EUROPEAN COMMISSION

SEVENTH FRAMEWORK PROGRAMME

Theme: ICT

Small or medium-scale focused research projects (STREP)

FP7-ICT-2013-10

Objective ICT-2013.6.5 Co-operative mobility
a) Supervised Automated Driving

GA No. 612035

Interoperable GCDC AutoMation Experience

Deliverable No.	i-GAME D2.2	
Deliverable Title	Generic real-time control system	
Dissemination level	Public	
Written By	Alejandro Morales Medina (TU/e)	15-02-2015
	Elham Semsar-Kazerooni (TNO)	15-02-2015
Checked by	Jeroen Ploeg (TNO)	20-02-2015
	Nathan van de Wouw (TU/e)	18-02-2015
	Hoai Hoang Bengtsson (Viktoria Swedish ICT)	10-03-2015
	-	-
Approved by	Almie van Asten (TNO)	01-04-2015
Status	Final	18-03-2015

Please refer to this document as:

DEL150318_i-GAME_D2.2 Generic real-time control system_FINAL

Acknowledgment:

Disclaimer:

i-GAME is co-funded by the European Commission, DG Research and Innovation, in the 7th Framework Programme. The contents of this publication is the sole responsibility of the project partners involved in the present activity and do not necessarily represent the view of the European Commission and its services nor of any of the other consortium partners.

Executive summary

This document presents the generic real-time control system designed for the benchmark vehicles within the iGame project. The benchmark vehicles should be able to perform two different scenarios: a platoon merging scenario (scenario 1) and an intersection crossing scenario (scenario 2). It is worth mentioning that there is also an emergency vehicle scenario, where two platoons of vehicles are informed about an approaching emergency vehicle, such that the platoons can make space for the emergency vehicle to pass. This scenario is for demonstration purposes only and, therefore, there is no control system involved.

Each scenario has its particular design approach but both share the same control architecture implemented in the benchmark vehicles which consists of:

- Perception layer: it tracks the host and target vehicles, and identifies the Most Important Objects for each scenario.
- Control layer: it gives the vehicle all its functionalities such as Cooperative Adaptive Cruise Control (CACC), Cruise Control (CC), Obstacle Avoidance (OA), Collision Avoidance (CA), Lateral Control (LC), and Virtual CACC (VCACC).
- Supervisory layer: it determines the functionality to be used by the vehicle based on the Interaction Protocol (presented in Deliverable 2.1).

The design approach for scenario 1 combines different vehicle functionalities (with their respective controllers) to perform a desired maneuver. For instance a vehicle making a gap, inside the platoon, will use CACC to follow the vehicle in front and OA to avoid the vehicle that intends to enter the platoon; the combined action of both functionalities achieves the desired behavior. So, in general, for each maneuver a determined combination of functionalities will be used to perform the required tasks.

The design approach for scenario 2 defines platoons of vehicles driving on different lanes of the intersection; these platoons are referred as virtual platoons. To form a virtual platoon, a virtual inter-vehicle distance is defined and then fed to the CACC; this combination is referred as VCACC. These virtual platoons allow the vehicles to be at a safe distance from each other at the center of the intersection, which allows the vehicles to cross in a safe manner.

The functionality and effectiveness of both design approaches are demonstrated by simulations.

The solutions presented in this document, and to be implemented in the benchmark vehicles, are guidelines for the GCDC participants on how the iGame scenarios can be implemented. Part of the presented control solutions must be adopted by the participants to guarantee a correct execution of the scenarios, whereas other portions of the control solutions may be freely chosen. This is explicitly indicated in the document.

Contents

1 Introduction

1.1 i-Game project

The i-GAME project is meant to facilitate development and real-life implementation of automated driving with a focus on cooperation done through communication between the vehicles and between vehicles and road-side equipments.

Within this project several work packages are defined. The focus of work package two (WP2) is on design, implementation, and testing of a reliable, fail-safe supervisory control system, incorporating both vehicles and roadside infrastructure [1]. In this work package, the entire control system including the supervisory layer as well as the real time controller should be designed such that the three scenarios defined and proposed in WP1 are executable by the benchmark vehicles to be developed by TNO and TU/e. These scenarios will be explained in the upcoming sections. In addition to the control system design, for verification purposes, in this work package a simulation toolset should be designed. Using a set of benchmark vehicles, the envisioned developments in WP2 will be thoroughly validated in practice in WP4. Moreover, WP2 should provide WP3 with the required elements of the interaction protocol, as part of the supervisory control system, which are needed for the developments in WP3. Finally, WP2 aims to support WP5 by means of dissemination of the results to the teams. It should be mentioned that the proposed solutions in the current document are meant to be implemented to the benchmark vehicles and the participants do not have to comply to the entire setup. At the end of this document, the necessary parts to be complied by the participants are highlighted.

1.2 i-Game scenarios

In total, three scenarios have been defined by the i-Game organizers. Among these three, two are considered as part of the competition, whereas one is for demonstration purposes, only. These scenarios are described in the following sections.

1.2.1 Description of Scenario 1

Scenario 1 is described in [2] as: Two platoons are approaching a construction site on a highway. The left platoon (A) and the right platoon (B) receive a message from a RSU saying that they are approaching a construction site with information about the position of and the speed limit on the construction site. The participating vehicles must merge the two platoons into the available lane for passing the site. This merging operation should take place as late as possible and still before the Competition Zone (CZ).

This scenario is visualized in Figure 1.

1.2.2 Description of Scenario 2

In this scenario, described in [2], three cooperative vehicles approach a T-intersection (Figure 2). All three competing vehicles pass the line of the Competition Zone (CZ) at exactly the same time with the same velocity, and are not allowed to respond to communication from other competing cars before this time. Once all the vehicles have passed into the CZ (front of the vehicle), they will collaborate to allow V1 to enter the main road first and in a safe manner (Figure 3). When V1 is on the main road after taking a left turn, all vehicles accelerate to cross and leave the intersection as soon as possible. The scenario ends when the last of the three vehicles (front of the vehicles) has left the CZ (Figure 4).

All vehicles negotiate a speed that is optimal (to complete the scenario as fast as possible) allowing passage for V1 without stopping or violating a minimal safety (Euclidean) distance between vehicles.

1.2.3 Description of Scenario 3

This scenario, shown in Figure 5, is described in [2] as: “An emergency vehicle (EV) is approaching a congested traffic situation and signals the intent to pass the traffic congestion

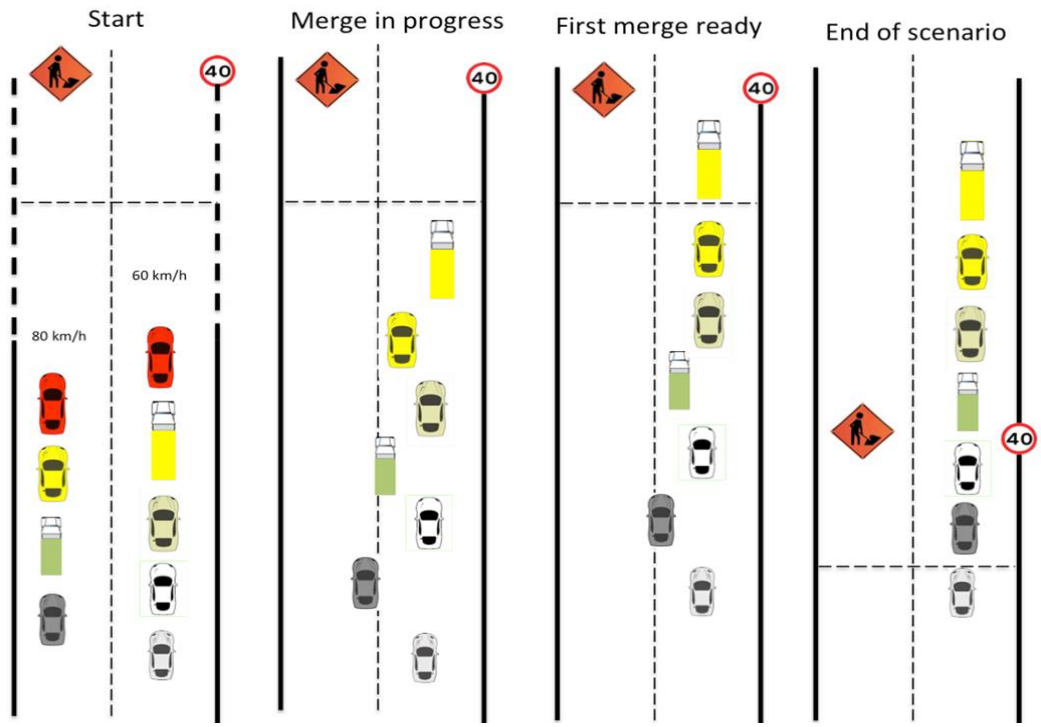

Figure 1: Description of scenario 1 of i-Game.

in a given position (left / middle / right side). The cooperating vehicles know at which time the EV will be close and act in a cooperative manner to create room for the EV. Vehicles continue with reduced speed during the maneuver. When the EV has passed the vehicles resume position within their lanes and resume speed." Execution of this scenario is not part of the competition but is meant for demonstration purposes only.

1.3 Description of the tasks and role of involved parties

This document, i.e.\Deliverable 2.2, should reflect the results and progress of Task 2.2 as defined in [1]: The focus of this task is the development and verification of a real-time control strategy which objective is to robustly, safely, and comfortably steer and drive a vehicle, in a cooperative manner with both road side and neighboring vehicles. To achieve such objective it is necessary to distinguish a generic upper control level (or supervisory level) and a vehicle-specific lower control level. The upper control level interprets the inputs from

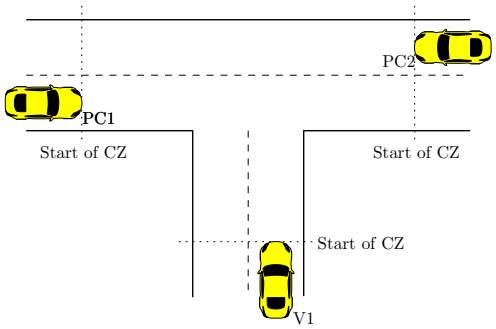

Figure 2: Description of scenario 2.

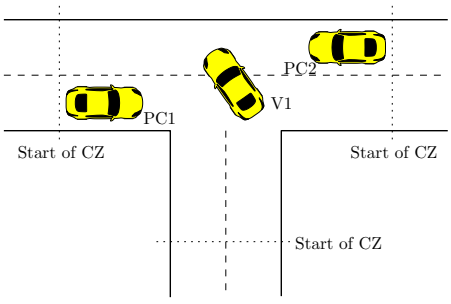

Figure 3: V1 entering the main road in scenario 2.

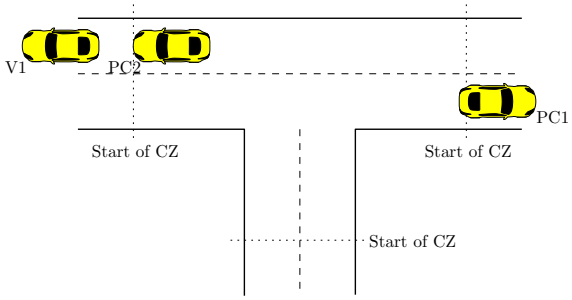

Figure 4: End of the scenario 2.

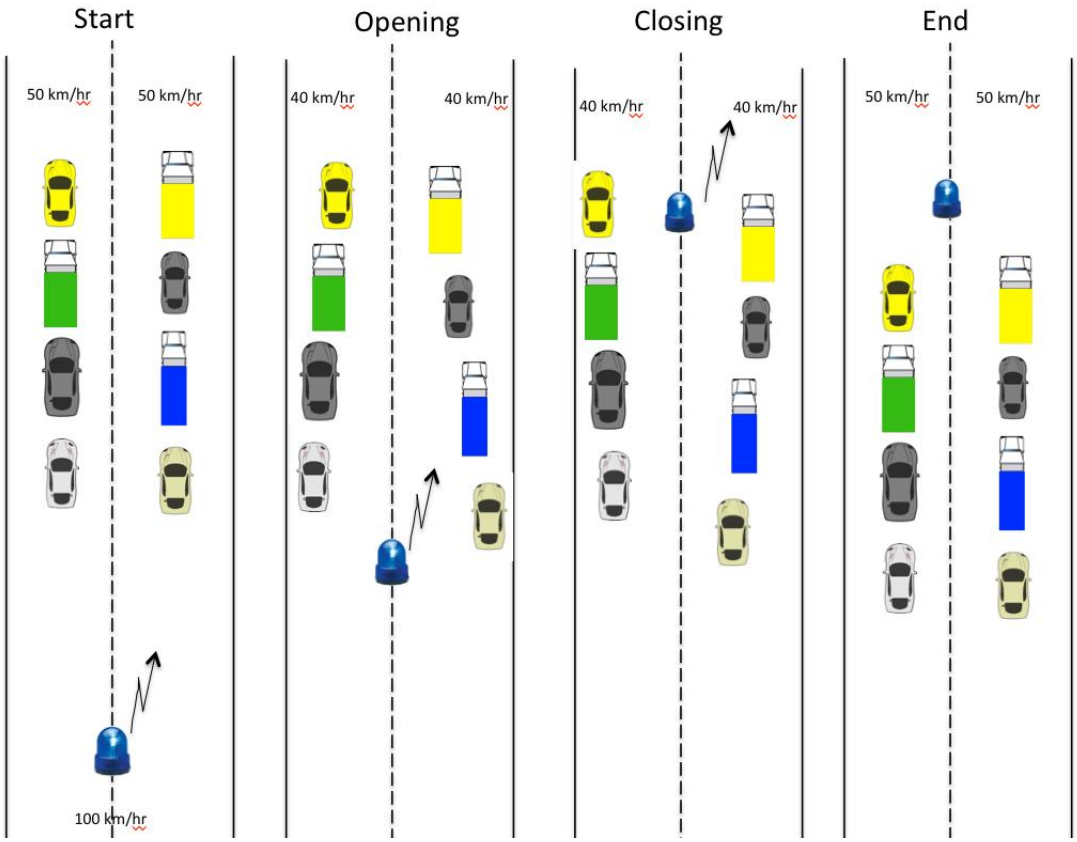

Figure 5: Description of scenario 3 of i-Game.

the interaction protocol and generates reference signals; which are used by the lower level control to achieve the desired dynamic behavior of the vehicle.

The aim of this task is to develop the supervisory real-time control system for the vehicles so that – no matter how the individual vehicle has defined its low-level steer/acceleration controller — the basic use cases described in [1] can be successfully executed. A (simulation) tool to evaluate the operation of the supervisory real-time controllers is one of the objectives here.

The performance of this task is led by TNO, integrated vehicle safety department, and is supported by TU/e and Viktoria Swedish ICT.

2 Implemented control architecture

The proposed control system architecture to be implemented to benchmark vehicles is shown in Figure 6. This system has the following layered architecture:

- Perception layer: includes host and target tracking algorithms. Also, the perceived objects are "classified"', e.g. as in-lane, left lane, right lane, closest front/back objects, etc.
- Control layer: contains the real-time control functions such as vehicle following, obstacle avoiding, lateral controller, etc.
- Supervisory layer: determines the desired role of the host vehicle, e.g., being a platooning vehicle or a gap-making one, etc., invokes the corresponding message sequence and maneuvers, and tunes vehicle settings, e.g. cruise speed, according to the vehicle's action role.

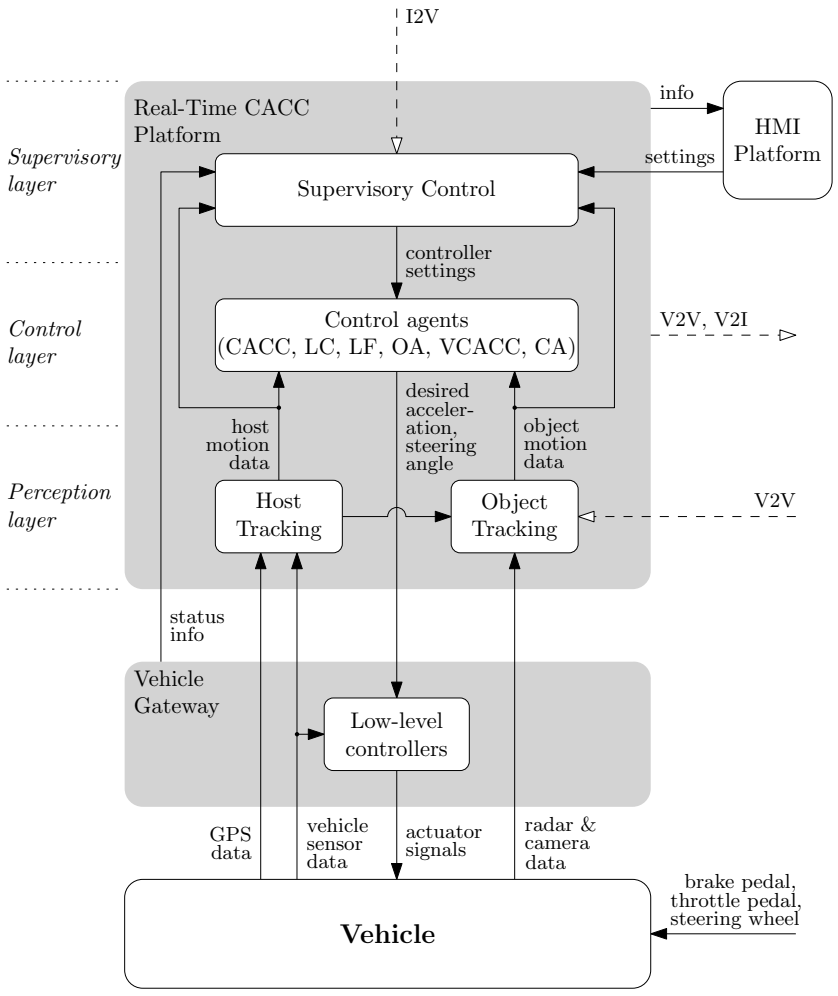

Figure 6: Control system architecture.

In the following exposition, some of the components which are relevant for the current document are explained.

2.1 Environmental perception layer

The environmental perception unit, includes target tracking (TT) as well as host tracking algorithms. Here, we do not go into the details of these algorithms and move on to the

object distinction/classification strategies used in target tracking algorithms.

2.1.1 Distinguished objects for scenario 1 implementation

In this TT algorithm, six objects are tracked and distinguished as the most important objects (MIOs) for the control system. The distinguished (most important) objects are the two objects directly in front, one direct rear object, two front objects on adjacent lanes and one rear object on right lane. This is shown in Figure 7, where 3 lanes of a road are shown and the host (red) vehicle is placed in the middle lane. The objects are labeled as 1, 2, 3, 4, 5 and 6. In this figure, it can be seen that in addition to main-lane front objects, two extra front objects on right and left lanes are classified. These are called adjacent lane MIOs and detected only if there is a maneuvering (i.e. gap making, or merging) situation. Also, the second rear object would be on right lane and is needed for merging maneuver. Similar adjacent lane backward MIO is not defined for the left lane since no merging to the left lane is considered. To be more specific, for a host vehicle merging into the right lane, the set of MIOs is $\{1, 2, 3, 4, 6\}$, for a vehicle which is making a gap, the set is $\{1, 2, 3, 5\}$, and for a vehicle in platoon the set of relevant objects is a subset of $\{1, 2, 3\}$, depending on the adopted vehicle-following control algorithm.

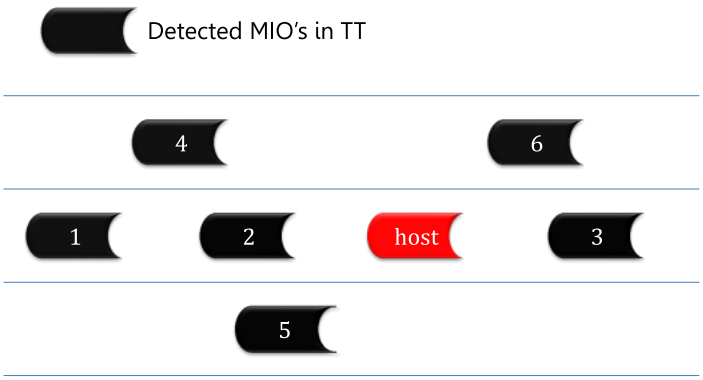

Figure 7: Target classification to be used in a benchmark vehicle for scenario 1.

In terms of nomination, these six objects (1-6) are respectively called, FWD MIOB, FWD MIOA, BWD MIOC, FWD MIOR, FWD MIOL, and BWD MIOR, respectively. The classification of most important objects (MIO) is a two-step process. First, all MIO candidates are selected based on their relative positions with respect to the host vehicle. That is, the object must be in front of the host vehicle for all the forward MIOs. In addition, it should be in adjacent lanes for the forward adjacent-lane MIOs. Similarly, the object should be at the rear for the backward MIOs. In addition, the rear object should be at the right lane for the backward MIOR. Also, the object must move in the same direction as the host, or stand still. However, since this set of MIO candidates is rather large, for computational efficiency, a subset of MIO candidates are selected based on the smallest Euclidean distance to the host vehicle. If two or more MIO candidates have the same smallest distance, a selection is made based on the bearing angle. For forward adjacent lane MIOs, i.e. FWD MIOL if merging is from left to right, an additional criterion is in effect. An object is recognized as FWD MIOL (or FWD MIOR), only if it is closer to the host as compared to FWD MIOA. This condition is put in place based on practical reasons, i.e. the adjacent lane MIO is relevant only if it can merge in front of the host (or vice versa). This requires FDW MIOL (or FWD MIOR) to be closer than FWD MIOA with respect to the host.

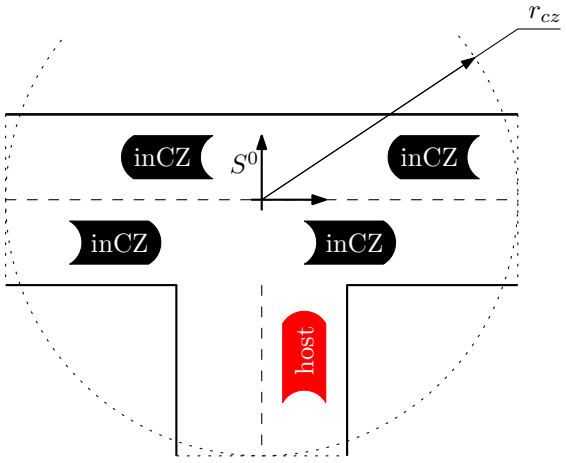

Figure 8: Target classification to be used in a benchmark vehicle for scenario 2.

2.1.2 Distinguished objects for scenario 2 implementation

The most important objects are the vehicles that are inside the Competition Zone (CZ), as shown in Figure 8. When the host vehicle enters the CZ it uses the position of each vehicle, with respect to the Intersection Reference Frame (IRF) S^0, to determine whether the vehicle is inside the CZ or not. The vehicles that are inside the CZ become potential targets to the host vehicle. The assignation of a definite target vehicle is done by the interaction protocol described in Deliverable D2.1.

2.2 Control layer

Before proceeding to the details of the control layer, let us define the dynamical model used for representing a nominal vehicle.

2.2.1 Dynamical model of a vehicle

We will use two vehicle models: a linearized model for the longitudinal behavior and a nonlinear unicycle model describing the lateral behavior of the vehicles.

The longitudinal model considers two vehicles traveling on the same lane. The linearized model for vehicle i, $i = 1, \ldots, m$, is given by

$$\begin{aligned} \dot{\delta}_i &= v_{i-1} - v_i \\ \dot{v}_i &= a_i \\ \dot{a}_i &= -\frac{1}{\tau} a_i + \frac{1}{\tau} u_i^x, \end{aligned} \tag{2.1}$$

where $\delta_i = q_{i-1} - q_i$ is the inter-vehicle distance with q_i being the absolute vehicle position, v_{i-1} the target vehicle velocity, v_i the host vehicle velocity, a_i the host vehicle acceleration, τ a time constant related to the vehicle's driveline dynamics, and u_i^x being the desired longitudinal acceleration input.

The nonlinear unicycle model behind the representation shown in Figure 9 is given by

$$\begin{aligned} \dot{x}_i &= v_i \cos \alpha_i \\ \dot{y}_i &= v_i \sin \alpha_i \\ \dot{\alpha}_i &= u_i^y \end{aligned} \tag{2.2}$$

where x_i and y_i are the coordinates of the host vehicle with respect to the frame $S^0 = \{O^0, [\vec{e}_x^{\,0}\ \vec{e}_y^{\,0}]^T\}$, α_i is the orientation of the vehicle with respect to the the frame S^0, and u_i^y is the desired yaw rate input.

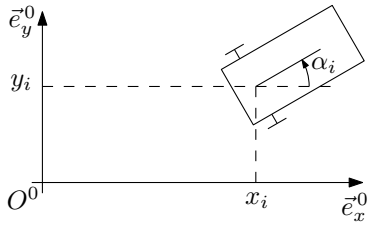

Figure 9: Unicycle representation.

2.2.2 Real-time control system

In order to implement i-Game scenarios to benchmark vehicles, we need to have appropriate real time controllers. These controllers should be capable of controlling vehicles both in lateral and longitudinal directions.

In developing the real-time control system for benchmark vehicles, an agent-based approach is employed to be able to decompose the complex control tasks into the simpler ones. In particular, the design is based on defining different agents for different functionalities envisaged for the platoon members, e.g., agents representing vehicle following, obstacle avoidance, lane change, intersection control, etc. The major control components are defined as:

- Cooperative Adaptive Cruise Control (CACC), which is an inter-vehicle distance controller that ensures the inter-vehicle distance δ_i converges to a reference distance. This controller also includes a cruise controller (CC), which is a velocity controller that ensures the vehicle velocity v_i converges to a reference (cruise) velocity.
- Obstacle avoidance (OA) controller, ensures that the required gap for merging of an agent within the platoon is made.
- Lateral controller which acts in two situations; first to guarantee lateral platooning goal (vehicle following or lane keeping) and second to execute the lateral motion needed to perform the maneuvers (i.e. lane change).
- Collision avoidance (CA) controller with the objective to stop the vehicle if it gets too close to any object. Note that the CA controller is needed for safety reasons and is not used in nominal operation.
- Virtual Cooperative Adaptive Cruise Control (VCACC), which defines a virtual inter-vehicle distance between vehicles traveling on different lanes of an intersection, and uses CACC to make sure that this virtual inter-vehicle converges to the reference distance.

For sake of implementation of i-Game scenarios, we are specifically interested in CACC/CC, OA, and CIC controllers. Since, CACC/CC is used in both scenarios 1 and 2, it is explained in details in the following subsection. The other two controllers are designed for implementation of scenarios 1 and 2, respectively. They will be explained in the corresponding sections focused on design strategies for scenarios 1 and 2.

2.2.3 Cooperative adaptive cruise control (CACC)

This controller, is meant for regulation of the inter-vehicle distances to a velocity-dependent desired distance. We adopt the following policy for the inter-vehicle spacing

$$\delta_{r,i}(t) = r_i + hv_i(t), \quad 2 \leq i \leq m, \tag{2.3}$$

where $\delta_{r,i}$ is the desired inter-vehicle distance (from vehicle i to vehicle $i-1$), h is the so- called time headway, and r_i is the standstill distance between vehicles i and $i-1$. The

primary goal of platooning is to regulate the inter-vehicle distances to $\delta_{r,i}(t)$, i.e.

$$e_i(t) = \delta_i(t) - \delta_{r,i}(t) \to 0, \text{ for } t \to 0. \tag{2.4}$$

This goal is achieved with the help of a cooperative adaptive cruise controller (CACC) [3]. The approach proposed in [3] is based on a one-vehicle look-ahead strategy, where the desired acceleration of the preceding vehicle is used as a feedforward within the controller of the ego vehicle. The CACC control law for vehicle i is given by

$$\dot{u}_{CACC,i}(t) = \frac{1}{h}\left(-u_{CACC,i}(t) + u^x_{i-1}(t) + k_p(e_i) + k_d(\dot{e}_i)\right), \tag{2.5}$$

where k_p and k_d are controller parameters, e_i is defined in (2.4), and $u^x_{i-1}(t)$ is the target (preceding) vehicle input (desired acceleration). This controller yields stable closed-loop dynamics for $k_p > 0$, $k_d > 0$,and $k_d > k_p\tau$. Moreover, in case that no target is found, CACC switches to a cruise controller. The CC control law is given by

$$u_{CC,i}(t) = k_{cc}(v_{CC,i}(t) - v_i(t)) + a_{CC,i}(t), \tag{2.6}$$

where k_{cc} is the control parameter, $v_{CC,i}(t)$ is the reference velocity profile, and $a_{CC,i}(t)$ is a feed-forward acceleration profile.

2.2.4 Lateral control (LC)

This controller is meant for following a trajectory in the intersection scenario (scenario 2). Let us consider the unicycle model described in 2.2. The objective of the Lateral Control (LC) is to make the angle $\theta_i(t)$ track the reference angle $\theta_{LC,i}(t)$. The lateral control law is given by

$$u_{LC,i}(t) = k_{lc}(\theta_{LC,i}(t) - \theta_i(t)) + \omega_{LC,i}(t), \tag{2.7}$$

where k_{lc} is a design constant, $\theta_{LC,i}(t)$ is the reference angle profile, and $\omega_{LC,i}(t)$ is a feed-forward yaw-rate profile.

3 Design approach for scenario 1

The design approach for scenario 1 is based on the definition of agents for different required functionalities. In such an agent-based control design approach, different controllers are assigned to different control goals, e.g. a controller for longitudinal vehicle following and another one aiming at obstacle avoidance, etc. Each of these controllers is called an agent. Then, there should be a strategy to put all these control efforts together. Summation and minimization are just two methods of aggregating these controller agents. In simple terms, if summation is the chosen aggregation method, then the controller can be defined as

$$u = f^1 + f^2 + f^3 + \ldots + f^n, \tag{3.1}$$

where $f^i, i = 1, \ldots, n$, represent different control agents each addressing different control objectives such as longitudinal vehicle following, lane change, or obstacle avoidance. In implementation of i-Game scenarios, we have chosen summation as a means of integration. One reason for such a selection is that, we desire to have CACC functionality active during the entire scenario execution both for safety and performance reasons. The other agents, e.g. obstacle avoiding, will add forces to that of CACC. From this point on we nominate each agent by its functionality, e.g. lane changing agent, etc. Note that not all these control agents need to be present, simultaneously. The major active agent in implementation of scenario 1 of i-Game is the obstacle avoiding controller (OA) which is responsible for making large enough gaps between vehicles to make the merging possible. This controller will be further explained in the following subsections.

3.1 Obstacle avoiding control

As the name describes, this controller agent has an avoidance function when two objects get closer than certain minimal threshold distance. The design concept is to provide a repulsive 'force' when two objects are getting close but no attractive force if the objects are far enough. This concept has been extensively used in several different application domains. To name a few, in flocking/swarming algorithms, the concept of agent-based design as well as designing an obstacle avoiding agent has been used, see [4, 5]. Also, this concept has been extensively used in robot motion planning [6] and design approaches based on artificial potential field/functions [7, 8].

In the current context, we use the repulsive action as a means to achieve a safe spacing between the platoon members, the maneuvering vehicles, and the relevant road users while a maneuver is in progress. To be more specific, the obstacle avoiding (OA) controller agent of a vehicle sees any object on the adjacent lanes as an obstacle. Hence, while a maneuver is in progress, if a vehicle in platoon sees an object on an adjacent lane, it will consider it as an obstacle. Then, the OA controller tries to keep a safe distance to this obstacle. As a result, this controller agent never tries to regulate the distance but only aims to realize a minimum safe distance to relevant objects. In other words, if the initial distance between the host and the adjacent lane object is bigger than a certain threshold, the OA controller agent will not be activated. Also, if the host is not maneuvering and is only part of a simple platoon, it won't see the adjacent lane objects (as part of target tracking policy) and its OA agent will not react to them.

Different obstacle avoidance strategies are proposed in the literature. Many of these strategies are based on defining a potential function which becomes infinitely large, when the distance is too small. Therefore, the corresponding control action, i.e. the gradient of the potential function, would be infinitely large as the distance to obstacle decreases. An example of such a function is the Yukawa potential function [9, 7]:

$$U = \beta \frac{e^{-\alpha\delta}}{\delta} \tag{3.2}$$

where U is the potential function, δ is distance to the obstacle, β is a gain factor, and α indicates the rate of fall off when getting far from the obstacle. The control effort corresponding to Yukawa potential function, i.e. $u_{Yukawa} = \nabla_\delta U = -\beta \frac{\alpha\delta+1}{\delta^2} e^{-\alpha\delta}$, is shown in

Figure 10. This type of control action does not suit the specific application of our interest since the (infinitely) large control force is not implementable. Moreover, we are not using this OA controller as a safety/emergency function. Therefore, for the sake of comfort, we prefer to have a limited control action even though the objects are getting very close. The following OA controller, u_{OA}, provides a finite 'force', i.e. deceleration, at small distances and an exponentially decreasing 'force' as the distance increases:

$$u_{OA} = -\beta(\alpha\delta + 1)e^{-\alpha\delta}, \tag{3.3}$$

where δ is distance to the obstacle, and β is a gain factor which shows the maximum effort that controller provides if the distance goes to zero. Also, α indicates the rate of fall off when getting far from the obstacle. When compared to a pure exponential action, as shown in Figure 10 by $u_{exp} = -\beta e^{-\alpha\delta}$, this controller is more effective in short distances, i.e. provides a more severe action, whereas the fall off rate is exponential. The control effort in (3.3) versus distance to an obstacle is shown in Figure 10. For all three control actions, the parameters are chosen as $\beta = 6, \alpha = 0.3$.

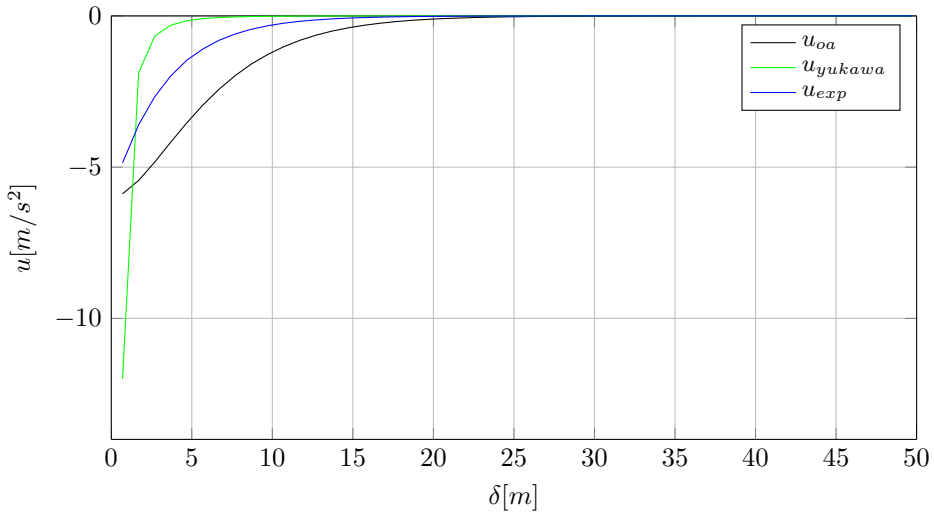

Figure 10: Control effort versus distance to the obstacle for three types of control definition.

Also, for some practical reasons a feed-forward term is included in the obstacle avoiding control action. Hence, for vehicle i, $i = 1, \ldots, m$, the obstacle avoidance control law is defined as

$$u_{OA,i} = -\beta(\alpha\delta_i^o + 1)e^{-\alpha\delta_i^o} + u_{obstacle}, \tag{3.4}$$

where $\delta_i^o = q_{obstacle} - q_i$, is the distance of vehicle i to the obstacle, e.g. the distance between the merging and gap making vehicles in case that vehicle i is making a gap, and $u_{obstacle}$ is the desired or actual "deceleration" of the obstacle. In other words, if the obstacle starts to accelerate, vehicle i won't adapt its acceleration to that of obstacle. The feed-forward term is meant to address two issues:

- To enable the gap making vehicle to decelerate at least as hard as the merging car does, when OA is activated: without this term, the merging car can decelerate that hard that it gets out of the field of view of the gap making car and the entire maneuver execution falls apart.
- To increase the safe execution of the scenario: while the merging maneuver has started and if vehicle 1 brakes, CACC agent at vehicle 3 cannot detect this braking before it is too late for a safe action. However, the feedforward action added to OA controller can take this deceleration into account up to the point that CACC comes back into the loop, i.e. when vehicle 3 sees vehicle 1 as a forward object, and reacts.

3.2 Model of a controlled vehicle in platoon

Now consider a homogeneous platoon of m vehicles, each having dynamical model as in (2.1). Moreover, these are controlled by a one-vehicle look-ahead CACC [3]. Then, using (2.5) and assuming that CACC is the only active controller, the ith controlled vehicle in platoon, $i = 2, \ldots, m$, can be described by the following state-space model

$$\begin{aligned}\begin{pmatrix} \dot{e}_i \\ \dot{v}_i \\ \dot{a}_i \\ \dot{u}_i \end{pmatrix} &= \begin{pmatrix} 0 & -1 & -h & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\tau} & \frac{1}{\tau} \\ \frac{k_p}{h} & -\frac{k_d}{h} & -k_d & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} e_i \\ v_i \\ a_i \\ u_i \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{k_d}{h} & 0 & \frac{1}{h} \end{pmatrix} \begin{pmatrix} e_{i-1} \\ v_{i-1} \\ a_{i-1} \\ u_{i-1} \end{pmatrix} \\ &:= A_0 x_i + A_1 x_{i-1},\end{aligned} \tag{3.5}$$

where e_i is defined as in (2.4), $u_i = u_i^x$ is the longitudinal control input, $x_i := \begin{pmatrix} e_i & v_i & a_i & u_i \end{pmatrix}^T$, k_p and k_d are CACC parameters, and the matrices A_0 and A_1 are defined, correspondingly. Then, in [3], it is shown that for a bounded external input applied to the first vehicle and subject to some constraints on the controller gains, the above dynamics with the spacing policy (2.3) will be exponentially stable and string stable. Here, u_i represents the entire external input (acceleration) implemented to vehicle i, which is equal to the CACC control effort of vehicle i, $u_{CACC,i}$, when CACC is the only active control agent. However, if another control agent like OA is also active then u_i can be expressed for example as $u_i = u_{CACC,i} + u_{OA,i}$. In that case, the vehicle control system can be expressed as in Figure 11, where

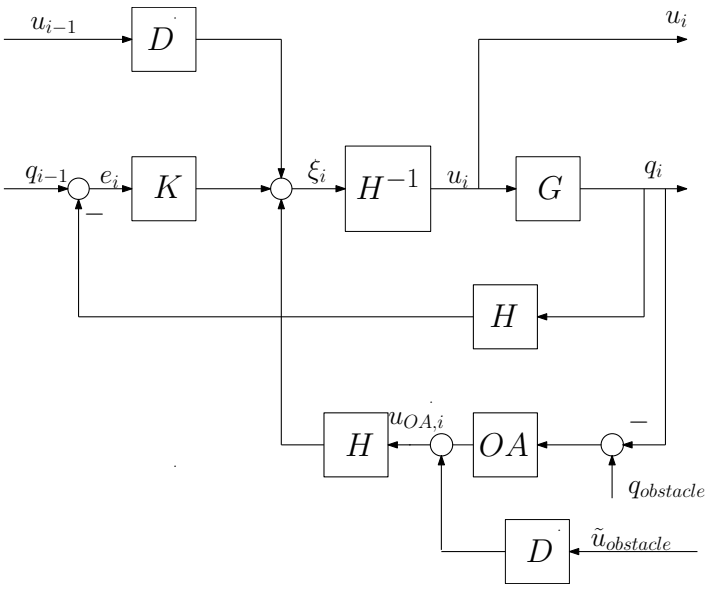

Figure 11: Block diagram of a controlled vehicle in a platoon with active CACC and OA controller agents.

$$\begin{aligned} G(s) &= \frac{q_i(s)}{u_i(s)} = \frac{1}{s^2(\tau s + 1)} \\ H(s) &= hs + 1 \\ K(s) &= k_p + k_d s \\ D(s) &= e^{-\Delta s}. \end{aligned} \tag{3.6}$$

Here, $q_i(s)$ and $u_i(s)$ are the Laplace transforms of the vehicle i position $q_i(t)$ and the desired acceleration $u_i(t)$, respectively, and Δ is the time delay induced by the wireless

communication network. Also, $q_{obstacle}$ is the obstacle's position, $\tilde{u}_{obstacle}$ is the deceleration of the obstacle, i.e equal to $u_{obstacle}$ if $u_{obstacle} < 0$ and equal to 0 if $u_{obstacle} >= 0$, and OA is the nonlinear function representing the OA controller. The parameters h, k_p and k_d are defined as before. Then, the corresponding system dynamics (3.5) can be rewritten to include OA controller as follows:

$$\dot{\bar{x}}_i = A_0\bar{x}_i + A_1\bar{x}_{i-1} + B_0u_{OA,i} + B_1u_{OA,i-1} \tag{3.7}$$

where $\bar{x}_i := \begin{pmatrix} e_i & v_i & a_i & u_{CACC,i} \end{pmatrix}^T$, $B_0 := \begin{pmatrix} 0 & 0 & \frac{\gamma_i}{\tau} & 0 \end{pmatrix}$, $B_1 := \begin{pmatrix} 0 & 0 & 0 & \frac{\gamma_{i-1}}{h} \end{pmatrix}$, $u_{OA,i}$ is the OA control effort of vehicle i, and A_0, A_1 are defined as before. Also, γ_i is a parameter indicating the activation/deactivation of OA controller of vehicle i. Therefore, it can take the values of 1 or 0 depending on being enabled or disabled, respectively. As it can be seen in (3.7), $u_{OA,i}$ and $u_{OA,i-1}$ influence the model through different input matrices. This is because $u_{OA,i}$ is directly influencing the acceleration dynamics of vehicle i, i.e. is part of the control feedback, whereas $u_{OA,i-1}$ is part of the feedforward action which only influences the controller dynamics $u_{CACC,i}$.

3.3 Steady-state behavior of obstacle avoiding controller

The steady state behavior of the OA controller is relevant for prediction of the maneuvering performance since it decides the gap which will be created by the OA controller. For that purpose, let us find the equilibrium point of a platoon vehicle dynamics given by (3.7) when $\gamma_i = 1$, i.e. OA controller of vehicle i is active. Then, from $\dot{\bar{x}}_i = A_0\bar{x}_i + A_1\bar{x}_{i-1} + B_0u_{OA,i} + B_1u_{OA,i-1}$, it can be derived that

$$\begin{aligned} -v_i - ha_i + v_{i-1} &= 0 \\ a_i &= 0 \\ -\frac{1}{\tau}(a_i - u_{CACC,i} - u_{OA,i}) &= 0 \\ \frac{k_p}{h}e_i - \frac{k_d}{h}v_i - k_da_i - \frac{1}{h}u_{CACC,i} + \frac{k_d}{h}v_{i-1} + \frac{1}{h}u_{CACC,i-1} + \frac{1}{h}u_{OA,i-1} &= 0 \end{aligned} \tag{3.8}$$

Knowing that at equilibrium $u_{i-1} = u_{CACC,i-1} + u_{OA,i-1} = 0$, we arrive at

$$\begin{aligned} e_i &= \frac{u_{CACC,i}}{k_p} = -\frac{u_{OA,i}}{k_p} \\ v_i &= v_{i-1} \\ a_i &= 0 \\ u_{CACC,i} &= -u_{OA,i}. \end{aligned} \tag{3.9}$$

Therefore, at steady state vehicles i and $i-1$ drive with the same constant speed. However, there is an error in their distance compared to the desired distance $\delta_{r,i}$ which is what we aimed at. Indeed, this error can be translated into the gap which is made for the maneuvering purpose, e.g. the gap made for the merging vehicle. To be more precise, let us assume that the traffic configuration is as shown in Figure 12, where vehicle i, the host vehicle, has an active CACC agent with respect to vehicle $i-1$ and an active OA with respect to merging vehicle denoted by 'obs'. Then, the following relations can be derived

$$\delta_i = \delta_c + L + \tilde{\delta}_i^o \Rightarrow e_i = \delta_c + L + \tilde{\delta}_i^o - r - hv_i, \tag{3.10}$$

where L is the merging vehicle length. Combining the first equation of (3.9) and equation (3.4) with the above equation we obtain that in steady state it holds that

$$u_{OA,i} = k_p(r + hv_i - L - \delta_c - \tilde{\delta}_i^o) = -\beta(\alpha\delta_i^o + 1)e^{-\alpha\delta_i^o} + u_{obstacle}. \tag{3.11}$$

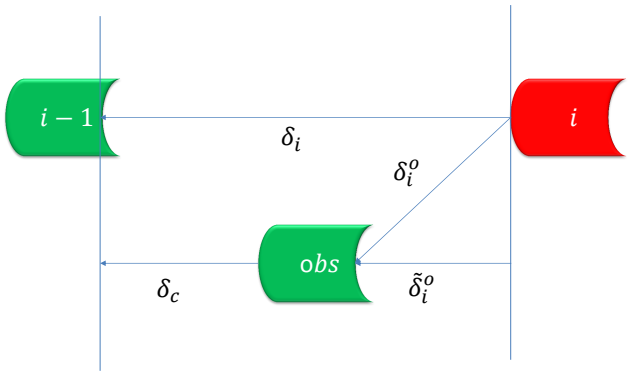

Figure 12: Distance definition in a nominal gap making configuration.

Also, δ_i^o and $\tilde{\delta}_i^o$ are related as

$$(\delta_i^o)^2 = (\tilde{\delta}_i^o)^2 + (L_w)^2, \tag{3.12}$$

where L_w is the lane width or in general the lateral offset of the obstacle with respect to the host vehicle i. Then, assuming that $u_{obstacle} = 0$, i.e. the obstacle is at steadystate, δ_c is constant and all vehicles drive at a nominal velocity of v_c, equations (3.11) and (3.12) give a solution for the equilibrium distance to the obstacle, i.e, $\delta_i^o = \delta_i^{o,ss}$. This solution can be found graphically as is shown in Figure 13 for a nominal velocity of 60 kph. Other parameters are chosen as $\beta = 6, \alpha = 0.3, k_p = 0.2, L_w = 3.5, h = 0.6, r = 2.5, L = 4.5, \delta_c = r + hv_c$. As it can be seen, with the given parameters, the gap, i.e. $d_i^{o,ss}$, is around $10m$. This graphical computation can be used to find appropriate controller parameters that result in the desired gap. A similar analysis can be done for the case where the cruise controller is replacing the CACC, i.e. when vehicle i detects no target. The gap made at equilibrium can be calculated by intersecting the $u_{OA,i}$ graph with the constant line of $-u_{CC} = k_{CC}(v_{c,o} - v_{c,i})$, where k_{cc} is the cruise controller gain and $v_{c,i}$ and $v_{c,o}$ are the cruise speeds of vehicle i and the obstacle, respectively. Clearly, the assumption is that $v_{c,i} >= v_{c,o}$.

By the above numerical analysis of the gap made by OA controller, we have all the elements required for performing scenario 1 of i-Game using the benchmark vehicles. Namely, based on platoon parameters, e.g. velocity, we first should decide how large the gap should be. Then, using the above approximation, an estimate of the OA parameters can be made. The parameters of CACC are chosen such that the stability and comfort criteria are guaranteed. Moreover, for in-vehicle implementation, in addition to inclusion of the relevant controller agents, e.g. OA and CACC, a wireless message set should be communicated between the vehicles. This set is presented in Deliverable D2.1. Also, the logical sequence of the scenario execution as well as other high-level decisions are made at the supervisory control layer, which is explained in more details in Deliverable 2.1.

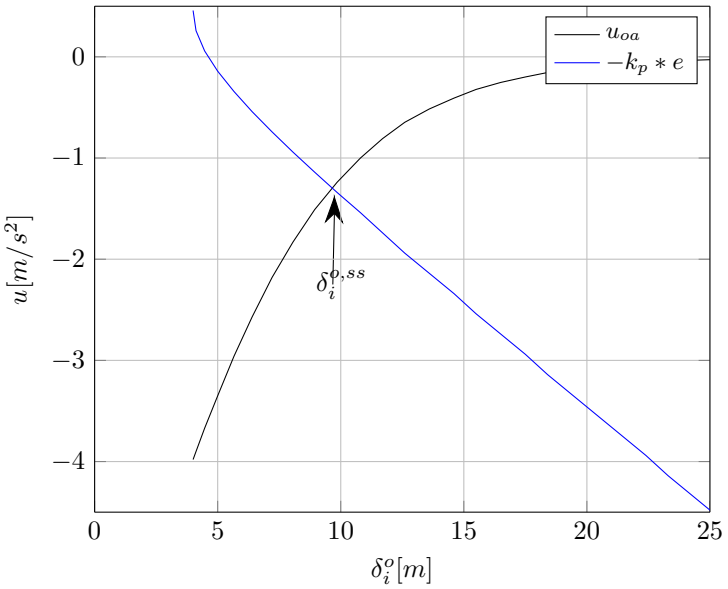

Figure 13: Calculation of the gap made by OA control agent in steady state.

4 Design approach for scenario 2

The automation of the intersection is achieved by forming virtual platoons of vehicles driving on different lanes. The idea behind the virtual platoon is to define a virtual inter-vehicle distance, between two vehicles driving on distinct lanes, using coordinate transformations. This virtual inter-vehicle distance is fed to an inter-vehicle distance controller such as Cooperative Adaptive Cruise Control (CACC) which generates the necessary space for the vehicles to cross the intersection is a safe manner. Hereafter we refer to the combination of virtual inter-vehicle distance and CACC as Virtual CACC (VCACC).

The presented solution, hereafter referred as Cooperative Intersection Control (CIC), is divided in two levels (as shown in Figure 14): a supervisory level that manages the formation of virtual platoons of vehicles approaching the intersection, and an execution level that takes care of the coordinate transformations needed to define the virtual gaps between vehicles on different lanes.

To define the virtual inter-vehicle distance subsystem (Section 4.2), in the execution level, it is necessary to define the possible trajectories (Section 4.1) for each lane.

The Target Vehicle Assignment (TVA) subsystem, in the supervision level, implements the interaction protocol. The TVA assigns each vehicle with a platoon index as soon as it enters the Competition Zone (CZ); this assignment is dependent on the lane ($k \in \{1, 2, 3\}$) and intention ($\eta \in \{s, l, r\}$ where s represents the intention to drive straight through the intersection, l represents the intention to take a left turn, and r represents the intention to take a right turn) of both the host vehicle m and the potential target vehicle $m - f$. For a more detailed description of the TVA subsystem, the reader is referred to deliverable D2.1.

The other subsystem in the supervision level is the control reconfiguration which uses the control mode step signal $cm \in \{1, 2, 3\}$ (note that $cm = 1$ represents CC, $cm = 2$ represents CACC, and $cm = 3$ represents VCACC) as an input to generate a smooth transition between the control modes. The details of this subsystem are presented in Section 4.4. An additional part of this subsystem is the VCACC cancellation logic that is presented in Section 4.3.

Once the virtual platoon is formed the leader will follow reference velocity, acceleration, orientation and yaw-rate signals, designed as described in Section 4.5, using the CC mode. The other members of the platoon will follow the leader at a reference virtual inter-vehicle distance, which will create the necessary distance between vehicles such as the whole virtual platoon can cross the intersection in a safe manner.

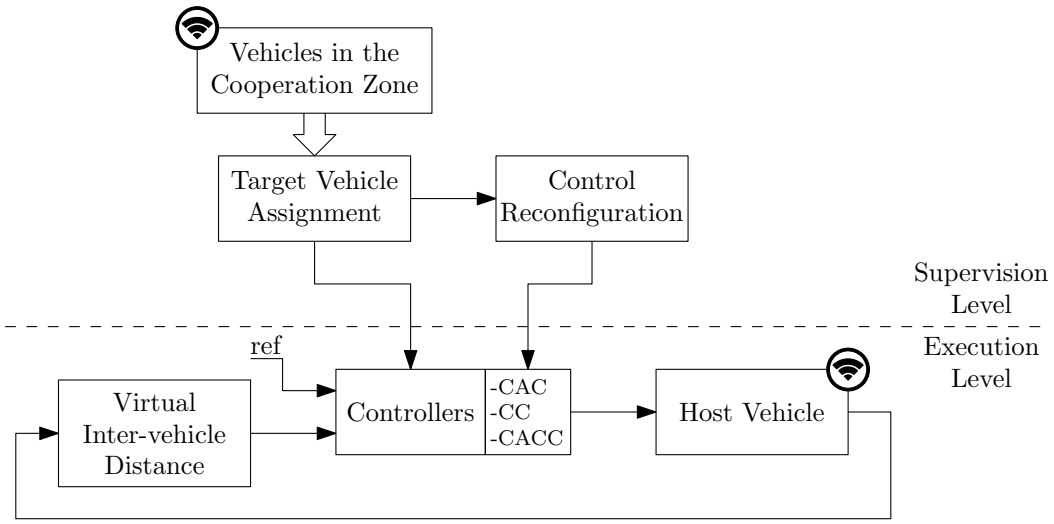

Figure 14: Control system architecture.

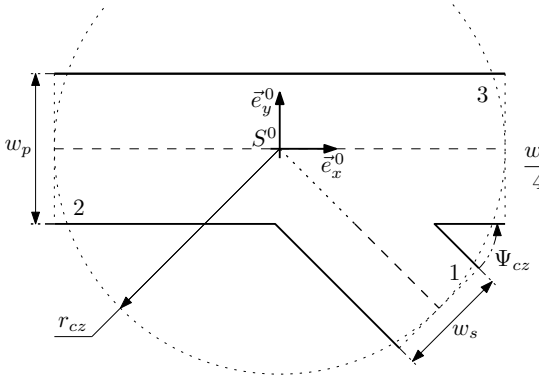

Figure 15: Generalized intersection geometry.

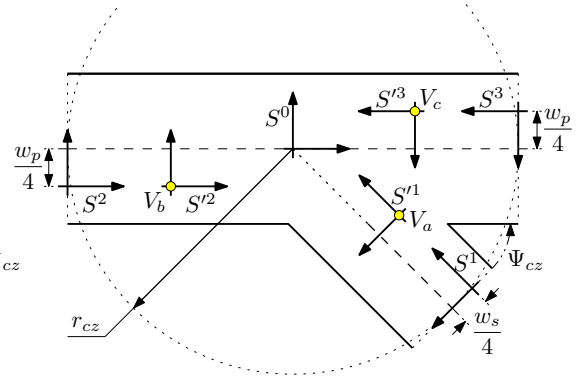

Figure 16: Vehicles' frames.

4.1 Possible Vehicle Trajectories

The scenario 2 of i-Game considers a cooperative T-intersection. Here we will consider a more generic intersection involving three roads. In particular this section presents a generalization of the intersection geometry and the related possible trajectories.

4.1.1 Intersection Geometry

Figure 15 depicts the geometry of the generalized intersection where w_p is the width of the main or principal road, w_s is the width of the secondary road, Ψ_{cz} is the angle between the secondary and primary road (which is $\pi/2$ in the i-Game scenario), r_{cz} is the radius of the circle that defines the boundary of the CZ, and $S^0 = \{O^0, \underline{\vec{e}}^0\}$ is the Intersection Reference Frame (IRF). The origin O^0 of the IRF is located at the intersection between the principal road middle line and the secondary road middle line. The orthonormal basis of the IRF is denoted as $\underline{\vec{e}}^0 = \begin{bmatrix}\vec{e}_x^0 & \vec{e}_y^0 & \vec{e}_z^0\end{bmatrix}^T$, where $\vec{e}_z^0 = \vec{e}_x^0 \times \vec{e}_y^0$ (pointing out of the page) and its orientation is such that the unitary vector $\vec{e}_x^0$ is parallel to the principal road, see Figure 15.

4.1.2 Vehicles' Frames

Each vehicle m is assigned with a set of frames $\mathcal{S}_m = \{S^{k_m}, S'^{k_m}\}$, where $k_m \in \{1, 2, 3\}$ is the lane on which the vehicle m entered the CZ, S^{k_m} is a stationary frame, and S'^{k_m} is a body-fixed frame. As an example let us consider the set of vehicles $\{V_m | m \in \{a, b, c\}\}$, depicted in Figure 16. The vehicle V_a entered the CZ on the lane $k_a = 1$ so it is assigned with the set of frames $\mathcal{S}_a = \{S^1, S'^1\}$. The assignment of the set of frames for vehicles V_b and V_c follow the same logic so, V_b is assigned with $\mathcal{S}_b = \{S^2, S'^2\}$, and V_c is assigned with $\mathcal{S}_c = \{S^3, S'^3\}$.

The stationary frame of lane k is defined as $S^k = \{O^k, \underline{\vec{e}}^k\}$, where O^k is the frame's origin, that is the point in which each vehicle enters the CZ, and $\underline{\vec{e}}^k$ is the frame's basis. The position of the frame's origin with respect to the origin of S^0 is given by

$$\vec{r}_{O^k} = \underline{r}_{O^k}^{0^T} \underline{\vec{e}}^0, \tag{4.1}$$

from which we can determine the coordinates as

$$\underline{r}_{O^k}^{0^T} = \vec{r}_{O^k} \cdot \underline{\vec{e}}^{0^T} = \begin{bmatrix} x_{O^k}^0 & y_{O^k}^0 & z_{O^k}^0 \end{bmatrix}. \tag{4.2}$$

So, we have that

$$\begin{aligned}
\underline{r}_{O^1}^{0^T} &= \begin{bmatrix} r_{cz}\cos\Psi_{cz} + \frac{1}{4}w_s \sin\Psi_{cz} & -r_{cz}\sin\Psi_{cz} + \frac{1}{4}w_s\cos\Psi_{cz} & 0 \end{bmatrix}, \\
\underline{r}_{O^2}^{0^T} &= \begin{bmatrix} -r_{cz} & -\frac{1}{4}w_p & 0 \end{bmatrix}, \\
\underline{r}_{O^3}^{0^T} &= \begin{bmatrix} r_{cz} & \frac{1}{4}w_p & 0 \end{bmatrix}.
\end{aligned} \tag{4.3}$$

Note that the origins O^k, $\forall k \in \{1,2,3\}$, are located at the intersection point between the lines perpendicular to the road (which delimit the CZ) and the middle line of each lane, rather than at the circumference of the CZ with radius r_{cz}.

The orthonormal frame $\underline{\vec{e}}^k$ is given by

$$\begin{bmatrix} \vec{e}_x^k \\ \vec{e}_y^k \\ \vec{e}_z^k \end{bmatrix} = \begin{bmatrix} \cos\gamma_k & \sin\gamma_k & 0 \\ -\sin\gamma_k & \cos\gamma_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{e}_x^0 \\ \vec{e}_y^0 \\ \vec{e}_z^0 \end{bmatrix} \tag{4.4}$$

or

$$\underline{\vec{e}}^k = R^T(\gamma_k)\underline{\vec{e}}^0, \tag{4.5}$$

where γ_k is the rotation angle between $\vec{e}_x^k$ and $\vec{e}_x^0$, and $R(\cdot)$ is the rotation matrix associated with a rotation about $\vec{e}_z^0$. The orientation of $\underline{\vec{e}}^k$ is such that the unitary vector $\vec{e}_x^k$ represents the forward longitudinal movement of the vehicles and $\vec{e}_y^k$ represents the lateral movement to the left. So, $\gamma_1 = \pi - \Psi_{cz}$, $\gamma_2 = 0$, and $\gamma_3 = \pi$. Therefore,

$$\begin{aligned} \underline{\vec{e}}^1 &= \begin{bmatrix} -\vec{e}_x^0\cos\Psi_{cz} + \vec{e}_y^0\sin\Psi_{cz} & -\vec{e}_x^0\sin\Psi_{cz} - \vec{e}_y^0\cos\Psi_{cz} & \vec{e}_z^0 \end{bmatrix}^T \\ \underline{\vec{e}}^2 &= \begin{bmatrix} \vec{e}_x^0 & \vec{e}_y^0 & \vec{e}_z^0 \end{bmatrix}^T \\ \underline{\vec{e}}^3 &= \begin{bmatrix} -\vec{e}_x^0 & -\vec{e}_y^0 & \vec{e}_z^0 \end{bmatrix}^T. \end{aligned} \tag{4.6}$$

The body-fixed frame corresponding to lane k, which is attached to a vehicle when it enters the CZ, is defined as $S'^k = \{O'^k, \underline{\vec{b}}^k\}$, where O'^k is the frame's origin with initial condition $O'^k(t_k) = O^k$, where t_k is the time at which the vehicle V_m enters the CZ, and $\underline{\vec{b}}^k$ is the frame's basis. The position of the origin of the frame S'^k with respect to the origin of the frame S^k is given by

$$\vec{r}_{O'^k/O^k} = \underline{r}^{k^T}_{O'^k/O^k}\underline{\vec{e}}^k, \tag{4.7}$$

from which we can determine the coordinates as

$$\underline{r}^{k^T}_{O'^k/O^k} = \vec{r}_{O'^k/O^k} \cdot \underline{\vec{e}}^{k^T} = \begin{bmatrix} x^k_{O'^k/O^k} & y^k_{O'^k/O^k} & z^k_{O'^k/O^k} \end{bmatrix}. \tag{4.8}$$

In order to simplify the further notation we redefine the components of $\underline{r}^{k^T}_{O'^k/O^k}$ as

$$\begin{bmatrix} x^k_{O'^k/O^k} & y^k_{O'^k/O^k} & z^k_{O'^k/O^k} \end{bmatrix} =: \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}. \tag{4.9}$$

The orthonormal frame $\underline{\vec{b}}^k$ is given by

$$\begin{bmatrix} \vec{b}_x^k \\ \vec{b}_y^k \\ \vec{b}_z^k \end{bmatrix} = \begin{bmatrix} \cos\theta_k & \sin\theta_k & 0 \\ -\sin\theta_k & \cos\theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{e}_x^k \\ \vec{e}_y^k \\ \vec{e}_z^k \end{bmatrix} \tag{4.10}$$

or

$$\underline{\vec{b}}^k = R^T(\theta_k)\underline{\vec{e}}^k, \tag{4.11}$$

where θ_k is the rotation angle between $\vec{b}_x^k$ and $\vec{e}_x^k$, and $R(\cdot)$ is the rotation matrix associated with a rotation about $\vec{e}_z^k$.

4.1.3 Straight Trajectory

Let d_k be a path coordinate related to the vehicles' longitudinal movement, which is defined using the moving basis $\underline{\vec{b}}^k$ of S'^k, and represents each vehicle's traveled distance inside the CZ. So, we can define this coordinate using the coordinates of the position vector in (4.9).

For the case of a straight trajectory, the orientation of $\underline{\vec{b}}^k$ does not change; so, the path coordinate is defined as

$$d_k = x_k, \tag{4.12}$$

since we are interested in the longitudinal movement of the vehicle. Figure 17 shows the possible path coordinates in the principal road which is the only road that allows straight trajectories.

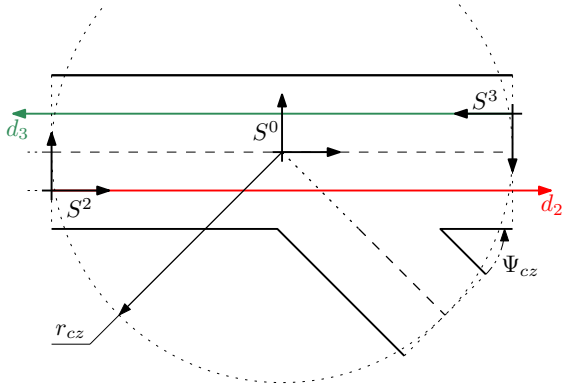

Figure 17: Straight trajectory.

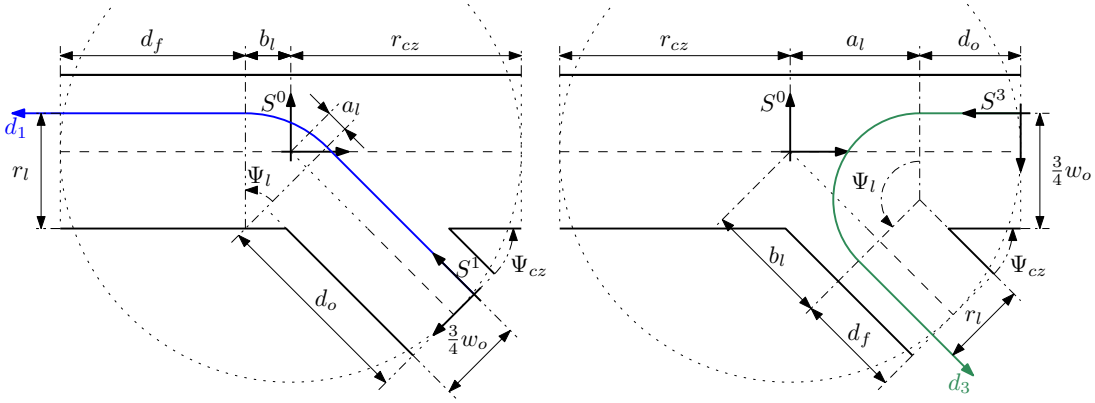

Figure 18: Left trajectory: for lane 1 (left), and for lane 3 (right).

4.1.4 Left-turn Trajectory

In this case, the orientation of $\vec{\underline{b}}^k$ changes when the vehicle takes the turn; this generates a curvilinear coordinate d_k. In order to define this coordinate we need to generalize the vehicle's trajectory first. Figure 18 shows two possible left-turn trajectories (starting from lane 1 and 3, respectively) that consist of two straight sections of length

$$\begin{aligned} d_o &= r_{cz} - a_l, \\ d_f &= r_{cz} - b_l, \end{aligned} \tag{4.13}$$

where

$$\begin{aligned} a_l &= \tfrac{1}{2} w_f \csc \Psi_l - (r_l - \tfrac{1}{4} w_o) \cot \Psi_l, \\ b_l &= -\tfrac{1}{2} w_f \cot \Psi_l + (r_l - \tfrac{1}{4} w_o) \csc \Psi_l, \end{aligned} \tag{4.14}$$

w_o is the width of the road on which the vehicle enters the CZ, and w_f is the width of the road on which the vehicle exits the CZ. See Appendix A for detailed derivations and definitions, and a left turn about an angle of Ψ_l radians with constant radius

$$r_l = \tfrac{3}{4} w_f. \tag{4.15}$$

Note that if the vehicle enters the CZ on lane 1 (Figure 18-left) then $w_o = w_s$, $w_f = w_p$, and $\Psi_l = \Psi_{cz}$. On the other hand, if the vehicle enters on lane 3 (Figure 18-right) then $w_o = w_p$, $w_f = w_s$, and $\Psi_l = \pi - \Psi_{cz}$.

Now we can define the curvilinear path coordinate d_k as a function of the coordinates in (4.9):

$$d_k = \begin{cases} x_k & x_k \le d_o, \;\; y_k = 0 \\ d_o + \psi_l(x_k, y_k) r_l & x_k > d_o, \;\; y_k \le r_l(1 - \cos \Psi_l) \\ d_o + \Psi_{cz} r_l + d_l(x_k, y_k) & \qquad\quad y_k > r_l(1 - \cos \Psi_l) \end{cases}, \tag{4.16}$$

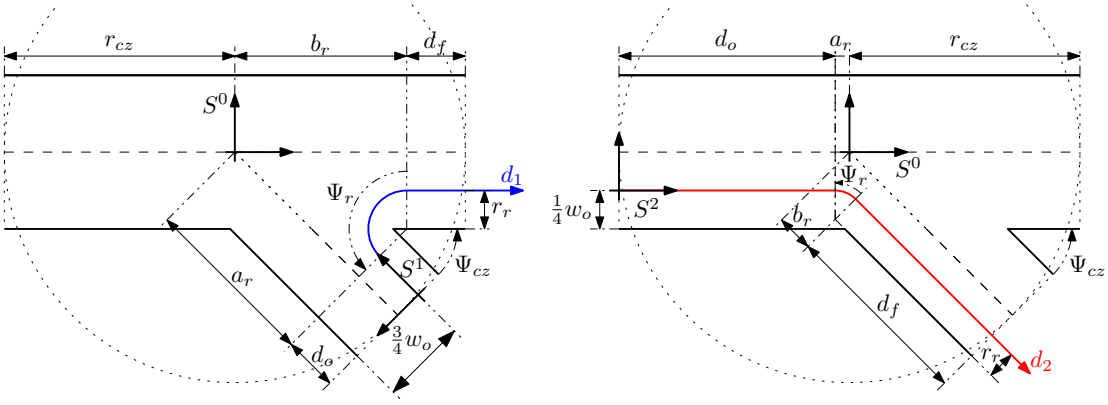

Figure 19: Right trajectory: for lane 1 (left), and for lane 2 (right).

where

$$\psi_l(x_k, y_k) = \arctan\left(\frac{x_k - d_o}{r_l - y_k}\right) \tag{4.17}$$

and

$$d_l(x_k, y_k) = \sqrt{(x_k - r_l \sin\Psi_l - d_o)^2 + (y_k - r_l(1 - \cos\Psi_l))^2}. \tag{4.18}$$

4.1.5 Right-turn Trajectory

In this case, the orientation of $\underline{\vec{b}}^k$ also changes when the vehicle takes the turn which generates a curvilinear coordinate d_k. In order to define this coordinate we need to generalize the vehicle's trajectory first. Figure 19 shows two possible right-turn of trajectories (starting from lane 1 and 2, respectively) that consist of two straight sections of length

$$\begin{aligned} d_o &= r_{cz} - a_r, \\ d_f &= r_{cz} - b_r, \end{aligned} \tag{4.19}$$

where

$$\begin{aligned} a_r &= \tfrac{1}{2}w_f \csc\Psi_r - (r_r + \tfrac{1}{4}w_o)\cot\Psi_r, \\ b_r &= -\tfrac{1}{2}w_f \cot\Psi_r + (r_r + \tfrac{1}{4}w_o)\csc\Psi_r, \end{aligned} \tag{4.20}$$

w_o is the width of the road on which the vehicle enters the CZ, and w_f is the width of the road on which the vehicle exits the CZ. See Appendix A for detailed derivations and definitions, and a right turn about an angle of Ψ_r radians with constant radius

$$r_r = \tfrac{1}{4}w_f. \tag{4.21}$$

Note that if the vehicle enters the CZ on lane 1 (Figure 19-left) then $w_o = w_s$, $w_f = w_p$, and $\Psi_r = \pi - \Psi_{cz}$. On the other hand, if the vehicle enters on lane 2 (Figure 18-right) then $w_o = w_p$, $w_f = w_s$, and $\Psi_r = \Psi_{cz}$.

Now we can define the curvilinear path coordinate d_k as a function of (4.9) as follows,

$$d_k = \begin{cases} x_k & x_k \le d_o, \;\; y_k = 0 \\ d_o + \psi_r(x_k, y_k) r_r & x_k > d_o, \;\; y_k > -r_r(1 - \cos\Psi_r) \\ d_o + \Psi_r r_r + d_r(x_k, y_k) & \qquad\quad y_k \le -r_r(1 - \cos\Psi_r) \end{cases}, \tag{4.22}$$

where

$$\psi_r(x_k, y_k) = \arctan\left(\frac{x_k - d_o}{r_r + y_k}\right) \tag{4.23}$$

and

$$d_r(x_k, y_k) = \sqrt{(x_k - r_r \sin\Psi_r - d_o)^2 + (y_k + r_r(1 - \cos\Psi_r))^2}. \tag{4.24}$$

4.2 Virtual Inter-vehicle Distance

To define the so-called virtual inter-vehicle distance between a pair of vehicles that are on different lanes, we first need to analyze the total traveled distance of each vehicle inside the CZ.

4.2.1 Total Traveled Distance

Let D_p be the total traveled distance, inside the Cooperation Zone, of a vehicle traveling along the principal road with intention to go straight; and D_s be the total traveled distance of a vehicle, either on the principal or secondary road, taking a left or right turn.

So, we have that

$$D_p = 2r_{cz} \tag{4.25}$$

and

$$D_s = d_o(a) + c + d_f(b) \tag{4.26}$$

where $d_o(a) = r_{cz} - a$, $d_f(b) = r_{cz} - b$, $\{a, b, c\} \in \{\{a_l, b_l, c_l\}, \{a_r, b_r, c_r\}\}$, $\{a_l, b_l\}$ are defined in (4.14), $\{a_r, b_r\}$ are defined in (4.20), $c_l = \Psi_l r_l$, and $c_r = \Psi_r r_r$.

By calculating

$$\begin{aligned} d_o(a) + d_f(b) &= 2r_{cz} - a - b \\ \Rightarrow 2r_{cz} &= d_o(a) + a + b + d_f(b) \\ \Rightarrow D_p &= d_o(a) + (a + b) + d_f(b) \end{aligned} \tag{4.27}$$

we can see that generically $D_s \neq D_p$ due to the fact that generically $c \neq a + b$. In other words, the distance traveled by a vehicle following a straight trajectory is not equal to the distance traveled by a vehicle taking a turn. So, in order to compare both traveled distances we introduce a mapping $T(\cdot)$ that defines a scaled distance $d'_s = T(d_s)$ such that

$$T(D_s) = D_p. \tag{4.28}$$

4.2.2 Coordinate scaling

We know that generically $D_s \neq D_p$ due to the fact that generically $c \neq a + b$, which is related to the fact that vehicles taking a turn travel a greater distance than the vehicles going straight through the intersection. Hence, it makes sense to scale d_s only for vehicles during the turn. This leads to the following scaling of the path coordinate d_s, while complying with the requirement in (4.28):

$$d'_s = T(d_s) = \begin{cases} d_s & d_s \leq d_o \\ d_o + \left(\dfrac{a+b}{c}\right)(d_s - d_o) & d_o < d_s \leq d_o + c \\ d_s + (a + b) - c & d_s > d_o + c \end{cases} . \tag{4.29}$$

This gives

$$T(D_s) = D_s + (a + b) - c, \tag{4.30}$$

since $D_s > d_o + c$. So, if we substitute the expression for D_s in (4.26) into (4.30) we obtain

$$T(D_s) = (d_o + c + d_f) + (a + b) - c = d_o + (a + b) + d_f = D_p, \tag{4.31}$$

where we used (4.27) in the last equality. In other words, (4.28) is indeed satisfied by the path coordinate transformation in (4.29).

The proposed scaling in the traveled distance will have an effect in the velocity $v_s = \dot{d}_s$ and acceleration $a_s = \ddot{d}_s$. The scaled velocity $v'_s = \dot{d}'_s$ and scaled acceleration $a'_s = \ddot{d}'_s$ are now given by

$$v'_s = \begin{cases} v_s & d_s \leq d_o \\ v_s \left(\dfrac{a+b}{c}\right) & d_o < d_s \leq d_o + c \\ v_s & d_s > d_o + c \end{cases} , \tag{4.32}$$

and

$$a'_s = \begin{cases} a_s & d_s \leq d_o \\ a_s \left(\dfrac{a+b}{c} \right) & d_o < d_s \leq d_o + c \\ a_s & d_s > d_o + c \end{cases} . \tag{4.33}$$

Now, we have defined a consistent set of path coordinates in the sense that the total traveled distance of a vehicle through the intersection after scaling now is always D_p. Next, we can define a virtual inter-vehicle distance as the difference between these scaled path coordinates. For instance, let a vehicle with the intention to take a turn, driving on either the primary or the secondary road with path coordinate d_s, be the target vehicle; also, let a vehicle with the intention to go straight driving on the primary road, with path coordinate d_p, be the host vehicle. Then, the virtual inter-vehicle distance $\delta_{s/p}$ between the target and host vehicle is defined as

$$\delta_{s/p} = d'_s - d_p, \tag{4.34}$$

where $d'_s = T(d_s)$. Similarly, if we invert the target and host vehicles roles we get

$$\delta_{p/s} = d_p - d'_s. \tag{4.35}$$

Now, let us consider two vehicles, both with the intention to take a turn while their trajectories cross. One of the vehicles is assigned the scaled path coordinate d'_{s1}, the other is assigned with the scaled path coordinate d'_{s2}. If the target vehicle is assigned with d'_{s1} and the host vehicle is assigned with d'_{s2} the inter-vehicle distance between them would be defined as

$$\delta_{s1/s2} = d'_{s1} - d'_{s2}, \tag{4.36}$$

and if the roles are inverted the inter-vehicle distance would be defined as

$$\delta_{s2/s1} = d'_{s2} - d'_{s1}. \tag{4.37}$$

Feeding this virtual inter-vehicle distance definition to a CACC controller will allow the vehicles to change their dynamics in order to execute their intention in a safe manner as will be shown in Chapter 5.

4.3 VCACC Cancellation

The purpose of CIC is to let a target vehicle cross the intersection in a safe manner by modifying the dynamics of the host vehicle following it. This is accomplished by the VCACC functionality that defines a virtual inter-vehicle distance, between vehicles driving on different lanes, and uses the CACC functionality to match the virtual inter-vehicle distance to a reference distance. Once the target vehicle has crossed the intersection (either by driving straight across the intersection or by completing a turn) VCACC is no longer needed by the host vehicle so it can change its control mode to CC if the vehicles end up on different lanes, or CACC if the vehicles end up on the same lane.

Depending on the situation, VCACC can be canceled in one of two ways: based on orientation, and based on position. We can condense all possible situations into three scenarios from which a general cancellation logic will be derived. In these scenarios, we will refer to the host vehicle's reference frames as S^h for the stationary frame, and S'^h for the body-fixed frame. Similarly, we will refer to the target's reference frames as S^t for the stationary frame, and S'^t for the body-fixed frame. Note that $t, h \in \{1, 2, 3\}$, depending of each vehicle's lane.

For the first scenario, let us consider the case in which the target vehicle is on lane 1 with the intention to take a left turn, and the host vehicle is on lane 3 with the intention to drive straight as shown in Figure 20. In Figure 20 (left), the VCACC is active since the target vehicle will take a left turn and the host vehicle will drive straight. After the target vehicle has taken the left turn, Figure 20 (right), VCACC is no longer needed because the two vehicles are now driving on the same road; thus, there is no need anymore for calculating the virtual inter-vehicle distance and we can use the normal CACC functionality. Therefore,

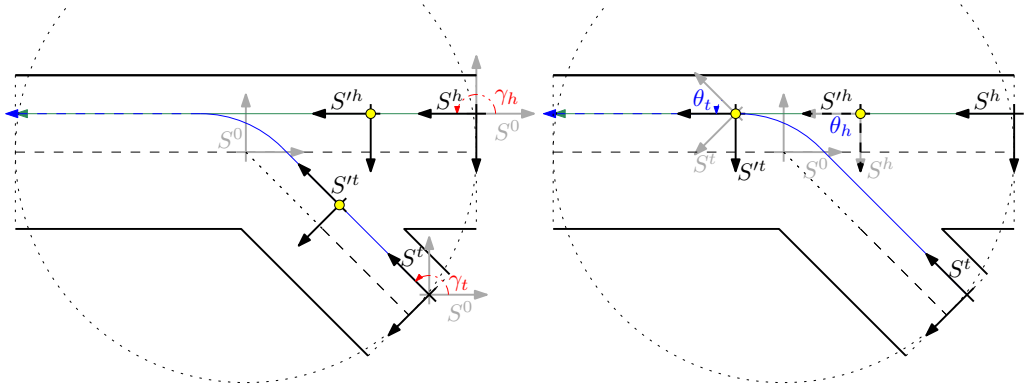

Figure 20: VCACC cancellation based on orientation: reference frames before the turn (left), and after the turn (right).

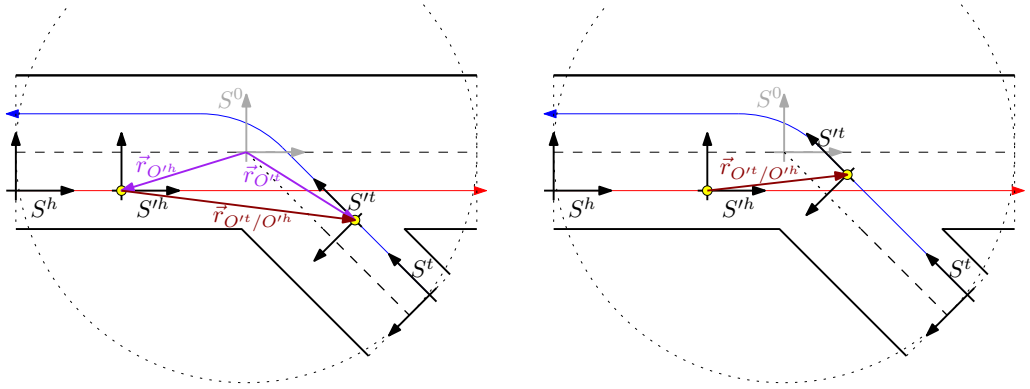

Figure 21: VCACC cancellation based on position: reference frames before the turn (left), and after the turn (right).

for this scenario we will cancel VCACC when both vehicles have the same orientation with respect to the IRF S^0, or in other words when

$$\|\theta_t + \gamma_t - (\theta_h + \gamma_h)\| \leq \epsilon \tag{4.38}$$

where $\epsilon \ll 1$ is used since in practice $\epsilon = 0$ would never lead to cancellation due to measurement noise on the heading sensor readings, γ_h and γ_t are the rotation angles of S^h and S^t with respect to S^0, and θ_h and θ_t are the rotation angles of S'^h and S'^t with respect to S^h and S^t, respectively. Note that this cancellation condition applies to any case in which the host and target vehicle end up on the same road.

In the second scenario, we consider the case in which the target vehicle is on lane 1 with the intention to take a left turn, and the host vehicle is on lane 2 with the intention to drive straight as shown in Figure 21. In Figure 21 (left), the VCACC is active since the target vehicle will take a left turn. After the target vehicle has passed in front the host vehicle, see Figure 21 (right), VCACC is no longer needed because the host vehicle has now free way to go straight using CC. Therefore, for this scenario we will cancel VCACC when the y-coordinate of the position vector

$$\vec{r}_{O'^t/O'^h} = \underline{r}^{h^T}_{O'^t/O'^h}\vec{\underline{b}}^h = \begin{bmatrix} x^h_{O'^t/O'^h} & y^h_{O'^t/O'^h} & z^h_{O'^t/O'^h} \end{bmatrix} \vec{\underline{b}}^h \tag{4.39}$$

of the target vehicle with respect to the host vehicle reference frame has changed sign. From Figure 21 (left), we know that

$$\vec{r}_{O'^t/O'^h} = \vec{r}_{O'^t} - \vec{r}_{O'^h} = (\underline{r}^{0^T}_{O'^t} - \underline{r}^{0^T}_{O'^h})\vec{\underline{e}}^0, \tag{4.40}$$

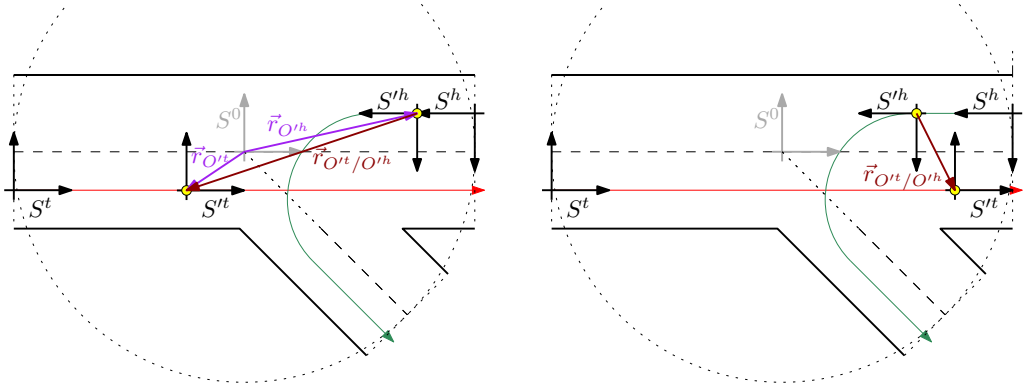

Figure 22: VCACC cancellation based on position: reference frames before the turn (left), and after the turn (right).

where $\underline{r}^0_{O'^t}$ and $\underline{r}^0_{O'^h}$ are the coordinates of the target and host vehicle with respect to the IRF S^0, respectively. We know that $\underline{\vec{e}}^h = R^T(\gamma_h)\underline{\vec{e}}^0$, and $\underline{\vec{b}}^h = R^T(\theta_h)\underline{\vec{e}}^h$, so we can write

$$\underline{\vec{e}}^0 = R(\gamma_h + \theta_h)\underline{\vec{b}}^h \tag{4.41}$$

since $R^T(\cdot) = R^{-1}(\cdot)$. Hence, we can rewrite the position vector in (4.40) as

$$\vec{r}_{O'^t/O'^h} = (\underline{r}^{0^T}_{O'^t} - \underline{r}^{0^T}_{O'^h})R(\gamma_h + \theta_h)\underline{\vec{b}}^h, \tag{4.42}$$

so the coordinates in (4.39) are given by

$$\begin{bmatrix} x^h_{O'^t/O'^h} & y^h_{O'^t/O'^h} & z^h_{O'^t/O'^h} \end{bmatrix} = (\underline{r}^{0^T}_{O'^t} - \underline{r}^{0^T}_{O'^h})R(\gamma_h + \theta_h). \tag{4.43}$$

Note that this cancellation condition applies to any case in which the vehicle's trajectories cross in just one point, except for the scenario depicted in Figure 22 and described below.

For the final scenario, we consider the case in which the target vehicle is on lane 2 with the intention to drive straight and the host vehicle is on lane 3 with the intention to take a left turn as shown in Figure 22. For this case we also use the coordinates of the position vector $\vec{r}_{O'^t/O'^h}$ defined in (4.43), but now we check when the x-coordinate changes sign. Note that if we invert the roles so that the target vehicle is on lane 3 and the host vehicle is on lane 2 we use the y-coordinate to cancel VCACC.

We can generalize these cancellation conditions to all the combinations of lanes and intentions as shown in Table 4.1, where $cn \in \{0, 1, 2, 3\}$ is the VCACC cancellation method ($cn = 0$ represents that no cancellation is needed since the vehicles entered on the same lane, $cn = 1$ represents the orientation cancellation, $cn = 2$ represents the position cancellation using the y-coordinate, and $cn = 3$ represents the position cancellation using the x-coordinate) which is a function $cn = p(k_h, \eta_h, k_t, \eta_t)$, where k_h and η_h are, respectively, the lane and intention of the host vehicle, and k_t and η_t are, respectively, the lane and intention of the target vehicle. Note that when the VCACC is canceled by $cn = 1$, the host vehicle will change its control mode to CACC ($cm = 2$); on the other hand, when the VCACC is canceled by either $cn = 2$ or $cn = 3$, the host vehicle will change its control mode to CC ($cm = 1$).

4.4 Control reconfiguration

Control reconfiguration with mixing is used to achieve the transition between different control modes. This reconfiguration method consist in calculating a convex combination of all

<table>
<tr><td></td><td></td><td></td><td colspan="8">Target</td></tr>
<tr><td></td><td>k</td><td></td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td></tr>
<tr><td></td><td></td><td>η</td><td>l</td><td>r</td><td>s</td><td>r</td><td>s</td><td>l</td></tr>
<tr><td rowspan="8">Host</td><td rowspan="2">1</td><td>l</td><td colspan="2" rowspan="2">$cn = 0$</td><td>$cn = 2$</td><td rowspan="2">$cn = 0$</td><td>$cn = 1$</td><td>$cn = 2$</td></tr>
<tr><td>r</td><td>$cn = 1$</td><td colspan="2">$cn = 0$</td></tr>
<tr><td rowspan="2">2</td><td>s</td><td>$cn = 2$</td><td>$cn = 1$</td><td colspan="2" rowspan="2">$cn = 0$</td><td rowspan="2">$cn = 0$</td><td>$cn = 2$</td></tr>
<tr><td>r</td><td colspan="2">$cn = 0$</td><td>$cn = 1$</td></tr>
<tr><td rowspan="2">3</td><td>s</td><td>$cn = 1$</td><td rowspan="2">$cn = 0$</td><td colspan="2">$cn = 0$</td><td colspan="2" rowspan="2">$cn = 0$</td></tr>
<tr><td>l</td><td>$cn = 2$</td><td>$cn = 3$</td><td>$cn = 1$</td></tr>
</table>

Table 4.1: VCACC cancellation method (cn) based on host and target vehicles' lane (k) and intention (η) ($cn = p(k_h, \eta_h, k_t, \eta_t)$).

the control efforts; so, the control input for the vehicle in the longitudinal direction is given by

$$u_m^x = \underline{\beta}_m^T \underline{u}_m^x \tag{4.44}$$

with $\underline{\beta}_m = \left[\beta_m^{ca} \quad \beta_m^{cc} \quad \beta_m^{cacc}\right]^T$, for $\|\underline{\beta}_m\|_1 = 1$ and $\beta_m^{ca}, \beta_m^{cc}, \beta_m^{cacc} \geq 0$; which are the properties of the convex combination.

Let the mixing signal β_{jm} be the element in the j-th row of $\underline{\beta}_m$. Each mixing signal is a function $\beta_j(\rho(cm(t)))$, where $\rho(cm(t))$ is a function that transforms the discrete changes in the control mode signal

$$cm(t) = \begin{cases} cm_1 & t \leq t_o \\ cm_2 & t > t_o \end{cases} \tag{4.45}$$

where t_o is the time in which the transition is commanded, cm_1 is the control mode before the transition is commanded, and cm_2 is the control mode after the transition is commanded, into a linear change defined by

$$\rho(cm) = \begin{cases} cm_1 & t \leq t_o \\ \dfrac{cm_2 - cm_1}{t_m}(t - t_o) + cm_1 & t_o \leq t < t_o + t_m \\ cm_2 & t \geq t_o + t_m \end{cases} \tag{4.46}$$

where t_m is the mixing time determined by the algorithm. The piece-wise linear change in $\rho(cm)$ is used as a reference to generate a smooth mixing signal.

For a more detailed explanation the reader is referred to [10]. For this work is enough to know that the control reconfiguration is achieved in a smooth manner.

4.5 Reference Signals

We will concentrate on designing the reference signals for the leader vehicle m which will take a turn to cross the intersection using CC. Additionally, we consider that the platoon members will cross the intersection with a straight trajectory. In other words, we will design the reference signals for the i-Game scenario 2.

The reference signals $v_m^{cc}(t)$, $a_m^{cc}(t)$, $\theta_m^{lc}(t)$, and $\omega_m^{lc}(t)$ are designed as a function of $d_m(t)$, which is the path coordinate inside the CZ. To design these reference signals, we first calculate the vehicle path coordinate d_m based on the vehicle's lane k_m and intention η_m. Then, for the longitudinal movement we identify the desired vehicle velocity and acceleration for the different parts of the path. In particular, for a left or right turn, we identify the desired velocity for the straight parts, the desired velocity for the turn, and the desired acceleration to change from one to the other or, for a straight path, we identify the constant desired velocity. For the lateral movement, we identify the desired orientation along the path. In particular, for a right or left turn the orientation only changes during the turn, and for a straight path the orientation will not change while the vehicle drives through the intersection. So, the desired yaw rate will be a piece-wise constant function that is different from zero only when the vehicle is turning.

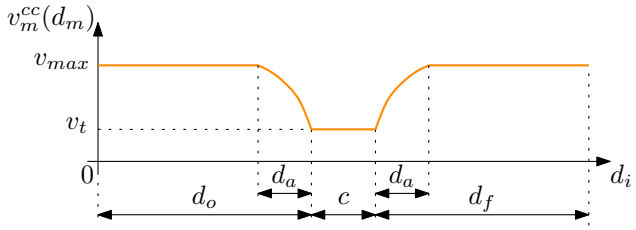

Figure 23: Reference velocity along the path coordinate.

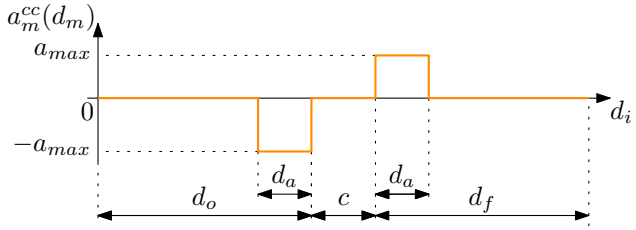

Figure 24: Reference acceleration along the path coordinate.

The outcome of this process is $v_m^{cc}(d_m(t))$, $a_m^{cc}(d_m(t))$, $\theta_m^{lc}(d_m(t))$, and $\omega_m^{lc}(d_m(t))$. Defining the reference signals as a function of the path coordinate simplifies the design process because we do not need to design $d_m(t)$ itself, we simply rely in the fact that $d_m(t) = \int_0^t v_m(s)ds$. With this approach, we want the vehicles to have certain velocity and acceleration at a certain position along the path instead of having certain velocity and acceleration at certain point in time. The next example will explain the design procedure of the desired trajectories in more detail.

Let us consider a vehicle that has the intention to take a turn in the intersection. Let v_{max} and v_t be the desired velocity for the straight parts of the trajectory and the desired velocity for the turn, respectively. Figure 23 depicts the reference velocity $v_m^{cc}(t)$ as a function of the path coordinate d_m, defined in Section 4.1. We can identify the straight parts of the trajectory, represented by the distances d_o and d_f, and the length of the arc $c \in \{c_l = \Psi_l r_l, c_r = \Psi_r r_r\}$.

The term d_a is the distance needed to change between v_{max} and v_t with a constant acceleration a_{max}, shown in Figure 24, which is given by

$$d_a = \left| \frac{v_{max}^2 - v_t^2}{2a_{max}} \right|. \tag{4.47}$$

Now, we can define the reference velocity as

$$v_m^{cc}(d_m) = \begin{cases} v_{max} & 0 \leq d_m < d_o - d_a \\ \sqrt{v_{max}^2 - 2a_{max}(d_m - d_o + d_a)} & d_o - d_a \leq d_m < d_o \\ v_t & d_o \leq d_m < d_o + c \\ \sqrt{v_t^2 + 2a_{max}(d_m - d_o - c)} & d_o + c \leq d_m < d_o + c + d_a \\ v_{max} & d_m \geq d_o + c + d_a \end{cases}, \tag{4.48}$$

and the reference acceleration as

$$a_m^{cc}(d_m) = \begin{cases} 0 & 0 \leq d_m < d_o - d_a \\ -a_{max} & d_o - d_a \leq d_m < d_o \\ 0 & d_o \leq d_m < d_o + c \\ a_{max} & d_o + c \leq d_m < d_o + c + d_a \\ 0 & d_m \geq d_o + c + d_a \end{cases}. \tag{4.49}$$

As the vehicle takes the turn, the orientation will change from 0 to the turning angle $\Psi \in \{\Psi_l, -\Psi_r\}$ as shown in Figure 25. Note that this figure represents a change in orientation

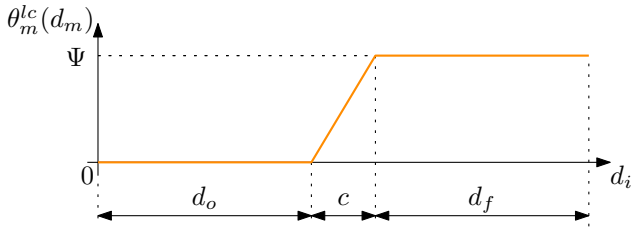

Figure 25: Reference orientation along the path coordinate.

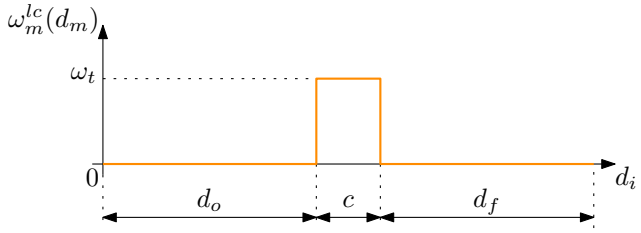

Figure 26: Reference yaw rate along the path coordinate.

in the positive direction which represents a left turn. In the case of a right turn, the orientation will decrease to achieve a negative orientation angle. While the vehicle takes the turn we also define the desired yaw rate, shown in Figure 26 where $\omega_t = r^{-1}v_t$ for $r \in \{r_l, r_r\}$.

Now, we can define the reference orientation angle as

$$\theta_m^{lc}(d_m) = \begin{cases} 0 & 0 \leq d_m < d_o \\ \dfrac{\Psi}{c}(d_m - d_o) & d_o \leq d_m < d_o + c \\ \Psi & d_m \geq d_o + c \end{cases}, \tag{4.50}$$

and the reference yaw rate as

$$\omega_m^{lc}(d_m) = \begin{cases} 0 & 0 \leq d_m < d_o \\ \dfrac{v_t}{r} & d_o \leq d_m < d_o + c \\ 0 & d_m \geq d_o + c \end{cases}. \tag{4.51}$$

5 Simulation Results

In the following subsections, the simulation results corresponding to the first and second scenario are presented.

5.1 Scenario 1

Here, a merging scenario is simulated where, a car is merging within a two-vehicle platoon. The merging car on the left lane accelerates to come somewhere in between of cars 1 and 2. Then, it starts to synchronize its speed with that of the platoon. Upon velocity synchronization, both the merging vehicle and gap making vehicle (vehicle 2) start to make needed space for merging. When the gap is big enough, the actual merging is performed. The simulation results can be seen in Figures 27-29. The following parameters have been used in the simulation setup. The cruise speed of the platoon is $16.7m/s(60\ kph)$ whereas the merging car has a higher cruise speed of $18.1m/s(65\ kph)$. Controller parameters are chosen as $\beta = 6, \alpha = 0.3, k_p = 0.2, L_w = 3.5, h = 0.6, r_i = 2.5, L = 4.5$, where L is the length of vehicles and L_w is the lane width. In Figure 27, the velocity profiles of the vehicles involved in the scenario execution are shown. Also, the three stages of speed synchronization, gap making, and merging are highlighted. Similarly, in Figure 28, the acceleration profile of all vehicles are presented. As can be seen, merging vehicle once brakes (around $t = 15s$) to adpat its speed to that of the platoon (from $18.1m/s$ to $16.7m/s$). When the speed synchronization is done, both the merging and the gap-making cars start to brake again (at $t = 20s$) to make an enough large gap. Finally, the merging is done at $t = 70s$ which results in a slight change both in velocity and acceleration profiles. The gaps made are better visualized in Figure 29, where the distance of the gap-making (GM) vehicle to the first (F) and the merging (M) vehciles are shown. Also, the distance of the merging vehicle to the front vehicle can be seen. As it can be seen, the distance from the merging to the gap making vehicle is initially negative. Eventualy, this distance increases to almost $12.5m$ which is the desired distance based on the vehicle following controller design. Also, the gap-making vehicle has a distance of $d_r = 12.5m$ to its front car which is increased to $30m$ at the end of the scenario execution. This disance is almost equal to $2 * d_r + L$ which is the gap needed for another car to fit in. In this figure, two moments of gap making and merging are highlighted.

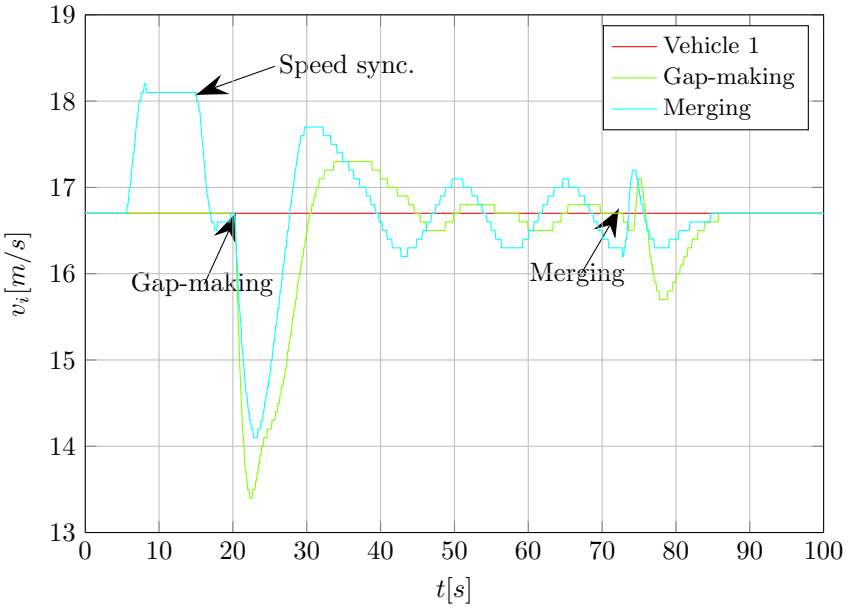

Figure 27: Velocity profile of the vehicles in a gap making/merging maneuver

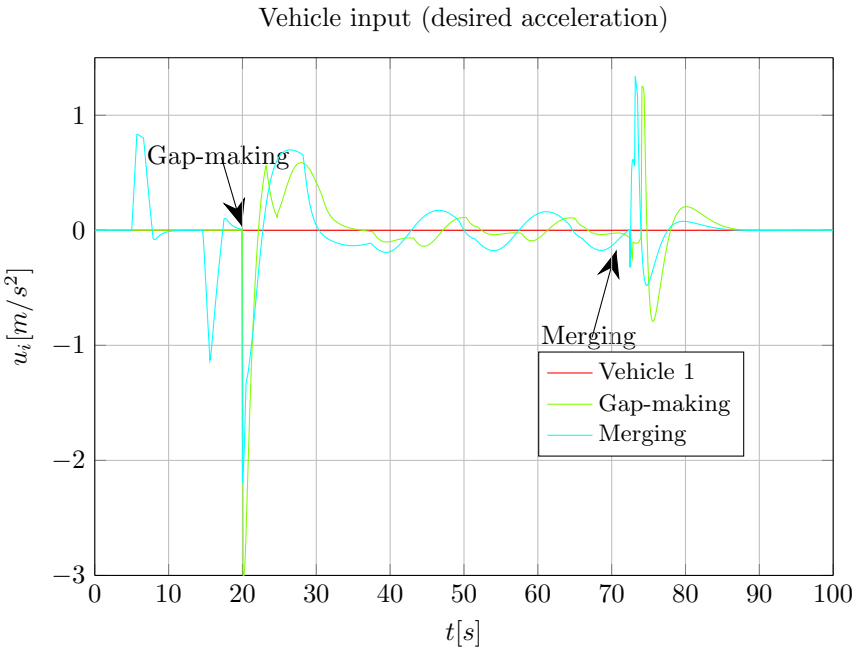

Figure 28: Acceleration profile of the vehicles in a gap making/merging maneuver

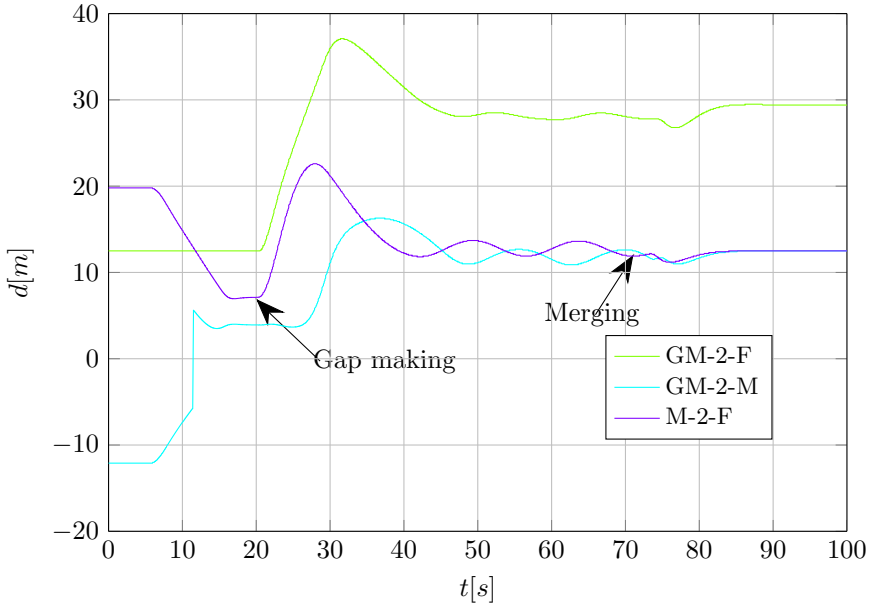

Figure 29: Distance of the gap-making (GM) vehicle to the first (F) and the merging (M) vehciles as well as the merging to the front vehicle

Symbol	Value	Name
τ	$0.1s$	Engine time constant.
a_{max}	$2m/s^2$	Maximum acceleration.
v_{max}	$30km/h = 8.33m/s$	Maximum velocity.
v_t	$20km/h = 5.56m/s$	Maximum turning velocity.
k_{lc}	$1s^{-1}$	Lateral controller constant.
k_{cc}	$1s^{-1}$	CC controller constant.
k_p	$0.2s^{-2}$	CACC controller proportional constant.
k_d	$0.7s^{-1}$	CACC controller derivative constant.
h	$0.5s$	headway time.
r^{cacc}	$10m$	Stand-still inter-vehicle distance.
r_{cz}	$100m$	Competition zone radius.
w_p	$9.2m$	Width of the primary road.
w_s	$5.4m$	Width of the secondary road.
Ψ_{cz}	$\pi/2$	Angle between the primary and secondary road.

Table 5.1: Simulation parameters.

5.2 Scenario 2

The parameter values used for this simulation are listed in Table 5.1. We will use the subscripts 1, 2, and 3 to identify the vehicles' signals. Note that this number coincides with the intersection vehicle counter m due to the fact that the vehicles enter at the same time to the CZ and the Target Vehicle Assignment (TVA), see deliverable D2.1 for details, uses the lane priority (in this case we assume the priorities to be the same as the lane number k) to update the counter.

5.2.1 Lane and Intention

First we identify the lane and intention of each vehicle:

- V1 enters the CZ on lane 1 with the intention to take a left turn; so, $\{k_1, \eta_1\} = \{1, \mathrm{l}\}$.
- V2 enters the CZ on lane 2 with the intention to drive straight; so, $\{k_2, \eta_2\} = \{2, \mathrm{s}\}$.
- V3 enters the CZ on lane 3 with the intention to drive straight; so, $\{k_3, \eta_3\} = \{3, \mathrm{s}\}$.

5.2.2 Intersection Frames

The vehicles' trajectories are depicted in Figure 30 (note that this figure is not to scale in terms of the simulation parameters). The lanes' stationary frames, with respect to S^0, are given by:

- $S^1 = \{O^1, \underline{\vec{e}}^1\}$, where the coordinates of the origin O^1 are

$$\begin{aligned} \underline{r}_{O^1}^{0^T} &= \begin{bmatrix} \frac{1}{4}w_s & -r_{cz} \end{bmatrix} \\ &= \begin{bmatrix} 1.35 & -100 \end{bmatrix} \end{aligned}, \tag{5.1}$$

  and the orthonormal frame $\underline{\vec{e}}^1$ is given by

$$\underline{\vec{e}}^1 = \begin{bmatrix} \vec{e}_y^0 & -\vec{e}_x^0 \end{bmatrix}^T. \tag{5.2}$$

- $S^2 = \{O^2, \underline{\vec{e}}^2\}$, where the coordinates of the origin O^2 are

$$\begin{aligned} \underline{r}_{O^2}^{0^T} &= \begin{bmatrix} -r_{cz} & -\frac{1}{4}w_p \end{bmatrix} \\ &= \begin{bmatrix} -100 & -2.3 \end{bmatrix} \end{aligned}, \tag{5.3}$$

  and the orthonormal frame $\underline{\vec{e}}^2$ is given by

$$\underline{\vec{e}}^2 = \begin{bmatrix} \vec{e}_x^0 & \vec{e}_y^0 \end{bmatrix}^T. \tag{5.4}$$

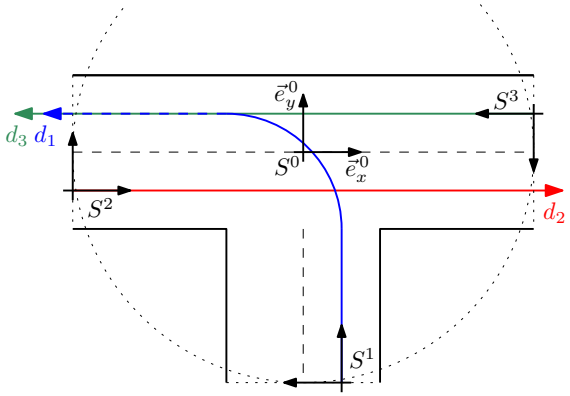

Figure 30: i-GAME scenario: Vehicles' trajectories.

- $S^3 = \{O^3, \underline{\vec{e}}^3\}$, where the coordinates of the origin O^3 are

$$\begin{aligned} \underline{r}_{O^3}^{0^T} &= \begin{bmatrix} r_{cz} & \frac{1}{4}w_p \end{bmatrix} \\ &= \begin{bmatrix} 100 & 2.3 \end{bmatrix} \end{aligned}, \tag{5.5}$$

  and the orthonormal frame $\underline{\vec{e}}^3$ is given by

$$\underline{\vec{e}}^3 = \begin{bmatrix} -\vec{e}_x^0 & -\vec{e}_y^0 \end{bmatrix}^T. \tag{5.6}$$

5.2.3 Target Vehicle Assignment

In this scenario, the vehicles enter the CZ at the same time; so, the lane priority will be used by the TVA subsystem presented in deliverable D2.1. Let us consider that the lanes' priorities coincide with the lane number making lane 1 the highest priority lane. This makes V1 the fist vehicle inside the CZ with counter number $m = 1$; thus, it is also the leader of the virtual platoon with index $i_1 = 1$ and control mode $cm_1 = 1(CC)$. The next vehicle to be considered is V2 which has counter number $m = 2$, since V1's and V2's trajectories are crossing the TVA assigns the platoon index $i_2 = 2$ and control mode $cm_2 = 3(VCACC)$. Finally, V3 has the counter number $m = 3$. The TVA checks the previous vehicle to enter the CZ, in this case V2, but since their trajectories are non-crossing, it checks the vehicle that entered before V2; in this case V1. V1's and V3's trajectories are crossing and, as a consequence, the TVA assigns V3 with the platoon index $i_3 = 2$ and control mode $cm_3 = 3(VCACC)$.

5.2.4 Reference Signals

Figure 31 shows V1's reference velocity v_1^{cc}, Figure 32 shows V1's reference acceleration a_1^{cc}, Figure 33 shows V1's reference orientation θ_1^{lc}, and Figure 34 shows V1's reference yaw rate ω_1^{lc}. Note that all reference signals are plotted as a function of the path coordinate d_1 that is calculated according to Section 4.1.

5.2.5 Vehicles Crossing the Intersection

Figure 35 shows a series of frames of the vehicles' position at several time instants. It can be seen how V2 and V3 slow down as they approach the intersection's center so that V1 can take the left turn in front of them. The frame at t=14.6s (time at which V1 has finished taking the turn) clearly shows why V2 can switch to CC and why V3 has to switch to CACC following V1 on the same lane.

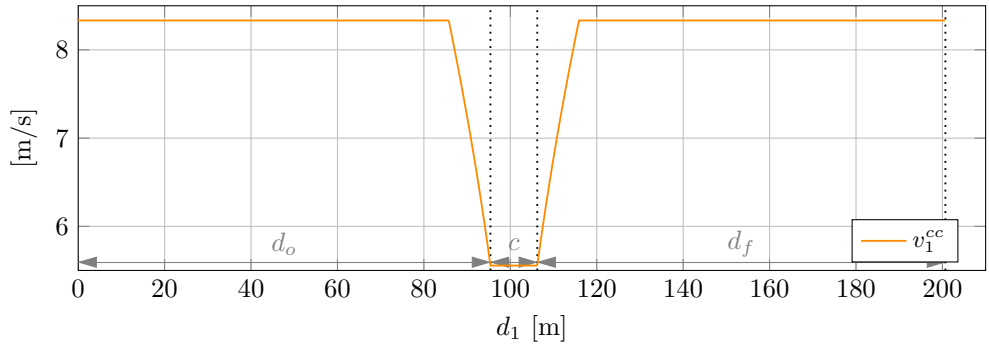

Figure 31: V1's reference velocity.

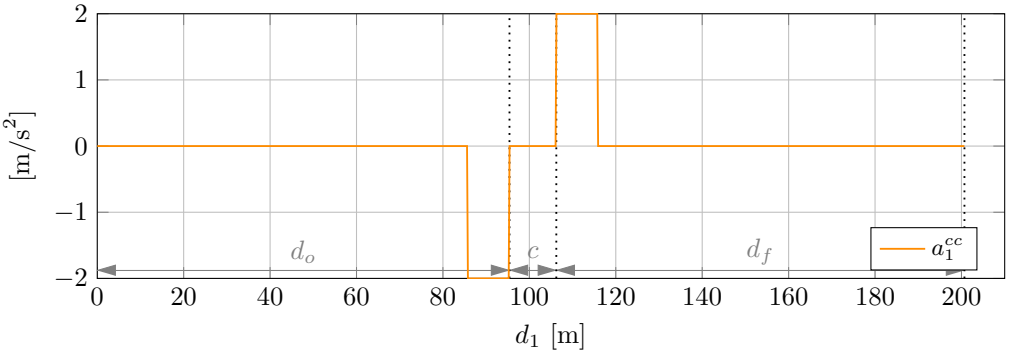

Figure 32: V1's reference acceleration.

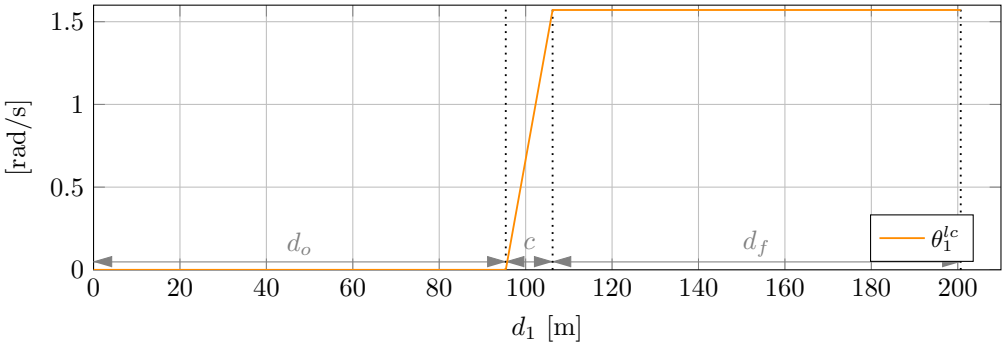

Figure 33: V1's reference orientation.

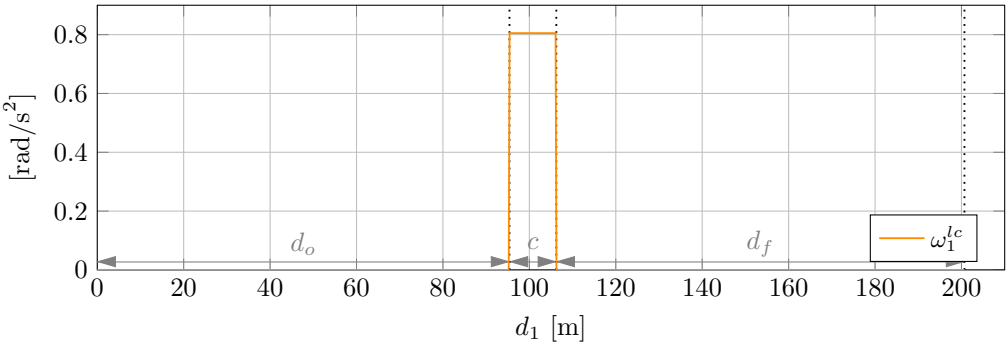

Figure 34: V1's reference yaw rate.

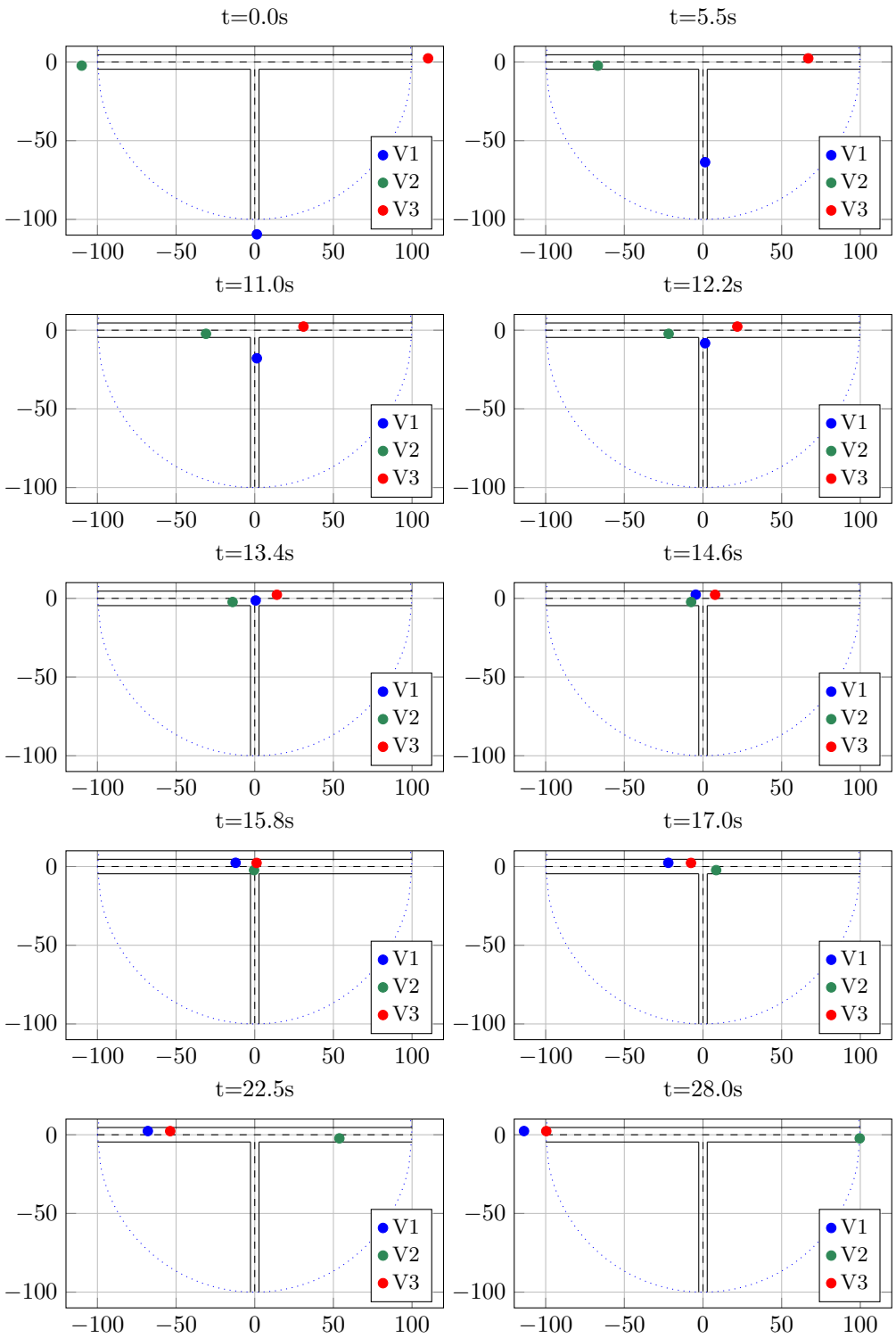

Figure 35: V2's and V3's response to V1's behavior.

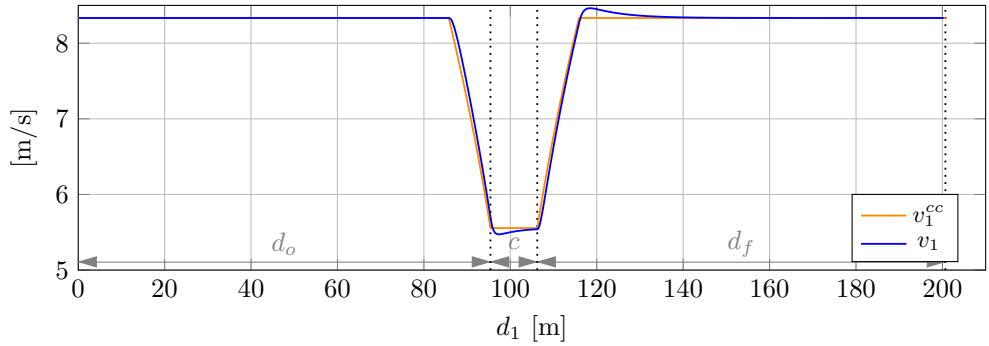

Figure 36: V1's response to its reference velocity.

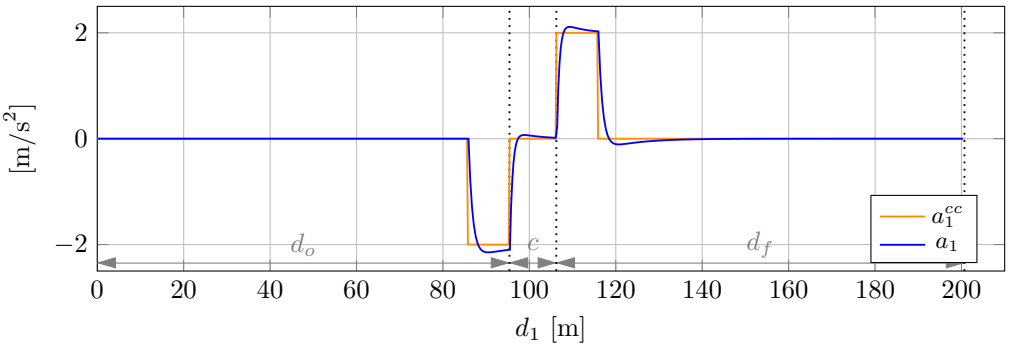

Figure 37: V1's response to its reference acceleration.

5.2.6 Response of V1 to the Reference Signals

The response of V1 to its reference signals is depicted in the following figures: Figure 36 shows velocity, Figure 37 shows acceleration, Figure 38 shows orientation, and Figure 39 shows yaw rate. From these figures, we can conclude that V1 is able to cross the intersection following reference signals designed as a function of its path coordinate. The reference signals are not followed exactly due to the performance of the longitudinal and lateral controllers.

5.2.7 Response of V2 and V3 to V1

When all the vehicles enter the CZ the virtual inter-vehicle distance between V1 and V2, and V1 and V3 is zero so, the VCACC decelerates V2 and V3 to increase the virtual gap. When the virtual inter-vehicle distance reaches the reference distance, the VCACC accelerates V2 and V3 to keep such reference. The responses of V2 and V3 are presented as a function of

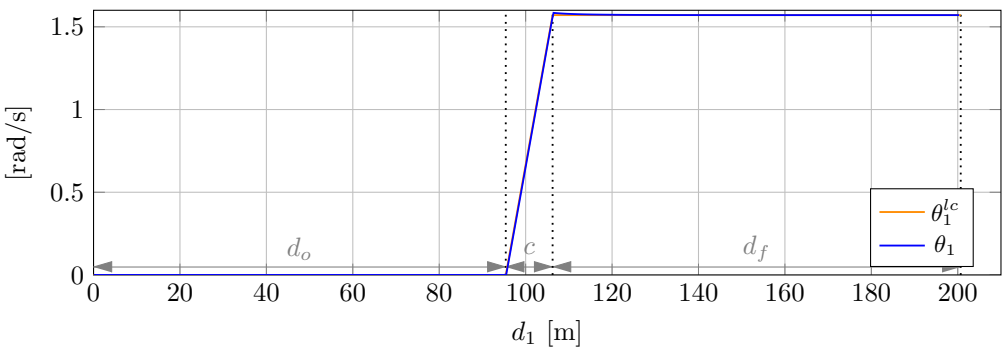

Figure 38: V1's response to its reference orientation.

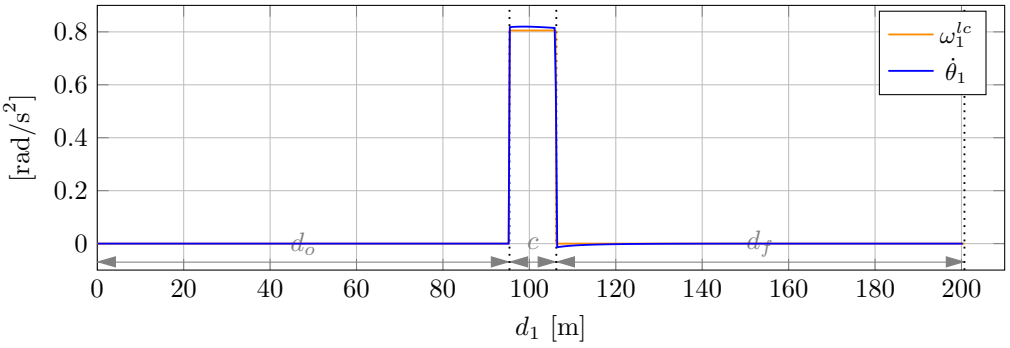

Figure 39: V1's response to its reference yaw rate.

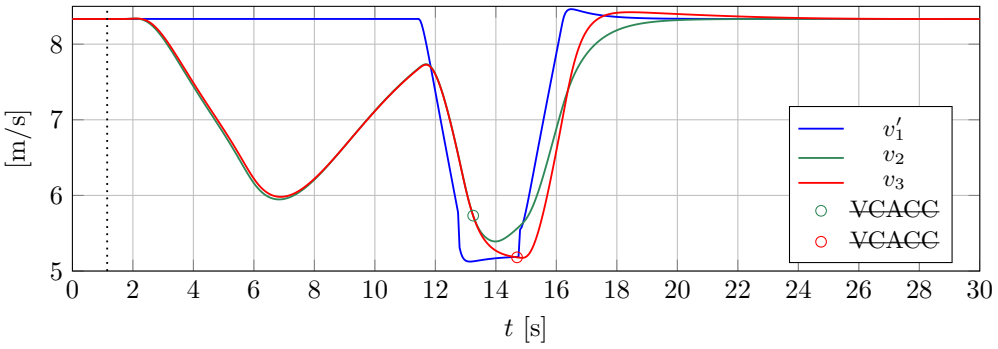

Figure 40: Response of V2 and V3 to V1's velocity.

time. Figure 40 shows their velocities; it can be seen how the velocity of V2 and V3 decrease after entering the CZ (represented by the vertical dotted black line) to let V1 take the turn. This is also visible when we examine their accelerations as shown in Figure 41. We can see how the vehicles start decelerating after entering the CZ. Note that the figures depict V1's scaled dynamics. In both figures ~~VCACC~~ represents the moment in which the VCACC is switched off.

As mentioned before V2 switches to CC after VCACC is canceled. In this case the cancellation condition is based on the y-coordinate of V1 with respect to V2's body-fixed frame, namely $y^2_{O'^1/O'^2}$ (defined in (4.43)) shown in Figure 42. It can be seen that the cancellation moments coincide with the moment in which this coordinate changes sign, which is the intended behavior.

On the other hand, V3 switches to CACC after VCACC is canceled. For this vehicle the cancellation condition is based on V1's and V3's orientation with respect to the Intersection

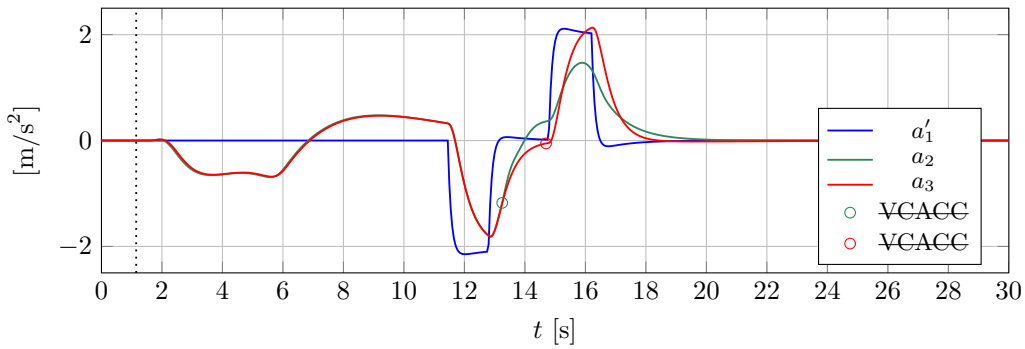

Figure 41: Response of V2 and V3 to V1's acceleration.

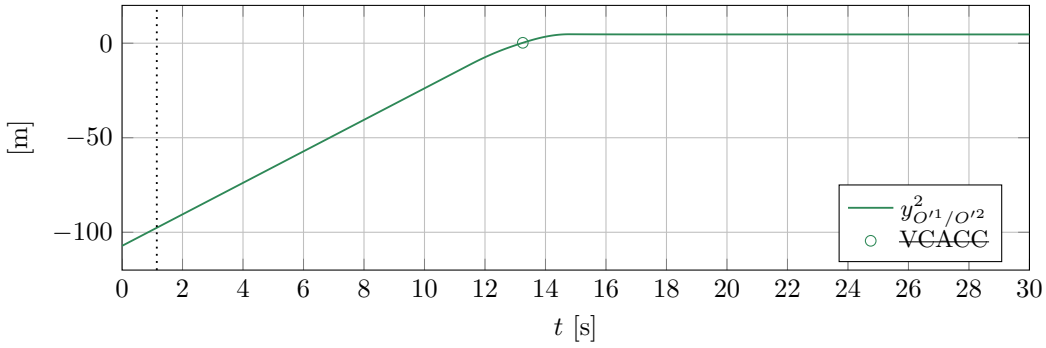

Figure 42: V2's condition to switch off VCACC.

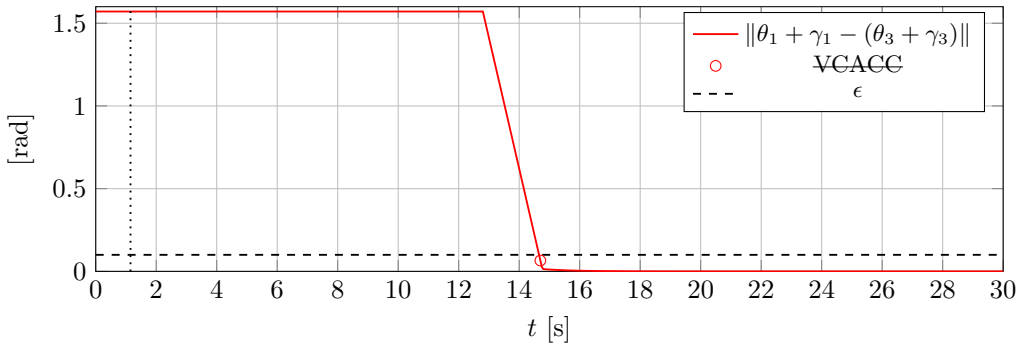

Figure 43: V3's condition to switch off CIC.

Reference Frame (IRF), namely $\|\theta_1 + \gamma_1 - (\theta_3 + \gamma_3)\| \leq \epsilon$ (in this case $\epsilon = 0.1$). Figure 43 shows the moment in which the aforementioned condition is met (the point representing the cancellation moment does not lie exactly at the dashed line due to the signal resolution).

We now check the virtual inter-vehicle distances. Figure 44 shows the behavior of V2 as it creates the virtual gap with respect to V1. It can be seen how V2 achieves the reference virtual inter-vehicle distance δ_2^{ic} before VCACC is canceled. After this moment V2 switches to CC, which also switches its control objective to follow the constant velocity v_{max} as shown in Figure 40.

Vehicle V3 also creates the virtual gap with respect to V1, as shown in Figure 45. The difference in this case is that V3 switches to CACC when VCACC is switched off; so, V3 has to still follow behind V1 but using the actual inter-vehicle distance rather than the virtual inter-vehicle distance. We can see how the virtual and actual inter-vehicle distances

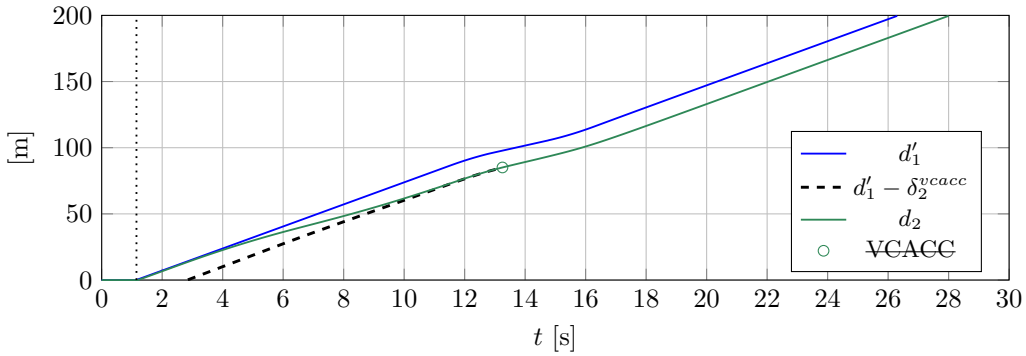

Figure 44: V2's virtual inter-vehicle distance.

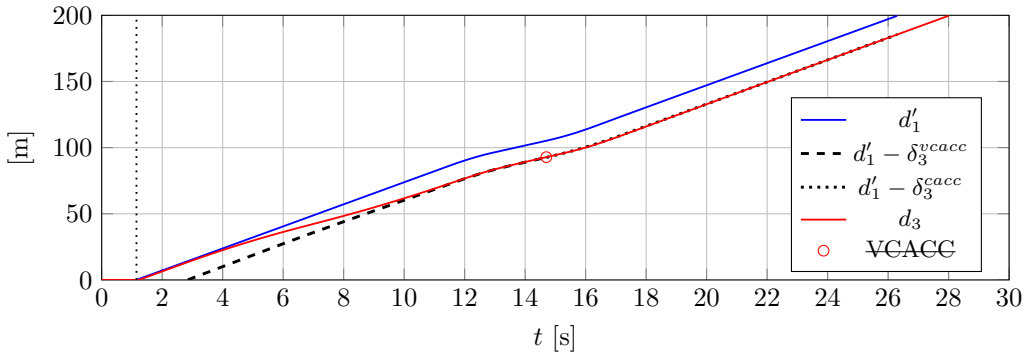

Figure 45: V3's virtual inter-vehicle distance.

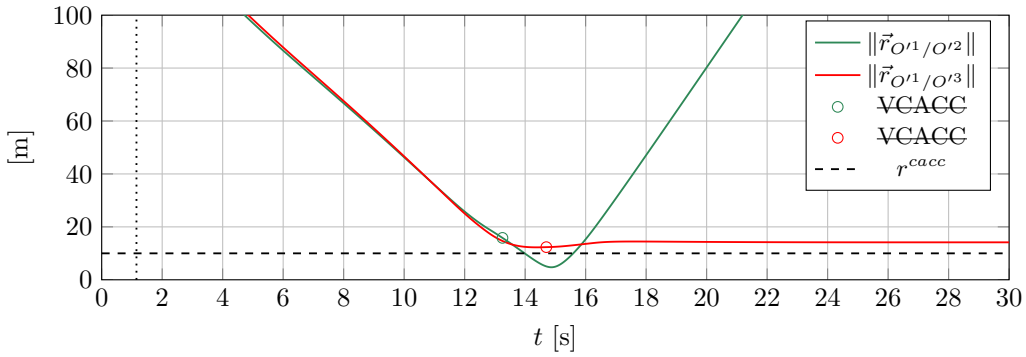

Figure 46: Euclidean distance between V1 and V2, and V1 and V3.

coincide at the moment VCACC is switched off, which is what was intended by scaling V1's dynamics.

Finally, Figure 46 shows the Euclidean distance between vehicles represented by the norm of the vectors $\vec{r}_{O'1/O'2}$ and $\vec{r}_{O'1/O'3}$. It can be noticed how the vehicles never go closer than the stand-still distance r^{cacc} (which is the constant term of the CACC reference distance $\delta^{cacc} = r^{cacc} + hv$) before the VCACC is switched off. After the VCACC is canceled, we can see that V1 and V2 come close together (represented by the minimum of $\|\vec{r}_{O'1/O'2}\|$), which is the moment in which the vehicles are side by side at a distance of $\frac{1}{2}w_p$.

This simulation illustrates the proposed method to solve the cooperative intersection problem applied to the i-GAME scenario. The results show that the method achieves the safe crossing of V1 through the intersection while V2 and V3 modify their dynamics to let V1 cross. Nevertheless, the algorithm was developed as to be dependent on the intersection geometry and the dynamical limitations of the vehicles which makes it capable of handling much more generic scenarios.

6 Minimum requirements for participating teams

The above proposed solutions are guidelines on how the i-Game scenarios can be implemented. These solutions will be implemented to the benchmark vehicles to be developed by TNO and TU/e. However, the participating teams do not have to comply to the entire solution proposed for each scenario. Here, we give an idea of what degrees of freedom are given to the participating teams in development of their systems.

As far as the implementation of scenario 1 is concerned, the participants are free in their choice of control system architecture. This includes the layers defined as well as majority of the content used at each layer. As some examples, the choice of target tracking algorithms, the number of distinguished objects, and the choice of real-time controllers, are arbitrary. However, some of the actions included at supervisory control layer, like the interaction protocol which includes the V2V message set and the logical sequence of the execution, should be consistent among the teams and be interpreted in a unified manner. As an example the pairup action should be done in the same manner by all the participants. However, the actual gap making controller is chosen arbitrary. There are also some parameters which need to be unified beforehand, like the minimum gap required, the minimum (or maximum) acceleration/deceleration tolerated, criteria used for issuing STOM message, etc.

For the implementation of scenario 2, the participants are free to design their own algorithm. The CIC is just a method to generate enough space for a vehicle, which is driving on the secondary road, to enter the primary road in a safe manner by means of a virtual platoon. The trajectory generation and following are open for the participants to design. The key concepts that has to be kept in mind are the vehicles' priorities based on the lane on which they entered the CZ, and the vehicle's lane and intention.

7 Conclusion

This document explains the details of the control system designed for the implementation of the i-Game scenarios. The underlying control system design strategy is based on an agent-based approach. In this approach, corresponding to each functionality required from the control system, a controller (a.k.a. agent) is designed. Therefore, in addition to the nominal control functions which are required for basic platooning function, such as cooperative adaptive cruise control, two major controllers were designed to be able to implement the i-Game scenarios. These are the obstacle avoiding and virtual cooperative adaptive cruise control. The former controller is responsible for making the necessary gaps required for performing a merging scenario and the latter is responsible for making the necessary virtual-gaps to achieve a safe intersection crossing.

The obstacle avoidance (OA) controller is designed based on a quasi-potential function. This controller provides a deceleration action when the merging car is close to the other cars involved in the maneuver. This repulsive force is used as a means of gap making. In case the cars involved in maneuvering are far enough from each other, then no action from OA is added. Hence, in that condition the only active agent would be the CACC which is responsible for regulating the inter-vehicle distances.

This report also presents the Cooperative Intersection Control (CIC) strategy which is used to automate a three lane intersection. Each vehicle that enters the intersection's Cooperation Zone is represented by the lane on which it entered and the intention it has (either to take a turn or to drive straight through the intersection). This information is used by the TVA to assign the vehicle's role; either as leader vehicle (that does not modify its dynamics) or as a follower vehicle (that modifies its dynamics).

The CIC is divided into a supervisory level and an execution level. On the execution level, this methodology uses an already existing inter-vehicle distance controller (CACC) to achieve the safe passage of vehicles through the intersection. It does so by defining a virtual inter-vehicle distance between vehicles that drive on different lanes of the intersection. To calculate the virtual inter-vehicle distance it is necessary to generalize the intersection's geometry, to define stationary and body-fixed vehicle frames, and generalize all the possible trajectories in order to define consistent path coordinates (Section 4.2). The combination

of the virtual inter-vehicle distance and the CACC controller is called VCACC through this document.

The supervisory level takes care of the target vehicle assignment (this subsystem and the control mode logic are described in Deliverable D2.1) that assigns a place for each vehicle in the virtual platoon (also referred as the vehicle priority), and of the VCACC cancellation that identifies when the VCACC is no longer needed and commands a control reconfiguration (Section 4.4).

The aforementioned subsystems and the dynamic equations presented in Section 3.2 are used to perform a simulation. From the simulation results (Chapter 5), we can conclude that the CIC complies with the expected functionality; which is that the lower priority vehicles modify their dynamics to let the higher priority vehicles cross the intersection first, in an automated and safe manner.

8 List of abbreviations and terminology

Table 8.1: List of abbreviations

Abbreviation	Full Name
CACC	Cooperative adaptive cruise control
VCACC	Virtual cooperative adaptive cruise control
CC	Cruise control
LC	Lateral control
CA	Collision avoidance
OA	Obstacle avoidance
MIO	Most important object
FWD MIO	Forward most important object
BWD MIO	Backward most important object
EV	Emergency vehicle
CZ	Competition Zone
RSU	Road side unit
TT	Target tracking
IRF	Intersection reference frame
CIC	Cooperative intersection control
TVA	Target vehicle assignment

List of Figures

A Calculation of Left- and Right-turn Trajectories

This appendix presents the calculations of the elements needed to define both left-turn and right-turn trajectories.

A.1 Left-turn

Figure 47 shows a trajectory that consist of two straight paths and a left turn with constant radius, where

$$\begin{aligned} d_o &= r_{cz} - a_l \\ d_f &= r_{cz} - b_l \\ r_l &= \tfrac{3}{4} w_f \\ \Psi_l &= \Psi_{cz} \\ w_o &= w_s \\ w_f &= w_p, \end{aligned} \tag{A.1}$$

note that r_l is the constant turning radius, Ψ_l is the turning angle, w_o is the width of the road on which the vehicle entered the Competition Zone, and w_f is the width of the road on which the vehicle exits the Competition Zone.

We use Figure 48 as a reference to calculate a_l and b_l. Note that this figure is extracted from Figure 47.

We first consider

$$\cos \Psi_l = \frac{h_4 - h_2}{a_4 + a_3}, \tag{A.2}$$

if we substitute $h_4 = r_l \cos \Psi_l$, $a_3 = h_5 \cos \Psi_l$, and $a_4 = \frac{1}{4} w_o$ in (A.2) we get

$$h_2 + h_5 \cos^2 \Psi_l = (r_l - \tfrac{1}{4} w_o) \cos \Psi_l. \tag{A.3}$$

From the intersection's geometry we know that

$$h_2 + h_5 = \tfrac{1}{2} w_f; \tag{A.4}$$

so, if we subtract (A.3) from (A.4) we get

$$\begin{aligned} h_5(1 - \cos^2 \Psi_l) &= \tfrac{1}{2} w_f - (r_l - \tfrac{1}{4} w_o) \cos \Psi_l \\ h_5 \sin^2 \Psi_l &= \tfrac{1}{2} w_f - (r_l - \tfrac{1}{4} w_o) \cos \Psi_l. \end{aligned} \tag{A.5}$$

Since $a_l = h_5 \sin \Psi_l$ we have that

$$a_l = \tfrac{1}{2} w_f \csc \Psi_l - (r_l - \tfrac{1}{4} w_o) \cot \Psi_l. \tag{A.6}$$

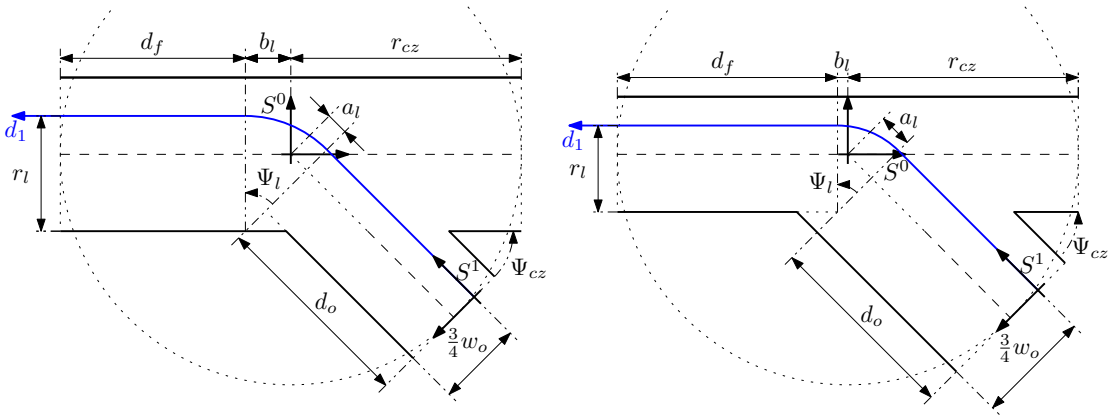

Figure 47: Left intention: $w_p > w_s$ (left), and $w_p < w_s$ (right).

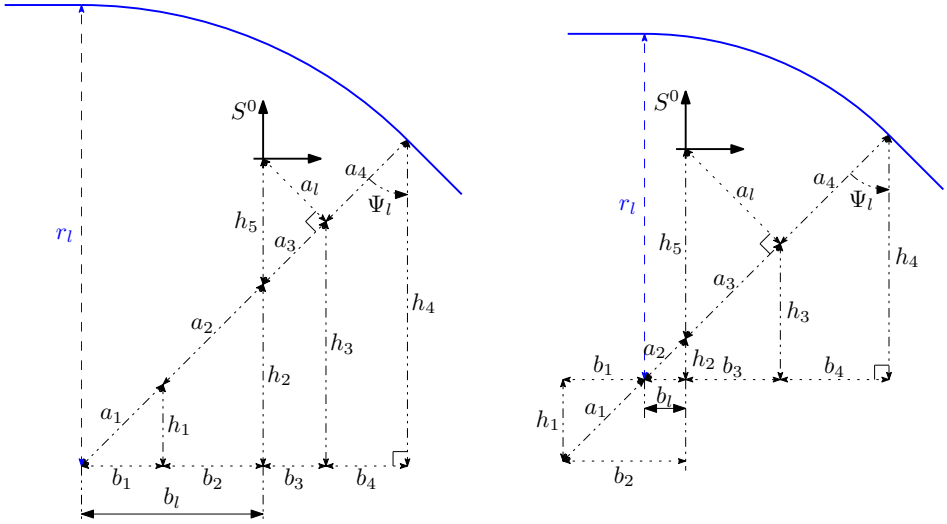

Figure 48: Geometry to calculate a_l and b_l: $w_p > w_s$ (left), and $w_p < w_s$ (right).

Now we consider

$$\tan \Psi_l = \frac{b_l}{h_2} \tag{A.7}$$

if we substitute $h_2 = \frac{1}{2}w_f - h_5$, and $h_5 = a_l \csc \Psi_l$ in (A.7) we get

$$\begin{aligned} b_l &= (\tfrac{1}{2}w_f - a_l \csc \Psi_l) \tan \Psi_l \\ &= \tfrac{1}{2}w_f \tan \Psi_l - a_l \sec \Psi_l, \end{aligned} \tag{A.8}$$

finally, if we substitute (A.6) in (A.8) we have

$$\begin{aligned} b_l &= \tfrac{1}{2}w_f \tan \Psi_l - (\tfrac{1}{2}w_f \csc \Psi_l - (r_l - \tfrac{1}{4}w_o) \cot \Psi_l) \sec \Psi_l \\ &= \tfrac{1}{2}w_f \sec \Psi_l (\sin \Psi_l - \csc \Psi_l) + (r_l - \tfrac{1}{4}w_o) \csc \Psi_l; \end{aligned} \tag{A.9}$$

so,

$$b_l = -\tfrac{1}{2}w_f \cot \Psi_l + (r_l - \tfrac{1}{4}w_o) \csc \Psi_l. \tag{A.10}$$

Note that the calculation of a_l and b_l holds for $w_p \geq w_s$ and $w_p < w_s$. Additionally, the derivation presented above also holds for the case in which the vehicle enters the competition zone on the principal road with $w_o = w_p$, $w_f = w_s$, and $\Psi_l = \pi - \Psi_{cz}$.

A.2 Right-turn

Figure 49 shows a trajectory that consist of two straight paths and a right turn with constant radius, where

$$\begin{aligned} d_o &= r_{cz} - a_r \\ d_f &= r_{cz} - b_r \\ r_r &= \tfrac{1}{4}w_f \\ \Psi_r &= \pi - \Psi_{cz} \\ w_o &= w_s \\ w_f &= w_p, \end{aligned} \tag{A.11}$$

note that r_r is the constant turning radius, Ψ_r is the turning angle, w_o is the width of the road on which the vehicle entered the Competition Zone, and w_f is the width of the road on which the vehicle exits the Competition Zone.

We use Figure 50 as a reference to calculate $a_r = a_1 + a_2$ and $b_r = b_1 + b_2$. Note that this figure is extracted from Figure 49. In this case we only present the case when $w_p > w_s$ since the geometry of the triangles does not change for $w_p < w_s$ as in Figure 48.

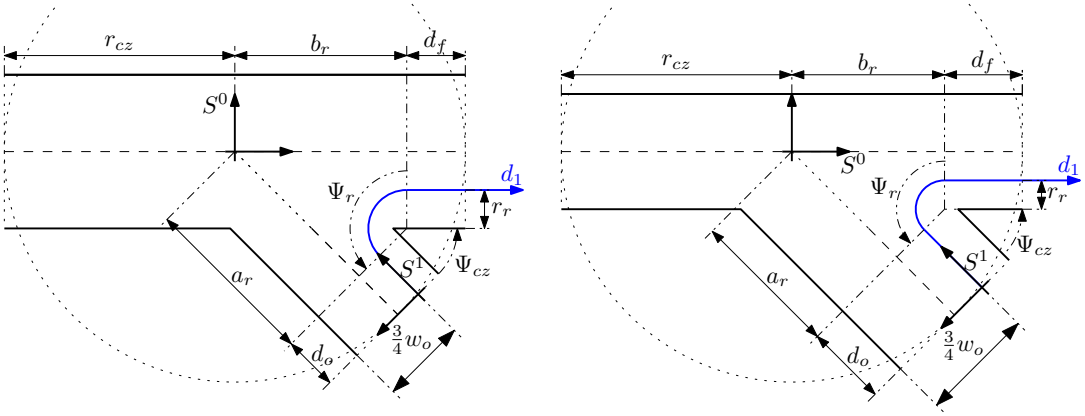

Figure 49: Right intention: $w_p > w_s$ (left), and $w_p < w_s$ (right).

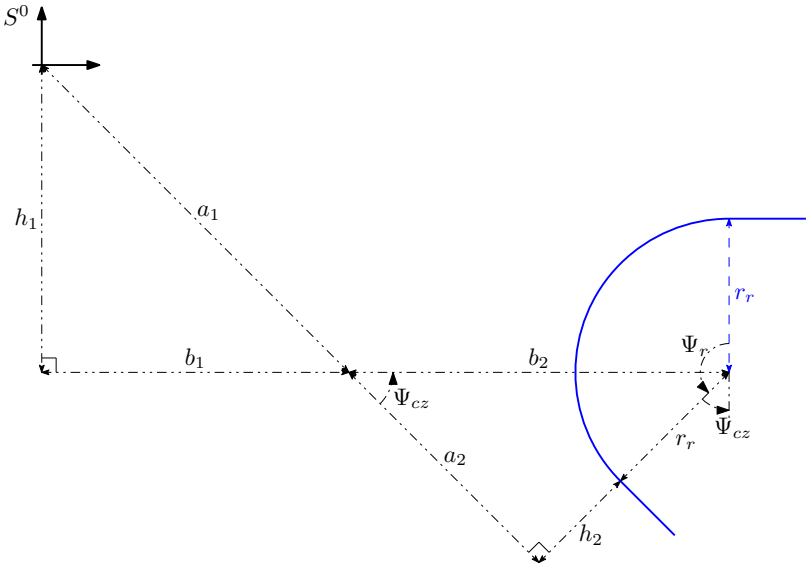

Figure 50: Geometry to calculate $a_r = a_1 + a_2$ and $b_r = b_1 + b_2$.

First we consider

$$\sin \Psi_{cz} = \frac{h_1}{a_1}, \tag{A.12}$$

and

$$\tan \Psi_{cz} = \frac{r_r + h_2}{a_2}, \tag{A.13}$$

so,

$$a_r = h_1 \csc \Psi_{cz} + (r_r + h_2) \cot \Psi_{cz}; \tag{A.14}$$

from the intersection's geometry we know that $\Psi_{cz} = \pi - \Psi_r$, $h_1 = \frac{1}{2} w_f$, and $h_2 = \frac{1}{4} w_o$. Thus,

$$a_r = \tfrac{1}{2} w_f \csc \Psi_r - (r_r + \tfrac{1}{4} w_o) \cot \Psi_r. \tag{A.15}$$

Now we consider

$$\tan \Psi_{cz} = \frac{h_1}{b_1}, \tag{A.16}$$

and

$$\sin \Psi_{cz} = \frac{r_r + h_2}{b_2} \tag{A.17}$$

so,

$$b_r = h_1 \cot \Psi_{cz} + (r_r + h_2) \csc \Psi_{cz}. \tag{A.18}$$

Therefore,

$$b_r = -\tfrac{1}{2} w_f \cot \Psi_r + (r_r + \tfrac{1}{4} w_o) \csc \Psi_r. \tag{A.19}$$

Note that the calculation of a_r and b_r holds for $w_p \geq w_s$ and $w_p < w_s$. Additionally, the derivation presented above also holds for the case in which the vehicle enters the competition zone on the principal road with $w_o = w_p$, $w_f = w_s$, and $\Psi_r = \Psi_{cz}$.

References

[1] "Interoperable GCDC automation experience (iGame): Description of work," 2013.

[2] "iGame deliverable 1.1: Specification of scenarios," 2014.

[3] J. Ploeg, N. van de Wouw, and H. Nijmeijer, "$\mathcal{L}_p$ string stability of cascaded systems: Application to vehicle platooning," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 786–793, 2014.

[4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, 2006.

[5] L. Iftekhar and R. Olfati-Saber, "Autonomous driving for vehicular networks with nonlinear dynamics," in *IEEE Intelligent vehicles symposium*, june 2012, pp. 723–729.

[6] M. Zavlanos and G. Pappas, "Dynamic assignment in distributed motion planning with local coordination," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 232–242, 2008.

[7] M. Wolf and J. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *IEEE Int. conf. on robotics and automation (ICRA)*, May 2008, pp. 3731–3736.

[8] D. Chong, S. Shadden, J. Marsden, and R. Olfati-saber, "Collision avoidance for multiple agent systems," in *IEEE Conf. on Decision and Control (CDC)*, December 2003, pp. 539–543.

[9] R. VOLPE and P. KHOSLA, "Manipulator control with superquadric artificial potential functions: Theory and experiments," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 6, pp. 1423–1436, 1990.

[10] A. Morales, "Control reconfiguration with mixing applied to a longitudinally-controled vehicle," Eindhoven University of Technology DC 2013.059, 2013.