Alma Mater Studiorum · Università di Bologna

SCHOOL OF ENGINEERING

DEPARTMENT OF
ELECTRICAL, ELECTRONIC, AND INFORMATION ENGINEERING
"Guglielmo Marconi"
DEI

MASTER'S DEGREE IN
ADVANCED AUTOMOTIVE ELECTRONIC ENGINEERING

MASTER'S THESIS IN
PLATFORMS AND ALGORITHMS FOR AUTONOMOUS DRIVING

Rapid Control Prototyping of an L2+ Highway Assist System

CANDIDATE:
Francesco Prignoli

SUPERVISOR:
Chiar.mo Prof. Paolo Pavan

CO-SUPERVISOR:
Chiar.mo Prof. Paolo Falcone

COMPANY SUPERVISOR:
Ing. PhD Roberto Rabbeni

Academic Year 2020/2021

Session I

Abstract

Advanced Driver Assistance Systems (ADAS) are proving to have huge potential in road safety, comfort, and efficiency. In recent years, car manufacturers have equipped their high-end vehicles with Level 2 ADAS, which are, according to SAE International, systems that combine both longitudinal and lateral active motion control. These automated driving features, while only available in highway scenarios, appear to be very promising towards the introduction of hands-free driving. However, as they rely only on an on-board sensor suite, their continuative operation may be affected by the current environmental conditions: this prevents certain functionalities such as the automated lane change, other than requiring the driver to keep constantly the hands on the steering wheel. The enabling factor for hands-free highway driving proposed by Mobileye is the integration of high-definition maps, thus leading to the so-called Level 2+.

This thesis was carried out during an internship in Maserati's Virtual Engineering team. The activity consisted of the design of an L2+ Highway Assist System following the Rapid Control Prototyping approach, starting from the definition of the requirements up to the real-time implementation and testing on a simulator of the brand new compact SUV *Maserati Grecale*. The objective was to enhance the current Level 2 highway driving assistance system with hands-free driving capability; for this purpose an *Autonomous Lane Change* functionality has been designed, proposing a Model Predictive Control-based decision-maker, in charge of assessing both the feasibility and convenience of performing a lane-change maneuver. The result is a Highway Assist System capable of driving the vehicle in a traffic scenario safely and efficiently, never requiring driver intervention.

Acknowledgments

The work presented in this thesis is the result of an internship carried out in Maserati which lasted four months; I would like to thank the company supervisor Roberto, who was able to understand my attitudes and support me in taking the right direction; in the same way, I thank Gianfranco, who was a constant point of reference throughout the duration of the activity, always able to guide me towards the solution to every obstacle. I also thank all the members of the Virtual Engineering team, Chiara, Davide, Matteo, and my classmate Leonardo, for having been a wonderful group, even outside of work.

Thanks go to the supervisors Prof. Paolo Pavan and Prof. Paolo Falcone, who supported me despite the complications due to the pandemic and the restrictions due to the confidentiality of the material.

Finally, I want to thank my family, who gave me the opportunity to complete this journey, as well as my girlfriend and all the people and friends who have been close to me.

Francesco Prignoli

Contents

List of Figures

Chapter 1

Introduction

Advanced Driver Assistance Systems (ADAS) have acquired great attention over the years, thanks to their ability to enhance driving comfort and even more safety; according to NHTSA, human error is responsible for about 94% of serious road accidents [1]. Here, automated driving could ideally remove the human factor by supporting and correcting the driver behavior, thus leading to a drastic reduction of car crashes.
SAE International provides a standard for the taxonomy and definitions of six levels of driving automation [2] (Figure 1.1): at the current state, *Partial Driving Automation* (Level 2) systems appear to offer the best trade-off between development and implementation costs and benefits in terms of safety, comfort, but also efficiency end economics.

Level 2 ADAS provide both steering and brake/acceleration capability at the same time by combining lateral and longitudinal motion control; typically they rely on a sensor suite composed of one long-range radar, four mid-range corner radars and a front camera. It is worth noting that the operative conditions of this kind of system are often limited to highways and limited-access roads; the main features that make these scenarios particularly favorable are:

- Predictability: the absence of vulnerable road users (i.e. pedestrians and cyclists) removes most of the sources of highest unpredictability, thus ensuring higher effectiveness of automated features.
- Road morphology: typically the road curvature is relatively low, thus allowing the lateral controller to apply small and smooth steering adjustments; moreover, both radars and cameras are not obstructed by narrow curves or high road slopes. Lastly, the lane boundaries are clearly visible, facilitating the camera in the detection.

Nevertheless, there are still several factors that prevent the current technologies from reaching the upper Level 3, thus requiring the driver to monitor constantly the environment and take control of the vehicle if needed; this typically means that the driver must always keep his hands on the steering wheel. Major limiting factors are those that interfere with the sensing system, such as adverse weather conditions (i.e. snow, fog, heavy rain), affecting the camera visibility and thus worsening both lane and object detection.
In 2017 Mobileye [3] conceived an additional level to those already defined by SAE, called

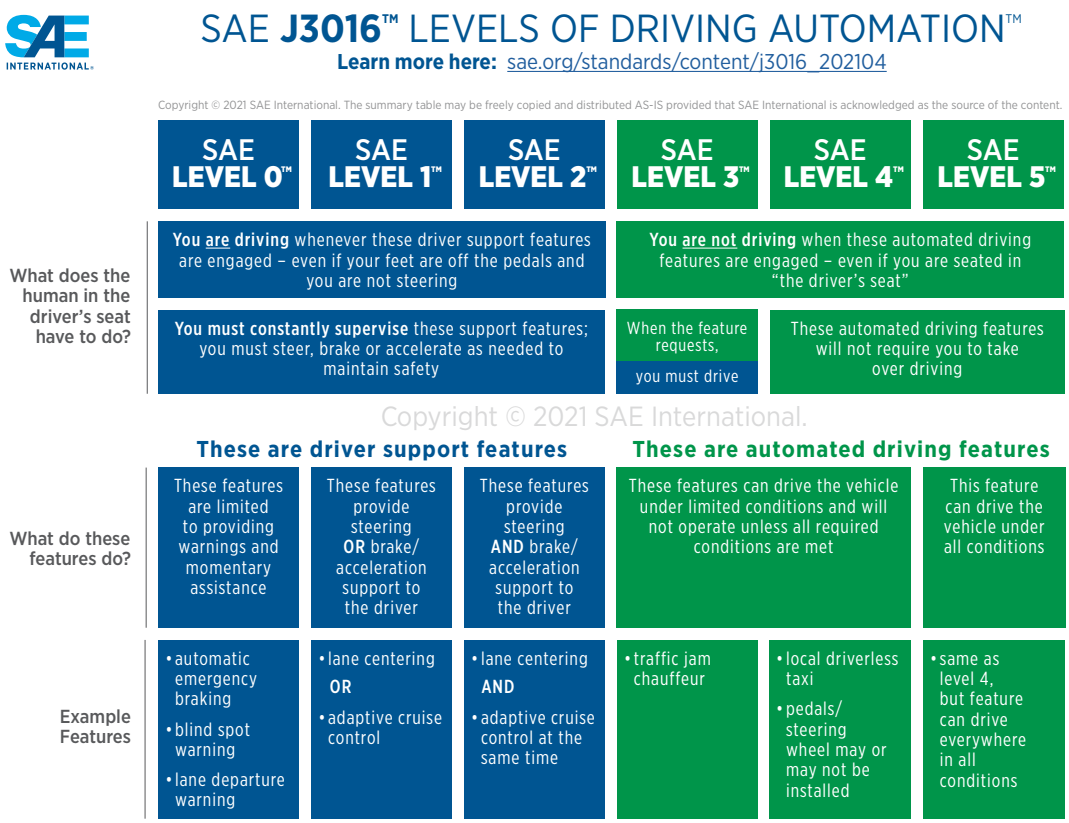


Figure 1.1: SAE J3016 Visual Chart [2]

L2+, reducing the gap between the Level 2 and 3; the key feature of the functionalities belonging to this category is the exploitation of a crowdsourced high-definition map, augmenting the sensing system and providing extended operative conditions, even in adverse scenarios. High-precision maps are intended to provide precise information about road geometry and semantics, thus enhancing the sensor fusion and extending the control system functionalities. This results in enabling the first semi-autonomous features, such as hand-free highway driving and automatic lane change.

Virtual Engineering

Virtual engineering has gained a central role in industrial development process, providing advanced tools and methodologies that make possible to move from the field to the lab. This leads to several benefits, such as:

- Time-to-market reduction: early testing and automatic testing drastically reduce the time-to-market, thus having an heavy economic impact.
- Costs reduction: virtual engineering methodologies are enabling factors for strong reuse, thanks to the exploitation of versatile tools, thus avoiding the need for application-specific

hardware and prototypes.

- Safety enhancement: testing and validation in virtual environments enhance safety, other than offering the possibility to realize arbitrary complex scenarios, without preventing repeatability.

As far as the ADAS world is concerned, virtual engineering solutions are spreading in both the development and validation domain, thanks to the availability of real-time simulators and highly detailed simulation models, including vehicle dynamics, electrical and mechanical components, ECUs, sensors, and complex traffic scenarios. These are the central elements for virtual validation and testing methodologies such as hardware-in-the-loop and driver-in-the-loop, but also for design methodologies, such as model-based design and rapid control prototyping.

Thesis outline

This thesis describes the results obtained during an internship carried out in the Virtual Engineering Team at Maserati. The design of a Level 2+ Highway Assist System is presented, enhanced with an Autonomous Lane Change functionality that makes it capable of fully hands-free highway driving.
Chapter 2 introduces the design objectives, detailing the functionalities that the system must implement and specifying the assumptions made about the sensing interface.
Chapter 3 describes how the Model Predictive Control framework was used to design both the longitudinal and the lateral controller, highlighting how it allows implementing the required functionalities in an intuitive and effective way.
Chapter 4 presents the solution that has been proposed for the implementation of the Autonomous Lane Change functionality: the problem has been split into trajectory planning and decision-making. The latter was implemented by using an approach that still exploits MPC in order to require minimal design effort while being precise and functional.
The first simulation results are shown in Chapter 5: these were performed offline exploiting a simplified model for the vehicle dynamics, allowing to speed up the tuning phase.
In the end, Chapter 6 describes the implementation of the designed HAS on a real-time simulator. This has been used for the verification and testing phase, exploiting various test cases, including Euro NCAP scenarios.

Chapter 2

Highway Assist System

Maserati introduced the Highway Assist System in 2018, empowering its vehicles with an L2 ADAS that combines longitudinal and lateral motion control when the vehicle is traveling on highways or limited access freeways.
These kinds of driver assistance systems result from the merger of multiple Level 1 ADAS, thus giving rise to automated driving functionalities that require minimal intervention from the driver: for this reason, they seem to be next to the introduction of hands-free highway driving. More in detail, the longitudinal dynamics is handled by the *Adaptive Cruise Control with Stop&Go*: a long-range front radar collects measures about the distance and speed of a leading target vehicle; basing on this information, the system is capable track the desired speed set by the driver while keeping a safe distance to the target and, if necessary, to perform a full-stop. If this happens, once the leading vehicle starts moving, the system is capable to start the motion without the driver's intervention. In addition, the *Lane Centering* functionality exploits the detection of lane boundaries and road curvature provided by the front camera sensor to maintain the vehicle close to the centerline, by actively controlling the steering wheel.
It is important to highlight that, at the current state, these systems are designed to provide driver assistance only and not to work autonomously: the driver must constantly supervise the environment in order to take the control of the vehicle as soon as the operative conditions are no longer matched. Nevertheless, as previously mentioned, L2+ functionalities offer extended sensing capabilities thanks to the exploitation of high-definition maps, so that ensuring the continuous operation of the system; this is an enabling factor for new automated driving features, such as hands-free highway driving and automatic lane change.

2.1 Assumptions and design objectives

The major assumptions are related to the sensing interface since signal acquisition and processing are beyond this dissertation's scope. More in detail, it is assumed that the vehicle is equipped with a sensor suite enhanced with high-definition maps so that the following signals are available:

- Distance, speed and acceleration of up to six vehicles; these are the closest front and rear

targets on the current, left and right adjacent lanes respectively (Figure 2.1); the front and rear radars range are assumed to be 200 and 100 meters respectively.

- Speed and acceleration of the ego vehicle.
- Road curvature preview; the camera range is assumed to be 60 meters.
- Lateral position and yaw errors with respect to the current lane centerline.
- Lane marks typology, i.e. solid or dashed.

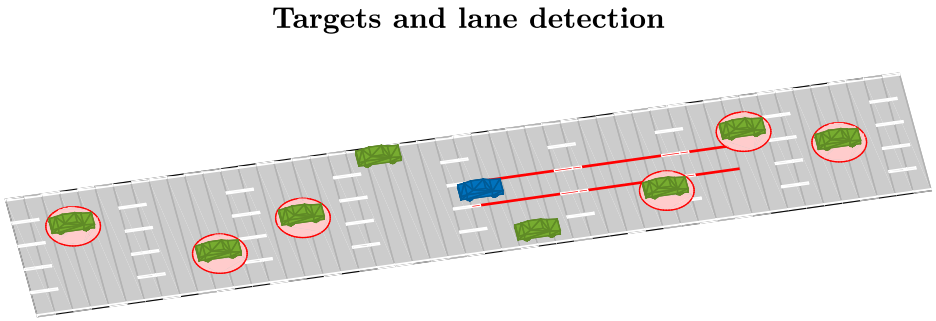


Figure 2.1: Radars and camera detections

Starting from this set of information, the goal is to design an L2+ Highway Assist System enhanced by the hands-free driving capability that implements the following functionalities:

- *Cruise Control (CC)*: the vehicle must be capable to track the desired speed value set by the driver;
- *Adaptive Cruise Control with Stop&Go (ACC)*: the vehicle must be capable to adapt its speed such that it keeps a safe distance from the leading target; if the latter stops, the vehicle must be capable of fully braking as well and restart as soon as the conditions are met;
- *Intelligent Adaptive Cruise Control (iACC)*: the vehicle must be capable to adapt its speed depending on other environmental conditions (i.e. road curvature);
- *Lane centering (LC)*: the vehicle must be capable to keep itself close to the lane centerline;
- *Autonomous Lane Change (ALC)*: the vehicle must be capable to perform a left or right lane change autonomously, namely both the maneuver and the decision-making.

It is worth noting that the resulting system should be capable to drive the vehicle ensuring comfort, safety and efficiency while requiring a minimal driver intervention.

Design considerations

The overall controller, namely the Highway Assist System, is structured as the composition of three main blocks:

- Longitudinal Controller: it implements the functionalities related to the longitudinal dynamics, i.e. the CC, ACC and iACC;
- Lateral Controller: it manages those functionalities involving the lateral motion control, namely the Lane Centering and the lane change maneuver.
- Lane Change Decision-Maker: it is in charge of continuously evaluating both the feasibility and convenience of a lane change maneuver.

Note that the separation between longitudinal and lateral controller is based on having assumed the decoupling between the longitudinal and lateral dynamics. Figure 2.2 shows the block scheme relative to the integration of these three main components.

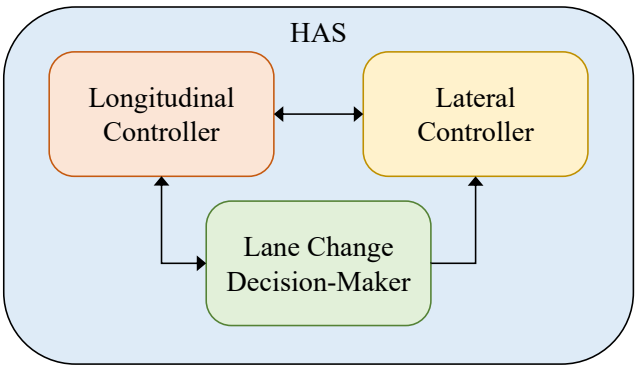


Figure 2.2: HAS high-level architecture

The resulting control system has to be designed by taking into account the final integration in the vehicle electrical architecture; more in detail, as a part of the ADAS Electronic Control Unit, the HAS will interface to the actuation domain through the proper ECUs, that are the Electronic Power Steering (EPS) for the lateral dynamics, and the Engine Control Module (ECM) and Brake System Module (BSM) for the longitudinal dynamics (Figure 2.3); hence, the designed HAS acts as a high-level controller that produces as outputs the desired steering angle and acceleration: these are received by the respective ECUs, that are in charge of controlling at low-level the actuators.

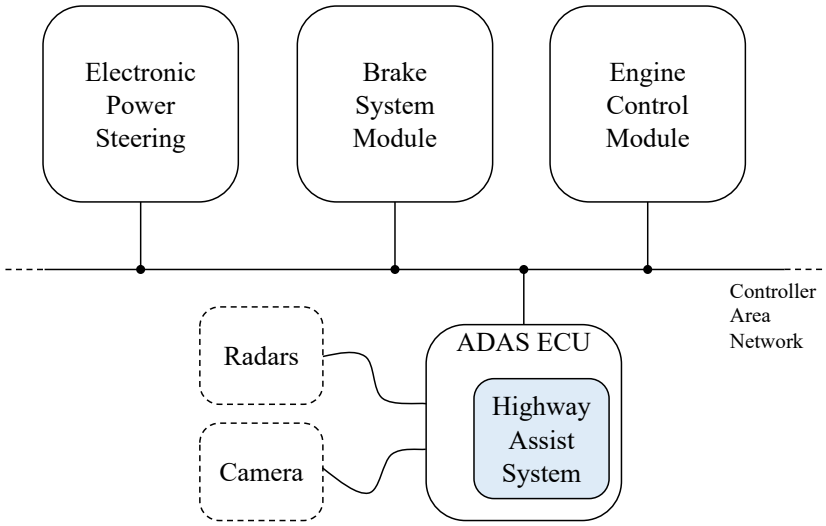


Figure 2.3: Communication scheme of the main ECUs involved by the HAS

Chapter 3

Vehicle motion control

The control framework that is used for both the longitudinal and lateral motion control is the *Model Predictive Control (MPC)* [4][5][6]. The main advantages for which this approach was chosen are:

- The ability to react to future events: this allows to take full advantage of the information provided by the detection system, ensuring smoother maneuvers.
- The predisposition to manage constraints: this allows to handle physical, safety and comfort constraints intuitively and with minimal implementation effort.

For these reasons, Model Predictive Control is particularly suited to vehicle motion control applications.

3.1 Model Predictive Control

The idea behind MPC is to exploit a discrete-time model to solve a constrained optimization problem over a finite *prediction horizon*. At each sampling time (*control interval*):

1. the model is updated with the current plant state (measure or estimate);
2. the optimization problem is solved over the next prediction horizon;
3. only the first move of the optimal control sequence is applied during the next control interval, whereas the remaining moves are discarded.

Thus, at each time step, the time horizon is shifted forward, from which also the name *Receding Horizon Control*.

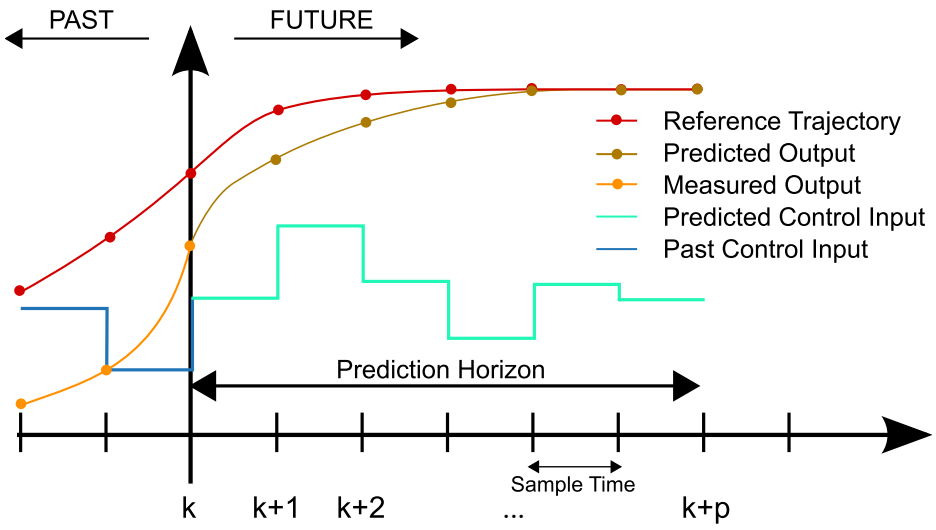


Figure 3.1: MPC scheme [7]

More in detail, the optimization problem is a *Quadratic Program (QP)*, since the control objective is described by a quadratic cost function J [8]. For the purpose of this thesis, its expression is[1]:

$$J(z_k) = \sum_{i=1}^{p} \sum_{j=1}^{n_y} W_{y,j}^2 [y_{ref,j}(k+i|k) - y_j(k+i|k)]^2 + W_u^2 u(k+i|k)^2 + W_{\Delta u}^2 \Delta u(k+i|k)^2 + W_\varepsilon \varepsilon_k^2 \tag{3.1}$$

where:

- k is the current *control interval*, namely the discrete sample time;
- $z_k = [u(k|k) \ \dots \ u(k+p-1|k) \ \varepsilon_k]^t$ is the control sequence (*QP decision*);
- p is the prediction horizon;
- n_y is the size of the model output y;
- $W_\star$ is the penalty weight associated to $\star$ (constant over the prediction horizon);
- ε_k is the *slack variable*, used for constraints softening.

Thus, the objective is to find the control sequence z_k that minimizes the cost function $J(z_k)$ subject to a set of inequality constraints specified as bounds on the output y, the control input u and control increment Δu:

$$\begin{cases} y_{j,min}(k+i|k) - \nu_{y_j,min}\varepsilon_k & \leq y_j(k+i|k) \leq y_{j,max}(k+i|k) + \nu_{y_j,max}\varepsilon_k \\ u_{min}(k+i|k) - \nu_{u,min}\varepsilon_k & \leq u(k+i|k) \leq u_{max}(k+i|k) + \nu_{u,max}\varepsilon_k \\ \Delta u_{min}(k+i|k) - \nu_{\Delta u,min}\varepsilon_k & \leq \Delta u(k+i|k) \leq \Delta u_{max}(k+i|k) + \nu_{\Delta u,max}\varepsilon_k \end{cases} \tag{3.2}$$

[1] Assuming scalar control input u.

Here $\nu_\star$ represents the tuning weight associated with the slack variable ε_k, through which the constraints are softened in the event that their violation is unavoidable; note that it is possible to define *hard* constraints by setting the respective $\nu_\star$ equal to zero.
Furthermore, it may be convenient to set constraints on a linear combination of the outputs [9]:

$$F_k y(k+i|k) \leq G_k + \nu_F \varepsilon_k \tag{3.3}$$

Overall, at each time step, the optimizer solves:

$$\min_{z_k} J(z_k)$$
$$\text{subject to} \begin{cases} (3.2) \\ (3.3) \end{cases}$$

3.2 Longitudinal control

Prediction model

The formulation of the vehicle longitudinal control problem starts with the definition of the dynamic model that is used in the prediction phase. As discussed in Section 2.1, the control input is given by the demanded acceleration a_{dem} and it is possible to approximate the dynamics of the low-level controller as a first order system with time constant τ:

$$\dot{a}_{ego} = \frac{1}{\tau}(a_{dem} - a_{ego})$$

where a_{ego} represents the actual acceleration of the ego vehicle. Then the speed v_{ego} and the traveled space s_{ego} are obtained by integration, such that:

$$\dot{s}_{ego} = v_{ego}$$
$$\dot{v}_{ego} = a_{ego}$$

Eventually, the distance d from a leading target vehicle is computed as:

$$d = d_0 + s_{target} - s_{ego}$$

where d_0 is the measured distance between the two vehicles at each time step, whereas s_{target} is the estimate of the traveled space by the target.
This set of equations gives rise to a Linear-Time-Invariant model, described by the following

state-space representation:

$$\frac{d}{dt}\begin{bmatrix} s_{ego} \\ v_{ego} \\ a_{ego} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} s_{ego} \\ v_{ego} \\ a_{ego} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} a_{dem} \tag{3.4}$$

$$y = \begin{bmatrix} v_{ego} \\ d \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_{ego} \\ v_{ego} \\ a_{ego} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} S_{target} \tag{3.5}$$

where $S_{target} \triangleq d_0 + s_{target}$. Note that, at each time step the first component of the state s_{ego} is updated with the zero value, whereas v_{ego} and a_{ego} with the current measures of the vehicle speed and acceleration respectively. On the other hand, S_{target} is the preview of the front target traveled space, thus, once discretized, it is a vector of length $p + 1$, where p is the prediction horizon; assuming a constant-acceleration model, the i-th component of the target space preview is:

$$s_{target}(i) = v_{target} i T_s + \frac{1}{2} a_{target} (i T_s)^2$$

It is worth noting that the choice to keep the target traveled space as a measured disturbance instead of including it within the state is justified by three main reasons:

- Efficiency: it reduces the number of states, decreasing the computational load of the MPC controller.
- Accuracy: it possibly allows to use of a custom estimator instead of a simple constant-speed or constant-acceleration model, which could be unsuitable if the prediction horizon is relatively long.
- Flexibility: it allows to easily handle the prediction of a switch among different targets (i.e. during a lane-change maneuver).

MPC parameters

Once the continuous-time model has been defined, it has to be discretized[2] with a proper sample time T_s, that coincides with the controller execution period and thus with the control interval. In this case, a trade-off between response time and prediction horizon length has been found by setting $T_s = 100\ ms$.

The choice of the prediction horizon must take into account that in a highway scenario the controller should react to the presence of a target by exploiting the full range of the front RADAR; given a prediction horizon of $P = p \cdot T_s$ seconds (where p is an integer number), the controller will react to the presence of targets within a space horizon equal to $S = v_{ego} \cdot P$ meters[3]. Hence,

[2] The adopted discretzation method is the *zero-order hold (ZOH)*.

[3] Assuming constant speed and not considering the constraint on the safe distance.

assuming a minimum speed of 90 km/h (25 m/s), a prediction horizon of 8 seconds is needed to exploit the 200 meters RADAR range; thus:

$$p = \frac{P}{T_s} = 80$$

Regarding the control horizon m, a good trade-off between computational load and performance has been found by setting it to half of the prediction horizon:

$$m = 40$$

In addition, the control horizon has been divided in a series of 20 *blocking intervals* with size 2, meaning that the optimization process will compute an optimal sequence of control moves U such that:

$$U(2k-1) = U(2k) \qquad k = 1 \dots \frac{m}{2}$$

This approach allows to increase even more efficiency and to smooth the computed control adjustments while keeping a relatively large control horizon.

Cost function and constraints

The problem of the longitudinal control can be stated as finding the optimal control sequence that minimizes the speed error with respect to a reference value while keeping the acceleration and jerk within predefined bounds and never exceeding the safe distance to the front target. This formulation perfectly suits the MPC problem statement by properly defining the cost function and constraints; the Equation (3.1) takes the form of:

$$J(z_k) = \sum_{i=1}^{p} W_v^2[v_{ref}(k+i|k) - v_{ego}(k+i|k)]^2 + W_u^2 u(k+i|k)^2 + W_{\Delta u}^2 \Delta u(k+i|k)^2 + W_\varepsilon \varepsilon_k^2 \tag{3.6}$$

where the weights are assumed to be constant over the whole prediction horizon p.
Here, W_v represents the penalty weight associated with the speed error: the higher this value, the more aggressive the tracking of the reference speed v_{ref}; W_u and $W_{\Delta u}$ are the weights related to respectively the demanded acceleration and jerk: the tuning of these parameters allows to achieve a smoother response, by penalizing abrupt acceleration and braking.

At this stage, the controller implements a basic Cruise Control system, that is able to track a reference speed. The key element that allows accounting for comfort and physical limitations and even more to include the ACC and iACC functionalities is the introduction of constraints in the optimization problem. Comfort and physical constraints involve the acceleration and jerk; it is worth noting that for a Full-Speed-Range ACC it may be appropriate to define speed-dependent bounds (Figure 3.2). Regarding the acceleration, typical values are $[-5; -3.5]\ m/s^2$ and $[2.5; 5]\ m/s^2$ for respectively the lower and upper limit ranges [10].

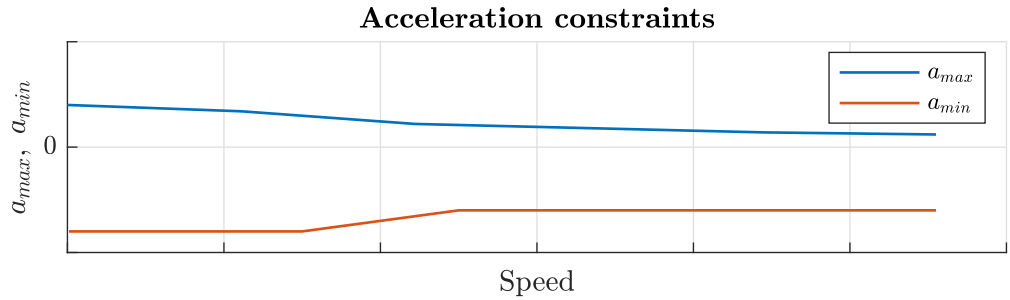


Figure 3.2: Speed-dependent constraints on the demanded acceleration

More in detail, the upper and lower bounds for the demanded acceleration are stored in a Look-Up Table, that is addressed by the current speed of the ego vehicle; since the prediction horizon is relatively large, the speed could experience large variation over the horizon and thus also the control action bounds; hence, at each time step, it is appropriate to exploit the prediction of the speed computed at the previous time step to estimate a preview of the constraints for the next horizon.

As far as the safe distance to the front target is concerned, it is computed as the product of the current speed and the so-called *Time Gap*:

$$d_{safe} = v_{ego} \cdot T_{gap}$$

Thus, the simple CC can be extended to an ACC by imposing a constraint on the linear combination of the outputs y (Equation (3.5)):

$$d \geq v_{ego} \cdot T_{gap} \iff \begin{bmatrix} T_{gap} & -1 \end{bmatrix} \begin{bmatrix} v_{ego} \\ d \end{bmatrix} \leq 0$$

Again, the Time Gap is a speed-dependent parameter, other than being adjustable by the driver (Figure 3.3).

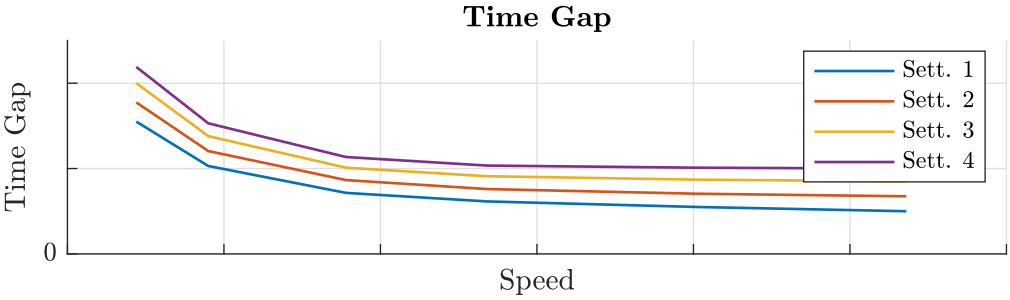


Figure 3.3: Speed-dependent Time Gap at four different driver settings

It is worth noting that at very low speed the concept of Time Gap is not applicable, since the computed safe distance would go to zero regardless of the value of the Time Gap; to solve this inconvenient, it is also defined an absolute constraint on d, representing the stopping distance

in case of full-stop:

$$d \geq d_{stop}$$

Following the same approach, it is immediate to include the iACC functionality by adding a constraint on the vehicle speed. During a curve, the lateral acceleration a_y can be computed as:

$$a_y = v_{ego}^2 \cdot \rho$$

where ρ is the curvature of the road. Hence, given the maximum acceptable lateral acceleration $a_{y,max}$, the constraint on the ego speed became:

$$v_{ego} \leq v_{max} \qquad v_{max} = \sqrt{\frac{a_{y,max}}{\rho}}$$

In addition, given the curvature preview, it is possible to feed the controller with the preview of the speed upper bound, thus giving the possibility to decelerate in time if needed, while respecting the other comfort constraints.

Recalling the cost function (Equation (3.6)), the last term $W_\varepsilon \varepsilon_k^2$ represents the cost associated to the *slack variable* ε_k at the time instant t_k: this allows handling situations in which constraints violation is unavoidable by softening them. A typical example is a cut-in scenario, that is a front vehicle coming from the adjacent lane performs a lane change despite the safe distance is not respected. If this happen, the optimization problem became unfeasible, thus preventing the controller to compute the next move. To solve this limitation, the constraint on the safe distance is softened by setting a proper weight $\nu_F > 0$ (see Equation (3.3)). Similarly, it is appropriate to soften the lower bounds of the acceleration and jerk, thus allowing to handle situations in which the comfort constraints may be too restrictive.

3.3 Lateral control

Prediction model

In opposite to the longitudinal motion, the model that describes the dynamics of the lateral position error e_y and yaw error e_ψ with respect to the reference path is Linear Time-Varying [11]:

$$\begin{aligned} \dot{x} &= A(v_x)x + B_1 u + B_2(v_x)d \\ y &= x \end{aligned}$$

where the state vector is $x = [e_y\ \dot{e}_y\ e_\psi\ \dot{e}_\psi]^t$, the control input u is the front wheel steering angle and the disturbance $d = \dot{\psi}_{des}$ is the yaw rate of the desired path. The latter can be approximated

as $\psi_{des} \simeq v_x \cdot \rho$, where ρ is the road curvature and v_x is the longitudinal component of the vehicle speed v_{ego}. The matrices are :

$$A(v_x) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\dfrac{2C_f + 2C_r}{mv_x} & \dfrac{2C_f + 2C_r}{m} & \dfrac{-2C_f l_f + 2C_r l_r}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\dfrac{2C_f l_f - 2C_r l_r}{I_z v_x} & \dfrac{2C_f l_f - 2C_r l_r}{I_z} & -\dfrac{2C_f l_f^2 + 2C_r l_r^2}{I_z v_x} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ \dfrac{2C_f}{m} \\ 0 \\ \dfrac{2C_f l_f}{I_z} \end{bmatrix} \qquad B_2(v_x) = \begin{bmatrix} 0 \\ -\dfrac{2C_f l_f - 2C_r l_r}{mv_x} - v_x \\ 0 \\ -\dfrac{2C_f l_f^2 + 2C_r l_r^2}{I_z v_x} \end{bmatrix}$$

As mentioned before, this time the prediction model is LTV; the solution to keep a QP optimization problem is to update at each time step the matrices computed with the current value of the ego vehicle speed; in addition, since during the prediction horizon the speed could experience large variations, it is possible to feed the controller with a preview of the matrices over the next horizon by exploiting the speed preview provided by the longitudinal controller.

MPC parameters

As before, the sample time $T_s = 100\ ms$ found to be a good trade off between the controller response time and the length of the prediction horizon; the latter has been sized basing on the range of the camera sensor: given the the prediction horizon p and the vehicle speed preview:

$$v_{ego}(k+i|k) \qquad i = 0 \dots p$$

during each optimization process the controller will react to a disturbance up to a distance:

$$s(k+p|k) = \sum_{i=0}^{p} v_{ego}(k+i|k) T_s$$

Hence, given the lowest speed value at which it is decisive to feed the full road curvature preview, it is possible to obtain the value of the prediction horizon. In this case, has been chosen

$$P = 4\ s \quad \Longrightarrow \quad p = \frac{P}{T_S} = 40$$

ensuring the exploitation of the full 60 meters curvature preview at $v_{ego} \geq 15$ m/s (54 km/h).

It is worth noting that at higher speed the controller must neglect the disturbance preview that

is farther than the camera range since these values are unknown; said α_k such that $s(k+\alpha_k|k) = camera\ range$, then the preview $[\rho(k+\alpha_k+1|k)\ \ldots\ \rho(k+p|k)]$ must be properly handled; three main approaches are:

- Extend the actual preview for the remaining horizon with the last known value; this trivial solution may be suitable for a highway scenario, where the road curvature is slowly varying and bounded by relatively small values.
- Update online all the weights of the cost function by setting $[W_\star(k+\alpha_k+1|k)\ \ldots\ W_\star(k+p|k)] = 0$; this is equivalent to truncate the prediction horizon at the first future time instant at which the disturbance is unknown.
- Update online the length of the prediction horizon by computing at each time step the value of α_k and setting $p = \alpha_k$; the result is similar to the previous one but, in addition, it may reduce the computational load.

The last method has been implemented since it suits both effectiveness and efficiency requirements.

As far as the control horizon is concerned, the choice is $m = p$: this is justified by the fact that some maneuvers (i.e. lane change) require a highly variable control sequence over the whole prediction horizon. Hence, limiting the control horizon may interfere with other constraints, resulting in a worsening of the reference tracking performance. Again, control sequence blocking with a block size equal to 2 is implemented to get smoother steering and reduce the computational load.

Cost function and constraints

The output of the prediction model is composed by the lateral position error, the yaw error and their derivatives; since the objective is to minimize the errors with respect to the reference path, the cost function can be written as:

$$J(z_k) = \sum_{i=1}^{p} \sum_{*\in\{e_y, e_\psi\}}^{p} W_*^2[*_{ref}(k+i|k) - *(k+i|k)]^2 + W_u^2 u(k+i|k)^2 + W_{\Delta u}^2 \Delta u(k+i|k)^2$$

Note that the tuning weights associated with the position and yaw error should be chosen such that the former is penalized more than the latter: the reason is that, at steady state while traveling a curve, a non zero yaw error is allowed due to the slip angle; hence it is not possible to zero both the position and yaw errors. In addition, the cost function includes the control action u and its rate Δu to penalize large steering angles and abrupt moves.

For the same reason, it is appropriate to impose constraints on these quantities, thus limiting the maximum and minimum allowed values. Note that, in order to handle different requirements associated with standard driving conditions and lane-change maneuvers, the constraints are updated online, i.e. the bounds on the lateral position error are increased.

Chapter 4

Autonomous lane change

Highway driving assistance systems seem to be promising technologies able to introduce hands-free driving in the near future, by combining longitudinal and lateral motion control. However, in a realistic highway scenario, the presence of several actors could worsen the driving experience and therefore make them almost useless. More in detail, the main drawback is that sooner or later the ego vehicle will approach a slower vehicle (i.e. trucks), obstructing the current path and preventing to reach the desired speed; this event requires the action of the driver, that typically have to take the manual control of the vehicle and perform a lane change maneuver. The *Autonomous Lane Change* functionality solves this annoying limitation: this chapter proposes an approach for the design of this functionality by decoupling the problems of trajectory planning and decision-making.

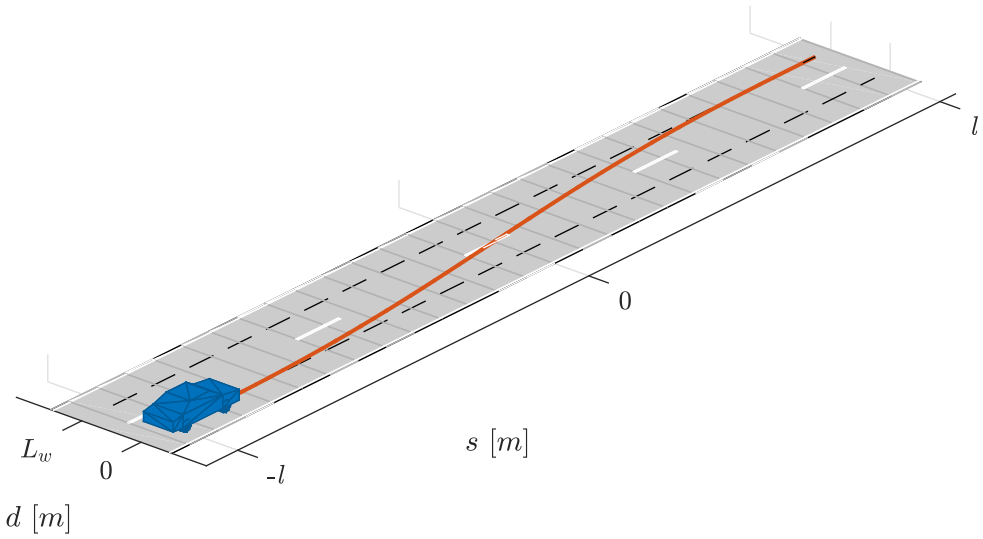
4.1 Trajectory planning

A simple but effective approach for the implementation of a lane-change maneuver is to feed the lateral motion controller with a constant reference for the lateral position error shifted by an offset equal to the lane width; nevertheless, a constant offset would lead to aggressive responses that must be suppressed by properly updating the weights and the constraints before starting the maneuver. A more flexible approach consists of computing a trajectory for the lateral position error according to all the safety and comfort constraints for the lateral motion, in terms of lateral speed, acceleration, and jerk; considering a left lane change and said $d(s)$ the lateral displacement with respect to the reference as a function of the traveled space on the reference path s, the following system of equations describes both the geometrical and comfort properties that the function $d(s)$ must satisfy:

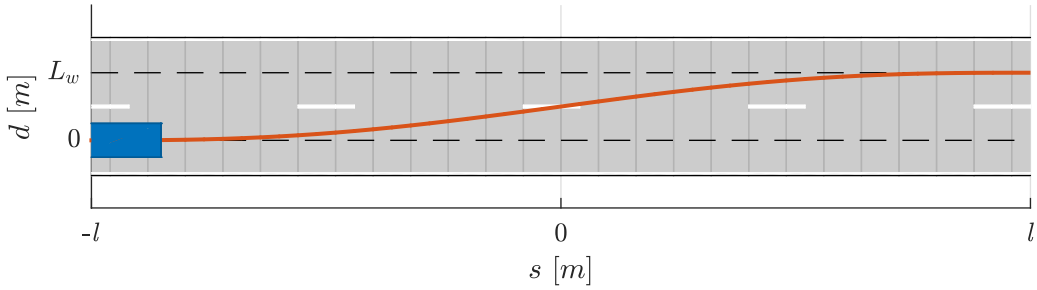
$$\begin{cases} d(s)\big|_{s=-l} & = 0 \\ d(s)\big|_{s=l} & = L_w \\ \dfrac{d}{dt}d(s)\big|_{s=\pm l} & = 0 \\ \dfrac{d^2}{dt^2}d(s)\big|_{s=\pm l} & = 0 \end{cases} \tag{4.1}$$

Here L_w is the lane width, whereas the terminal points are located in $s = \pm l$ (Figure 4.1); thus l represents the half-length of the maneuver and the initial distance to the line crossing; the last two pairs of equations impose to have zero lateral speed and acceleration in the terminal points, ensuring a smooth maneuver. The lowest order polynomial function able to solve this system of six equations is a fifth-order polynomial, and the solution is:

$$d(s) = \frac{15}{16} L_w \left(\frac{1}{5} \left(\frac{s}{l} \right)^5 - \frac{2}{3} \left(\frac{s}{l} \right)^3 + \frac{s}{l} + \frac{8}{15} \right) \tag{4.2}$$



(a) 3D view



(b) Bird's-eye plot

Figure 4.1: Polynomial reference offset for a left lane change

In addition, it is possible to obtain the analytical expressions for the lateral speed, yaw angle

and yaw rate relative to the reference path[1]:

$$\begin{cases} v_y^r = v_{ego}\dfrac{d}{ds}e_{y,ref} & = \dfrac{15}{16}\dfrac{v_{ego}L_w}{l}\left(1-\left(\dfrac{s}{l}\right)^2\right)^2 \\ \psi^r = \dfrac{d}{ds}e_{y,ref} & = \dfrac{15}{16}\dfrac{L_w}{l}\left(1-\left(\dfrac{s}{l}\right)^2\right)^2 \\ \dot{\psi}^r = v_{ego}\dfrac{d^2}{ds^2}e_{y,ref} & = -\dfrac{15}{4}\dfrac{v_{ego}L_w}{l^3}s\left(1-\left(\dfrac{s}{l}\right)^2\right) \end{cases} \tag{4.3}$$

Note that the same trajectory can be adapted for a right lane change maneuver by changing the signs of the previous expressions. Overall, the reference output for the lateral motion controller is composed by (Figure 4.2):

$$y_{ref} = \begin{bmatrix} e_{y,ref} \\ \dot{e}_{y,ref} \\ e_{\psi,ref} \\ \dot{e}_{\psi,ref} \end{bmatrix} = \begin{bmatrix} d \\ v_y^r \\ \psi^r \\ \dot{\psi}^r \end{bmatrix}$$

[1] It is assumed the small-angle approximation: $\arctan\psi \simeq \psi$ for $\psi \ll 1$

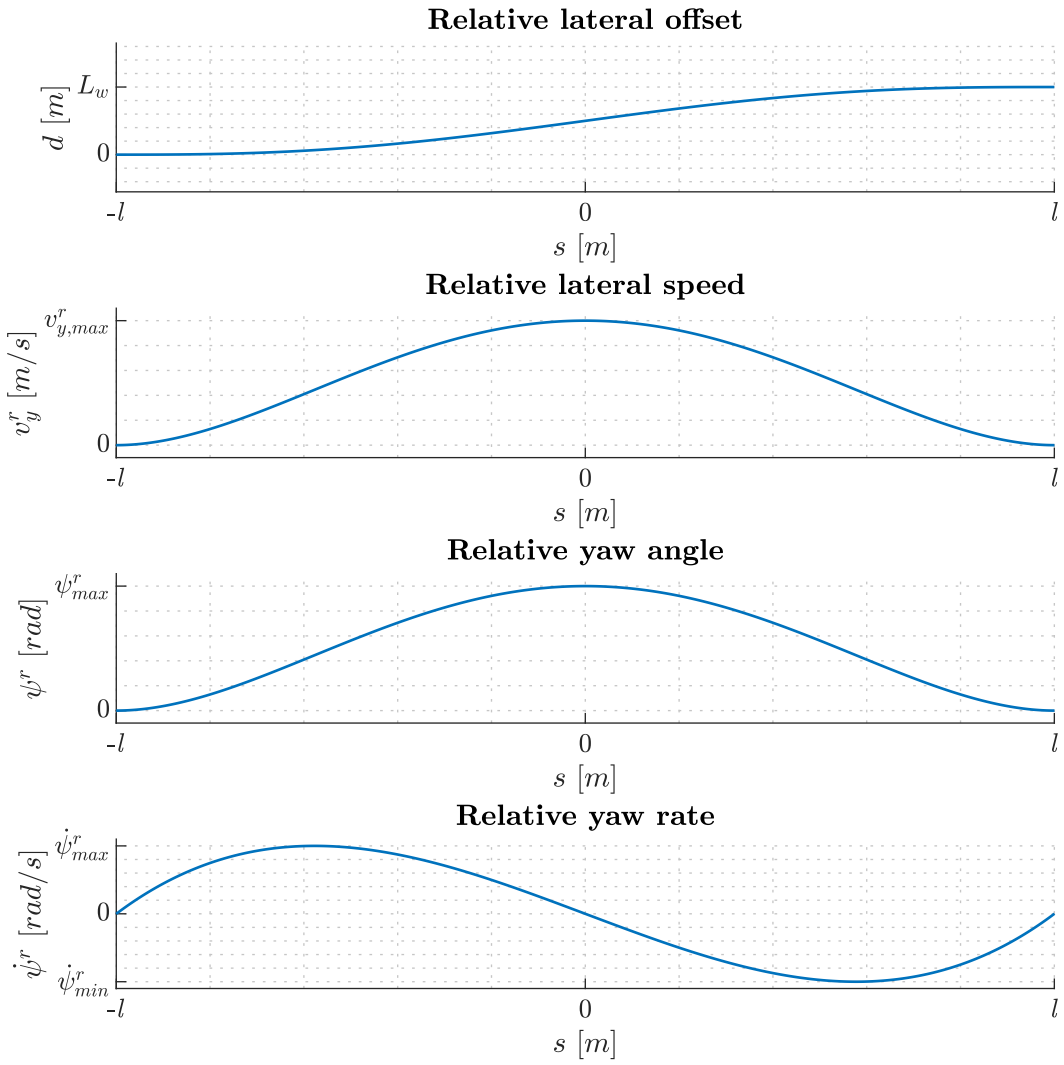


Figure 4.2: Reference output trend for a left lane change maneuver

Once fixed the lane width L_w and the ego vehicle speed v_{ego}, the only parameter left to be defined online is the maneuver half-length l: this must be done according to the constraints on the lateral motion and starting from the expressions of the maximum lateral speed, acceleration and jerk it is possible to compute l by inversion. Given the travel speed of the maneuver, these quantities can be obtained as successive time derivatives [2] of the lateral offset $d(s)$, exploiting the relation $\frac{d}{dt} = \frac{ds}{dt}\frac{d}{ds} = v_{ego}\frac{d}{ds}$:

[2] Assuming constant speed.

$$v_y^r(L_w, l, v_{ego}, s) = \frac{15}{16}\frac{v_{ego}L_w}{l}\left(\left(\frac{s}{l}\right)^2 - 1\right)^2$$
$$a_y^r(L_w, l, v_{ego}, s) = \frac{15}{4}\frac{v_{ego}^2L_w}{l^3}s\left(\left(\frac{s}{l}\right)^2 - 1\right)$$
$$j_y^r(L_w, l, v_{ego}, s) = \frac{15}{4}\frac{v_{ego}^3L_w}{l^3}\left(3\left(\frac{s}{l}\right)^2 - 1\right)$$

Finally, the maximum values expressed as functions of the lane width, the current speed and the half-length of the maneuver, are (Figure 4.3):

$$v_{y,max}^r(L_w, l, v_{ego}) = \frac{15}{16}\frac{v_{ego}L_w}{l} \tag{4.4}$$
$$a_{y,max}^r(L_w, l, v_{ego}) = \frac{5\sqrt{3}}{6}\frac{v_{ego}^2L_w}{l^2} \tag{4.5}$$
$$j_{y,max}^r(L_w, l, v_{ego}) = \frac{15}{2}\frac{v_{ego}^3L_w}{l^3} \tag{4.6}$$

Hence, by limiting these quantities it is possible to retrieve online the value for the half-length l that ensures a comfortable maneuver. A drawback is that, since the length of the trajectory is the only tunable parameter, it has to be chosen the such that it matches all among the bounds on the maximum speed, acceleration and jerk, despite it may be too conservative.

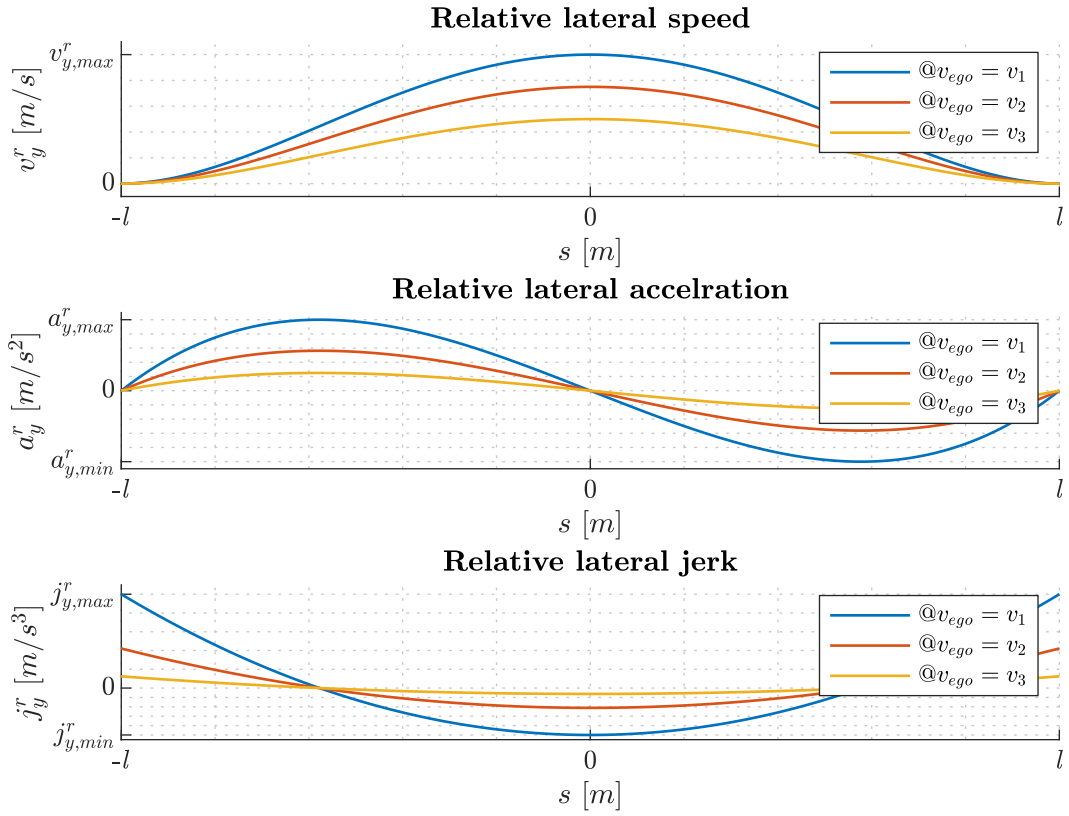


Figure 4.3: Lateral relative speed, acceleration and jerk at different ego vehicle longitudinal speeds: $v_1 > v_2 > v_3$

Implementation

As previously mentioned, the reference output for a lane change maneuver can be computed analytically; thus it is possible to generate the preview for the reference y_{ref} of the lateral motion controller, allowing a smoother response and smaller tracking errors (Figure 4.4). As soon as a lane change is requested, the algorithm starts to integrate the current speed of the vehicle[3], whose results is the variable s shifted by l; the latter is computed online by inverting the Equations (4.4) to (4.6). At each time instant, the reference preview is updated starting from the current value of s: it is possible to pass from the space domain to the time domain by integration of the speed preview provided by the longitudinal controller, obtaining the traveled space preview for each future time step of the prediction horizon. It is worth noting that since the position and yaw errors are computed with respect to the current lane, as soon as the vehicle crosses the line, e_y will experience a gap with magnitude equal to the lane width; to avoid abrupt steering, the same offset must be added to the reference $e_{y,ref}$.

[3]Assumption: $\dot{s} = v_{ego}\cos(e_\psi) \simeq v_{ego}$

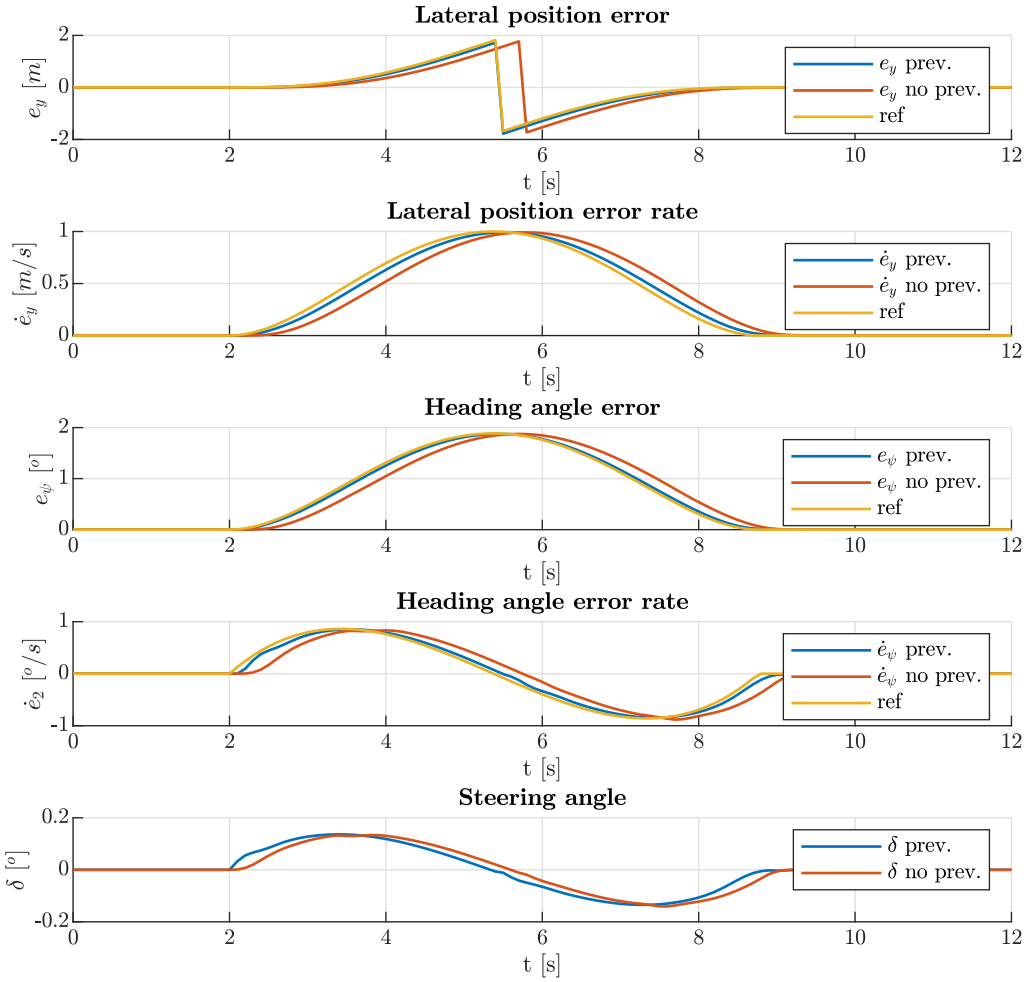


Figure 4.4: Comparison of lane change maneuvers at $v_{ego} = 110\ [km/h]$ with and without reference preview

Another benefit of planning a trajectory is that it is possible to estimate the distance (and time) left to the line crossing:

$$d_{cross} = l - s$$

This information is essential for avoiding unnecessary braking when the vehicle starts a lane change maneuver in presence of a front target. Given the vehicle speed preview, it is possible to estimate the future time instant at which the line crossing will occur:

$$t_{cross,k} = t_k + \alpha_k \cdot T_s$$

where k is the current discrete time instant. Thus the traveled space preview of the front target

is obtained by merging the one related to the current lane and the one related to the side lane as:

$$[S_{target}(k|k) \ \dots \ S_{target}(k+\alpha_k|k)] = [S_{front}(k|k) \ \dots \ S_{front}(k+\alpha_k|k)]$$

$$[S_{target}(k+\alpha_k+1|k) \ \dots \ S_{target}(k+p|k)] = [S_{side}(k+\alpha_k+1|k) \ \dots \ S_{side}(k+p|k)]$$

where p is the prediction horizon. Figure 4.5 shows the impact of previewing the target switch due to the lane change: as soon as the ego vehicle approaches a stationary target it starts braking; after 0.5 seconds a lane change is requested: if the target distance preview is only related to the current lane, then the ego vehicle continues to brake as it would perform a full-stop, regardless it started a lane change maneuver; as soon as it crosses the line, it stops to decelerate since the lane is now free. On the other hand, by previewing the line crossing, the longitudinal controller has to fulfill the constraints on the safe distance only for a fraction of the prediction horizon, avoiding unnecessary braking.

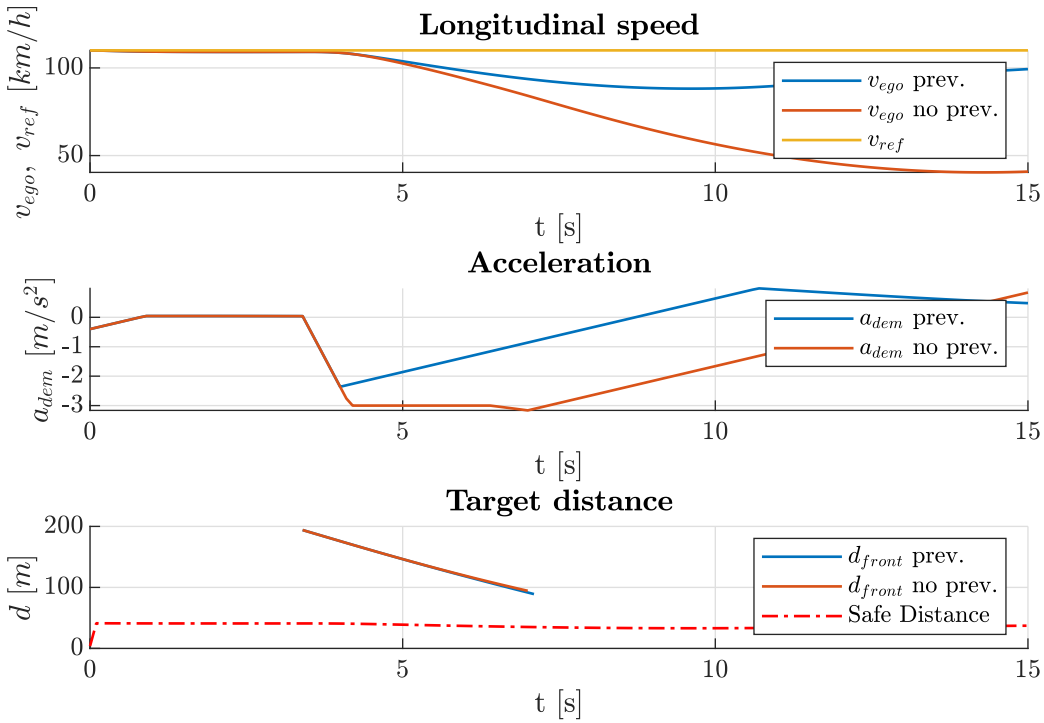


Figure 4.5: Comparison of lane change maneuver in presence of a stationary target with and without side target distance preview

4.2 Decision-making

The decision-making for autonomous lane change maneuver is the result of two main tasks:

1. Check the feasibility of the maneuver, that is determine if it will cause a collision or violation of the safe distance with other actors in the surrounding.

2. Evaluate the benefit of performing the lane change.

Instead of treating them singularly, it is proposed an approach that allows to perform the decision-making by reusing the already designed longitudinal motion controller. Thus, a duplicate MPC controller is exploited for checking the constraint violation and, at the same time, for evaluating the cost associated with the maneuver; this result is obtained by merging the previews of the target distances related to the current lane and to the destination lane: once again, it is mandatory to have an estimate of the future time instant at which the crossing will occur and this can be easily done by having planned a trajectory.
In order to let the controller to react to the presence of vehicles behind the ego one, it is first extended the output of the prediction model:

$$\frac{d}{dt}\begin{bmatrix} s_{ego} \\ v_{ego} \\ a_{ego} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} s_{ego} \\ v_{ego} \\ a_{ego} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} a_{dem} \tag{4.7}$$

$$\begin{bmatrix} v_{ego} \\ d_{front} \\ d_{rear} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_{ego} \\ v_{ego} \\ a_{ego} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S_{front} \\ S_{rear} \end{bmatrix} \tag{4.8}$$

where $S_{front} \triangleq d_{front,0} + s_{front}$ and $S_{rear} \triangleq d_{rear,0} - s_{rear}$. Then the constraints on the front and rear safe distances became:

$$\begin{cases} d_{front} \geq v_{ego} \cdot T_{gap} \\ d_{rear} \geq v_{ego} \cdot T_{gap} \end{cases} \iff \begin{bmatrix} T_{gap} & -1 & 0 \\ T_{gap} & 0 & -1 \end{bmatrix} \begin{bmatrix} v_{ego} \\ d_{front} \\ d_{rear} \end{bmatrix} \leq 0$$

It is worth noting that this time there is no need to handle situations of constraints violation by means of slack variables: if the controller is not able to compute a feasible control sequence then the maneuver must be prevented and declared unfeasible; hence, all constraints are defined as ”hard”. In addition, it may be appropriate to strengthen some of them, such that too aggressive maneuvers are evaluated as unfeasible: more in details, the lower bound of the longitudinal jerk is increased, such that the controller is prevented to compute control moves that cause abrupt deceleration.

Implementation

At each time step the secondary MPC longitudinal controller is initialized with the same state as the primary one, except for the target distance preview; this one is computed as the vehicle would start a lane change maneuver:

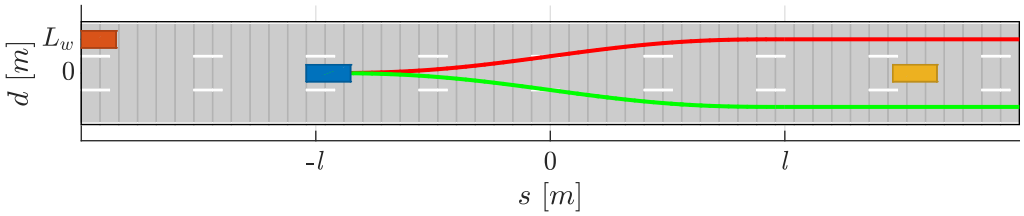
$$[S_{front}(k|k) \ ... \ S_{front}(k+\alpha_k|k)] = [S_{f,current}(k|k) \ ... \ S_{f,current}(k+\alpha_k|k)]$$

$$[S_{front}(k+\alpha_k+1|k) \ ... \ S_{front}(k+p|k)] = [S_{f,side}(k+\alpha_k+1|k) \ ... \ S_{f,side}(k+p|k)]$$

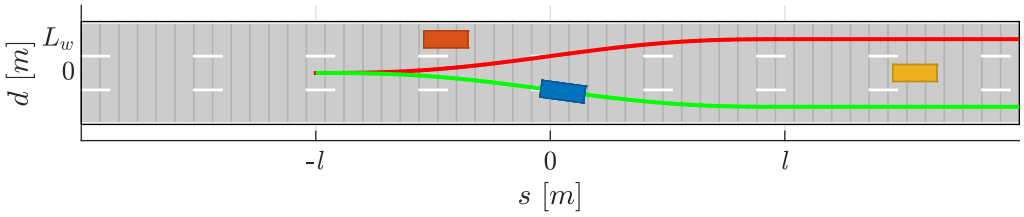
$$[S_{rear}(k|k) \ ... \ S_{rear}(k+\alpha_k|k)] = [S_{r,current}(k|k) \ ... \ S_{r,current}(k+\alpha_k|k)]$$

$$[S_{rear}(k+\alpha_k+1|k) \ ... \ S_{rear}(k+p|k)] = [S_{r,side}(k+\alpha_k+1|k) \ ... \ S_{r,side}(k+p|k)]$$

where again α_k is the future time instant at which the vehicle will cross the line.
The computed optimal acceleration demand is discarded, whereas the cost preview is extracted and used as an indicator of the cost associated with the lane change; it is worth noting that, if the controller is not able to solve the optimization problem due to constraints violation, then the cost is set to infinite, thus preventing the starting of the maneuver; on the other hand, if it is finite, it is compared to the cost preview computed by the primary controller, representing the cost associated to keep the vehicle on the current lane. To include the cost associated with the maneuver itself, the computed cost preview is multiplied by a factor $\gamma > 1$. Overall, if the lane change cost is lower than the current lane cost, then a lane change is requested to the lateral controller.
In order to include the possibility to have both the left and right lane in the decision making there is no need to solve a third optimization problem: the same secondary controller is initialized alternatively at each time step with the distance preview of respectively the left and right targets; then, the computed cost previews are collected together with the related side lane direction. Despite the resulting sample time at which each lane cost is evaluated is doubled, the computational load remains unchanged.



(a) @$t = t_0$



(b) @$t = t_{cross}$

Figure 4.6: Unfeasible left lane change (red); feasible and convenient right lane change (green)

Chapter 5

Offline simulation

The designed Highway Assist System has been implemented in the Matlab/Simulink environment exploiting the *Model Predictive Control Toolbox* [12].
The first closed-loop testing phase has been based on a simplified simulation model, composed by:

- *Vehicle Dynamics*: single-track dynamic bicycle model [13]; the inputs are the longitudinal acceleration and the front wheel steering angle; the outputs are the vehicle pose and dynamics signals associated with the vehicle motion (i.e. pose, velocity, acceleration).
- *Scenario & Sensors*: this block generates the synthetic data for simulating the sensor output signals [1] with the possibility of switching between different scenarios; the latter are designed using the *Automated Driving Toolbox* [14]. The output is a bus composed by: RADARs signals, that are distance, speed and acceleration of the front and rear targets in the current, left and right lanes respectively; camera signals, that are road curvature preview, lateral position and yaw errors, lane width and lane markers topology (solid or dashed).
- *Driver Signals*: Highway Assist System enable button, reference speed, time-gap setting.
- *HAS*: contains the lateral and longitudinal motion controllers and the Autonomous Lane Change logic.
- *ECM - BSM* and *EPS*: represents respectively the Engine Control Module, Braking System Module and Electronic Power Steering, that are the ECUs in charge of tracking the reference signals produced by the HAS; these systems are approximated by a PI controller.

Figure 5.1 shows the overall closed-loop simulation model; it is worth nothing that different time domains are highlighted with different colors: black for continuous-time, red for discrete-time and yellow for the mixed time domain, that are the sensing and actuation systems.

[1] Unless otherwise stated, assumed as ideal, i.e. not affected by measurement noise.

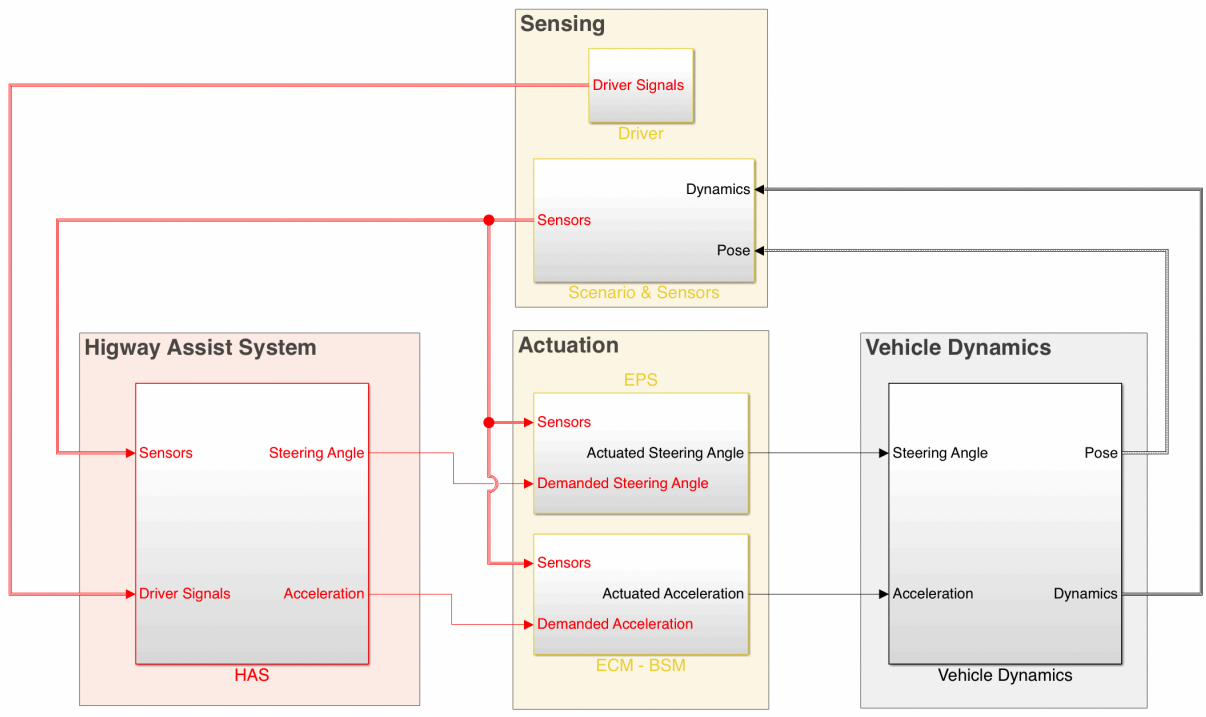


Figure 5.1: Simulink virtual test bench for closed-loop simulation

5.1 Longitudinal Controller

In this section are presented the simulation results relative to the functionalities implemented by the longitudinal controller, namely the Cruise Control, Adaptive Cruise Control (with Stop&Go), and Intelligent Adaptive Cruise Control.

Cruise Control

This feature expect the vehicle to be capable to track a reference speed set by the driver, while respecting comfort constraints on acceleration and jerk.
Figure 5.2 shows the simulation results of a Cruise Control maneuver; the scenario comprises a free straight road, in which the vehicle travels at the reference speed equal to $v_{ref} = 90\ km/h$; after 5 seconds the driver increases the reference speed to 130 km/h; after 25 seconds the driver set back the reference to 90 km/h. It can be seen that the vehicle speed tracks the reference while the computed acceleration demand complies with both acceleration and jerk constraints.

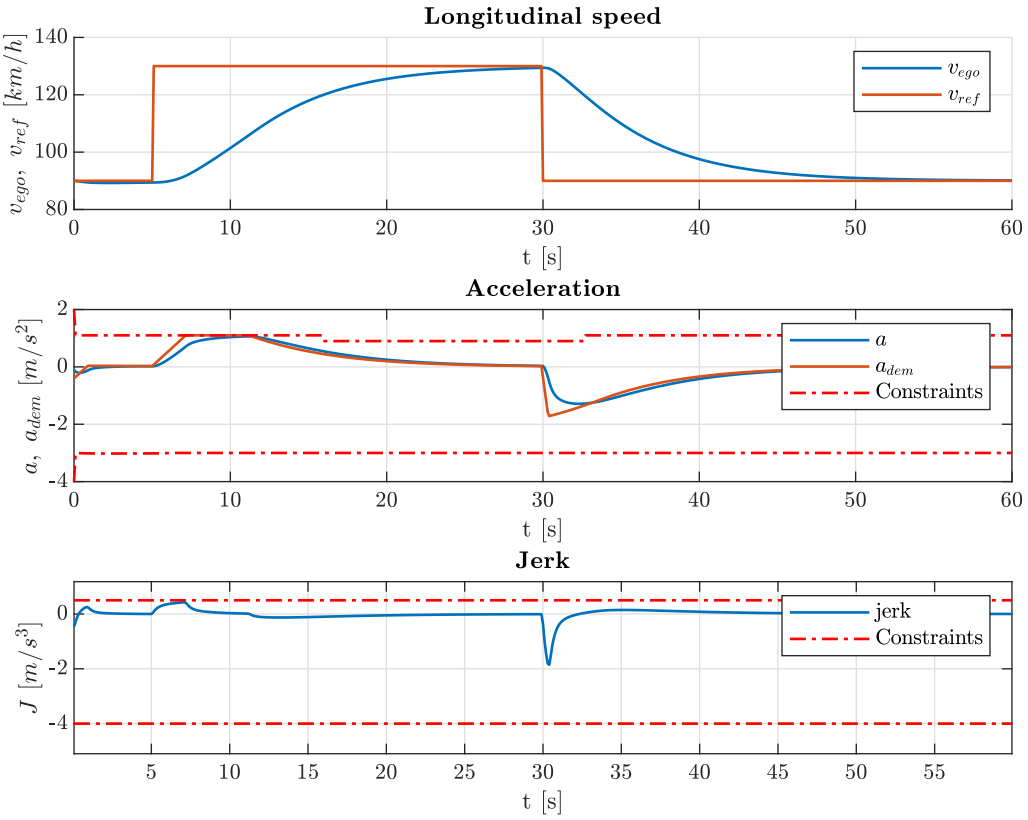


Figure 5.2: Longitudinal controller response to reference speed variations

It is possible to tune properly the weights W_v related to speed tracking error $v_{ref} - v_{ego}$ depending on the desired driving style: increasing W_v leads to a more aggressive response, whereas decreasing this parameter the behavior will be smoother (Figure 5.3).

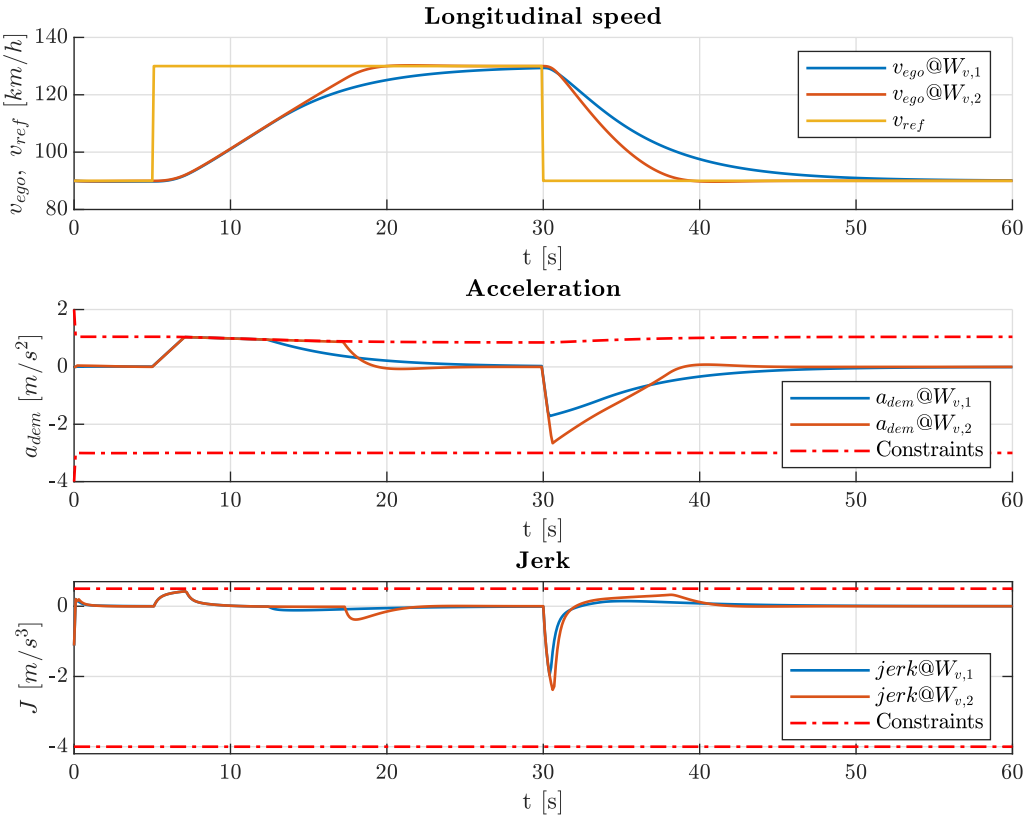


Figure 5.3: Comparison of longitudinal controller response with speed tracking error tuning weight $W_{v,1} < W_{v,2}$

Adaptive Cruise Control

The vehicle must be capable to decrease its speed in order to keep a safe distance from the leading target.

Figure 5.4 shows the simulation results for the Adaptive Cruise Control functionality; the scenario consists of a straight road occupied by an actor whose velocity profile v_{target} is shown in the upper plot: it starts with constant speed at 90 km/h, then it brakes at $-2\ m/s^2$ until it stops and accelerates again at $2\ m/s^2$ up to the initial speed. The ego vehicle starts at $v_{ego} = 130\ km/h$ and the reference speed is kept at the same value for the complete duration of the maneuver; it is possible to observe that the vehicle is able to adapt its speed in order to never exceed the safe distance; it is also able to perform a ”Stop&Go”, that is a full-stop and restart as the front actor accelerate again.

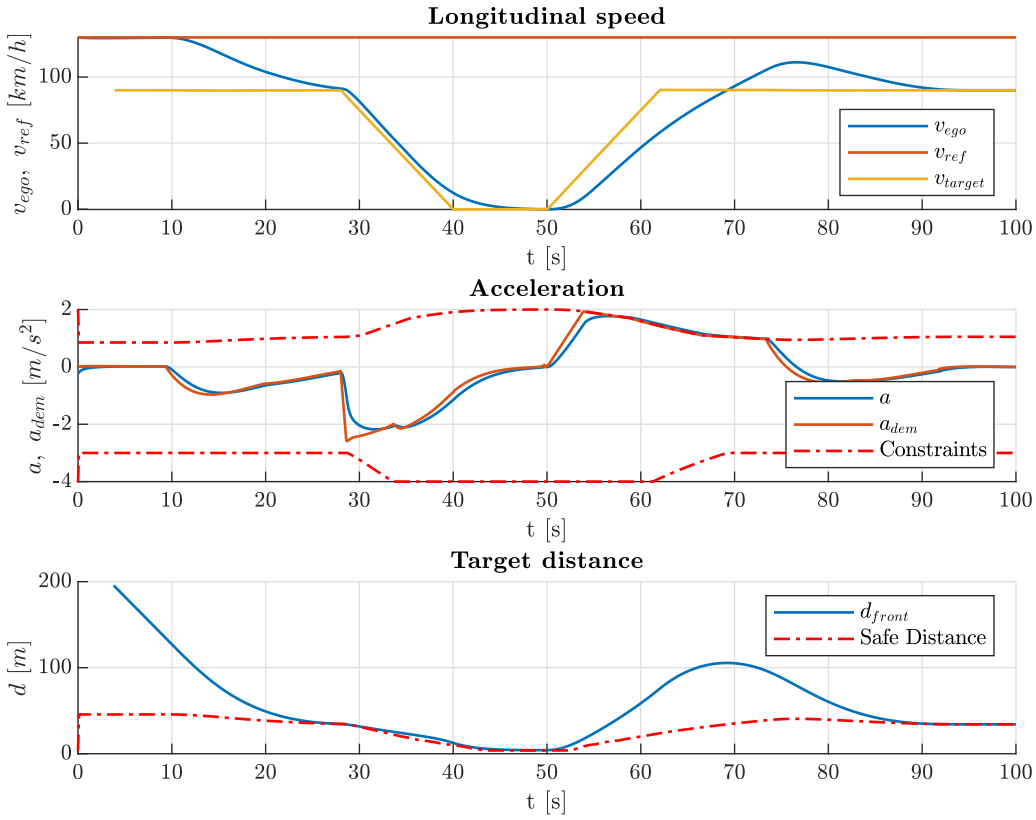


Figure 5.4: Adaptive Cruise Control with Stop&Go

As mentioned in Section 3.2, the MPC framework allows to handle situations of constraint violation by means of *constraint softening*, that is, if there no exist a solution for the optimization problem compliant with the constraints, a slack variable ε is used for softening the defined constraints. Figure 5.5 shows the simulation results of a cut-in scenario: the ego vehicle is traveling on the left lane of a two-lane road and it is accelerating to reach the reference speed; suddenly a slower vehicle coming from the right performs a left lane change, obstructing the ego vehicle path; after a while, it turn back to the right lane. Despite there no exist any control action that allows to brake the vehicle while respecting the comfort and safety constraints on the acceleration and safe distance respectively, the controller is still able to avoid the collision by finding a trade-off in softening these constraints.

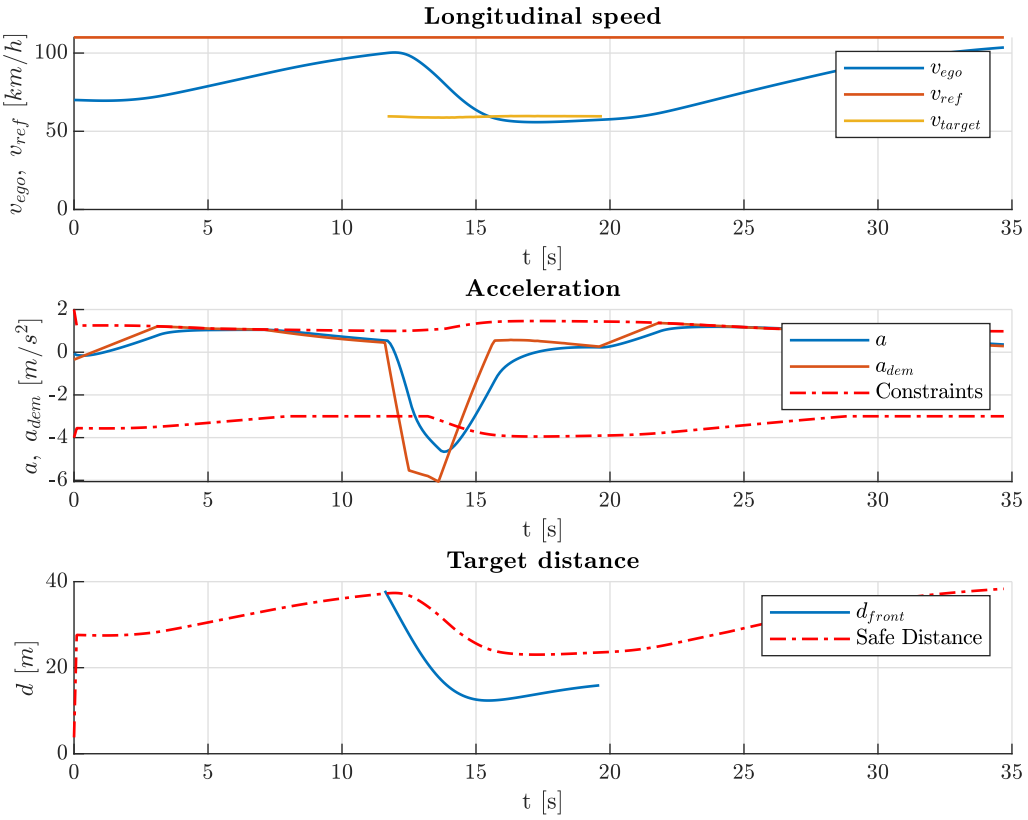


Figure 5.5: Cut-in scenario: the MPC controller is able to avoid the collision by softening the constraints

Intelligent Adaptive Cruise Control

The Intelligent Adaptive Cruise Control (iACC) functionality consists in adapting the ego vehicle speed when approaching a curve, such that the lateral acceleration is always kept within specified comfort bounds and thus allowing to perform the curve safely and comfortably. It must be said that in highway scenarios the presence of curves with a relatively low radius of curvature is not common, however, to test the designed functionality, it has been chosen a scenario characterized by a sequence of curves with high curvature; in addition, to stress even more the system, the reference speed has been set to value much higher than a plausible speed limit for this road morphology. Figure 5.6 shows the simulation results: from the upper plot, it can be observed how the previewing mechanism allows the controller to react in advance to the presence of a curve by starting to brake in time, thus never exceeding the maximum allowed speed.

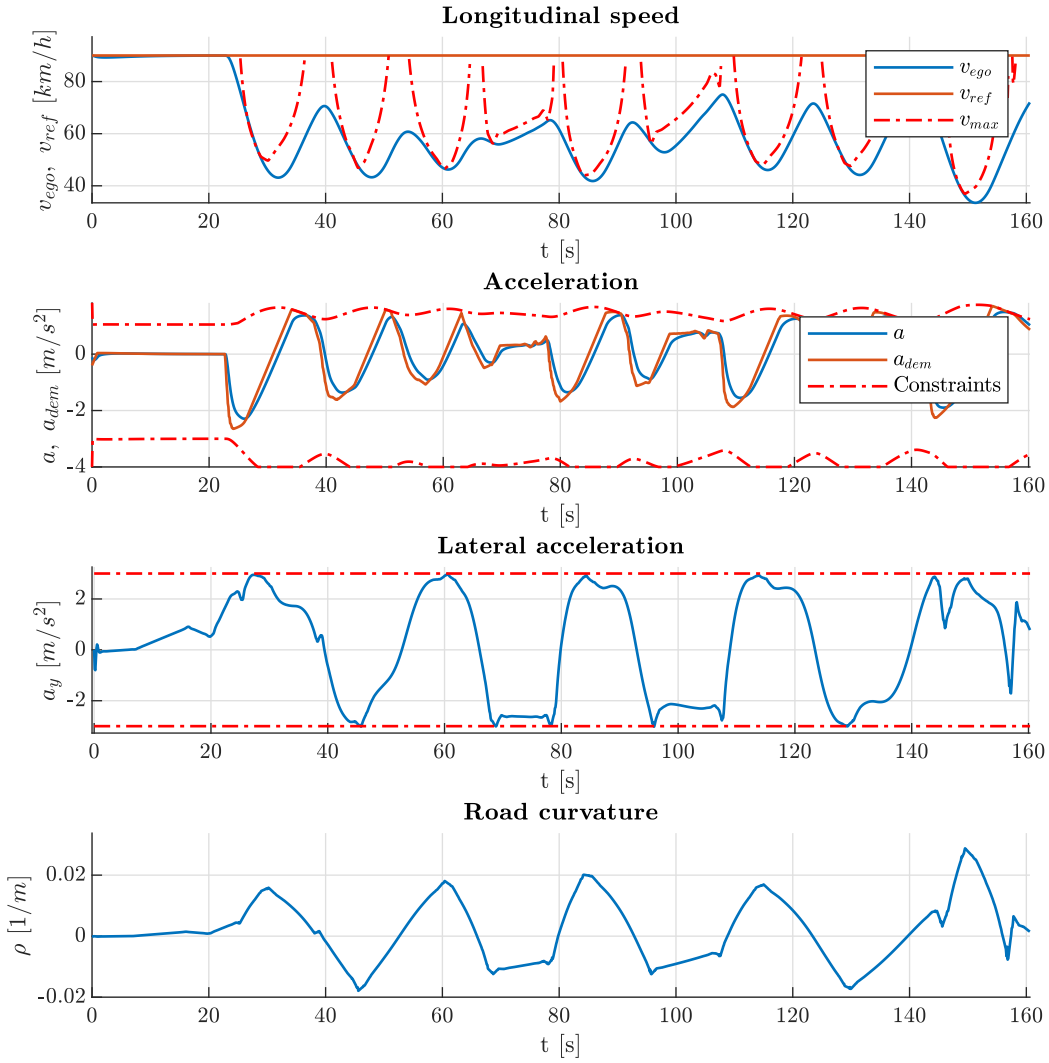


Figure 5.6: Intelligent Adaptive Cruise Control on a curved road scenario

As discussed in Chapter 2, the designed Highway Assist System is an L2+ ADAS since it exploits high-definition maps other than the front camera to sense the environment. This feature can be exploited to ”augment” the camera range (60 meters), thus feeding the longitudinal controller with a full preview of the road curvature. **??** shows the simulation results: it can be seen how the controller is able to react with more advance, ensuring smoother braking and acceleration.

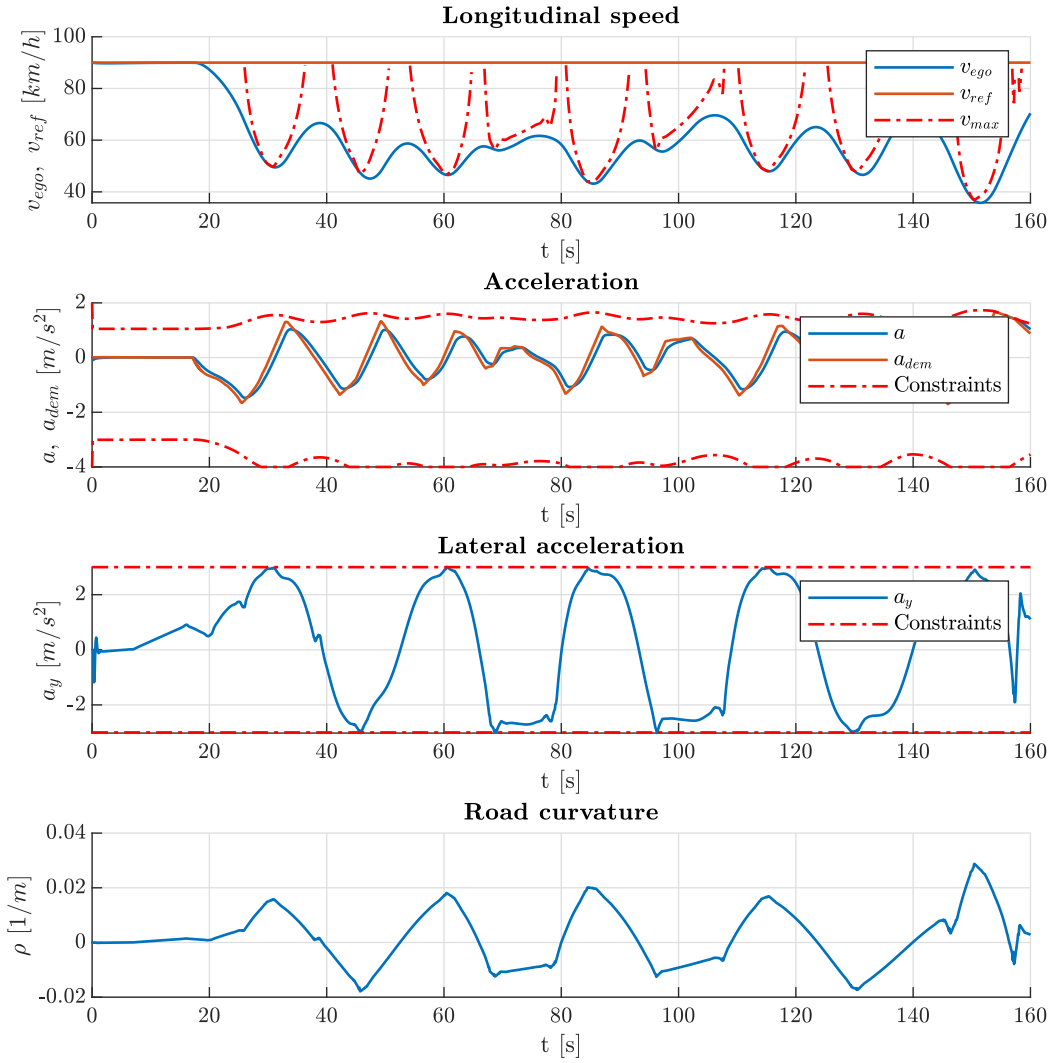


Figure 5.7: Intelligent Adaptive Cruise Control with augmented road curvature preview

5.2 Lateral Controller

The main functionality implemented by the lateral controller is Lane Centering. In addition, it must be capable of performing a lane change according to the planned trajectory.

Lane centering

The Lane Centering functionality has the main objective of minimizing the vehicle lateral position error with respect to the centerline by controlling the steering angle. The test scenario consists of a curved road with decreasing radius of curvature (Figure 5.8); more in detail, the road profile is described by a clothoid model with curvature derivative $\dot{\rho} = 10^{-5}$ $[m^{-2}]$ and 1

km length. This scenario allows to test the response of the lateral motion controller when dealing with a variable curvature.
Figure 5.9 shows the simulation results when the vehicle travels at a constant speed of 110 km/h; it can be seen that despite the lateral position and yaw errors increase as the radius of curvature decreases, they keep small for the entire duration of the maneuver, reaching a maximum of about 5 centimeters and 0.5 degrees respectively. Moreover, it must be said that this test has been performed with the iACC functionality disengaged, that is the speed is not constrained: the bottom right plot shows how the vehicle experiences a very high lateral acceleration as the curvature radius gets smaller.

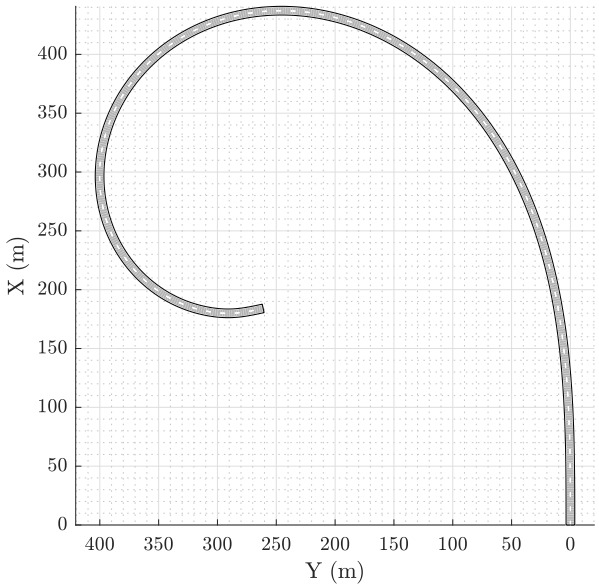


Figure 5.8: Clothoid road scenario

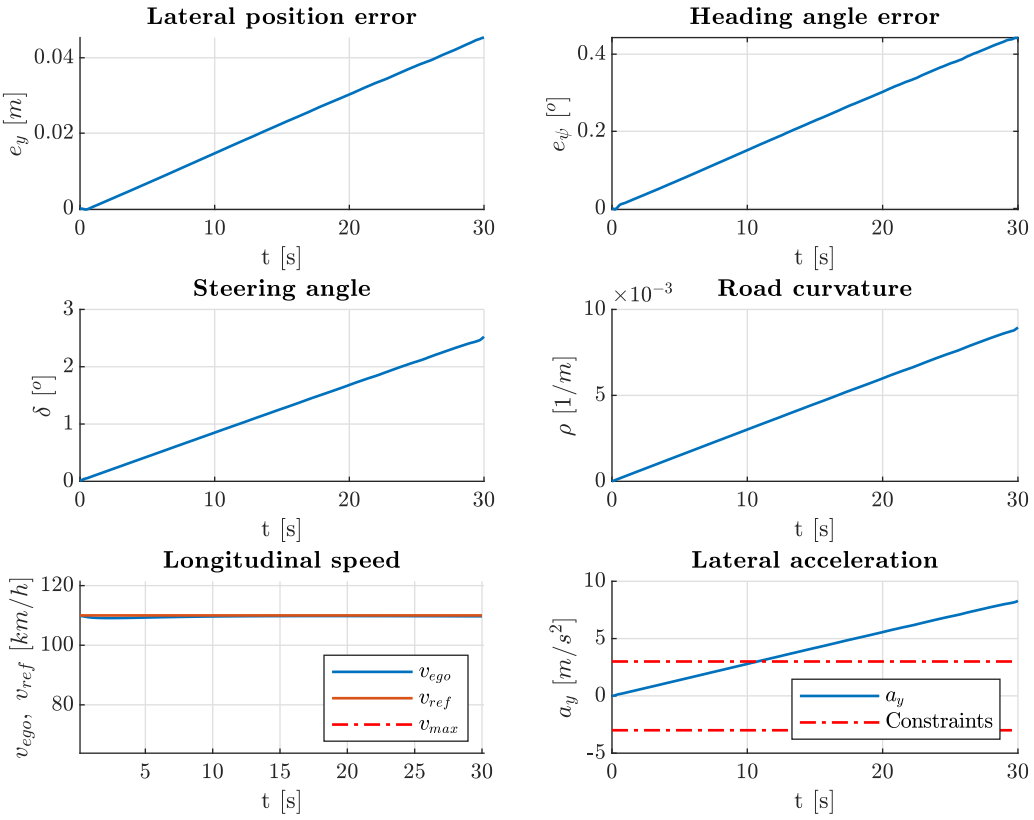


Figure 5.9: Simulation results with iACC disengaged

More realistic results are obtained by engaging the iACC, thus limiting the vehicle speed, as well as the lateral acceleration. In this case, the lateral position error is even lower, reaching a maximum of less than 2 centimeters; similarly, the maximum yaw error is about halved. Hence, by limiting lateral acceleration, it not only ensures greater comfort, but also improves the performance of the lateral controller

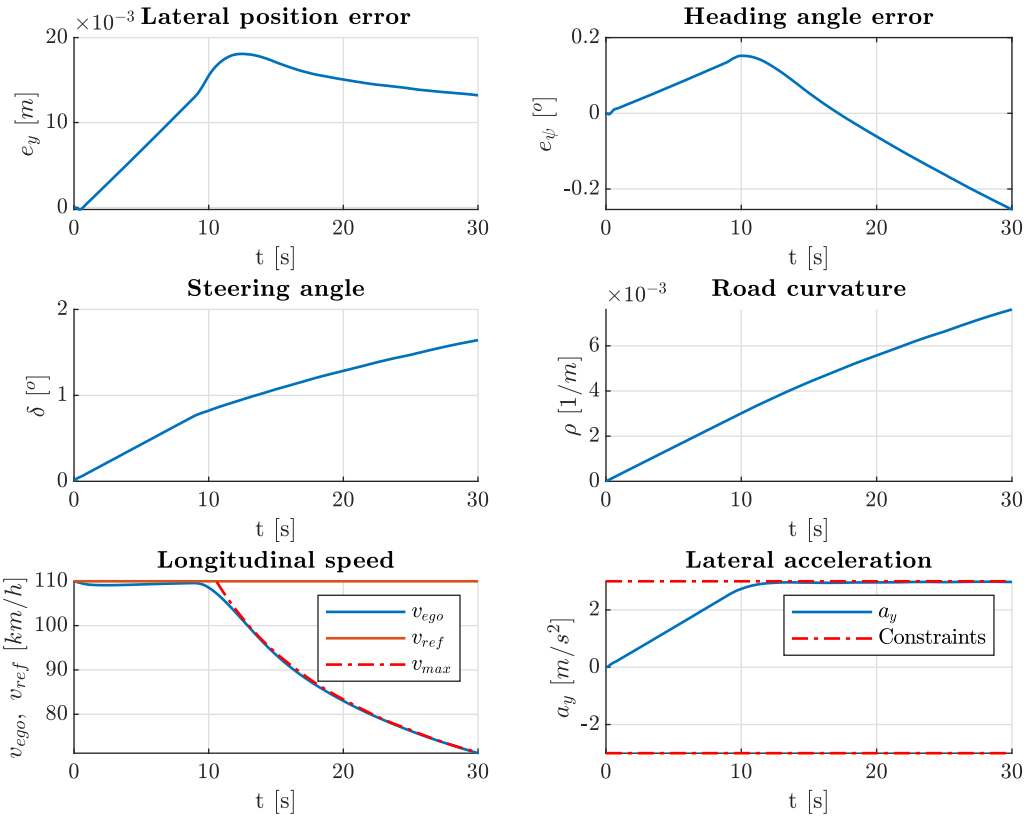


Figure 5.10: Simulation results with iACC engaged

As discussed in Chapter 4, the Autonomous Lane Change functionality is decoupled in two tasks:

1. planning a feasible and comfortable trajectory;
2. evaluating the feasibility and benefit of changing lane.

Lane change trajectory following

Figures 5.11 to 5.13 show the simulation results for a left lane change performed on a straight road scenario, at three different values of maximum allowed lateral speed v_y^r, while the vehicle is traveling at $v_{ego} = 100\ km/h$. Table 5.1 resumes the computed maneuver lengths for each test case; it can be seen that the vehicle is able to perform successfully the lane change in all the three test cases, according to the planned trajectory.

	$v^r_{y,max}\ [m/s]$	$a^r_{y,max}\ [m/s^2]$	$j^r_{y,max}\ [m/s^3]$	$2 \cdot l\ [m]$ @110 $[km/h]$
Case 1	1.00	0.46	0.70	206.25
Case 2	1.50	1.03	2.37	137.50
Case 3	2.00	1.82	5.62	103.13

Table 5.1: Constraints on lateral dynamics used for computing the length of the maneuver.

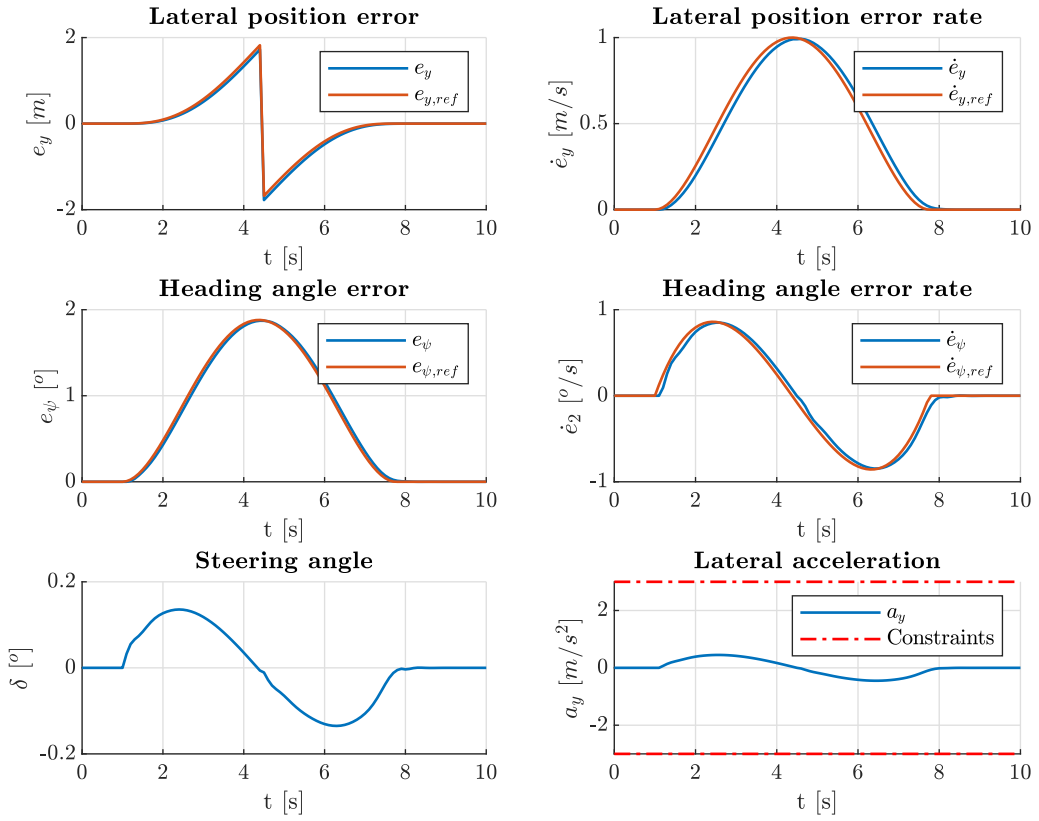


Figure 5.11: Left lane change with maximum relative lateral speed $v^r_{y,max} = 1\ [m/s]$

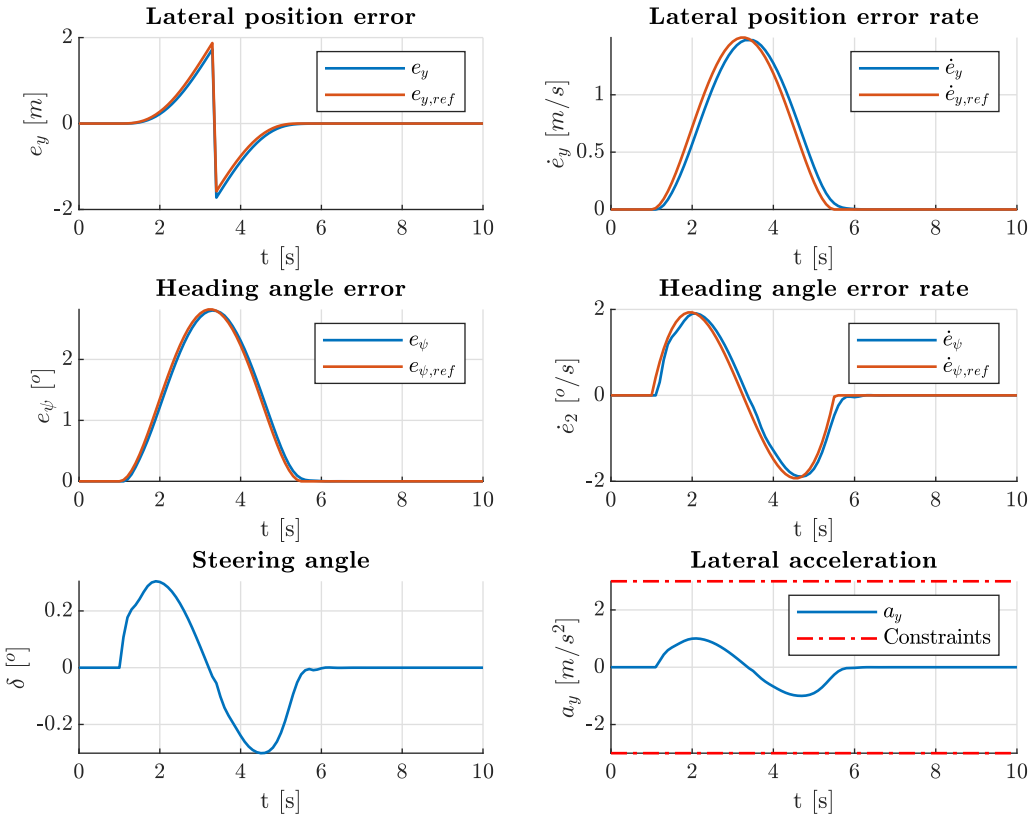


Figure 5.12: Left lane change with maximum relative lateral speed $v^r_{y,max} = 1.5\ [m/s]$.

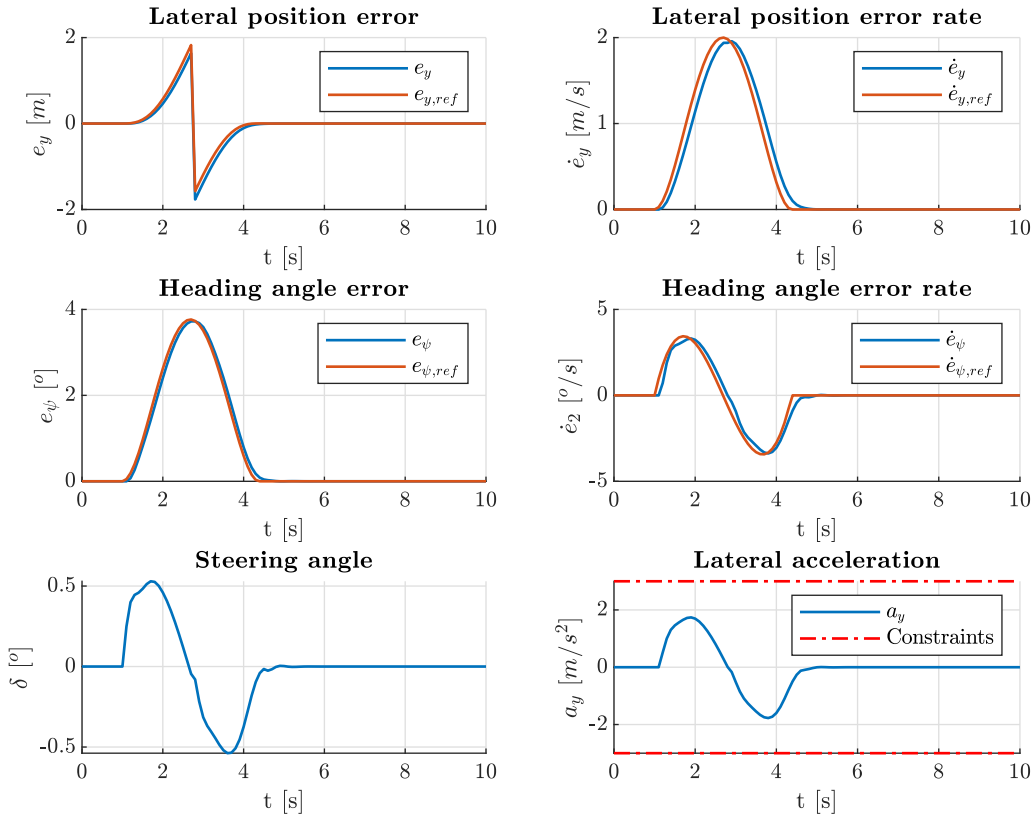


Figure 5.13: Left lane change with maximum relative lateral speed $v^r_{y,max} = 2\ [m/s]$

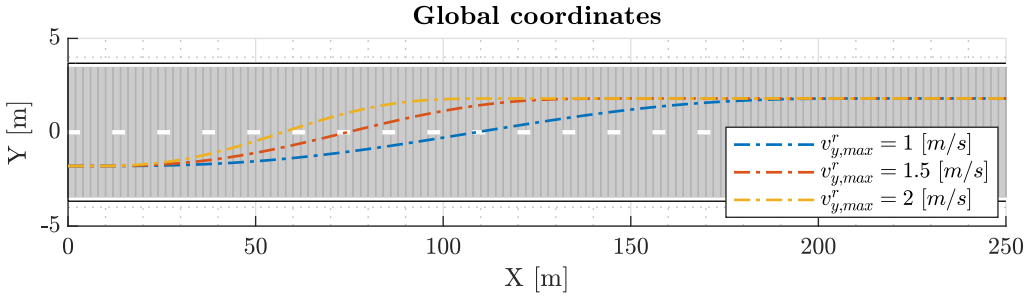


Figure 5.14: Comparison between left lane changes of different length at $v_{ego} = 100\ km/h$

5.3 Lane Change Decision-Maker

In Section 4.2 has been proposed an MPC-based approach for the lane change decision-making, encompassing both the feasibility and convenience analysis. Figure 5.15 shows the scenario exploited to test the decision-making algorithm for a left lane change: the ego vehicle starts in the right lane with initial and reference speed of respectively 110 km/h and 130 km/h; it is preceded by an actor that is traveling at 108 km/h (constant). The left lane is occupied by a

second actor with speed of 90 km/h. The third plot of Figure 5.16 shows the computed cost previews for the current, the left and the right lane: as expected, as the ego gets closer to the front right target, the cost for a right lane change C_{right} increases and became infinite as soon as the maneuver would lead to the front safe distance violation ($t \simeq 5\ s$). At $t = 15\ s$ the ego vehicle overtakes the actor on the right lane and at $t = 18.5\ s$ the cost back to being finite, despite the rear right target distance is below the safe value (second plot): this results from the preview that, if the the ego vehicle starts the maneuver at this time instant, at the moment of line crossing the distance to the rear target will comply the safe value. Indeed, it can be observed that, since the cost preview $C_{right} < C_{current}$, after 0.5 s (tunable threshold) a right lane change request is sent to the lateral controller (fourth plot) and, as soon as the vehicle crosses the line ($t = 22.3\ s$), the safe distance is respected. In other words, by looking at the second plot, the blue line represents the distances of the front (if positive) and rear (if negative) targets: at any time instant it is always outside the area enclosed by the front and rear safe distances (red dashed line).

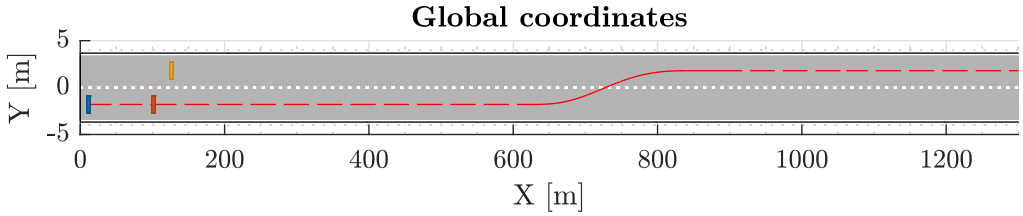


Figure 5.15: Test scenario #1: the red dashed line is the resulting path followed by the ego vehicle (blue); the orange and yellow patches represent the other actors in their initial positions.

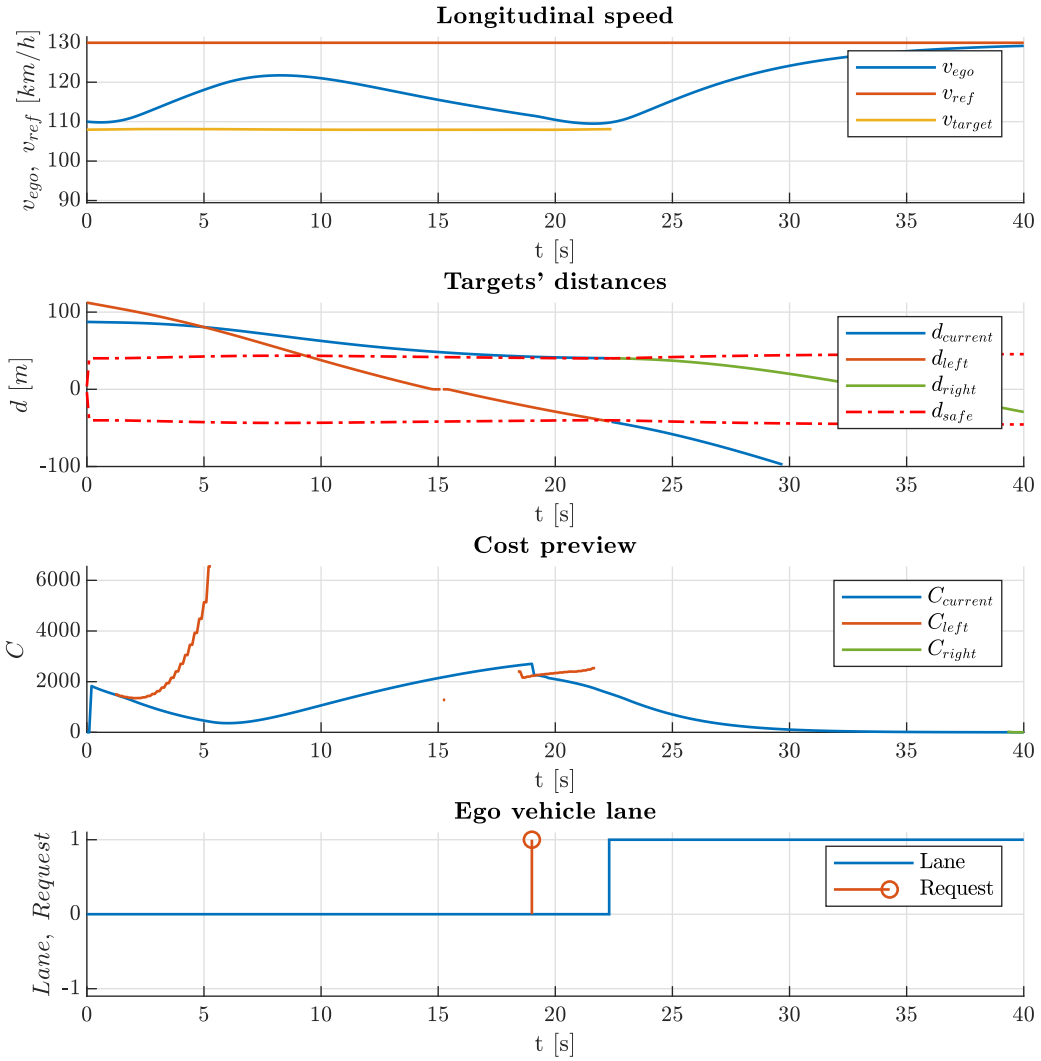


Figure 5.16: Simulation results for Autonomous Lane Change when a slower target occupies the destination lane.

A second scenario is shown in Figure 5.17: this time the slowest actor (yellow) travels on the right lane and, as before, it prevents the ego vehicle to reach the desired speed. In addition, the left lane is occupied by a second actor (orange) that is moving at higher speed than the yellow one. Figure 5.18 shows how, as soon as the orange actor is next to overcame the yellow one, the decision-maker rises a left lane change request to the lateral controller ($t = 18.1\ s$). As before, thanks to the previewing technique, the maneuver starts when the left target is still violating the safe distance, but, as soon as the vehicle crosses the line ($t = 21.3\ s$), the constraint is respected (second plot). Now, since the reference speed is even higher than the orange actor speed, as soon as the rear distance to the yellow actor is large enough, the decision-maker requests a right lane change ($t = 35.2\ s$).

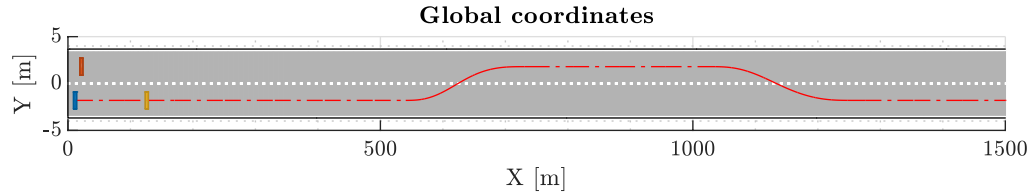


Figure 5.17: Test scenario #2: the red dashed line is the resulting path followed by the ego vehicle (blue); the orange and yellow patches represent the other actors in their initial positions.

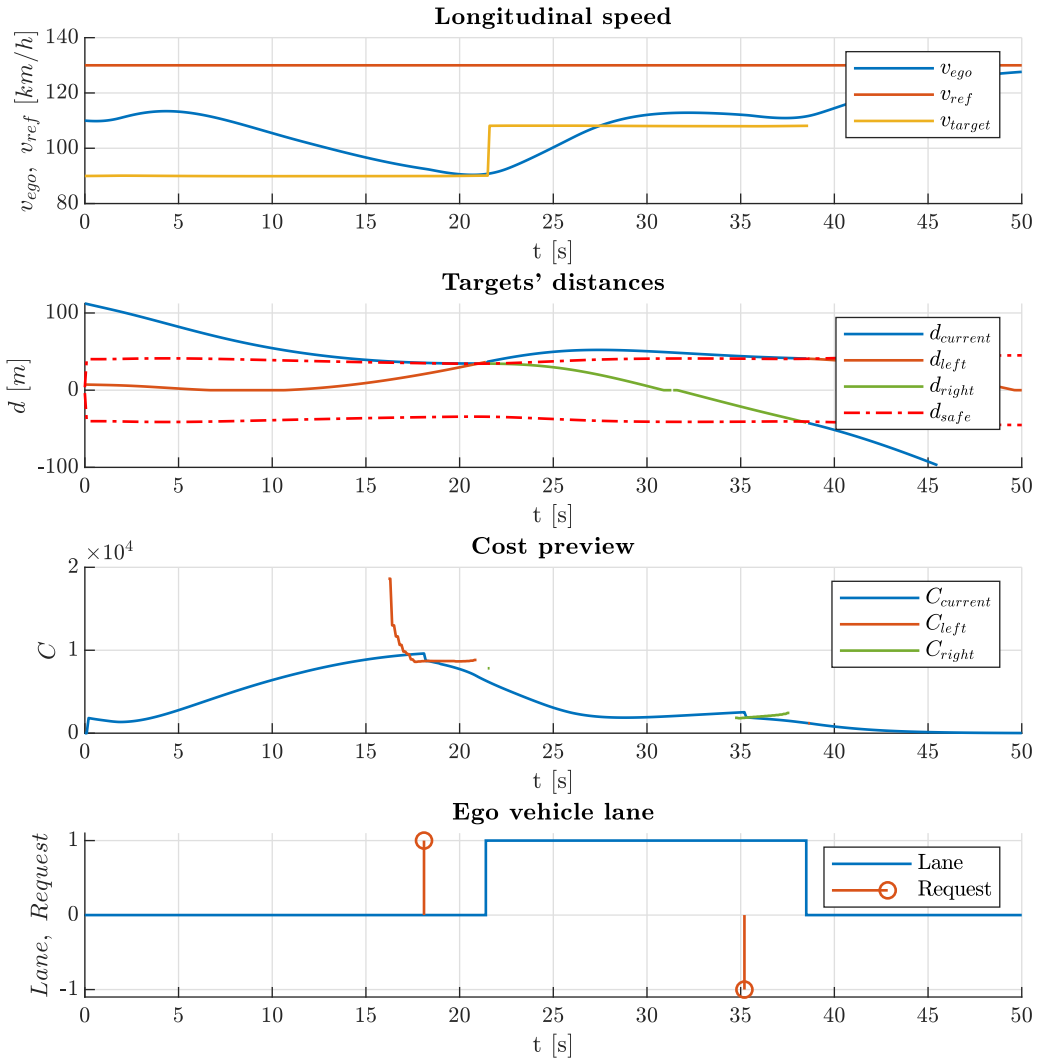


Figure 5.18: Simulation results for Autonomous Lane Change when a faster target occupies the destination lane.

To understand how the designed functionality behaves in a more chaotic scenario, it has been tested in a four lanes highway traffic scenario, populated by 17 actors traveling at speeds in the

range [90; 110] km/h, while the ego vehicle reference speed is set to 130 km/h. In addition, a delay of 4 seconds has been included before starting each lane change, representing the time interval in which the turn indicator is switched on to declare the maneuver to the other actors. The results showed that the vehicle was able to complete the maneuver with an average speed of about 117 km/h and to perform 8 lane change maneuvers while never violating the front and rear safe distance constraints (Figure 5.19).

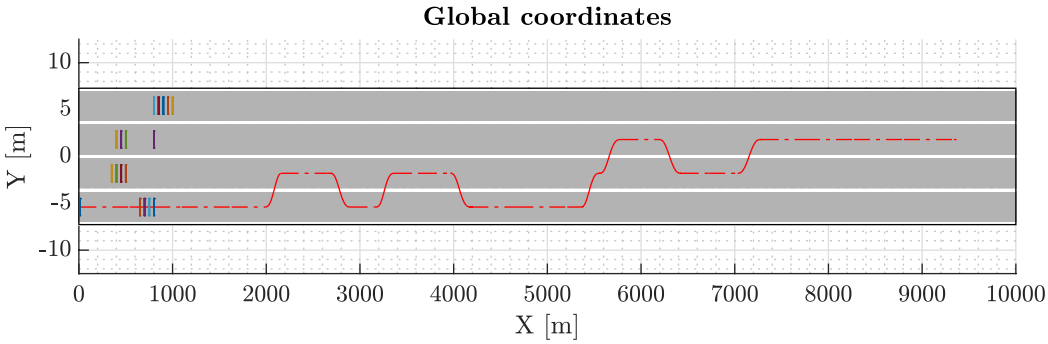


Figure 5.19: Four lanes highway traffic scenario

It must be said that the final number of lane changes showed to be relatively high for a travel of less than five minutes, resulting in a too aggressive behavior for a traffic scenario. On the other hand, the algorithm is behaving correctly and the frequent lane changes are caused by the driver setting a reference speed much higher than the other actors. Nevertheless, it is possible to properly increase the cost associated to the lane change maneuver to reduce the number of triggers and thus promote the driving comfort.

Chapter 6

Real-time simulation

The core of RCP methodology is the real-time implementation of the designed control system. For this purpose, Maserati provided a dSPACE real-time simulator dedicated to the hardware-in-the-loop (HIL) validation of the brand new *Maserati Grecale* ADAS control unit.

6.1 Real-time simulator

The simulator exploited for the real-time testing of the designed Highway Assist System is a dSPACE customized SCALEXIO system [15]; the key features are:

- High-performance multi-core processing unit, suitable for real-time simulation of a full vehicle dynamic model.
- Flexible I/O capabilities, including the possibility of *rest-bus* simulation, i.e. the simulation of CAN bus traffic.

dSPACE provides a toolchain for the simulator management composed by:

- *Configuration Desk*: allows to set up the hardware I/O configuration, to define the simulation models and the real-time tasks properties (i.e. period, priority, scheduling algorithm). Moreover, it manages the code-generation and compilation process by directly interfacing with Matlab/Simulink environment.
- *Control Desk*: instrumentation software used for the real-time interaction with the simulator; here it is possible to design control interfaces, by means of which the user can access to and possibly modify the variables of the models, offering plotting and measurement tools.
- *Model Desk*: provides features for online model parameterization; it also include a ”road generator” for the creation of scenarios and maneuvers.
- *Motion Desk*: 3-D online visualization tool; it reads the data directly from the HIL simulator and provides real-time animations (Figure 6.1).

As far as the plant model is concerned, it is used the dSPACE *Automotive Simulation Models (ASM)*, that is a tool suite for simulating the vehicle dynamics, engine, ECUs, traffic environment a sensors, provided as open Simulink models. It is worth noting that this time, the simulation model is highly realistic: just to mention, it includes a 13 DoF drivetrain model with manual and automatic transmission, other than a 13 DoF vehicle multibody system model, consisting of a car body, four wheels and steering system. As mentioned before, the vehicle simulation model is parametrized according to the Maserati Grecale technical specifications.

Overall, once the vehicle model is compiled, it can run as standalone on the real-time simulator. To include the HAS controller, it has been exploited the multi-core capability of the SCALEXIO processing unit, thus avoiding to use an additional real-time hardware. More in detail, the ASM and the HAS models run as two different tasks on two different cores; to allow this, it has been specified a communication interface through which the HAS receives sensors data and sends the demanded acceleration and steering angle to the vehicle.

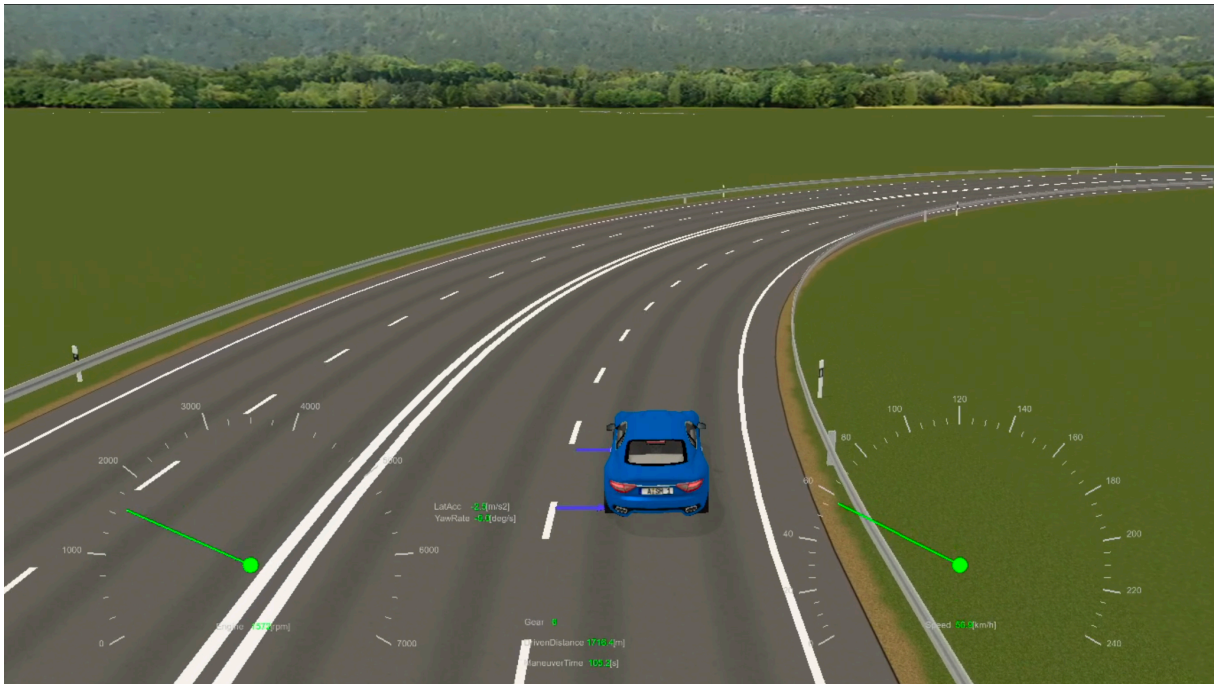


Figure 6.1: Motion Desk capture during a real-time simulation

6.2 Euro NCAP scenarios

The *European New Car Assessment Programme* (Euro NCAP) defines a *Test & Assessment Protocol* dedicated to Highway Assist systems [16] in which are defined several scenarios for evaluating the performance of these functionalities. Part of these test cases have been implemented in the real-time simulation environment and thus exploited to assess the correct functioning of the designed HAS.

Car-to-Car stationary target (CCRS)

According to Euro NCAP definition, the *Car-to-Car stationary target* scenario consists in driving the vehicle-under-test (VUT) toward a stationary target vehicle, while the longitudinal control system (ACC) is active; Figures 6.2 to 6.5 show the simulation results when the VUT is traveling on a straight road at different reference speeds: respectively 70, 90, 110 and 130 km/h. It can be observed that in all the four scenarios the ACC is able to stop the VUT according to the computed safe distance; in addition, except for the last case, the vehicle deceleration never exceeds the lower bound, meaning that the front RADAR range is large enough to allow a smooth deceleration compliant to the comfort bounds of the longitudinal controller; when the initial speed is 130 km/h, despite the lower bound for the VUT acceleration is exceeded, the controller is still able to drive the vehicle to a full-stop within the safe-distance (Figure 6.5).

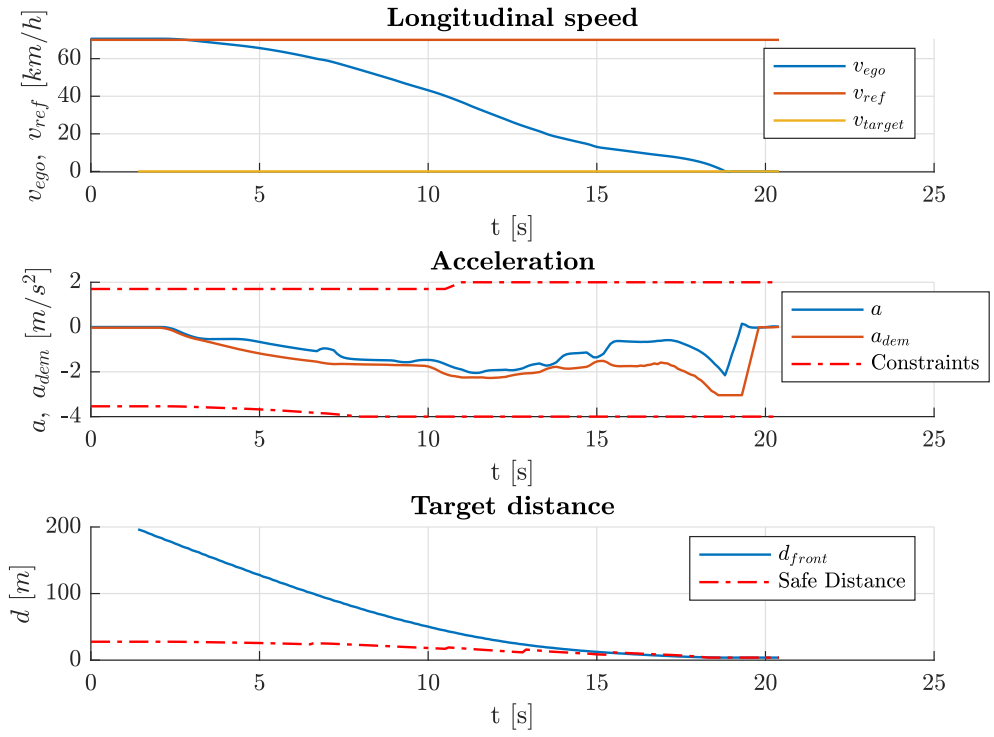


Figure 6.2: CCRS at $v_{ref} = 70\ km/h$

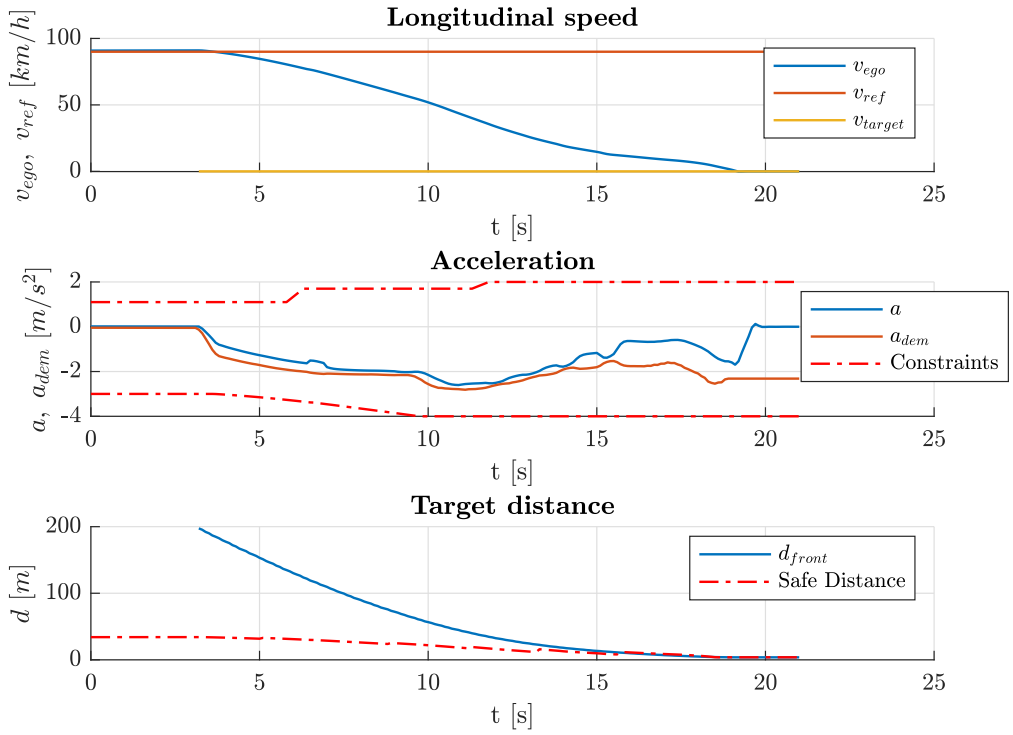


Figure 6.3: CCRS at $v_{ref} = 90\ km/h$

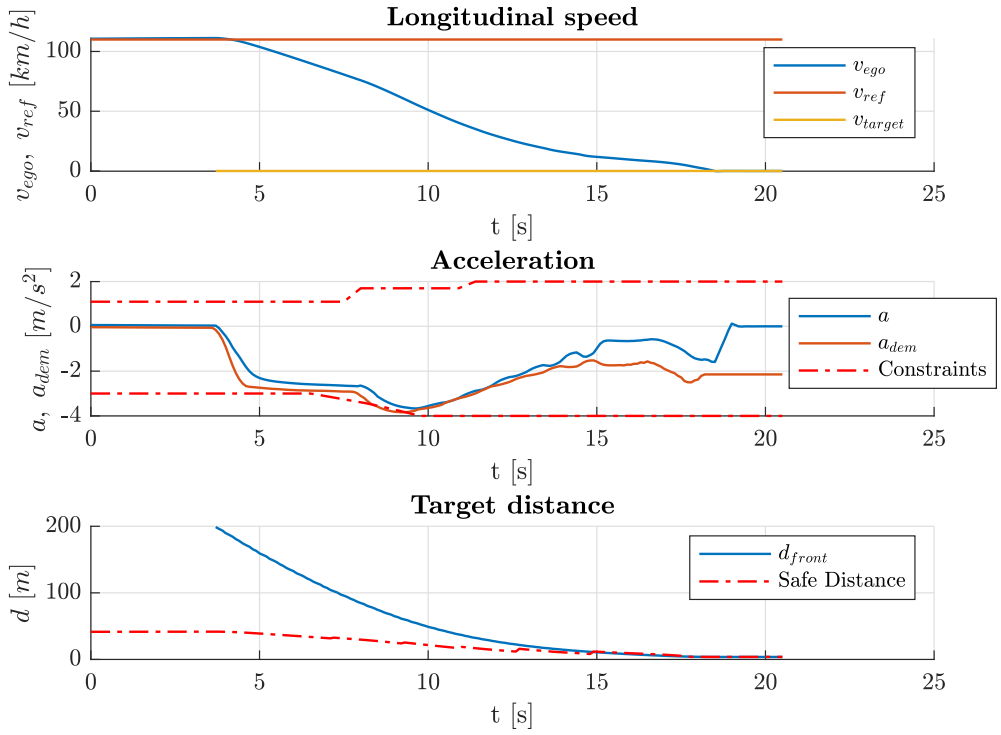


Figure 6.4: CCRS at $v_{ref} = 110\ km/h$

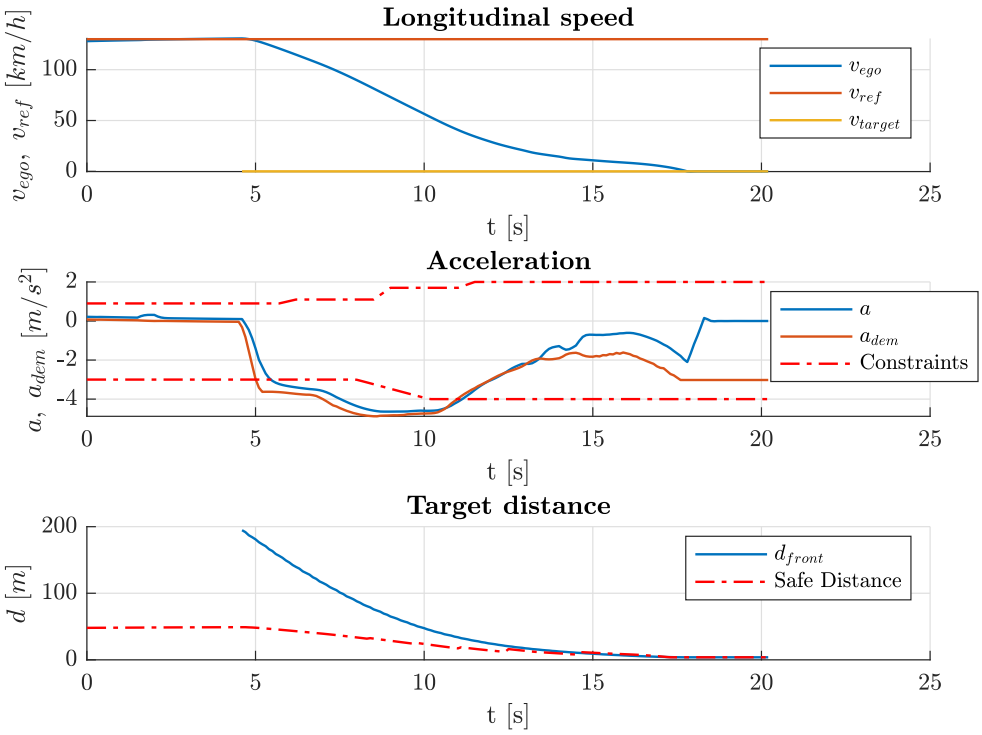


Figure 6.5: CCRS at $v_{ref} = 130\ km/h$

Car-to-Car moving target (CCRM)

In the *Car-to-Car moving target* scenario, the Global Vehicle Target (GVT) is traveling at constant speed slower than the VUT; Figures 6.6, 6.7 and 6.9 and Figures 6.9 to 6.11 show respectively the simulation results for different VUT speeds (90. 110, 130 km/h) when the GVT is traveling at respectively 20 and 60 km/h. In all cases the ACC is is able to adapt the ego vehicle speed to the target speed value while keeping the safe distance.

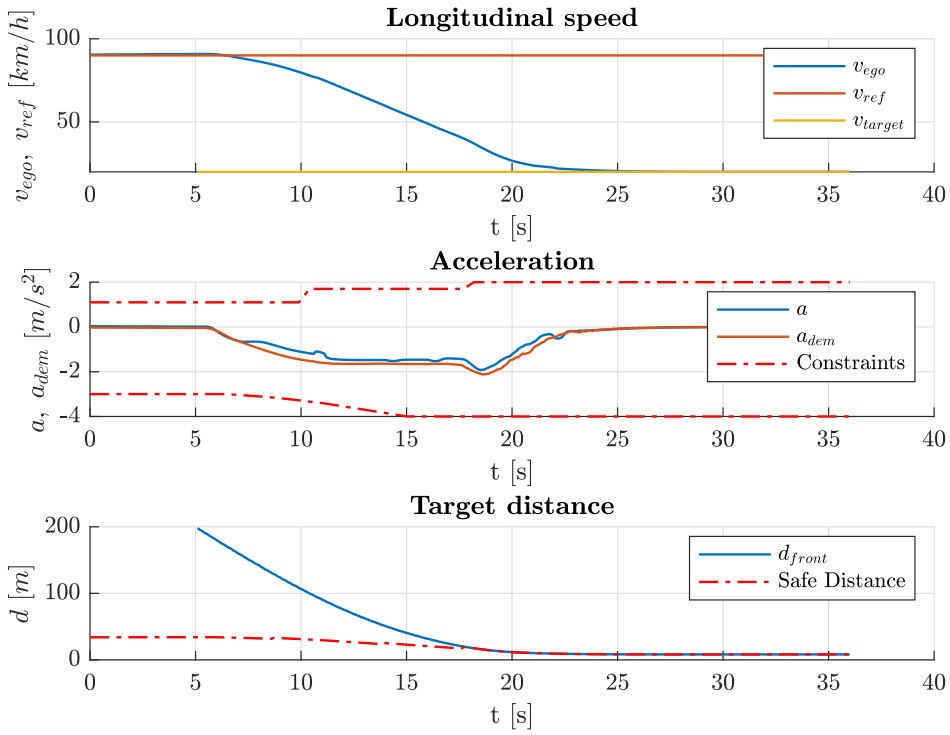


Figure 6.6: CCRM at $v_{ref} = 90\ km/h$ and $v_{target} = 20\ km/h$

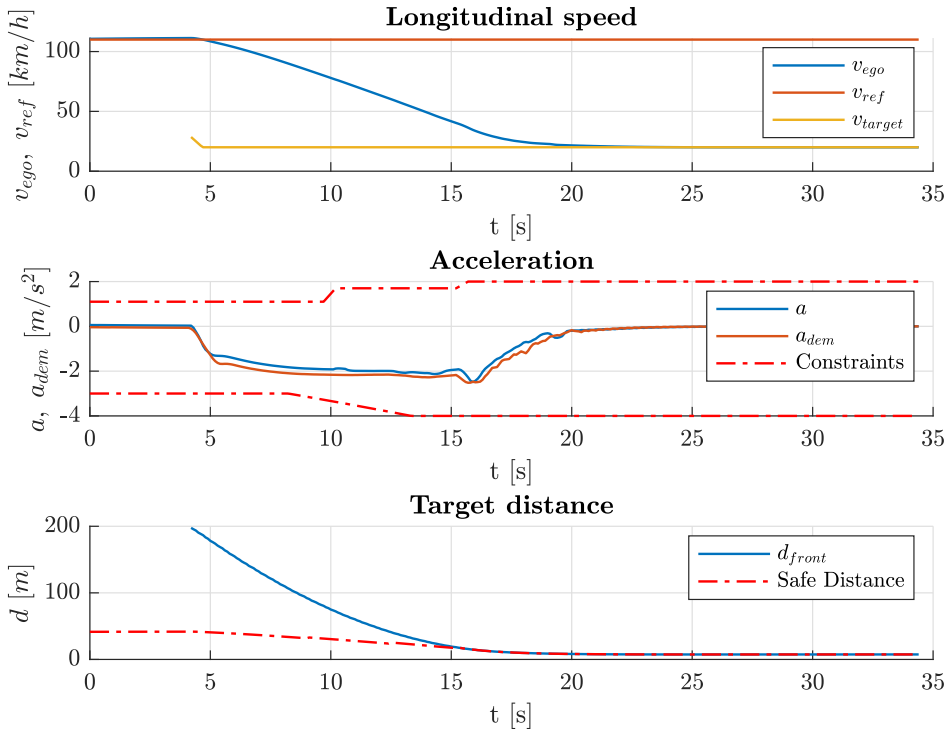


Figure 6.7: CCRM at $v_{ref} = 110\ km/h$ and $v_{target} = 20\ km/h$

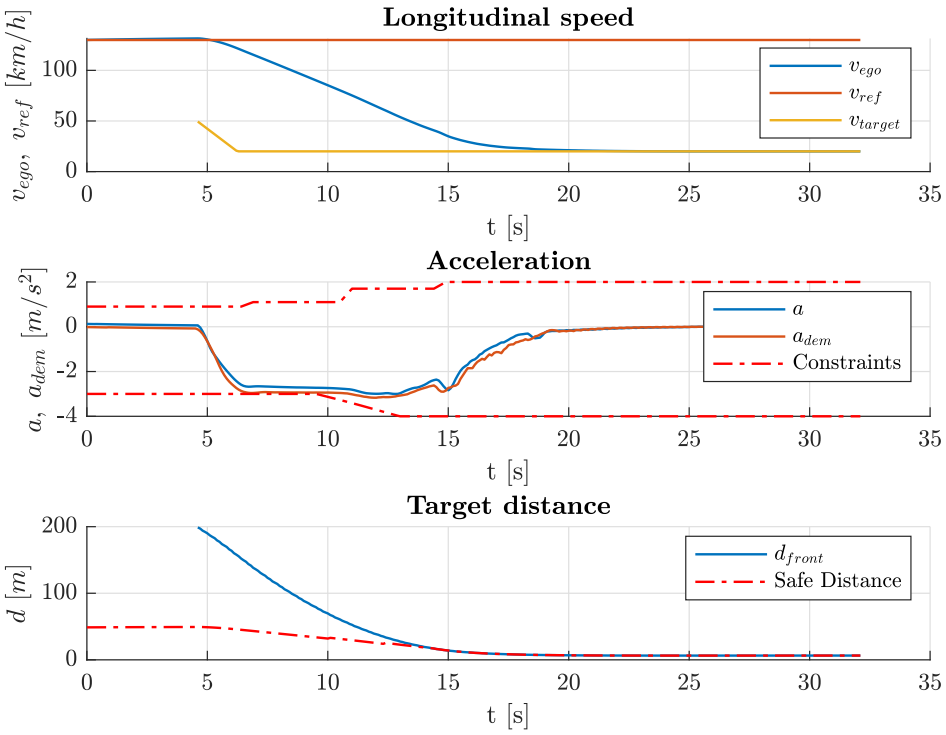


Figure 6.8: CCRM at $v_{ref} = 130\ km/h$ and $v_{target} = 20\ km/h$

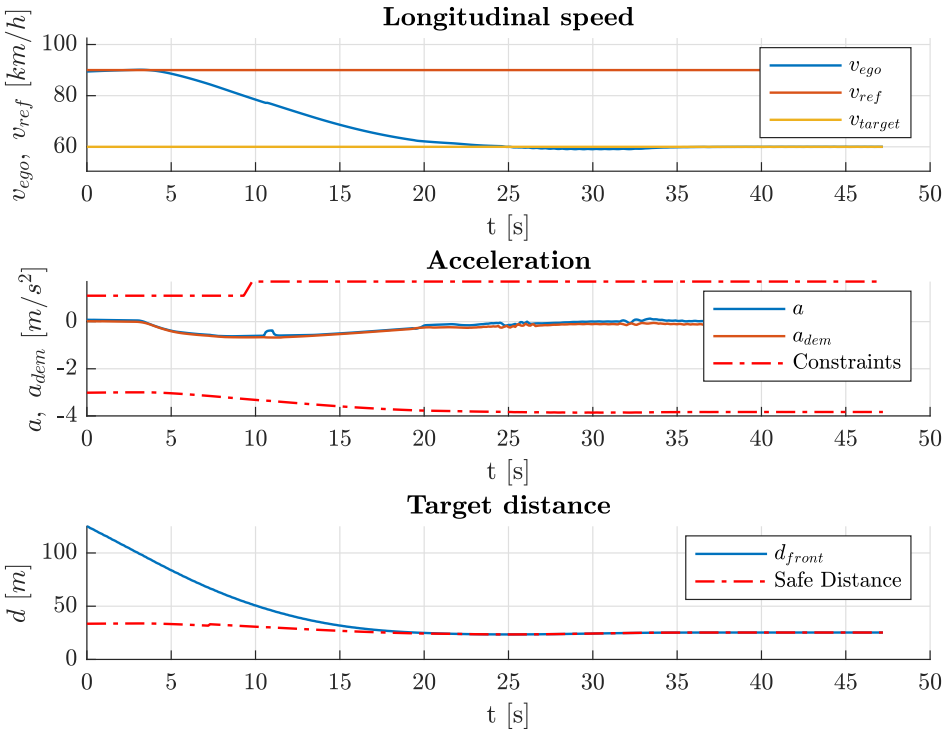


Figure 6.9: CCRM at $v_{ref} = 90\ km/h$ and $v_{target} = 60\ km/h$

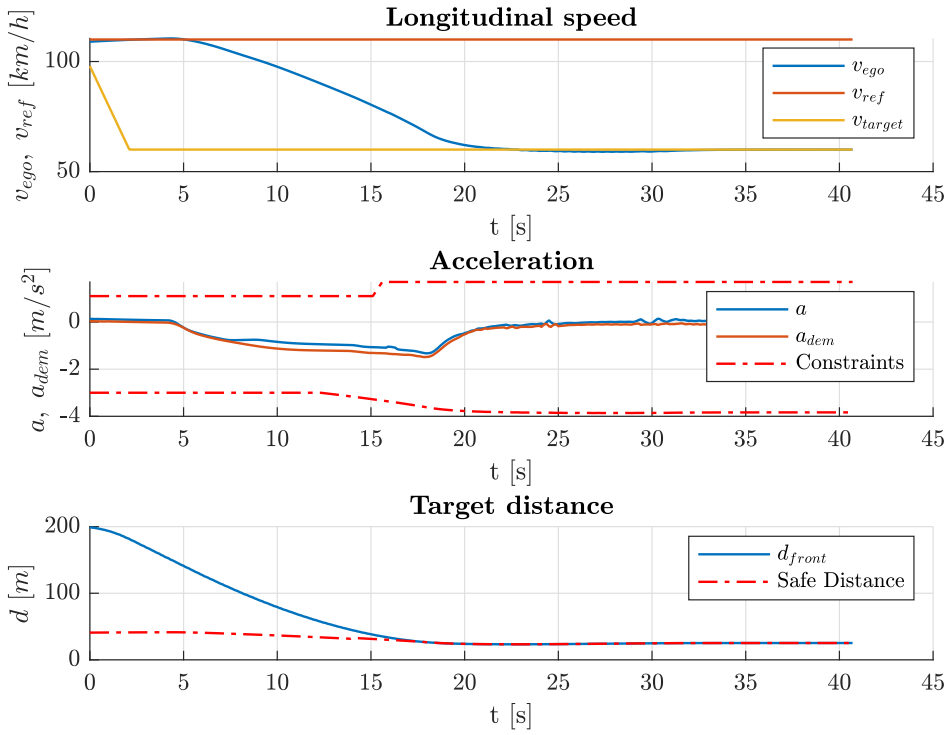


Figure 6.10: CCRM at $v_{ref} = 110\ km/h$ and $v_{target} = 60\ km/h$

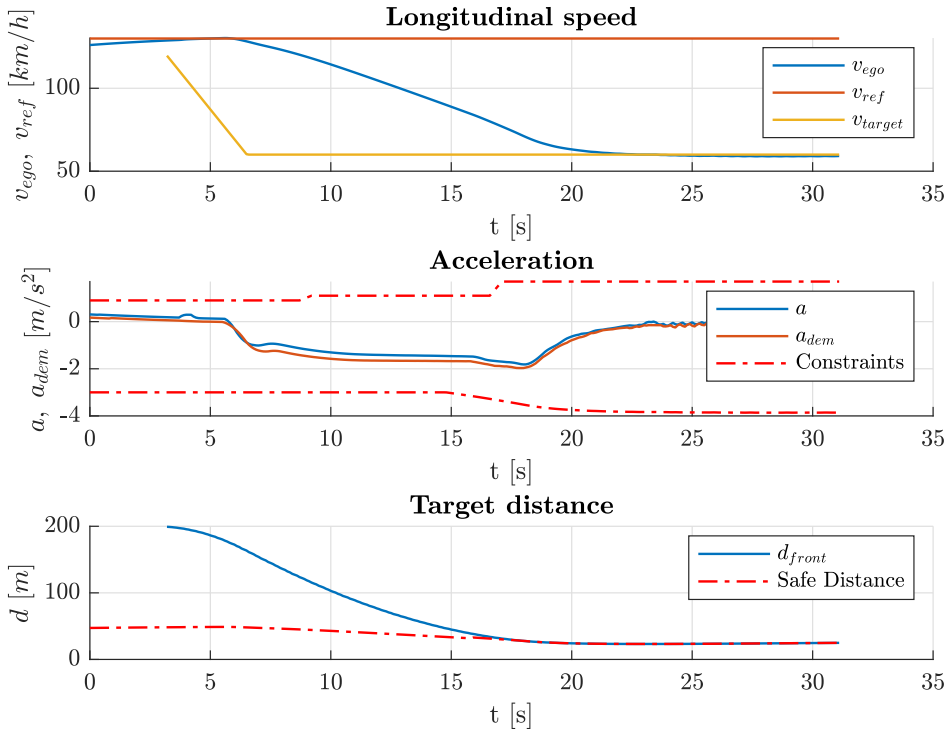
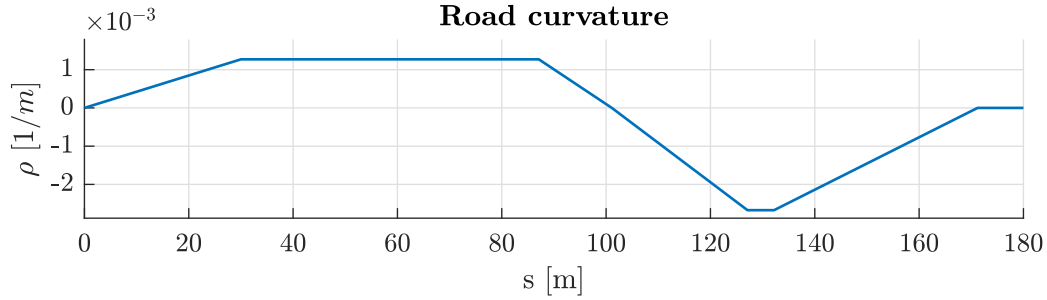


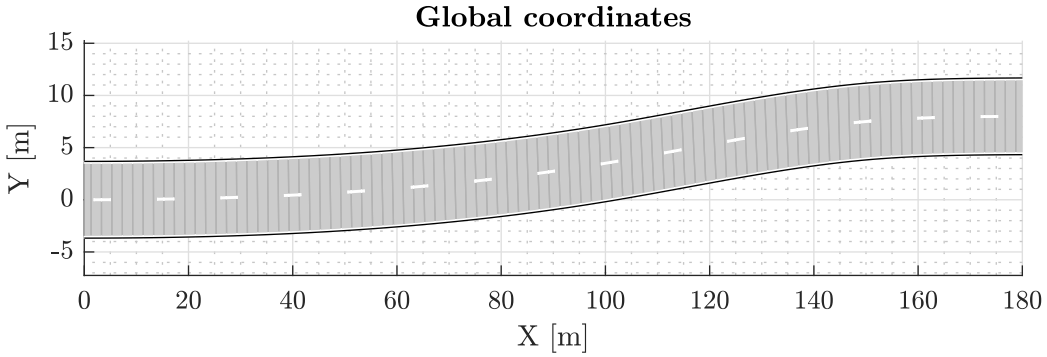
Figure 6.11: CCRM at $v_{ref} = 130\ km/h$ and $v_{target} = 60\ km/h$

S-Bend

Euro NCAP provides a scenario for testing the steering assistance functions called *S-Bend*, consisting in a sequence of two curves, whose curvature profile is shown in Figure 6.12a.



(a) S-Bend curvature profile



(b) Bird's eye plot

Figure 6.12: S-Bend scenario

Figures 6.13 to 6.15 show the simulation results when the VUT is traveling respectively at 80, 100 and 120 km/h: in all cases the lateral controller is able to keep the vehicle inside the lane with a very small deviation to the centerline.

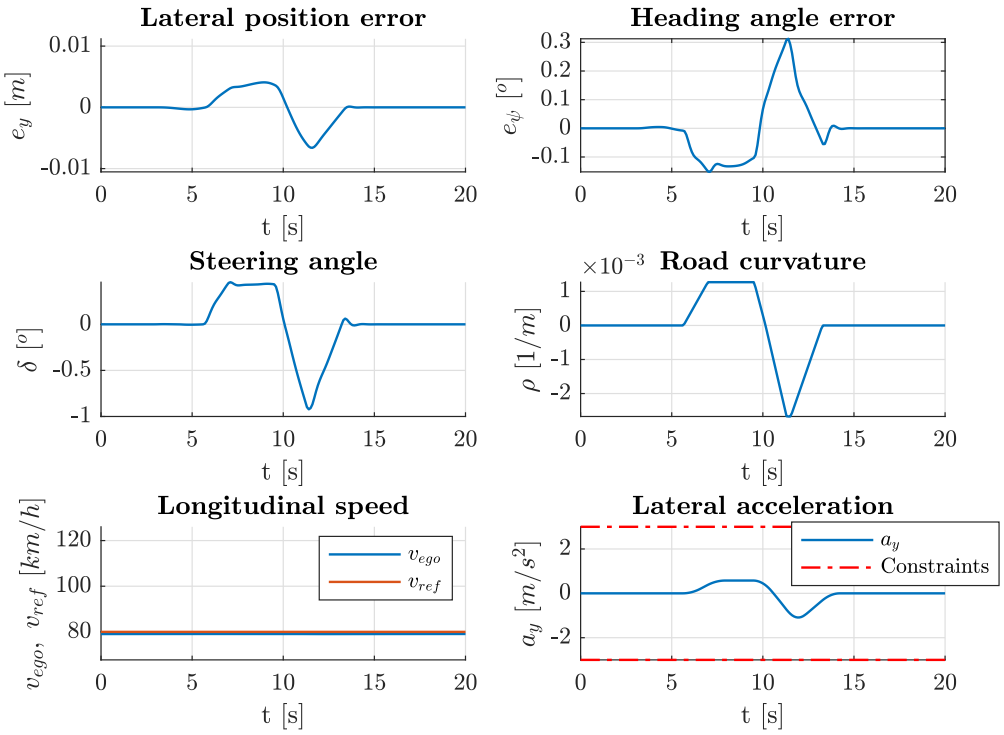


Figure 6.13: S-Bend at $v_{ref} = 80\ km/h$

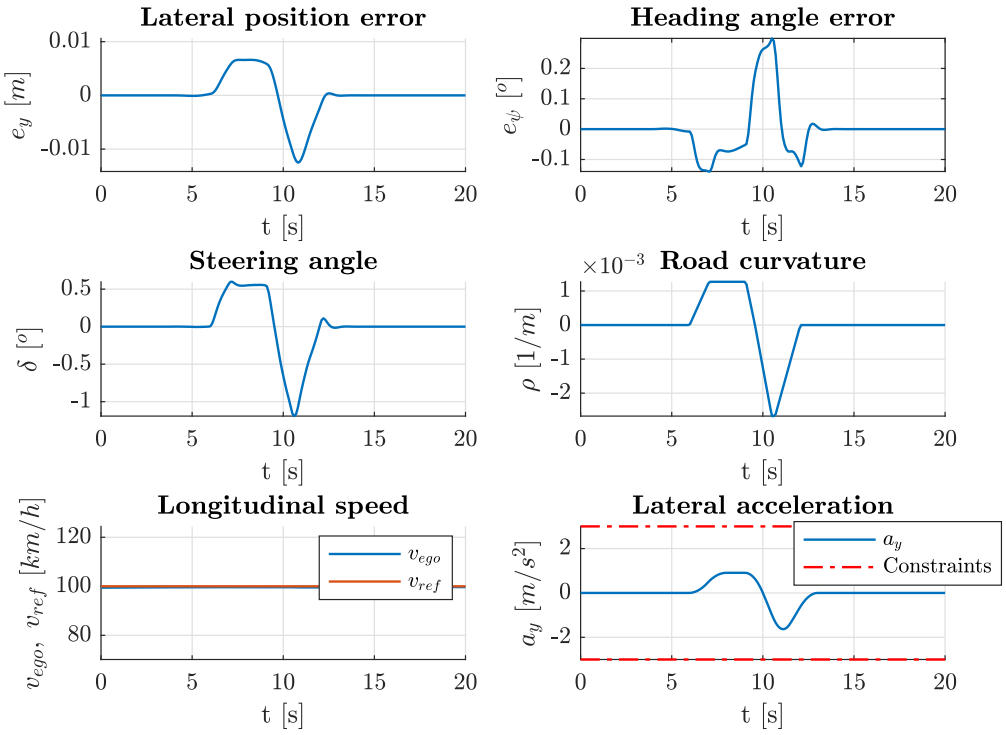


Figure 6.14: S-Bend at $v_{ref} = 100\ km/h$

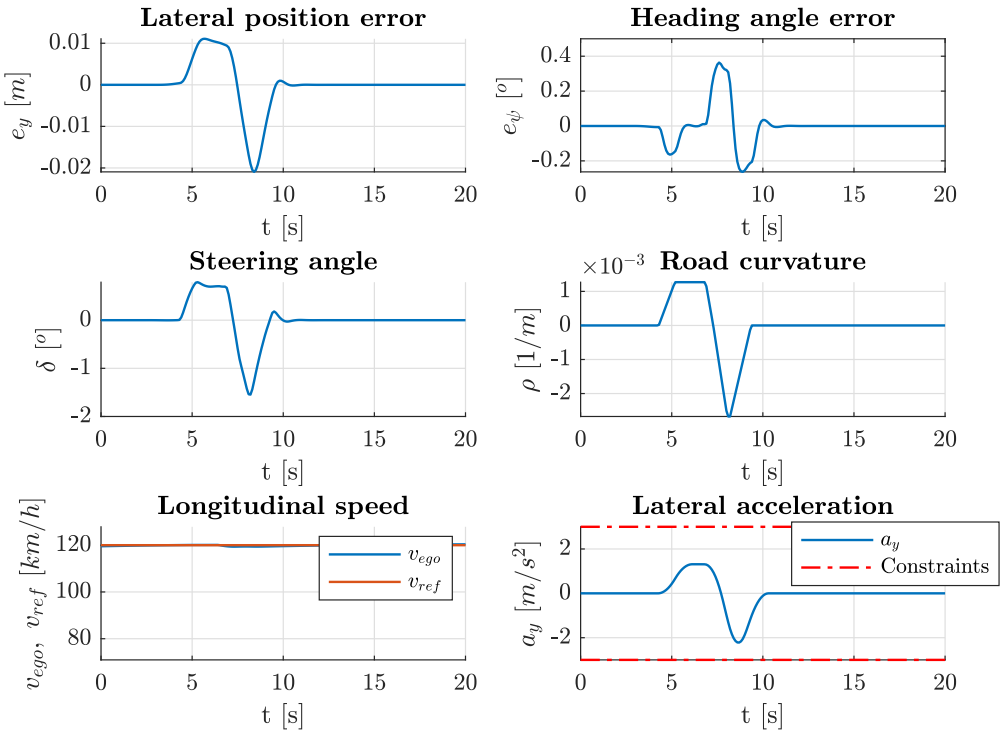


Figure 6.15: S-Bend at $v_{ref} = 120\ km/h$

6.3 Other scenarios

In addition to the scenarios provided by Euro NCAP, others have been designed for testing the performance of the lateral controller while traveling on a curved road and during a lane change maneuver.

In order to stress the Lane Centering functionality, it has been chosen a scenario composed by a sequence of tight curves, whose radius of curvature reaches values even below 50 meters (Figure 6.16). The test has been performed with the iACC engaged, meaning that the vehicle speed is decreased when approaching a curve, so that the lateral acceleration is kept within some comfort bounds.

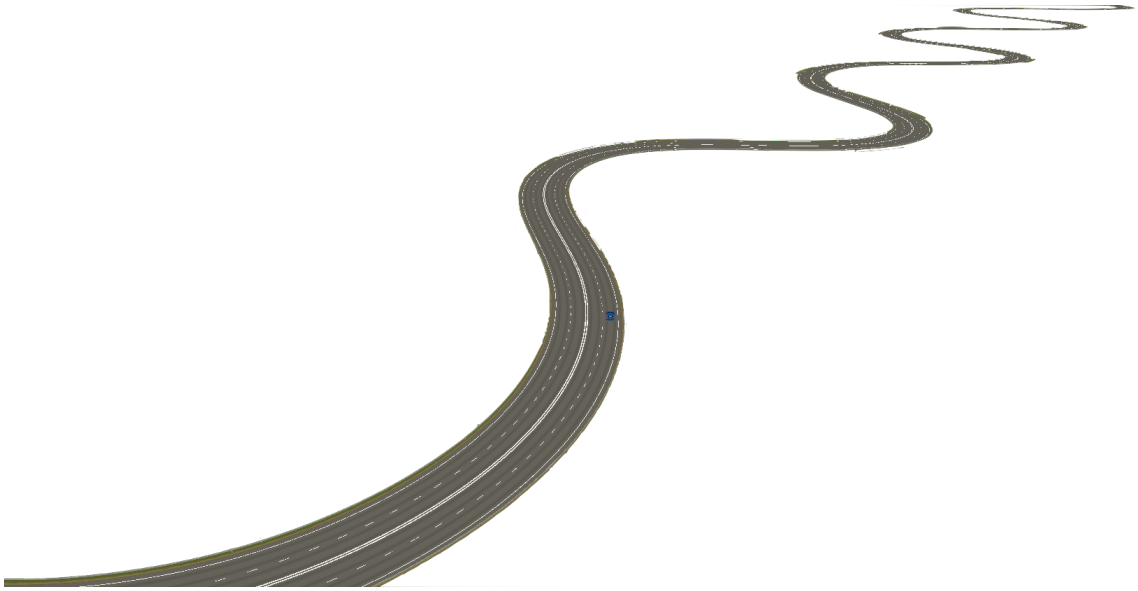


Figure 6.16: Curved road scenario

Figure 6.17 shows the simulation results: despite the high values of curvature, the lateral position error never exceeds 2 centimeters. It must be said that these results are strongly dependent on the accuracy of the camera sensor in the estimation of the distance to the lane boundaries, other than the road curvature.

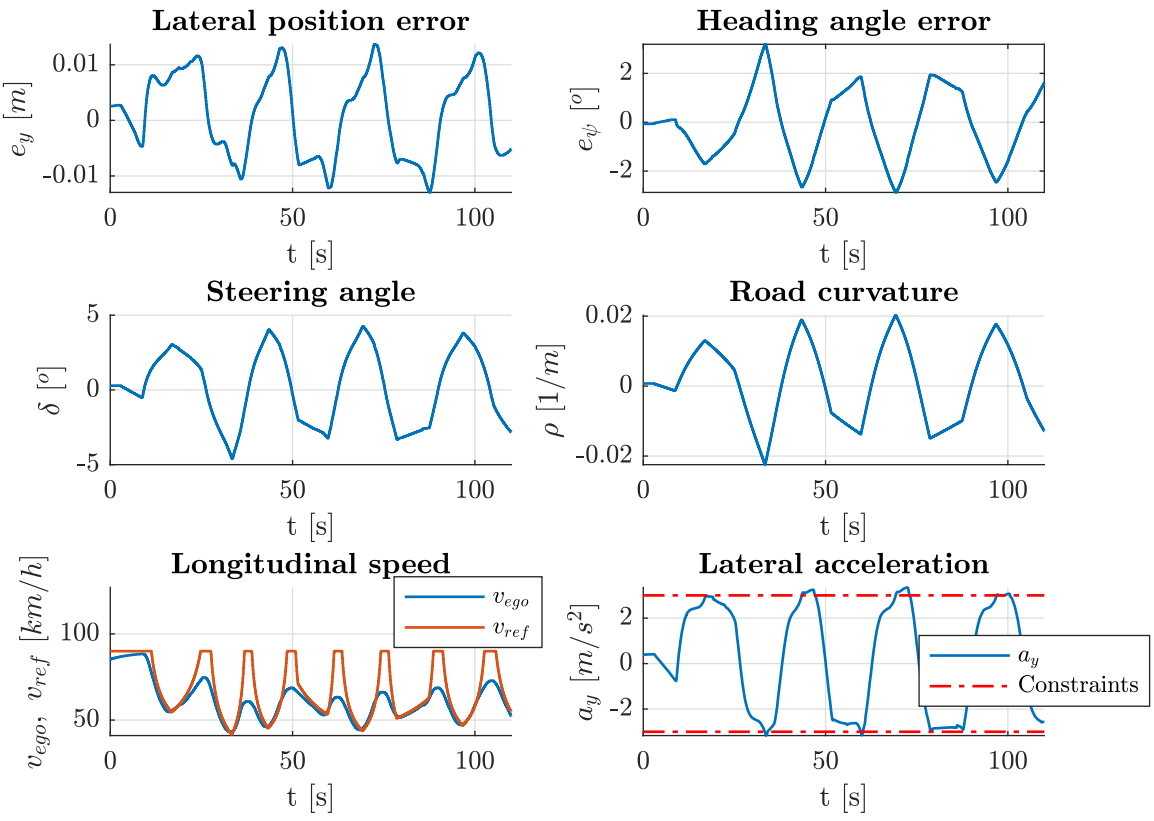


Figure 6.17: Simulation results for the Curved road scenario

As far as the lane change maneuver is concerned, Figures 6.18 and 6.19 show the simulation results of a double lane change performed respectively on a straight and curved road, while the vehicle is traveling at 90 km/h. The trajectory has been planned imposing the maximum relative lateral speed equal to 1 [m/s]; in both cases, the lateral controller is able to track the planned trajectory.

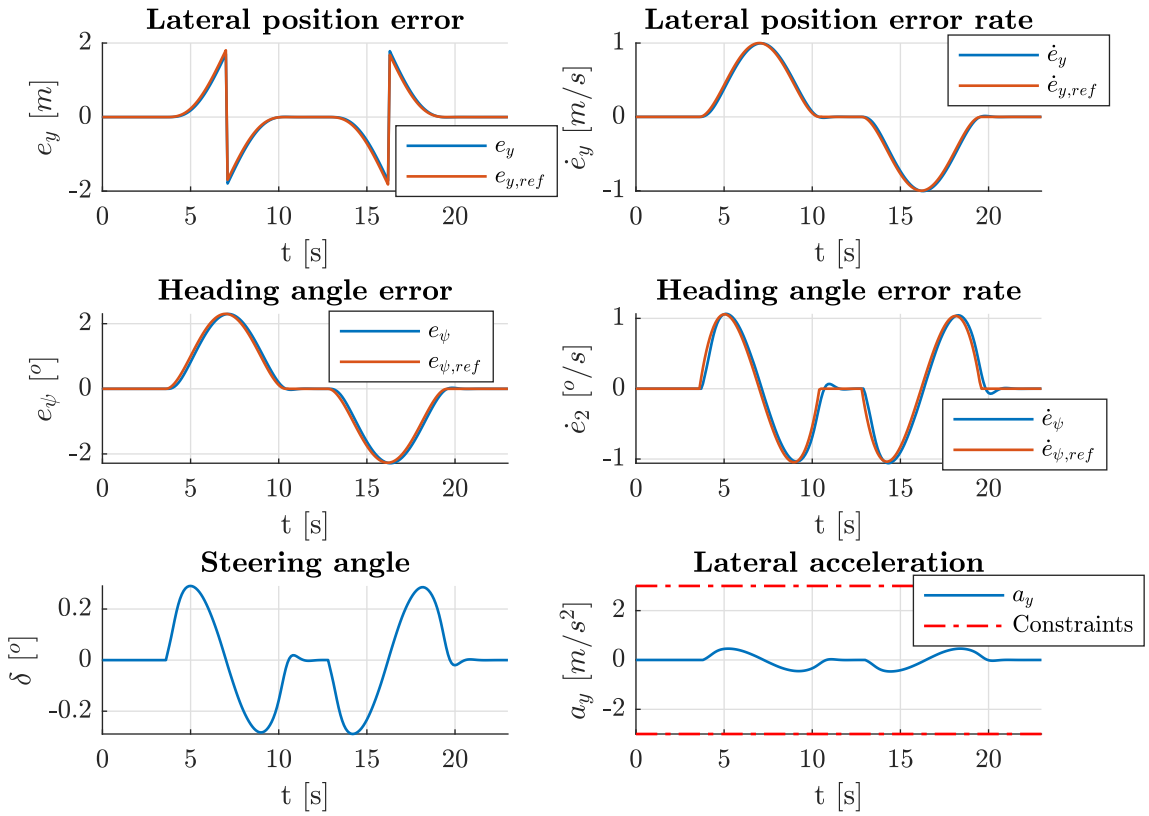


Figure 6.18: Lane change maneuver on straight road

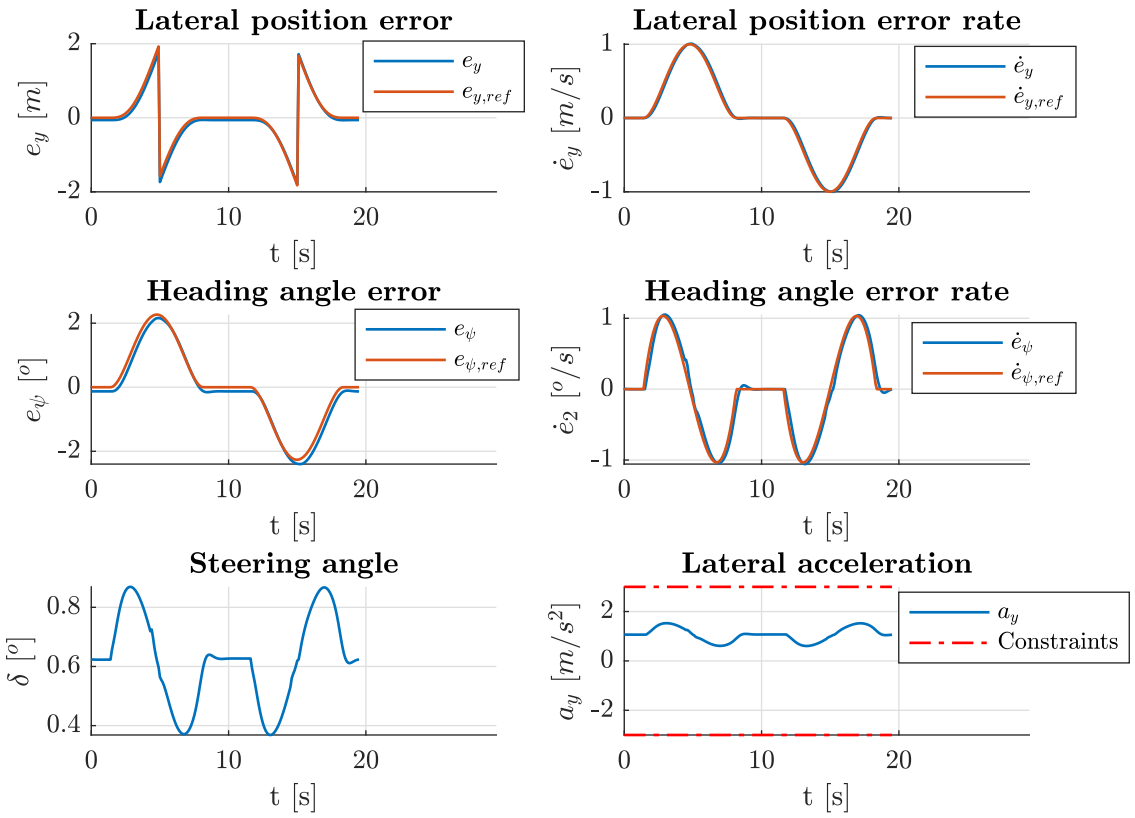


Figure 6.19: Lane change maneuver on curved road with constant radius of 600 meters

Chapter 7

Conclusions

Advanced Driving Assistance Systems are proving to be promising technologies for enhancing road safety, improving efficiency, and promoting comfort. Among these, the current Level 2 automated driving functionalities are able to provide both steering and acceleration/braking assistance in suitable scenarios, such as highways or limited access roads; however, highway driving assistance systems still have limited operative conditions, thus requiring the driver to constantly keep the hands on the wheel; even more, the greatest lack is the capability of automatic lane change; this means that as soon as the vehicle approaches a slower leading target, it follows him by decreasing the current speed so that the safe distance is kept. Nevertheless, this prevents reaching the desired speed, regardless of the presence of free lanes. To overcome this annoying limitation, driver intervention is required, who has to take control of the vehicle and perform a lane change.

Latest technologies focused on extending the operative conditions of Level 2 highway assist systems that rely only on onboard sensors by empowering the perception system with information extracted from high-definition maps. This led to the introduction of the so-called L2+ automated driving features: precise and detailed data about the road morphology and semantic allow to overcome the limitations of the sensors and even to extend their capabilities. Hence, high-definition maps are providing the enabling factors for new features, namely hands-free highway driving and automatic lane change.

7.1 Results

In this thesis, it has been shown the design of a full L2+ Highway Assist System (HAS), following the *rapid control prototyping* approach, starting from the design up to the verification and testing on a real-time simulator.

The designed system implements the most advanced commercially available automated driving features for the longitudinal and lateral motion control, namely the Intelligent Adaptive Cruise Control and Lane Centering respectively; in addition, the L2+ HAS has been enhanced with an Autonomous Lane Change functionality, meaning that the system is able not only to perform a lane change maneuver at the driver's request but also to continuously assess the feasibility and

the convenience of changing lane so that to autonomously start the maneuver.
The above functionalities have been implemented by partitioning the overall system into three subsystems, i.e. the Longitudinal Controller, Lateral Controller, and Lane Change Decision-Maker: the result is a modular structure, which has the advantages of being flexible and scalable. Model Predictive Control is the core of each subsystem: in Chapter 3 it has been shown how this framework makes it possible to handle physical, safety, and comfort constraints in a simple and effective way; furthermore, it allows to adjust the performances flexibly and intuitively, by varying the tuning weights of the objective function. It is worth noting that it is possible to change the driving style simply by changing the current set of weights; these are typically pre-computed, offering to the driver a finite number of choices, but they can also be ”learned” online employing other techniques such as Artificial Intelligence, to replicate the driver’s driving style. In chapter 4 has been proposed a design solution for the Autonomous Lane Change functionality, which is the most challenging. The problem was decoupled into two tasks: trajectory planning and decision-making. The former has been integrated into the lateral controller: as soon as it receives a lane change request, the planner computes the full preview of the reference output according to a 5th order polynomial; the trajectory is computed online to ensure comfortable maneuver by bounding lateral velocity, acceleration, and jerk. On the other hand, the decision-maker has been designed using an MPC-based approach, ensuring effectiveness and minimal design effort. In this case, the objective was not to control the vehicle motion but to assess both the feasibility and convenience of changing lanes. To do this, a duplicate longitudinal controller was fed with the distance preview related to the target belonging to the destination lane; thus, it is immediate to assess the feasibility of the maneuver: if the controller is not able to solve the optimization problem it is implied that a constraints violation occurred, hence the maneuver must be prevented. On the other hand, to evaluate the convenience it was retrieved the cost preview and compared to the one of the actual longitudinal controller.
In Chapter 5 it has been shown the offline simulation results for each of the designed functionalities, basing on a simplified model for the vehicle dynamics. This simulation environment was used for the first tuning phase. The results showed that the longitudinal and lateral controllers are able to control the vehicle according to the specified constraints; in addition, it has been shown that the use of a slack variable for constraint softening extends the capabilities of the controller, allowing it to handle situations in which constraints violation are unavoidable.
As far as the decision-maker is concerned, it behaves exactly as expected, meaning that it is able to evaluate the convenience of performing a left or right lane change and as soon as the destination lane is available, it requests a lane change, matching both the front and rear safe distance constraints.
Lastly, in Chapter 6 has been described the real-time implementation on a dSPACE simulator provided by Maserati; here the simulation model was highly detailed in all its components, such as the vehicle dynamics, steering, driveline, engine, virtual ECUs and sensors. Here, the designed system has been tested on several scenarios defined by Euro NCAP in the Test &

Assessment Protocol for Highway Assist Systems [16] and the vehicle passed all the tests.

7.2 Future work

As previously mentioned, the designed L2+ Highway Assist System was able to work as expected in a real-time simulation environment. The next steps include:

- Study of the effects of measurement noise: so far it has been assumed that the sensing interface is able to provide as output ideal signals, i.e. not affected by measurement noise and offset free. Advanced techniques are available for processing the signals produced by radars and camera sensors, such as Kalman filters, multi-object tracking, computer vision, and sensor fusion. In addition, the use of high-definition maps allows to further reduce the uncertainty on some data, such as road curvatures. Despite this, considering purely ideal signals is not realistic, so it is advisable to study the impact of measurement noise on system performance.

- Integration of a lateral motion planner: at the current state, the lateral controller is fed with a null reference output, meaning that it tries to keep the vehicle close to the centerline, minimizing the position and yaw errors; this approach is suitable for highway scenarios, where the roads are designed so that to ensure smooth transition among sections with different curvature, i.e. employing clothoid. However, a motion planner may improve even more comfort by planning a trajectory that does not necessarily overlap with the lane centerline: *curve-cutting* allows to reduce the lateral acceleration by following a trajectory with curvature lower than the current lane, although this implies an offset with respect to the centerline (Figure 7.1).

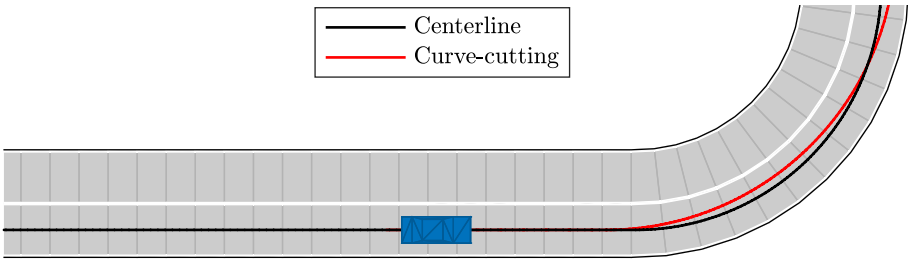


Figure 7.1: Curve-cutting: small lateral offset is allowed to decrease the curvature of the trajectory.

- Improve the prediction of targets motion: to estimate the preview of the targets traveled space it has been used a simple constant-acceleration model, despite it could be inaccurate since the prediction horizon is relatively large, i.e. eight seconds. More accurate prediction models may be used to improve the performance of the longitudinal controller. Furthermore, it would be interesting to predict also the lateral motion of the targets, thus

allowing the controller to act in advance to a possible lane change; for this purpose it could be exploited the detection of the target's turn indicators and the motion estimation. Once again, it is important to highlight the scalability of the designed system, since it is already set up to receive an externally calculated preview.

- Enhance the triggering algorithm for the Autonomous Lane Change: in Section 5.3 it has been shown how the decision-maker is perfectly able to assess the feasibility of the maneuver but it may request a lane change too frequently, resulting in potentially uncomfortable behavior. At the current state, the decision-maker does not take into account the measure of the benefit that the lane change would bring, since it only compare the cost previews of changing lane and keeping the current lane respectively. An improvement may consist in integrating the gain (i.e the difference between the two costs) and requesting the lane change as soon as it reaches a certain threshold: this would delay the trigger if the gain is relatively small.
  Furthermore, what limits the evaluation of the convenience of changing lane is the prediction horizon of the decision-maker; this must be the same to the one of the longitudinal controller (otherwise the cost previews would no longer be comparable), preventing from increasing it arbitrarily since it would lead to a large increment of the computational effort. A possible trade-off may be to split the feasibility and convenience assessments by adding a secondary decision-maker, still MPC-based but with a simpler model and longer prediction horizon: this would be in charge of evaluating only the convenience by computing the costs previews of both adjacent and current lanes.
  It must be said that regardless the prediction horizon, the convenience is assessed in the short term, due to the limited range of the radars. To overcome this limitation different technologies may be exploited, such as *Vehicle-to-everything (V2X)* communication.

- Equip the Autonomous Lane Change with abortion capability: it may happen that once the decision-maker requests a lane change, the traffic conditions in the surroundings suddenly change so that the maneuver is no longer convenient. In these situations it may be appropriate to abort the lane change; to do this, it is possible to exploit the decision-maker itself by setting the starting lane as a target so that during the maneuver it assesses both the convenience and feasibility of the abortion.

- Field testing: the last step is definitely the field testing, that is the implementation on a real vehicle, despite it may be a complex task; indeed, so far it as been assumed the availability of all the sensors output signals and that they have already been processed. The solution that would takes the least time and effort is to use an in-vehicle prototyping system, such as the dSPACE MicroAutoBox III, and interface it on the CAN bus as an additional node. Nevertheless, typically signal processing is performed inside the ADAS control unit, thus preventing to access them from the CAN bus; in that case it may be necessary to intervene from the ECU supplier to make the required signals available or directly integrating the

designed functionality in the control unit software.

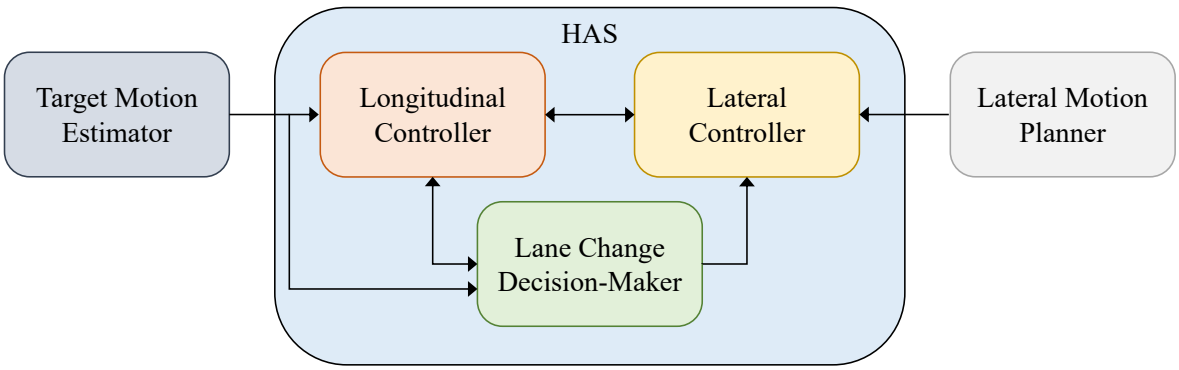


Figure 7.2: Possible integration of a target motion estimator and lateral motion planner

Bibliography

[1] *Automated Vehicles for Safety | NHTSA*. [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety.

[2] *J3016C: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International*. [Online]. Available: https://www.sae.org/standards/content/j3016_202104 (visited on 07/07/2021).

[3] *Defining the 'Plus' in L2+*, en-US. [Online]. Available: https://newsroom.intel.com/articles/defining-plus-l2/.

[4] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017. doi: 10.1017/9781139061759.

[5] E. F. Camacho and C. Bordons, *Model predictive control.* Springer, 2003, isbn: 978-18-523-3694-3.

[6] J. Rawlings, D. Mayne, and M. Diehl, *Model predictive control: theory, computation and design*, 2nd edition. Santa Barbara: Nob Hill Publishing, LLC, 2020, isbn: 9780975937754. [Online]. Available: https://sites.engineering.ucsb.edu/~jbraw/mpc/MPC-book-2nd-edition-3rd-printing.pdf.

[7] M. Behrendt. (Oct. 2009). "Mpc scheme basic," [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/1/11/MPC_scheme_basic.svg. Martin Behrendt, CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons.

[8] *Optimization Problem - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/mpc/ug/optimization-problem.html.

[9] *Constraints on Linear Combinations of Inputs and Outputs - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/mpc/ug/constraints-on-linear-combinations-of-inputs-and-outputs.html.

[10] *BS ISO 15622:2018 Intelligent transport systems. Adaptive cruise control systems. Performance requirements and test procedures.* BSI Standards Limited, 2018, isbn: 978-0-580-98139-5.

[11] R. Rajamani, *Vehicle dynamics and control.* Springer, 2012, isbn: 978-14-614-1433-9.

[12] *Model Predictive Control Toolbox*, en. [Online]. Available: https://www.mathworks.com/products/model-predictive-control.html.

[13] *Vehicle Body 3DOF*, 3DOF rigid vehicle body to calculate longitudinal, lateral, and yaw motion - Simulink. [Online]. Available: https://www.mathworks.com/help/vdynblks/ref/vehiclebody3dof.html?s_tid=doc_ta.

[14] *Automated Driving Toolbox*, en. [Online]. Available: https://www.mathworks.com/products/automated-driving.html.

[15] *SCALEXIO*, en. [Online]. Available: https://www.dspace.com/en/inc/home/products/hw/simulator_hardware/scalexio.cfm.

[16] *Assisted Driving - Highway Assist Systems, Test & Assessment Protocol*. Euro NCAP, Sep. 2020, Version 1.0. [Online]. Available: https://cdn.euroncap.com/media/58813/euro-ncap-ad-test-and-assessment-protocol-v10.pdf.

[17] H. Winner, "Adaptive cruise control," in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed. London: Springer London, 2012, pp. 613–656, isbn: 978-0-85729-085-4. doi: 10.1007/978-0-85729-085-4_24.

[18] *MPC Modeling - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/mpc/gs/mpc-modeling.html.

[19] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," en, *Control Engineering Practice*, vol. 41, pp. 124–133, Aug. 2015, issn: 09670661. doi: 10.1016/j.conengprac.2015.04.006. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0967066115000726.