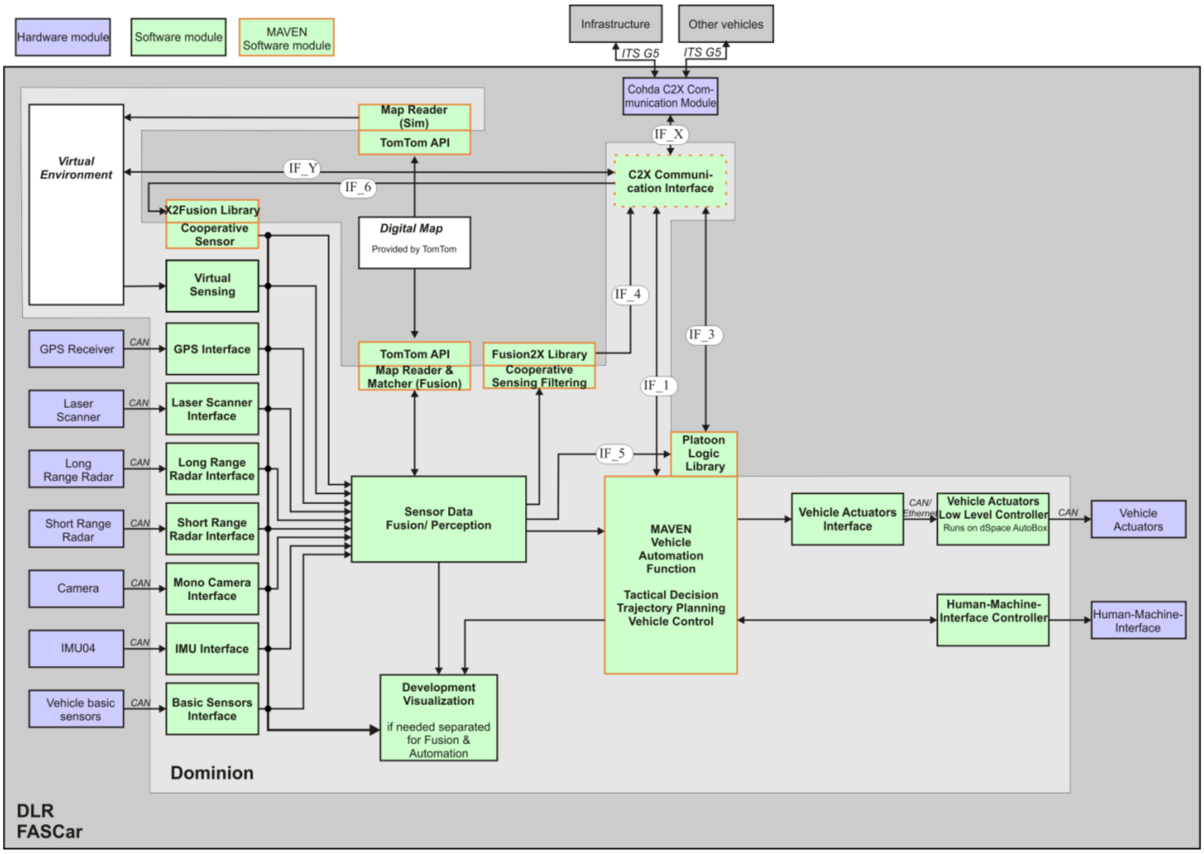

Figure 1: DLR vehicle architecture

As shown in the figure, there are specifically designed interfaces for the C2X Communication Interface and the Platoon Logic Library which are similar in the HMETC and the DLR vehicle, although not all interfaces are present at both vehicles.
The interfaces at DLR are shown in Table 1.

Table 1: Interface description of the DLR vehicle

Interface	Description
IF_1	Bi-Directional interface between the C2X Communication Interface and the MAVEN Vehicle Automation Function. This interface is used to send and receive the CAM data (vehicle position, speed, orientation etc.)
IF_3	Bi-directional interface between the C2X Communication Interface and the Platoon Logic Library. This interface includes all platooning related information from the vehicle itself and the other existing or possible platoon members.
IF_4	With this interface, detected objects are forwarded to the C2X Communication Interface, used to create the Collective Perception Messages (CPM)
IF_5	This interface provides object data for the platoon logic, necessary for the detection of platoon break-ups and the creation of platoons
IF_6	This interface provides received CPMs to the sensor data fusion.
IF_X	This is the general interface between the C2X Software and the Cohda Module
IF_Y	Virtual vehicles also send virtual messages which need to be received by the ego

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

	vehicle. Also the sent messages from the ego vehicle need to be received by the virtual vehicles. This is the purpose of this interface.

2.2 HMETC

Hyundai's main goal in MAVEN is the research of cost effective solutions for automated driving vehicle architecture. This means for example the usage of series or near-series components and the integration of required computing platforms to run the MAVEN control logic. Figure 2 shows Hyundai's Software architecture for the MAVEN project which doesn't differ much from DLR's approach. This is for two reasons:

1. Enabling compatibility to perform MAVEN cases (especially by using the same set of C2X communication messages)
2. Using a common software architecture shared and used by several OEMs, research institutes and universities (common also in the robotic sector) → "sense, act and control"

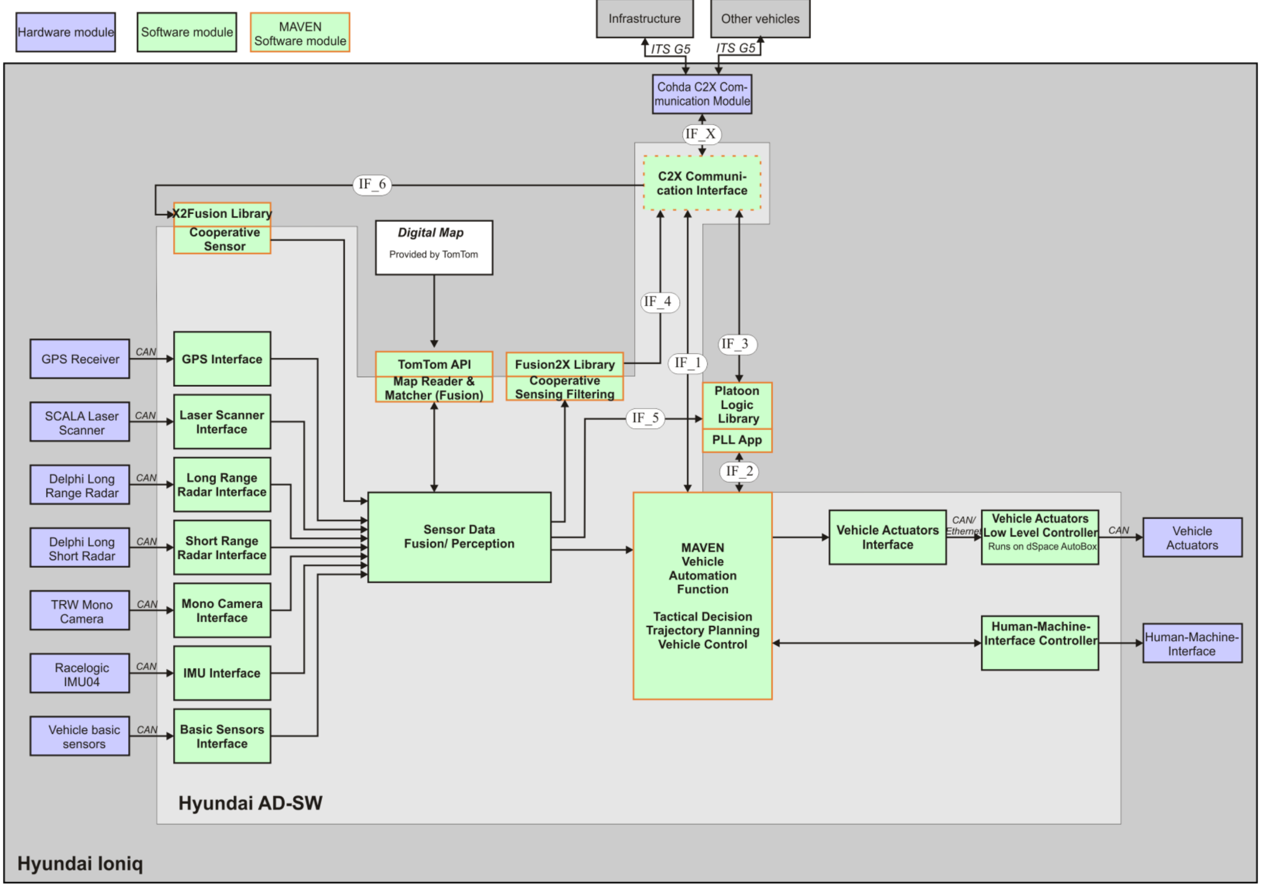

Figure 2: HMETC vehicle architecture

As shown in the figure, there are specifically designed interfaces for the C2X Communication Interface and the Platoon Logic Library which are similar in the HMETC and the DLR vehicle, although not all interfaces are present at both vehicles.
The interfaces at HMETC are shown in Table 2.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Table 2: Interface description of the HMETC vehicle

Interface	Description
IF_1	Bi-directional interface between the C2X Communication Interface and the MAVEN Vehicle Automation Function. This interface is used to send and receive the CAM data (vehicle position, speed, orientation etc.)
IF_2	Bi-directional interface between the platoon logic library application and the vehicle automation function. (At DLR, this interface is not needed as the library is directly linked by the automation function application)
IF_3	Bi-directional interface between the C2X Communication Interface and the Platoon Logic Library. This interface includes all platooning related information from the vehicle itself and the other existing or possible platoon members.
IF_4	With this interface, detected objects are forwarded to the C2X Communication Interface, used to create the Collective Perception Messages (CPM)
IF_5	This interface provides object data for the platoon logic, necessary for the detection of platoon break-ups and the creation of platoons
IF_6	This interface provides received CPMs to the sensor data fusion.
IF_X	This is the general interface between the C2X Software and the Cohda Module

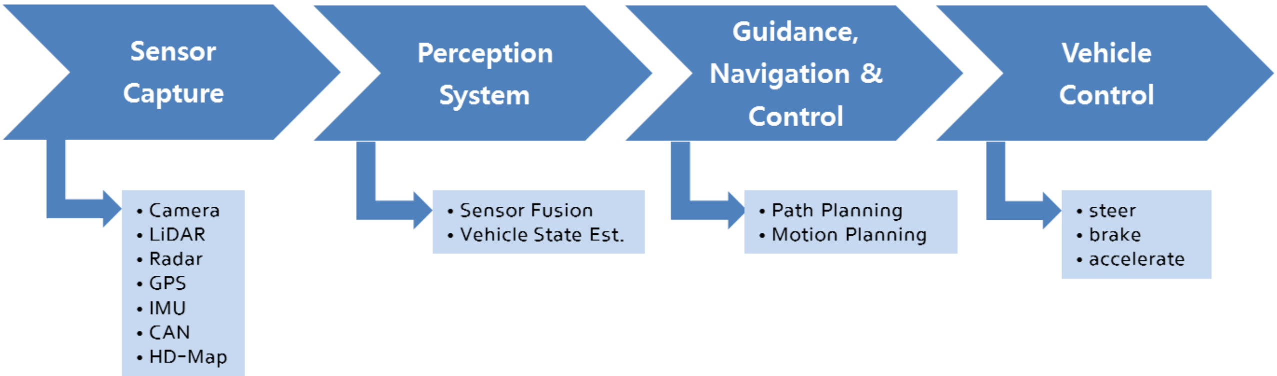

Figure 3: HMETC data processing approach "sense, act & control"

Figure 2 illustrates the vehicle architecture of the HMETC vehicle. The architecture contains different sensors, software's and interfaces. Some of the main modules are explained briefly.

- AD_SW: Hyundai's test vehicle framework running the MAVEN functions is based on the commonly used ROS (short for Robot Operating System), which is taking care of the communication between several so called nodes (sending and/or receiving endpoints/control units/functions), scheduling and maintenance tasks. On top of this base framework control logic and specific MAVEN functions are integrated to fulfil the required MAVEN use cases like manoeuvring control upon road infrastructure advisories, handling of collective perception information or platooning.

- Sensors: the following On-board sensors are used:
    - 1x Ibeo front + 1x Ibeo rear LiDAR,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

- 1X Mobileye front Camera,
- 4 x Aptiv SRR4 corner radars,
- 1 x Aptiv MRR3 front radar

Also, the following Cooperative Sensors are used:

- Cohda MK5 OBU (V2X communication module)

- Sensor Data Fusion / Perception: This module receives the information from different sensors, information from the Highly Automated Driving (HAD) map and also information collected via V2X communication. Then it fuses the information and delivers the fused information to vehicle automation.

- V2X communication interface: This module is used to transmit and receive the required information needed to improve the environmental perception and run platooning algorithms (such as CAM messages and CPM messages), as well to receive information from the infrastructure needed for I2V advices (SPAT/MAP and LAMs).

- Vehicle automation (also called Guidance, Navigation and Control module (GNC) module in the following): This module consists of various sub-modules: Decision Making Module (DMM), Path Planner (PP), Motion Planner (MP), and Vehicle Control (VC). The DMM analyses the intender route, the outputs of the sensor fusion module about the vehicle surrounding and the information received via V2X communication. It uses this information to take decisions regarding possible manoeuvres and threats as inputs to the path planner based on some priority mechanisms. Based on these, the path planner plans a trajectory and provides inputs for the motion planning. Vehicle control consists of several feedback and feedforward controllers to guarantee that vehicle follows the planned motion.

- Platoon logic: The same platoon logic implementation as for the DLR car is used.

2.2.1 *Sensor interfaces, data fusion and planning*

Environmental perception sensors like Radar, LiDAR and front camera are used for object detection and classification as well as for vehicle localization/positioning tasks ((D)GPS receiver and IMU). Sensors used by HMETC are “smart” sensors which mean they offer outputs of pre-processed sensor data (e.g. object detection, classification and tracking). More specifically, dedicated ECUs (Electronic Control Unit) perform a data fusion of data captured respectively by single LiDAR or Radar units, as well as an object tracking (360° around the vehicle). Pre-processed data (classified objects) are fed into the sensor data fusion / perception module to combine objects detections from different sources with vehicle dynamics sensor data (vehicle velocity, acceleration, yaw rate). The data set is completed with V2X inputs (extracted from received CAM and CPM messages), positioning information generated by (D)GPS receivers and an Highly Automated Driving (HAD) map database. Localization of the vehicle on the HAD map can be improved using LiDAR point cloud data (raw data) and TomTom’s RoadDNA database complementing the HAD map. RoadDNA allows a position pattern matching process to improve current position accuracy without necessarily using expensive DGPS receivers (requiring RTK - Real Time Kinematic - service), see MAVEN deliverable D5.2 [1]. Sensor data capturing and data conversion is done by a sensor capture node which distributes filtered and compiled data to the different post-processing nodes as shown in Figure 4.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

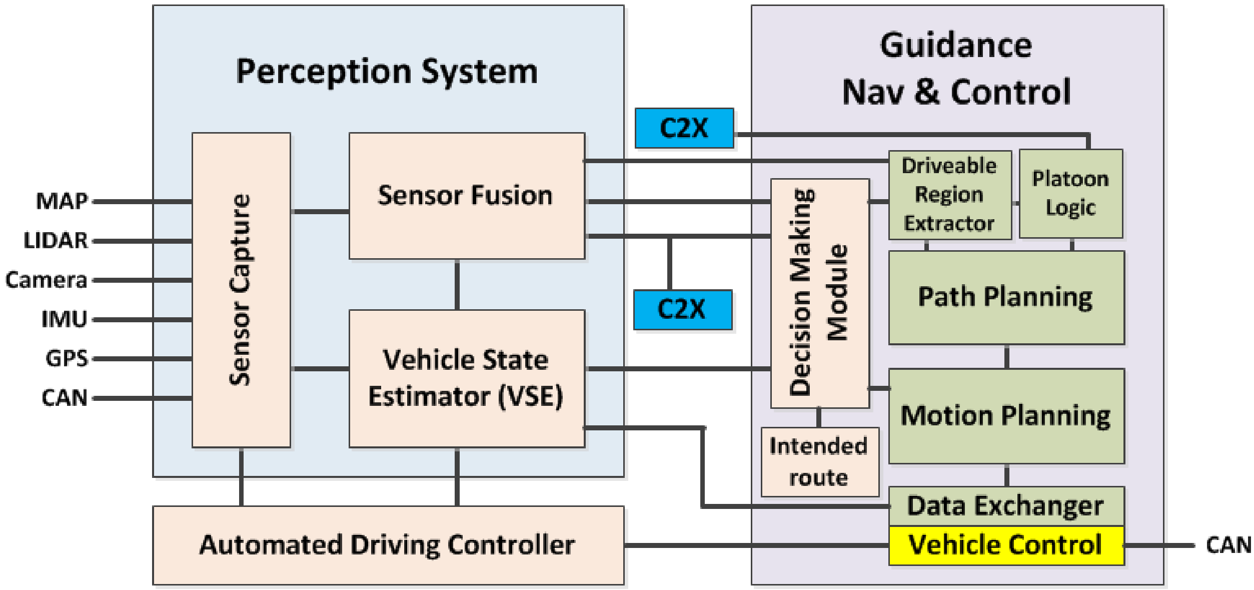

Figure 4: HMETC AD_SW building blocks (Perception and Guidance, Navigation and Control)

The sensor fusion module collects inputs from the individual sensors (including the V2X communication module) and provides a consolidated representation of the environment to the Guidance, Navigation and Control module (GNC). The guidance and control module is devoted to compute the planned trajectory and derive motion objectives to be converted into vehicle control signals. In this block, the Decision Making Module (DMM) can support a threat assessment based on the vehicle route and the obstacles detected in the drivable region. As such, it is used to drive deceleration/stopping, as well as lane change decisions. For this purpose, the Decision Making module takes as inputs the list of detected and tracked objects as well as the list of lanes information from the Sensor Fusion module. This information is crossed with the ego vehicle state (heading, speed, position, etc.) from the Vehicle State Estimator (VSE) and with the intended route as received from the global route planner module. Moreover, the DMM can receive triggers to adapt the vehicle speed or change the lane based on V2X receptions from the infrastructure and has to also consider the presence of traffic signs, speed limits, etc. from HAD maps. Based on all these inputs, the DMM generates two outputs: a so called “feasible manoeuvre” and an “object threat list”. A feasible manoeuvre such as lane change, go straight, stop, keep distance and possible associated speed and or distance values are computed based on the priority of the various inputs like intended route, object list, C2X speed and lane change advice, traffic signs from HAD map data, etc. The object threat list is a set of objects whose position and dynamics currently constitute a threat (e.g. risk of collision) when compared with position and dynamics of the ego vehicle. Feasible manoeuvre and object threat list are given as inputs to the path and motion planner module. Based on them, the path planner module continuously computes a reference goal point and, by exploring all the possible paths to reach it, selects the most suitable one consisting of a set of intermediate waypoints. Finally, the motion planning module takes the waypoints as input and translates them into an objective position, speed and heading. The output of this motion planning is the input for the vehicle automation function (actuator control) enabling the vehicle to reach the target in a safe a comfortable way.
The platooning logic is also interfaced with the path planner and sensor fusion module in order to collect information needed to be exchanged via V2X with other cooperative automated vehicles, and accordingly enable the calculations of the platooning state machine.

2.2.2 *V2X communication*

Main part of the MAVEN project is the V2X communication enabling communication with different (automated) vehicles and the infrastructure. MAVEN V2X communications enable different implementations of automated driving vehicles to form platoons and the collective perception of other road users or vehicles without V2X technology. A detailed description of the V2X messages

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

used to run these use cases can be found in the MAVEN deliverable D5.1 (MAVEN Deliverable 5.1, "V2X communications for infrastructure-assisted automated driving", Feb 2018).
To support V2X communications, the HMETC vehicle supports a V2X communication module, whose interfacing with the general AD_SW framework is handled via a dedicated V2X interface node. The dedicated V2X interface node is in charge of collecting from the other AD_SW modules (sensor fusion and GNC) the needed information to be transmitted by the V2X communication module. On the reception side, the V2X interface node receives the data extracted by the V2X communication modules out of received V2X messages. The interface node then distributes this data to the different AD SW modules that reuse it. For the communication between the V2X communication unit and the V2X interface node, UDP sockets are used. The data exchange is organized in well-defined data structures that have been defined according to the SW modules that are providing and reusing the V2X exchanged information (see MAVEN deliverable D5.1 [2]), as well as the V2X messages employed. As an example, the V2X interface node receives dedicated subset of data separately from the sensor fusion and GNC block, or from the platoon logic modules over specific UDP interfaces. This information is then reused to populate MAVEN CAM messages and their expansions for the platooning algorithms. Similarly, the SF and GNC modules provide data structure to populate CPMs at the V2X communication module over another UDP interface. When the V2X modules receives V2X messages, similar UDP interfaces are used to provide data structures to the SW modules of the vehicle automation that reuse it.
It is here important to mention that the HMETC framework includes an emulation approach used to emulate V2X transmissions and receptions in absence of other cooperative automated vehicles or infrastructure [1]. The AD_SW emulation module can be used to record the sensor fusion module detections of real objects to be disseminated via V2X collective perceptions as well as V2X messages from real infrastructure (e.g. speed or lane change advices recorded at a given traffic light). These recordings are converted into ROS bag files and stored in the format as they would be received over the above mentioned UDP interfaces. The ROS bag files can be then “replayed” within the AD SW logic of the ego-vehicle when performing tests on the test track. This replaying of the bag file is emulating the receptions of V2X messages from the transmitting vehicle or infrastructure (virtual stations in this case).

2.2.3 *Virtual Test Environment*

Before the AD_SW logic is validated and integrated in the real vehicle hardware, tests using a virtual environment are getting executed. For this purpose, HMETC uses a ROS-based simulation framework. The HMETC simulation framework currently supports two major aspects:

1. Guidance, Navigation & Control (GNC) simulation
    - Virtual vehicle test drive on proofing ground → correctness verification of the simulated vehicle trajectory and dynamics can be performed with reference to the landmarks (lane markings) of the proving ground used which are accurately reproduced in the simulation environment.

    - Simulated vehicle and vehicle sensors → correctness verification of the reaction to the inputs for the control logic which runs on the same hardware type used in the test vehicle. Here the behaviour of the simulated vehicle can be checked for all the investigated MAVEN use cases (lane/speed adaptations in response to Infra-advices), ADAS reactions to objects detected by local sensors via collective perception, etc.

2. Perception simulation
    - Replay of recorded sensor data from proofing ground test drives → projection of sensor fusion into monitoring camera view
    - Check of environment → object detection and classification according with sensors models

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

- Platooning scenario generation (including behaviour configuration for other road users) → validation of platoon logic and in particular reaction to V2X communication.

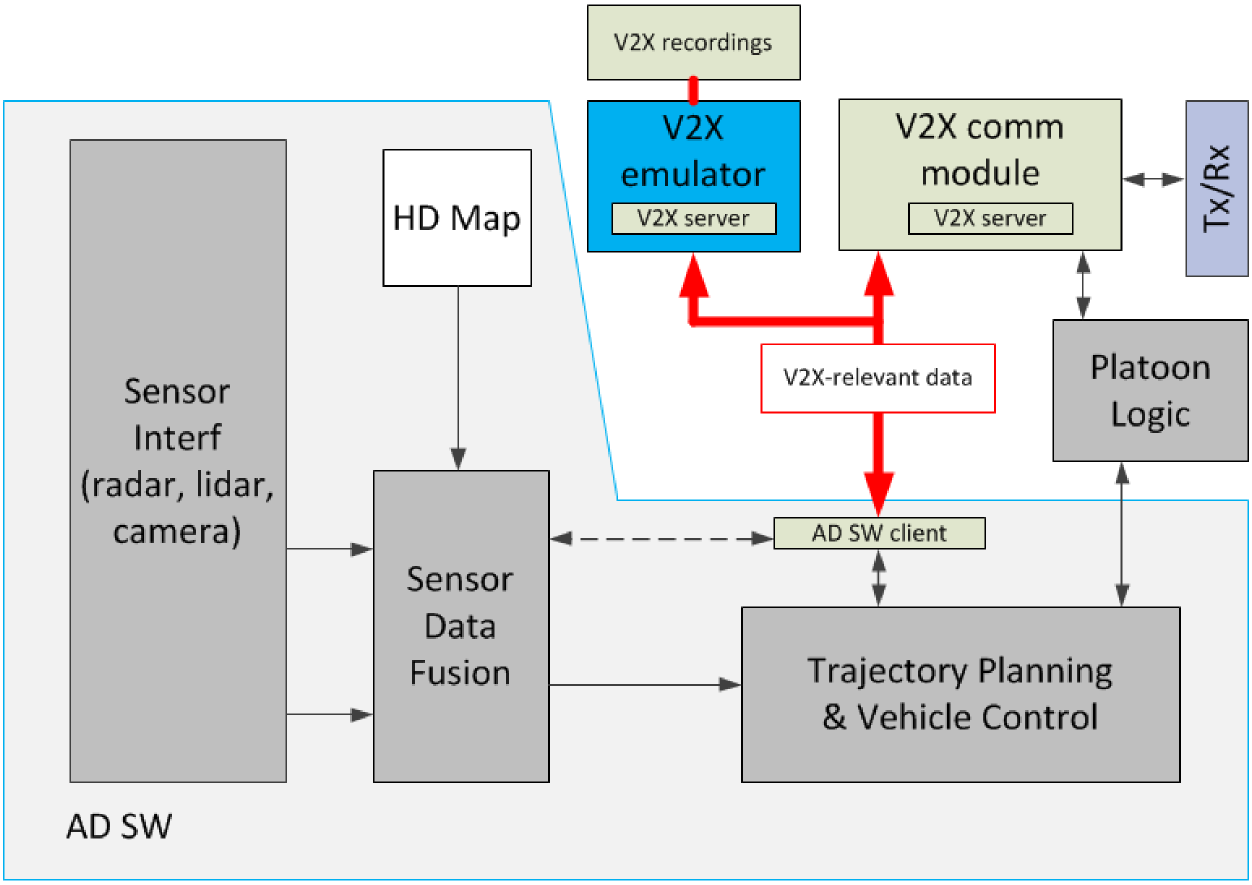

Figure 5: HMETC framework including V2X emulation approach

Figure 6 shows some example screenshots of the simulation framework. The simulator uses a dedicated vehicle dynamics model derived from test drives and specifications to model the test vehicle physical behaviour as good as possible. Simulated sensor data (e.g. vehicles, pedestrians, traffic signs) are used to validate the control logics reaction to other objects' behaviour. With this approach it is possible to reduce the time for proofing ground validation and improve the manoeuvring control logic (for example lane changes, GLOSA behaviour). Especially the low level control signals must be checked before executing the real test vehicle validation to prevent damage of components or unexpected behaviour in presence of other traffic participants.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

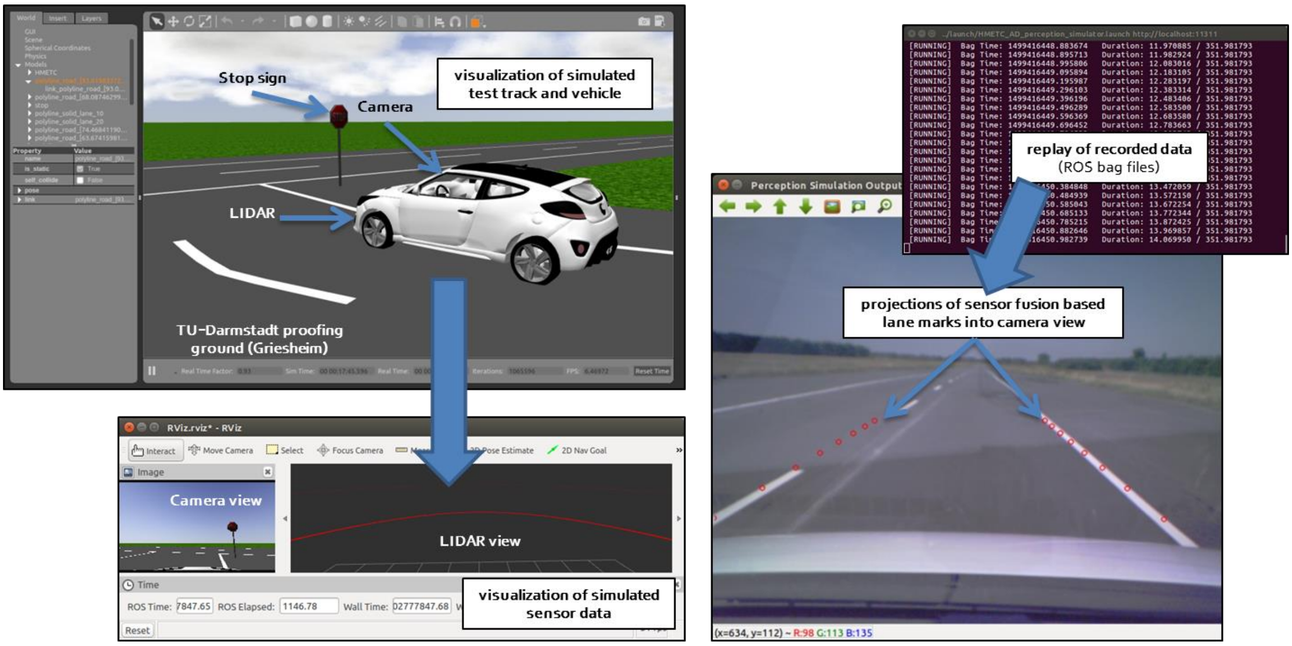

Figure 6: Guidance, Navigation & Control (GNC) and Perception simulation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

3 Vehicle automation function

In this chapter, the vehicle automation function on module level is described for DLR and HMETC vehicles.

3.1 DLR

Figure 7 illustrates the general concept of the vehicle automation used in the MAVEN project by DLR. As it is shown, the vehicle automation is composed of three main modules: Tactical decision, trajectory planner and vehicle control. The tactical decision module defines the driving strategy, the trajectory planner plans a trajectory and the vehicle control guarantees that the vehicle follows the planned trajectory. Each of these modules is described in more detail below.

3.1.1 *Trajectory planner*

The approach used in MAVEN by DLR to plan a trajectory is based on optimal control. In this approach, an optimal control problem OCP is defined in which driver is modeled as penalty function and the vehicle with its dynamic model of motion. Then the OCP is solved with Sequential Quadratic Programming SQP solver. In order to solve OCP an initial solution is required. As the dynamic vehicle model is nonlinear, the quality of the optimal solution and optimization convergence rate depends on the quality of the initial solution. Trajectory planning for automated vehicles unlike the trajectory planning for a robot manipulator has the advantage of having 2D environment which makes the planning problem easier. As the road geometry is well defined and known, planning an initial solution can be done based on the information delivered via digital map. In order to plan an initial trajectory, the environment where the vehicle moves, also called workspace in this context, can be defined as a set of connected graph. There are several methods such as Dijkstra or A* which find the shortest path in this graph. A^* algorithm due to its efficiency and simplicity is used to find the initial trajectory.

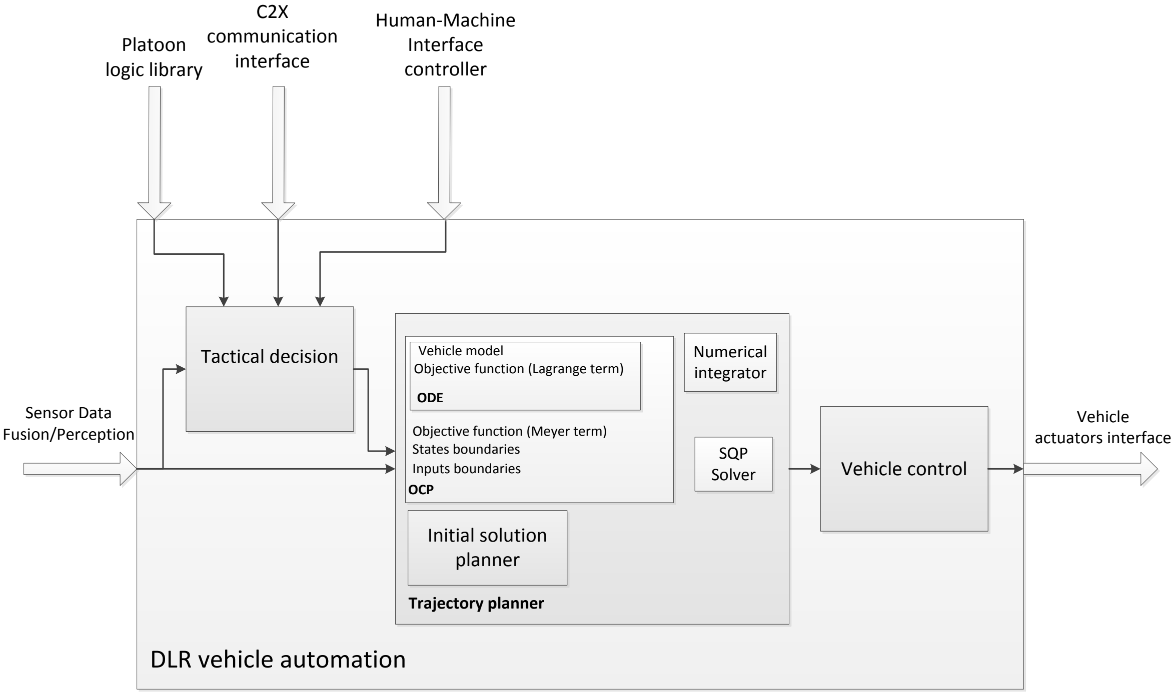

Figure 7: DLR vehicle automation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

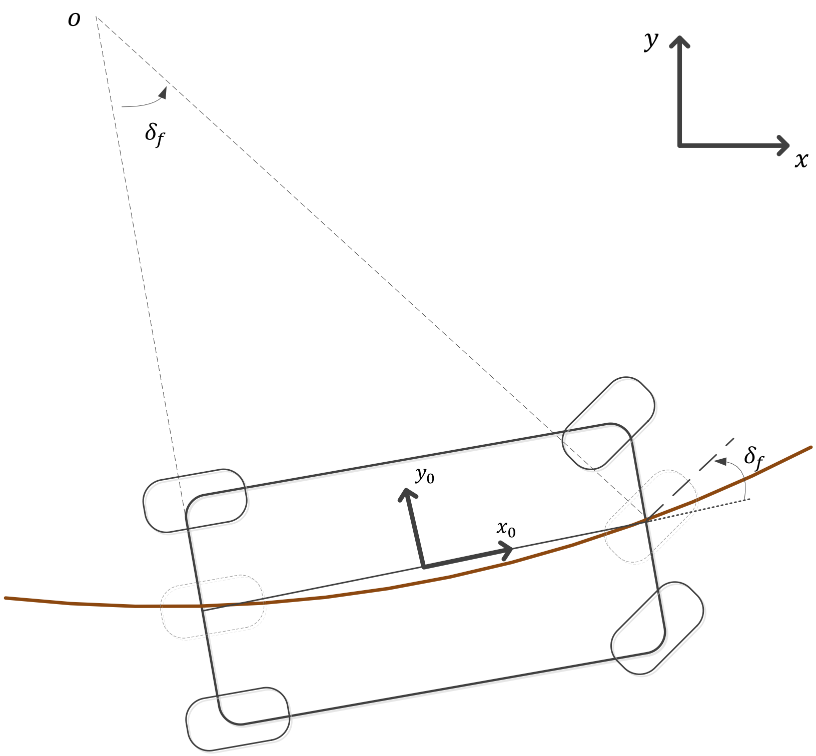

Figure 8: Simple kinematic vehicle model used to find initial solution

Based on the trajectory and a simple kinematic model of the vehicle (see Figure 8), required initial control value, here steering angle and longitudinal acceleration can be calculated. These control values are used as initial solution of the optimal control problem which is explained below.

As explained before, optimal control approach is used to plan a trajectory in DLR vehicles and driver is modeled as penalty function and the vehicle with its dynamic model of motion. Generally an optimization problem can be written in the following form:

$$\min_{\underline{x},\underline{u}} J(\underline{x}(t_f), \underline{u}(t_f)) \tag{1}$$

with dynamic system and nonlinear constraints as

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \tag{2}$$

$$g_l \leq g(\underline{x}, \underline{u}) \leq g_u \tag{3}$$

as well as states and input restrictions as

$$\underline{X_l} \leq \underline{x} \leq \underline{X_u} \tag{4}$$

$$\underline{U_l} \leq \underline{u} \leq \underline{U_u} \tag{5}$$

J referes to the objective function, $\underline{x}$ and $\underline{u}$ refer to states and inputs, g to nonlinear constraints and index l and u to lower and upper value respectively. Equation 1 considers only the states values and inputs values at the final time t_f also named Meyer term. To deal with the objective function which consider the complete optimization horizon, objective function of equation 1, can be written as equation 6 in which integral part, also called Lagrange term, is defined as an extra state inside ODE, ordinary differential equation, of the dynamic system.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

$$J(\underline{x},\underline{u}) = J\left(\underline{x}(t_f),\underline{u}(t_f)\right) + \int_{t_0}^{t_f} f_J\,(\underline{x}(t),\underline{u}(t))dt \qquad (6)$$

Desired driving behavior can be defined inside objective function of the optimal control problem, equation 6, and any boundaries and constraints can be defined in equations 3, 4, 5 or inside Lagrange term of equation 6 as penalty function.
Due to the problem complexity such as high length of driving course, solving the optimal control problem globally requires high computational time. In addition, the driving environment has a dynamic behavior. Therefore, the global optimization problem covering the complete driving task by using "Moving-horizon approach" can be portioned into several local optimal sub-problems which are easier to solve and the problem can be updated by receiving the new information. Figure 9 illustrates an optimal sub problem of τ seconds also named a horizon. After solving this horizon, a part of this horizon, ξ named also increment, is sent to the vehicle control and then the optimal sub problem is updated and a new horizon will be solved (here from t_{n+1} till $t_{n+1} + \tau$).

3.1.2 *Tactical decision*

Define an optimal control problem in a general form which satisfies all driving situations and conditions is impossible. Therefore, based on the forehead situation, out of planning horizon, driving strategy must be chosen and based on chosen strategy *OCP* must be reformulated. As an example maximum velocity is an important parameter which can be taken from the road traffic sign or can be defined based on driving area (Urban, Highway). But traffic network dynamics can also impact driving speed. Therefore a safe and feasible speed based on the current situation must be defined. For example driving with high velocity through a sharp curve is not desirable and also results in discomfort. Further consideration is not limited only to driving velocity but to reformulation of OCP. For example a traffic light with red phase which is far enough to not be considered inside optimization horizon, or an advised velocity from AGLOSA system, or any action needed for platooning scenario , can result in deferent parametrization of OCP . In that event *Tactical decision* module by having information about vehicle states and other traffic participants and road geometry reformulates OCP.

3.1.3 *Vehicle Control*

As the vehicle model used inside the trajectory planner is simplified, it cannot reflect the complete behaviour of the real vehicle, in another hand is not possible to consider all external disturbances and effects inside the trajectory planner due to their complexity and also their calculation time expense. Hence driven trajectory after applying the actuator values does not match with the planned trajectory. Therefore, set of close loop controller is used to minimize the error and difference between calculated trajectory and the driven one.

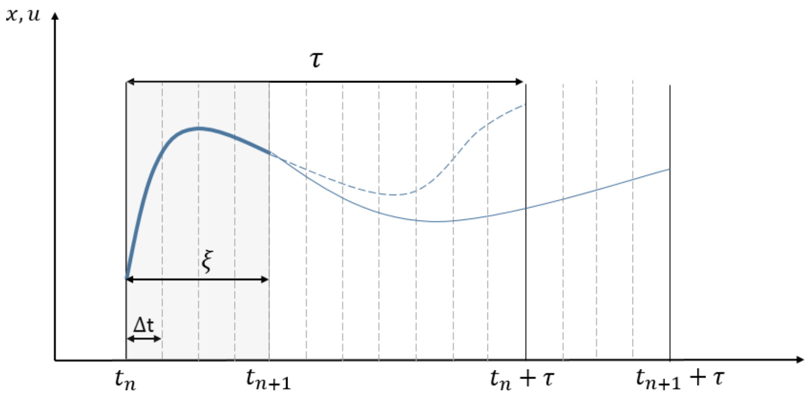

Figure 9: Moving horizon

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

3.2 HMETC

3.2.1 *Path and motion planning*

Referring to Figure 4, based on the outputs of the sensor data fusion, HD maps information, intended route and decision making module outputs, a free space detection and drivable region identification algorithm running at the drivable region extractor will distinguish between driveable and non-driveable areas. Thanks to this algorithm, the drivable region extractor provides a polygon representing the region where the vehicle can drive and hence possible trajectories/paths can be calculated. With this information a trajectory will be calculated by the path planner (PP) module. The path planner will generate an objective point and intermediate waypoints out of the feasible maneuver received by the DMM module (“do a lane change to the right, before the next junction”) after selecting one of several calculated possible paths (optimal path based on cost functions). Main inputs in this whole process are the HD map road attributes (number of lanes, lane width, lane center and boarder points, slope, curvature radius, speed limits, road classes, constraints) and the current ego-vehicle’s state information (position, speed, heading). The path planning logic logic is aiming for stabilizing the vehicle trajectory as close as possible to the target lane center line, which will keep the vehicle in the middle of a road lane (see Figure 10 and MAVEN deliverable D5.2 [1]. As requested by the platooning control algorithm, the planned path is calculated with a time horizon of 6s, where 30 intermediate waypoints are equidistantly separated by 200ms. The result of this path planning (multiple waypoints) is fed into the motion planning module which calculates a “short term” path (or trajectory) for the next couple of meters which consists of a trajectory profile including position, speed and heading to be applied. For this calculation, the Motion Planning module, also takes into account a list of threats received by the DMM module. In fact, the whole process also takes into account other (non-cooperative) road users (for example cars hooking in front of the Ego vehicle or pedestrians walking into the drive path) to prevent accidents or hard braking. For this reason the object perception and prediction function in the Perception module will try to predict the behaviour of other road roads users for the next seconds and feed it as possible threats to the MP module via the decision making module. The uncertainty of these predictions will rise with moving time (as well with higher driving speeds or faster direction changes of the object → pedestrians).
Finally, the short term path calculated by the MP module is translated into lateral and longitudinal control signals which are sent to the vehicle actuators (engine, brake, steering, and gearbox).

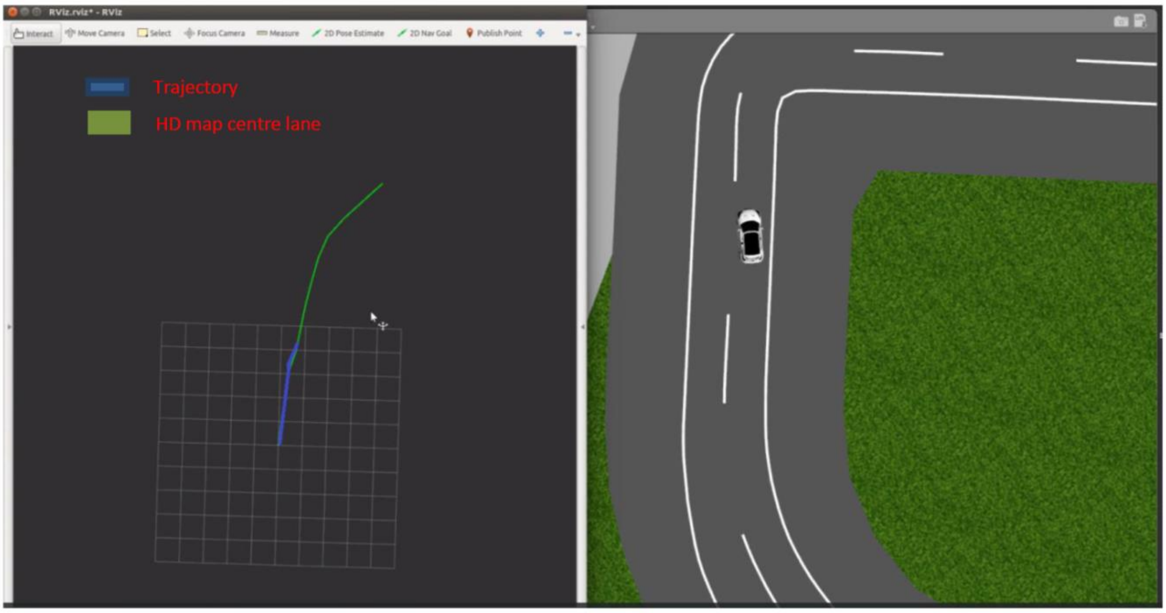

Figure 10: Calculated trajectory following lane center line

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

3.2.2 *Vehicle control*

The last module in the GNC control logic chain is the Trajectory Follower (TF, see Figure 11). The TF implements a control loop of the vehicle's actuators. This Node takes input from the motion planner (trajectory profile in terms of desired heading, position and velocity) for the next intermediate objective waypoint (about a couple of meters). These values are fed into two PID controllers (proportional–integral–derivative control loop) dedicated to lateral and longitudinal control implemented in a dSpace MicroAutobox. Both controllers calculate based on current vehicle state (vehicle state error → offset from the desired state) steering torque & steering angle as well as acceleration values which are sent to the vehicle actuators via CAN messages to reach the desired waypoint. The test driver is always able to intervene either by pressing an emergency stop button or by taking control of the steering wheel and or acceleration / brake pedal.

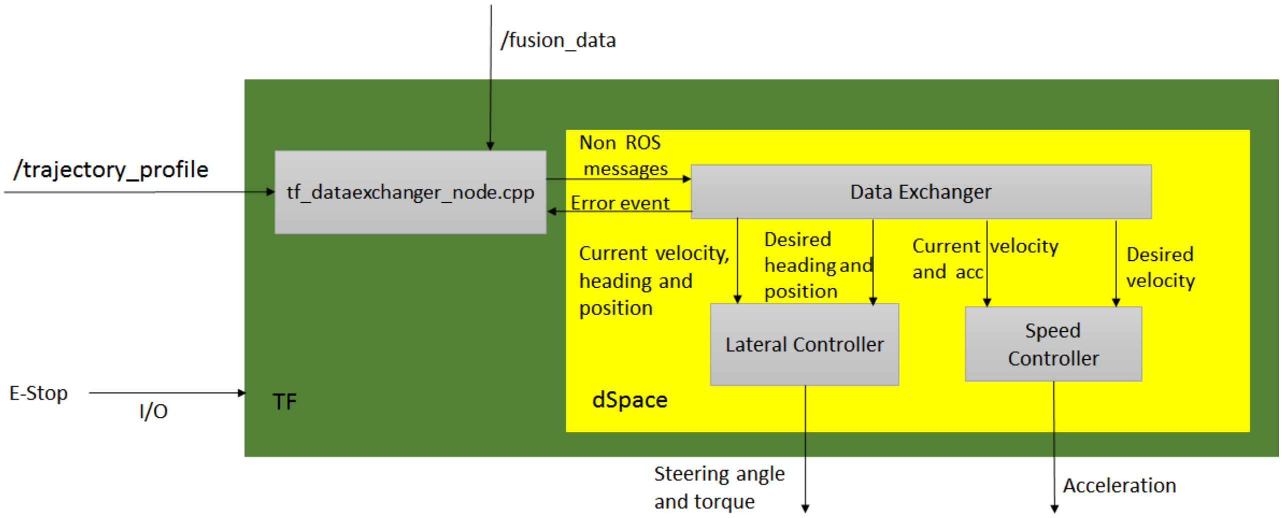

Figure 11: Vehicle low level controller (PID)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

4 Platoon Logic Library

In MAVEN project one of the main focuses is on platooning in urban area which probably will make the transport operation more efficient and reduce the environment. Platooning connected to the intelligent traffic management results in high throughput on intersection. In order to have a flexible and robus platooning behaviour, a platoon logic is proposed which will be applied in DLR and HMETC vehicles. This section is focussing on the communication aspects of the platoon use cases of MAVEN.

4.1 General assumptions

In MAVEN, the general approach for platooning is based on robustness, flexibility and simplicity. The reason for this is that MAVEN targets at urban driving on ring roads where situations are more complex than on highways, where sensors easily come to its limits (e.g. GPS due to buildings and trees), where communication range is also reduced, and where situations require sudden and flexible reactions.

Although MAVEN focuses on hierarchical traffic management, it therefore has been decided that basic platoon functionalities are communicated on a V2V basis without putting the infrastructure in the loop. This results in higher flexibility and simplicity of the resulting communication patterns. On the other hand, the behaviour of the vehicles in a platoon and the decision of vehicles to form or leave a platoon can be influenced by the infrastructure indirectly, e.g. by GLOSA messages, lane change advisories or cooperative sensing.

With regard to platooning, MAVEN targets on the following use cases:

UC1: Platoon initialization
UC2: Joining a platoon
UC3: Traveling in a platoon
UC4: Leaving a platoon
UC5: Platoon break-up
UC6: Platoon termination

Besides the exchanged messages described in section 4.2 it is important that the platoon logic is harmonized in each vehicle. Therefore, section 4.3 describes the currently specified conditions for state changes in each vehicle.
Afterwards, section 5.4 shows how state machines and communication behave in different scenarios.

4.2 Message definition

According to D5.1 [2], two different message types are used. Both are described in the following.

4.2.1 *Extended CAM on SCH0 (ECAM)*

This message is a standard CAM message with platoon-related extensions. It contains information for other vehicles and for the infrastructure.
This message includes the following CAM extensions:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

	Data Field/Element	Description
MAVEN Automated Vehicle Container	*RouteAtIntersection*	Planned route at next intersection (in/out lane)
	IntersectionRoute	Planned route in terms of next intersections to cross
	DesiredSpeedRange	Desired min and max speed for driving in a platoon
	AccelerationCapability	Supported max positive and negative accelerations
	PlatoonId	Id of the platoon that the vehicle is currently in
	PlatoonParticipants	List of following vehicle IDs (tx by platoon leader only when approaching a cooperative intersection)
	desiredPlatoonSpeed	Speed the platoon desires to adopt (txd by platoon leader only when approaching a cooperative intersection)

Table 3: Platooning-related content of extended CAM on SCH0 (ECAM)

In addition, the following standard CAM fields are directly used for the platoon handshake procedure and platoon mode:

- Unique Vehicle ID
- Speed (Actual velocity)
- referencePosition,
- Heading,
- lanePosition,
- drivingDirection,
- longitudinalAcceleration
- vehicleLength
- exteriorLights (used for platoon break-up initiated by non-platooning connected vehicle)

The ECAM is always sent on the normal CAM channel (SCH0) with low frequency.

4.2.2 *Extended CAM on SCHx (Platooning CAM or PCAM)*

This message is a specialized message for platoon coordination. It is sent on a special channel and used by the following platoon members only. Therefore, this is a V2V message. It consists of a high frequency and a low frequency part as described in the following.
The PCAM is sent whenever the message state machine is in any of the “sending PCAM” states.

	Data Field/Element	Description
Automated Vehicle Container HighFreq.	*Heading*	Vehicle heading
	Speed	Vehicle speed
	LongitudinalAcceleration	Vehicle longitudinal acceleration
	LanePosition	Lane the vehicle is currently driving
	PlannedPath	Planned vehicle trajectory in terms of future positions and headings
	PlannedLane	Lane the vehicle plans to drive to
	EmergencyFlag	Indicates that an emergency situation is locally ongoing
Automated Vehicle Container LowFreq.	*PlatoonId*	Id of the Platoon that the vehicle is currently in
	PlatoonFollowers	List of following vehicle IDs
	PlatoonVehicleState	State of the platoon that the vehicle is currently in
	PlatoonFormingState	Forming state of the platoon that the vehicle is currently in
	PlatoonDistanceState	Distance state of the platoon that the vehicle is currently in
	PlannedPath	Planned vehicle trajectory in terms of future positions and headings
	PlannedLane	Lane the vehicle plans to drive to

Table 4: Platooning-related content of extended CAM on SCHx (PCAM). Remark: PlannedPath and PlannedLane can be either in HighFreq. or in LowFreq., depending on the requirements of the situation.

4.3 Platoon member state machines

Each vehicle which is able to form a platoon has an implemented a set of four state machines covering the different possible states for the platooning, as shown in Figure 12. The basis for all actions is done in the main platooning state machine which shows the current platooning state of

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

the vehicle. Besides, there is a platoon forming state machine, which is only active during platooning. Furthermore, there is the message state machine which is responsible for the frequencies in which each part of the platoon related messages is sent. Finally, there is the distance state machine which is responsible for managing the distance to the vehicle ahead or opening up a gap.
The states are described closely in the next sub sections.

4.3.1 *The platooning state machine*

The platooning state machine represents the overall state of the vehicle, so it describes whether the automated vehicle is currently not able to drive in a platoon, has the wish to create or join a platoon, is currently driving in a platoon or currently leaving one.

4.3.1.1 *States*

- *State "not able"*

This is the initial state of the state machine. In this state, the vehicle is not able to form a platoon or participate at one. Therefore, the vehicle has to keep the normal distances to other vehicles when in this mode.

- *State "want to form"*

In this state, the vehicle basically wants to form a platoon. This state is not linked to any situation. It more serves as a state showing that the vehicle basically is interested in platooning and that the system currently has the ability to form a platoon. This ability is linked to current system states and driver choices, e.g. the driver always has the chance to disable platooning in general.

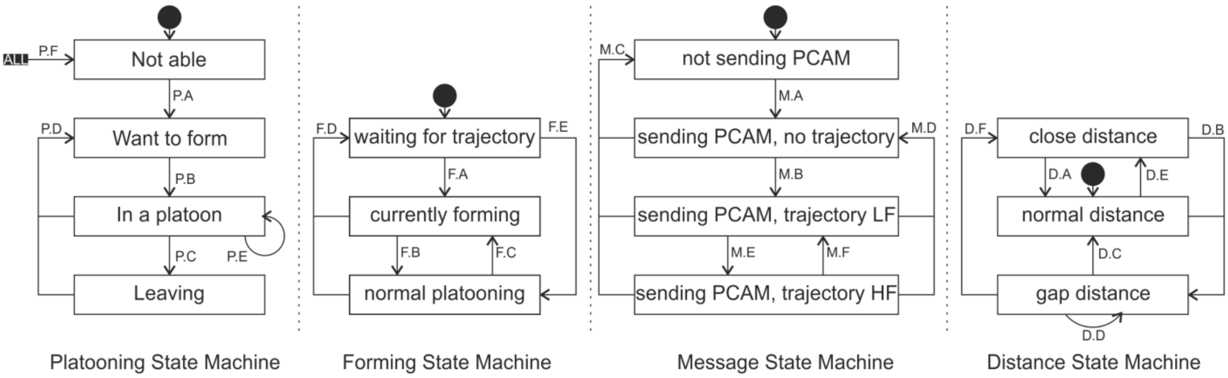

Figure 12: The four different state machines used for platooning

- *State "in a platoon"*

During this state the vehicle is acting as a full platoon member. Being in this state activates the Forming state machine. Besides, the vehicle still is interested in forming a platoon with other following vehicles as long as it is the last vehicle in the platoon.

- *State "leaving"*

This state indicates that the vehicle is currently leaving the platoon. As long as the vehicle is in this state, it is not interested in forming another platoon. The state "leaving" is only reached when single vehicles leave the platoon. In case of a platoon splitting the vehicles stay in state "in a platoon" and only change the platoon leader ID.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

4.3.1.2 *State transitions and their conditions*

- *Condition P.A: “not able”→ “want to form”*

This transition is triggered when the system is switched on or when the driver manually activates the platoon functionality in the driving automation, as long as the system detects that all systems needed for platooning are up and running. This is always independent from the current situation.

- *Condition P.B: “want to form” → “in a platoon”*

This transition is triggered at a following vehicle when another vehicle able to form a platoon is in a given distance $d_{BuildPlatoon}$ ahead, with no vehicle in between, and when the given details of the ECAM message (see further down) like required platoon accelerations, speeds and the driving direction on the upcoming intersection are met. Furthermore, it is a mandatory criterion that PCAM messages from the vehicle ahead are received. In addition, also situational data is input to this transition. This data may include being on the same lane or the estimated costs to change the lane, or the behaviour of other vehicles in the vicinity. The transition is triggered for a leading vehicle when it receives a PCAM with its own vehicle ID as platoon leader ID.

- *Condition P.C: “in a platoon”→ “leaving”*

This transition is triggered when the vehicle automation or the driver decided to leave the existing platoon. This can also happen instantly, e.g. in case of a detected emergency situation.

- *Condition P.D: “in a platoon”/ “leaving”→ “want to form”*

This transition is triggered when the state “in a platoon” is interrupted and a formation of the platoon or driving in a platoon is no longer possible by the situation. This can be due to e.g. evasive manoeuvres, obstacles or other/vulnerable road users between the vehicles, impossible lane changes to the lane of the platoon, or platoon break-up conditions. Furthermore, this condition is met when received PCAMs are not as expected, e.g. when the own state changes to “in a platoon” and the corresponding platoon member is not reflecting own participation.
This transition is also met for a leading vehicle in a platoon, when the direct follower leaves. Furthermore, this transition is triggered when the procedure of leaving (in state “leaving”) ends (e.g. when a lane change out of the platoon lane is finished) and the vehicle is again able to build a new platoon.

- *Condition P.E: “in a platoon” → “in a platoon”*

This transition is triggered whenever the platoon leader ID changes. This can happen

1. when a the platoon enlarges in the front or
2. when vehicles ahead in the platoon are leaving.

The first situation is occurring when a platoon leader gets behind another single automated vehicle able to drive in a platoon or behind another platoon with matching criteria with a distance below dBuildPlatoon. It is a mandatory criterion that PCAM messages from the vehicle ahead are received.
In addition, also situational data is input to this transition. This data may include being on the same lane or the estimated costs to change the lane, or the behaviour of other vehicles in the vicinity.
The second situation will occur when either the platoon leader is leaving or when the platoon splits up ahead of the corresponding vehicle.

- *Condition P.F: all states → “not able”*

This transition is triggered whenever the system is switched off or when a malfunction of the platooning system is detected.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

4.3.2 *The forming state machine*

This state machine is responsible for the platoon forming procedure. It is very much related to the Distance state machine and is used to reflect the state of platoon forming to others.

4.3.2.1 *States*

- *State "waiting for trajectory"*

  This state means that the vehicle is currently waiting to start the forming process, which can start not before any trajectory information has been received in valid PCAMs.

- *State "currently forming"*

  This state means that the vehicle is currently busy reaching the desired range of time headway and distance to the vehicle ahead. Valid trajectory information of the leader is needed for this.

- *State "normal platooning"*

  In this state, the vehicle is acting as normal platoon member. The distances to the vehicle ahead are in the desired range. Valid trajectory information of the leader is needed for this.

4.3.2.2 *State transitions and their conditions*

- *Condition F.A: "waiting for trajectory" → "currently forming"*

  This transition is triggered when a valid PCAM has been received from the vehicle ahead including a valid trajectory.

- *Condition F.B: "currently forming" → "normal platooning"*

  This transition is triggered when the desired time headway and distances have been reached.

- *Condition F.C: "normal platooning" → "currently forming"*

  Whenever the time headway and distances leave the desired range, this transition is triggered.

- *Condition F.D: "normal platooning"/"currently forming" → "waiting for trajectory"*

  Whenever there is no trajectory information received by the vehicle ahead, this transition is triggered. This can happen in normal situations, e.g. for the platoon leader when another vehicle joins the platoon ahead, or when the PCAMs are lost for any reason. This transition is also triggered whenever the platooning state machine changes from "in a platoon" to any other state.

- *Condition F.E: "waiting for trajectory" → "normal platooning"*

  This transition is triggered when the vehicle has not been in a platoon and is directly getting the platoon leader, as forming is only done at followers.

4.3.3 *The message state machine*

The message state machine is responsible for defining the content of the messages sent by the vehicles. This state machine is always active.

4.3.3.1 *States*

- *State "not sending PCAM"*

  In this state the vehicle is not sending any PCAM messages.

- *State "sending PCAM, no trajectory"*

  In this state, PCAM messages are sent, but trajectory data is omitted.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

- *State “sending PCAM, trajectory LF”*

In this state, PCAM messages are sent including trajectory data. The trajectory data is sent in low frequency.

- *State “sending PCAM, trajectory HF”*

In this state, PCAM messages are sent including trajectory data. The trajectory data is sent in high frequency, as it is needed in very high precision, e.g. in case of an emergency action with low planning horizons.

4.3.3.2 *State transitions and their conditions*

- *Condition M.A: “not sending PCAM” → “sending PCAM, no trajectory”*

This transition is triggered when the vehicle is in state “want to form” or “in a platoon” and there is another vehicle able to join a platoon within the distance $d_{SendPCAM}$ behind.

Attention: This should be true: $d_{SendPCAM} > d_{BuildPlatoon}$

- *Condition M.B: “sending PCAM, no trajectory” → “sending PCAM, trajectory LF”*

This transition is triggered when the platooning state machine changes its state to “in a platoon” and the vehicle gets the platoon leader. Furthermore, this transition is triggered when the platooning state machine is in state “in a platoon” and the vehicle gets any followers.

- *Condition M.C: “sending PCAM, no trajectory/trajectory LF/trajectory HF” → “not sending PCAM”*

This transition is triggered when the platooning state machine changes its state to or is in “not able”, or when the vehicle is in state “want to form” and there is no vehicle able to form a platoon within the distance $d_{SendPCAM}$ behind.

- *Condition M.D: “sending PCAM, trajectory LF/trajectory HF” → “sending PCAM, no trajectory”*

This transition is triggered whenever the corresponding vehicle gets the last one in the platoon or when the direct follower, a former member of the own platoon, is either leaving the state “leaving” or when it reaches another lane or a safe distance greater than $d_{BuildPlatoon}$.

- *Condition M.E: “sending PCAM, trajectory LF” → “sending PCAM, trajectory HF”*

This transition is triggered whenever the vehicle has to perform an emergency maneuver requiring a more frequent update of the trajectory information.

- *Condition M.F: “sending PCAM, trajectory HF” → “sending PCAM, trajectory LF”*

This transition is triggered when there is no more need to provide trajectory information in high frequency, e.g. when the emergency situation passed.

4.3.4 *The distance state machine*

This state machine is responsible for managing the distance to the vehicle ahead and for opening up gaps in front in order to allow merging of other vehicles to the own lane. Therefore, this state machine is active independent of the current platoon state machine as it can change the behaviour of single automated vehicles, too.

4.3.4.1 *States*

- *State “normal distance”*

This state represents the normal driving distance and time headway.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

- *State “close distance”*

In this state, the targeted distance/time headway is reduced. Therefore, this state can only happen when the vehicle is in platooning mode “in platoon”.

- *State “gap distance”*

In this state, a gap is opened up ahead of this vehicle.

4.3.4.2 *State transitions and their conditions*

- *Condition D.A: “close distance” → “normal distance”*

The condition is triggered whenever there is the need to enlarge the distance to the vehicle ahead. This happens when the vehicle ahead or the vehicle itself is leaving the platoon or when there is no more trajectory data received from the vehicle ahead.

- *Condition D.B: “close distance”/“normal distance” → “gap distance”*

This transition is triggered whenever there is a vehicle on an adjacent lane

  - With its tail within a given range from own tail to preceding tail (blue area in fig.2) or its front within a given range from own front to preceding front (orange area in fig. 2),
  - Which is driving in the range +/-10% of my own velocity,
  - Which is intending to change the lane, e.g. by a set indicator, by a known lane closure ahead or by other more sophisticated means.

- *Condition D.C: “gap distance” → “normal distance”*

Whenever the conditions from D.B are not met anymore and there is no vehicle directly in my platoon ahead sending trajectory data or the vehicle is not in any platoon, the transition back to “normal distance” is triggered.

- *Condition D.D: “gap distance” → “gap distance”*

Whenever the conditions from D.B are met with another vehicle, this transition is performed.

- *Condition D.E: “normal distance” → “close distance”*

Whenever there is a vehicle directly in my platoon ahead sending trajectory data, the transition back to “close distance” is triggered.

- *Condition D.F: “gap distance” → “close distance”*

Whenever the conditions from D.B are not met anymore and there is a vehicle directly in my platoon ahead sending trajectory data, the transition back to “close distance” is triggered.

4.4 Use Case description

The following shows the MAVEN use cases which include platooning and the related communication. Each use case is split up into scenarios showing the behaviour in different situations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

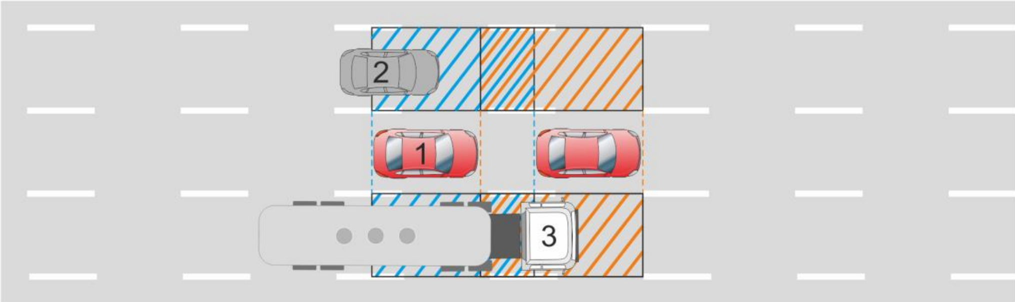

Figure 13: Ranges around vehicle 1 in which another vehicle is considered as candidate for a lane change to V1's lane. As the range is in relation to the tail of the other vehicle, V2 is not a candidate. The truck 3 is a candidate, because its front is in the orange area.

4.4.1 *UC1: Platoon initialization*

This use case describes the forming of a platoon when there is no platoon existing. This use case is divided into three scenarios.

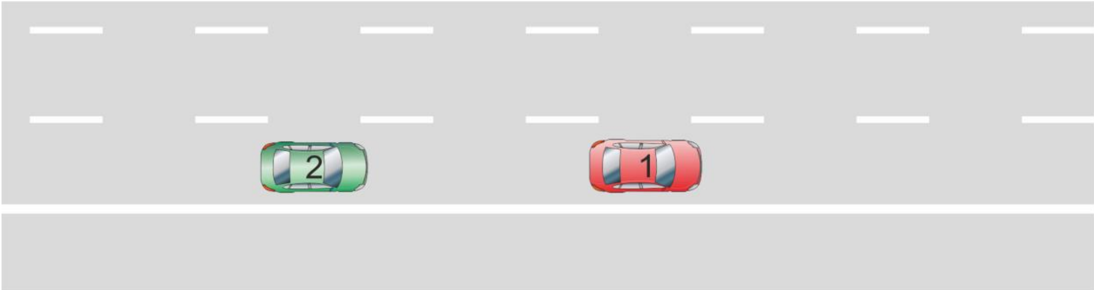

Figure 14: UC 1, Platoon initialization. Figure shows the initial situation.

4.4.1.1 *Scenario UC1a, Normal initialization*

Initial situation:
Two vehicles V1 and V2 are equipped with the needed systems to be able to form a platoon. Both vehicles therefore have the following states:

	Platooning	Forming	Message	Distance	Followers
V1	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None
V2	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None

As result, both vehicles are sending ECAM messages only, with low frequency.

Step 1:
V1 detects that Condition M.A is fulfilled. The message state machine changes its state to "sending PCAM, no trajectory", resulting in sending PCAM without the trajectory data.

Step 2:
V2 receives ECAMs in the background. By receiving the PCAM, V2 meets Condition P.B, switches to state "in a platoon" with the forming state "waiting for trajectory" and the message state "sending PCAM, no trajectory". V2 therefore starts sending PCAMs with the appropriate flags, the platoon leader ID of V1 and no followers.

Step 3:
V1 receives the corresponding PCAM and changes its state to "in a platoon" (Condition P.B), sets the platoon leader ID to its own vehicle ID and adds the vehicle ID of V2 to the list of followers. V1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

changes its message state to “sending PCAM, trajectory LF” and transmits a PCAM with flags, trajectory, leader ID and list of followers accordingly. Furthermore, the forming state is directly changed to “normal platooning” according to Condition F.E.

Step 4:
V2 receives the trajectory information, meets condition F.A, changes the forming state to “currently forming” and starts the platoon formation.

Step 5:
When V2 reaches the desired time headway and distance, condition F.B is met and V2’s forming state changes to “normal platooning”, which is reflected in the PCAM.
Result:
V1 and V2 have formed a platoon with the following states:

	Platooning	Forming	Message	Distance	Followers
V1	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V2
V2	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.1.2 *Scenario UC1b, Unable to initialize, Condition P.B not met*

Initial situation:
Equal to UC1a.

Step 1:
V1 detects that Condition M.A is fulfilled. The message state machine changes its state to “sending PCAM, no trajectory”.

Step 2:
V2 has a not matching Condition P.B, due to lost messages from V1, a situation where a platoon forming is not useful (e.g. different lanes, vehicles in between), etc. Therefore, V2 stays in state “want to form”.

Step 3:
The distance between V1 and V2 raises until Condition M.C is met. V1 stops sending PCAMs.

Result:
No platoon has been formed.

	Platooning	Forming	Message	Distance	Followers
V1	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None
V2	Want to form	Waiting for trajectory	Not sending PCAM	Normal Distance	None

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

4.4.1.3 *Scenario UC1c, Unable to initialize, PCAM lost*

Initial situation:
Equal to UC1a.

Step 1:
V1 detects that Condition M.A is fulfilled. The message state machine changes its state to "sending PCAM, no trajectory", resulting in sending PCAMs.

Step 2:
V2 receives PCAM and ECAM, meets Condition P.B, switches to state "in a platoon" with the forming state "waiting for trajectory" and the message state "sending PCAM, no trajectory". V2 therefore starts sending PCAM appropriately, including the platoon leader ID of V1 and no followers.
Unfortunately, the sent PCAMs are lost.

Step3:
As V1 does not receive a positive reply, it does not change its state. V2 recognizes this, as V1's state stays at "want to form" and the list of followers stays empty.
Now, the states are as follows:

	Platooning	Forming	Message	Distance	Followers
V1	Want to form	Waiting for trajectory	sending PCAM, no trajectory	Normal distance	None
V2	In a platoon	waiting for trajectory	sending PCAM, no trajectory	Normal distance	None

Step 4:
Condition P.D is met as there is no positive feedback from V1, so V2 changes its state back to "want to form".

Step 5:
As long as Condition P.B is met, V2 will try to form a platoon with V1, so the former steps are repeated.

Step 6:
No request has been acknowledged. The distance between V1 and V2 raises until Condition M.C is met. V1 stops sending PCAMs.

Result:
No platoon has been formed.

	Platooning	Forming	Message	Distance	Followers
V1	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None
V2	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

4.4.2 *UC2: Joining a platoon*

This use case describes the joining of an existing platoon. This use case is divided into five scenarios.

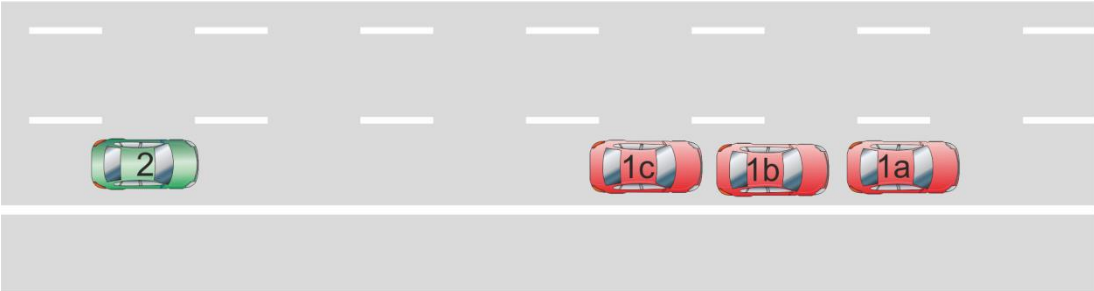

Figure 15: UC 2, joining a platoon. Figure shows the initial situation.

4.4.2.1 *Scenario UC2a, Normal joining*

Initial situation:
The vehicle V2 is approaching an existing platoon with vehicles V1a/b/c. The vehicles therefore have the following states:

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c
V1c	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V2	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None

All the vehicles are sending ECAMs in the background. V1c is sending PCAMs without trajectory, V2 is not sending PCAMs yet. V1a and V1b are sending PCAMs with trajectory V1a/b/c are all reflecting their platoon membership by using the vehicle ID of V1a as platoon leader ID. V1a has 2 followers, V1b has 1 follower, and V1c has no followers.

Step 1:
V2 receives PCAMs without a trajectory from V1c. It meets condition P.B, switches to state “in a platoon” with the forming state “waiting for trajectory” and the message state “sending PCAM, no trajectory”. V2 therefore starts sending PCAM appropriately, including the platoon leader ID of V1a and no followers.

Step 2:
V1c receives the corresponding PCAM, adds the vehicle ID of V2 to the list of followers and changes the message state machine to “sending PCAM, trajectory LF”. Either directly (when PCAM of V2 is received) or with a delay due to hopping from V1c to V1b to V1a, this information is brought to the other platoon members.

Step 3:
V2 receives the trajectory information, meets condition F.A, changes the forming state to “currently forming” and starts the platoon formation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

Step 4:
When V2 reaches the desired time headway and distance, condition F.B is met and V2's forming state changes to "normal platooning", which is reflected in the PCAM.

Result:
V1a/b/c and V2 have formed a platoon with the following states:

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c, V2
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c, V2
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V2
V2	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.2.2 *Scenario UC2b, Unable to join, Condition P.B not met*

Initial situation:
Equal to UC2a.

Step 1:
V1c detects that Condition M.A is fulfilled. The message state machine of V1c changes its state to "sending PCAM, no trajectory", resulting in sending PCAM.

Step 2:
V2 has a not matching Condition P.B, due to lost messages from V1c, a situation where a platoon forming is not useful (e.g. different lanes, vehicles in between), etc. Therefore, V2 stays in state "want to form"

Step 3:
The distance between V1c and V2 raises until Condition M.C is met. V1c stops sending PCAM.

Result:
No platoon joining has been performed.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c
V1c	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V2	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

4.4.2.3 *Scenario UC2c, Unable to join, PCAM lost*

Initial situation:
Equal to UC2a.

Step 1:
V1c detects that Condition M.A is fulfilled. The message state machine of V1c changes its state to "sending PCAM, no trajectory", resulting in sending PCAM.

Step 2:
V2 receives PCAM, meets Condition P.B, switches to state "in a platoon" with the forming state "waiting for trajectory" and the message state "sending PCAM, no trajectory". V2 therefore starts sending PCAM appropriately, including the platoon leader ID of V1a and no followers. Unfortunately, the sent PCAMs are not received by V1c, but by V1a/b.

Step3:
As V1c does not receive a positive reply, it does not add V2 as follower. V2 recognizes this, as V1c's list of followers stays empty. Nevertheless, V1a/b have received the message and added V2 to the list of followers.
Now, the states are as follows:

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c, V2
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c, V2
V1c	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V2	In a platoon	waiting for trajectory	sending PCAM, no trajectory	Normal distance	None

Step 4:
Condition P.D is met for V2 as there is no positive feedback from V1c, so V2 changes its state back to "want to form".

Step 5:
V1a/b receive the status change of V2 to "want to form". Therefore, V2 is considered as a vehicle which left the platoon. V1a/b remove V2 from the list of followers.

Step 6:
As long as Condition P.B is met, V2 will try to form a platoon with V1c, so the former steps are repeated.

Step 7:
No request has been acknowledged by V1c. The distance between V1c and V2 raises until Condition M.C is met. V1c stops sending PCAM.

Result:
No platoon joining has been performed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c
V1c	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V2	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None

4.4.2.4 *Scenario UC2d, Normal joining of two platoons*

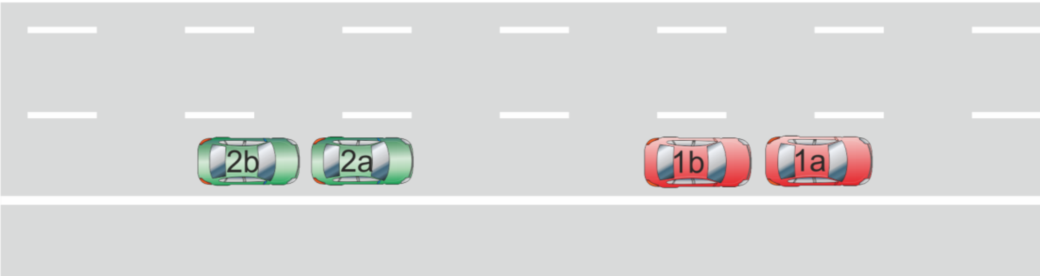

Figure 16: UC 2, joining two platoons. Figure shows the initial situation.

Initial situation:
A platoon of vehicles V2a/b is approaching another platoon with vehicles V1a/b/c. The vehicles therefore have the following states:

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b
V1b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V2a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V2b
V2b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

As result, V1b and V2b are sending ECAMs in the background, and PCAMs without trajectory. V1a and V2a are sending PCAMs with trajectory. V1a/b are reflecting their platoon membership by using the vehicle ID of V1a as platoon leader ID. V2a/b are doing the same with the vehicle ID of V2a. V1a and V2a have 1 followers each, and V1b and V2b have no followers.

Step 1:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

V2a receives PCAMs without a trajectory from V1b. Whenever condition P.E is met, V2a changes its platoon leader ID to the one of V1a and provides one follower. As consequence, condition F.D is met and the forming state machine changes to “waiting for trajectory”.

Step 2:
V1b receives the modified PCAM of V2a and changes its number of followers to three. Furthermore, V1b changes its message state to “sending PCAM, trajectory LF” according to M.B. Either directly (when PCAM of V2a is received) or with a delay due to hopping from V1b to V1a, this information is brought to the other platoon members.

Step 3:
V2a receives the updated PCAM of V1b including the trajectory and changes its forming state to “currently forming” according to F.A.
V2a directly starts reducing the time headway to V1b.

Step 4:
When V2a reaches the desired time headway and distance, condition F.B is met and V2a’s forming state changes to “normal platooning”, which is reflected in the PCAM.

Result:
V1a/b and V2a/b have formed one single platoon with the following states:

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V2a, V2b
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V2a, V2b
V2a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V2b
V2b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.2.5 *Scenario UC2e, Vehicle ahead joins existing platoon*

Initial situation:
A platoon of vehicles V2a/b is approaching a single vehicle V1. The vehicles therefore have the following states:

	Platooning	Forming	Message	Distance	Followers
V1	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None
V2a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V2b
V2b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

As result, V2b is sending ECAMs in the background, and PCAMs without trajectory. V2a is sending PCAMs with trajectory. V1 is not sending any PCAMs. V2a/b are reflecting their platoon

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

membership by using the vehicle ID of V2a as platoon leader ID. V2a has 1 follower, and V2b has no followers.

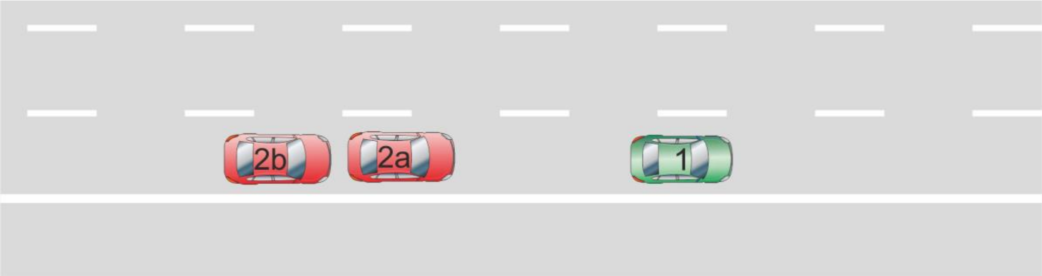

Figure 17: UC 2, joining front vehicle to a platoon. Figure shows the initial situation.

Step 1:
V1 detects that Condition M.A is fulfilled. The message state machine changes its state to “sending PCAM, no trajectory”, resulting in sending PCAM without the trajectory data.

Step 2:
V2a receives PCAMs without a trajectory from V1. Whenever condition P.E is met, V2a changes its platoon leader ID to the one of V1 and provides one follower. As consequence, condition F.D is met and the forming state machine changes to “waiting for trajectory”.

Step 3:
V1 receives the modified PCAM of V2a. It therefore changes its platooning state to “in a platoon” with the own ID as platoon leader ID and two followers. Furthermore, V1 changes its message state to “sending PCAM, trajectory LF” according to M.B.

Step 4:
V2a receives the updated PCAM of V1 including the trajectory and changes its forming state to “currently forming” according to F.A.
V2a directly starts reducing the time headway to V1.

Step 5:
When V2a reaches the desired time headway and distance, condition F.B is met and V2a’s forming state changes to “normal platooning”, which is reflected in the PCAM.

Result:
V1 and V2a/b have formed a platoon with the following states:

	Platooning	Forming	Message	Distance	Followers
V1	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V2a, V2b
V2a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V2b
V2b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

4.4.3 *UC3: Traveling in a platoon*

In this use case, the platooning vehicles have the following states:

	Platooning	Forming	Message	Distance	
Leader	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	All followers
Follower	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	Everyone behind
Last	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

All followers try to keep their own respective distance to the corresponding vehicle ahead.

4.4.3.1 *Scenario UC3a, Lane change*

During normal driving, one frequent situation will be the lane change. This scenario will cover the uninterrupted lane change, as an interruption will commonly result in vehicles which leave the platoon (see UC4).

Initial situation:
A platoon built of 4 vehicles 1a/b/c/d is travelling on the right lane of a multiple-lane road. There is a slower V2 ahead of the platoon. This information is known by all platoon members thanks to cooperative sensing.

Step 1:
V1a decides to change lane in order to overtake V2. V1a therefore changes the value of the planned lane ID and provides information about the targeted trajectory in the PCAM, and sets the indicator.

Step 2:
V1b/c/d receive the updated PCAM and start planning the optimal individual lane change trajectory based on the own situation and the trajectory of the vehicle ahead. The primary goal is to closely follow the leader, but slight modifications required by the situation are possible. Furthermore, all three vehicles set the indicator. V1b/c/d update their planned lane ID in their PCAMs accordingly.

Step 3:
Each vehicle changes lane and passes V2 on the own optimal trajectory.

Step 4:
After passing, V1a detects that it prefers the right lane again and therefore restarts the procedure with the right lane as planned lane ID.

Result:
Vehicle 2 is overtaken.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

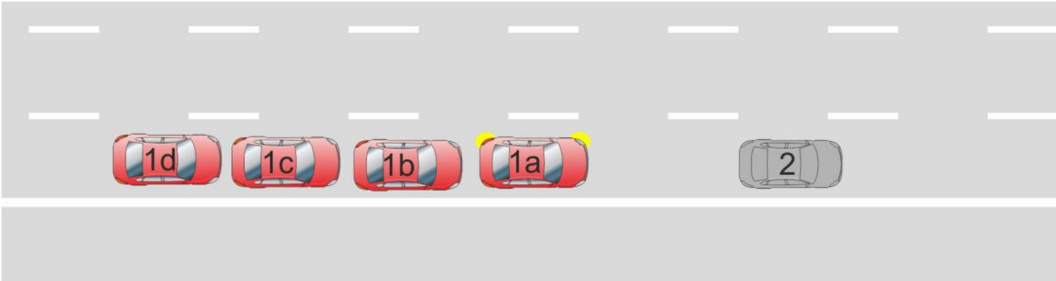

Figure 18: UC 3, travelling in a platoon. Figure shows the situation where a lane change is about to start.

4.4.4 *UC4: Leaving a platoon*

Leaving a platoon can be based on several circumstances. In the following, three scenarios are shown.

4.4.4.1 *Scenario UC4a, Intended leaving of single vehicle*

Initial situation:
There is a platoon of 4 vehicles 1a/b/c/d. V1b wants to leave and to change the lane to the left.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c, V1d
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c, V1d
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

Step 1:
As V1b wants to leave, Condition P.C is met. It sets its state to “leaving” and provides this information to the others by sending the PCAM accordingly. V1b continues to send trajectory data as long as V1c is close.

Step 2:
V1a receives this message and removes the vehicle and its followers (i.e. all vehicles) from the list of followers. As it is now logically separated from the others, condition P.D is met and it has to change the state to “want to form”. This information is provided to the others in the PCAM. As long as V1b is close to V1a, V1a continues to send trajectory data in its PCAM.

In parallel, by receiving the messages from V1b, V1c understands that it is now the formal leader of the platoon. Therefore, it triggers transition P.E and changes the platoon leader ID to its own vehicle ID. As V1b is no longer considered as platoon member, V1c is braking to enlarge the distance to V1b.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

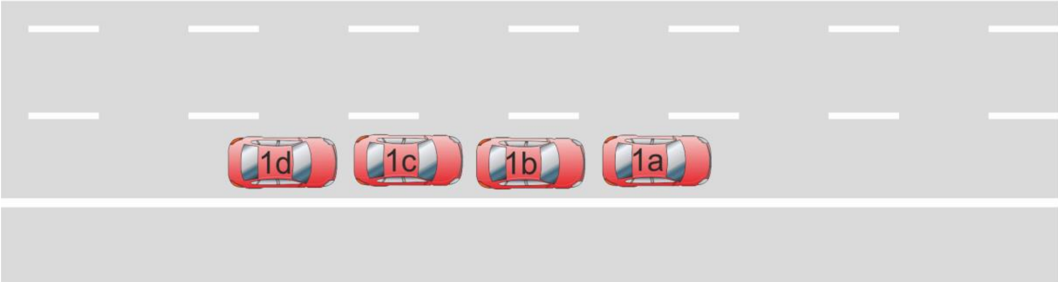

Figure 19: UC 4, leaving a platoon.

Step 3:
V1b changes the lane individually. When the lane change is completed, condition P.D is met resulting in a state change of V1b to “want to form”. V1a detects that V1b is no longer closely behind and therefore changes its message state to “sending PCAM, no trajectory”, according to M.D. V1b does the same with regard to V1c. As other vehicles are still in range, V1a and V1b continue sending PCAM. The states are now:

	Platooning	Forming	Message	Distance	Followers
V1a	Want to form	Waiting for trajectory	sending PCAM, no trajectory	Normal distance	None
V1b	Want to form	Waiting for trajectory	sending PCAM, no trajectory	Normal distance	None
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

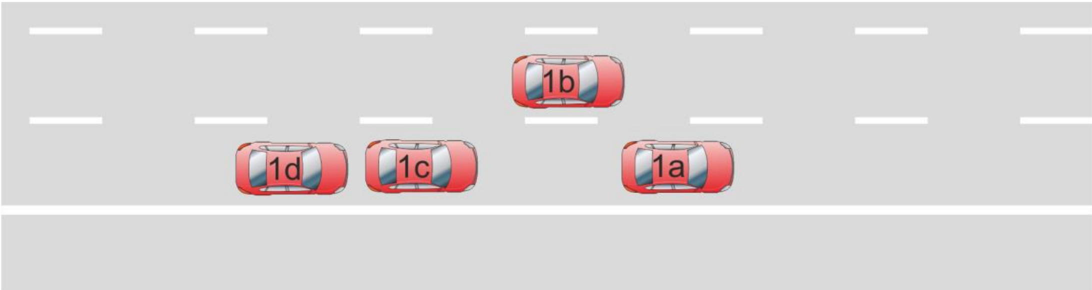

Figure 20: UC 4, leaving a platoon, when vehicle 1b has left

Step 4:
V1a and V1c are in range and build a platoon according to the procedure shown in UC2e. Afterwards, there is a platoon V1a/c/d and a single vehicle V1b on the other lane. As in the meantime V1b has reached a greater distance to the other vehicles, condition M.C is met and it stops sending PCAM.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1c, V1d
V1b	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.4.2 *Scenario UC4b, leaving due to packet loss*

Initial situation:
There is a platoon of 4 vehicles 1a/b/c/d.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c, V1d
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c, V1d
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

Step 1:
V1c does not receive any PCAMs of V1b for at least one second, so it cannot follow this vehicle with a low distance any more. Therefore, V1c changes the platoon leader ID to its own vehicle ID, triggers the distance state condition D.A, starts to enlarge the distance and sends this information in its PCAM.

Step 2:
V1a/b/d receive this changed PCAM. V1a/b remove V1c/d from the list of followers.

In case V1a also did not receive PCAMs from V1b for at least one second, it would have applied the removal of V1b accordingly.

Step 3:
Whenever the distance between V1b and V1c gets large enough according to M.D, V1b stops sending trajectory data.

Result:
In case that V1a received PCAMs from V1b, the final states would be as follows:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b
V1b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.4.3 *Scenario UC4c, Interrupted lane change*

In contrast to UC3a, this scenario will cover an interrupted lane change, where the platoon leader needs to perform a lane change while the tail of the platoon is not able to follow.

Initial situation:
A platoon built of 4 vehicles 1a/b/c/d is travelling on the right lane of a multiple-lane road. There is a slower vehicle 2 ahead of the platoon. On the left lane, there is another vehicle 3 which is driving in the same speed than the platoon. This vehicle is left behind of the last platoon member 1d. This information is known by all platoon members thanks to cooperative sensing.

Step 1:
V1a decides to change lane in order to overtake V2. V1a therefore changes the value of the planned lane ID and optionally provides information about the targeted trajectory in the PCAM, and sets the indicator.

Step 2:
V1b/c/d receive the updated PCAM and start planning the optimal individual lane change trajectory based on the own situation and the trajectory of the vehicle ahead. Furthermore, all three vehicles set the indicator. V1b/c/d update their planned lane ID in their PCAMs accordingly.

Step 3:
While V1a starts the lane change, V1d recognizes that V3 is not making room for a lane change. Therefore, it decides to leave the platoon, triggering Condition P.C, changing state to “leaving” and sending this information in the actual PCAM. This also triggers Condition F.D in terms of forming. As this also triggers D.A, V1d directly starts to enlarge the distance to V1c.

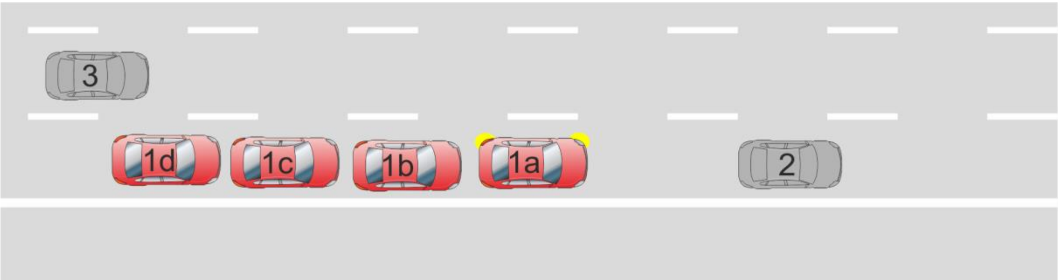

Figure 21: UC 4, leaving a platoon with obstacle on adjacent lane

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Step 4:
V1a/b/c receive the message and remove V1d from the list of followers.

Step 5:
Condition P.D is met for V1d, so the state changes back to “want to form”, making this vehicle again available for other platoons. As V1d has no vehicles able to form a platoon behind, the message state machine changes to “not sending PCAM”.

Step 6:
As the distance between V1c and V1d is rising, V1c meets condition M.D and stops including the trajectory in its PCAMs.

Result:
Now, V1d is no longer driving in a platoon.
The states are now as follows:

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b, V1c
V1b	In a platoon	Normal platooning	sending PCAM, trajectory LF	Close distance	V1c
V1c	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V1d	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None

4.4.4.4 *Scenario UC4d, Interrupted lane change, platoon splitting*

In contrast to UC4c, this scenario will cover an interrupted lane change uncoupling not only a single vehicle, but a part of the platoon.

Initial situation:
A platoon built of 4 vehicles 1a/b/c/d is travelling on the right lane of a multiple-lane road. There is a slower vehicle 2 ahead of the platoon. On the left lane, there is another vehicle 3 which is driving in the same speed than the platoon. This vehicle is left behind of the last platoon member 1d. This information is known by all platoon members thanks to cooperative sensing.

Step 1:
V1a decides to change lane in order to overtake V2. V1a therefore changes the value of the planned lane ID and optionally provides information about the targeted trajectory in the PCAM, and sets the indicator.

Step 2:
V1b/c/d receive the updated PCAM and start planning the optimal individual lane change trajectory based on the own situation and the trajectory of the vehicle ahead. Furthermore, all three vehicles set the indicator. V1b/c/d update their planned lane ID in their PCAMs accordingly.

Step 3:
While V1a starts the lane change, V1c recognizes that V3 is not making room for a lane change. Therefore, it decides to split the platoon. This is done by changing the platoon leader ID to its own

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

vehicle ID, triggering Condition P.E and sending this information in the actual PCAM. As this is also triggering D.A, V1c directly starts to enlarge the distance to V1b.

Step 4:
V1a/b/d receive the message. V1a/b remove V1c/d from the list of followers.

Step 5:
As the distance between V1b and V1c is rising, V1b meets condition M.D and stops including the trajectory in its PCAMs.

Result:
Now, V1a/b and V1c/d are driving in two separated platoons.
The states are now as follows:

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b
V1b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.5 *UC5: Platoon break-up*

In the platoon break-up use case there are other vehicles on adjacent lanes which want to/must change their lane to the lane where the platoon is driving on, resulting in a splitting up of the platoon.

The following scenarios show how the platooning algorithm is handling such situations.

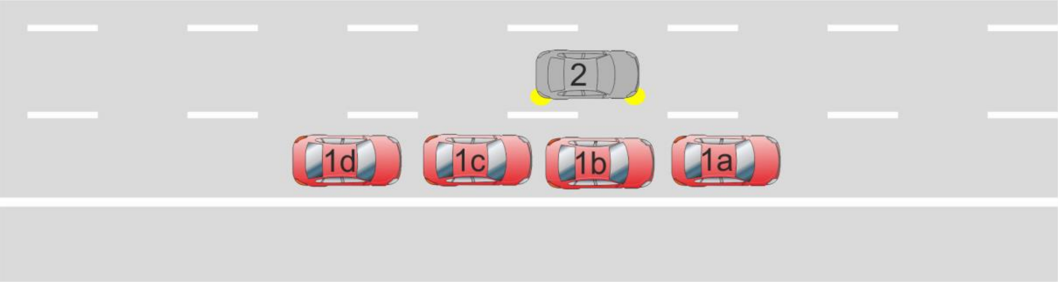

Figure 22: UC 5, platoon break-up. Figure shows the situation where the breaking-up has just started.

4.4.5.1 *Scenario UC5a, Normal break-up*

Initial situation:
A platoon built of 4 vehicles V1a/b/c/d is travelling on the right lane of a multiple-lane road. There is a vehicle V2 left of the platoon which is driving at the same speed then the platoon and indicating to go to the right lane. This information is detected either by a platoon member or the infrastructure and known by all platoon members thanks to cooperative sensing.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Step 1:
V1c detects that condition D.B of the distance state machine is met. Therefore, V1c's trajectory planner takes V2 into account and instantly opens the gap by decelerating. This information is provided to the others by PCAM.

Step 2:
When the gap ahead of V1c is large enough, V2 changes the lane. V1c sees that it is detached from the platoon members ahead and therefore automatically gets the new leader of V1d. V1c therefore changes its platoon leader ID to its own vehicle ID. Besides, V1c's distance state machine changes to "normal distance", as V1c does not receive any trajectory data from V2 and it has not built a platoon with V2. This information is provided to the others in the PCAM.

Step 3:
V1a/b/d receive the message. While V1a/b remove V1c and V1d from the list of followers, V1d automatically adopts V1c as new platoon leader. As condition M.D is triggered for V1b, it automatically changes its message state to "sending PCAM, no trajectory"

Result:
V1a/b are one platoon, V1c/d are another.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b
V1b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.5.2 *Scenario UC5b, Normal break-up with trailer*

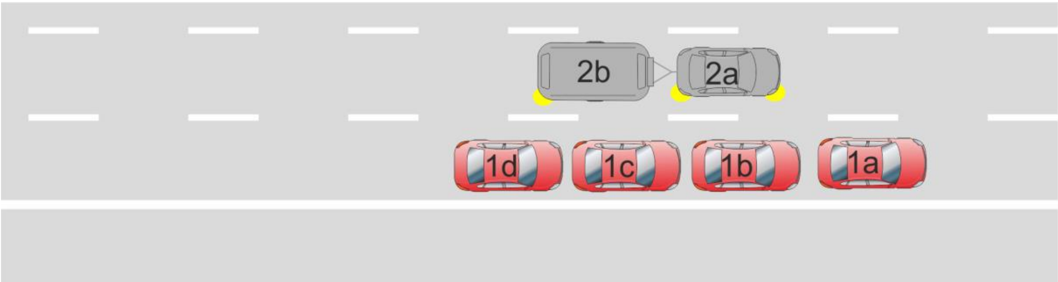

Figure 23: UC 5, platoon break-up with a trailer. Figure shows the situation where the breaking-up has just started.

Initial situation:
A platoon built of 4 vehicles V1a/b/c/d is travelling on the right lane of a multiple-lane road. There is a vehicle V2a with its trailer V2b left of the platoon, driving at the same speed then the platoon and indicating to go to the right lane. Unfortunately, the detector of this was not able to see that this is a vehicle/trailer combination, so both parts are interpreted as separate vehicles. This information is known by all platoon members thanks to cooperative sensing.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Step 1:
V1c detects that condition D.B of the distance state machine is met with V2a. Therefore, V1c's trajectory planner takes V2a into account and instantly opens the gap by decelerating.

In the same moment V1d detects that condition D.B of the distance state machine is met with V2b. Therefore, V2b's trajectory planner takes V2b into account and instantly opens the gap by decelerating.

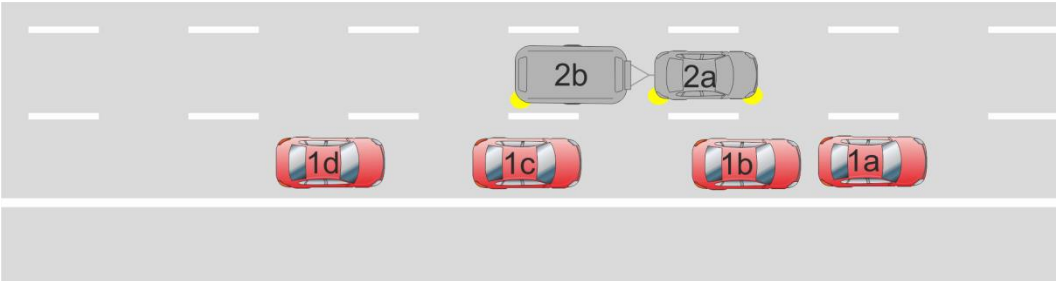

Figure 24: UC 5, platoon break-up with a trailer. Figure shows the situation when V1c and V1d make room.

Step 2:
V1d detects that there is no need to keep up the gap ahead, so condition D.F is met (as V1c continues to send trajectory data) and V1d returns to close distance driving.

Step 3:
V1c now meets condition D.D with V2b, so it is opening the gap based on V2b.

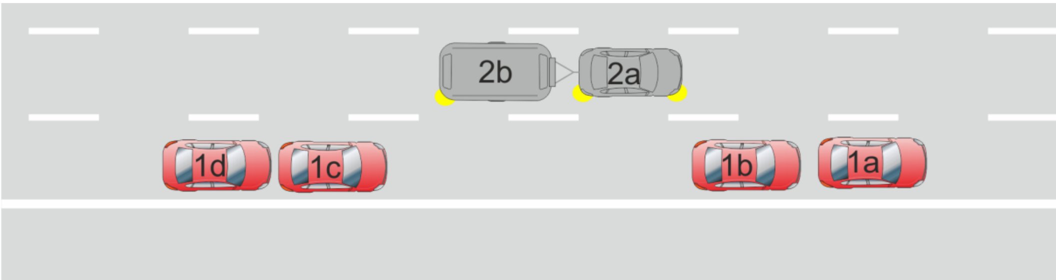

Figure 25: UC 5, platoon break-up with a trailer. Figure shows the situation when V1c and V1d restored their platoon.

Step 4:
When the gap is large enough, V2a/b change the lane. V1c sees that it is detached from the platoon members ahead and therefore automatically gets the new leader of V1d. V1c therefore changes its platoon leader ID to its own vehicle ID. Besides, V1c's distance state machine changes to "normal distance", as V1c does not receive any trajectory data from V2a/b and it has not built a platoon with V2a/b. This information is provided to the others in the PCAM.

Step 5:
V1a/b/d receive the message. While V1a/b remove V1c and V1d from the list of followers, V1d automatically adopts V1c as new platoon leader. As condition M.D is triggered for V1b, it automatically changes its message state to "sending PCAM, no trajectory"

Result:
V1a/b are one platoon, V1c/d are another. V2a/b was able to change the lane as requested.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b
V1b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None
V1c	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1d
V1d	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

4.4.6 *UC6: Platoon termination*

When there are only two platoon members left and one of them is leaving, platoon termination takes place.

4.4.6.1 *Scenario UC6a, Intended leaving*

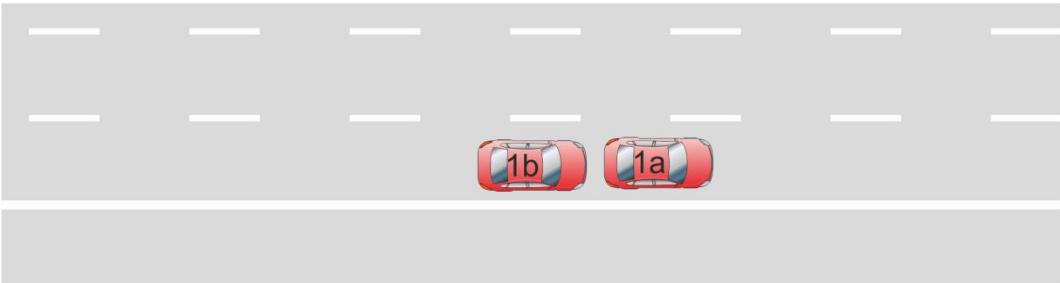

Figure 26: UC 6, platoon termination.

Initial situation:
There is a platoon of 2 vehicles V1a/b. V1b wants to leave and to change the lane to the left.

	Platooning	Forming	Message	Distance	Followers
V1a	In a platoon	Normal platooning	sending PCAM, trajectory LF	Normal distance	V1b
V1b	In a platoon	Normal platooning	sending PCAM, no trajectory	Close distance	None

Step 1:
As V1b wants to leave, Condition P.C is met. It sets its states to “leaving” and provides this information to V1a by sending the PCAM accordingly. This also triggers Condition F.D, resulting in a forming state “waiting for trajectory”. At the same time, condition D.A is met and the vehicle starts to enlarge the distance.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

Step 2:
V1a receives this message and removes the vehicle from the list of followers. As it is now alone, condition P.D is met and it has to change the state to “want to form”. This information is provided to the others in the PCAM. This also changes the forming state to “waiting for trajectory” according to condition F.D. Due to the enlarging of the gap, V1a detects that V1b is no longer closely behind and therefore changes its message state to “sending PCAM, no trajectory”, according to M.D.

Step 3:
V1b changes the lane individually. As this finishes the state “leaving”, condition P.D is triggered, and V1b returns to platooning state “want to form”. Besides, this is triggering M.C and V1b stops sending PCAM.

Step 4:
The distance between V1a and V1b rises. At some point, condition M.C is met for V1a and V1a stops sending PCAM.

	Platooning	Forming	Message	Distance	Followers
V1a	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None
V1b	Want to form	Waiting for trajectory	Not sending PCAM	Normal distance	None

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

5 Cooperative perception

5.1 Introduction

The DLR automated vehicles FASCar E, FASCar II and ViewCar 2 are equipped with the following sensors:

- LIDAR Laser Scanner
- Camera
- ACC Radar Sensor
- Ultrasonic Sensors
- Short Range Radar Sensors

The properties of the used sensors differ in

- their reliability in different weather and lighting conditions,
- their accuracy and precision,
- their viewing range,
- their resolution
- and physical constraints on their applicability.

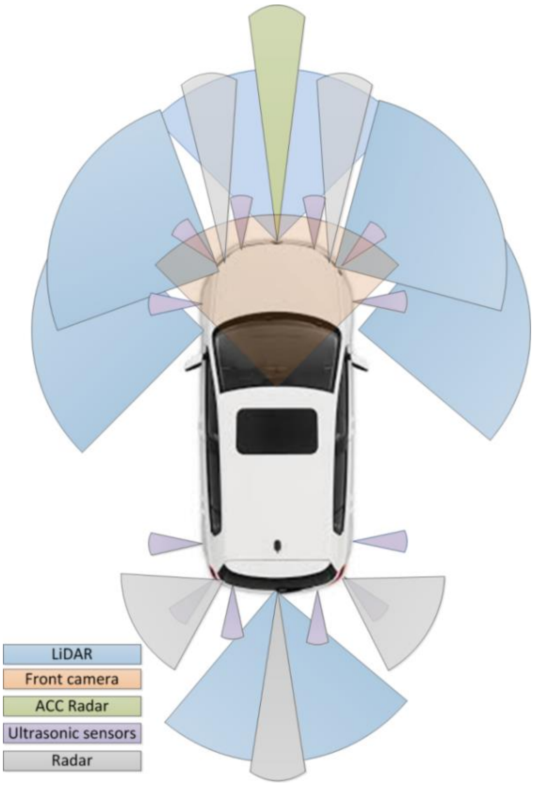

Figure 27: Sensor equipment of automated vehicles

For example, an ACC radar sensor emits electromagnetic waves and registers the Doppler frequency and delay time of their echoes. The wavelength of the electromagnetic waves is in the range of a few millimetres, which is long compared to the wavelength of laser light, which is about one micrometre to less than one micrometre. The long wavelength enables the radar signal to pass fog, rain and snow while the laser scanner will be jammed with nearby false alarm echoes in dense

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

snow, heavy rain or even in the event of large exhaust gas piles in front of the vehicle. However, because of the long wavelength of the radar wave, it cannot be focused as sharp as a laser or light beam. Therefore, the field of view of the radar is a coil whose diameter widens to one or two meters in diameter at long detection ranges. The lack of focus yields continuous echoes from almost everything on the field of view. Therefore, ACC range radar sensors use to filter out objects, which do not move. This is because otherwise the detector output would be populated by a large number of false positive detections. Consequently, using of an ACC range radar detector is not the right choice when collisions with static objects must be avoided.

This example illustrates, that the different sensors have their distinct characteristics. Combining them in a way, such that the strengths of one sensor levels out the weaknesses of the other sensor is a reasonable approach. Sensor fusion algorithms aim to find an optimal solution for the task of estimating a model of the environment of the vehicle given tangible sensor outputs. This task is accomplished by accounting for a-priori knowledge. A-priori knowledge includes

a) the different properties of the sensors involved (sensor models) and
b) physical constraints that can allow to assess if an object hypothesis is sound or not (motion models).

The practical implementation usually includes calibrating and/or learning of sensor and motion models.
Within MAVEN, algorithms are developed and tested to incorporate information provided by infrastructure road side units and by other vehicles into the sensor data processing chain of the vehicle. Advanced sensor fusion algorithms for this objective are already available from literature. The purpose of this chapter is to give an introduction and overview of the relevant sensor fusion concepts and to point out the scope of the MAVEN project in the context of cooperative perception.

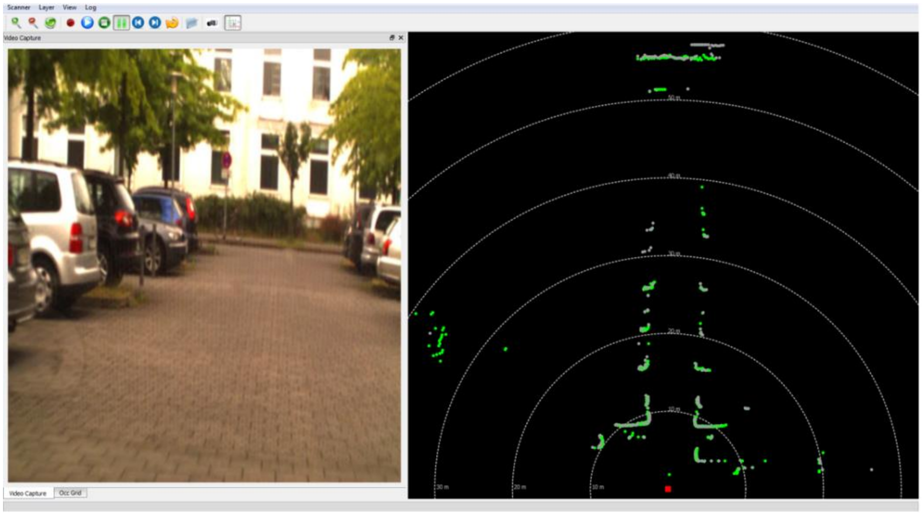

Figure 28: Laser scan of the DLR vehicle FASCAR and corresponding video image

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

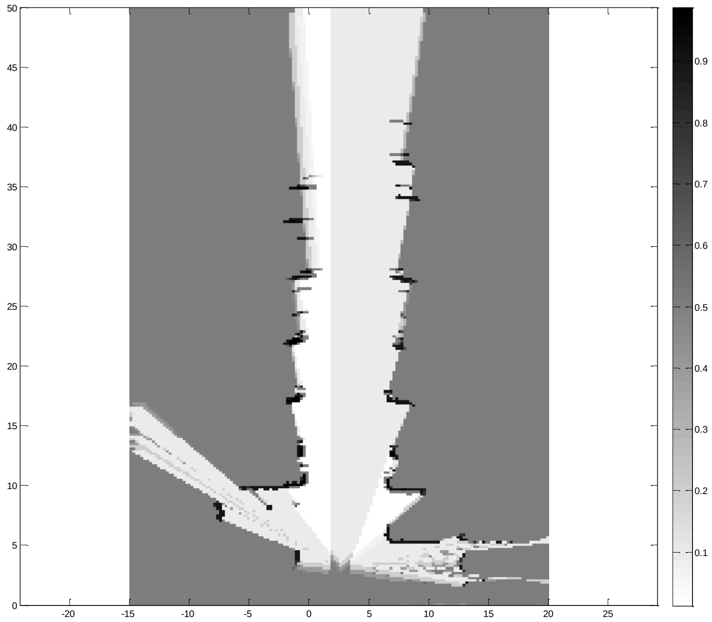

Figure 29: Occupancy grid map generated from the laser scan

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

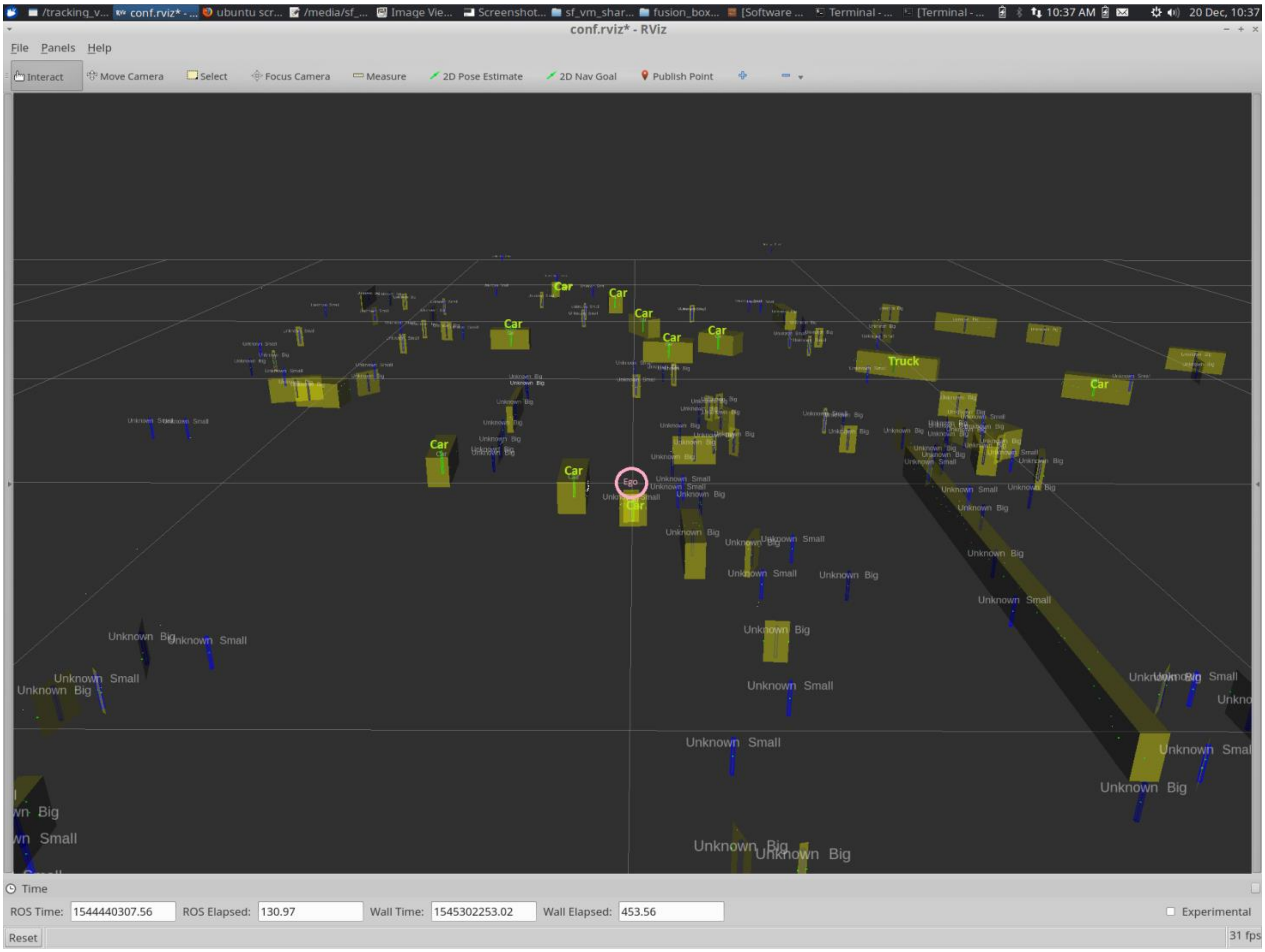

Figure 30: LIDAR Scan of the ViewCar2 vehicle in a complex urban scene. The scene contains cars, many static objects (markes as "Unknown Small") and a Truck. The field of view covers 100m in each direction (Grid size is 25m)

5.2 Fusion paradigms

Data fusion can be applied at different levels in the processing chain. In the context of autonomous driving, it is reasonable to differentiate the concepts

- data fusion based on raw data (low level fusion),
- data fusion based on object detections (object level fusion)
- and data fusion based on object trajectories (track level fusion).

These three concepts yield different outcomes. While a common output of low level fusion based on raw data is an occupancy grid map, fusion based on object detections outputs lists of objects, associated objects or trajectories.

Notes:

- It is reasonable using the output of the low level fusion as input for high level fusion, e.g. object detection.
- The less aggregated the data to be fused the bigger is the potential of data quality improvement by a good data fusion technique. The reason is that the later the stage, the higher is the level of aggregation of data and the more data is lost before fusion.
- A higher level, aggregated fusion output can in principle be used to produce an output that is compatible with a low level fusion output. For this operation, an up-sampling operation is needed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

There are the following algorithm classes suitable for the different data aggregation levels:

- Low Level
    - Bayesian Networks
- Object Level
    - Association (e.g. Global Nearest Neigbor association) combined with Kalman filter
    - PHD-Filter
    - Multi-Bernoulli-Filter
- Tracklet Level
    - Network flow combined with Kalman filter

The state of the art algorithmic approaches are studied within MAVEN in the context of cooperative sensing. Publicly available data sets like KITTI[1], nuscenes[2] and IOSB[3] can be used for algorithm development, unit and integration testing, however they are not suited for testing cooperative perception functions. Within MAVEN, an own dataset is being produced for testing. The dataset contains sensor data from automated vehicles containing connected vehicle from DLR and HMETC and non-connected vehicles in mixed traffic.

5.3 Fusion System Architecture

The sensor data fusion algorithms for cooperative perception are composed of re-usable software building blocks. The building blocks are available as ROS nodes. The following configurations are being investigated within MAVEN:

1. Fusion with Cooperative Perception Message (CPM) from a cooperative sensing vehicle
2. Fusion with Cooperative Awareness Message (CAM) from another vehicle
3. Fusion with CPM from a road side unit (RSU)

The following methods are appropriate:

a) Track Level Fusion, e.g. using a Network Flow based association Technique and Kalman Filtering
b) Object Level Fusion, e.g. with using an association technique and Kalman Filtering
c) Low Level Fusion, e.g. using Bayesian Network

The corresponding system architectures are depicted in building block charts in the figures in this section.

5.4 Notes on the Cooperative Perception Message definition

The current cooperative perception message as it is proposed for standardization, limits the options for cooperative sensor data fusion to object level and track level fusion (Section 5.2). This means, that the use of low level sensor fusion is not yet supported. Cooperative low level fusion has high potential to improve the performance of perception and should be considered for future advancement of the CPM message definition. Because communications channel capacity is limited in V2X communication, data compression techniques need to be introduced in this context.

The current cooperative perception message as it is proposed for standardization limits the ranges for the confidence values of position and speed to relatively small values of less than 1m and less

[1] http://www.cvlibs.net/datasets/kitti/
[2] https://www.nuscenes.org/
[3] https://www.iosb.fraunhofer.de/servlet/is/71820/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

than 1m/s respectively. This means that when there is a sensor with more uncertainty, for instance a stereo camera in longitudinal direction or a radar sensor in lateral direction, the confidence values are truncated. In order to compute an optimal result, data fusion algorithms need correct confidence values, even when the uncertainty is more than one meter. Therefore, the limits of the confidence values should be corrected to larger values, e.g. 50m.

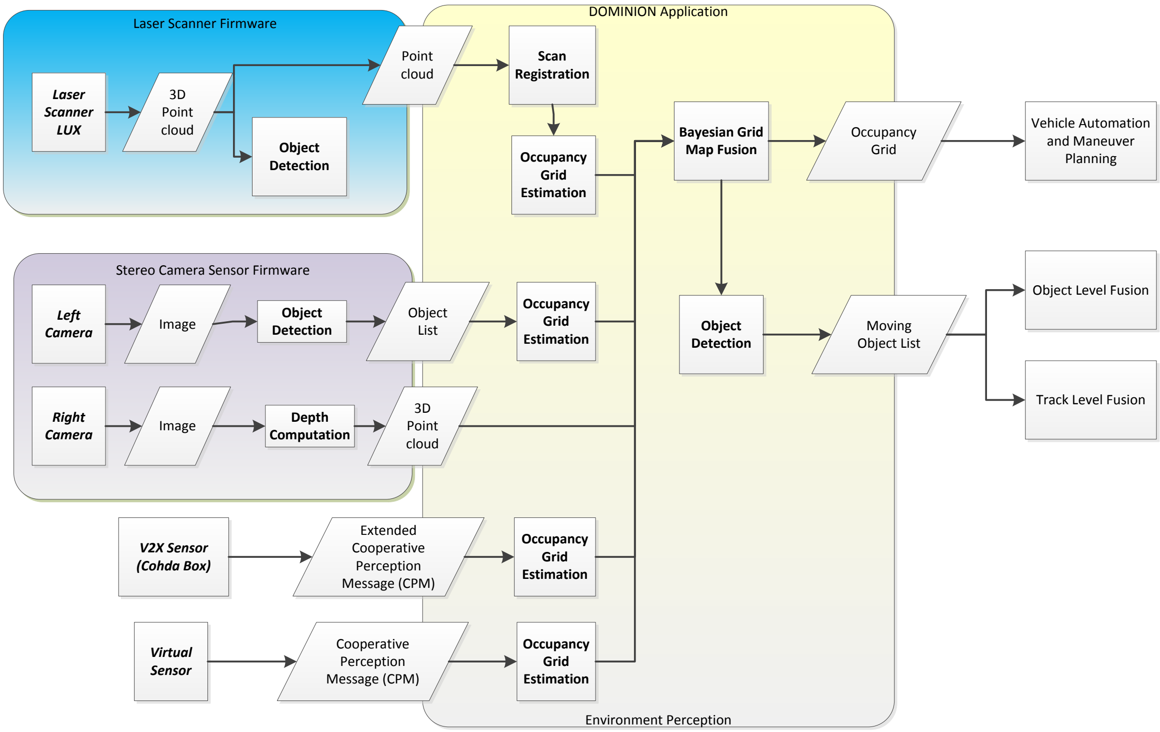

Figure 31: Low level fusion building blocks

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

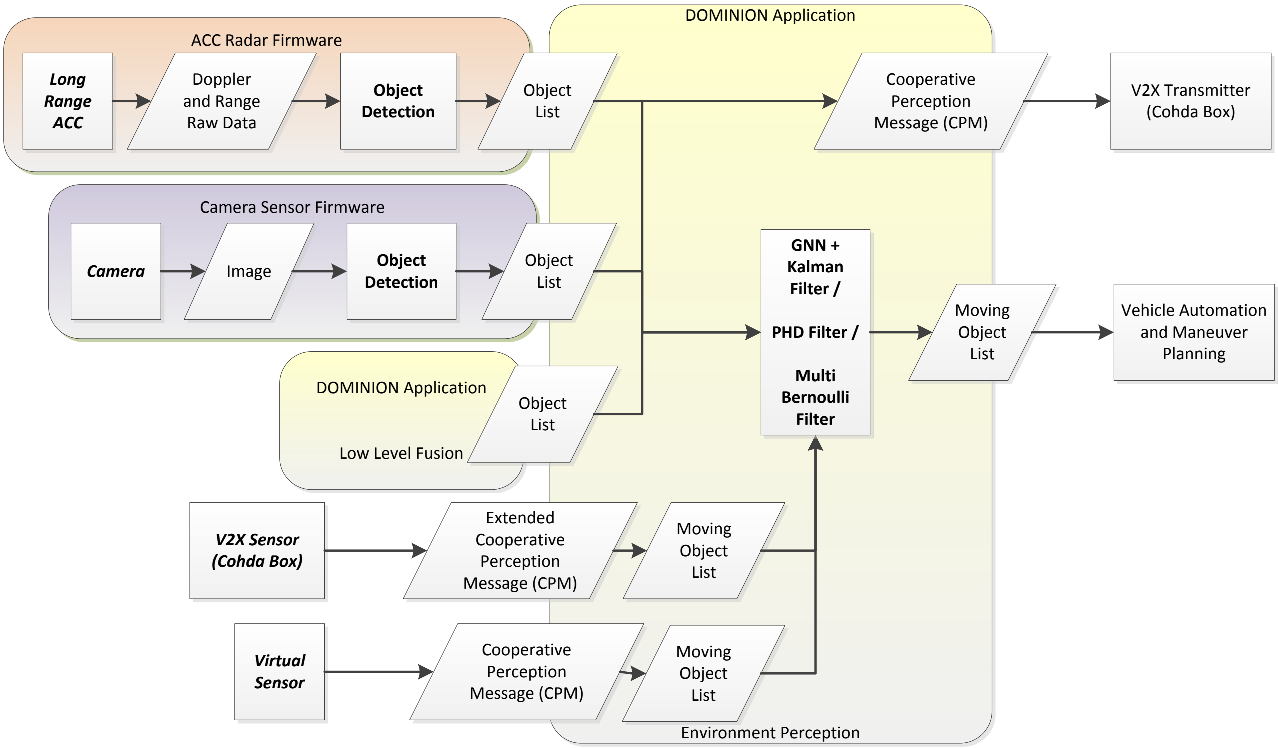

Figure 32: Object level fusion building blocks

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

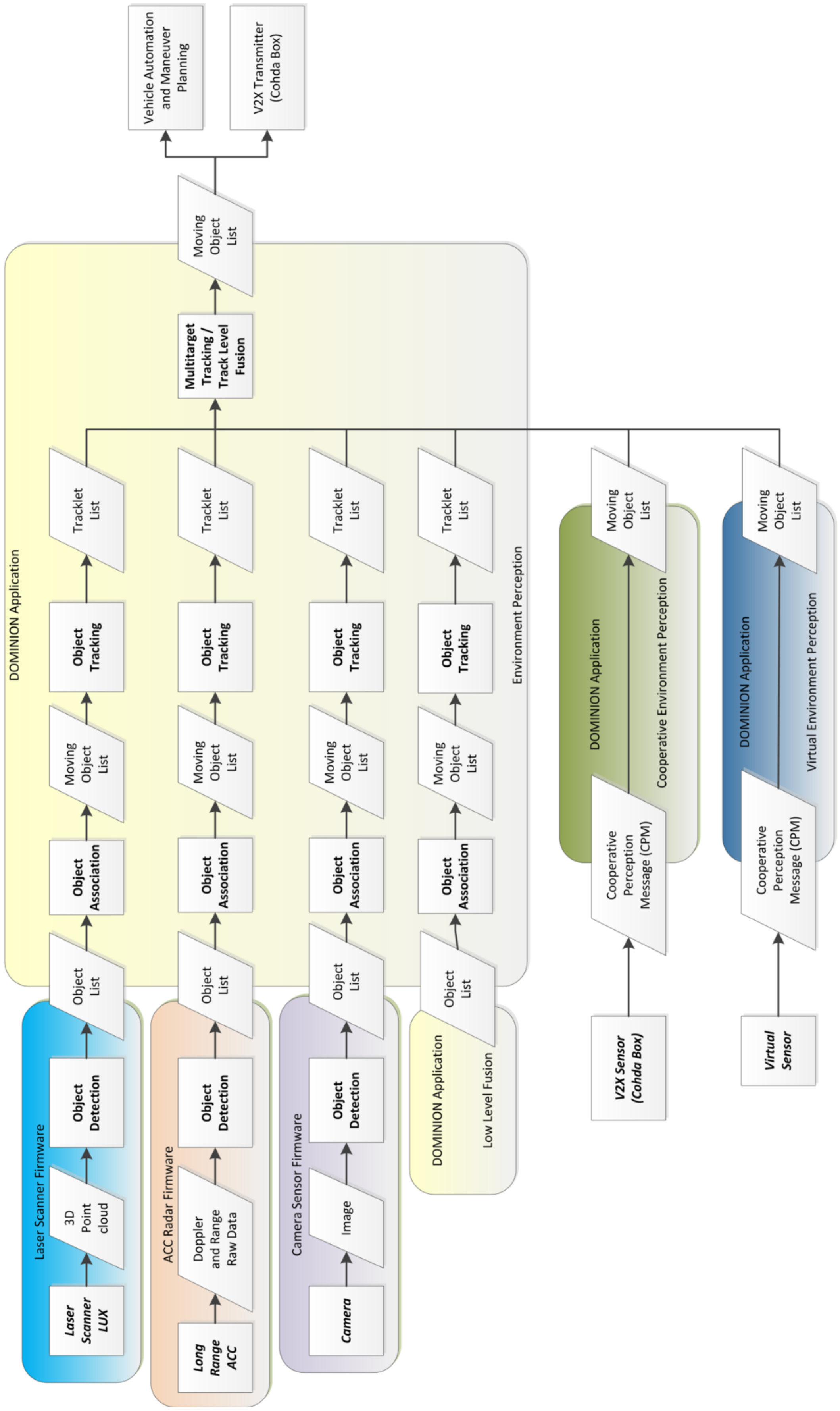

Figure 33: Track level fusion building blocks

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

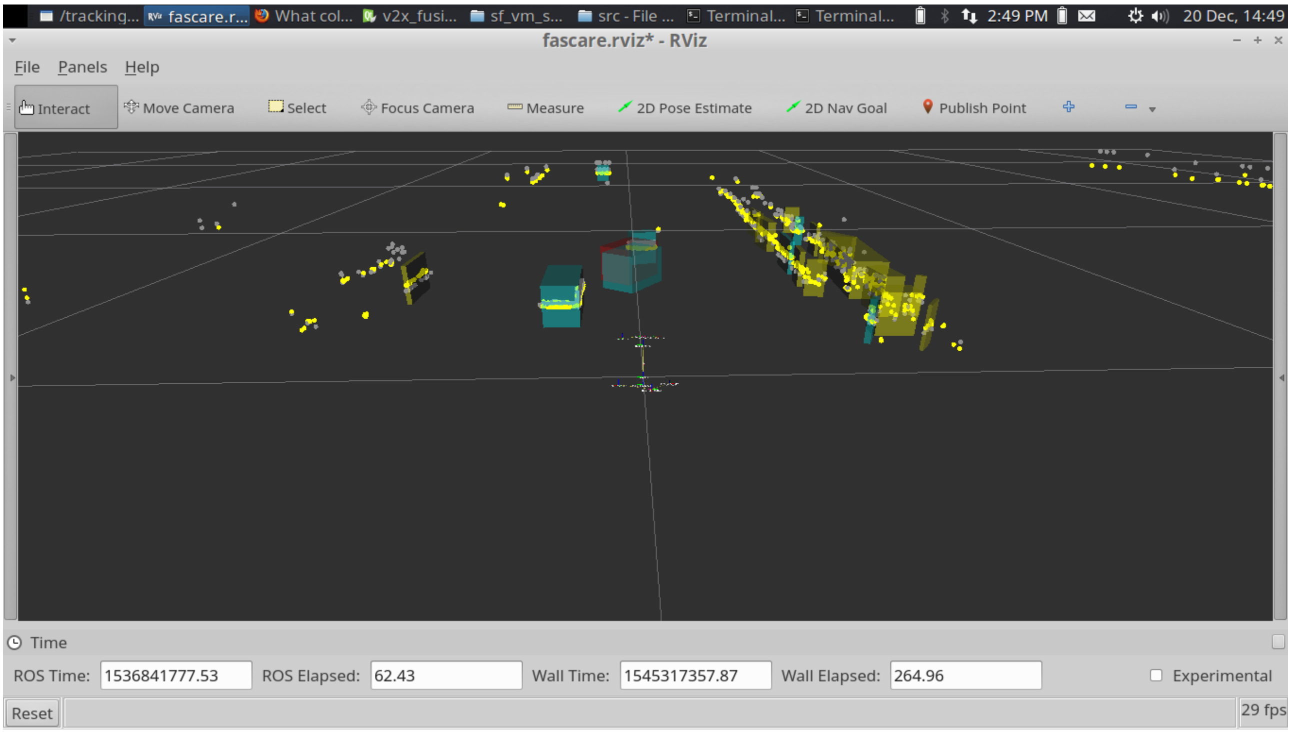

Figure 34: Object level fusion results of fusing CAM with LIDAR recorded on the HMETC test track in Griesheim. There are two vehicles in front of the ego-vehicle: a non-connected in the left lane and a connected vehicle sending CAM-Messages in the same lane. The red box denotes association between CAM message and LIDAR object in front of ego-vehicle.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

6 Conclusion and outlook

In this deliverable the vehicle automation on modular level has been explained for DLR and HMETC. The main blocks of vehicle automation, such as sensors, communication, HD map reader, vehicle controller and etc. have been explained briefly. The simulation environment and virtual sensors used in this environment has been explained.
The state-of-the-art for trajectory planner and environment perception has been briefly reviewed and an advanced algorithm concept has been presented for DLR and HMETC. To deal with unsimilarities of the presented concept of trajectory planning for DLR and HMETC in case on platooning, a unique platoon logic to manage the platooning states has been designed and presented by DLR. As explained this module will be implmeneted by DLR and will be integrated in DLR automation as well as HMETC automation.
The presented concept will be implemented and tested first in simulation and then with vehicles. Deliverables D3.2 and D3.3 explain the result of the implemented concept of D3.1.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

7 References

[1] MAVEN Deliverable 5.2, "ADAS functions and HD map". (July 2018)
[2] MAVEN Deliverable 5.1, "V2X communications for infrastructure-assisted automated driving". (Feb 2018).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690727. The content of this document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.