Chapter 6

SUPERVISOR

6.1 Introduction

A number of urban driving situations are nowadays still too challenging to be handled by an autonomous driving system, and intervention from humans inside the vehicle may be necessary. In this chapter, a novel human-in-the-loop approach based on a traded control architecture is proposed to improve the performance of the ADS by increasing the interaction between the human and the ADS and maintaining a mutual understanding environment. The chapter starts by describing the functional architecture of the supervisor in section 6.2. Then, it introduces the concept of candidate suitability in section 6.3, which is used to determine the complexity level of the current driving (section 6.4), the level of driving automation (section 6.5), and, ultimately, the Required Involvement Level (RIL) for the human pilot on board (section 6.6). Next, the developed human-machine interface is described in section 6.7. The chapter concludes with a use case in an urban environment to illustrate the performance of the supervisor, presented in section 6.8.

6.2 Functional Architecture

The block diagram of Figure 6.1 depicts the architecture of this module, where it can be seen that the module receives the set of decision variables Θ from the trajectory generator, and uses that information to evaluate the suitability of the candidates and estimate the complexity of the driving scene. On the other hand, the supervisor interacts with the human pilot using an onboard HMI and a DMS to set the LoDA and the RIL.

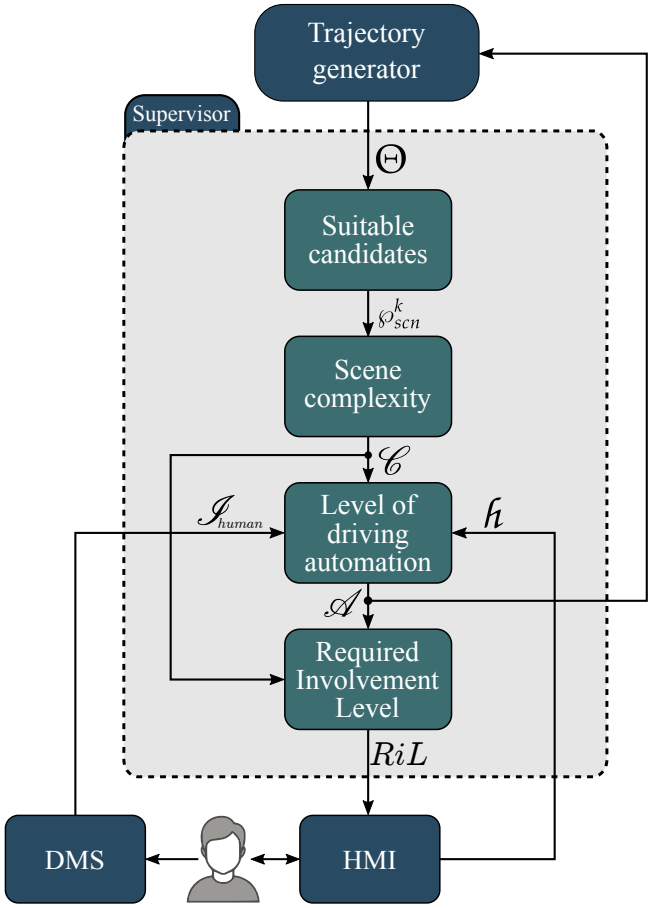


FIGURE 6.1: Supervisor architecture and its interaction with other modules of the ADS.

6.3 Candidates Suitability

The complexity level of the driving scene is estimated based on how appropriate the trajectory set Γ is according to the current driving scene, i.e. if there are several trajectory candidates able to handle the driving scene properly, the scene complexity is low, but if there are few -or none- candidates which are able to handle the current situation, the scene complexity increases. This approach does not measure the complexity based on some predefined complexity indicators, but it rather obtains how well the trajectory generator is handling the current situation, which is more relevant for the trading control strategy. To that end, the concept of candidate suitability is proposed next. The suitability of a trajectory candidate $\tau^c \in \Gamma$ is a binary value computed using its decision variables, as follows:

$$\wp^c = \begin{cases} 1, & \text{if } \mathcal{D}^c_p > \wp^p_{min} \; \forall \, p \in \{0, ..., 3\} \\ 0, & \text{otherwhise.} \end{cases} \qquad c \in \{0, ..., N_T - 1\} \tag{6.1}$$

being $\wp^p_{min}, p \in \{0, ..., 3\}$ the minimum values required for the DVs of the candidate (except the length) to determine if it is suitable or not. So, while a valid candidate is

able to handle the situation even compromising safety or comfort, a suitable candidate handles the situation correctly in all evaluated aspects.

Next, two driving scenes are presented to show how the suitability of the trajectory set varies in different situations. The configuration parameters used for these experiments are depicted in Table 6.1.

TABLE 6.1: Configuration parameters for inter-distance model testing scenarios.

Parameter	Value
N_P	800
N_{SP}	4
$\wp^0_{min}$ (Lon. Comfort)	0.45
$\wp^1_{min}$ (Lat. Comfort)	0.3
$\wp^2_{min}$ (Safety)	0.6
$\wp^3_{min}$ (Utility)	0.2

From this configuration, it can be concluded that the maximum possible number of valid trajectories would be 3200, in case all paths and all speed profiles are valid. Also, it can be observed that safety had the largest priority when evaluating the suitability of the candidates, this is because the main objective of the traded control on an automated vehicle is to increase safety during the driving task.

The first driving scene is a single-lane roundabout with heavy traffic, where the EV is performing a merging maneuver. Figure 6.2 shows the EV, the positions of Ω_{OV}, the candidate set Γ and the waypoints $\mathcal{W}$. The color code used to plot the candidates indicates their suitability: the green candidates are suitable, while the red candidates are not.

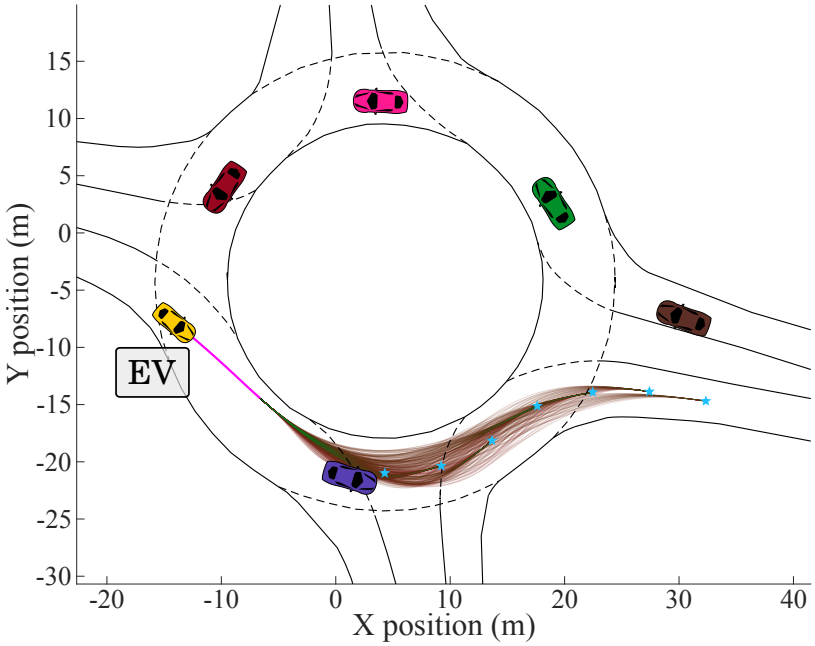


FIGURE 6.2: Driving scene of a roundabout to test the candidates' suitability.

Figure 6.3 shows the reachability map of Θ using the same color code as Figure 6.2. The figure depicts the value of the decision variables for each candidate, as well as the suitability limits $\wp_{min}$ using a cyan line. All candidates with every DV over the limit line are suitable.

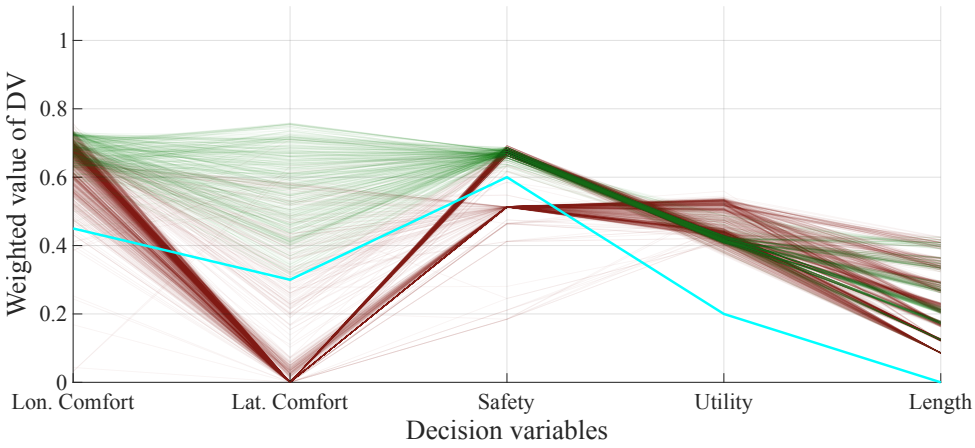


FIGURE 6.3: Reachability map of Θ for a roundabout scenario. Suitable candidates are plotted in green.

For this scenario, there were $N_T = 1464$ valid trajectory candidates to evaluate. The DV with the greatest influence on the suitability of the candidates was lateral comfort (only 31.7% of the candidates were suitable for this DV), followed by safety, with a percentage of 58.1%. The performance for the longitudinal comfort (98.7% of suitable candidates) and the utility (100% of suitable candidates) DVs was generally good for the trajectory set. After taking into account all DVs, 384 out of 1464 were suitable (26.2%).

The second driving scenario is a two-lane highway with a medium level of traffic. Two different planning instants are presented to show how the suitability of Θ varies depending on the type of maneuver. Figure 6.4 shows the traffic status for the selected planning instants on the driving scenario. In the first of them (Figure 6.5a) the EV is following the blue vehicle and all candidates are practically straight due to the geometry of the navigation corridor. On the other hand, during the second planning instant (Figure 6.5b) the EV is performing a lane-change maneuver, and the candidates are being generated on the adjacent-left corridor. It can be observed in this driving scene that the suitable candidates are in the middle of the trajectory set; the candidates ending on the first waypoints require higher lateral accelerations, while some of the candidates ending in the final waypoints reduce safety due to the proximity with the blue vehicle.

Figure 6.5 shows the suitability maps of Θ for the two planning instants of interest. Figure 6.5a shows that most of the candidates were suitable, there were only a few of them that did not reach the minimum limit regarding lateral comfort (94.5% of the candidates were suitable for this DV). These candidates with poor performance on the lateral comfort are mostly short candidates with low curvature but with higher values on its derivative, due to the proximity of the Bézier control points. In general, this traffic scene is simple for the trajectory generator since all 3200 candidates were valid and 3024 of them (94.5%) were suitable.

Figure 6.5b shows that during the overtaking maneuver when the driving scene gets more complex to handle for the trajectory generator. The performance on all

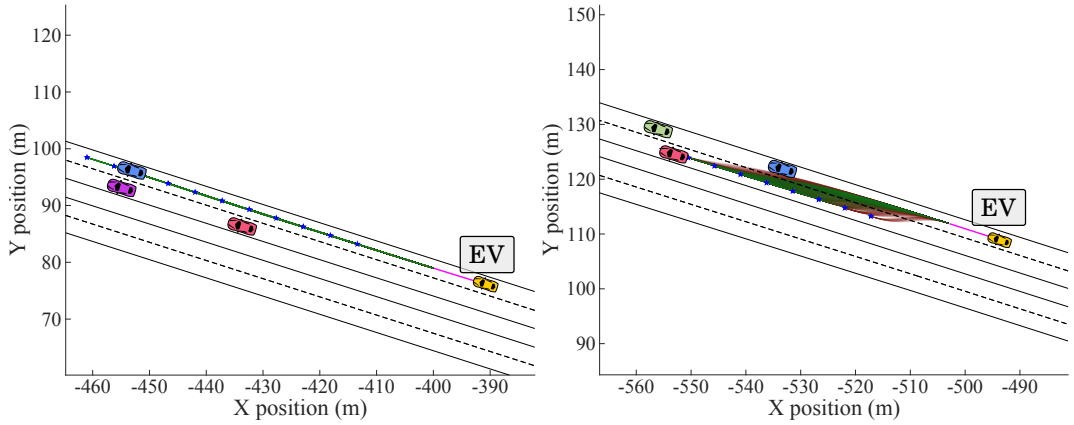


FIGURE 6.4: Suitability of candidates in a highway scenario. (A) Candidates set when following leader vehicle. (B) Candidates set during an overtaking maneuver.

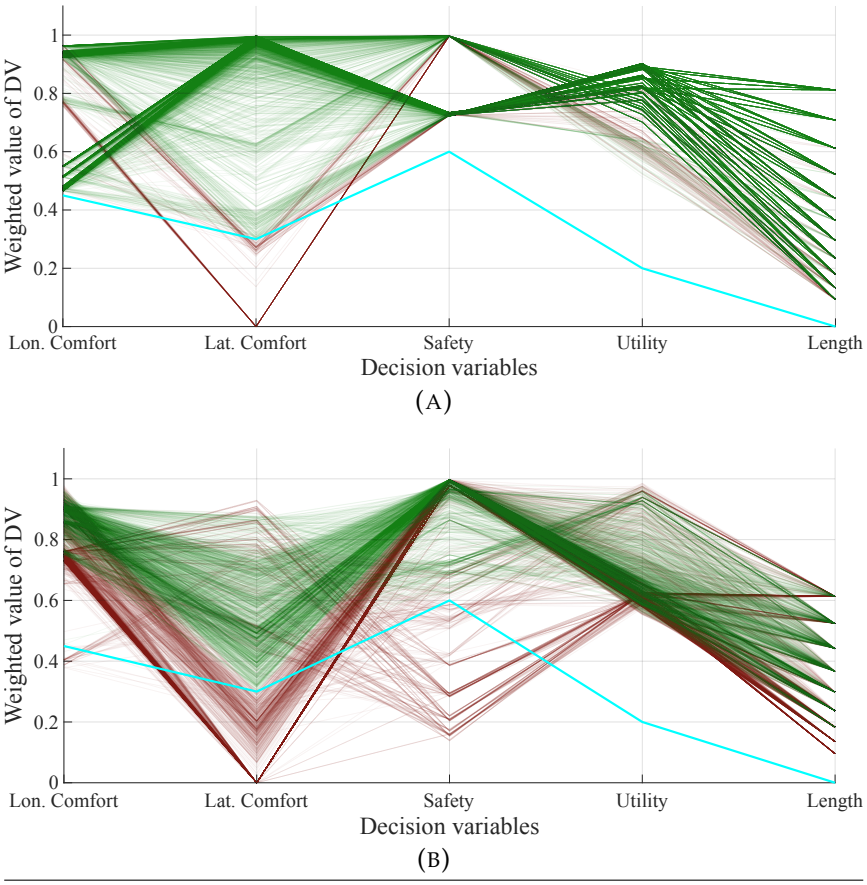


FIGURE 6.5: Suitability of candidates in a following scenario. (A) Suitability map based on the DV of the candidates. (B) Suitability map based on the DV of the candidates

DVs was reduced, compared to the first scene, but the two DVs with the worst performance were the lateral comfort (with only 63.9% of the candidates being suitable on this DV) and the safety (88.9% of suitable candidates). The longitudinal comfort had 99.4% suitability, and the utility, 100%. After combining all of the DVs, 1485 out

of the 2832 valid candidates, were suitable, which represents 53.1%.

Let N_S be the number of suitable candidates for the current driving scene and $\wp^k_{scn}$ be the suitability percentage of the driving scene for a planning instant k. The value of $\wp^k_{scn}$ is computed as the mean of the ratio between N_S and N_T for the last N_{su} planning instants, as follows:

$$\wp^k_{scn} = \frac{1}{N_{su}} \sum_{i=0}^{N_{su}-1} \frac{N_S^{k-i}}{N_T^{k-i}} \tag{6.2}$$

This operation provides a steadier value over time to avoid bouncing effects on the scene complexity estimation, and it also reduces the dynamics of the variable, so an adequate or poor performance must be consolidated before updating the complexity level, and therefore, on the RIL of the human pilot.

6.4 Scene Complexity

The complexity level of the scene is determined with an FSM that uses the suitability percentage of the driving scene to determine the current state out of the set of possible states:

$$S_C = \{\texttt{Complex, Medium, Simple, Basic}\}$$

The state diagram to determine the complexity level $\mathscr{C} \in S_C$ is depicted in Figure 6.6, and the description of the state-changing conditions is presented in Table 6.2.

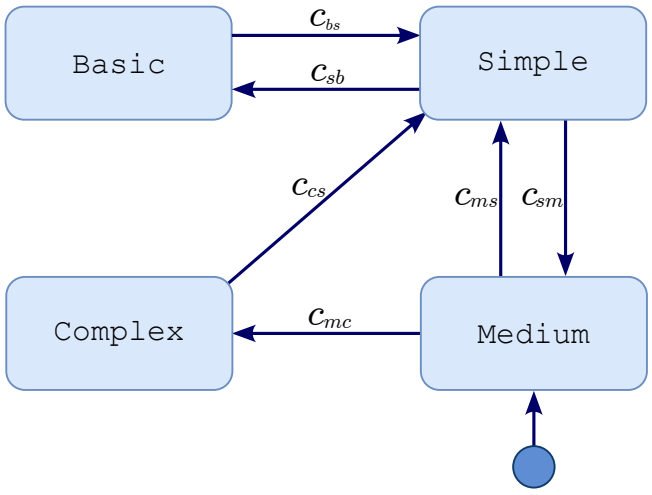


FIGURE 6.6: FSM to estimate scene complexity.

The complexity level varies when $\wp^k_{scn}$ exceeds a series of thresholds: $\mathscr{C}_{comp}$, $\mathscr{C}_{med}$, $\mathscr{C}_{simp}$ and $\mathscr{C}_{bas}$. If $\mathscr{C} =$ `Complex`, this state is maintained until the scene is considered simple, so the behavior of the FSM is more stable in complex scenarios. Let t_{cc} be the period of time the system has been on the current complexity level; once the complexity level state has been updated, it is not possible to change to a simpler state while t_{cc} is lower than a predetermined threshold t_{CL}, although, it is possible to change to a more complex level without this time condition. When the autonomous driving mode is engaged, the initial status of the complexity level is $\mathscr{C} =$ `Medium` by default.

TABLE 6.2: State transition conditions for the complexity level FSM.

Name	Definition
c_{bs}	$\wp^k_{scn} < \mathscr{C}_{simp}$
c_{sb}	$\wp^k_{scn} > \mathscr{C}_{bas} \wedge t_{cc} > t_{CL}$
c_{cs}	$\wp^k_{scn} > \mathscr{C}_{simp} \wedge t_{cc} > t_{CL}$
c_{mc}	$\wp^k_{scn} < \mathscr{C}_{comp}$
c_{ms}	$\wp^k_{scn} > \mathscr{C}_{simp} \wedge t_{cc} > t_{CL}$
c_{sm}	$\wp^k_{scn} < \mathscr{C}_{med}$

6.5 Level of Driving Automation

The SAE J3016 standard [3] is used as a starting point to define the levels of driving automation used in this work. Nevertheless, even though the proposed levels of automation are based on the SAE levels, they do not correspond exactly to the standard definition. Given that a complete disengagement from the human driver may cause dangerous situations if the person is required to intervene, SAE Level 3 is discouraged by many authors [25], as an alternative to this situation the LoDA 3 was modified to require more or less engagement from the human depending on the driving scene complexity, and the human was maintained on-the-loop by using a DMS to estimate the engagement status and an HMI to display the decision-making status. On the other hand, the LoDA 1 of the standard was not implemented because a driver assistance system is out of the scope of this project. Table 6.3 presents the proposed behavior of the autonomous driving system according to each LoDA. Depending on the LoDA, different driving styles are adopted by the EV, and the level of interaction with the people on board varies as well.

The proper LoDA for the ego-vehicle is automatically determined by the supervisor using the FSM depicted in Figure 6.7. The set of possible states of the LoDA FSM is:

$$S_L = \{\texttt{LoDA}_0\,, \texttt{LoDA}_2\,, \texttt{LoDA}_3\,, \texttt{LoDA}_4\,, \texttt{Safe-Stop}\}$$

Besides the automation levels described in Table 6.3, there is a fifth state named `Safe-Stop`, which is proposed to handle critical situations when the human operator is not involved at all, this state can only be reached from $\mathscr{A} = \texttt{LoDA}_4$. If this state is reached, the speed profiles for the trajectories apply a constant deceleration until the ego-vehicle gets to 0 km/h. The $\mathscr{A} = \texttt{Safe-Stop}$ state is maintained until the human operator uses the HMI to resume autonomous driving or take over the wheel.

The transitions between states (described in Table 6.4) depend on (i) the complexity level $\mathscr{C}$, (ii) the responsiveness of the human operator to the HMI requests, and (iii) the involvement level measured by the DMS system.

The involved variables in Table 6.4 are described next:

- $\hbar$ an action requested by the human pilot from the HMI.
- $\hbar = AD$ is a shift-to-autonomous mode request.
- $\hbar = MD$ is a shift-to-manual mode request.
- $\hbar = RS$ is a resume driving request.

TABLE 6.3: Description of the levels of driving automation implemented for the traded control task.

Automation level	Description
LoDA 0	Manual mode. The human pilot is in charge of everything. The supervisor only displays warnings on the HMI.
LoDA 2	Automated mode with conservative driving parameters. Safer candidates are chosen rather than risky ones, and more conservative driving parameters are established. The supervisor suggests the human pilot to take over the wheel if the scene becomes complex. The involvement level requested to the human pilot is high, in case the human is not as involved as the supervisor requests, a visual and audible warning will trigger on the HMI.
LoDA 3	Automated mode with normal-driving parameters. The supervisor may request the human pilot to be ready to take control if the scene becomes complex but also can change to the highest LoDA if the scene has a basic complexity level. The involvement level requested to the human pilot depends on the complexity of the scene, in case the human is not as involved as the supervisor requests, a visual and audible warning will trigger on the HMI.
LoDA 4	Highly automated mode with normal-driving parameters. The supervisor does not request involvement from the human pilot in any situation. In case the scene becomes consistently complex during a determined period of time, if the DMS detects that the human pilot is completely disengaged from the driving task, a safe-stop maneuver is performed, if the human pilot is engaged in the driving task, the level of automation is changed back to LoDA 3.

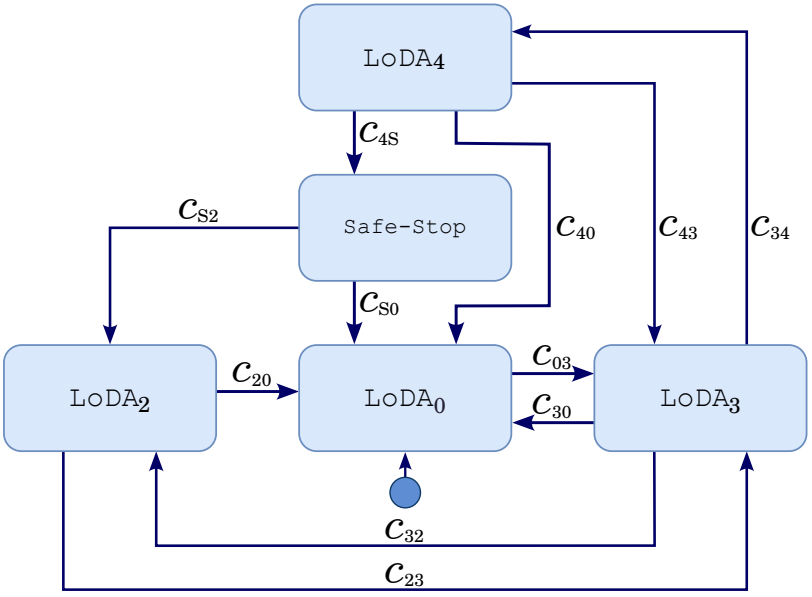


FIGURE 6.7: FSM to establish the LoDA.

- t_r is the response time from the human after a request prompted in the HMI.
- $\mathscr{I}_{human}$ is the current involvement of the human pilot estimated by the DMS.

See Section 6.7 for more details about the possible actions using the HMI.

The initial state for the LoDA is $\mathscr{A} = \texttt{LoDA}_0$ i.e. manual driving. The only way to leave this state is through a $\hbar = AD$ request from the HMI. The autonomous mode

TABLE 6.4: State transition conditions for the LoDA FSM.

Name	Definition
c_{03}	$\hbar = AD$
c_{32}	$\mathscr{C} = \texttt{Complex} \wedge \hbar = \emptyset \wedge t_r > t_{WR}$
c_{23}	$\mathscr{C} = \texttt{Simple}$
c_{34}	$\mathscr{C} = \texttt{Basic}$
c_{43}	$\mathscr{C} = \texttt{Complex} \wedge t_{cc} > t_{CS}$
c_{4S}	$\mathscr{C} = \texttt{Complex} \wedge \mathscr{I}_{human} = LOW$
c_{S2}	$\hbar = RS$
c_{S0}	$\hbar = MD$
c_{20}	$\hbar = MD$
c_{30}	$\hbar = MD$
c_{40}	$\hbar = MD$

is initialized in $\mathscr{A} = \texttt{LoDA}_3$. If the complexity of the scene gets to $\mathscr{C} = \texttt{Complex}$, the supervisor displays a warning using the HMI and starts counting the response time t_r to detect if the human takes control of the vehicle; if the response time gets greater than t_{WR} seconds, without any response from the human pilot, the state is changed automatically to $\mathscr{A} = \texttt{LoDA}_2$. On the other hand, if $\mathscr{A} = \texttt{LoDA}_3$ and the complexity reduces to $\mathscr{C} = \texttt{Simple}$, the automation level is changed to highly automated mode $\mathscr{A} = \texttt{LoDA}_4$. Once in that level, the ADS handles all situations, and only leaves this state if the scene complexity is $\mathscr{C} = \texttt{Complex}$ during a period of time higher than t_{CS} seconds, the new state depends on the current value of $\mathscr{I}_{human}$: if the human pilot is ready to take control ($\mathscr{I}_{human} = HIGH$), the following state is $\mathscr{A} = \texttt{LoDA}_3$, on the other hand, if the driver is not engaged in the driving task ($\mathscr{I}_{human} = LOW$), the following state would be $\mathscr{A} = \texttt{Safe-Stop}$ and the EV will stop. Section 7.2.4 details the onboard DMS of the EV and Section 8.3.4 describes a real-life demonstrator where the ADS was integrated with a DMS on the testing vehicle. The automation level $\mathscr{A} = \texttt{LoDA}_0$ can be reached from any state using the HMI on board, i.e. the human pilot wants to take control of the vehicle, and this is the only way to get to this state.

In case $\mathscr{A} = \texttt{LoDA}_2$, different driving parameters of the trajectory generator and the maneuver planner are updated to have a more conservative driving style. The list of modified parameters is depicted in Table 6.5. Regarding the trajectory generator, the maximum driving speed v_{max} is reduced, also the comfort acceleration limits $\mathcal{L}$ are reduced, the weights of the merit function are updated to prioritize safety and reduce the priority of utility, and the minimum distance of the inter-distance model d_c is increased. Regarding the maneuver planner, the term v_n is reduced, which increases the size of the gaps to consider lane changes, and reduces the utility of adjacent navigation corridors.

6.6 Required Involvement Level

Once the complexity level of the driving scene $\mathscr{C}$ and the level of driving automation $\mathscr{A}$ are established, both variables are combined to request an involvement level to the human pilot, which is a binary variable:

TABLE 6.5: Safe driving parameters for LoDA 2.

Parameter	Value
v_{max}	$v_{max,safe}$
$\mathcal{L}$	$\mathcal{L}_{safe}$
$\mathcal{W}$	$\mathcal{W}_{safe}$
d_c	$d_{c,safe}$
v_n	$v_{n,safe}$
$\mathscr{U}_{op}$	$\mathscr{U}_{op,safe}$

$$RiL = \{\text{LOW}\,,\,\text{HIGH}\} \tag{6.3}$$

To that end, the correspondence matrix presented in Table 6.6 is used.

TABLE 6.6: Traffic-regulation restrictions on speed profile generation.

$\mathscr{A}$ \ $\mathscr{C}$	Complex	Medium	Simple	Basic
$\texttt{LoDA}_2$	HIGH	HIGH	HIGH	HIGH
$\texttt{LoDA}_3$	HIGH	HIGH	LOW	LOW
$\texttt{LoDA}_4$	LOW	LOW	LOW	LOW

Notice that as the level of automation increases, the requested involvement from the human operator gets lower. In $\mathscr{A} = \texttt{LoDA}_2$, the human must be involved all the time no matter the scene complexity, in case of $\mathscr{A} = \texttt{LoDA}_3$, the required involvement level varies according to the complexity of the scene, more complex scenes require higher involvement; finally, if $\mathscr{A} = \texttt{LoDA}_4$, no involvement is required from the human operator at all. If the involvement level estimated by the DMS $\mathscr{I}_{human}$ is lower than the one requested, an audiovisual alarm will be prompted in the HMI. Even though the development of a DMS was not in the scope of this work, the supervisor was integrated with a third-party DMS described in section 7.2.4.

6.7 Human-Machine Interface

The EV has an onboard HMI to interact with the human pilot which shows the navigation map, the position of the EV, the current trajectory, the requested involvement level, the estimated driver status, and different warnings, among other information. Furthermore, there are different actions the human pilot can perform using buttons on the HMI. Let $\mathcal{H}$ be the finite set of possible actions, defined as:

$$\mathcal{H} = \{AD\,,\,MD\,,\,SS\,,\,RS\,,\,RNR\,,\,ANR\,,\,COT\} \tag{6.4}$$

The current action $\hbar \in \mathcal{H}$ activates the autonomous mode driving when $\hbar = AD$. In case $\hbar = MD$, the human pilot takes control of the vehicle and autonomous driving is disengaged. If the current action is $\hbar = SS$, the level of driving automation is

updated to $\mathscr{A} =$ `Safe-Stop`, and a safe-stop maneuver is performed by the trajectory generator; the autonomous driving can be resumed if $\hbar = RS$. When the current action is $\hbar = RNR$, a new global route is requested from the HMI to a specific destination point. Also, if the current route is completely blocked, the maneuver planner computes an alternative route (if there is any available), which is shown in the HMI and has to be confirmed by the human pilot; this confirmation is $\hbar = ANR$. An example of this situation is presented in the demonstrator described in section 8.3.3. Finally, if an overtaking maneuver on an opposite lane is being performed by the EV, this maneuver can be manually canceled by the human from the HMI, in this case, $\hbar = COT$. Figure 6.8 shows the information displayed on the HMI during a navigation mission.

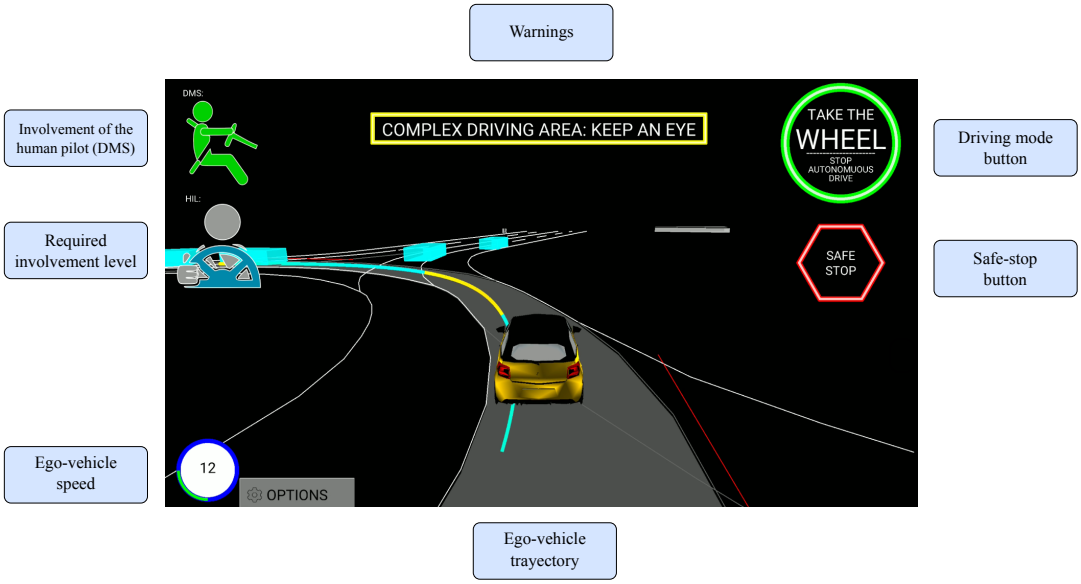


FIGURE 6.8: HMI elements description.

The required involvement level is shown in the HMI with different icons according to the value of RiL. These icons are presented in Figure 6.9a. On the other hand, the $\mathscr{I}_{human}$ level is also presented in the HMI using the icons of Figure 6.9b.

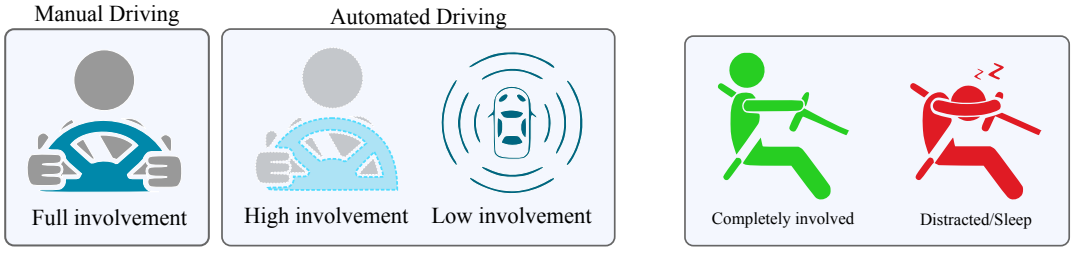


FIGURE 6.9: Traded control HMI icons. (A) Requested involvement levels. (B) DMS involvement levels.

The computation of a new global route takes into account the current position of the EV (p_{ego}) and the destination point (p_{des}). The latter can be selected from the HMI using a bird-view of the digital map. After the maneuver planner has computed the global route, it is presented in the HMI for the human pilot to confirm it. Figure 6.10 presents an example of a global route as presented in the HMI for the testing track

of the Centre for Automation and Robotics. The global route is highlighted in green, and it starts from the current position of the EV and ends at the selected destination point.

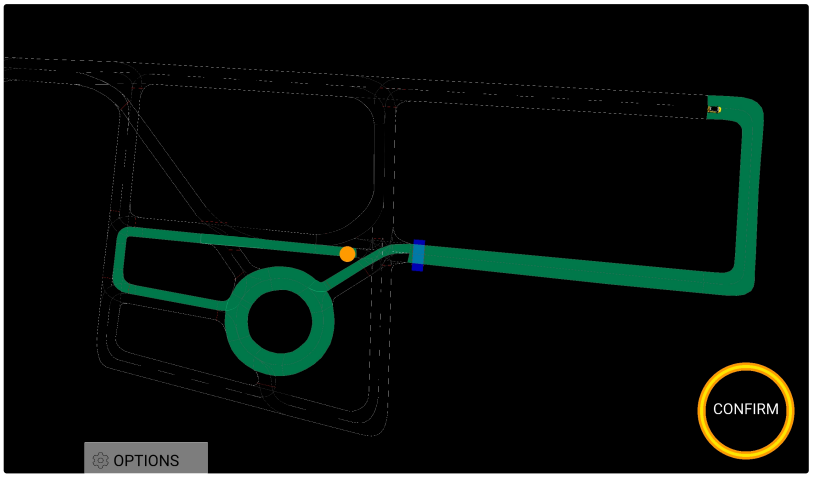


FIGURE 6.10: Selecting a global route from the HMI.

6.8 Use case for Supervisor

The performance of the supervisor is evidenced in an urban scenario with crossroads and roundabouts in a realistic simulation environment. The scene difficulty, level of driving automation and required involvement level of the human pilot are adapted to the traffic status as the vehicle travels along the urban environments, as described later in this section. Table 6.7 shows the configuration parameters used for the experiment, being $\mathcal{L}_{max} = \{acc_{\mathcal{L},max}, dec_{\mathcal{L},max}, lat_{\mathcal{L},max}\}$ the maximum acceleration limits in normal mode and $\mathcal{L}_{safe,max} = \{acc_{\mathcal{L},safe,max}, dec_{\mathcal{L},safe,max}, lat_{\mathcal{L},safe,max}\}$ the maximum acceleration limits in safe driving mode.

The complete trajectory followed by the EV during the experiment is presented in Figure 6.11. The color of the path corresponds to the current value of $\mathscr{A}$ along the travel: orange is for $\mathscr{A} = \texttt{LoDA2}$, green is for $\mathscr{A} = \texttt{LoDA3}$ and blue for $\mathscr{A} = \texttt{LoDA4}$. The figure also identifies some of the traffic scenarios found along the route of the experiment. In the setup of the experiment, the ego-vehicle first encountered a four-way crossroads (CR) with three OVs incoming and outgoing from it; then, two T-shape intersections (T1 and T2) and finally two roundabouts (R1 and R2) on the global route. R1 was configured to have heavy traffic, and R2 to be traffic-free. It can be observed how the LoDA is reduced when approaching the intersections and roundabouts, and it is maintained that way during the complex areas.

In the interest of showing the traffic status at different moments of the experiment, four relevant time steps —marked in Figure 6.11 as A, B, C, D— are detailed in Figure 6.12, which depicts the status of the HMI for those time steps. It can be observed that for A, B, and C, there are OVs present on CR, T1, and R1, respectively. In those cases, the supervisor is displaying a warning of a complex scenario on the HMI, and there is a high involvement level required from the human pilot. For time step D, there are no OVs and the scenario is a straight road, so there is no complex scenario warning, and the required involvement level is low.

TABLE 6.7: Configuration parameters for supervisor module use case.

Parameter	Value	Units
v_{max}	50	km/h
$v_{max,safe}$	25	km/h
$\mathcal{L}_{max}$	$\{2.5, 2.0, 2.0\}$	m/s^2
$\mathcal{L}_{safe,max}$	$\{2.0, 1.5, 1.5\}$	m/s^2
$\mathcal{W}$	$\{0.7, 0.7, 0.6, 0.4\}$	–
$\mathcal{W}_{safe}$	$\{0.9, 0.3, 0.7, 0.1\}$	–
d_c	7	m
$d_{c,safe}$	13	m
v_n	1.0	–
$v_{n,safe}$	0.3	–
t_{ur}	0.2	s
$\mathscr{C}_{comp}$	25%	–
$\mathscr{C}_{med}$	45%	–
$\mathscr{C}_{simp}$	65%	–
$\mathscr{C}_{bas}$	80%	–
$\wp_{min}$	$\{0.6, 0.6, 0.7, 0.2\}$	–
t_{CL}	10	s
t_{WR}	4	s
t_{CS}	15	s
N_{su}	5	–
N_P	900	–
N_{SP}	4	–

Figure 6.13 plots the data of the traded control module during the experiment. This data includes candidates' suitability (6.13a), scene complexity level (6.13b), level of driving automation (6.13c) and required involvement level (6.13d). The dotted vertical lines emphasize the moments where the EV crossed the yielding lines of the regulatory elements highlighted in Figure 6.11.

The evolution of the experiment over time as the EV navigates through the different regulatory elements is described next:

- Before crossing CR:
  - At the beginning of the experiment, the complexity level was set to $\mathscr{C} =$ `Medium`, and the LoDA was set to $\mathscr{A} =$ `LoDA3`.
  - Since the EV was driving on a straight road with no traffic, the suitability scene $\wp^k_{scn}$ increased at the beginning of the experiment and, after 10 s, the complexity level changed to $\mathscr{C} =$ `Simple`;

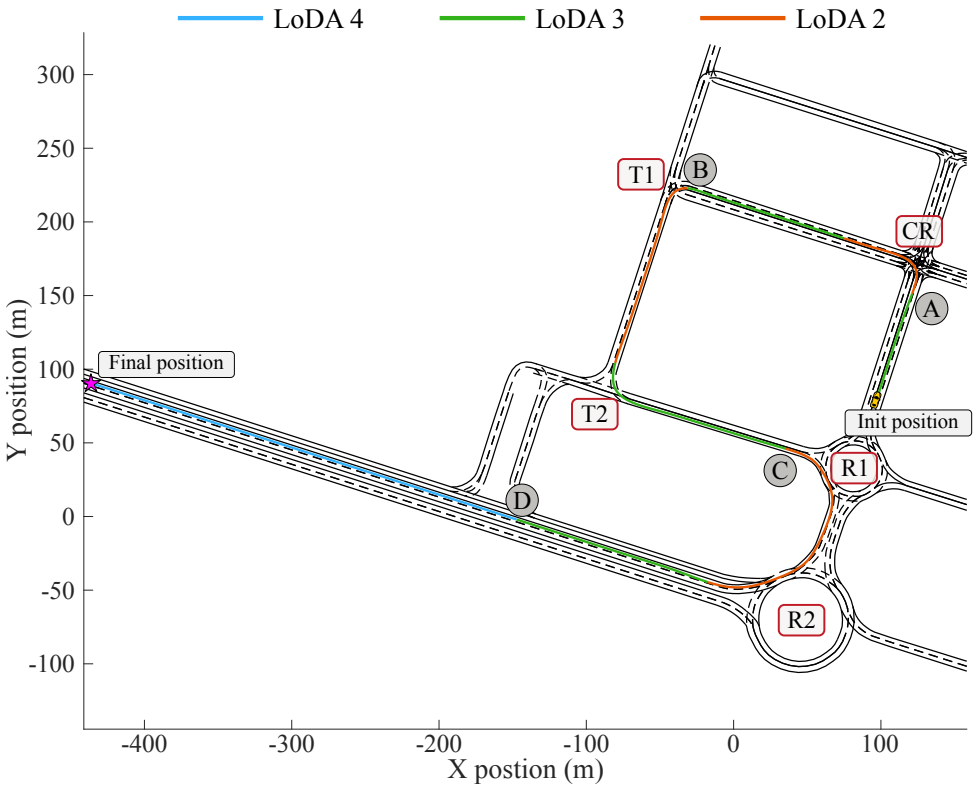


FIGURE 6.11: Complete trajectory followed by the EV in the use case for the supervisor.

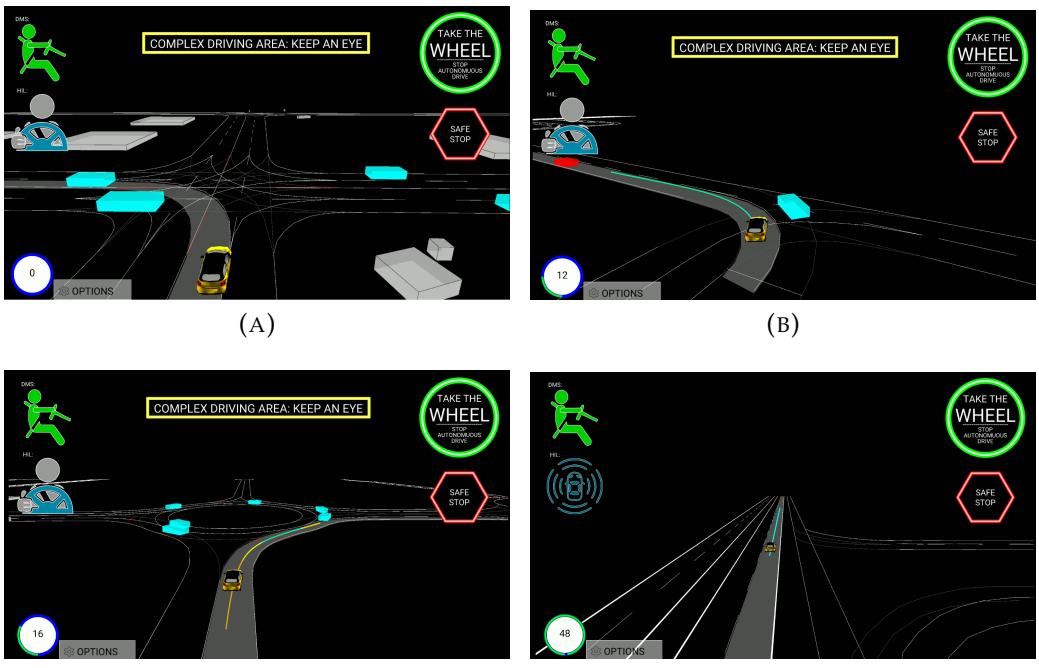


FIGURE 6.12: HMI status regarding four relevant time steps during the use case, marked in Figure 6.11 as A, B, C, D.

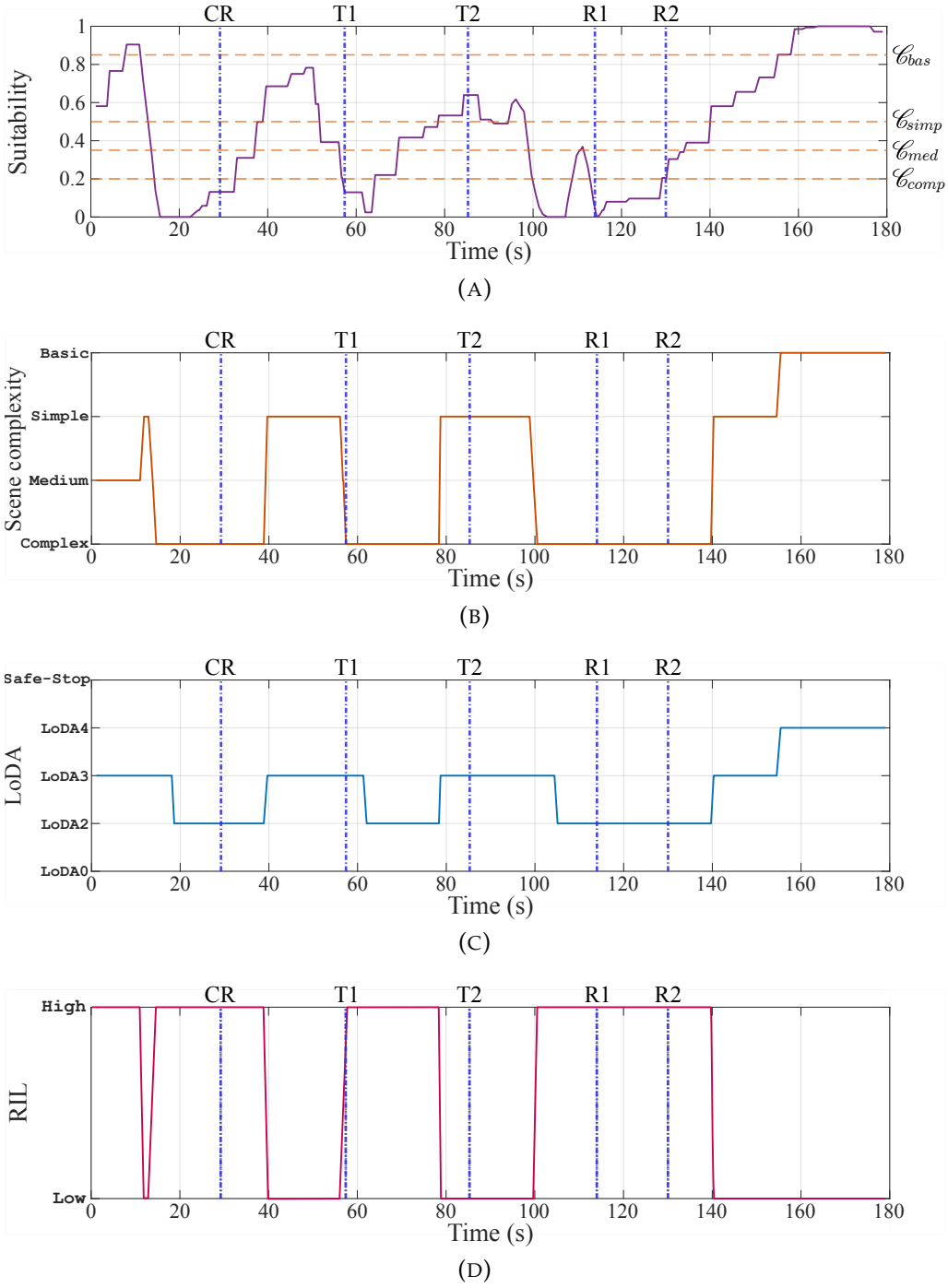


FIGURE 6.13: Traded control data during the supervisor use case. (A) Candidates' suitability. (B) Scene complexity level. (C) Level of driving automation. (D) Required involvement level.

- As the EV approached the CR regulatory element, the value of suitability reduced, getting to $\wp^k_{scn} < (C_{comp})$ 5 seconds before stopping on the yielding line. The scene complexity was updated at that moment.
- A warning of a complex zone was displayed on the HMI, and, since the human pilot decided to leave the vehicle in autonomous mode, the LoDA reduced 4 seconds later to $\mathscr{A} =$ `LoDA2`. It remained that way until the EV crossed completely the CR zone.

- Between CR and T1:
  - The suitability increased again after completing CR and the complexity and the LoDA changed to $\mathscr{C} =$ `Simple` and $\mathscr{A} =$ `LoDA3`, respectively.
  - When crossing T1, the suitability is reduced due to the traffic conditions (this is the time step shown in Figure 6.12b).
  - Figure 6.14 shows the current state of the driving scene at that moment, as well as the set of trajectory candidates distinguishing the suitable and not suitable ones. At this time step the EV is crossing the yielding line of the T intersection with a green OV very close to it on the adjacent lane and a purple leader OV in the current lane. All candidates that invade the adjacent lane lead the EV to hard braking and dangerous situations, hence, all of these candidates are not suitable.
  - The reachability map of the decision variables set Θ is displayed in Figure 6.15, which also shows the suitability limits $\wp_{min}$ using a cyan line. In this case, there were 3136 valid trajectory candidates and only 215 of them were suitable, which results in an instant scene suitability percentage of 6.8%.

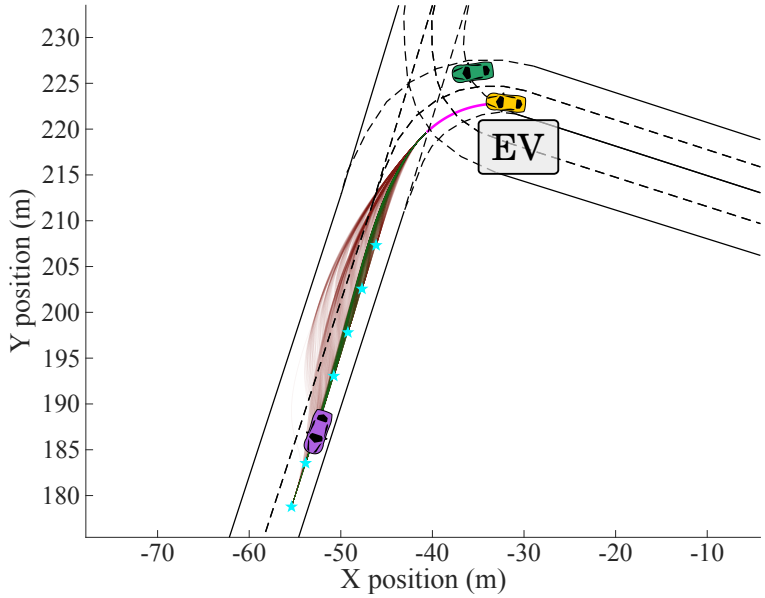


FIGURE 6.14: Driving scene during T1 for the supervisor use case experiment.

- During T1 and T2:
  - Once T1 was completed the suitability of the scene started to increase and the scene complexity got to $\mathscr{C} =$ `Simple` before reaching T2, remaining that way until the EV approached the first roundabout R1.
- Crossing R1 and R2:
  - When the EV approached R1 the suitability was affected due to the high traffic and the road topology (see Figure 6.12c), the complexity level was updated to $\mathscr{C} =$ `Complex` and the LoDA was reduced 4 seconds later since the human pilot did not take control over the vehicle.

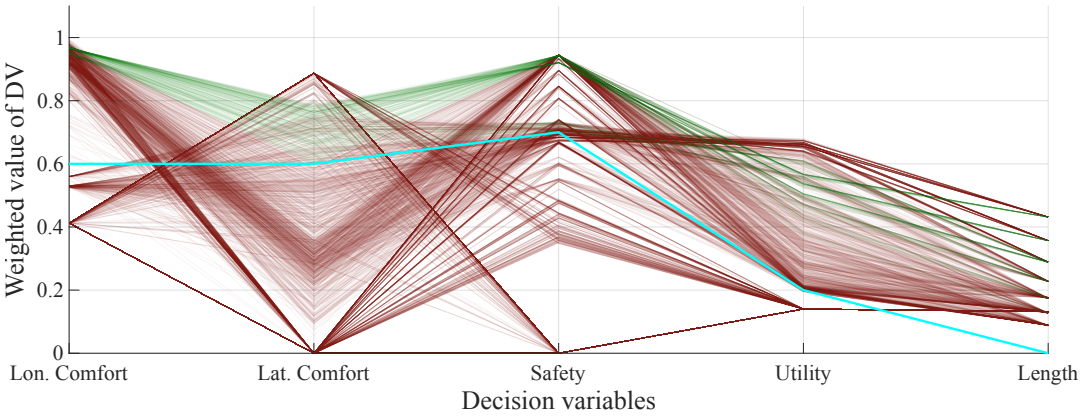


FIGURE 6.15: Reachability map of Θ for T1 RE. Suitable candidates are plotted in green.

  - The traded control status remained this way until both roundabouts were crossed.

- Highway zone:

  - Towards the end of the experiment, the EV reached a highway scenario with no traffic and it could almost reach the maximum level of suitability, getting to a complexity level $\mathscr{C} = \texttt{Basic}$ and reaching the maximum level of driving automation $\mathscr{A} = \texttt{LoDA4}$.

Figure 6.16 shows the complete speed profile during the travel of this use case. The figure shows the limits v_{max} and $v_{max,safe}$ using green and red dotted lines, respectively; and the time periods where $\mathscr{A} = \texttt{LoDA2}$ are highlighted in orange. The speed profile shows a smooth behavior of the EV for handling all traffic scenarios of the experiment. The vehicle only stopped on the yielding line of CR and R1, and reduced the speed to cross T1, T2, and R2. Besides, it can be observed how the speed limit $v_{max,safe}$ was fulfilled when the automation level imposed a conservative driving behavior.

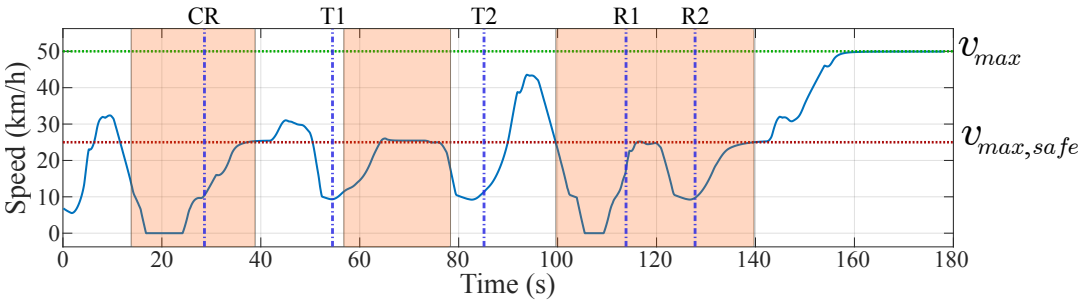


FIGURE 6.16: Speed profile for the complete experiment of the supervisor use case.

Chapter 7

EXPERIMENTAL SETUP

7.1 Introduction

The algorithms described in chapters 4-6 were validated and tested both in real and simulation environments. This chapter describes the architecture of the testing vehicle of Autopia in section 7.2. It also presents the inter-process communication framework used in the ADS, in section 7.3. Finally, section 7.4 depicts the framework implemented to evaluate the ADS algorithms using a software-in-the-loop architecture with realistic simulation scenarios.

7.2 Testing Vehicle

All the autonomous driving algorithms proposed in this Ph.D. thesis were validated in a real vehicle from Autopia's fleet. The testing vehicle is a standard Citröen DS3 modified to include different sensors and actuators; these modifications allow to control the vehicle autonomously. Figure 7.1 shows the hardware components incorporated into the vehicle.

FIGURE 7.1: Testing vehicle components.

7.2.1 Sensors

Regarding the onboard sensors, the vehicle was equipped with a Trimble BX982 Global Navigation Satellite System (GNSS) receiver, which is able to provide a high accuracy position ($\sim 2\ cm$) due to differential real-time kinematic GNSS correction from the reference station; this sensor is configured to provide a 20 Hz data rate.

On the other hand, three LiDAR sensors were placed in the vehicle: two Velodyne VLP-16 on the roof and one Ibeo Lux on the front. The VLP-16 provides information of 16 vertical layers (between -15 deg and $+15$ deg) with a horizontal field of 360 deg and a distance range of 100 m. The Ibeo Lux has 4 vertical layers (between -1.6 deg and $+1.6$ deg and a horizontal view of 85 deg and a range of 200 m.

7.2.2 Actuators

The vehicle is controlled by handling four commands: steering wheel, brake, throttle, and gears. The first two commands are controlled using two actuators mounted on the vehicle, as depicted in Figure 7.1, while the throttle is controlled with an electric signal generated with a digital-to-analog converter. The gears may be configured using a set of relays controlled by a digital module, nevertheless, in this work they are controlled by the automatic gearbox of the vehicle. The reference positions of these commands are generated by the tracking control module.

Regarding the steering wheel, a DC gearbox-motor (12V-24V, 60A) was coupled to the steering wheel shaft, and a low-level PID control was used to control the position of the motor by sending a reference position normalized between $[-1 : 1]$.

With the current hardware configuration, the position of the steering wheel can be set between $[-3\pi \text{ rad} : 3\pi \text{ rad}]$ with a maximum angular speed of 4.712 rad/s.

The brake actuator was another DC gearbox motor (12V-24V, 60A) mounted on the braking pedal in a way that does not interfere with the manual operation of this command. This motor is also controlled by a low-level PID controller that sets the current position of the motor according to a reference position normalized between 0 and 1.

For safety reasons, the actuators are disabled if the reference signal is not refreshed for a period of time of 250 ms. Also, these commands can be deactivated manually at any time by using the actuator switches located on the roof of the vehicle (see image 10 in Figure 7.1).

7.2.3 Other equipment

There are two main processing units installed in the vehicle. The first of them is a CQ77G InCar-PC, with an Intel Core i7-8700T 6x4.0 GHz CPU, an NVIDIA GeForce GTX 1060 GPU, and 32 GB of RAM. This PC is running a Linux-based OS and it is used to execute the autonomous driving algorithm proposed in this work.

The second processing unit is an Nvidia Drive PX2 platform, an embedded system suited for sensor data fusion and processing tasks. This device has two Tegra X2 SoCs with ARM v8 architecture CPUs with 12 cores in total, where each of the SoCs has four ARM-A57 and two Denver cores and 8 GB LPDDR4 memory, and an integrated GPU with Pascal architecture. This unit is used for running the perception and motion prediction tasks.

Both devices are displayed on a 10" screen located in the front of the vehicle. This screen offers a developer-oriented interface, where different programs can be launched and debugged.

The vehicle is also equipped with different modules that allow exchanging information with other vehicles and with the infrastructure. The communication modules in the vehicle are a Commsignia roadside unit ITS-RS4 & an onboard unit ITS-OB4. These devices are compatible with IEEE 802.11p and offer a high-level V2X software development kit, which provides basic services such as Cooperative Awareness and Decentralized Environmental Notification; and day-one applications such as Forward Collision Warning and Lane Change Warning.

The last component is an Android-based 10" tablet mounted in the front of the vehicle to be used as a final user HMI. The user interface described in section 6.7 is executed in this device.

7.2.4 Driver monitoring system

The implementation of a driver monitoring system was out of the scope of this work, but in order to implement a functional human-in-the-loop architecture, the ADS was designed to interact with a third-party monitoring system, which was developed by the company ROVIMATICA [163]. This DMS is able to estimate the drowsiness of the human pilot on board using a near-infrared camera and processing the imagery data with a CNN that receives as input the head pose and eye blinking/yawning rates. Figure 7.2 shows the status of the DMS at different moments during a driving experiment. In Figure 7.2a the driver was focused on the driving task and the status is ok (green circle at the bottom of the figure); conversely, in Figure 7.2b the driver was not engaged in the driving task, and the alarm of the DMS was activated (red circle at the bottom of the figure).

(A) (B)

FIGURE 7.2: DMS user interface. (A) Involved driver. (B) Asleep driver.

7.3 Inter-Process Communication

This section describes how the multiple processes running in multiple devices on the testing vehicle communicate with each other efficiently. Even though there are several options to establish an inter-process communication infrastructure, the alternative adopted in the testing platforms of Autopia is the Lightweight Communication and Marshalling (LCM) API [164]. This API provides a set of libraries and tools for inter-process communication in real-time systems, which is based on a publish-subscribe message passing model using UDP multicast. This architecture allows processes to broadcast data across a specific channel designated by a unique name, and subscribe to the channels they require to complete their task. Furthermore, by employing UDP multicast, the bandwidth required for one message transmission is independent of the number of subscribers.

The advantages of LCM include (i) the possibility of defining customized messages accordingly to the requirements of the application instead of adapting the application to predefined messages; (ii) the multi-platform and multi-language compatibility brings flexibility to the system and allows incorporating new modules in different types of platforms; (iii) it provides marshalling capabilities to encode and decode the messages instead of using human-readable strings [165]; and (iv) the API provides different tools to debug the system, such as a logger, a log-player for offline replaying or a spy tool to visualize data in real-time.

Figure 7.3 depicts the interaction between the different modules of the automated vehicle and the LCM API in the system's network. It can be seen that there are three main categories of components connected to the LCM API: Perception, Actuators, and ADS. There are also three other modules that do not belong to any category: HMI, Motion prediction, and V2X communications. A brief description of the different modules is presented below.

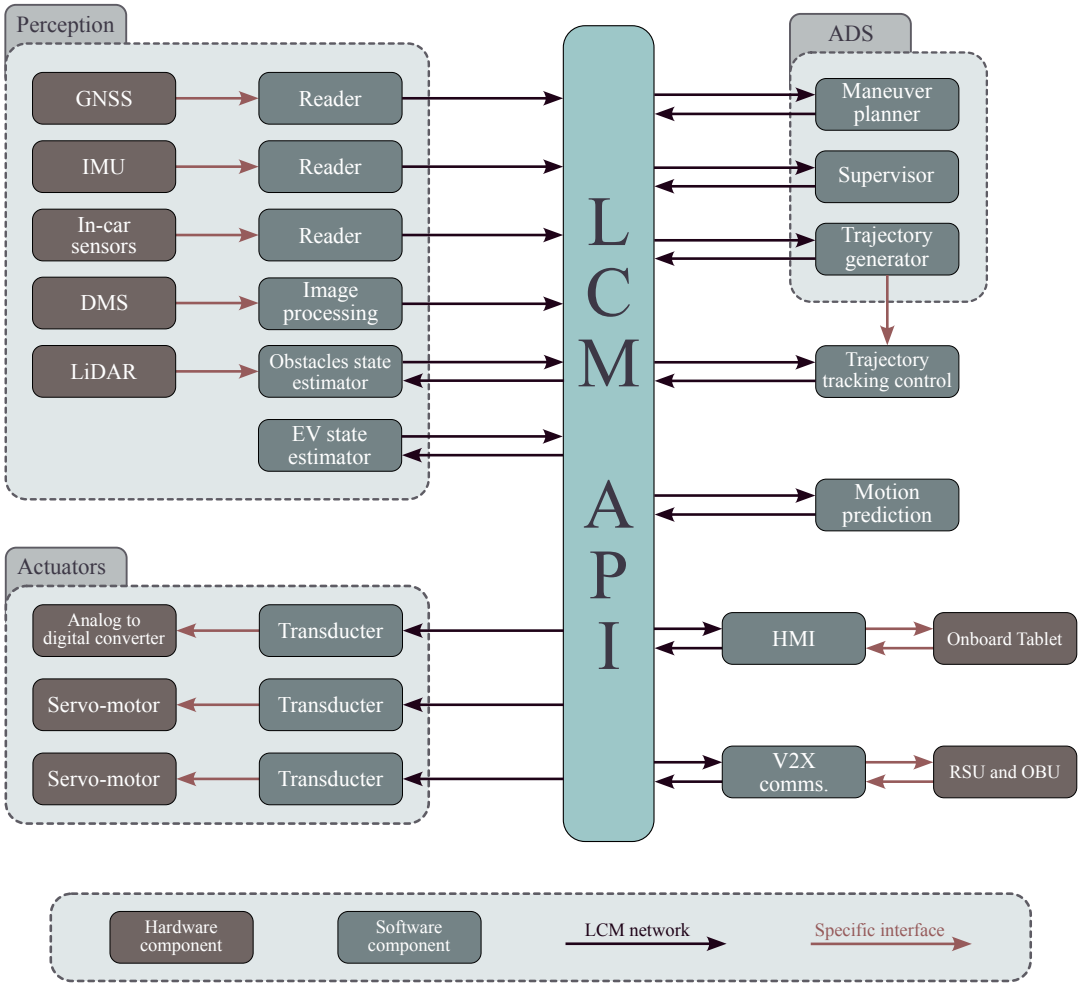


FIGURE 7.3: LCM network implemented in the testing vehicle.

7.3.1 Perception modules

Each of the sensors described in section 7.2.1 sends its information using a specific interface to a reader module in charge of creating an LCM message and publishing it to the LCM network using a specific-sensor channel and predefined data rate. Any module that needs the information of a sensor must subscribe to the corresponding LCM channel. In the case of the DMS sensor, the image processing module implements the CNN described in section 7.2.4 and publishes a boolean variable indicating the status of the driver. In-car sensors refer to the set of proprioceptive sensors included in the vehicle by default, which measure different variables like the gearbox status, the brake system, and the speed of the vehicle, among others; Their data is transmitted via CAN bus.

The obstacles state estimator uses the LiDARs information and the current state of the ego-vehicle to estimate the status of the obstacles present in the scene; the description of this module is presented in [166]. The output of this module is a finite set of obstacles with their current state and approximate size.

Finally, the EV state estimator fuses information from the GNSS and the onboard sensors to obtain a more reliable state using a Kalman filter strategy.

Table 7.1 depicts the LCM messages regarding the perception modules and their data rates.

TABLE 7.1: LCM messages of perception modules.

LCM channel	Data rate	Description
CAN_CAR_GEARBOX	100 *Hz*	Contains information about the status of the gearbox such as the driving mode or the current gear.
CAN_CAR_CONSUMPTION	10 *Hz*	Contains the current fuel consumption level of the vehicle.
CAN_CAR_STEERING	100 *Hz*	Contains the data about the state of the steering wheel, namely its current angle, speed, or torque, among others.
CAN_CAR_BRAKE_ABS	100 *Hz*	Contains the pressure applied to the braking system, the yaw rate, and the lateral acceleration of the vehicle.
CAN_CAR_CABIN	20 *Hz*	Contains the vehicle ignition state and a binary state of the braking pedal.
CAN_CAR_ENGINE	100 *Hz*	Contains the status of the engine like RPM or current torque.
CAN_CAR_SPEED	25 *Hz*	Contains the longitudinal speed and acceleration of the vehicle.
GNSS_DATA	20 *Hz*	Contains data regarding the GNSS like the latitude, longitude, UTM coordinates, number of satellites, and quality of service, among others.
DMS_STATUS	1 *Hz*	Contains two binary variables, one indicating the current status of the human driver, and the other indicating the reliability of the measure.
EV_STATE	20 *Hz*	Contains the current state of the ego-vehicle, position, heading, speed, and acceleration.
OBSTACLES_STATE	12.5 *Hz*	Contains a set of obstacles, each of them described with a pose and a bounding box.

7.3.2 Actuators modules

Each actuator module described in section 7.2.2 is controlled with an LCM channel and a normalized reference value. In order to achieve this, there are three low-level software modules subscribed to the corresponding LCM channel which decode the data and set a goal position (or voltage in the case of the throttle) to the corresponding actuator. The messages to control the actuators are sent by the trajectory tracking control module at a constant rate. Table 7.2 shows the LCM messages associated with each actuator.

7.3.3 ADS modules

The ADS is composed of the three modules described in previous chapters: the maneuver planner, the trajectory generator, and the supervisor.

The maneuver planner subscribes to different LCM channels of the perception modules such as the state of the EV, or the state of the obstacles, but also to the HMI messages regarding the global route. It publishes the global route when requested or in case an alternative route is needed.

TABLE 7.2: LCM messages of actuator modules.

LCM channel	Data rate	Description
CONTROL_THROTTLE	20 Hz	Contains a normalized value between 0 and 1, being 0 no throttle at all and 1 full throttle.
CONTROL_BRAKE	20 Hz	Contains a normalized value between 0 and 1, being 0 no braking at all and 1 full braking.
CONTROL_STEERING	20 Hz	Contains a normalized value between -1 and 1, being -1 full steering to the right and 1 full steering to the left.

The trajectory generator also uses the perception information and publishes the reference trajectory τ_{ref} so that it can be displayed by the HMI and used by the motion prediction module.

The supervisor uses data from the DMS and the HMI and publishes the status of the traded control variables such as the current level of driving automation, the requested involvement level, or different warnings in case a complex scenario arises or the driver is not sufficiently involved.

Table 7.3 summarizes the LCM messages published by the ADS modules. None of these messages is published at a fixed data rate, so this field is not included in the table.

TABLE 7.3: LCM messages of ADS modules.

LCM channel	Description
NEW_GLOBAL_ROUTE	Contains a set of the global route lanelets IDs
TRAJECTORY_REFERENCE	Contains information regarding the current trajectory like the path coordinates, the speed, and the acceleration.
TRADED_CONTROL_STATUS	Contains the status of the traded control variables such as the LoDA level, the scene complexity, the RiL, and the driver involvement level.
HMI_WARNINGS	Contains a set of binary values that activates different warnings on the HMI, it includes GNSS poor quality, a complex area found, or low involvement level from the human.

7.3.4 Other modules

The four remaining modules in the system are the trajectory tracking control, the HMI, the motion prediction, and the V2X communications.

The trajectory tracking control receives the reference trajectory through a specific interface. It needs the information on the EV state in order to compute the proper commands to track the reference trajectory properly.

The HMI subscribes to different LCM perception channels to display information about the driving scene. It also subscribes to the traded control LCM channel to display the required involvement level and to the warnings channel to alert the human pilot when necessary. It publishes different requests to the ADS, like computing a

new global route, switching the driving mode, performing a safe stop maneuver, or canceling an overtaking maneuver.

The motion prediction module creates the prediction grids $\mathcal{R}_{OG}$ using data from the obstacles, the state of the EV, and the current trajectory; a detailed architecture of this module can be found in [167]. These grids are published into the LCM network.

Finally, the V2X communications module publishes messages regarding the data received from other traffic agents using Collective Perception Message (CPM) and Cooperative Awareness Message (CAM) standards.

Table 7.4 summarizes the LCM messages published by the mentioned modules.

TABLE 7.4: LCM messages of independent modules.

LCM channel	**Description**
`HMI_NEW_ROUTE_REQUEST`	Contains the coordinates of the destination point selected in the HMI.
`HMI_DRIVING_MODE_REQUEST`	Contains a binary value to establish the desired driving mode: 1 is autonomous mode, 0 is manual mode .
`HMI_SAFE_STOP_REQUEST`	Contains a binary value where 1 indicates that a safe-stop maneuver is requested, and 0 indicates that autonomous driving may be resumed from the safe-stop mode.
`HMI_CANCEL_OVERTAKING_REQUEST`	Contains a binary value where 1 indicates that the current overtaking maneuver must be canceled by the ADS.
`MOTION_PREDICTION`	Contains the set of grids with the predicted positions of the obstacles present on the current scene in a defined time horizon.
`CMS_CAM`	Contains the current state of other vehicles driving on the scene received via V2V communications.
`CMS_CPM`	Contains the perceived obstacles by other traffic agents, received via V2I or V2V communications.

7.4 Simulation Environment

This section describes the software architecture implemented to facilitate the interaction between the high-level software for AVs and a realistic state-of-the-art simulator. The number of driving situations and environment variables makes validation almost intractable with real vehicles on open roads, and testing reproducibility can only be achieved via simulation. A robust simulation framework allows to test the control algorithms of AVs safely in complex scenarios and to improve wrong behaviors when specific conditions are met.

7.4.1 Simulator description

The simulation software used in this thesis is SCANeR Studio by AV Simulation [168], which is a software suite for automotive simulation, addressing both testing and driving for AVs, human-machine interactions or driving assistance systems. This tool provides all the necessary modules to build a realistic virtual world: road environment, vehicle dynamics, traffic, sensors, real or virtual drivers, headlights, weather conditions, and scenario customization. Figure 7.4 shows the physical setup of Autopia's simulator.

FIGURE 7.4: Physical setup of the simulator.

7.4.2 Vehicle model

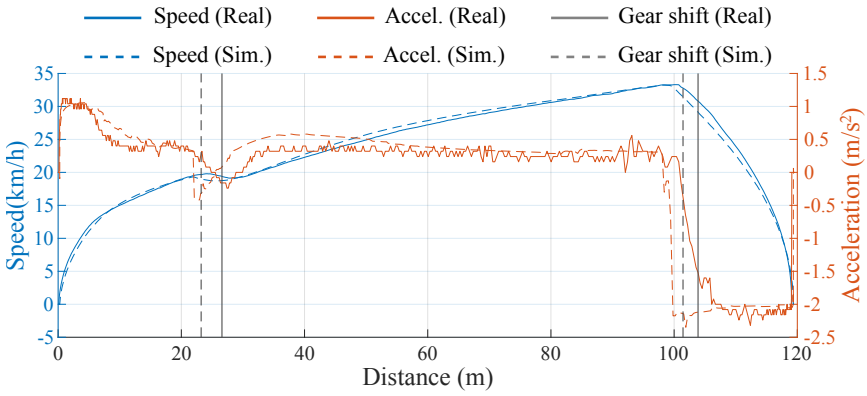
SCANeR Studio provides the possibility of using a high-detail dynamic model of the vehicles named CALLAS model. The CALLAS model is a nonlinear model including vertical, lateral, and longitudinal dynamics. The key features of this model include the engine performance according to the environmental conditions; the transmission includes dynamics related to the clutch, continuous variable transmission, and the torque converter; the suspension uses non-linear dynamics sensible to acceleration and temperature; electronic components like ABS, ESP, cruise control or automatic gearing are included in the model as well.

A CALLAS model was used to replicate the testing vehicle of Autopia on the simulation software, which includes a detailed 3D model of the real vehicle, and moreover, customized dynamics parameters like the torque curves of the engine, the gear-shifting policy, the speed of the steering wheel or the brake pressure. Figure 7.5 shows both the real vehicle and the simulation twin.

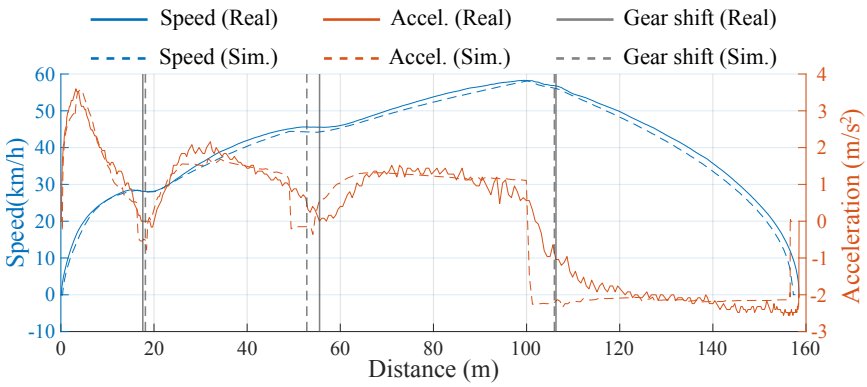
In order to verify the performance of the CALLAS model when compared to the real vehicle, a testing experiment was proposed where different throttle reference values were set to each vehicle during 100 m and then a constant brake reference was applied until the vehicle completely stops. The longitudinal speeds and accelerations of both vehicles after completing two experiments are shown in Figure 7.6. During the first experiment (Figure 7.6a) a 30% throttle reference was applied, followed by a 30% braking reference. In the second experiment (Figure 7.6b), a 50% throttle reference was set during the first 100 m followed by a 30% brake reference.

(A) (B)

FIGURE 7.5: (A) Real and (B) simulated testing vehicles.



(A)



(B)

FIGURE 7.6: Longitudinal dynamics comparison between the real vehicle and the CALLAS model. (A) 30% throttle command. (B) 50% throttle command.

The figures show a very similar performance between the real and the simulated vehicle, reaching an almost identical speed at 100 m and having a comparable deceleration profile. It can be observed that both vehicles performed gear shifting at similar moments during the experiment (marked with vertical grey lines in the figures), being the largest error the first gear shifting during the first experiment, where the simulated vehicle changed the gear 3.4 m before the real one. During the gear-shifting process, the real vehicle showed smoother acceleration curves.

To compare the steering wheel behavior of the real and simulated vehicle, different reference positions were set over time when the vehicles were stopped. The

resulting steering wheel angles for both vehicles are plotted in Figure 7.7. In this case, the steering wheel positions were almost identical, with a maximum error of 0.12 rad in very specific moments.

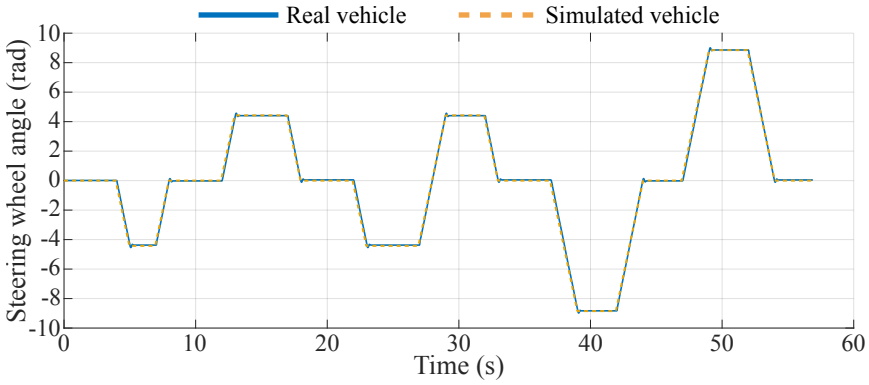


FIGURE 7.7: Testing vehicle components.

7.4.3 Software-in-the-loop architecture

SCANeR Studio includes an API that allows users to personalize their interactions with it by creating their own modules that can access and modify data from a running simulation. This API also enables the developer to control vehicles and pedestrians, change the scenario's parameters, and retrieve sensory data. The network protocol to manage this information notifies the state changes of the elements involved in the simulation and allows sending messages to individual modules in order to control them. One advantage of this technology is that the users can create their own modules in any programming language that can load the given dynamic libraries, in order to transmit and receive the proper network signals.

This modular architecture was used to implement a customized middleware to handle the simulated vehicle as if it were the real one, so the same ADS software could be used to control both. This software-in-the-loop approach simplified the deployment of the ADS algorithms since the exact same software could be verified in simulation before validating it on the real vehicle.

Two custom-built SCANeR modules were developed with the aim of implementing the proposed architecture: the first for handling the perception and the second for handling the actuators of the simulated vehicle. The perception process receives messages from SCANeR Studio and transmits them through LCM periodically, with the same data rates as the modules in the real vehicle. This module transmits information about the vehicle state, the built-in sensors, the steering wheel state, the GNSS location, and the state of the surrounding obstacles. The actuator process, on the other hand, receives messages from the LCM API and sends them to the SCANeR Studio network in order to control the vehicle. This module also emulates the behavior of the actuators implemented in the real vehicle to handle the steering wheel and the breaking pedal. This architecture is depicted in Figure 7.8.

An example of a software-in-the-loop simulation execution is presented in Figure 7.9. The running simulation in SCANeR studio is shown in Figure 7.9a, while the status of the ADS software is depicted in Figure 7.9b.

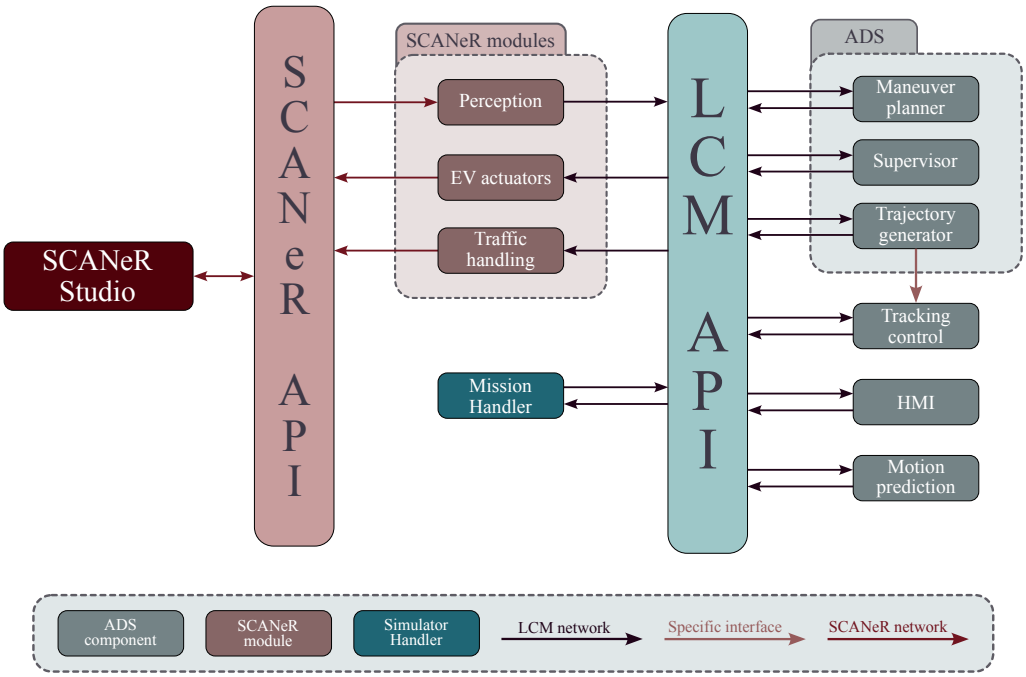


FIGURE 7.8: Software architecture to interact with the simulator.

(A)

(B)

FIGURE 7.9: Software-in-the-loop simulation in a double-lane intersection. (A) Simulator status. (B) ADS software status.

Mission handler

In case the developer needs to evaluate the ADS algorithms using realistic traffic scenarios based on datasets, the simulation can be handled from a Mission Handler module that reads the information from previously selected datasets and generates the virtual scene in the simulator. Then the ADS algorithms interact with the simulator by reading the state of the vehicles and sending commands to the EV using the LCM API.

Using this approach, it is possible to test the complete system in a simulated scenario where, even though the virtual vehicles are spawned from the original positions of the dataset and acquire the same planned behavior, they still have the ability to react to the ego-vehicle's actions during simulation.

The mission handler is in charge of sending the appropriate commands to a traffic handling module (see Figure 7.8), which receives commands through the LCM API and sets different goals for the OVs e.g. driving speed or lane-change commands during the simulation. Thus, the simulated scene recreates the real one as accurately as possible by reading a simulation script that contains a set of commands synchronized with the dataset vehicle evolution while the simulation is running.

The simulation script language used by the mission handler was developed in this Ph.D. thesis, as well as the algorithm to verify its integrity and execute the commands by sending the corresponding actions to SCANeR Studio. This scripting language contains different actions, including creating a vehicle in a specific position, setting a traveling speed, forcing a lane-change maneuver, or configuring a predetermined itinerary for an OV. Table 7.5 details each of the possible commands included in a simulation script.

The dataset-based simulated scenarios of section 5.7.2, which were used to validate the speed profile algorithm in roundabouts, as well as the scenario of the demonstrator depicted in section 8.4.3, were implemented using the mission handler described above to replicate the behavior of the real vehicles.

TABLE 7.5: Set of functions to handle a realistic simulation based on scripts.

Function name	Parameters	Description
`set_ego_destination`	`<UTM_east>`, `<UTM_north>`	This function will set the destination point for the ego-vehicle. It is mandatory for all script files and it must be the first instruction on the file since the destination is required in order to start the mission.
`wait_until`	`<time>`	This function will prevent the following instructions from being executed until this time is reached on the simulation. The parameter `<time>` is a float.
`create_vehicle`	`<name>`, `<id>`, `<pos_x>`, `<pos_y>`, `<heading>`, `<speed>` , `<sign>`	This function will create a new vehicle in the scene with a given state. The parameter `name` defines the type of vehicle that will be created (for example "Citroen C4"). The coordinates for the vehicles are referenced to the SCANeR scenario frame. ID 0 is reserved for the ego-vehicle. The heading's units are degrees and the speed's units are km/h. The `<sign>` parameter is an optional boolean value that decides if the obstacle must obey traffic signs or discard them.
`set_vehicle_max_speed`	`<id>`, `<goal_speed>`	This function will set the maximum speed for a given vehicle. The speed is a value in km/h.
`set_vehicle_lane_change`	`<id>`, `<shift_value>`, `<time_to_shift>`	This function will perform a lane change for a given vehicle. The `shift_value` will perform a left-lane change when set to 1 and a right-lane change when set to -1. The `time_to_shift` parameter determines the maximum time to perform the maneuver.
`set_vehicle_itinerary`	`<id>`, `<add>`, `<n_directions>`, `<direction[1]>`, `<direction[2]>`, ...	This function will set the itinerary of a vehicle. A complete itinerary will be formed by an array of intersection IDs and the number of the desired exit at each intersection. The exit number starts at 0 being the right-most exit of the intersection. The parameter 'add' is set to 1 to append the itinerary, and 0 to overwrite the existing one.

Chapter 8

DEMONSTRATORS

8.1 Introduction

Being able to run the algorithms in real-time and on real scenarios was one of the objectives of this Ph.D. thesis since it evidences how the system is able to handle real-life problems such as perception uncertainty, limited computation capability, and hardware-software interactions, among others. This chapter depicts seven different demonstrators conducted in the real vehicle in different types of urban scenarios, which are introduced in section 8.2. Each demonstrator highlights different capabilities of the ADS, and, together, they illustrate the scope of this Ph.D. thesis. The demonstrators are part of Autopia's results in two European projects: Prystine (2018-2021) [169] (see section 8.3) and New Control (2019-2023) [170] (see section 8.4). A set of videos for the demonstrators are listed in section 8.5.

8.2 Demonstrators Description

A brief description of each demonstrator is presented next.

- **Urban intersection:** In this demonstrator the EV and two other vehicles are approaching an intersection from different roads. The EV has to perform a left turn at the intersection and manage a pedestrian crossing located after the intersection, where a pedestrian will cross when the EV approaches. The results of this demonstrator show the ability of the maneuver planner to handle different types of REs, and also how the trajectory generator properly adapts to the

strategic restrictions, by stopping before the yielding lines while maintaining smooth driving at all times. Figure 8.1 shows the experimental setup for the demonstrator, plotting the initial positions of the vehicles, the pedestrian, and the REs on the road.

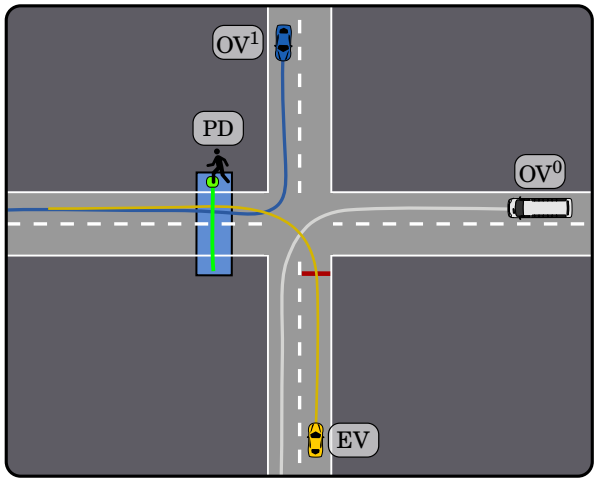


FIGURE 8.1: Experimental setup of the urban intersection demonstrator.

- **Overtaking in a two-way street:** In this demonstrator, the EV has to perform an overtaking maneuver, but on the opposite lane there is an oncoming vehicle, which means that the EV has to wait until the opposite lane is free before overtaking the leader vehicle on its lane. During the experiment, it can be observed how the maneuver planner selects the most effective lane while taking into account the safety gaps. The experimental setup of this demonstrator is depicted in Figure 8.2.

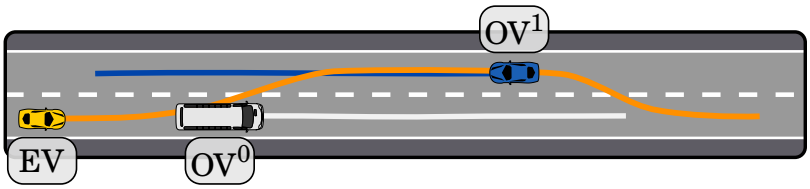


FIGURE 8.2: Experimental setup of the overtaking in a two-way street demonstrator.

- **Cooperative re-routing on blocked road:** In this demonstrator, depicted in Figure 8.3, the EV is driving on a two-way road and suddenly discovers that all the lanes on the current road are blocked by some roadworks. The maneuver planner finds an alternative global route to reach the destination point. The new route is prompted to the human driver through the HMI, and it has to be manually accepted in order to be considered by the ADS. This demonstrator highlights the human-machine interaction skills of the system, and it also presents an analysis of the dynamic variables like speeds and accelerations during the experiment.

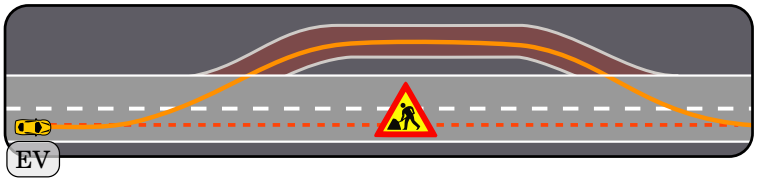


FIGURE 8.3: Experiment setup of the road-works demonstrator.

- **Human-in-the-loop automated driving:** This demonstrator presents a scenario where the EV is driving in autonomous mode while the DMS is estimating the drowsiness level of the human driver (see Figure 8.4). At some point during the automated driving, the human driver falls asleep but the navigation continues normally. After some time, a complex scenario is found and a safe-stop maneuver is performed due to the driver's unavailability. Autonomous driving is resumed manually by the human once awake. This demonstrator illustrates the human-in-the-loop capabilities of the system since the supervisor is able to find the most pertinent level of driving automation according to the driving scene and safely handle a complex situation while the human is not ready to take control of the vehicle.

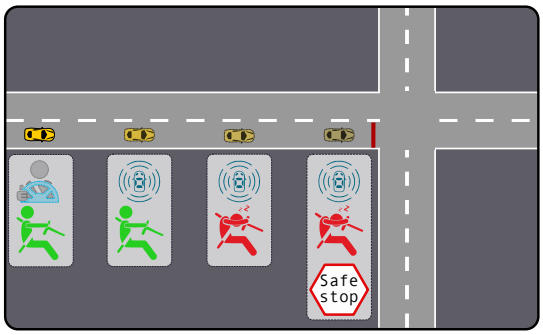


FIGURE 8.4: Experimental setup of the human-in-the-loop automated driving demonstrator.

- **Unexpected behavior of an OV in a roundabout:** In this demonstrator, an OV suddenly stops while driving inside a roundabout. The EV, which has merged into the roundabout after that particular OV, avoids it using the inner lane of the roundabout safely, to take the intended exit afterward. This demonstrator emphasizes the trajectory generation and evaluation process on a complex scenario and shows the resulting path and speed profiles during the complete experiment. The experimental setup is presented in Figure 8.5.

- **Merging into a roundabout with traffic:** The EV has to merge into a roundabout between two OVs in a safe, yet comfortable manner. The speed profile generation process while merging into the roundabout is presented, as well as the dynamic variables and the gaps with respect to the OVs for the complete experiment. This demonstrator shows the ability of the trajectory generator to efficiently adapt the speed profile of the EV during difficult situations. The experimental setup is presented in Figure 8.6.

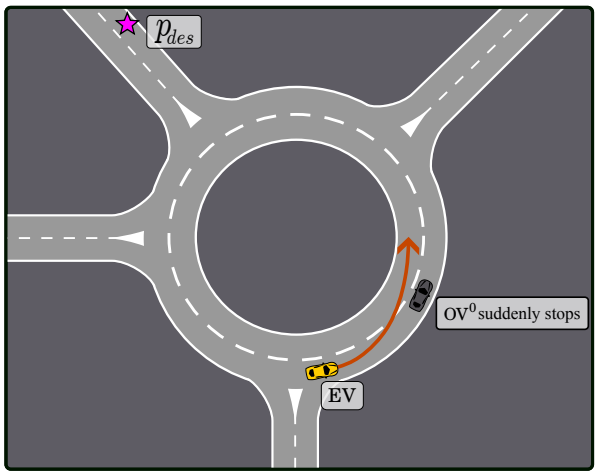


FIGURE 8.5: Experimental setup of the unexpected behavior on a roundabout demonstrator.

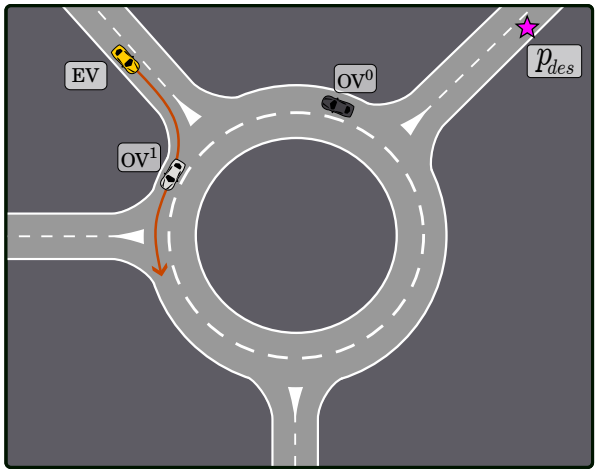


FIGURE 8.6: Complete path of the merging into a roundabout with traffic demonstrator.

- **Integration with IAMP:** This demonstrator shows the performance comparison of the trajectory generator when using two motion prediction strategies: Constant Speed Motion Prediction (CSMP) and Interaction Aware Motion Prediction (IAMP) [167]. This is the only demonstrator tested on simulation, nevertheless, it is a dataset-based realistic scenario. The layout of the driving scenario of this demonstrator is presented in Figure 8.7.

8.2.1 Configuration parameters

The default configuration parameters used for these demonstrators are presented in Table 8.1. In the cases where a specific parameter was modified for any of the demonstrators, it is mentioned in the description of the corresponding experiment.

The table shows different parameters regarding the maneuver planner, the trajectory generator, and the supervisor modules. For the maneuver planner, the parameters involve the utility weights for the corridor selection and the parameters regarding the traffic regulation state. Regarding the trajectory generator, the parameters

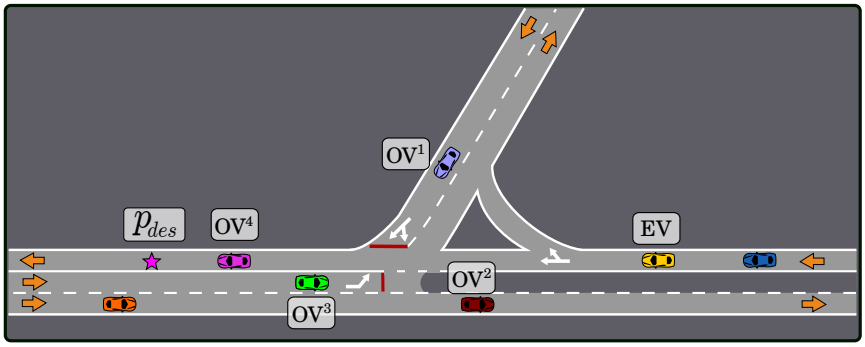


FIGURE 8.7: Experimental setup for T-junction scenario.

TABLE 8.1: Default configuration parameters of the ADS for the demonstrators.

Module	Parameter	Value	Unit
Maneuver planner	$\mathscr{U}_{op}$	6	–
	$\mathscr{U}_{fe}$	10	–
	v_n	1	–
	$\beta_{pos,rdb}$	$\{0, -2, -5, -7\}$	–
	d_{gc}	9	m
	d_{om}	10	m
	t_{o1}	3	s
	t_{o2}	0.5	s
Trajectory generator	N_P	800	–
	N_{SP}	4	–
	v_{max}	40	km/h
	$\mathcal{L}$	$\{1.1, 1.4, 1.6\}$	m/s^2
	B_{max}	4	m/s^2
	$B_{max,OV}$	1.5	m/s^2
	$\mathcal{W}$	$\{0.4, 0.9, 0.7, 0.99\}$	–
	d_c	5	m
	λ	1.38	–
	v_{sl}	7	km/h
	v_{aw}	10	km/h
	t_{ur}	0.2	s
	d_{sl}	14	m
	d_y	5	m
	δ_{nl}	15	m
Supervisor	$\mathscr{C}_{comp}$	25%	–
	$\mathscr{C}_{med}$	45%	–
	$\mathscr{C}_{simp}$	65%	–
	$\mathscr{C}_{bas}$	85%	–
	$\wp_{min}$	$\{0.5, 0.5, 0.6, 0.2\}$	–
	t_{CL}	15	s
	t_{WR}	3	s
	t_{CS}	15	s
	N_{su}	5	–

include the number of trajectory candidates, the speed and acceleration limits, the weights of the merit function, and the parameters needed by the inter-distance and merging models. Finally, for the supervisor module, the configuration parameters set the scene complexity thresholds, the suitability limits for the decision variables, and the time limits for the LoDA level updating. The safe parameters (mentioned in Table 6.5) used when $\mathscr{A} = \texttt{LoDA2}$, are set with the same values as the default parameters, except for the human-in-the-loop automated driving demonstrator.

Vehicle parameters for planning

Even though a real vehicle is modeled using multiple parameters of different natures, only those listed in Table 8.2 were used in the motion planning algorithm.

TABLE 8.2: Vehicle parameters used in the planning task.

Parameter	Value	Units
rear-axis-to-front distance	3.3	m
rear-axis-to-back distance	0.6	m
Wheelbase	2.464	m
Steering transmission ratio	16.36	$-$
Maximum steering wheel speed	4.712	rad/s
Maximum curvature	0.23	m^{-1}

The rear-axis-to-front distance defines the separation between the rear axis of the vehicle (which is used as the reference frame of the EV) and the front of the vehicle; this parameter is taken into account when computing the occupancy polygon of a path candidate, when calculating the gaps with the OVs, or when computing the speed profiles. The rear-axis-to-back distance is similar to the previous parameter, but uses the rear of the vehicle instead; this parameter is involved in the calculation of the gaps with the OVs. The wheelbase is used to compute the occupancy polygon of a path candidate. The steering transmission ratio gives the relation between the steering wheel angle and the front wheels; it is used together with the maximum steering wheel speed in the limit-speed profile computation. The maximum curvature defines the sharpest turning radio the EV can reach and it is used to discard invalid paths.

8.3 Prystine Project Demonstrators

One of the goals of Prystine project was to be able to handle a human-in-the-loop architecture with a dynamic assignation of the LoDA level. The first four demonstrators presented in this section are embedded in this European project. Three of these demonstrators were carried out on the proving grounds of Applus+ IDIADA, in Tarragona, Spain. The fourth demonstrator was carried out on the testing track of the Centre for Automation and Robotics in Madrid, Spain.

8.3.1 Urban intersection

The layout of this demonstrator locates the EV and two other vehicles (OV^0, OV^1) approaching an intersection from different roads. The EV has to cross the intersection, which involves driving over two different REs before reaching the destination.

The first of them (ψ^0) is a right of way RE, and the EV is approaching from its yielding lane, so OV^0 has higher priority. The second RE (ψ^1) is a pedestrian crossing with a pedestrian (PD) waiting in the crossing area. A snapshot of the real experiment on the proving grounds is shown in Figure 8.8.

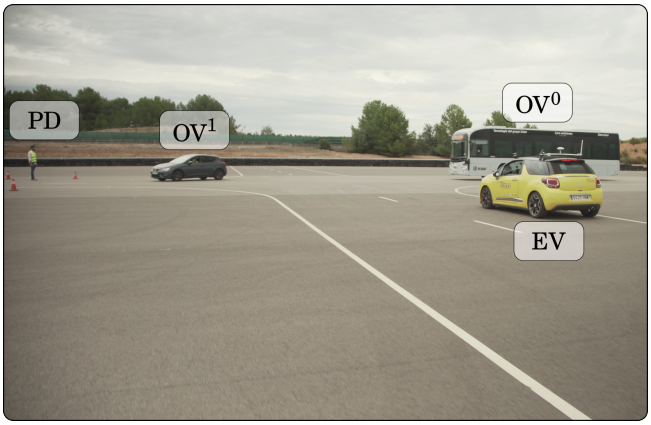


FIGURE 8.8: Traffic status snapshot during the urban intersection demonstrator.

The evolution of the experiment is presented in Figure 8.9 and it is described in detail next:

- Once the autonomous mode was engaged, the maneuver planner immediately detected the right of way RE ψ^0 (see Figure 8.9a), changing the traffic regulation state from $\varsigma_R = \texttt{Go}$ to $\varsigma_R = \texttt{try}$ to face the intersection.
- While driving in the latter state, the trajectory generator did not find a feasible trajectory to merge before OV^0 reached the intersection, so the EV stopped on the yielding line during the interval $t = [6.8\ s, 13\ s]$ (Figure 8.9b).
- The condition c_{tg} was triggered at $t = 13\ s$ because OV^0 did not intersect the EV trajectory any longer, hence $GD^1_\zeta \rightarrow \infty$. At that moment $\varsigma_R = \texttt{Go}$ and $id_{\psi_o} = id^0_\psi$ and, by doing this, this RE was omitted until its YL was completely crossed.
- Since ψ^0 was no longer taken into account, the pedestrian crossing ψ^1 was detected and the traffic regulation state changed to $\varsigma_R = \texttt{Aware}$. This state only lasted one iteration and it was afterward updated to $\varsigma_R = \texttt{Stop}$ because there was a pedestrian located in the crossing area.
- The EV started to move and crossed the yielding line of ψ^0 at $t = 17.1\ s$ (Figure 8.9c) and stopped before the yielding line of ψ^1. This state was maintained until $t = 27.1\ s$, when the pedestrian left the lane and $trf^1_\psi \geq t_{o2}$, triggering the condition c_{yg} (Figure 8.9d).
- Once ψ^1 was crossed, and until the end of the experiment, the traffic regulation status was $\varsigma_R = \texttt{Go}$.

The evolution of the traffic regulation state ς_R and the EV speed during the demonstrator are plotted in Figure 8.10. To provide a higher context to the figure, three events were highlighted using vertical dotted lines: the crossing of the yielding

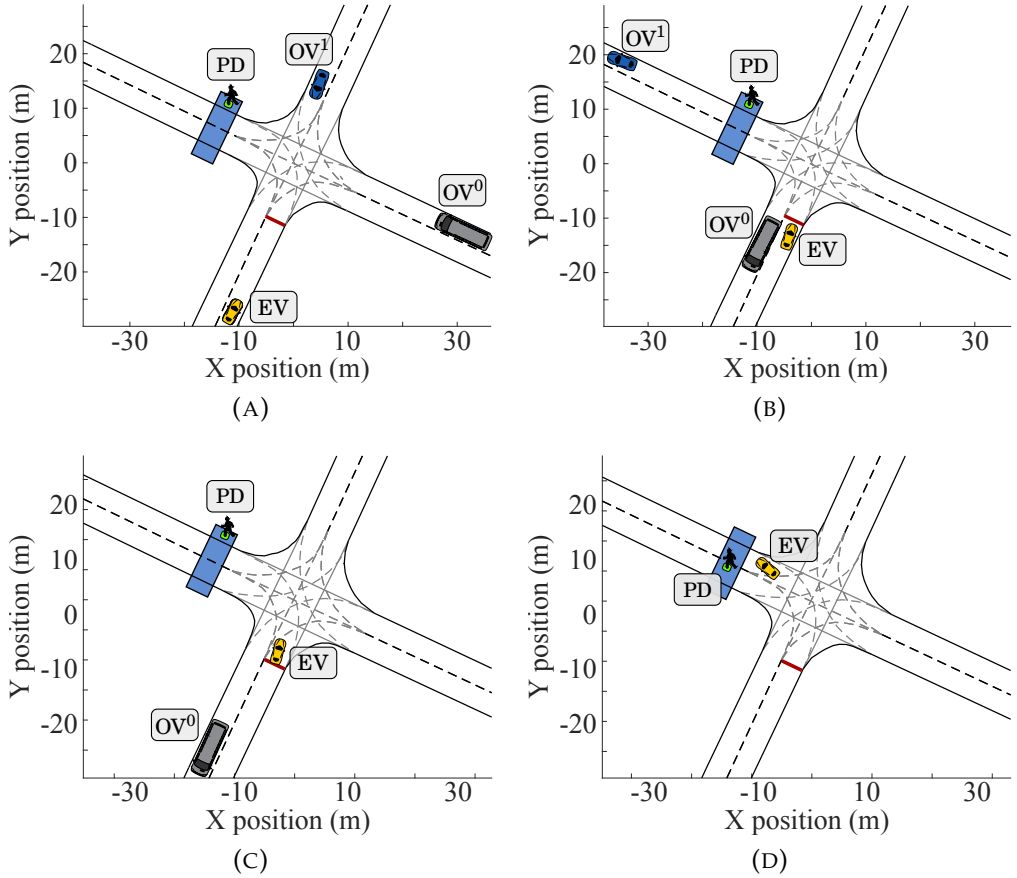


FIGURE 8.9: Status evolution of the urban intersection demonstrator. (A) Traffic status at $t = 1\ s$. (B) Traffic status at $t = 13\ s$ (C) Traffic status at $t = 17.1\ s$ (D) Traffic status at $t = 27.1\ s$.

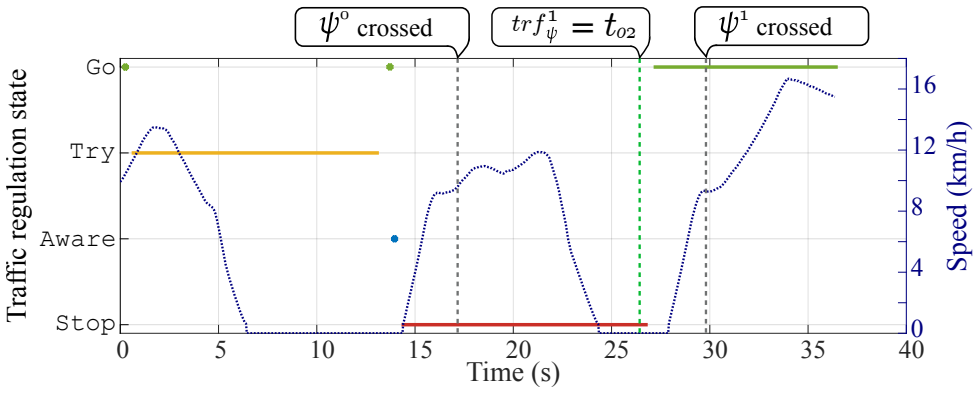


FIGURE 8.10: Traffic regulation state ς_R and speed profile all along the urban intersection demonstrator.

line of ψ^0, the moment when the road on the pedestrian crossing had been free for over t_{o2} seconds after the EV had stopped, and the crossing of the yielding line of ψ^1.

The speed profile shows how the EV started to brake at $t = 2.2\ s$ with the $\varsigma_R = \texttt{Try}$ state, to stop smoothly at the intersection, then, at $t = 14\ s$ it started moving even though the traffic regulation state was $\varsigma_R = \texttt{Stop}$, this is because that state

is regarding the pedestrian crossing, which is still far from the EV. The distance to the yielding lines during the travel is plotted in Figure 8.11, which shows how the EV stopped at 5.7 m and 5.9 m from YL^0_ψ and YL^1_ψ, respectively; which is expected, given that the stop distance parameter was set to $d_y = 5\ m$. At the last section of the experiment, the traffic regulation state is set to $\varsigma_R =$ `Go` as soon as the road on the pedestrian crossing has been free for more than t_{o2} seconds, and the EV is able to drive and reach the destination point.

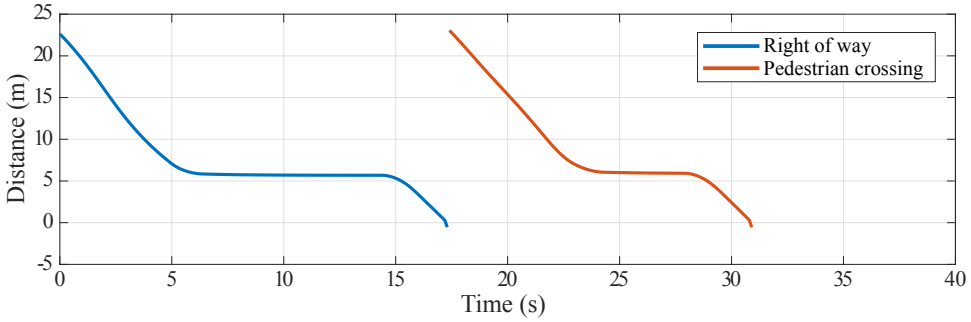


FIGURE 8.11: Distance to the yielding lines of ψ^0 and ψ^1 during the intersection demonstrator.

8.3.2 Overtaking in a two-way street

In this demonstrator the EV has to overtake another vehicle (OV^0) while there is an oncoming vehicle (OV^1) in the opposite lane; hence, the EV has to wait until the opposite lane is free before performing the overtaking maneuver. Figure 8.12 shows the status of the vehicles during the experiment. Notice that in order to increase the perception distance of the EV and, therefore, to be able to detect OV^1 correctly while it is in front of OV^0, the Collective Perception Service was implemented between the EV and OV^0 [166].

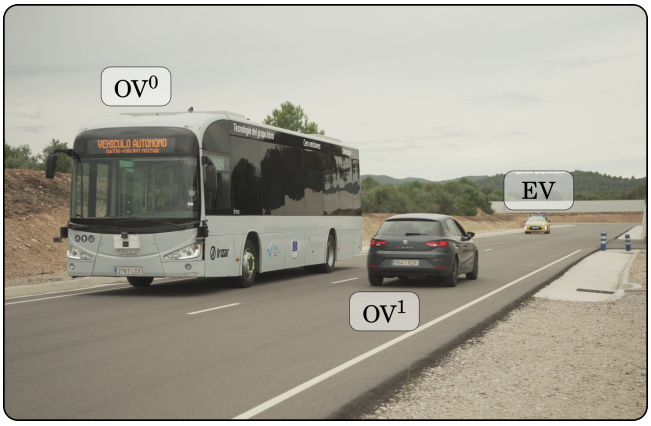


FIGURE 8.12: Traffic status during the overtaking in a two-way street demonstrator.

There are two possible corridors for the EV in this case. The first one is the right corridor (ζ^0), which is also the initial corridor of the EV. The second one is the left corridor (ζ^1), which is located on an opposite-way lane, so the δOP^1_ζ flag is activated.

Figure 8.13 shows the probability assigned to each corridor during the travel, using a color code to represent their status, as follows: blue indicates that $\zeta^q = \zeta^T$, $\zeta^q \in \mathcal{C}$; green means that ζ^q is available, but $\zeta^q \neq \zeta^T$; and grey is referred to the situation where the gaps of ζ^q are not valid at the moment.

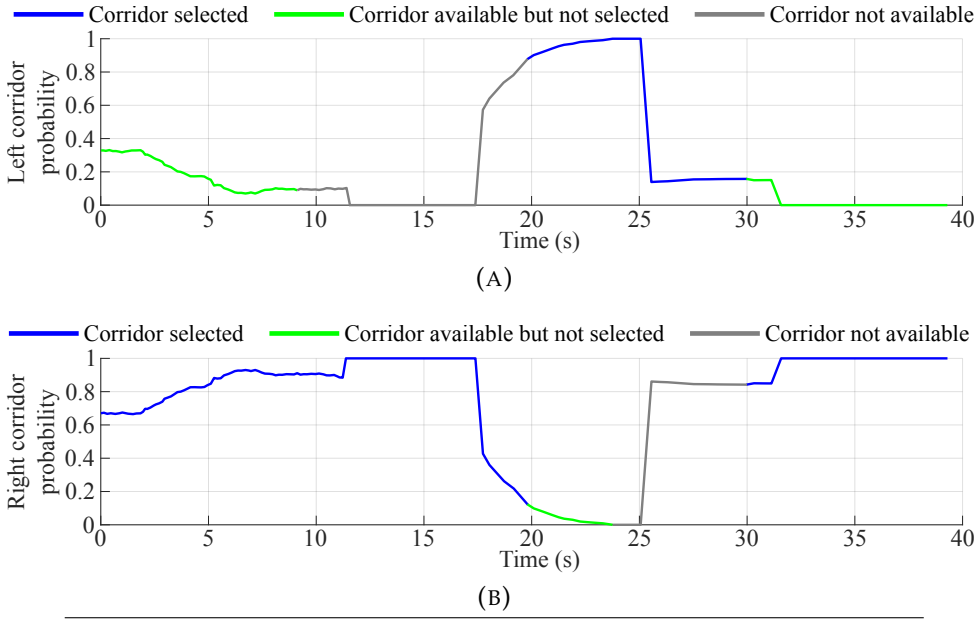


FIGURE 8.13: Probability of $P(\zeta^q = \zeta^T)$ for the (A) left and (B) right corridors during the overtaking demonstrator.

The evolution of the overtaking demonstrator is presented in Figure 8.14. A detailed description is presented next:

- At the beginning of the experiment, the right-hand corridor lane was selected as the target corridor since $P(\zeta^0 = \zeta^T) = 0.66$. At that moment, the gap with respect to OV^1 was larger than 100 m, so it was considered safe, but the value of $P(\zeta^1 = \zeta^T)$ was low.

- At $t = 8$ s (see Figure 8.14a), the gap of ζ^1 was not considered safe anymore and this corridor became unreachable until $t = 19.8$ s (Figure 8.14b), when the target corridor also changed, as $P(\zeta^1 = \zeta^T) > P(\zeta^0 = \zeta^T)$, triggering a lane-change maneuver.

- At the beginning of the overtaking maneuver the HMI prompted the option of canceling the overtaking maneuver (see Figure 8.15), this option was available as long as the gap of ζ^0 was still safe due to the proximity between the EV and OV^0. Figure 8.14c shows the status of the vehicles when the gap of ζ^0 was not valid anymore, and it was not possible to go back.

- When the EV over passed OV^0 at $t = 25.5$ s, the probability of the target corridor changed notably, and $P(\zeta^1 = \zeta^T) < P(\zeta^0 = \zeta^T)$, but the gap acceptance model did not allow to select this corridor yet.

- At $t = 30$ s (Figure 8.14d) the EV started the lane-changing maneuver back to the original lane, in order to complete the overtaking maneuver.

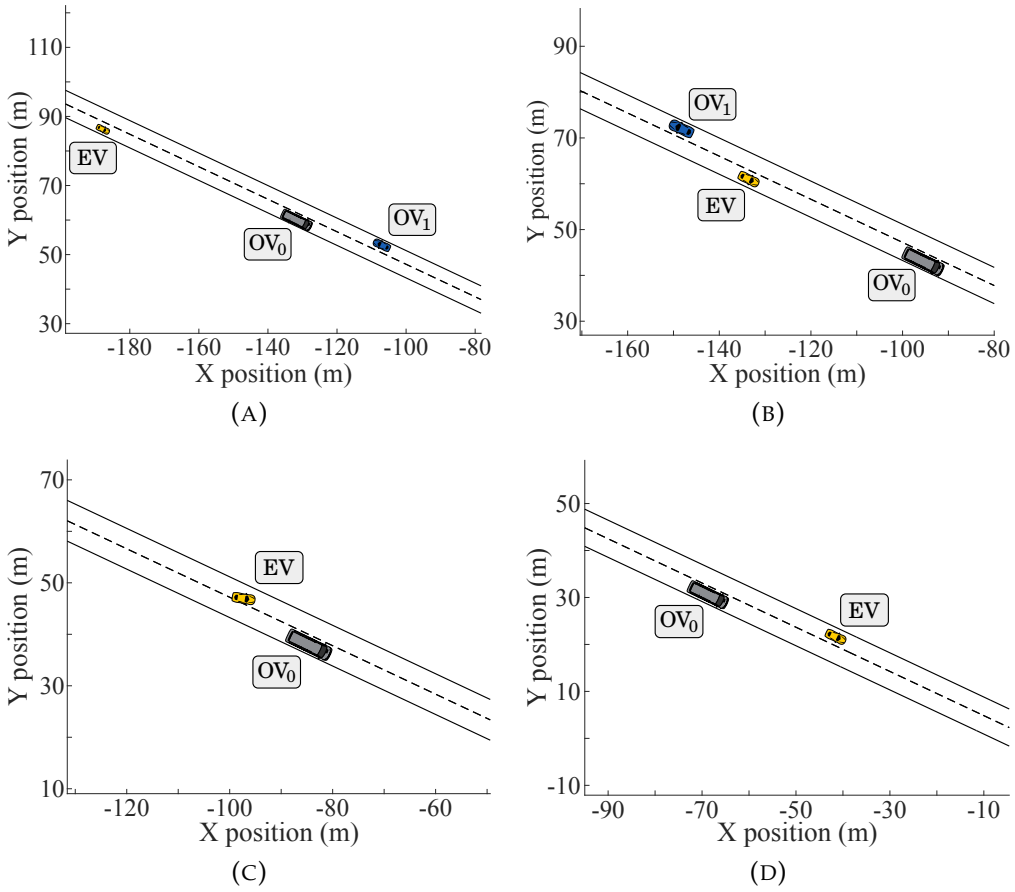


FIGURE 8.14: Time evolution of the overtaking traffic scene. (A) traffic status at $t = 8\ s$. (B) traffic status at $t = 19.5\ s$ (C) traffic status at $t = 23.7\ s$ (D) traffic status at $t = 30\ s$.

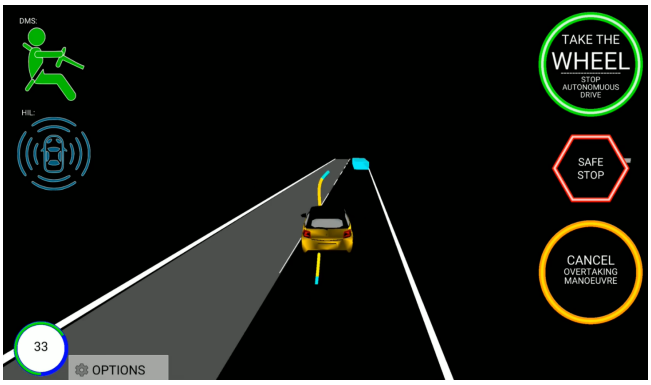


FIGURE 8.15: HMI status during the overtaking maneuver.

The gaps between the EV and OV^0, OV^1 during the experiment are plotted in Figure 8.16. Every positive gap means that the OV is ahead of the EV. For this experiment, OV^0 started 45 m ahead of the EV and, due to the collaborative perception, OV^1 was detected at a distance of 120 m. Four time instants are highlighted in the figure using vertical dotted lines, which match with the four time steps plotted in Figure 8.14. This figure provides accurate information about the behavior of the maneuver planner, as it shows that (i) the opposite lane became unreachable when OV^1

was 78 m away; (ii) OV^1 was detected until it was 17 m behind the EV, and at that moment the opposite lane became available; (iii) the right-most corridor stopped being available when the gap with respect to OV^0 was 23.7 m and (iv) the right-most corridor was considered safe again when the lag gap was $GG^0_\zeta = 31.5$ m.

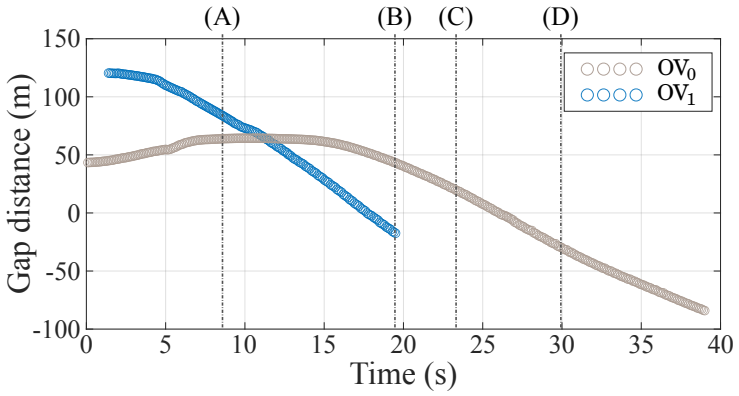


FIGURE 8.16: Gaps with respect to OV^0 and OV^1 during the overtaking demonstrator.

8.3.3 Cooperative re-routing on blocked road demonstrator

In this demonstrator, the EV started driving on a traffic-free two-way road, and then it found that all navigation corridors on the global route were blocked by some roadworks; at that point, the EV finds an obstacle-free navigation corridor available which, although not included on the global route, could also reach the destination point. Figure 8.17 shows the status of the trajectory generation process when the roadworks were detected.

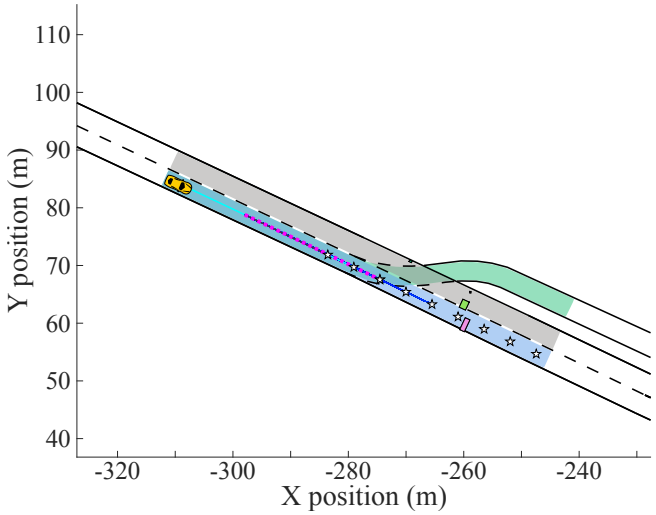


FIGURE 8.17: Trajectory set in the original route of the cooperative re-routing on blocked road demonstrator.

The figure shows two obstacles located on the current and adjacent-left navigation corridors; the current corridor is colored in blue, and the adjacent-left corridor is colored in grey, meaning it was not available for selection. A third corridor, plotted

in green, is plotted in the figure, which was not blocked by any obstacle. When the maneuver planner identified that all corridors on the original route were blocked, it searched for an alternative global route to reach the destination.

In this case, the maneuver planner found an alternative route and proposed it to the human driver through the HMI. The request is depicted in Figure 8.18. The navigation corridors of any alternative route are not considered by the maneuver planner until it is manually confirmed by the human.

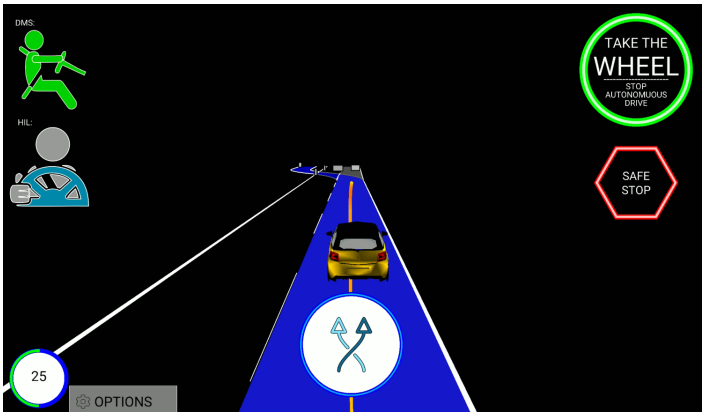
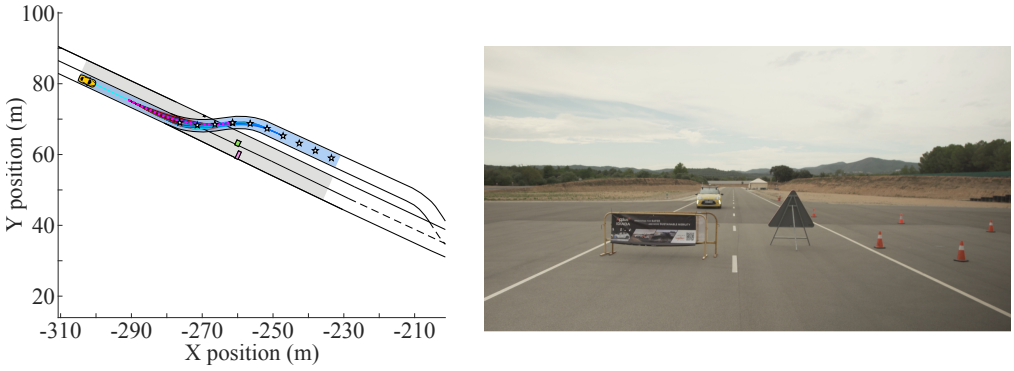


FIGURE 8.18: Alternative route request.

Since the human driver confirmed the alternative route during the demonstrator, the obstacle-free corridor was selected as ζ^T, and the trajectory candidates were created over its waypoints, as shown in Figure 8.19a. The real-life status of the demonstrator at this moment is depicted in Figure 8.19b.



(A) (B)

FIGURE 8.19: New route acceptance. (A) Trajectory set in the new route. (B) Snapshot of the real vehicle during the demonstrator.

The complete path followed by the EV during this demonstrator is depicted in Figure 8.20. It shows how the alternative route was used, avoiding the blocking roadworks, to finally reach the destination point.

The speed and acceleration profiles during the travel are plotted in Figure 8.21. For this specific setup, the maximum speed was reduced to $v_{max} = 25\ km/h$, with the intention of increasing the time window where the new route was proposed to the human driver on the HMI. It can be observed how the EV gets to maximum speed after $t = 12.4\ s$, and how it started to brake at $t = 17.6\ s$ in order to make the left turn on the alternative route in a comfortable way. The lateral acceleration

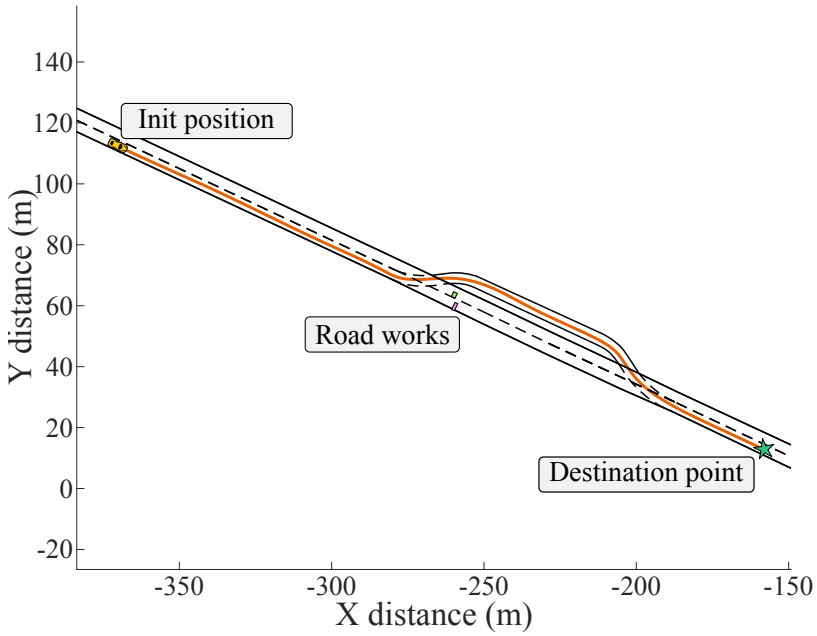


FIGURE 8.20: Complete trajectory for roadworks demonstrator.

exceeded the limit $lat_{\mathcal{L},max}$ during the entrance and exit of the alternative route due to the longitudinal speed error when tracking the reference speed. The longitudinal acceleration shows a smooth curve along the demonstrator, except at $t = 9\ s$ and $t = 24\ s$, when a gear-shifting was performed. Also, it only exceeded the positive limit at the beginning of the experiment due to the error between the reference speed and the real speed of the EV.

8.3.4 Human-in-the-loop automated driving

This demonstrator tested how the autonomous driving system performed using the human-in-the-loop approach in a driving scenario where the drowsiness and focus levels of the driver were constantly being monitored by the system described in section 7.2.4. In this demonstrator, the EV started driving in a rather simplistic scenario, where the human on board simulated falling asleep as the vehicle was in autonomous mode. Some configuration parameters were varied with respect to Table 8.1, the list of the modified parameters and their values for this demonstrator is presented in Table 8.3.

TABLE 8.3: Configuration parameters for the human-in-the-loop automated driving demonstrator.

Parameter	Value	Units
v_{max}	30	km/h
$v_{max,safe}$	20	km/h
$\mathcal{L}_{max}$	$\{1.3\,,\,1.5\,,\,1.0\}$	m/s^2
$\mathcal{L}_{safe,max}$	$\{1.0\,,\,1.0\,,\,0.9\}$	m/s^2
$\mathcal{W}$	$\{0.7\,,\,0.5\,,\,0.5\,,\,0.99\}$	–
$\mathcal{W}_{safe}$	$\{0.7\,,\,0.4\,,\,0.8\,,\,0.4\}$	–

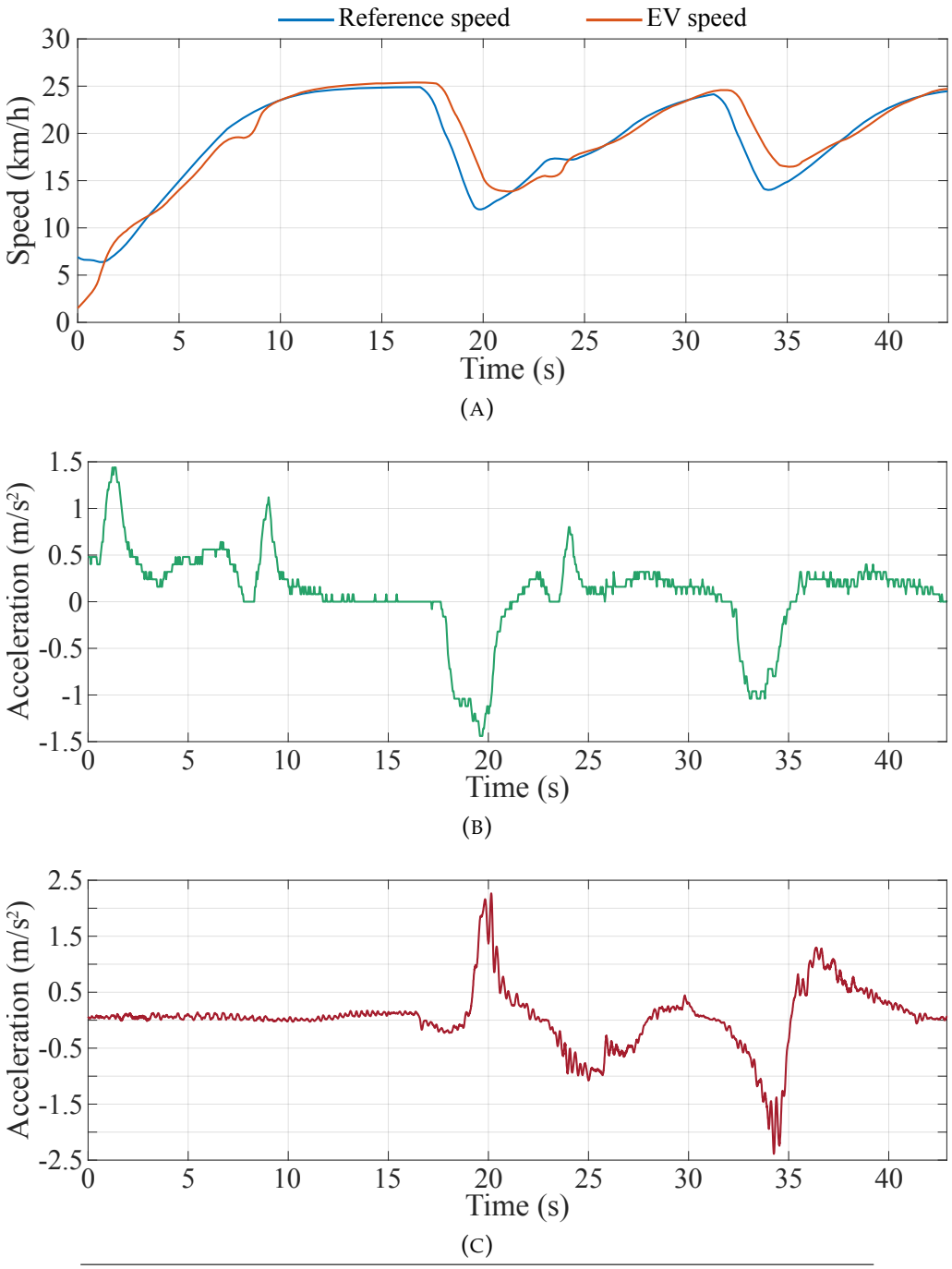


FIGURE 8.21: Dynamic variables for the roadworks demonstrator. (A) Longitudinal speed. (B) Longitudinal acceleration. (C) Lateral acceleration.

All mentioned parameters have to do with the trajectory generator; they are set to have a more conservative driving style when required. It can be observed how the acceleration limits and the maximum speed are reduced for the conservative driving style, and also the weights of the merit function are modified to increase the relevance of the safety and reduce the relevance of the utility.

Figure 8.22 shows the complete path followed by the EV during the demonstrator, using a color code to represent the level of driving automation. The description of the demonstrator's evolution is presented next:

- The EV started at $\mathscr{A} =$ `LoDA3`, reaching $\mathscr{A} =$ `LoDA4` after a while since there were no OVs or challenging situations in the scene.
- The human driver fell asleep between the first and second turns, as marked in Figure 8.22, but the autonomous driving process continued normally.

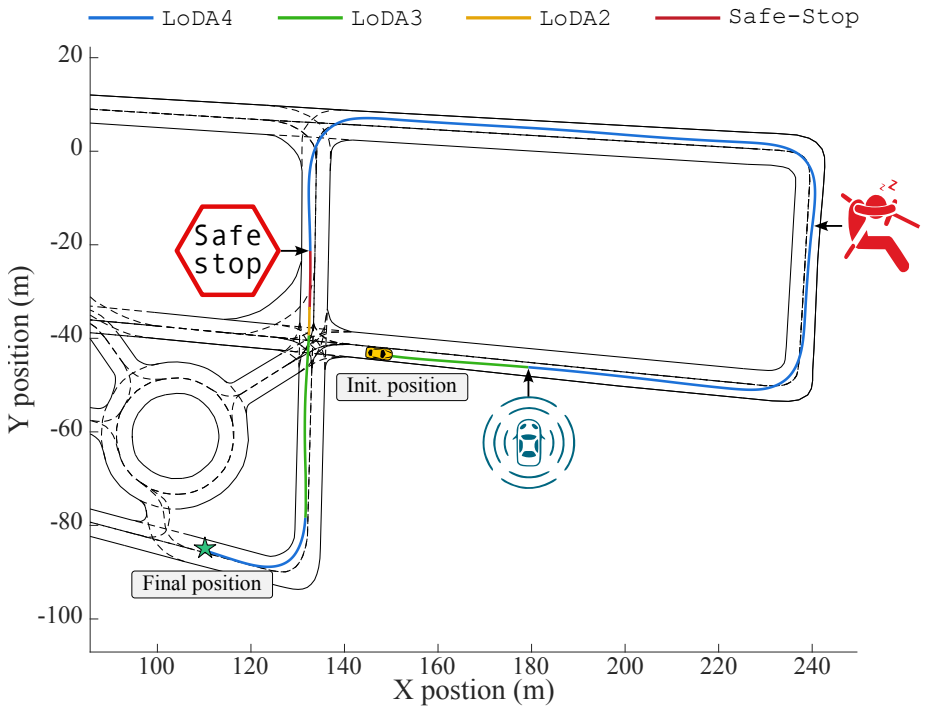
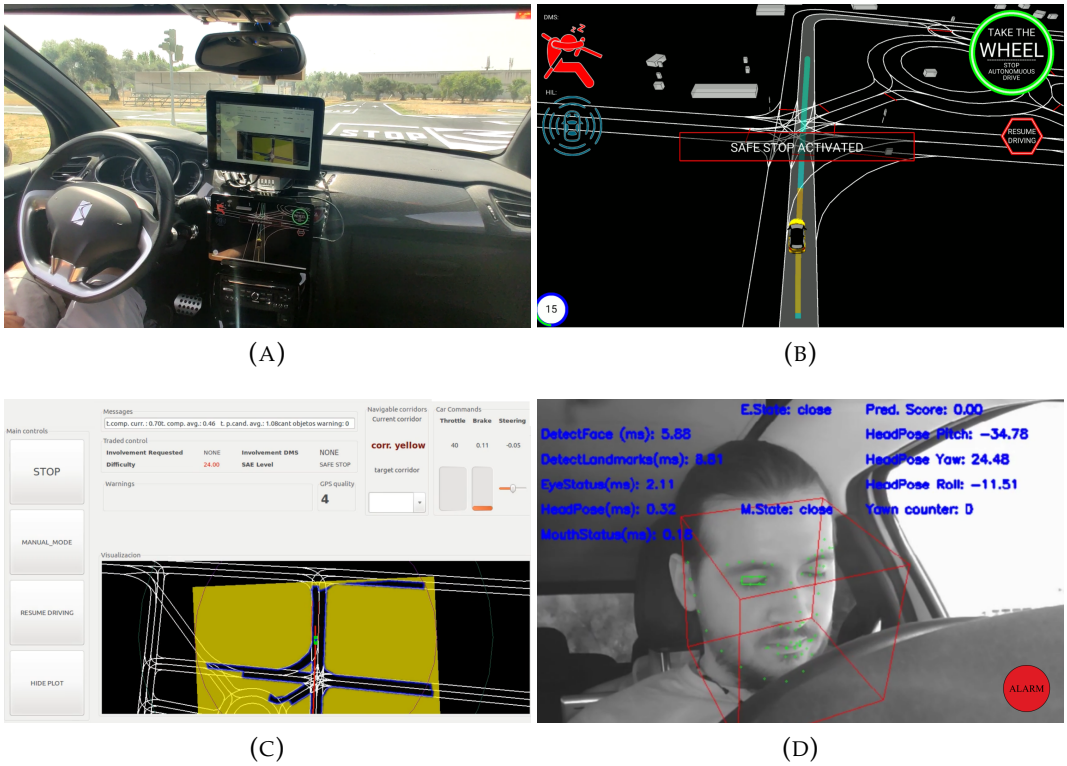


FIGURE 8.22: Complete travel for the human-in-the-loop automated driving demonstrator.

- In order to create a complex situation without including any other vehicles on the driving scene for safety reasons, an extra condition in the suitability computation was implemented temporarily in this demonstrator: every trajectory candidate which intersected with the yielding line of a must-stop RE is considered unsuitable. Thus, the scene complexity is increased when this type of REs is found.
- Using the mentioned strategy, the supervisor detected a complex scenario after the third turn and, as the human driver was not ready to take over the control of the vehicle, the level of driving automation was updated to $\mathscr{A} =$ `Safe-Stop`.
- The status of the demonstrator at this moment is presented in Figure 8.23, which shows the view from inside the EV (8.23a), the HMI display (8.23b), the graphic user interface of the ADS software (8.23c) and the camera image of the DMS system (8.23d). The DMS image shows that the driver is asleep with his eyes closed and his head down. The HMI shows the human driver is not involved and also that the safe-stop mode has been activated.
- This mode is maintained until it is manually deactivated using the $h = RS$ (Resume driving) option located at the right side of the HMI.



(A) (B)

(C) (D)

FIGURE 8.23: Demonstrator status during safe-stop maneuver. (A) Inside of the vehicle. (B) HMI. (C) Planner GUI. (D) DMS.

Figure 8.24 shows the LoDA during the travel, and Figure 8.25 shows the involvement of the human driver estimated by the DMS, and the required involvement level computed by the supervisor.

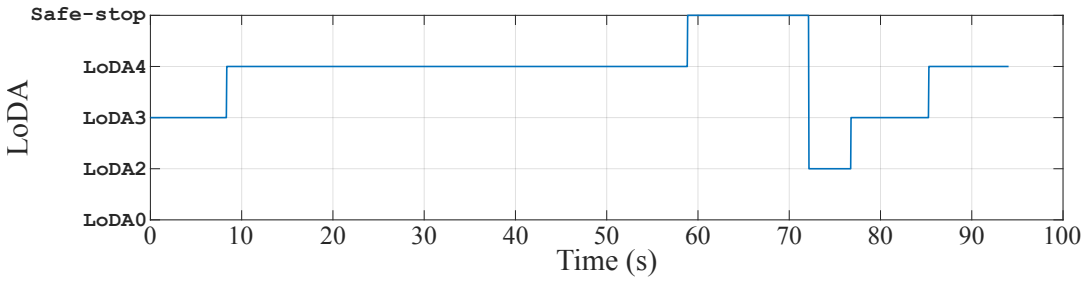


FIGURE 8.24: Level of driving automation during the DMS demonstrator.

The level of driving automation was set to $\mathscr{A} = \texttt{LoDA4}$ from $t = 8.3\ s$. The driver fell asleep at $t = 25.4\ s$, and the status of the DMS was changed from High to Low. The scene complexity increased at $t = 58.8\ s$ and since the driver was still not involved, the LoDA level changed to $\mathscr{A} = \texttt{Safe-Stop}$. The driver woke up at $t = 63.4\ s$, and resumed autonomous driving from the HMI at $t = 72\ s$, at that moment, the LoDA was set to $\mathscr{A} = \texttt{LoDA2}$, which requires a high involvement level. During this status, the driver stopped focusing on the road ($t = 75\ s$), and an audio-visual warning was triggered on the HMI, this situation is depicted in Figure 8.26. From that moment on, autonomous driving continued with no complications.

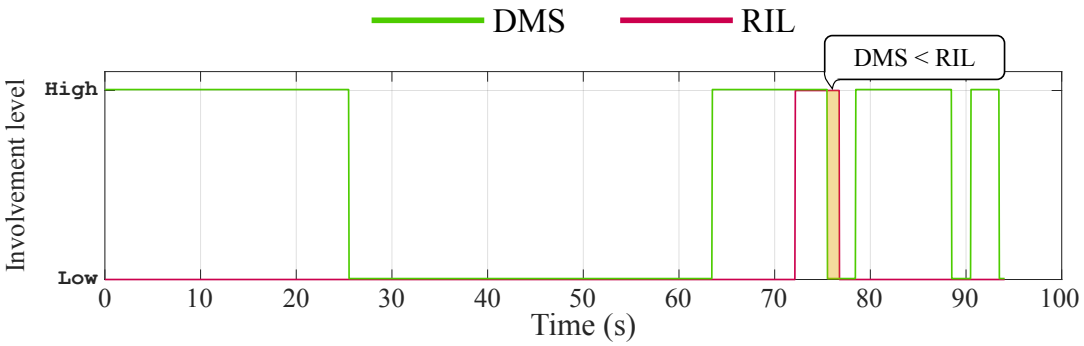
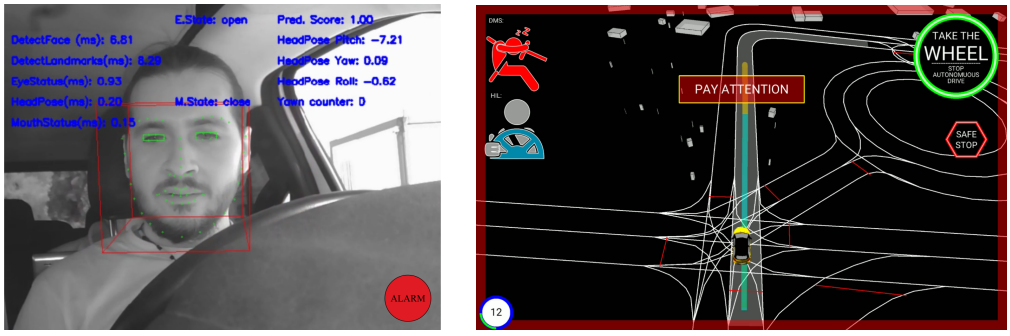


FIGURE 8.25: Measured and requested involvement on the DMS demonstrator.



(A) (B)

FIGURE 8.26: Demonstrator status after resuming autonomous driving. (A) DMS. (B) HMI.

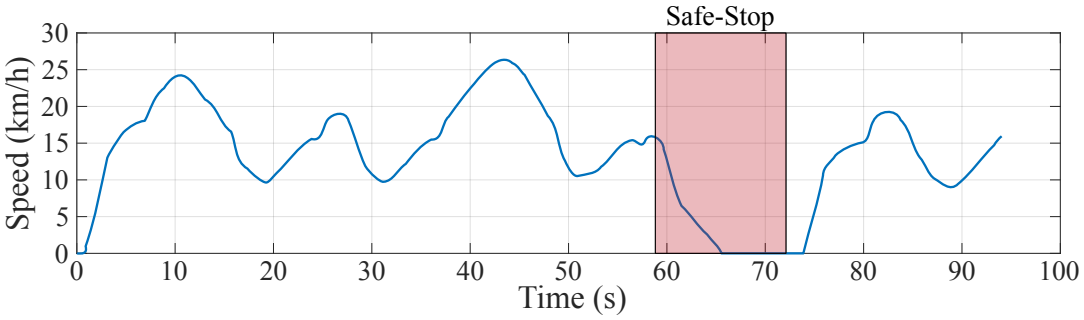


FIGURE 8.27: Speed during the DMS demonstrator.

The speed of the EV during this demonstrator is presented in Figure 8.27. It shows how the speed is reduced to make turns 1 and 2, then, the EV almost reached maximum speed at $t = 44\ s$. Once the safe-stop mode was activated, the speed was immediately decreased until 0 km/h. When autonomous driving was resumed, the safe speed limit was respected because $\mathscr{A} =$ `LoDA2`.

8.4 New Control Project Demonstrators

One of the goals of the New control European project was to implement adaptive and predictive motion planning algorithms which consider the vehicle limitations,

and the environment situation to achieve a holistic decision-making system. The algorithms should be validated in urban and suburban simulated scenarios, in order to implement them in demonstrator vehicles.

The following demonstrators were carried out in the testing track of the Centre for Automation and Robotics, and they exhibit part of the CSIC's contribution to this project. Some design parameters were modified for these demonstrators regarding the default parameters presented in section 8.2.1. The modified parameters are listed in Table 8.4.

TABLE 8.4: Configuration parameters for the New Control real vehicle demonstrators.

Parameter	Value	Units
N_p	1100	–
v_{max}	30	km/h
$\mathcal{L}_{max}$	$\{1.7\,,\,1.5\,,\,1.9\}$	m/s^2
$\mathcal{W}$	$\{0.9\,,\,0.6\,,\,0.8\,,\,0.5\}$	–

In this case, longitudinal comfort and safety were prioritized over lateral comfort and utility. Also, the number of candidates and the acceleration limits were increased to improve the flexibility of the trajectory generator.

8.4.1 Unexpected behavior of an OV in a roundabout

In this demonstrator, the EV had to avoid a vehicle that stopped suddenly while driving on the right-most lane of a roundabout. The EV had to manage the situation by changing to the inner lane of the roundabout and then leaving the roundabout at the intended exit. Figure 8.28 shows the valid trajectory candidates (704 in total) once the grey OV has stopped and the maneuver planner has decided to make a lane-change maneuver.

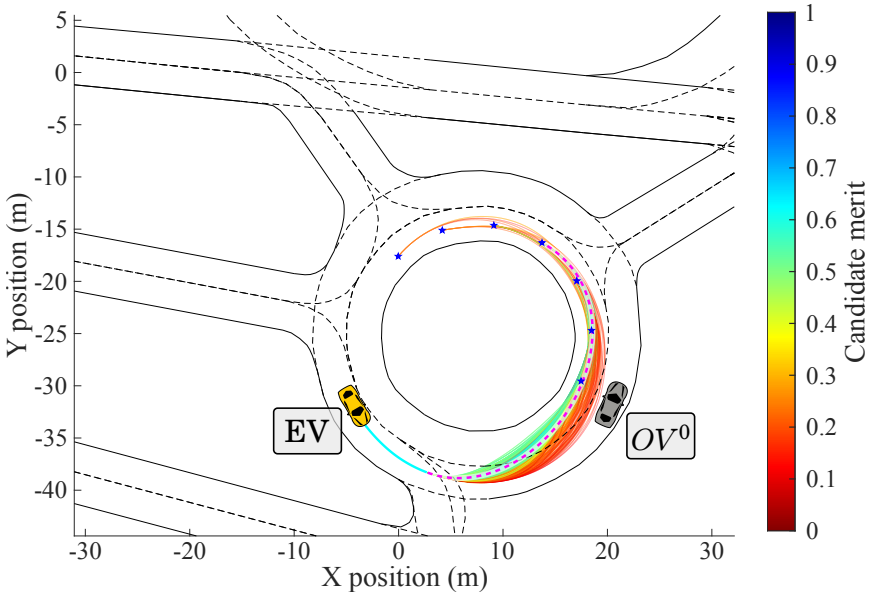


FIGURE 8.28: Trajectory set during vehicle-evasion maneuver.

It can be observed how the candidates closer to the stopped vehicle had the lowest merit, compared to the further-most candidates. This is appreciated in the reachability map Γ, which is plotted in Figure 8.29. As can be seen in the image, there is a large number of candidates with very poor performance on the safety DV, which is due to the closeness with the stopped vehicle. Many of these unsafe candidates have a good performance on the lateral comfort DV, as expected since they have broader curves.

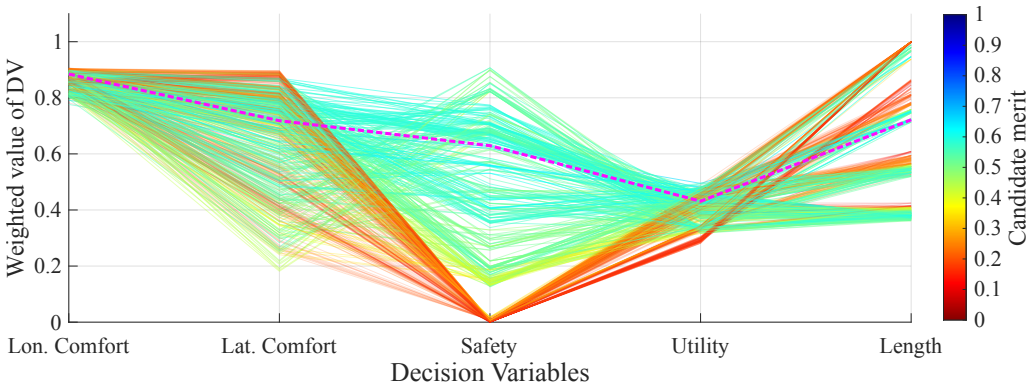


FIGURE 8.29: Reachability set during vehicle-evasion maneuver.

The selected candidate is highlighted in both figures using a dashed magenta line. In Figure 8.28 it can be seen that the best trajectory candidate is not the one furthest nor closest to the grey OV, but it is rather in between, which depicts a trade-off between the lateral comfort and the safety of the candidate, as perceived in Figure 8.29.

The traffic status of the demonstrator on the analyzed scene is presented in Figure 8.30. It shows the EV driving on the roundabout and the grey OV stopped a few meters ahead.

After the EV has avoided the stopped vehicle using the inner lane of the roundabout, it has to leave the roundabout at the intended exit. Figure 8.31 shows the trajectory set once the outer-most corridor of the roundabout is considered safe again by the maneuver planner. In this case, the geometry of the road was more irregular, and there were only 60 valid trajectories. In such case, the best trajectory candidate

FIGURE 8.30: Real traffic status during the vehicle-evasion maneuver.

has one of the longest paths, and one of the best performances on lateral acceleration, safety, and utility DVs, as can be seen in the reachability map shown in Figure 8.32.

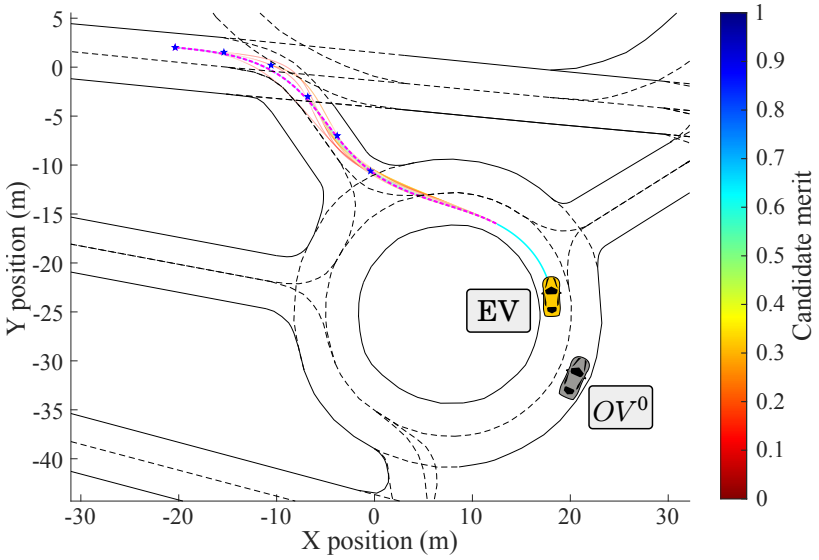


FIGURE 8.31: Trajectory set during roundabout exit.

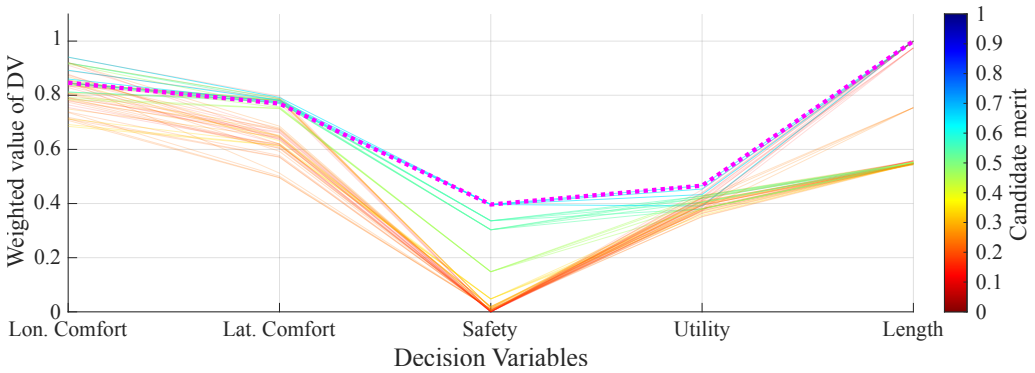


FIGURE 8.32: Reachability set during roundabout exit.

The complete path of this demonstrator is depicted in Figure 8.33. It shows that, even though there was a tight space inside the roundabout to avoid the grey OV, the maneuver was completed satisfactorily with a safe enough gap. It can also be observed that the EV followed a smooth path during the whole demonstrator.

The path's curvature is plotted in Figure 8.34. It can be observed that the curvature was continuous all along the travel. The largest curvature was $\kappa = -0.2m^{-1}$, which was reached at $t = 12\ s$ and it corresponded to the entrance of the roundabout. The lane change maneuver started at $t = 16.1\ s$, and the maximum curvature during that maneuver was $\kappa = 0.13\ m^{-1}$.

8.4.2 Merging into a roundabout with traffic

In this demonstrator, the merging-before model presented in section 5.5.4 was tested in a real-life scenario, where the EV had to merge into a roundabout between two OVs, in a safe manner. The results include the trajectory generation process during

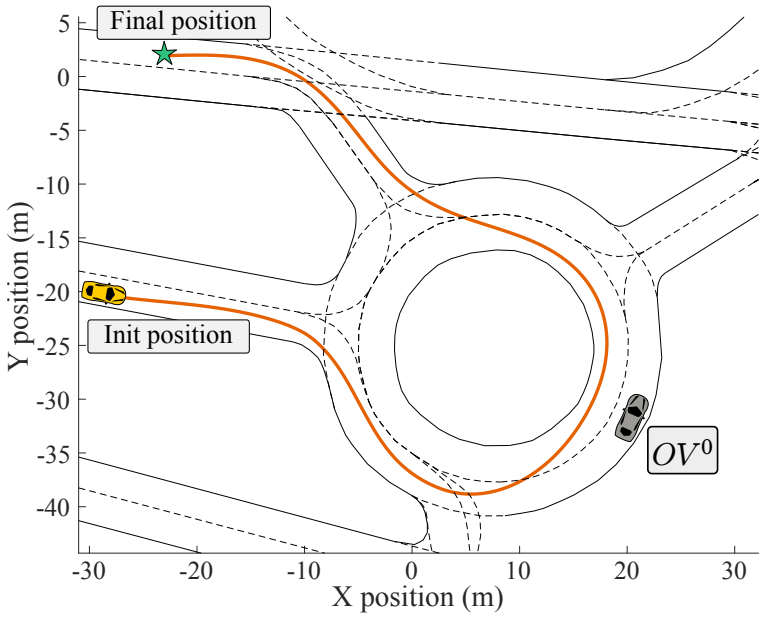


FIGURE 8.33: Complete path of the unexpected behavior on a roundabout demonstrator.

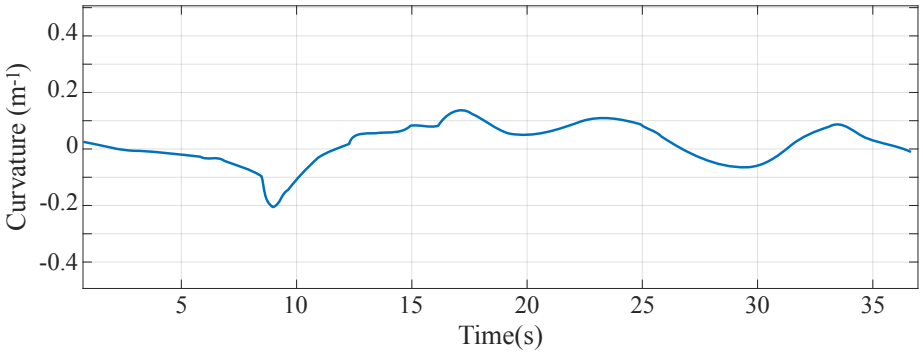


FIGURE 8.34: Path curvature of the unexpected behavior on a roundabout demonstrator.

the merging maneuver, as well as a general analysis of speeds, accelerations, and distances with respect to other OVs during the whole travel.

The trajectory generation process when merging into the roundabout is detailed next. In Figure 8.35 it is presented the trajectory set Γ as well as the positions of the vehicles. There is a leader (OV^1) and a lag (OV^0) vehicle present in the scene and the merging gap between them is around 25 m. Each trajectory is plotted using a color code to represent its merit and the best candidate τ^{best} is highlighted using a magenta dashed line.

Figure 8.36 depicts the speed profile for the best trajectory candidate. The image shows the limit-speed profile Λ^{best}, the resulting speed profile V^{best}, and the current speed of the OVs. As can be seen, the initial speed of the EV is $v_{ego} = 7\ km/h$, while the traveling speeds of the OVs are $v^0_{OV} = 12\ km/h$ and $v^1_{OV} = 16\ km/h$. The EV accelerates at the beginning of the trajectory to guarantee a safe gap with respect to OV^0 until it reaches the limit-speed profile, which restrains the comfort limits.

The position evolution of the traffic agents during this maneuver is presented in

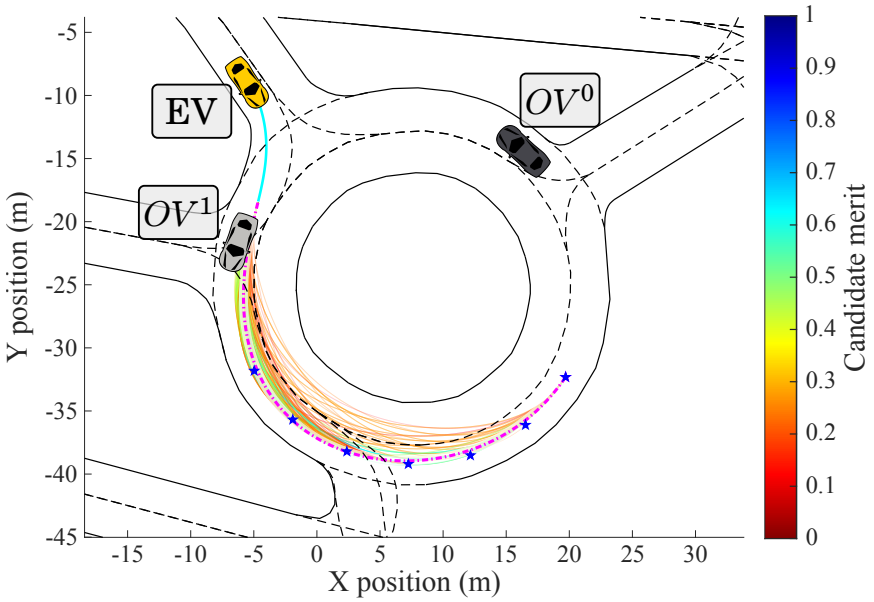


FIGURE 8.35: Trajectory set the merging maneuver.

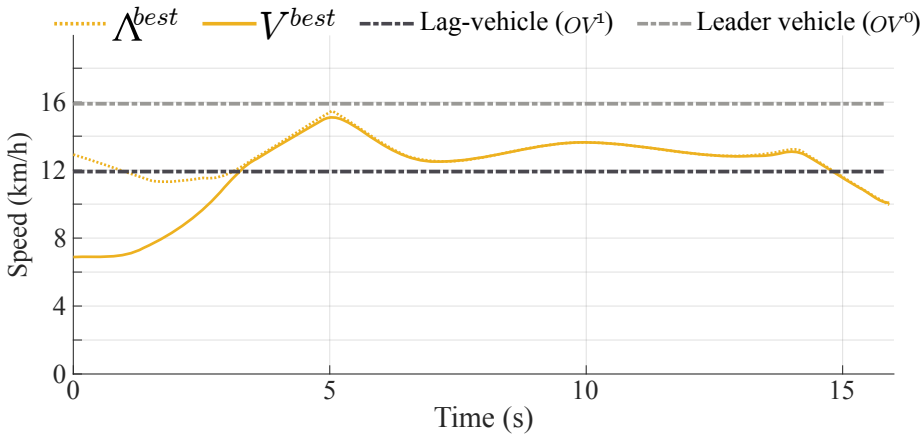


FIGURE 8.36: Speed profile of the best trajectory candidate during the merging maneuver.

Figure 8.37, where it can be seen that the EV is able to maintain its position between the safe positions $\mathcal{G}_G$ and $\mathcal{G}_D$.

Figure 8.38 shows the status of the vehicles during the merging maneuver in the demonstrator. Figure 8.38a presents an aerial view of the traffic scene, while 8.38b shows the point of view of OV^0.

The reference and real speeds of the EV during the demonstrator are plotted in Figure 8.39. The figure shows how the limit speed $v_{sl} = 7\ km/h$ is set as the reference speed from $t = 8.5\ s$ as the EV approaches to the roundabout and does not find a trajectory to merge before OV^1. At $t = 12.5\ s$, the trajectory generator finds a trajectory able to merge between the OVs and the EV starts to accelerate to complete that maneuver. Once into the roundabout, the speed is limited by both the navigation speed of OV^1 and the curvature of the road and, in order to comfortably leave the roundabout at the intended exit, the speed is reduced at $t = 25.8\ s$.

The longitudinal and lateral accelerations during the demonstrator are plotted

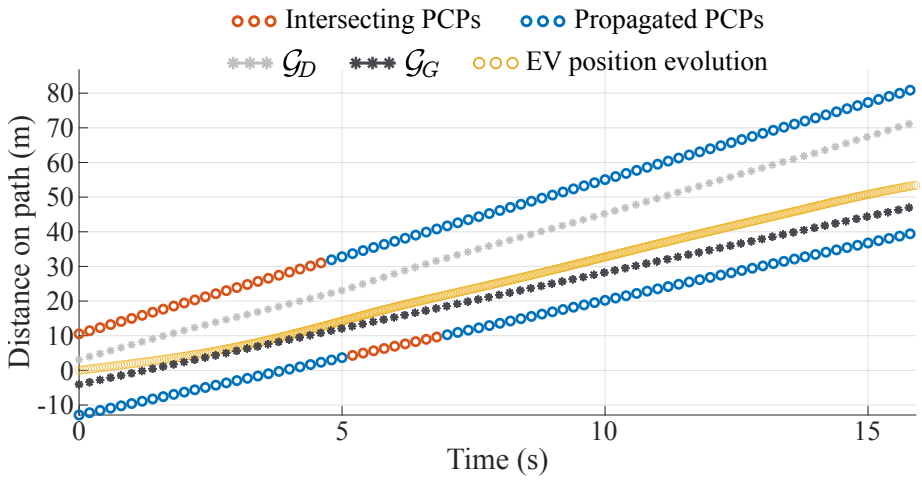


FIGURE 8.37: Position-on-path evolution and PCPs during the merging maneuver.

(A) (B)

FIGURE 8.38: Demonstrator status during the merging maneuver. (A) Aerial view. (B) Lag vehicle's point of view.

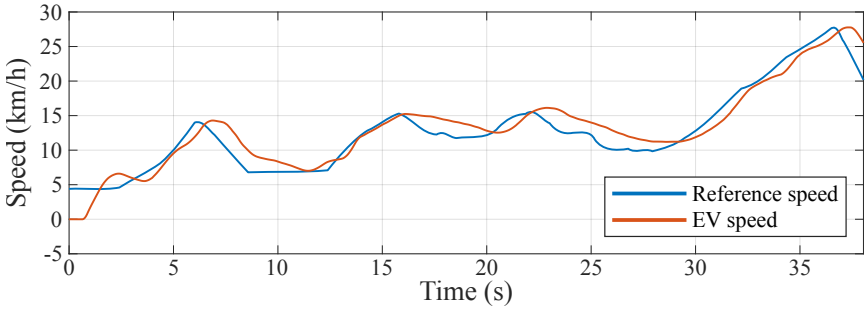


FIGURE 8.39: Reference and real speeds for the EV during the merging into a roundabout demonstrator.

in Figure 8.40. From the longitudinal acceleration data (Figure 8.40a) it can be concluded that the trajectory planner was able to drive the EV smoothly without sudden acceleration changes during the demonstrator. Also, the acceleration limits of Table 8.4 were respected. Regarding the lateral acceleration, the highest values were found once inside the roundabout after $t = 17\ s$. In three different moments ($t = [17.5\ s, 18.5\ s, 26.4\ s]$) the maximum acceleration limit was exceeded, which can be explained with the large tracking error of the longitudinal speed at those

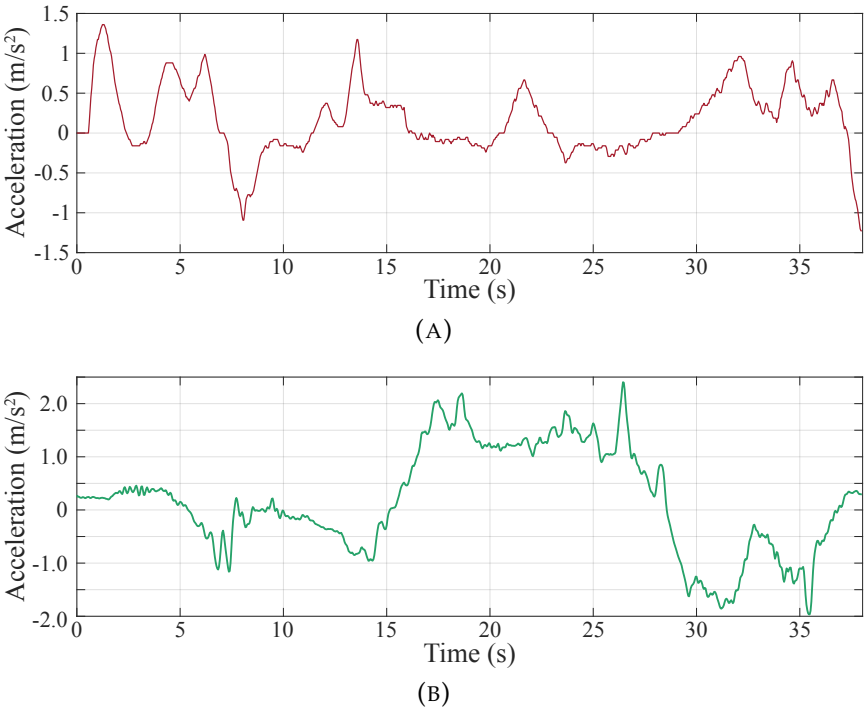


FIGURE 8.40: EV accelerations for the merging into a roundabout with traffic demonstrator. (A) Longitudinal acceleration (B) Lateral acceleration.

moments (See Figure 8.39).

Finally, the gaps regarding the OVs during the complete demonstrator are plotted in Figure 8.41, being highlighted the time slot when the merging maneuver was performed. The figure also shows the safe gap with respect to the lag OV, calculated using (5.40) with the parameters shown in Table 8.1 and its speed during the merging maneuver. The average gaps for this time slot are presented in Figure 8.42.

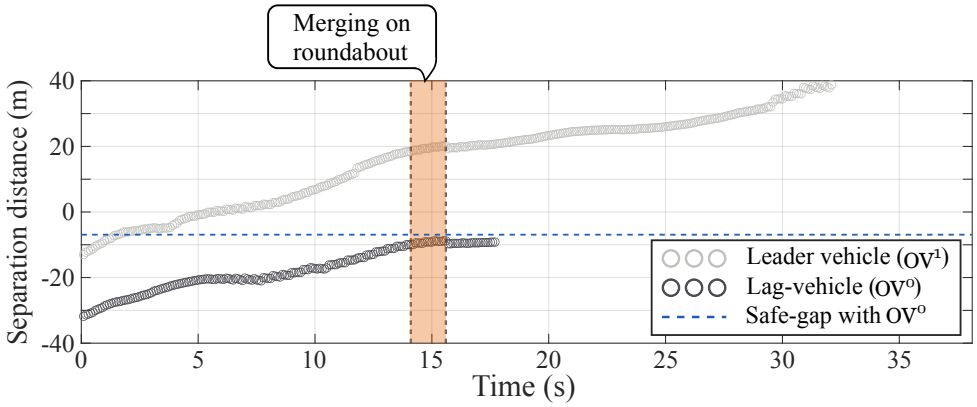


FIGURE 8.41: Gaps along time regarding OVs during merging demonstrator.

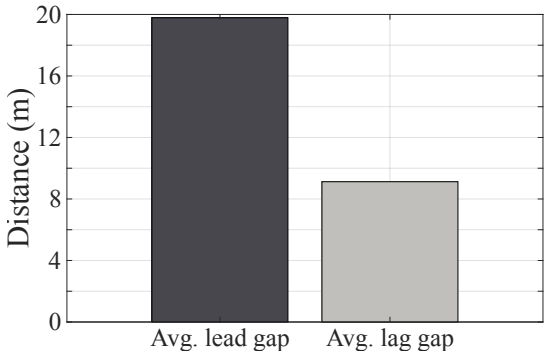


FIGURE 8.42: Average lead and lag gaps OVs during merging demonstrator.

8.4.3 Integration with IAMP

As mentioned in section 5.5.2, the motion planning process can use different motion prediction strategies in order to compute the PCPs used in the speed profile generation process. The motion planning task is a work of constant negotiation with the rest of the traffic agents. An incorrect prediction due to either excess or deficiency could cause the motion planning algorithm to adopt overly conservative or reckless behaviors that can eventually become a dangerous scenario. This demonstrator compares (i) a feedback strategy between the interaction-aware motion prediction (presented in [167]) and the motion planning strategy, with (ii) the linear-speed motion prediction model. The IAMP algorithm produces a probabilistic estimation of the future positions and intentions of the surrounding vehicles by taking into account traffic rules, vehicle interaction, road geometry, and the reference trajectory of the ego-vehicle.

To evaluate the performance of the motion planner, a total of four experiments were executed in the demonstrator, two of them using CSMP and the other two using IAMP. The experiments were compared using the KPIs described in Table 5.19, computed from the resulting trajectory of the EV after completing the scenario.

The configuration parameters modified for this demonstrator are presented in Table 8.5. The acceleration limits and the maximum speed were increased concerning the default values, the number of trajectory candidates was reduced to improve the computational cost, and the weights of the merit function were configured to prioritize longitudinal comfort and safety rather than lateral comfort and utility.

TABLE 8.5: Configuration parameters for the New Control real vehicle demonstrators.

Parameter	Value	Units
N_p	500	–
v_{max}	50	km/h
$\mathcal{L}_{max}$	$\{2.5, 2.0, 2.2\}$	m/s^2
$\mathcal{W}$	$\{0.9, 0.6, 0.8, 0.5\}$	–

In the layout of the demonstrator, there is a T-junction in a suburb of Heckstrasse, Aachen. The main road has the right of way and there is a left turn lane into the side road. The EV is located on the main road and has to cross the intersection completely.

The distance between the EV and p_{des} is $s = 140\ m$. There are two OVs that have a larger interaction with the EV during this experiment: OV^1 and OV^3. In the case of OV^3, it has to make a left turn to incorporate into the side road, crossing the ego-lane; as for OV^1, it incorporates from the side road to the main road, on the ego-lane. The scenario was recreated in SCANeR Studio using the software-in-the-loop architecture depicted in Figure 7.8 and using the Mission handler module to replicate a traffic scene from the `inD` public dataset [171].

Figure 8.43 shows the speed and acceleration profiles for the demonstrator. For the CSMP-based experiments, the EV slowed down and stopped on the main road for 2 seconds approximately, even though it should not stop at all since it was driving on the priority road.

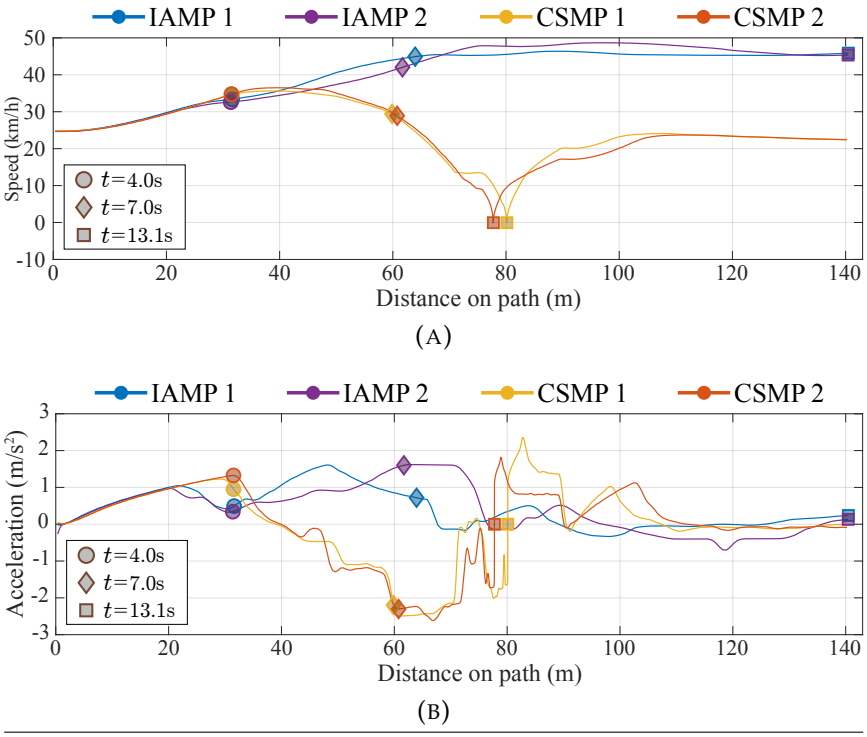


FIGURE 8.43: Speed and acceleration profiles for the different experiments on the integration with IAMP demonstrator. (A) Speed profiles. (B) Longitudinal acceleration profiles.

Note that this kind of maneuver (stopping on a priority road) is not only inefficient but also very dangerous and it may cause accidents with OVs driving behind the EV since they would not expect this type of behavior. All experiments had very similar speed profiles during $s \leq 40\ m$, (which is the first $t \approx 4.6\ s$ of travel). At that moment, the predictions for OV^1 using CSMP went through the ego-lane without taking into account the roads' priorities (see Figure 8.44a), which caused the EV to reduce its speed. On the other hand, the predictions of the IAMP strategy take into account the traffic rules and the predictions of OV^1 ended at the yielding line, as can be seen in Figure 8.44b. As the EV kept traveling, it reached almost v_{max} for the IAMP-based experiments at $s = 70\ m$, while it kept reducing the speed at a rate of $\gamma_x \approx -2.4\ m/s^2$ for the CSMP-based experiments. Finally, the EV reached the destination point at $t \approx 13.1\ s$ for the IAMP-based experiments; as for the CSMP-based experiments, the EV was completely stopped in the middle of the road at that moment. Due to this speed reduction from the EV, both OV^1 and OV^3 merged into the

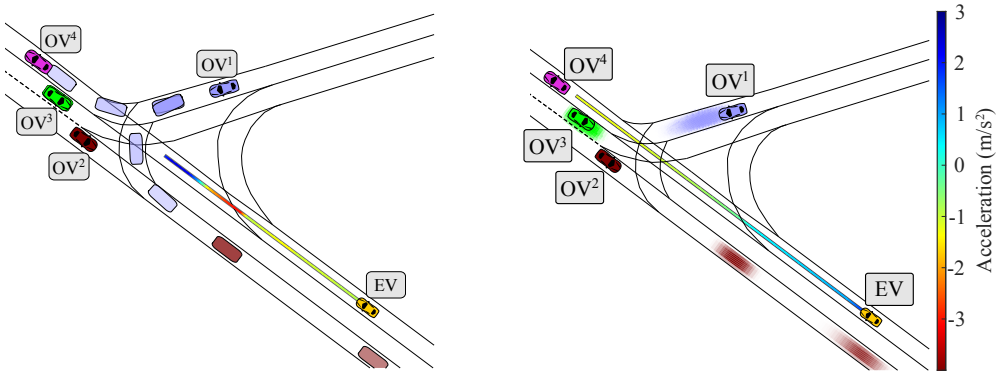


FIGURE 8.44: Predictions for 2, 4 and 6 s for the demonstrator at $t = 5.8\ s$. The colors of the predictions match the corresponding OV. (A) CSMP. (B) IAMP.

intersection before the EV, which affected the driving speed of the EV during the last section of the travel, due to the low speed of OV^1 after it merged into the main road.

The performance indicators of the experiments for this demonstrator are shown in Figure 8.45. The longitudinal comfort-related indicators are better for the IAMP-based experiments than for the CSMP-based experiments, this is due to the accelerations involved in the yielding maneuver on the T-junction. On the other hand, the CSMP-based experiments showed a better performance on the lateral comfort-related indicators, this can be explained by the lower speeds of these experiments, which imply a low lateral acceleration. Nevertheless, the difference between the maximum lateral acceleration between the CSMP and the IAMP-based experiments is only $\Delta\gamma_y \approx 0.28\ m/s^2$. Regarding the RTTCE, both strategies presented a similar performance, none of which indicates any dangerous situation. The main difference in the performance of the experiments is presented in the travel time, which was $\bar{t} = 13.1\ s$ for IAMP-based experiments, and $\bar{t} = 25.7\ s$ for CSMP-based experiments.

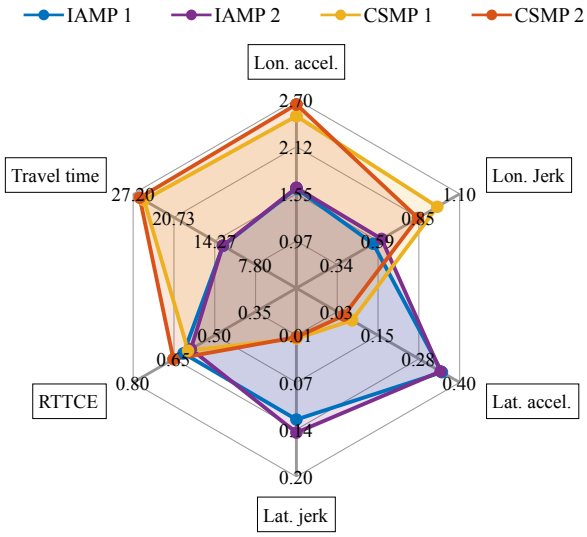


FIGURE 8.45: Performance indicators for the different experiments on the integration with IAMP demonstrator.

8.5 Demonstrators' Videos

Most of the demonstrators presented in this section have corresponding videos which provide the reader with a broader context and a more complete point of view about the functioning of the automated driving system proposed in this thesis. The videos of the demonstrators are summarized below:

- **Urban intersection:** This video shows the evolution of the experiment from different points of view while displaying the status of the HMI at all times. Available at www.youtube.com/watch?v=iOn0VP7fZ3A.
- **Overtaking in a two-way street:** This video presents the evolution of the overtaking maneuver and how the EV waits until the opposite-way OV crosses completely. Available at www.youtube.com/watch?v=0F4ov4ltFE4.
- **Cooperative re-routing on blocked road:** The video presents how the alternative route is presented to the human driver and is used after it is manually accepted. Available at https://www.youtube.com/watch?v=iClwErSexs8.
- **Human-in-the-loop automated driving :** This video depicts a synchronized status of the inside of the vehicle, the HMI, the DMS GUI, and the DMS video streaming during the complete travel.
  Available at https://www.youtube.com/watch?v=TLV9myrIkG8.
- **Unexpected behavior & merging into a roundabout:** In this case, the same video presents both (i) unexpected behavior inside a roundabout and (ii) merging into a roundabout with traffic demonstrators.
  Available at https://www.youtube.com/watch?v=Zha-9FbH9A0.

Chapter 9

CONCLUDING REMARKS

9.1 Introduction

This last chapter summarizes the developments presented in this manuscript in section 9.2. It also presents future works and perspectives regarding the proposed autonomous driving system in section 9.3. The chapter ends with the scientific dissemination outcomes obtained in this Ph.D. thesis, listed in section 9.4.

9.2 Conclussions

This Ph.D. thesis presents the implementation of an autonomous driving system for urban scenarios considering a human-in-the-loop approach. The proposed system is formed by maneuver planner, motion planner, and supervisor modules, which interact with each other to provide a comfortable, yet safe driving behavior. The system was tested and validated in diverse real-life demonstrators with an instrumented vehicle, and in realistic simulation environments with a software-in-the-loop architecture where the driving agents dynamically adjust their behavior according to the evolution of the ego-vehicle. In the following sections, some concluding remarks for each of the three main modules of the system are presented.

9.2.1 Maneuver planning

A maneuver planner for automated vehicles was designed and implemented within a hierarchical architecture to make tactical decisions about lane selection, regulatory-elements handling, and trajectory-generation triggering. The lane-selection algorithm uses a utility model which considers both the current traffic status and the global route, while it ensures safety by applying a gap-acceptance model. The regulatory element handling is performed using an FSM that can manage right-of-way, must-stop, and pedestrian-crossing signals. The trajectory generation process is triggered based on different traffic events like the detection of a new obstacle, a change in the navigation corridor, or approaching the end of the current trajectory. The maneuver planner establishes a set of restrictions that are taken into account by a motion planner to generate a reachable trajectory set, and finally, a valid trajectory for the EV. The trajectory set is involved in the decision-making process, as the number of valid candidates is taken into account to override a lane-changing maneuver, and also the trajectory generator has the possibility of finding available gaps at intersections or roundabouts to decide if the vehicle merges without stopping or performs a full stop in the yielding line of the signal. This approach provides flexibility and robustness to the system in complex driving scenarios.

The decision-making architecture was validated in different urban scenarios like intersections, multi-lane roads, and roundabouts; showing good performance in all of them using a unique strategy.

9.2.2 Trajectory planning

The motion planner strategy proposed in this Ph.D. thesis consisted in computing a set of trajectory candidates and selecting the best of them according to a merit function. This approach provides a scenario-independent trajectory generation strategy, able to handle safely many driving scenarios while maintaining comfort constraints.

The trajectory candidate generation process consists of finding a valid path and then computing a set of possible speed profiles to follow. The path generation process is based on Quintic Bézier curves, which allows to maintain a G^2 continuity along the travel and provides a diverse set of alternatives to handle curved roads, lane changes, or roundabouts. The speed generation algorithm considers two different inter-distance models according to the traffic layout and also takes into account comfort and restrictions based on regulatory elements to obtain collision-free trajectories.

The merit-based selection strategy is based on five decision variables: longitudinal comfort, lateral comfort, safety, utility, and length. The strategy was tested by executing a set of experiments on urban-like scenarios with different weight configurations and comparing the final trajectories and relevant KPI. These experiments showed that the driving style of the ADS can be customized by modifying the priority of the decision variables. In addition to that, the results showed that when the comfort-related variables had higher priority, the resulting acceleration profiles were smoother and had lower overall values, both in longitudinal and lateral comfort. When safety was considered the most important decision variable, the resulting trajectories did not invade adjacent lanes and kept larger distances with respect to the obstacles, reducing the risk of possible collisions. When the priority was the utility, the generated trajectories were faster and exhibit bolder acceleration profiles.

The robustness of the algorithm was tested in different types of scenarios, but a special focus was given to dataset-based scenarios in roundabouts, where a set of experiments were performed under different driving conditions, evaluating not only the dataset-based configurations but modified versions of the original scenes. The system was able to safely handle all driving scenarios and maintain the comfort constraints. The performance of the algorithm was compared with human-driven data under the same environment and with another state-of-the-art insertion maneuver strategy. The comparison showed that the trajectory generator was able to handle real-traffic situations similarly to human drivers on different performance indicators like maximum longitudinal and lateral accelerations, average jerk values, time to close encounter with OV, or travel time.

9.2.3 Traded control

A novel traded control approach is presented, where the level of automation and the required involvement level of the driver is automatically determined from estimations of the driving scene complexity level and the driver's drowsiness. The scene complexity is evaluated from the suitability of the trajectory candidates set, which provides an objective way to determine how well the ADS is handling the current driving scene.

To maintain safety when driving in the traded control architecture, alternative levels of driving automation were proposed, which do not leave the control of the vehicle disengaged at any moment unless required by the driver. The automation interacts with the human driver using an HMI that displays warnings, the requested level of involvement, and the planning decisions. The human driver can also use the HMI to change the driving mode, select the destination point, cancel an overtaking maneuver, or perform a safe-stop maneuver, among others.

The implemented ADS was validated in a simulated urban scenario, where it required higher involvement levels from the human when the scenes were more critical (e.g. approaching an intersection or roundabout with traffic).

9.3 Future Works

Although this Ph.D. thesis has introduced a human-in-the-loop motion planner for urban environments, and it has been successfully validated in relevant or even operational scenarios, there are still different areas for further exploration. Some potential improvements related to the contributions are presented below.

9.3.1 Maneuver planning

Even though the current lane-selection model has proven to have a good performance in different urban layouts, the safety of lane changes can be improved by taking into account other considerations like the uncertainty of the perception, the spacial occlusions, or the involvement level of the human driver. Also, the computation of the gap acceptance can be enhanced by considering the dynamic restrictions of the vehicle. Regarding the handling of regulatory elements, future works will be oriented toward the incorporation of further elements such as traffic lights or reserved lanes.

It is very important to have good integration between the tactical and operational layers of the decision-making process. The proposed architecture goes in that

direction as the number of valid candidates is taken into account when evaluating the feasibility of a lane-change maneuver. However, the level of integration can be improved by implementing a fall-back maneuver strategy in which the maneuver planner can guarantee a safe alternative in case the current trajectory is no longer valid. This could be done by requesting backup candidates in alternative lanes or out-of-lane driving zones.

The current approach has a strong dependency on the precision of the digital map information. More research needs to be done regarding this issue, e.g. implementing algorithms for dynamic navigable-zone detection and/or lane detection that can minimize potential errors or omissions on the original digital maps.

9.3.2 Trajectory planning

Since the trajectory generation algorithm produces a large variety of candidates on the target corridor, but only a percentage of those candidates meet the validity conditions, a machine-learning approach could be used to determine the most adequate parameters to create the path candidates and the acceleration profiles based on the driving context. As a result, the number of candidates to evaluate could be reduced, optimizing the computation time of the algorithm.

The waypoints used to create the trajectory set are computed from the centerline of the navigation corridors. The strategy to sample the navigation space to generate the trajectory candidates could be improved by applying an approach such as artificial potential fields, which not only takes into account the road geometry, but also the position of the obstacles.

To obtain a more naturalistic driving style from the trajectory generation, it can be implemented a learning-based system that automatically infers the most appropriate weights for the merit function to match a human driving profile in a given driving context. Moreover, different criteria besides the proposed decision variables can be used in the trajectory selection process; an interesting approach would be to incorporate ethics criteria in the merit function.

The merging-before speed profile generation may be easily applied to other merging scenarios an AV may encounter, like ramp merging on highways or lane changing with traffic. Future work will be oriented towards the deployment and performance evaluation of the trajectory generator in these driving situations, both in simulation and real scenarios.

Despite being tested in a large variety of driving situations, there are some driving situations where the robustness of the motion planning algorithm can be increased, such as scenarios where the OVs are accelerating heavily, dealing with occlusions on the exteroceptive sensors, or losing precision on the location data.

9.3.3 Traded control

The role of the human driver in a traded control architecture is not completely defined in the state-of-the-art, and different strategies of the human-automation interaction can be further studied. One of these possibilities is to increase the involvement of the human in the driving process, so it is not only a supervisor but also a commander who can actively participate in the tactical decisions of the vehicle. Thus, the driver is more focused on the driving process and it is kept in the control loop all the time, but at the same time, the automation is in charge of several tasks.

So far, the supervisor can establish a safe driving mode when the scene becomes complex and the human driver does not intervene to take over control of the vehicle. These capabilities can be extended by including other factors to activate safe driving mode besides the complexity of the scene and the driver engagement status, e.g. the uncertainty of the perception module regarding the surrounding environment or the possible failures of any proprioceptive sensor.

The performance of the traded control strategy can be further evaluated by conducting a study with different human drivers in simulated complex urban environments to assess the human factors associated with the proposed strategy. Different factors like acceptance, trust in the decisions, or the ability to communicate efficiently and clearly through the HMI, can be evaluated from a human factor perspective.

9.4 Scientific Dissemination

This Ph.D. thesis has been developed under the framework of several European research projects and a number of scientific articles have been published in journals and events related to autonomous driving vehicles. These publications are listed below.

9.4.1 Journal articles

- J. F. Medina-Lee, V. Trentin, J. L. Hortelano, A. Artuñedo, J. Godoy and J. Villagra. (2023). "$IA(MP)^2$ Interaction-aware motion prediction for an efficient motion planning in complex driving situations," *IEEE Transactions on Intelligent Vehicles*, (submitted).
- J. F. Medina-Lee, J. Godoy, A. Artuñedo and J. Villagra. (2022). "Speed Profile Generation Strategy for Efficient Merging of Automated Vehicles on Roundabouts With Realistic Traffic," *IEEE Transactions on Intelligent Vehicles*, `https://doi.org/10.1109/TIV.2022.3211350`.
- J. F. Medina-Lee, A. Artuñedo, J. Godoy, J. Villagra. (2021). "Merit-Based Motion Planning for Autonomous Vehicles in Urban Scenarios". *Sensors*, 21, 3755. `https://doi.org/10.3390/s21113755`.

9.4.2 Conferences

- J. F. Medina-Lee, V. Jiménez, J. Godoy and J. Villagra. (2022). "Maneuver Planner for Automated Vehicles on Urban Scenarios". In IEEE International Conference on Vehicular Electronics and Safety (ICVES), Bogota, Colombia, (pp. 1-7), https://doi.org/10.1109/ICVES56941.2022.9987128.

- J. F. Medina-Lee, A. Artuñedo, J. Godoy and J. Villagra. (2020). "Reachability Estimation in Dynamic Driving Scenes for Autonomous Vehicles". In IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, (pp. 2133-2139), https://doi.org/10.1109/IV47402.2020.9304672.

- J. F. Medina-Lee, J. Villagra, and A. Artuñedo. (2020). "Traded control architecture for automated vehicles enabled by the scene complexity estimation". In 4th International Conference on Computer-Human Interaction Research and Applications (CHIRA).

- J. F. Medina-Lee, V. Trentin, and J. Villagra, (2019). "Framework for motion prediction of vehicles in a simulation environment". In XL Jornadas de Automática (pp. 520-527). Universidade da Coruña, Servizo de Publicacións.

9.4.3 Book chapter

- J. Villagra, J. Vallvé Navarro, J. F. Medina-Lee, J. Solà Ortega, A. Artuñedo, and J. Andrade-Cetto. (2020). "Technological advances for intelligent vehicles: shared control in highly complex contexts and the automation of transport in segregated environments". In Boletín del Grupo Español del Carbón, 58, (pp. 15-22).

- J. Villagra, F. Jimenez, J. F. Medina-Lee, et al. (2023). Human-machine interaction. In J. Villagra, F. Jimenez (Eds.), *Decision-making Techniques for Autonomous Vehicles* (pp. 333-351). Elsevier.

Bibliography

[1] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016. DOI: 10.1109/TIV.2016.2578706.

[2] N. H. T. S. Administration, *Critical reasons for crashes investigated in the national motor vehicle crash causation survey*, 2018.

[3] S. International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (Surface Vehicle Recommended Practice: Superseding J3016 sep 2016)," SAE International, Tech. Rep., 2016.

[4] N. Gavanas, "Autonomous road vehicles: Challenges for urban planning in european cities," *Urban Science*, vol. 3, no. 2, 2019, ISSN: 2413-8851. DOI: 10.3390/urbansci3020061.

[5] F. Garrido and P. Resende, "Review of decision-making and planning approaches in automated driving," *IEEE Access*, vol. 10, pp. 100 348–100 366, 2022. DOI: 10.1109/ACCESS.2022.3207759.

[6] M. Morsali, E. Frisk, and J. Åslund, "Spatio-temporal planning in multi-vehicle scenarios for autonomous vehicle using support vector machines," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 611–621, 2021. DOI: 10.1109/TIV.2020.3042087.

[7] K. Liu, J. Gong, A. Kurt, H. Chen, and U. Ozguner, "Dynamic modeling and control of high-speed automated vehicles for lane change maneuver," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 329–339, 2018. DOI: 10.1109/TIV.2018.2843177.

[8] A. C. Charalampidis and D. Gillet, "Speed profile optimization for vehicles crossing an intersection under a safety constraint," in *Proceedings European Control Conference (ECC)*, 2014, pp. 2894–2901. DOI: 10.1109/ECC.2014.6862507.

[9] B. Park, Y.-C. Lee, and W. Y. Han, "Trajectory generation method using bézier spiral curves for high-speed on-road autonomous vehicles," in *Proceedings IEEE International Conference on Automation Science and Engineering (CASE)*, 2014, pp. 927–932. DOI: 10.1109/CoASE.2014.6899437.

[10] J. Villagra, V. Milanés, J. Pérez, and J. Godoy, "Smooth path and speed planning for an automated public transport vehicle," *Robotics and Autonomous Systems*, vol. 60, no. 2, pp. 252–265, 2012, ISSN: 0921-8890. DOI: 10.1016/j.robot.2011.11.001.

[11] C. Hubmann, M. Aeberhard, and C. Stiller, "A generic driving strategy for urban environments," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1010–1016. DOI: 10.1109/ITSC.2016.7795679.

[12] S. Masi, P. Xu, and P. Bonnifait, "Roundabout crossing with interval occupancy and virtual instances of road users," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4212–4224, 2022. DOI: 10.1109/TITS.2020.3042870.

[13] W. Zhan, C. Liu, C.-Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 459–464. DOI: 10.1109/ITSC.2016.7795595.

[14] H. Chen, X. Wang, and J. Wang, "A trajectory planning method considering intention-aware uncertainty for autonomous vehicles," in *Proceedings Chinese Automation Congress (CAC)*, 2018, pp. 1460–1465. DOI: 10.1109/CAC.2018.8623263.

[15] U. Baumann, Y.-Y. Huang, C. Gläser, M. Herman, H. Banzhaf, and J. M. Zöllner, "Classifying road intersections using transfer-learning on a deep neural network," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 683–690. DOI: 10.1109/ITSC.2018.8569916.

[16] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1590–1595. DOI: 10.1109/IVS.2017.7995936.

[17] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1379–1384. DOI: 10.1109/IVS.2018.8500556.

[18] L. Chen, X. Hu, B. Tang, and Y. Cheng, "Conditional dqn-based motion planning with fuzzy logic for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 2966–2977, 2022. DOI: 10.1109/TITS.2020.3025671.

[19] A. Aksjonov and V. Kyrki, "Rule-based decision-making system for autonomous vehicles at intersections with mixed traffic environment," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2021, pp. 660–666. DOI: 10.1109/ITSC48978.2021.9565085.

[20] L. Wei, Z. Li, J. Gong, C. Gong, and J. Li, "Autonomous driving strategies at intersections: Scenarios, state-of-the-art, and future outlooks," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2021, pp. 44–51. DOI: 10.1109/ITSC48978.2021.9564518.

[21] Y. Zhang, L. Angell, and S. Bao, "A fallback mechanism or a commander? a discussion about the role and skill needs of future drivers within partially automated vehicles," *Transportation Research Interdisciplinary Perspectives*, vol. 9, p. 100 337, 2021, ISSN: 2590-1982. DOI: 10.1016/j.trip.2021.100337.

[22] R. Parasuraman and C. D. Wickens, "Humans: Still vital after all these years of automation," *Human Factors*, vol. 50, no. 3, pp. 511–520, 2008. DOI: 10.1518/001872008X312198.

[23] L. Bainbridge, "Ironies of automation," *Automatica*, vol. 19, no. 6, pp. 775–779, 1983. DOI: 10.1016/0005-1098(83)90046-8.

[24] G. Salvendy, *Handbook of human factors and ergonomics*. John Wiley & Sons, 2012.

[25] T. W. Seppelt Bobbie D.and Victor, "Potential solutions to human factors challenges in road vehicle automation," in *Road Vehicle Automation 3*. Springer International Publishing, 2016, pp. 131–148, ISBN: 978-3-319-40503-2. DOI: 10.1007/978-3-319-40503-2_11.

[26] Y. Xing, C. Huang, and C. Lv, "Driver-automation collaboration for automated vehicles: A review of human-centered shared control," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1964–1971. DOI: 10.1109/IV47402.2020.9304755.

[27] H. Ahn, K. Berntorp, P. Inani, A. J. Ram, and S. Di Cairano, "Reachability-based decision-making for autonomous driving: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 1907–1921, 2021. DOI: 10.1109/TCST.2020.3022721.

[28] T. Inagaki, "Adaptive Automation: Sharing and Trading of Control," in CRC Press, 2003, vol. 2001.10, pp. 147–169. DOI: 10.1201/9781410607775.

[29] C. Huang, F. Naghdy, H. Du, and H. Huang, "Review on human-machine shared control system of automated vehicles," in *Proceedings International Symposium on Autonomous Systems (ISAS)*, 2019, pp. 47–51. DOI: 10.1109/ISASS.2019.8757749.

[30] M. Bojarski, P. Yeres, A. Choromanska, *et al.*, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911*, 2017. DOI: 10.48550/arXiv.1704.07911.

[31] S. Hecker, D. Dai, and L. Van Gool, "End-to-end learning of driving models with surround-view cameras and route planners," in *Proceedings European Conference on Computer Vision (ECCV)*, 2018, pp. 449–468. DOI: 10.1007/978-3-030-01234-2_27.

[32] W. Zeng, W. Luo, S. Suo, *et al.*, "End-to-end interpretable neural motion planner," in *Proceedings IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8652–8661. DOI: 10.1109/CVPR.2019.00886.

[33] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson, "If, when, and how to perform lane change maneuvers on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 4, pp. 68–78, 2016. DOI: 10.1109/MITS.2016.2565718.

[34] J. Nilsson, J. Fredriksson, and E. Coelingh, "Rule-based highway maneuver intention recognition," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015, pp. 950–955. DOI: 10.1109/ITSC.2015.159.

[35] J. Daniel, J.-P. Lauffenburger, S. Bernet, and M. Basset, "Driving risk assessment with belief functions," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 690–695. DOI: 10.1109/IVS.2013.6629547.

[36] Y. Chen, J. Zha, and J. Wang, "An autonomous t-intersection driving strategy considering oncoming vehicles based on connected vehicle technology," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2779–2790, 2019. DOI: 10.1109/TMECH.2019.2942769.

[37] J. Palatti, A. Aksjonov, G. Alcan, and V. Kyrki, "Planning for safe abortable overtaking maneuvers in autonomous driving," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2021, pp. 508–514. DOI: 10.1109/ITSC48978.2021.9564499.

[38] G. Xiong, Z. Kang, H. Li, W. Song, Y. Jin, and J. Gong, "Decision-making of lane change behavior based on rcs for automated vehicles in the real environment," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1400–1405. DOI: 10.1109/IVS.2018.8500651.

[39] J. Ziegler, P. Bender, M. Schreiber, *et al.*, "Making bertha drive: An autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014. DOI: 10.1109/MITS.2014.2306552.

[40] C. Urmson, J. Anhalt, D. Bagnell, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008. DOI: 10.1002/rob.20255.

[41] M. Montemerlo, J. Becker, S. Bhat, *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008. DOI: 10.1002/rob.20258.

[42] T. Gindele, D. Jagszent, B. Pitzer, and R. Dillmann, "Design of the planner of team annieway's autonomous vehicle used in the darpa urban challenge 2007," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2008, pp. 1131–1136. DOI: 10.1109/IVS.2008.4621268.

[43] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 913–929, 2013. DOI: 10.1109/TRO.2013.2254033.

[44] T. Gu, J. M. Dolan, and J.-W. Lee, "Automated tactical maneuver discovery, reasoning and trajectory planning for autonomous driving," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5474–5480. DOI: 10.1109/IROS.2016.7759805.

[45] K. Esterle, P. Hart, J. Bernhard, and A. Knoll, "Spatiotemporal motion planning with combinatorial reasoning for autonomous driving," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1053–1060. DOI: 10.1109/ITSC.2018.8570003.

[46] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014. DOI: 10.1109/TRO.2014.2312453.

[47] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1671–1678. DOI: 10.1109/IVS.2017.7995949.

[48] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 5–17, 2018. DOI: 10.1109/TIV.2017.2788208.

[49] M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for autonomously navigating urban intersections," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 825–830. DOI: 10.1109/IVS.2017.7995818.

[50] Z. Sunberg and M. J. Kochenderfer, "Improving automated driving through pomdp planning with human internal states," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20 073–20 083, 2022. DOI: 10.1109/TITS.2022.3182687.

[51] H. Bey, M. Sackmann, A. Lange, and J. Thielecke, "Pomdp planning at roundabouts," in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 2021, pp. 264–271. DOI: 10.1109/IVWorkshops54471.2021.9669232.

[52] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2076–2087, 2021. DOI: 10.1109/TITS.2020.3036984.

[53] M. Garzón and A. Spalanzani, "Game theoretic decision making for autonomous vehicles' merge manoeuvre in high traffic scenarios," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 3448–3453. DOI: 10.1109/ITSC.2019.8917314.

[54] Y. Ali, Z. Zheng, M. M. Haque, and M. Wang, "A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 220–242, 2019. DOI: 10.1016/j.trc.2019.07.011.

[55] N. Smirnov, Y. Liu, A. Validi, W. Morales-Alvarez, and C. Olaverri-Monreal, "A game theory-based approach for modeling autonomous vehicle behavior in congested, urban lane-changing scenarios," *Sensors*, vol. 21, no. 4, 2021. DOI: 10.3390/s21041523.

[56] S. Li, K. Shu, C. Chen, and D. Cao, "Planning and decision-making for connected autonomous vehicles at road intersections: A review," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, pp. 1–18, 2021. DOI: 10.1186/s10033-021-00639-3.

[57] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2020. DOI: 10.1109/TITS.2019.2913998.

[58] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1590–1595. DOI: 10.1109/IVS.2017.7995936.

[59] Q. Huy, S. Mita, H. T. N. Nejad, and L. Han, "Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles," *IEICE TRANSACTIONS on Information and Systems*, vol. 96, no. 2, pp. 314–328, 2013. DOI: 10.1587/transinf.E96.D.314.

[60] A. Plebe, M. Da Lio, and D. Bortoluzzi, "On reliable neural network sensorimotor control in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 711–722, 2020. DOI: 10.1109/TITS.2019.2896375.

[61] L. Chen, Y. He, Q. Wang, W. Pan, and Z. Ming, "Joint optimization of sensing, decision-making and motion-controlling for autonomous vehicles: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4642–4654, 2022. DOI: 10.1109/TVT.2022.3150793.

[62] M. Da Lio, F. Biral, E. Bertolazzi, *et al.*, "Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 244–263, 2015. DOI: 10.1109/TITS.2014.2330199.

[63] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007. DOI: 10.3141/1999-10.

[64] T. Toledo, C. F. Choudhury, and M. E. Ben-Akiva, "Lane-changing model with explicit target lane choice," *Transportation Research Record*, no. 1934, 157–165, 2005, ISSN: 03611981. DOI: 10.1177/0361198105193400117.

[65] W. J. Schakel, V. L. Knoop, and B. van Arem, "Integrated lane change model with relaxation and synchronization," *Transportation Research Record*, vol. 2316, no. 1, pp. 47–57, 2012. DOI: 10.3141/2316-06.

[66] B. Khelfa, I. Ba, and A. Tordeux, "Predicting highway lane-changing maneuvers: A benchmark analysis of machine and ensemble learning algorithms," *Physica A: Statistical Mechanics and its Applications*, vol. 612, p. 128 471, 2023. DOI: 10.1016/j.physa.2023.128471.

[67] J. Gong, Y. Xu, C. Lu, and G. Xiong, "Decision-making model of overtaking behavior for automated driving on freeways," in *Proceedings IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2016, pp. 1–6. DOI: 10.1109/ICVES.2016.7548162.

[68] L. Claussmann, A. Carvalho, and G. Schildbach, "A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams," in *Proceedings European Control Conference (ECC)*, 2015, pp. 2976–2982. DOI: 10.1109/ECC.2015.7330990.

[69] P. de Beaucorps, T. Streubel, A. Verroust-Blondet, F. Nashashibi, B. Bradai, and P. Resende, "Decision-making for automated vehicles at intersections adapting human-like behavior," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 212–217.

[70] H. Weinreuter, B. Szigeti, N.-R. Strelau, B. Deml, and M. Heizmann, "Decision making at unsignalized inner city intersections using discrete events systems," *tm - Technisches Messen*, vol. 89, no. 2, pp. 134–146, 2022. DOI: 10.1515/teme-2021-0140.

[71] S. Noh, "Decision-making framework for autonomous driving at road intersections: Safeguarding against collision, overly conservative behavior, and violation vehicles," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3275–3286, 2019. DOI: 10.1109/TIE.2018.2840530.

[72] D. González, J. Pérez, and V. Milanés, "Parametric-based path generation for automated vehicles at roundabouts," *Expert Systems with Applications*, vol. 71, no. 1, pp. 332–341, 2017. DOI: 10.1016/j.eswa.2016.11.023.

[73] L. Labakhua, U. Nunes, R. Rodrigues, and F. S. Leite, "Smooth trajectory planning for fully automated passengers vehicles: Spline and clothoid based methods and its simulation," in *Informatics in Control Automation and Robotics*. Springer Berlin Heidelberg, 2008, pp. 169–182. DOI: 10.1007/978-3-540-79142-3_14.

[74] J. A. R. Silva and V. Grassi, "Path planning at roundabouts using piecewise linear continuous curvature curves," in *Proceedings Latin American Robotics Symposium (LARS) and Brazilian Symposium on Robotics (SBR)*, 2017, pp. 1–6. DOI: 10.1109/SBR-LARS-R.2017.8215288.

[75] M. Rodrigues, A. McGordon, G. Gest, and J. Marco, "Autonomous navigation in interaction-based environments—a case of non-signalized roundabouts," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 425–438, 2018. DOI: 10.1109/TIV.2018.2873916.

[76] C. Hidalgo, R. Lattarulo, J. Pérez, and E. Asua, "Hybrid trajectory planning approach for roundabout merging scenarios," in *Proceedings IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–6. DOI: 10.1109/ICCVE45908.2019.8964985.

[77] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts," in *Proceedings IEEE Conference on Decision and Control (CDC)*, 2018, pp. 321–326. DOI: 10.1109/CDC.2018.8619275.

[78] T. Stanger and L. del Re, "A model predictive cooperative adaptive cruise control approach," in *Proceedings American Control Conference*, 2013, pp. 1374–1379. DOI: 10.1109/ACC.2013.6580028.

[79] A. Danesh, W. Ma, C. Yu, R. Hao, and X. Ma, "Optimal roundabout control under fully connected and automated vehicle environment," *IET Intelligent Transport Systems*, vol. 15, no. 11, pp. 1440–1453, 2021. DOI: 10.1049/itr2.12117.

[80] B. Filocamo, J. A. Ruiz, and M. A. Sotelo, "Efficient management of road intersections for automated vehicles - the frfp system applied to the various types of intersections and roundabouts," *Applied Sciences*, vol. 10, no. 1, pp. 1–25, 2020. DOI: 10.3390/app10010316.

[81] A. S. M. Bakibillah, M. A. S. Kamal, C. P. Tan, S. Susilawati, T. Hayakawa, and J.-i. Imura, "Bi-level coordinated merging of connected and automated vehicles at roundabouts," *Sensors*, vol. 21, no. 19, pp. 1–15, 2021. DOI: 10.3390/s21196533.

[82] D. HeB, R. Lattarulo, J. Pérez, J. Schindler, T. Hesse, and F. Koster, "Fast maneuver planning for cooperative automated vehicles," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, 1625 –1632. DOI: 10.1109/ITSC.2018.8569791.

[83] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015. DOI: 10.1016/j.trc.2015.09.011.

[84] J. Bohren, T. Foote, J. Keller, *et al.*, "Little ben: The ben franklin racing team's entry in the 2007 darpa urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 598–614, 2008. DOI: 10.1002/rob.20260.

[85] A. Bacha, C. Bauman, R. Faruque, *et al.*, "Odin: Team victortango's entry in the darpa urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 467–492, 2008. DOI: 10.1002/rob.20248.

[86] G. Tanzmeister, D. Wollherr, and M. Buss, "Grid-based multi-road-course estimation using motion planning," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 1924–1935, 2016. DOI: 10.1109/TVT.2015.2420752.

[87] C. Zhang, D. Chu, S. Liu, Z. Deng, C. Wu, and X. Su, "Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 2, pp. 29–40, 2019. DOI: 10.1109/MITS.2019.2903536.

[88] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009. DOI: 10.1002/rob.20285.

[89] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4889–4895. DOI: 10.1109/ICRA.2011.5980223.

[90] T. Gu, J. Snider, J. M. Dolan, and J.-w. Lee, "Focused trajectory planning for autonomous on-road driving," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 547–552. DOI: 10.1109/IVS.2013.6629524.

[91] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996. DOI: 10.1109/70.508439.

[92] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011. DOI: 10.1177/0278364911406761.

[93] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009. DOI: 10.1109/TCST.2008.2012116.

[94] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *Proceedings IEEE Conference on Decision and Control (CDC)*, 2010, pp. 7681–7687. DOI: 10.1109/CDC.2010.5717430.

[95] J. hwan Jeon, R. V. Cowlagi, S. C. Peters, *et al.*, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proceedings American Control Conference (ACC)*, 2013, pp. 188–193. DOI: 10.1109/ACC.2013.6579835.

[96] X. Lan and S. Di Cairano, "Continuous curvature path planning for semi-autonomous vehicle maneuvers using rrt," in *Proceedings European Control Conference (ECC)*, 2015, pp. 2360–2365. DOI: 10.1109/ECC.2015.7330891.

[97] C. Guo, K. Kidono, R. Terashima, and Y. Kojima, "Humanlike behavior generation in urban environment based on learning-based potentials with a low-cost lane graph," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 46–60, 2018. DOI: 10.1109/TIV.2017.2788194.

[98] J. Funke, P. Theodosis, R. Hindiyeh, *et al.*, "Up to the limits: Autonomous audi tts," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2012, pp. 541–547. DOI: 10.1109/IVS.2012.6232212.

[99] H. Mouhagir, V. Cherfaoui, R. Talj, F. Aioun, and F. Guillemard, "Using evidential occupancy grid for vehicle trajectory planning under uncertainty with tentacles," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–7. DOI: 10.1109/ITSC.2017.8317808.

[100] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, 346–359, 2012. DOI: 10.1177/0278364911423042.

[101] H. Tehrani, Q. Huy Do, M. Egawa, K. Muto, K. Yoneda, and S. Mita, "General behavior and motion model for automated lane change," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1154–1159. DOI: 10.1109/IVS.2015.7225839.

[102] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *Proceedings International Conference on Intelligent Robotics and Applications*, 2012, pp. 588–597. DOI: 10.1007/978-3-642-33503-7_57.

[103] L. Zheng, P. Zeng, W. Yang, Y. Li, and Z. Zhan, "Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1882–1891, 2020. DOI: 10.1049/iet-its.2020.0355.

[104] A. Artuñedo, J. Villagra, and J. Godoy, "Real-time motion planning approach for automated driving in urban environments," *IEEE Access*, vol. 7, 180039 –180053, 2019. DOI: 10.1109/ACCESS.2019.2959432.

[105] R. Lattarulo, L. González, E. Martí, J. Matute, M. Marcano, and J. Pérez, "Urban motion planning framework based on n-bézier curves considering comfort and safety," *Journal of Advanced Transportation*, vol. 2018, 2018. DOI: 10.1155/2018/6060924.

[106] J. Moreau, P. Melchior, S. Victor, M. Moze, F. Aioun, and F. Guillemard, "Reactive path planning for autonomous vehicle using bézier curve optimization," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1048–1053. DOI: 10.1109/IVS.2019.8813904.

[107] I. Bae, J. H. Kim, J. Moon, and S. Kim, "Lane change maneuver based on bezier curve providing comfort experience for autonomous vehicle users," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 2272–2277. DOI: 10.1109/ITSC.2019.8916845.

[108] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2997–3004, 2019. DOI: 10.1109/LRA.2019.2923954.

[109] A. Artuñedo, J. Villagra, J. Godoy, and M. D. d. Castillo, "Motion planning approach considering localization uncertainty," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5983–5994, 2020. DOI: 10.1109/TVT.2020.2985546.

[110] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical Systems and Signal Processing*, vol. 100, pp. 482–500, 2018, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2017.07.019.

[111] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 740–753, 2016. DOI: 10.1109/TMECH.2015.2493980.

[112] Y. Meng, Y. Wu, Q. Gu, and L. Liu, "A decoupled trajectory planning framework based on the integration of lattice searching and convex optimization," *IEEE Access*, vol. 7, 130 530–130 551, 2019. DOI: 10.1109/ACCESS.2019.2940271.

[113] L. Claussmann, M. O'Brien, S. Glaser, H. Najjaran, and D. Gruyer, "Multi-criteria decision making for autonomous vehicles using fuzzy dempster-shafer reasoning," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2018, 2195 –2202. DOI: 10.1109/IVS.2018.8500451.

[114] A. Artuñedo, J. Villagra, and J. Godoy, "Jerk-limited time-optimal speed planning for arbitrary paths," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8194–8208, 2022. DOI: 10.1109/TITS.2021.3076813.

[115] J. J. Martinez and C. Canudas, "A safe longitudinal control for adaptive cruise control and stop-and-go scenarios," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 2, pp. 246–258, 2007, ISSN: 10636536. DOI: 10.1109/TCST.2006.886432.

[116] V. Milanes, J. Villagra, J. Godoy, and C. Gonzalez, "Comparing fuzzy and intelligent pi controllers in stop-and-go manoeuvres," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 770–778, 2012. DOI: 10.1109/TCST.2011.2135859.

[117] F. Damerow and J. Eggert, "Risk-aversive behavior planning under multiple situations with uncertainty," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015, pp. 656–663. DOI: 10.1109/ITSC.2015.113.

[118] J. Li, X. Xie, H. Ma, X. Liu, and J. He, *Speed planning using bezier polynomials with trapezoidal corridors*, 2021. DOI: 10.48550/ARXIV.2104.11655.

[119] M. Wang, Z. Wang, L. Zhang, and D. G. Dorrell, "Speed planning for autonomous driving in dynamic urban driving scenarios," in *Proceedings IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020, pp. 1462–1468. DOI: 10.1109/ECCE44975.2020.9235659.

[120] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, "Simultaneous trajectory planning and tracking using an mpc method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4273–4283, 2018. DOI: 10.1109/TII.2018.2815531.

[121] X. Tang, K. Yang, H. Wang, *et al.*, "Driving environment uncertainty-aware motion planning for autonomous vehicles," *Chinese Journal of Mechanical Engineering*, vol. 35, no. 1, p. 120, 2022. DOI: 10.1186/s10033-022-00790-5.

[122] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2510–2525, 2019. DOI: 10.1109/TCST.2018.2871397.

[123] R. Lattarulo and J. Pérez Rastelli, "A hybrid planning approach based on mpc and parametric curves for overtaking maneuvers," *Sensors*, vol. 21, no. 2, 2021. DOI: 10.3390/s21020595.

[124] S. Yu, E. Sheng, Y. Zhang, Y. Li, H. Chen, and Y. Hao, "Efficient nonlinear model predictive control of automated vehicles," *Mathematics*, vol. 10, no. 21, 2022. DOI: 10.3390/math10214163.

[125] A. Reda, A. Bouzid, and J. Vásárhelyi, "Deep learning-based automated vehicle steering," in *Proceedings International Carpathian Control Conference (ICCC)*, 2021, pp. 1–5. DOI: 10.1109/ICCC51557.2021.9454613.

[126] M. Baltzer, E. Altendorf, S. Meier, and F. Flemisch, "Mediating the interaction between human and automation during the arbitration processes in cooperative guidance and control of highly automated vehicles: Basic concept and first study," *Advances in Human Aspects of Transportation Part I*, pp. 439–450, 2014.

[127] J. C. F. de Winter, S. M. Petermeijer, and D. A. Abbink, "Shared control versus traded control in driving: A debate around automation pitfalls," *Ergonomics*, vol. 0, no. 0, pp. 1–27, 2022. DOI: 10.1080/00140139.2022.2153175.

[128] T. B. Sheridan, "Automation, authority and angst-revisited," in *Proceedings of the Human Factors Society Annual Meeting*, 1991, pp. 2–6. DOI: 10.1177/15419312910350010.

[129] Z. Lu, R. Happee, C. D. Cabrall, M. Kyriakidis, and J. C. de Winter, "Human factors of transitions in automated driving: A general framework and literature survey," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 43, pp. 183–198, 2016, ISSN: 1369-8478. DOI: 10.1016/j.trf.2016.10.007.

[130] D. A. Abbink, T. Carlson, M. Mulder, *et al.*, "A topology of shared control systems—finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018. DOI: 10.1109/THMS.2018.2791570.

[131] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 475–491, 2020. DOI: 10.1109/THMS.2020.3017748.

[132] M. R. van Paassen, R. P. Boink, D. A. Abbink, M. Mulder, and M. Mulder, "Four design choices for haptic shared control," in *Advances in Aviation Psychology, Volume 2*, Routledge, 2017, pp. 237–254.

[133] C. Sentouh, A. T. Nguyen, M. A. Benloucif, and J. C. Popieul, "Driver-automation cooperation oriented approach for shared control of lane keeping assist systems," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 1962–1978, 2019. DOI: 10.1109/TCST.2018.2842211.

[134] M. Li, X. Song, H. Cao, *et al.*, "Shared control with a novel dynamic authority allocation strategy based on game theory and driving safety field," *Mechanical Systems and Signal Processing*, vol. 124, pp. 199–216, 2019. DOI: 10.1016/j.ymssp.2019.01.040.

[135] B. Ma, Y. Liu, X. Na, Y. Liu, and Y. Yang, "A shared steering controller design based on steer-by-wire system considering human-machine goal consistency," *Journal of the Franklin Institute*, vol. 356, no. 8, pp. 4397–4419, 2019. DOI: 10.1016/j.jfranklin.2018.12.028.

[136] T. Melman, J. de Winter, and D. Abbink, "Does haptic steering guidance instigate speeding? a driving simulator study into causes and remedies," *Accident Analysis and Prevention*, vol. 98, pp. 372–387, 2017, ISSN: 0001-4575. DOI: 10.1016/j.aap.2016.10.016.

[137] S. Petermeijer, D. Abbink, and J. de Winter, "Should drivers be operating within an automation-free bandwidth?: Evaluating haptic steering support systems with different levels of authority," *Human Factors: the journal of the human factors and ergonomics society*, vol. 57, no. 1, pp. 5–20, 2015. DOI: 10.1177/0018720814563602.

[138] D. G. Kidd, J. B. Cicchino, I. J. Reagan, and L. B. Kerfoot, "Driver trust in five driver assistance technologies following real-world use in four production vehicles," *Traffic Injury Prevention*, vol. 18, no. sup1, S44–S50, 2017. DOI: 10.1080/15389588.2017.1297532.

[139] H. Muslim and M. Itoh, "A theoretical framework for designing human centered automotive automation systems," *Cognition, Technology & Work*, vol. 21, no. 4, pp. 685–697, 2019. DOI: 10.1007/s10111-018-0509-8.

[140] T. Inagaki and T. B. Sheridan, "A critique of the SAE conditional driving automation definition, and analyses of options for improvement," *Cognition, Technology and Work*, 2018. DOI: 10.1007/s10111-018-0471-5.

[141] D. A. Drexler, A. Takacs, T. D. Nagy, P. Galambos, I. J. Rudas, and T. Haidegger, "Situation awareness and system trust affecting handover processes in self-driving cars up to level 3 autonomy," in *Proceedings IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, 2019, pp. 000 179–000 184. DOI: 10.1109/IWOBI47054.2019.9114533.

[142] F. Biondi, I. Alvarez, and K.-A. Jeong, "Human–vehicle cooperation in automated driving: A multidisciplinary review and appraisal," *International Journal of Human–Computer Interaction*, vol. 35, no. 11, pp. 932–946, 2019. DOI: 10.1080/10447318.2018.1561792.

[143] C. Gold, D. Damböck, L. Lorenz, and K. Bengler, ""take over!" how long does it take to get the driver back into the loop?" *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 1938–1942, 2013. DOI: 10.1177/1541931213571433.

[144] E. Dogan, M.-C. Rahal, R. Deborne, P. Delhomme, A. Kemeny, and J. Perrin, "Transition of control in a partially automated vehicle: Effects of anticipation and non-driving-related task involvement," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 46, pp. 205–215, 2017. DOI: 10.1016/j.trf.2017.01.012.

[145] J. Radlmayr, C. Gold, L. Lorenz, M. Farid, and K. Bengler, "How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving," *Proceedings Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 2063–2067, 2014. DOI: 10.1177/1541931214581434.

[146] M. Bahram, M. Aeberhard, and D. Wollherr, "Please take over! an analysis and strategy for a driver take over request during autonomous driving," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 913–919. DOI: 10.1109/IVS.2015.7225801.

[147] F. O. Flemisch, C. A. Adams, S. R. Conway, K. H. Goodrich, M. T. Palmer, and P. C. Schutte, "The h-metaphor as a guideline for vehicle automation and interaction," NASA, Tech. Rep., 2003.

[148] P. Lindemann, T.-Y. Lee, and G. Rigoll, "Supporting driver situation awareness for autonomous urban driving with an augmented-reality windshield display," in *Proceedings IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2018, pp. 358–363. DOI: 10.1109/ISMAR-Adjunct.2018.00104.

[149] K. Sonoda and T. Wada, "Displaying system situation awareness increases driver trust in automated driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 185–193, 2017. DOI: 10.1109/TIV.2017.2749178.

[150] AutoMate-project, *Grant agreement id: 690705, h2020-eu.3.4.* 2019. [Online]. Available: http://www.automate-project.eu/.

[151] Vi-DAS-project, *Grant agreement id: 690772, h2020-eu.* 2019. [Online]. Available: http://vi-das.eu/.

[152] ADAS&ME-project, *Grant agreement id: 688900, h2020-eu.* 2019. [Online]. Available: http://www.adasandme.com/.

[153] A. Domina and V. Tihanyi, "Ltv-mpc approach for automated vehicle path following at the limit of handling," *Sensors*, vol. 22, no. 15, 2022. DOI: 10.3390/s22155807.

[154] A. Artuñedo, J. Godoy, and J. Villagra, "A decision-making architecture for automated driving without detailed prior maps," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1645–1652. DOI: 10.1109/IVS.2019.8814070.

[155] F. Poggenhans, J.-H. Pauls, J. Janosovits, *et al.*, "Lanelet2: A high-definition map framework for the future of automated driving," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, 1672–1679. DOI: 10.1109/ITSC.2018.8569929.

[156] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973. DOI: 10.3138/FM57-6770-U75U-7727.

[157] A. Artuñedo, J. Godoy, and J. Villagra, "A primitive comparison for traffic-free path planning," *IEEE Access*, vol. 6, pp. 28 801–28 817, 2018. DOI: 10.1109/ACCESS.2018.2839884.

[158] L. Svensson and J. Eriksson, *Tuning for ride quality in autonomous vehicle: Application to linear quadratic path planning algorithm*, 2015.

[159] A. Blake, A. Bordallo, K. Brestnichki, *et al.*, "FPR - Fast Path Risk Algorithm to evaluate collision probability," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, 2019. DOI: 10.1109/LRA.2019.2943074.

[160] J. Eggert and T. Puphal, "Continuous risk measures for adas and ad," in *Proceedings Future Active Safety Technology Symposium*, 2017, pp. 1–8.

[161] K. P. Yoon and C.-L. Hwang, *Multiple attribute decision making: An introduction*. SAGE Publications, Inc; N.º 1 edition, 1995.

[162] A. Breuer, J.-A. Termöhlen, S. Homoceanu, and T. Fingscheidt, "Opendd: A large-scale roundabout drone dataset," in *Proceedings IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294301.

[163] N. Druml, A. Ryabokon, R. Schorn, *et al.*, "Programmable systems for intelligence in automobiles (prystine): Final results after year 3," in *Proceedings Euromicro Conference on Digital System Design (DSD)*, 2021, pp. 268–277. DOI: 10.1109/DSD53832.2021.00049.

[164] A. S. Huang, E. Olson, and D. Moore, "Lightweight communications and marshalling for low latency interprocess communication," *Computer Science and Artificial Intelligence Laboratory Technical Report, Massachusetts Institute of Technology, MA*, 2009.

[165] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4057–4062. DOI: 10.1109/IROS.2010.5649358.

[166] J. Godoy, V. Jiménez, A. Artuñedo, and J. Villagra, "A grid-based framework for collective perception in autonomous vehicles," *Sensors*, vol. 21, no. 3, 2021. DOI: 10.3390/s21030744.

[167] V. Trentin, A. Artuñedo, J. Godoy, and J. Villagra, "Multi-modal interaction-aware motion prediction at unsignalized intersections," *IEEE Transactions on Intelligent Vehicles*, pp. 1–17, 2023. DOI: 10.1109/TIV.2023.3254657.

[168] *Scaner*, 2023. [Online]. Available: https://www.avsimulation.com/scaner-studio/.

[169] Prystine-project, *Grant agreement id: 783190, h2020-eu.* 2021. [Online]. Available: https://prystine.eu/.

[170] N. Control-project, *Grant agreement id: 826653, h2020-eu.* 2023. [Online]. Available: https://www.newcontrol-project.eu/.

[171] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934. DOI: 10.1109/IV47402.2020.9304839.