POLITECNICO DI TORINO

MSc Automotive Engineering

A.Y. 2023/24

Driver assistance system design laboratory report

Carlo Vittorio Colucci 329703

Riccardo Bressani 323665

Luca Marchetto 323437

July 8, 2024

Projects introduction

In modern vehicles the passengers' safety plays a crucial role in the development process; with this aim, the introduction of advanced driver assistance systems (ADAS) is a very effective solution, that is being adopted by the totality of the car manufacturers. At the same time the enhancement, provided by these systems, to the vehicle driving comfort is becoming more and more relevant in the customers' choice. Their continuous improvement, as well as the implementation of more advanced functionalities, are object of continuous investigation and research in the automotive sector.

In these projects, the control architectures of an Adaptive Cruise Control and a Lane Keeping Assist systems are implemented and discussed.

The same vehicle is considered for the two projects; the vehicle is a mid-engine two-seater sports car with a 3.5 liter spark ignition engine, corresponding to the vehicle 7 of the list provided in the file 'Dataset_for_students.pdf'. The primary geometrical data and inertial properties of the vehicle are:

length = 4,250 mm	width = 1,900 mm	height = 1,160 mm
a = 1,600 mm	b = 1,050 mm	l = 2,650 mm
m = 1,480 kg	t_l = 1,502 mm	t_l = 1,578 mm
J_x = 590 kg m^2	J_y = 1,730 kg m^2	J_z = 1,950 kg m^2
J_{xz} = -50 kg m^2	h_G = 430 mm	

$J_{w,f}$ (each) = 0.7 kg m^2	$J_{w,r}$ (each) = 0.7 kg m^2	J_e = 0.48 kg m^2	J_t = 0.08 kg m^2

f_0 = 0.013	K = 6.5 · 10^{-6} s^2/m^2	$R_{e,f}$ = 310 mm	$R_{e,r}$ = 315 mm

Aerodynamic data:

S = 1.824 m^2	C_x = 0.335	C_z = -0.34
C_{My} = 0	$(C_y)_{,\beta}$ = -2.3 1/rad	$(C_N)_{,\beta}$ = -0.3 1/rad

Gear ratios (front wheel drive):

τ_I = 0.3111	τ_{II} = 0.4751	τ_{III} = 0.6859	η_t = 0.87
τ_{IV} = 0.9141	τ_V = 1.1614	τ_f = 0.2469	

The tyres experimental data are extrapolated from the provided Excel file 'TyreCharacteristics', assuming that the vehicle is equipped with 2 245/35 R19 tyres at the front and 2 255/35 R20 at the rear. This choice is coherent with the type of vehicle considered. The dataset is used to extract constant values of cornering and self aligning stiffness, according to the linear tyre model adopted in the Segel model.

Contents

1 Project 1: Adaptive Cruise Control (ACC) design

Keeping an appropriate distance from a leading vehicle is important for safety reasons and the goal of this controller is to actively ensure a target gap between the two vehicles, or even more than two as explained in the Section 1.2. In particular the project is focused on a decentralized non-connected ACC, based on a leader-follower approach.

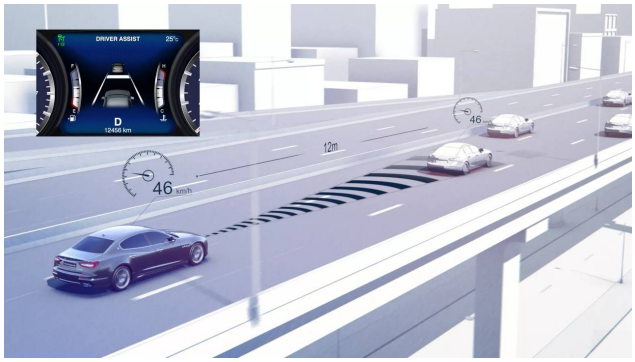


Figure 1: ACC representative figure

1.1 Exercise 1: Single lead vehicle

The first part of the project consists in considering two vehicles: a leader and a follower, called Ego Vehicle, equipped with the ACC system. The goal is to design the ACC controller in order to maintain a constant time gap from the vehicle in front. The system is subsequently tested on different speed profiles of the leader vehicle.
The ACC system architecture is hierarchical: the upper level controller determines the desired vehicle acceleration, while the lower level one provides the throttle/brake inputs required to achieve the desired acceleration. Both the controllers are developed in the following sections. An overview of the testing bench architecture for the two vehicles, during certain driving missions is represented:

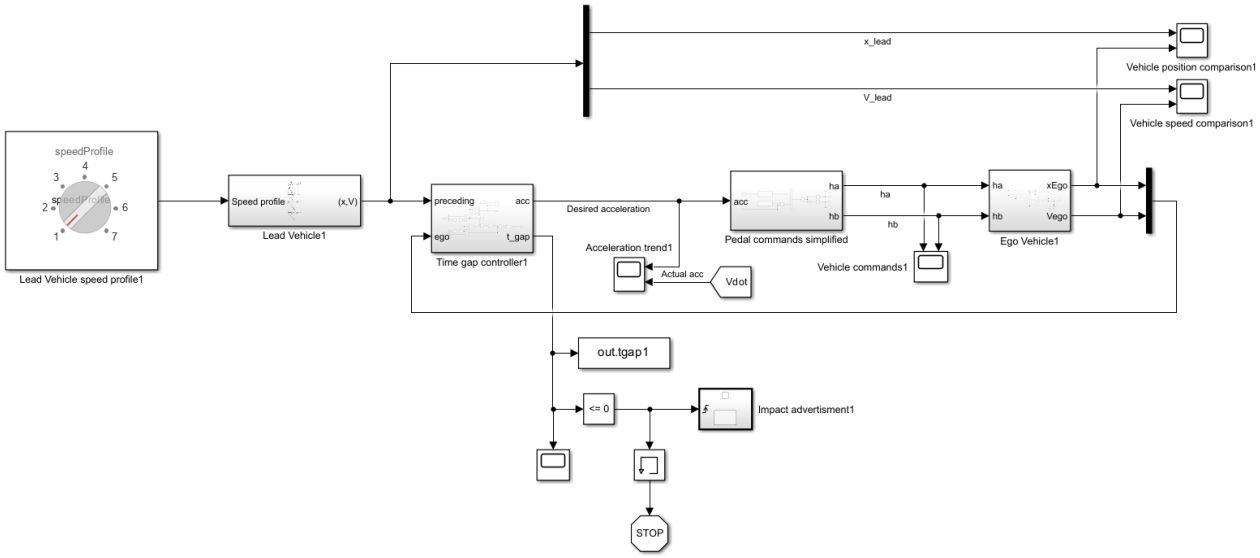


Figure 2: ACC controller testing architecture

1.1.1 Upper level ACC controller

A time based controller is implemented because, in a string of vehicles, it ensures the platoon stability, meaning that the spacing errors from the leader vehicle to the subsequent ones do not increase. The controller works exploiting the following equation:

$$a_i = -\frac{1}{h}\left(\lambda\delta_i + \dot{\varepsilon}_i\right) \tag{1}$$

where:

- a_i is the desired acceleration command;
- h is the desired time gap;
- λ is a proportional gain to be tuned;
- $\delta_i = \varepsilon_i + h\dot{x}_i$ is the spacing error of the ego vehicle with respect to the desired position;
- $\varepsilon_i = x_i - x_{i-1}$ is the distance between ego and leader vehicles.

The equation is implemented in the following Simulink block diagram:

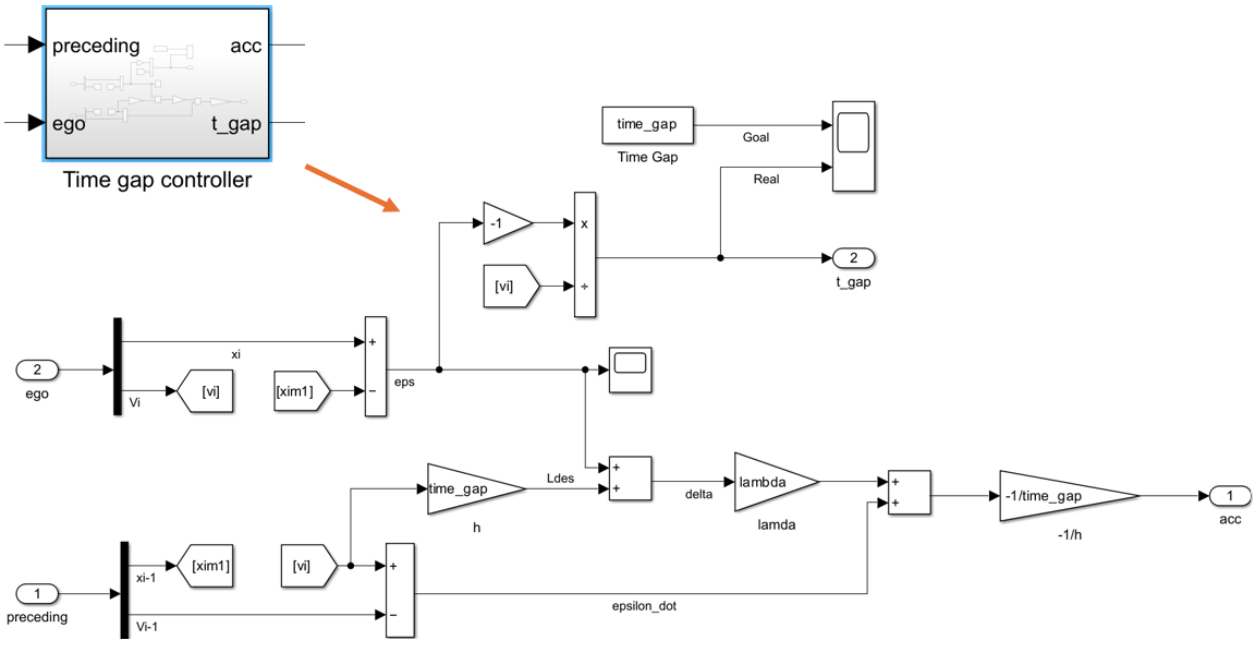


Figure 3: Time gap controller

1.1.2 Lower level ACC controller

The result of the subsystem presented in the previous section represents a desired acceleration that the vehicle should follow in each time instant. By the way, this is not enough for the system implementation, since this reference must be translated into vehicle commands, in the form of percentage of throttle (h_a) and brake (h_b) actuation. To do so, a non-steady state longitudinal dynamic model is adopted, assuming rigid parts and no slip at the tires: the reason for this is that slip would be difficult to measure and would not provide significant improvements to the calculated commands. Moreover, the considered model can be easily reversed to obtain the commands h_a and h_b. The first step is the computation of the longitudinal force:

$$F_x = \left[m_e \cdot \dot{V} + (A + B \cdot V^2 + C \cdot V^4)\right]$$

where m_e is the equivalent mass accounting also for the inertia of the rotating components and A, B, C are the resistance coefficients considering road grade (if present), rolling and aerodynamic resistances. The acceleration $\dot{V}$ is assumed equal to the command generated by the upper level controller, to trace the desired reference, while the speed V is directly taken by the real vehicle model, for an accurate estimation of the resistant force.

Then, for each iteration, both the pedal commands are calculated by means of the fol-

lowing expressions:

$$h_a = F_x \cdot \frac{R_c \cdot \tau_g \cdot \tau_f}{\eta_t \cdot T_{e,max}}$$

$$h_b = -F_x \cdot \frac{R_c}{T_{b,max}}$$

where:

- R_c is the tire loaded radius;
- τ_g is the gearbox transmission ratio;
- τ_f is the differential transmission ratio;
- η_t is the total driveline efficiency;
- $T_{e,max}$ is the max ICE torque according to the current angular speed of the crankshaft, computed inside the real vehicle model;
- $T_{b,max}$ is the max braking torque that can be exerted on the vehicle at each instant, computed by the ABS function, inside the real vehicle model.

The negative sign present in the braking pedal command expression is used to make positive the output value in braking conditions (negative longitudinal force).
Finally, by means of saturation blocks, the commands are kept below their maximum threshold and set none when the target condition doesn't require the pedal actuation.
The pedals command controller blocks architecture is shown:

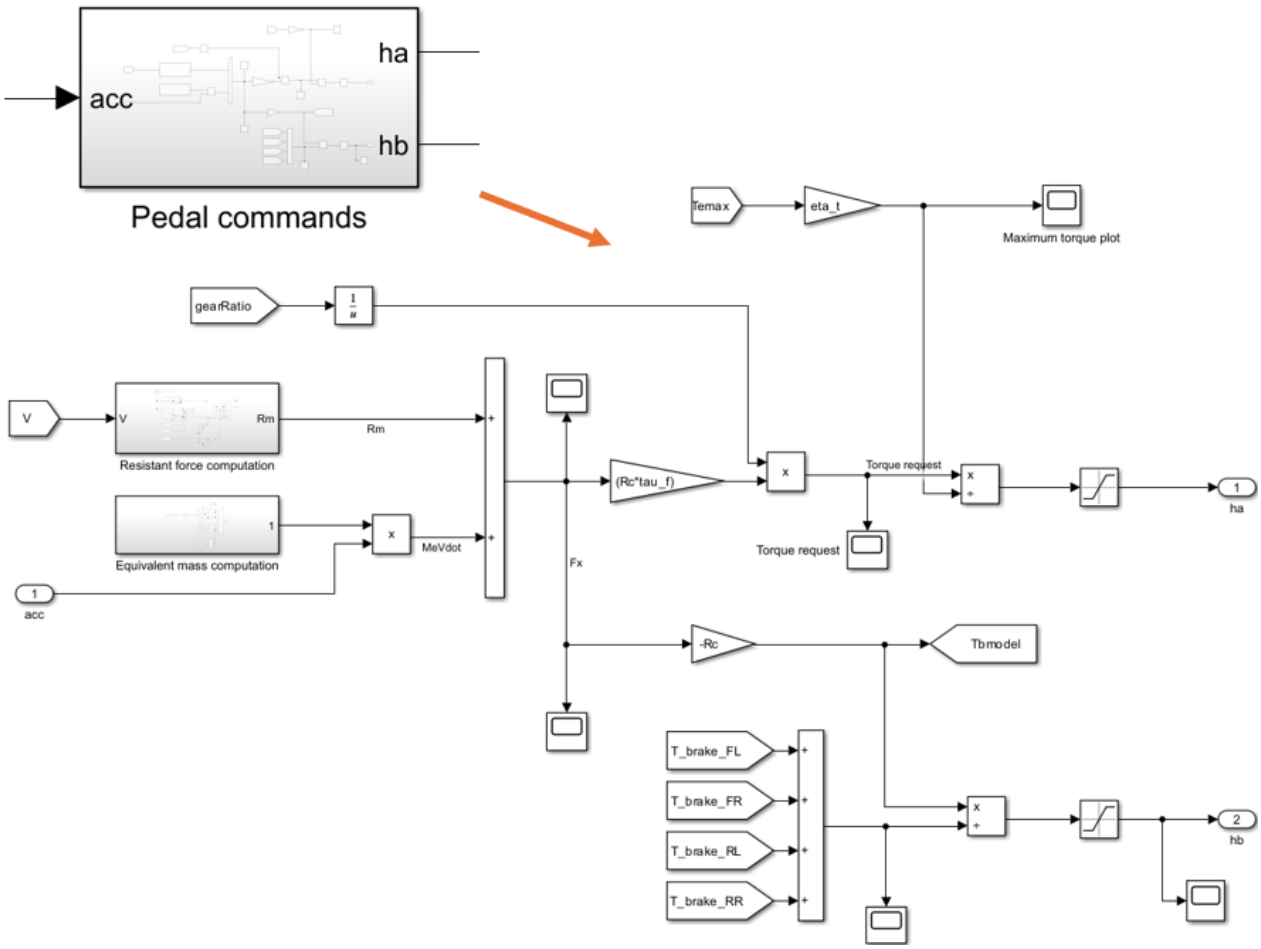


Figure 4: Pedals command controller

These pedal commands are then collected by the accurate vehicle model, whose design is not of interest for this laboratory activity, and the real resulting vehicle acceleration is evaluated, retrieving its speed and positioning as well.

1.1.3 Controller analysis and results comparison

ù Tuning the equivalent proportional gain term λ and the time gap h (control parameters of the Eq.1) means to obtain a control for the Ego vehicle that keeps the time gap, from the leading vehicle, as close as possible to the target value in a smooth way, avoiding too strong accelerations compromising the passengers' comfort and safety.
The two values are tested separately in order to understand how they affect the controller behaviour. The tracked parameter, chosen to carry out this analysis, is the time gap between the two vehicles during the driving mission, since it is a good indicator to track the overall controller behaviour.

Firstly, the effect of the variation of λ is analysed, keeping h = 5 s; speed profile 6 is taken as reference for the tuning, being a quite severe mission. The tuned value is subsequently tested on all the other profiles. The results are shown below:

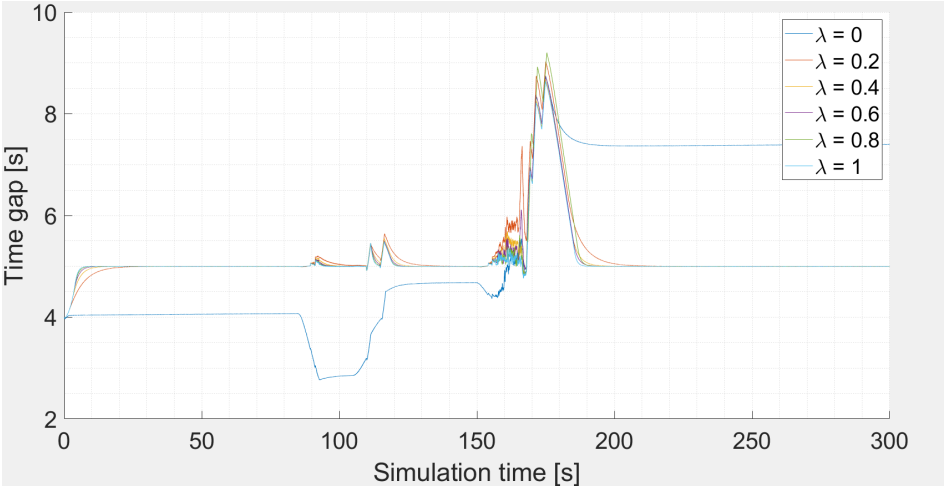


Figure 5: Lab01-Ex1/Proportional gain sensitivity

The peak of the time gap between 150 and 200s is not related to the controller's behaviour, but to the Ego vehicle acceleration capability, indeed it is not able to follow the steep speed increase of the car in front.

To better understand the different trends, a zoom on two quite representative transients is reported and commented:

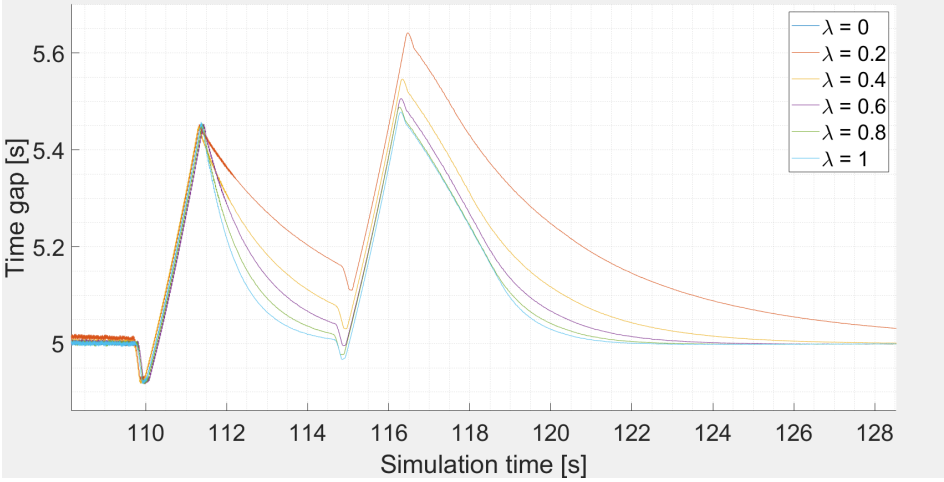


Figure 6: Proportional gain sensitivity zoom

The first trivial consideration is that, as expected, the controller with $\lambda = 0$ is not working correctly: indeed, it is able to minimize the speed difference between the vehicles, but not the error on the relative position. For what concerns the other values, it can in general be observed that the higher the value of λ, the quicker the transient response. Surely, a fast response is desirable for safety reasons; on the other hand, having a too quick reaction could negatively affect both the occupants' comfort and the fuel economy. For this reason, a very high value of λ would not necessarily provide an optimal controller behaviour. Moreover, further increasing λ, the vehicle transient behaviour tends to saturate: for very fast controllers, the system transient is not anymore limited by the controller itself, but by the vehicle dynamic response.

The value of $\lambda = 0.6$, apparently providing a good trade-off between the mentioned aspects, is adopted for the following simulations.

Always for what concerns safety, the most critical parameter is the desired time gap h. From a theoretical point of view, it is known that, if the vehicle response is modeled by a first order transfer function of the form:

$$\ddot{x}_i = \frac{1}{1 + \tau s} a_1$$

where a_1 is the reference acceleration requested by the controller and τ is a time constant representing the vehicle dynamic response, to guarantee the string stability it should be verified $h \geq 2\tau$. Although, in this case a more accurate longitudinal dynamics vehicle model is adopted, so an explicit value of τ is not known; for this reason, different simulations are carried out progressively increasing the value of h, looking for the first one that ensures to avoid the collision with the vehicle in front.

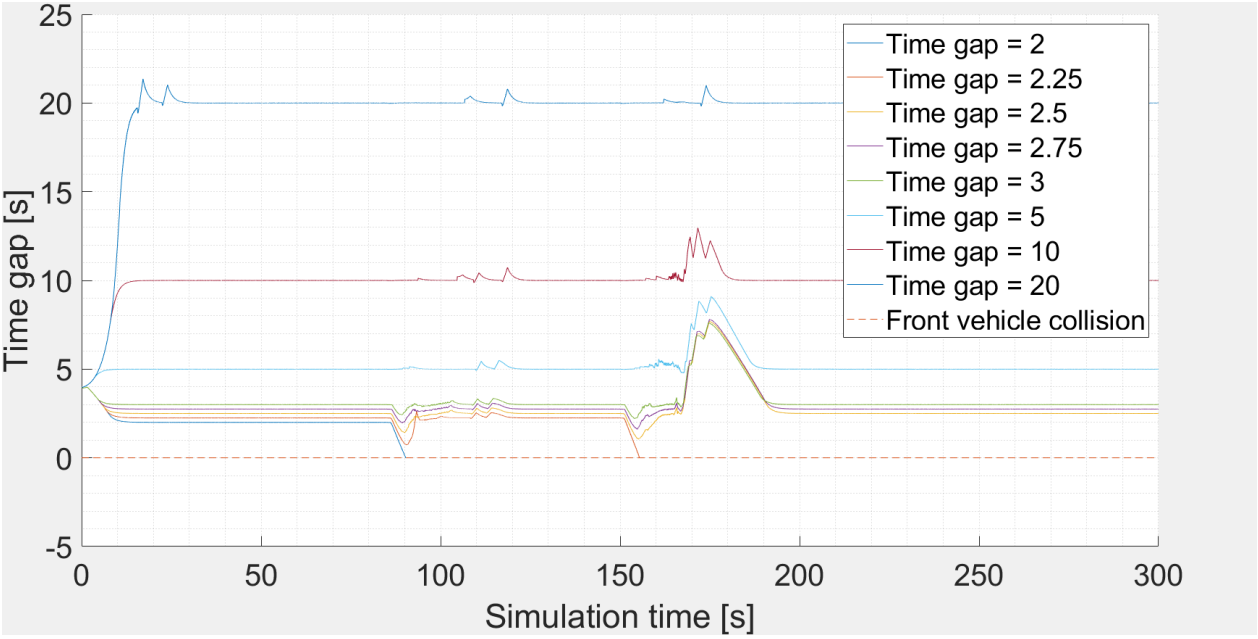
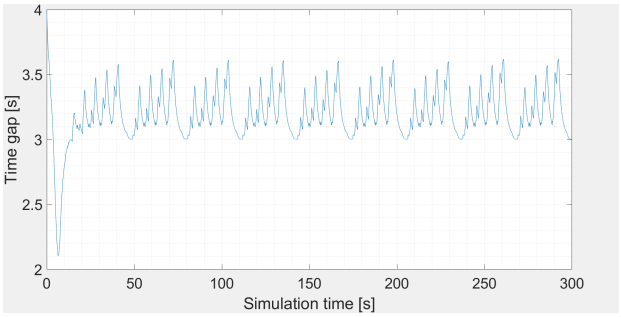
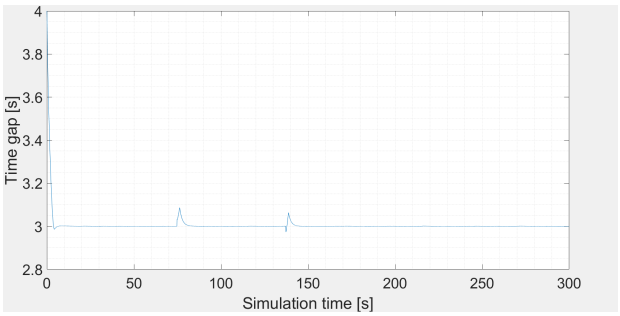


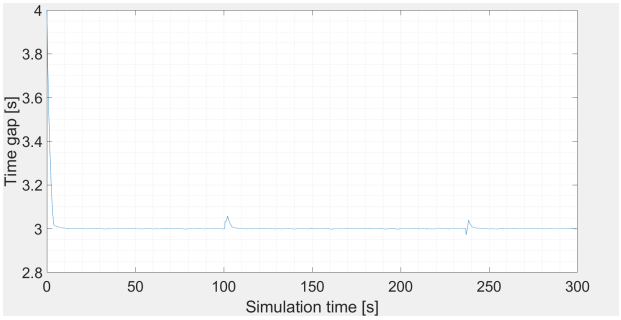
Figure 7: Time gap sensitivity

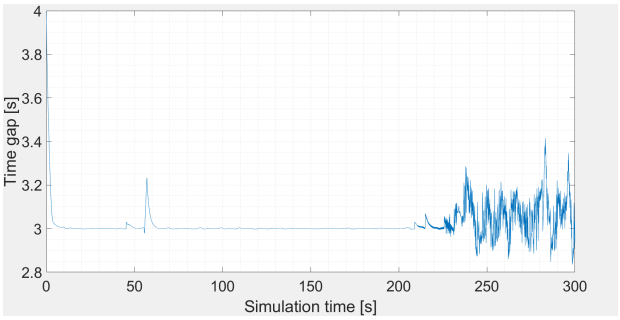
It can be observed that the first two values fail in maintaining the distance, while $h = 2.5s$ is the first value that provides a satisfactory result. Taking a small safety margin, the other speed profiles are tested with $h = 3s$ to determine if this value can be considered a safe lower limit. On the contrary, no upper limit for h value is observed; indeed, the slower acceleration command (induced by high h values) doesn't have critical safety implications due to the higher distances between the vehicles.



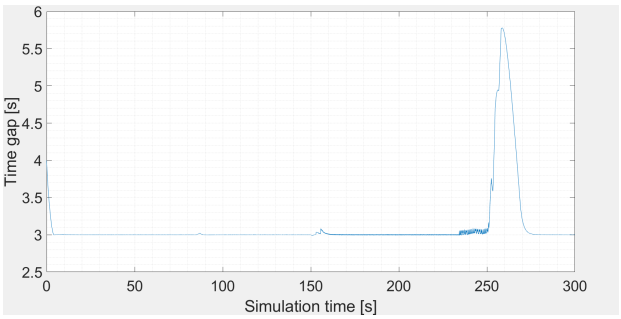


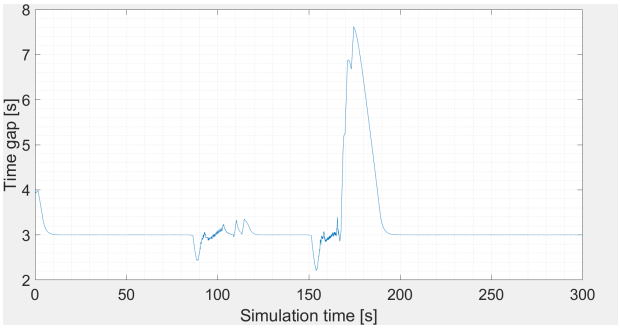
(a) Speed profile 1 (left), 2 (right)



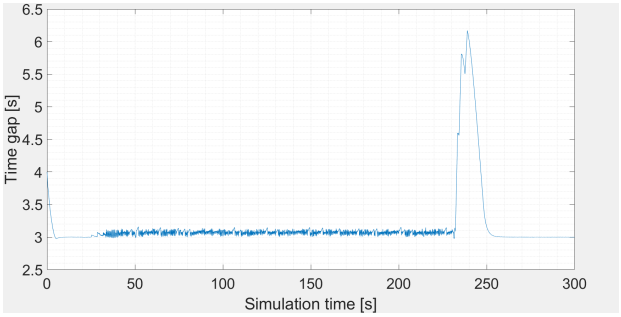


(b) Speed profile 3 (left), 4 (right)





(c) Speed profile 5 (left), 6 (right)



(d) Speed profile 7

Figure 8: Controller validation on different speed profiles

Looking at the speed profiles it's noticeable how in sharp speed profiles of the leading

vehicle, like the profiles 5, 6 and 7, the controlled Ego vehicle struggles to keep the desired time gap, due to the high required accelerations that the vehicle itself doesn't succeed in providing, due to limited ICE or braking system performances. By the way, no safety issues are reported: the time gap in all the profiles never falls below 2 s. Moreover, the tested profiles can be considered far more severe than a real application, showing accelerations level never reached by road vehicles. For this reason the validation of the minimum time gap can be considered even safer for the real application.

1.1.4 Controller effectiveness check

To verify the stability of the controller in different scenarios, some simulations are carried out changing the road conditions (considering a wet road) to evaluate if the parameters chosen for the dry situation can still be considered valid.

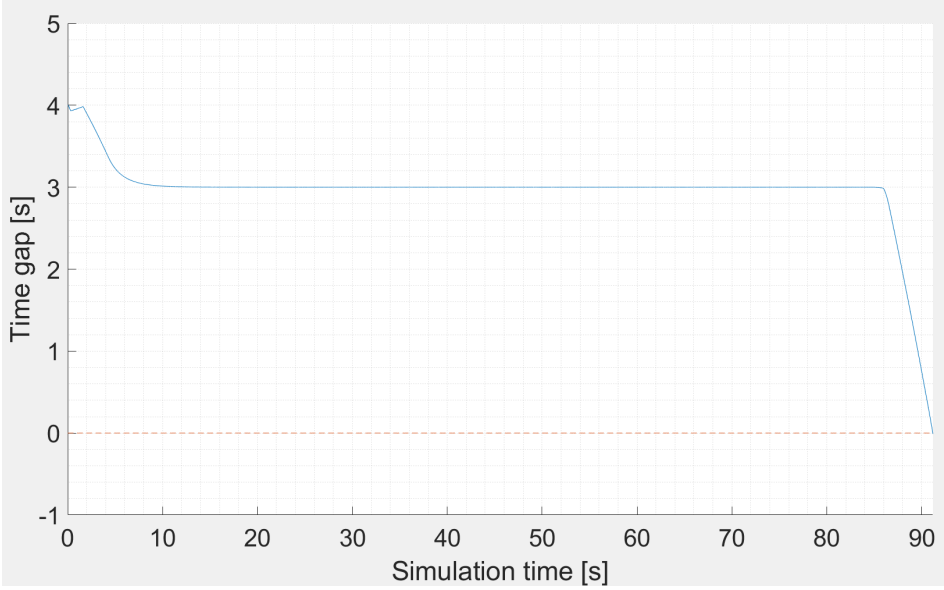


Figure 9: Time gap for profile 6 on wet road and h=3 s

The same parameters used for the speed profiles with dry road ($\lambda = 0.6$ and $h = 3$) are tested on wet road, speed profile 6. It is observed how in this case the system is not anymore able to guarantee a safe behaviour. Indeed the simulation is interrupted because the time gap reaches 0 (collision occurs between the vehicles); to determine the reason for this failure, the vehicle commands, as well as the actual and reference acceleration signals are reported below:

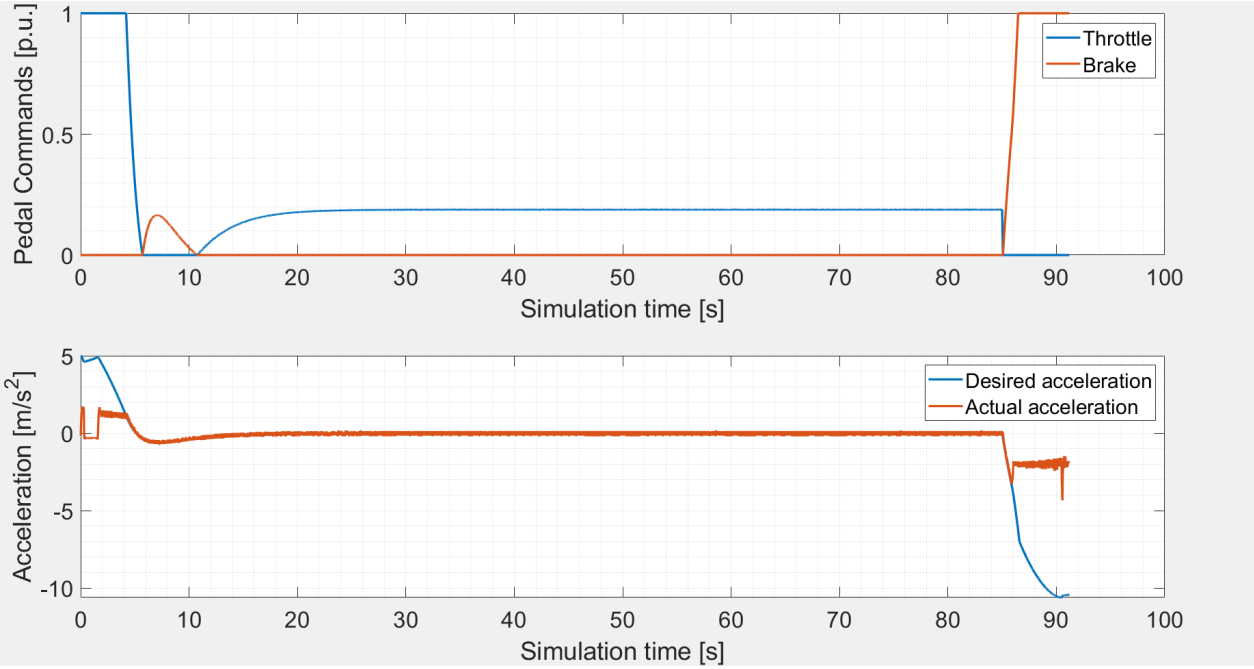


Figure 10: Command inputs analysis in wet conditions

The profiles show how the failure is not caused by the controllers functioning, which are correctly asking for 100% of braking action; by the way, the reduced grip level is limiting the vehicle deceleration capability, causing the accident. Therefore, a higher minimum h value, which can guarantee safe operation also in wet conditions, must be identified.
In the following figure, the value is progressively increased to find a suitable one.

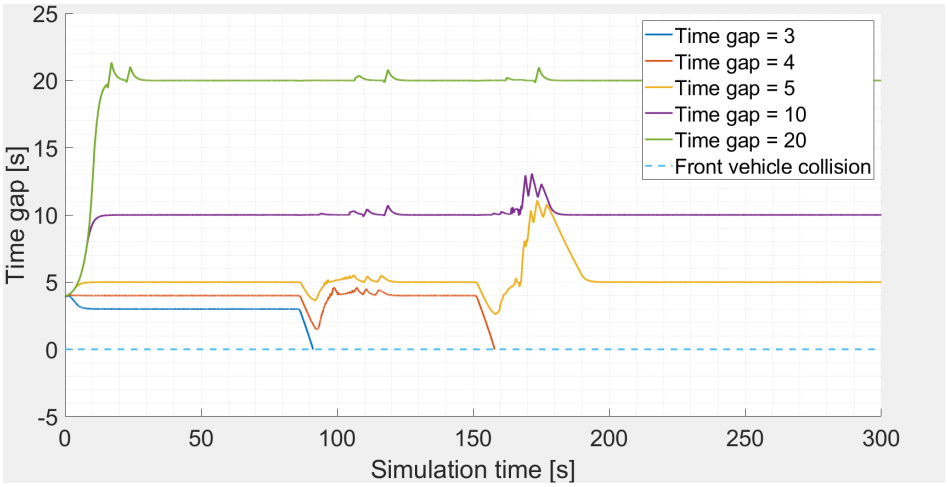


Figure 11: Time gap sensitivity in wet conditions

In this case it is shown how, to obtain a satisfactory behaviour, the minimum time gap must be increased up to 5 s. Two different approaches can be adopted to set the minimum

value for the time gap. The simplest case could be to take the one retrieved in wet conditions, knowing that it will be suitable also in a dry scenario. A more refined strategy, instead, could implement a model able to recognise the road conditions in real time and adapt the value to the actual situation.

1.2 Exercise 2: Vehicles platoon

A further situation of interest is represented by a vehicles platoon, whose vehicles (apart from the leading one) are embedded with the ACC system. Only the last vehicle dynamics is modelled in an accurate way, because it's the study object and to lessen the computational cost.

In this condition the implementation of this controller is crucial to determine the so called platoon stability, that would not be otherwise reached by implementing a different controller, such as the one based on the distance gap between two consecutive vehicles.

1.2.1 ACC implementation in a vehicles string architecture

The considered Simulink model features a difference with respect to the previous exercise: the Ego vehicle is now following a queue where a lead vehicle is imposing its speed profile and the following ones are equipped with the same ACC already developed. The general architecture is reported below:

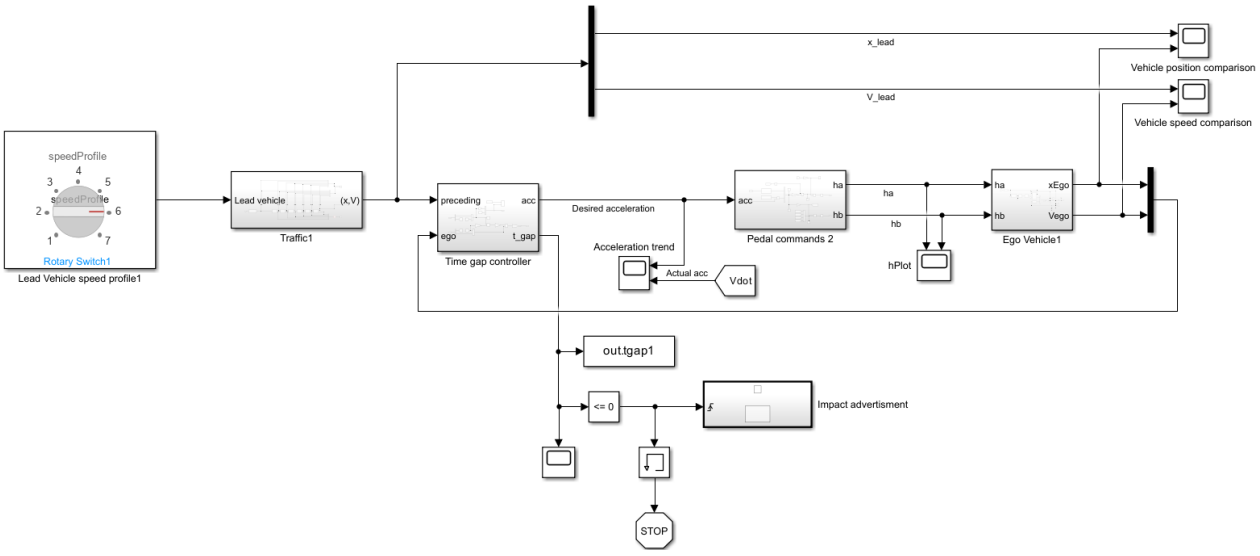


Figure 12: Vehicles platoon architecture

In the image the block 'Traffic' is representing the vehicles platoon and is implemented as follows:

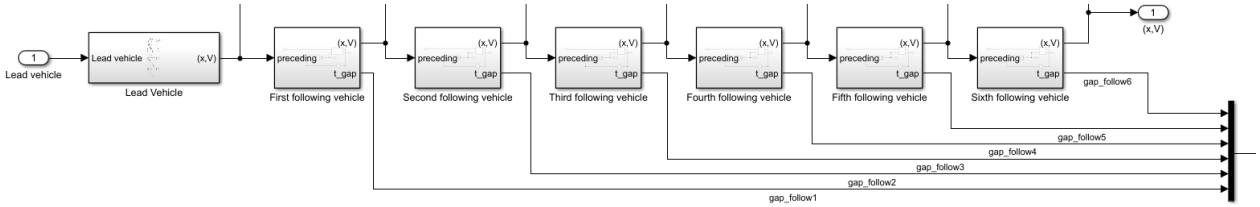


Figure 13: Traffic architecture

As already mentioned in Section 1.2, each follower vehicle is equipped with the same time gap upper level controller of the Ego vehicle, while the lower level one and the longitudinal dynamics are simply modeled by the following first order transfer function:

$$\frac{1}{1+0.5s}$$

representing the dynamic response of the vehicle and contained in the 'Simplified vehicle dynamics' block of the following figure.

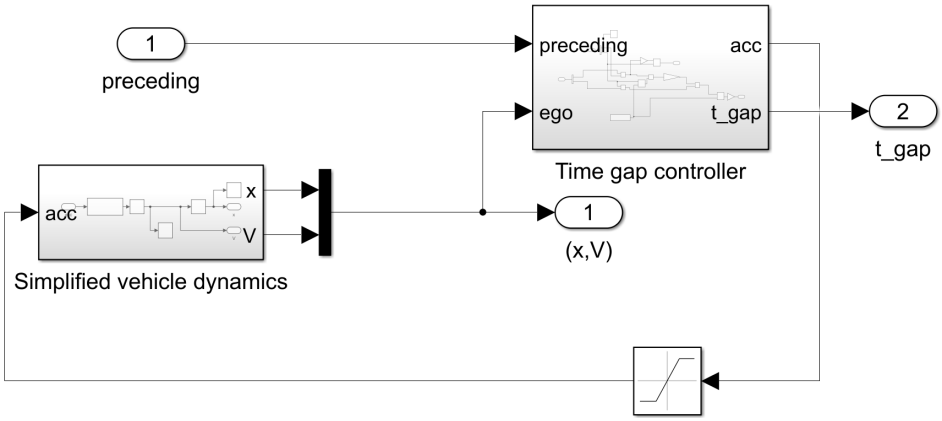


Figure 14: Follower vehicle architecture

The effect of the presence of the queue between the leader and Ego vehicle with respect to the simple case studied in Exercise 1 is shown in the following speed profiles of Ego vehicle, considering the same mission for the leader:

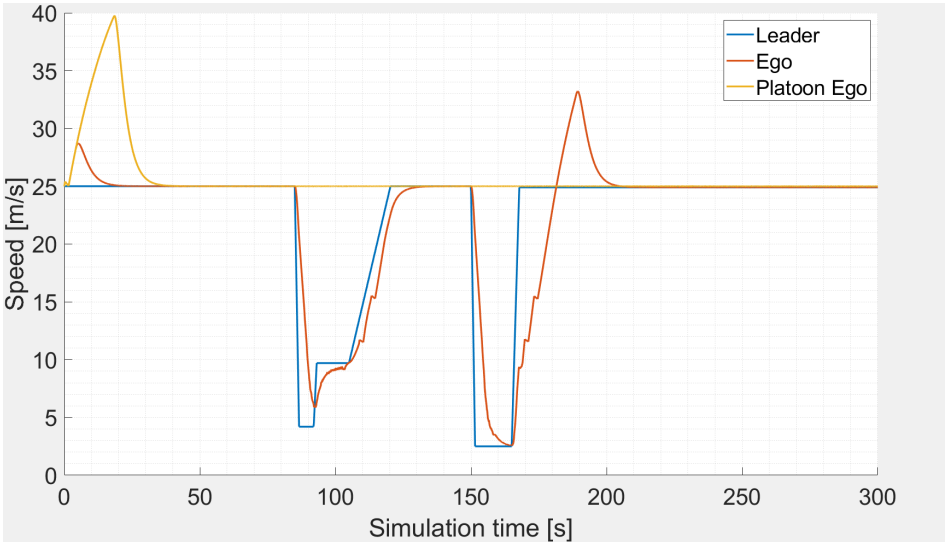


Figure 15: Profile 6 queue effect

The explanation of this particular effect is carried out in the Section 1.2.3.

1.2.2 Controller overview over different driving missions

Analyzing a string of vehicles, with the leader one attending a certain driving mission and the follower ones controlled by an ACC system, the platoon stability is achieved when the spacing error, computed between a certain vehicle and the preceding one, going progressively farther from the leader, remains equal or decreases with respect to the one computed between the two vehicles ahead of the considered.
A vehicles string, from the theory, is stable if the following conditions are verified:

1. $\|G(s)\|_\infty \leq 1$;
2. $sign\left[g(t)\right] = sign\left[g(t+dt)\right]; \quad \forall t \geq 0.$

Where:

- $G(s)$ is the transfer function defined as: $G(s) = \frac{\delta_i(s)}{\delta_{i-1}(s)}$;
- δ is the spacing error of the ego vehicle with respect to the desired position (as explained in Eq.1);
- $g(t)$ is the $G(s)$ impulse response.

The kind of ACC system embedded in the follower vehicles is based on a time gap control instead of an inter-vehicle spacing control because, in this way, platoon stability is ensured.

Exploiting the driving mission 1, the steady-state spacing error trend of the vehicles string is presented to give evidence of this phenomenon:

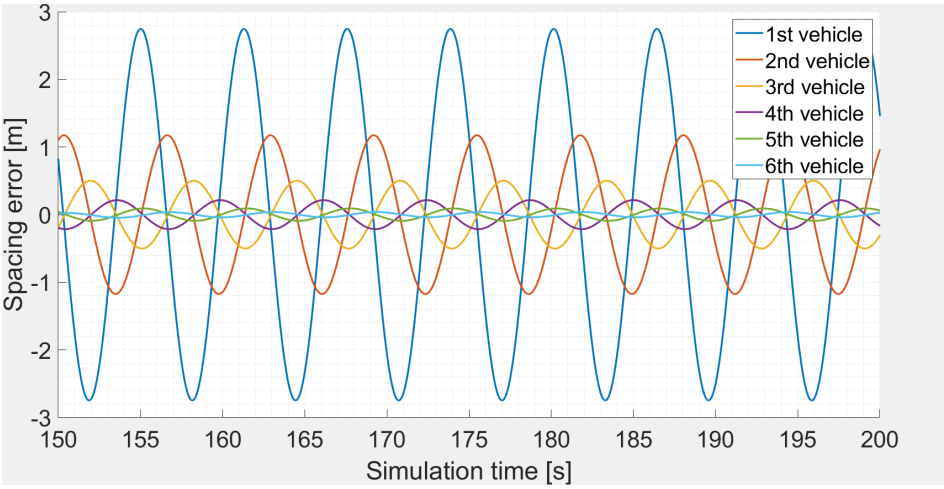


Figure 16: Platoon stability evidence

To understand the vehicles string behaviour, different driving missions are tested. Starting from the driving mission 3, studying the time gap plot, plateaus are visible in the first part of the simulation, when the vehicles are still trying to close the time gap from the preceding ones. Indeed their initial conditions impose higher distances with respect to the desired one. Since all the vehicles have a limited maximum speed, at first they all saturate at this value, so the time gap remains constant. Each vehicle can close its gap only when the preceding one starts to slow down, having reached its target.

The time gap plot is shown as follows:

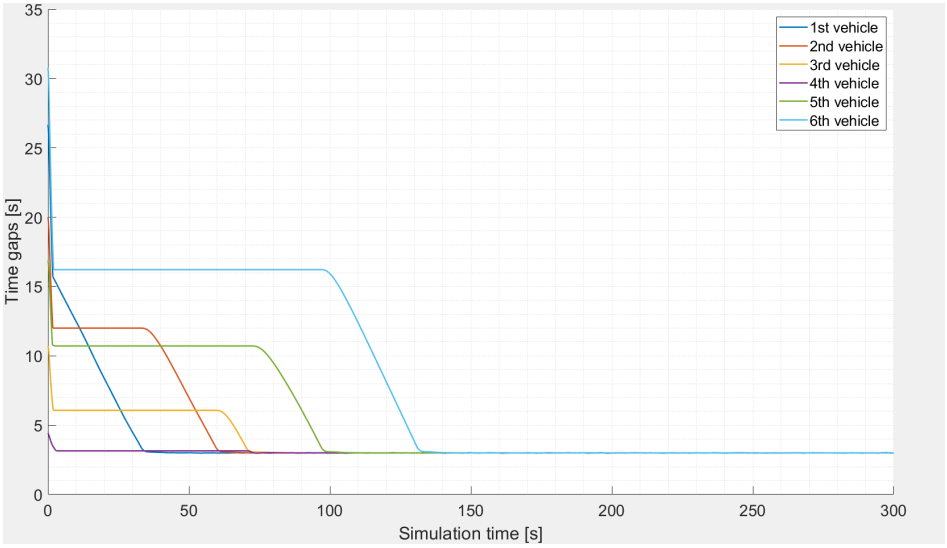


Figure 17: Profile 3 time gaps

A further evidence of this behaviour is highlighted by the speeds profile where, in the plateau regions, the vehicles speeds correspond to their upper limit ($25m/s$):

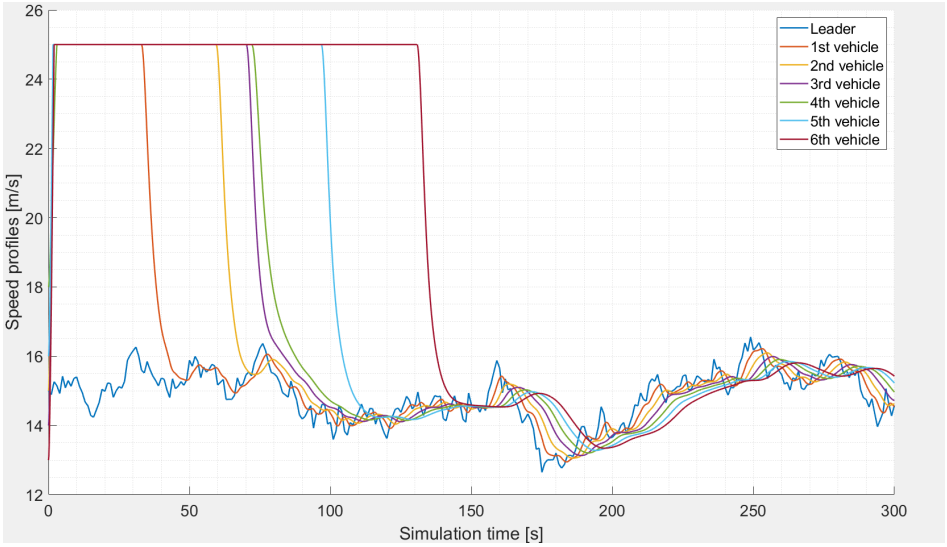


Figure 18: Profile 3 vehicles speed

Another interesting behaviour is observed in the mission 1, where the link between spacing error and time gaps is noticeable, that is the speed. Thus the platoon stability effect is evident in the time gaps trend as well. Indeed fluctuations for the first vehicle time gap are more evident then for the following ones, as it is represented in the following plot:

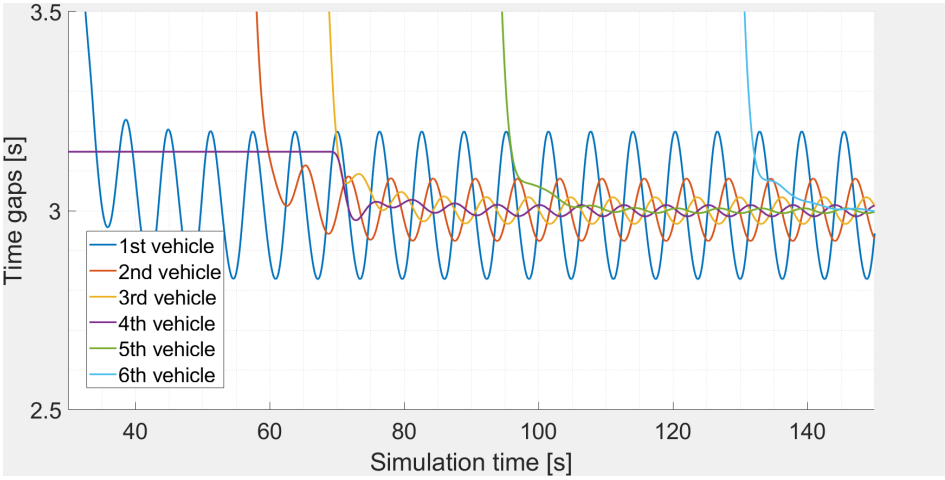


Figure 19: Mission 1 time gaps

Moving to mission number 5, where one sudden acceleration and deceleration are present (assimilable to step inputs), sine-like fluctuations can be observed; this can be attributed to the PD (proportional derivative) nature of the controller: this causes a dynamic response featuring damped oscillations, with an initial overshoot and a progressive attenuation of the amplitude.

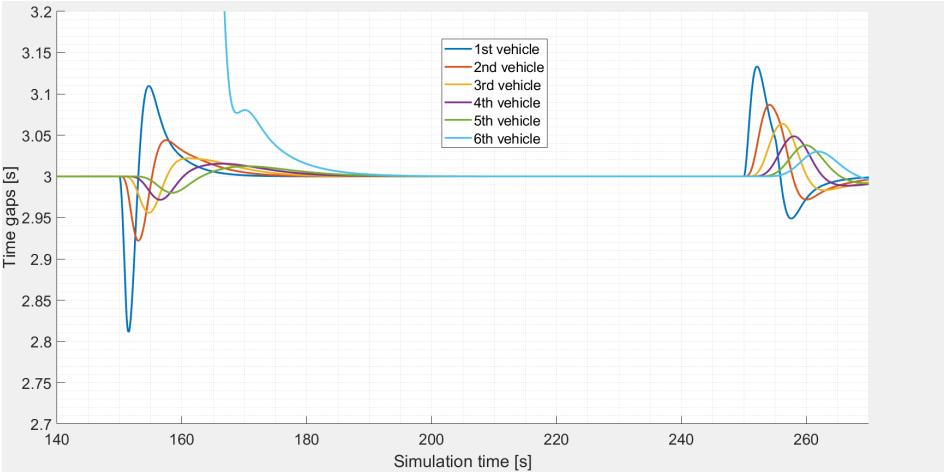


Figure 20: Mission 5 sine-like fluctuations

Analysing the system behaviour in the driving mission 6, the first follower vehicle impacts the leading one, because the imposed lower speed limit doesn't allow it to slow down enough in order to avoid the collision. Looking at the first follower vehicle time gap, a value below 0 is reached standing for the collision event:

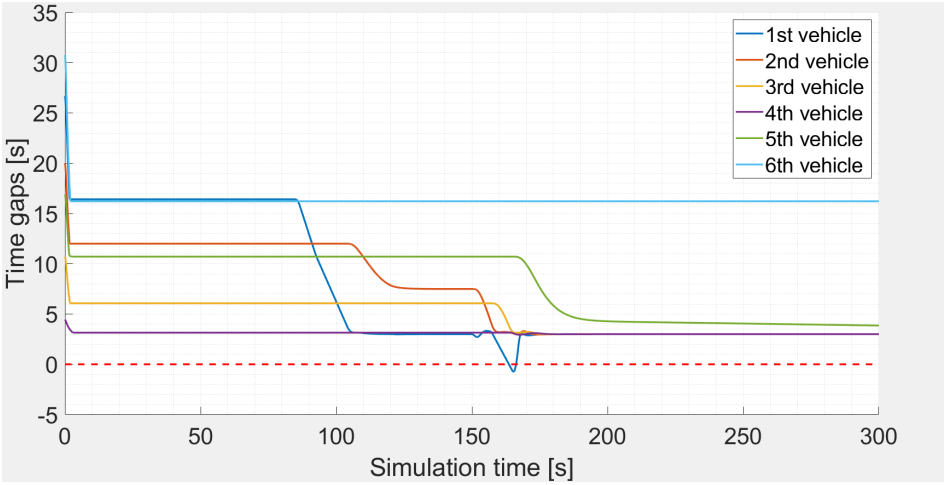


Figure 21: Time gap for profile 6 with vehicle 1 collision

Further considerations about the system behaviour over this mission are undertaken in the following Section 1.2.3.

The behaviour of the platoon for speed profile 7 is shown below. As anticipated for profile 3, there is an initial phase in which the vehicles reach the saturation speed and cannot close the time gap; after that, the speeds of each vehicle stabilize to a consensual value, showing a small delay in the transient response which follows the second step variation.

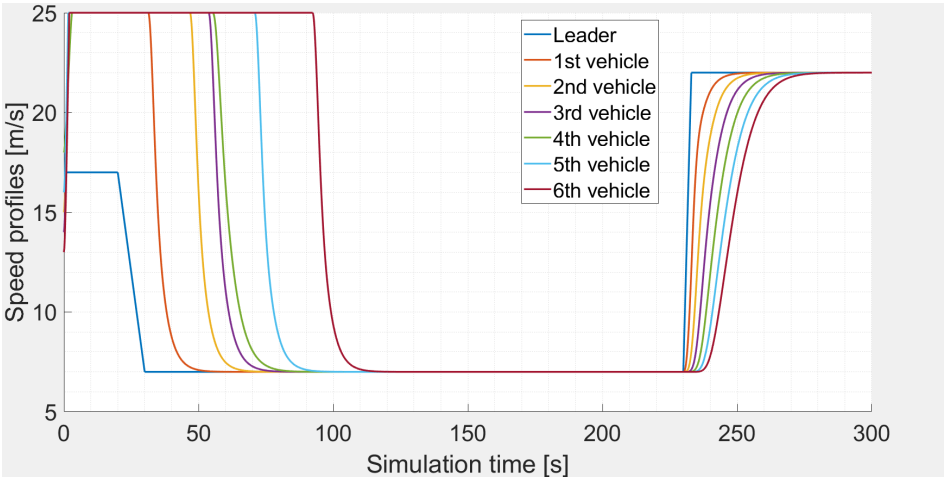


Figure 22: Speed evolution of the vehicles platoon for profile 7

The above mentioned behaviour is confirmed by the time gap graph. The transient response follows the same trend explained for mission 5, with a damped oscillating evolution.

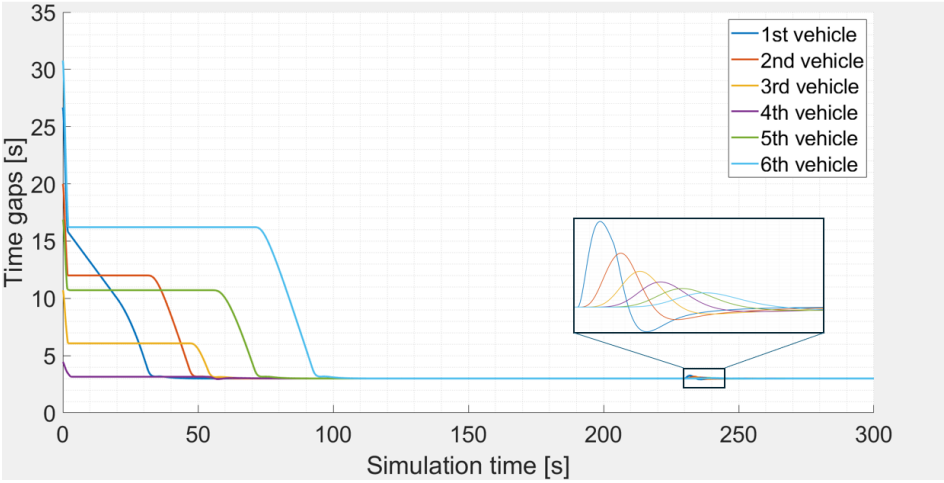


Figure 23: Time gap evolution of the vehicles platoon for profile 7

1.2.3 ACC implications

Going on with the analysis of the system behaviour, when undertaking the driving mission 6, interesting considerations about the Ego vehicle are carried out:

- its initial position, with respect to the preceding car, implies an initial time gap to be recovered, but this gap is too large, indeed the speed of the vehicle saturates to its maximum value, but never succeeding in reaching the target time gap, as shown in Fig. 24. This limit may be overcome by relaxing the constraints on the vehicle speed, exploiting its maximum performance;

- moving progressively farther from the leader, the vehicle speed profiles tend less to emulate the leader one; this is a result of the string stability, that plays a sort of filtering effect on the vehicles speed. This behaviour is clearly shown in the following picture:

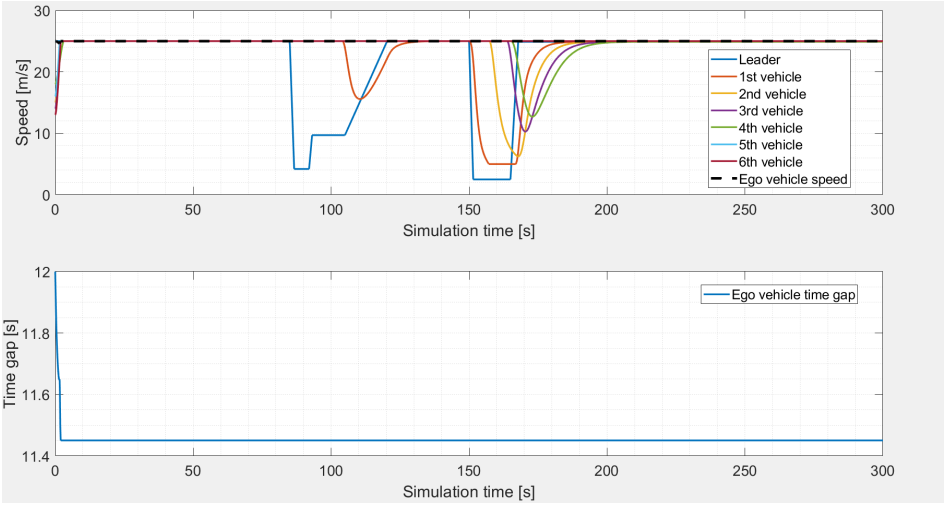


Figure 24: String speeds profile and Ego vehicle time gap

1.2.4 Traffic conditions sensitivity

As anticipated in the previous Section, the platoon stability plays a filtering effect on the speed profiles; thus, decreasing the number of vehicles within the string implies that the Ego vehicle tends to emulate more accurately the leader speed profile.
Each time that a car is removed from the string, the position initial condition of the Ego vehicle is shifted and put equal to the removed vehicle one, in order to have coherence in the cars relative positioning. For what concerns the speed values, the initial one v_0 is kept equal to $25m/s$ in all the situations, while the upper and lower boundaries are set, for the first analysis, respectively to 25 and $5m/s$.
This above explained trend is verified showing the Ego speed profiles for different queuing lengths, where the follower vehicles number, moving between the leader and the Ego, is progressively reduced from 6 to 0:

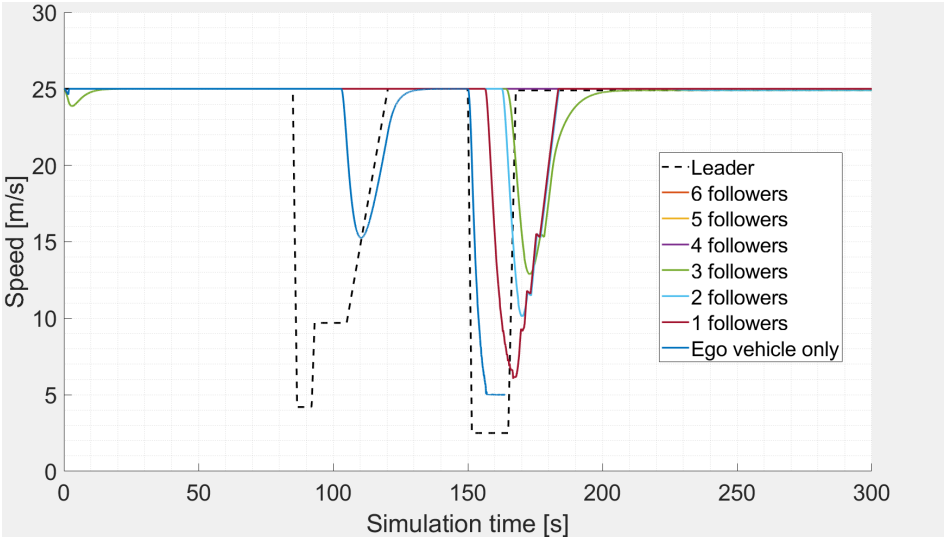


Figure 25: Profile 6 speed profiles with progressively reduced queue

Particular attention must be given to the case in which Ego vehicle is directly following the leader: in this case, due to saturation applied on the lower constraint speed, it crashes into the preceding one.
Moreover the time gaps trend is shown:

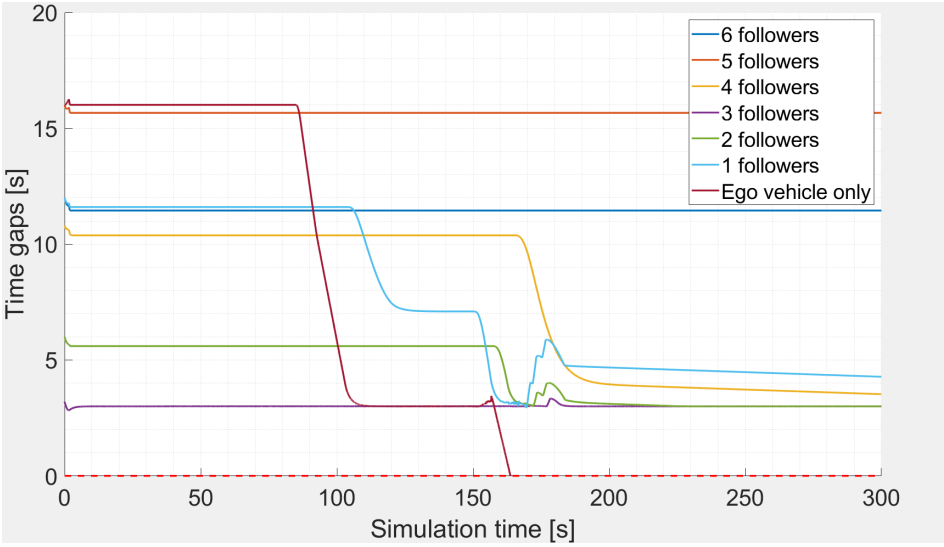


Figure 26: Profile 6 time gaps for with progressively reduced queue

1.2.5 Ego vehicle speed constraints sensitivity

As anticipated in Section 1.2.3, releasing the constraints on the vehicles speed stands for a better exploitation of the controller performances. Indeed they are now set to $v_{max} = 50m/s$

and $v_{min} = 2m/s$. These values are chosen to allow the Ego vehicle to close the time gap, even when testing the controller with the more challenging leader speed profiles. In fact it is possible to evaluate the performances obtained by the system, by analysing speed profiles for mission 6:

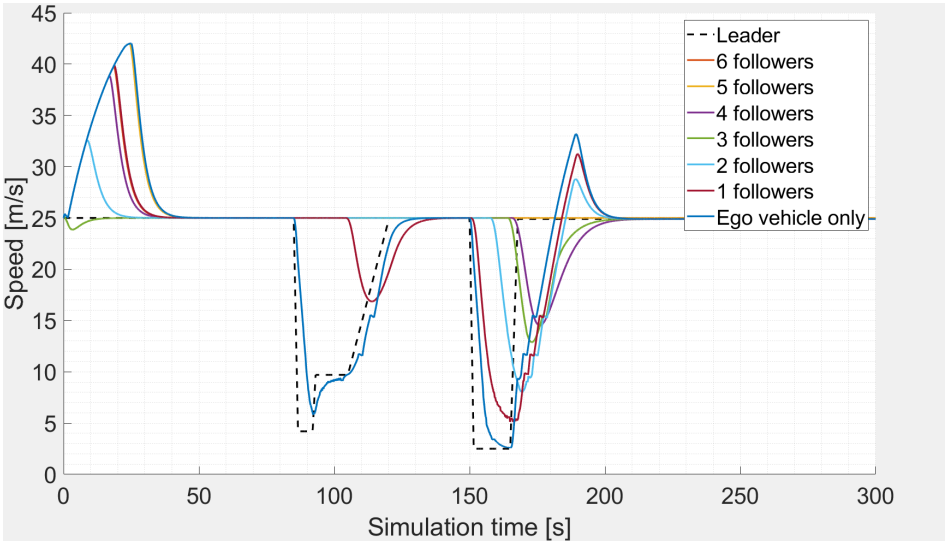


Figure 27: Profile 6 speed profiles with progressively reduced queue and released speed constraints

It is noticeable how the Ego vehicle, also when directly following the leader, doesn't crash into it.

Moreover, it succeeds in maintaining the desired time gap in every situation, because not anymore limited in terms of performances. To better understand the situation, the time gap plot is reported here:

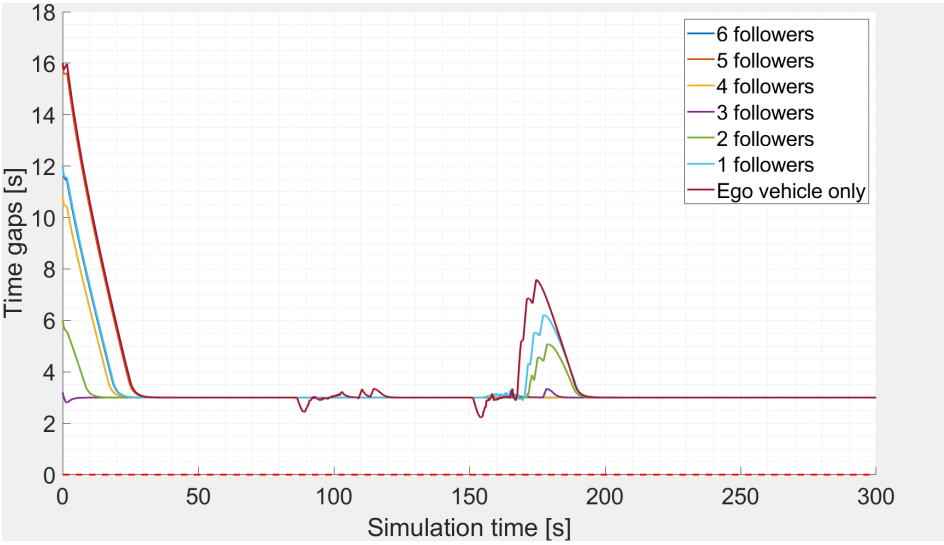


Figure 28: Profile 6 time gaps with progressively reduced queue and released speed constraints

With respect to the time gaps profiles, shown in Fig. 26, the plateaus visible in the first part are not anymore present because, as previewed in Section 1.2.2, relaxing the constraints the time gap decreases quicker until reaching the target value.

1.3 Project 1 results

To test a different lower level controller, a more basic approach is adopted, where the longitudinal force requested is used only to evaluate if the vehicle is requiring acceleration or braking; then the pedal commands h_a and h_b are respectively set equal to their max and min value in the first condition while, in the second one, their settings is the opposite. The lower lever controller architecture is reported here:

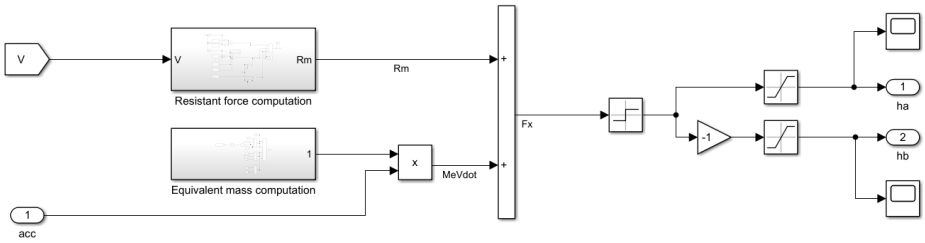


Figure 29: Basic lower controller architecture

The longitudinal required force is computed as in the more accurate controller and its sign is taken as reference to compute the commands. The saturation blocks (whose upper

and lower boundaries are set to 1 and 0) are placed to avoid that the commands assume the same value contemporarily.

1.3.1 Different lower level controllers comparison

The different behaviour for the two implemented controllers is compared, taking as reference the case in which the Ego vehicle is directly following the leader one in the most critical mission 6. It is noteworthy to highlight the pedal commands profile in both cases:

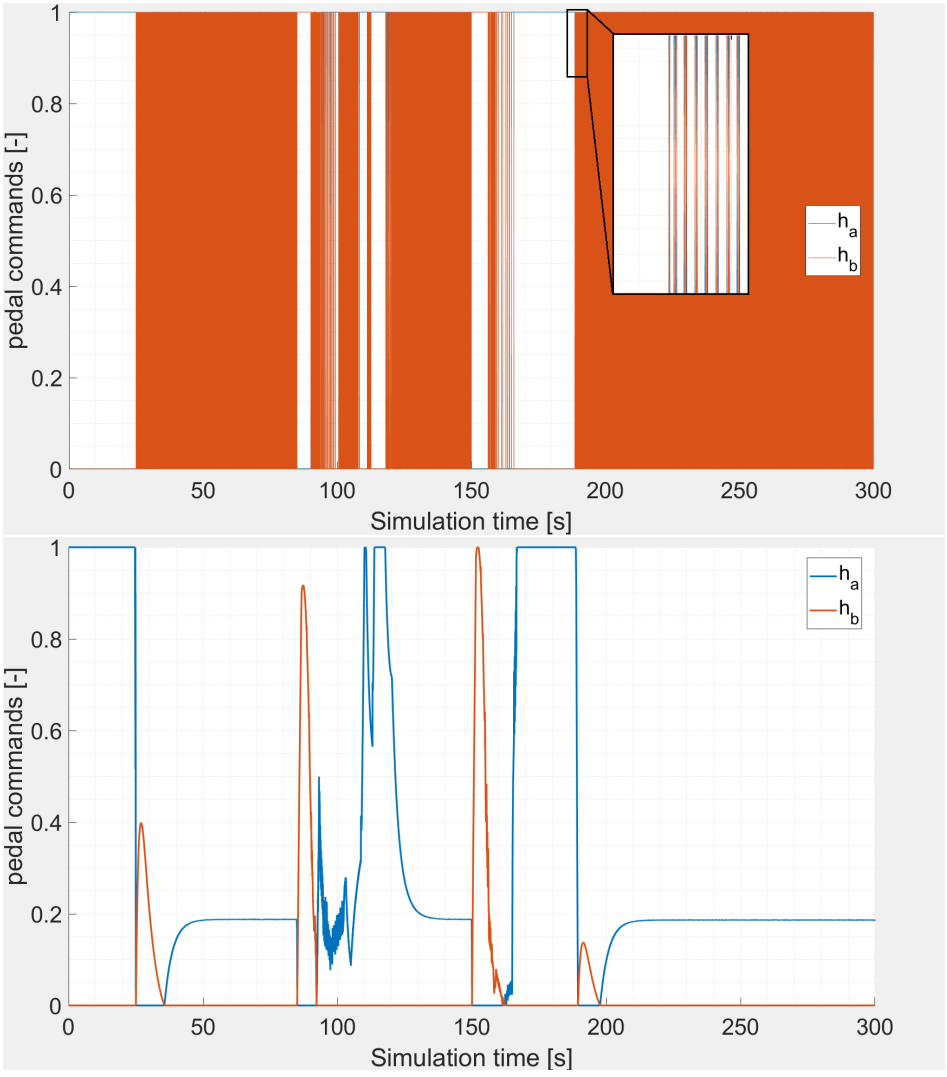


Figure 30: Pedal commands profiles for basic (up) and advanced (down) controllers

Regarding the basic controller, from the zoom it's noticeable how it produces constant commands equal to 1 or 0 when maximum acceleration or braking maneuvers are requested. In all the other cases, when the advanced controller generates intermediate commands, the

basic one is able only to switch continuously between max and min value of h_a and h_b. The speed profiles are reported and the drawbacks, related to the basic controller functioning logics, are clearly visible:

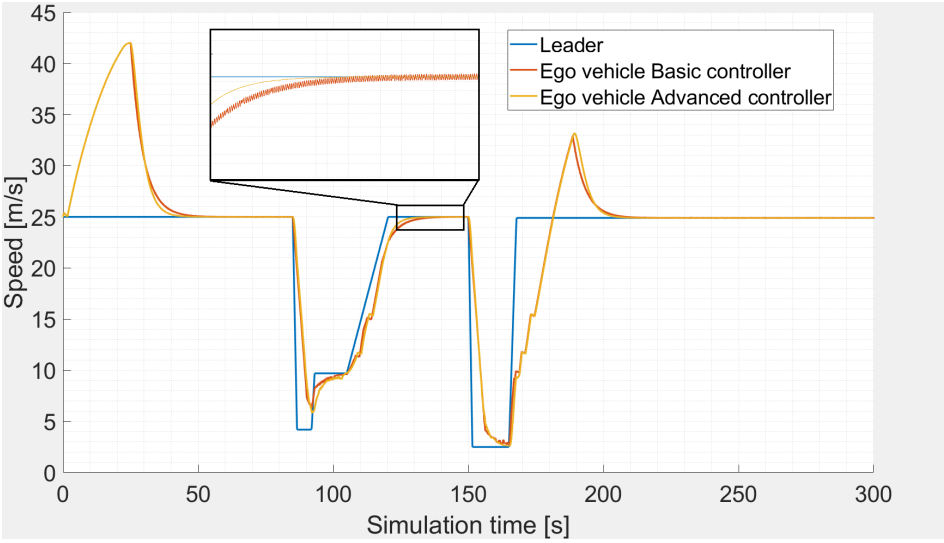


Figure 31: Speed profile comparison between the controllers

In general, both the controllers are able to maintain the target time gap similarly, but the behaviour of the commands in intermediate acceleration/deceleration zones generates some ripple, resulting in a more noisy and less smooth trend.
Moreover, the effectiveness of the two implemented logic is proved by the time gap plot shown here:

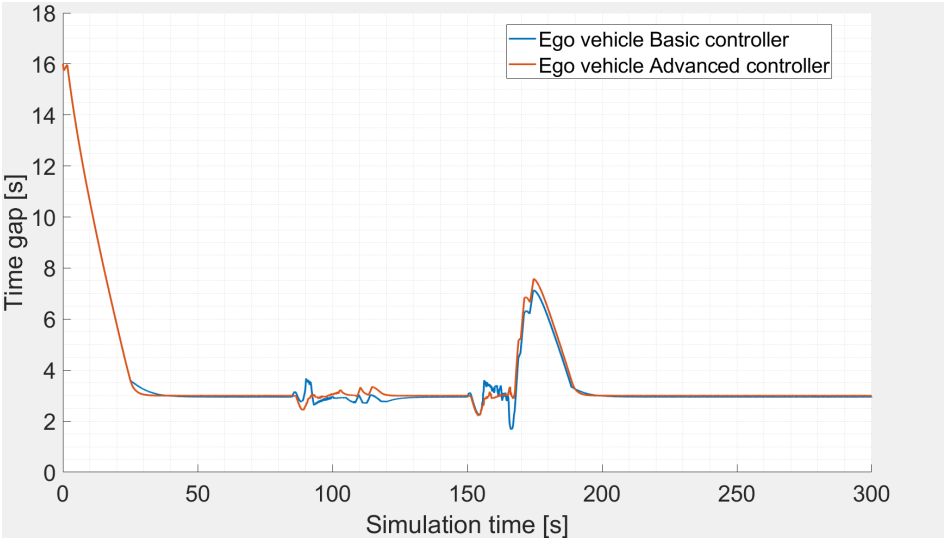


Figure 32: Time gap comparison between the two controllers

The convergence speed is basically the same in the two cases and, except for some short transient periods, they reach the desired time gap.

1.3.2 Metrics for controller performances evaluation

To evaluate the performances of the two controllers, the fast Fourier transform 'FFT' is applied on the obtained speed profile in the two cases: this is done to visualize the different harmonic contents of the two signals:

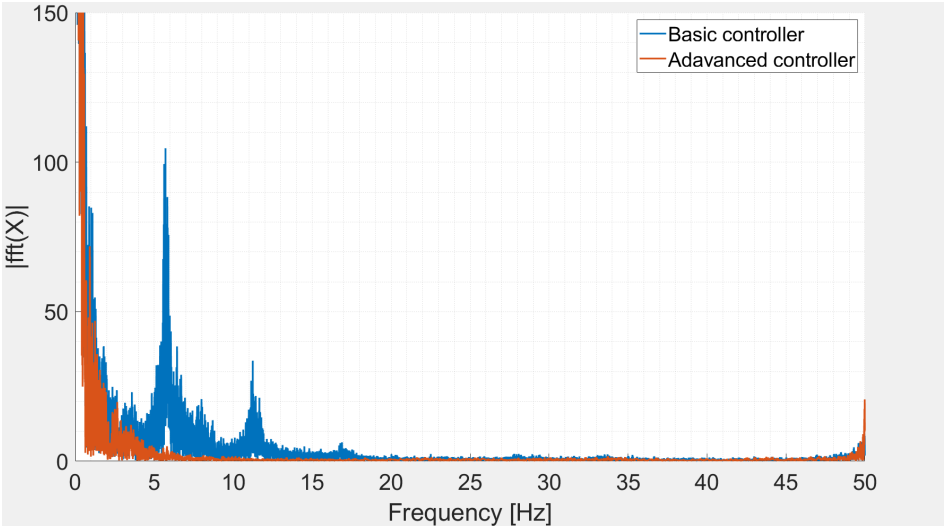


Figure 33: Ego speed profile FFT for the two controllers

For what concerns the basic controller signal, two additional peaks are visible with respect to the other case, in particular for frequencies above 5 Hz. This can be considered an objective indicator to evidence the worse behaviour of this controller.

If a more detailed model for the ego vehicle was available, the human sensitivity function 'ISO2631' could be applied on the output acceleration signal and its filtered 'rms' value would be a relevant parameter to represent the comfort index for the vehicle occupants. However, this procedure would not be representative considering the adopted model, since its level of sophistication is not suitable to represent the range of frequencies related to human sensitivity.

1.3.3 Lower level controller choice

Between the two possible controller analysed so far, the chosen one is the advanced one, based on a more detailed model of the vehicle to retrieve the output commands. Indeed it guarantees a smoother ride, especially during constant speed or moderate acceleration phases:

the basic controller is clearly not implementable due to severe drivability issues linked to acceleration ripples as can be seen in Fig. 30 and 31. In addition, this logic wouldn't even be representative of a real system behaviour, where the pedal commands application would always be characterized by a smoother transition rather than the step command resulting from the basic implementation.

2 Project 2: Lane Keeping Assist (LKA) design

During long highway trips, the risk to dangerously loose attention or to fall asleep is consistent. In such cases, an embedded control system for the lane keeping may be crucial to ensure the passengers' safety. In the following section the procedure to implement and validate this kind of controller is shown.
The test vehicle for this study is the same of the previous project.

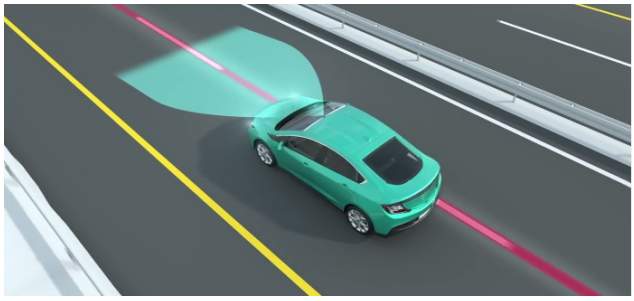


Figure 34: Lane keeping

2.1 Exercise 1: Controller implementation

Two different controller architectures are studied: the PIDF general control logic for linear systems and the specific control logic for lane keeping that is the Stanley controller. The simulations are carried out on a simplified 3 d.o.f. Segel model, based on the following assumptions:

- the longitudinal and lateral motion of the suspension as well as the yaw and pitch motion of the suspension are rigidly linked to the motion of the vehicle body;
- reference is made to a certain configuration of the vehicle. It may be the static equilibrium condition with the vehicle at standstill or travelling at a constant speed;
- the kinematics of suspensions is linearized around the reference position;
- pitch and roll angles are small enough to linearize their trigonometric functions;
- also, the displacements in z direction and all linear and angular velocities, with the exception of the forward speed and the rotation speed of wheels, are considered as small quantities;
- the vertical plane xz through the center of mass is a symmetry plane for the vehicle and its parts;

- sideslip angles of the wheels are small enough to linearize the cornering forces and the aligning torque.
- the dynamics of the sprung and unsprung mass are decoupled, due to the different orders of magnitude of their natural frequencies.

Therefore the state variables of the model are the vehicle sideslip angle β, the yaw angle ψ and the roll angle ϕ.

2.1.1 PIDF controller implementation

In the following section, a control strategy based on two PIDF controllers is implemented; these controllers separately act on the cross track error (e_{ct}) and heading error(e_h), defined as:

$$e_{ct} = (Y_r - Y_a) \cdot cos(\Psi_r) - (X_r - X_a) \cdot sin(\Psi_r)$$

$$e_h = \Psi_r - \Psi_a$$

where subscript r is referred to reference quantities, while subscript a to actual vehicle ones. They represent respectively the displacement from the center of the vehicle front axle to the closest point of the trajectory and the error between the vehicle longitudinal axis and the reference direction. The control scheme is shown below:

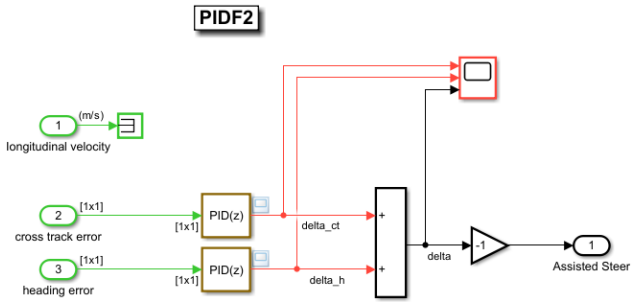


Figure 35: PIDF architecture

To retrieve starting values for the two PIDF controllers parameters, a simplified DST model is separately developed in order to allow the Simulink PID tuner linearization procedure, obtaining satisfactory preliminary results.
To validate these values a reference track is generated and the vehicle with the LKA embedded is tested on it. A trial and error iterative procedure leads to the selection of suitable parameters. The tuning is performed on a vehicle speed of 22.22 m/s (80 km/h) and on a

reference trajectory composed by a starting straightway, two bends of radius $250m$, one to the left and one to the right, and an arrival straightway; its representation follows:

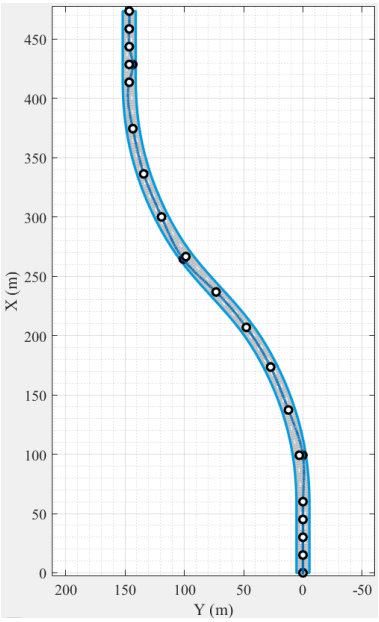


Figure 36: Test bench trajectory with double curve of 250m radius

The blue line depicted within the track represents the trajectory that driver would follow when the LKA is not engaged. The trajectory is meant to interpolate set waypoints, implemented to introduce a disturbance, with respect to the lane center, to test the controller operation.
The final tuned values are reported in the following table, where the PIDF1 is the controller of the cross-track error and PIDF2 of the heading error:

10^{-3} x	$PIDF_1$	$PIDF_2$
K_p	43.63	349.98
K_i	1.00	1.11
K_d	207.74	166.10
K_f	1.25	33770

Table 1: PIDs tuned values

The Simulink Bird's Eye View is a useful tool to perform a first evaluation of the controller behaviour from a graphical point of view, showing the real driving maneuver that a vehicle, embedded with the controller, would undergo on the previously introduced driving scenario. A picture of the analysis setup is shown:

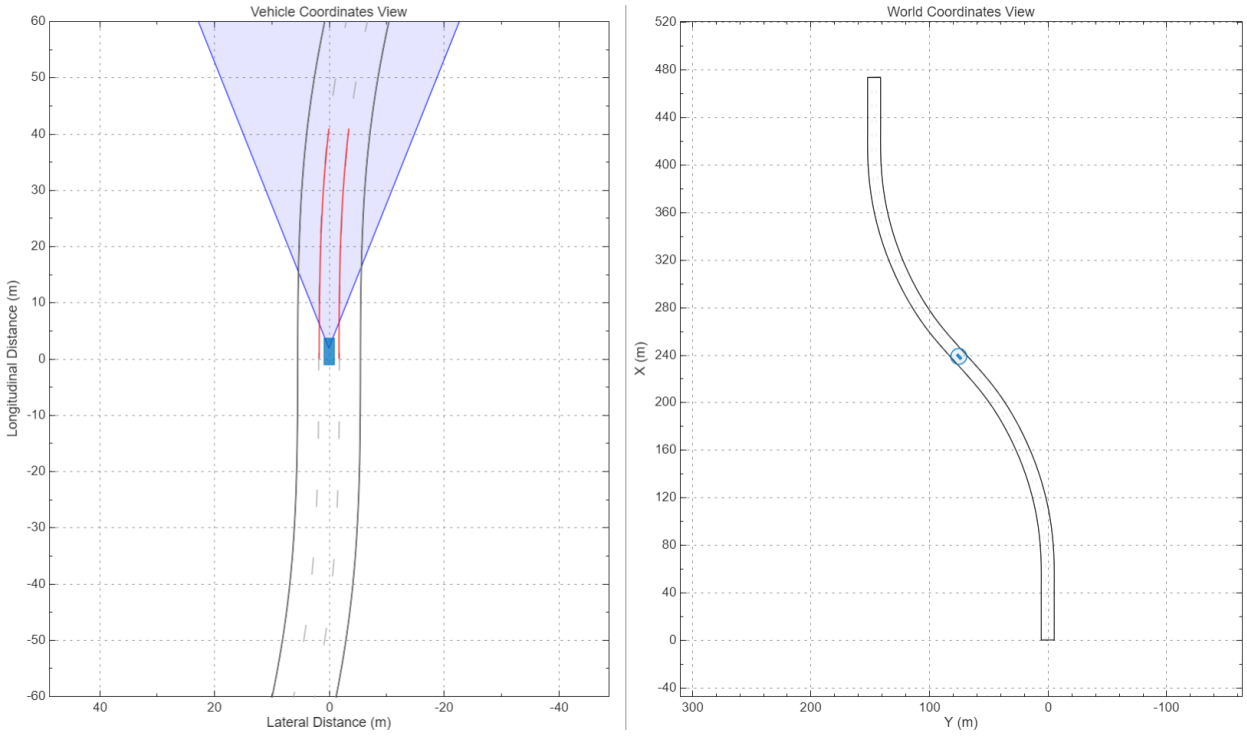


Figure 37: Bird's eye view on the test bench trajectory

A first consideration is that the vehicle is always able to stay within the initial central lane, as a result, the cross-track error never saturates to its maximum value of $0.5m$, as well as for the heading error. The obtained trends are considered satisfactory, validating the overall computed parameters, as shown as follows:

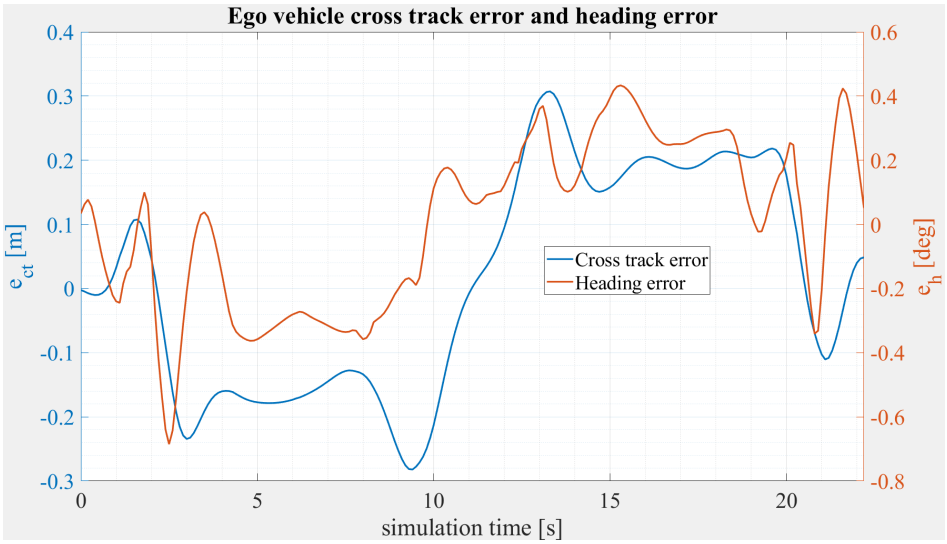


Figure 38: Cross-track and heading errors for the test bench simulation - PIDF

Safe distance effects

In this section an analysis of the effect of the variation of the safe distance is performed, in terms of lateral acceleration rms value tracking. This safety parameter represents the minimum distance between the vehicle COG and the lane boundaries under which the LKA activates. Different simulations are carried out progressively decreasing this value and observing the trend of $a_{y,rms}$:

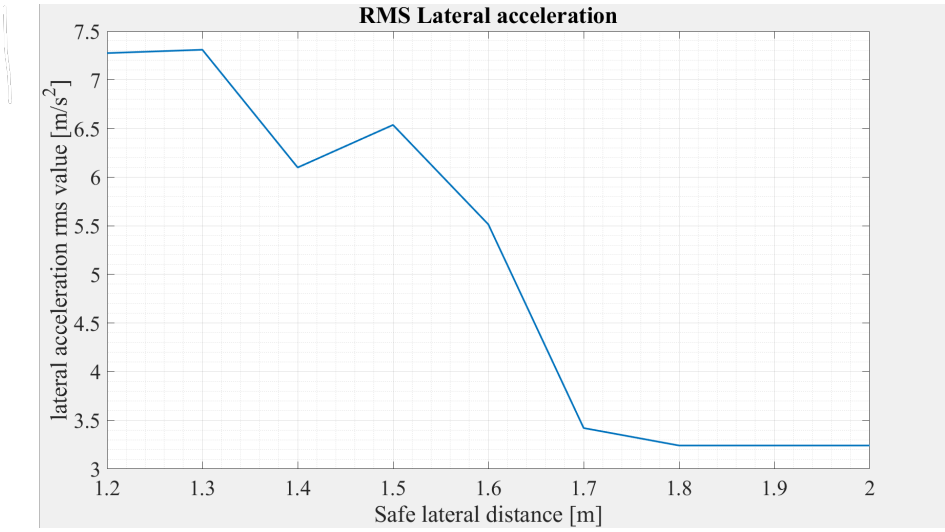


Figure 39: RMS lateral acceleration trend for varying safe lateral distances

Regarding these values, some considerations must be carried on:

- No difference is observed for values above $1.8m$. This is motivated since, being the lane width equal to $3.6m$, having a safe distance of $1.8m$ already means that the LKA is always active; thus, a further increase of the value doesn't provide any change;
- The values of vehicle lateral acceleration are not always realistic, since the Segel model is based on the assumption of linear tyre behaviour, which is not anymore valid for high values of sideslip angles, resulting from a high steering angle demand from the controller; this leads to values of a_y that a real vehicle could not reach;
- the values of $a_{y,rms}$ are strongly affected by the driver's behaviour when the system is not engaged: in this case a steep increase is observed as soon as the safe value is relaxed, mainly because the driver is specifically conceived to provide aggressive inputs to the vehicle, while a real driver is expected to provide smoother commands in normal driving conditions.

For these reasons, and being the LKA developed mainly for safety reasons rather than autonomous driving, the value of $a_{y,rms}$ is not the main indicator of the controller effectiveness, as far as the vehicle remains inside the lane; for this reason, a possible choice for this parameter is $1.7m$, letting the driver a range of manual driving of $20cm$, in case the lane is $3.6m$ wide, maintaining at the same time the system functionality. The errors retrieved in the simulation with safe distance equal to $1.7m$ are shown:

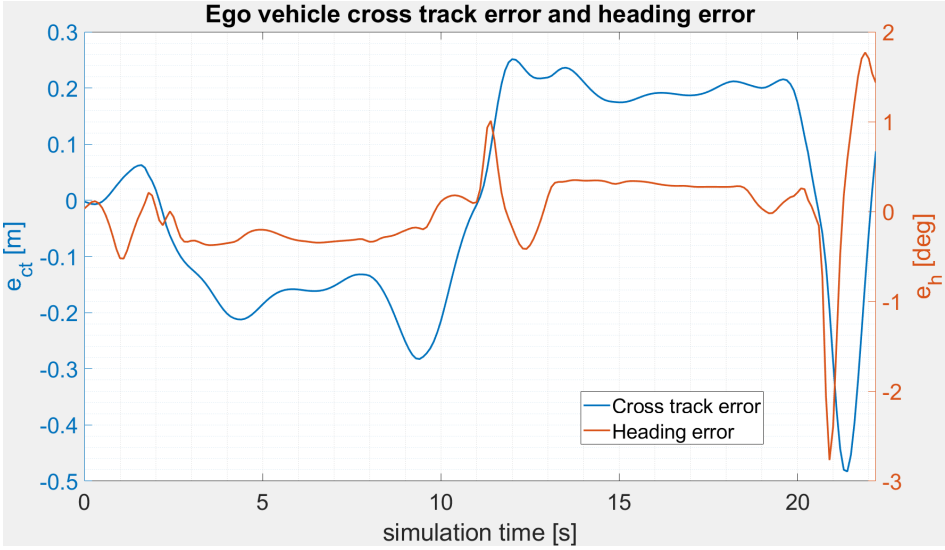


Figure 40: Cross-track and heading errors for the test bench simulation with safe distance = 1.7m

For reference, being the lane width equal to $3.6m$ and the vehicle largest track to $1.578m$, the condition of lane departure corresponds to a value of cross track error $\simeq \pm 1m$; in this case, the controller is able to almost entirely maintain the vehicle within the lane.

2.1.2 Stanley controller implementation

The second controller typology to be implemented is the Stanley, whose architecture is reported here:

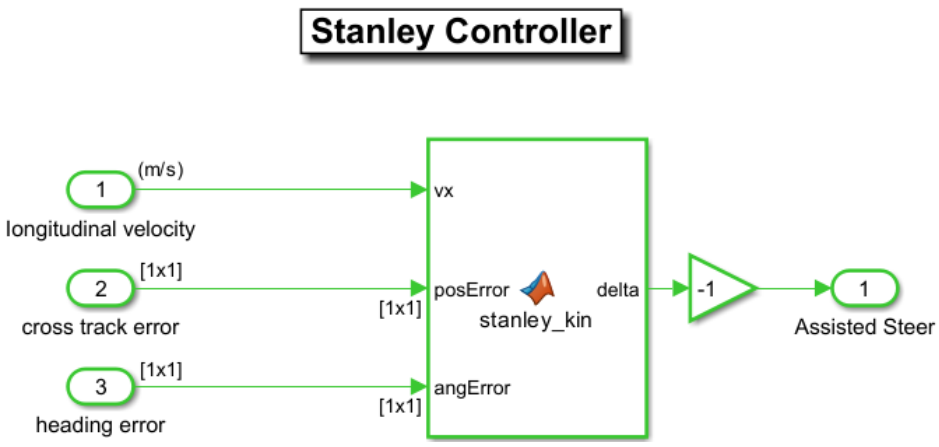


Figure 41: Stanley controller architecture

The equation on which the Stanley controller is based on is the following:

$$\delta_0 = e_h + atan(\frac{K_s \cdot e_{ct}}{v_b + v_x})$$

Where $K_s = 0.11$ is the tuned gain and $v_b = 1$ is a small parameter introduced to avoid numerical errors. The controlled value of the steering angle is saturated according to given thresholds.

As done for the previous controller, the characteristic parameters validation is carried out by means of the Simulink Bird's Eye View simulation (Fig.37) on the test bench and of the errors tracking during the cycle. Analog considerations for the driving simulation can be done: the vehicle performs the lane holding over the whole driving mission. For what concerns the cross-track and the heading errors, the resulting trends are reported:

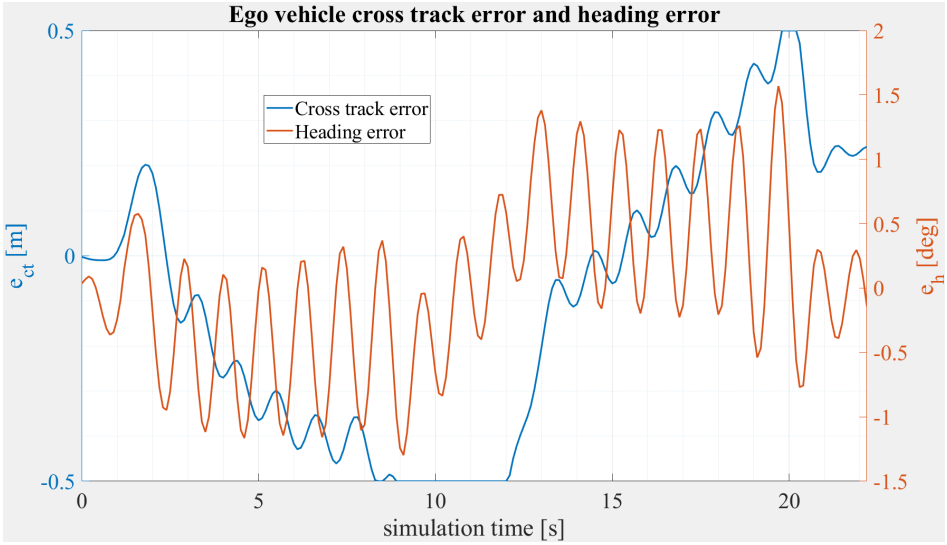


Figure 42: Cross-track and heading errors for the test bench simulation - Stanley

Comparing these results to the ones obtained with the PIDF controller, it is noticeable how the Stanley is less performant providing higher maximum errors and in general a more oscillatory behaviour.

Safe distance effects

As already done for the PIDF, an analysis of the effect of the safe distance value is undertaken for the Stanley controller as well, observing the variation of the rms value of the vehicle lateral acceleration:

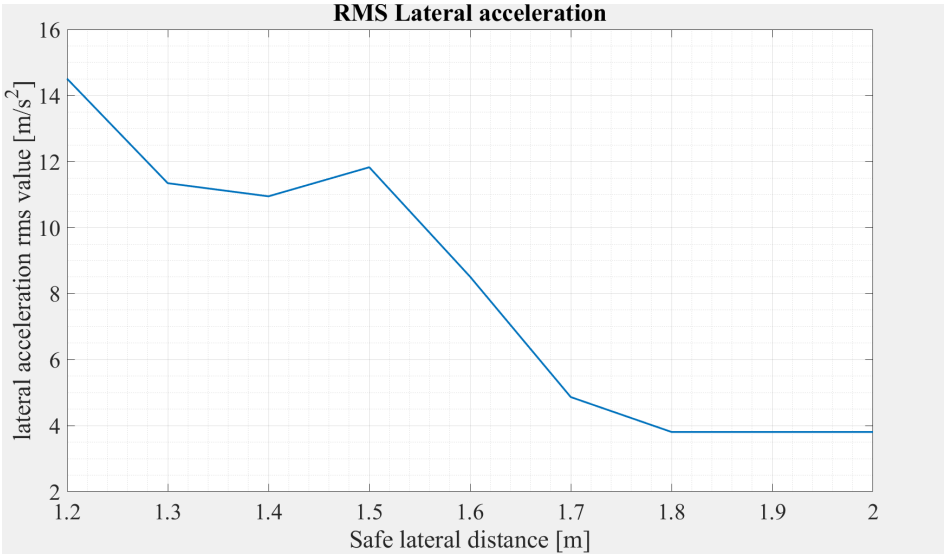


Figure 43: RMS lateral acceleration trend for varying safe lateral distances - Stanley

For the same considerations of the PIDF controller (Section 2.1.1) a safe distance value of $1.7m$ is chosen, ensuring the vehicle to stay within the lane:

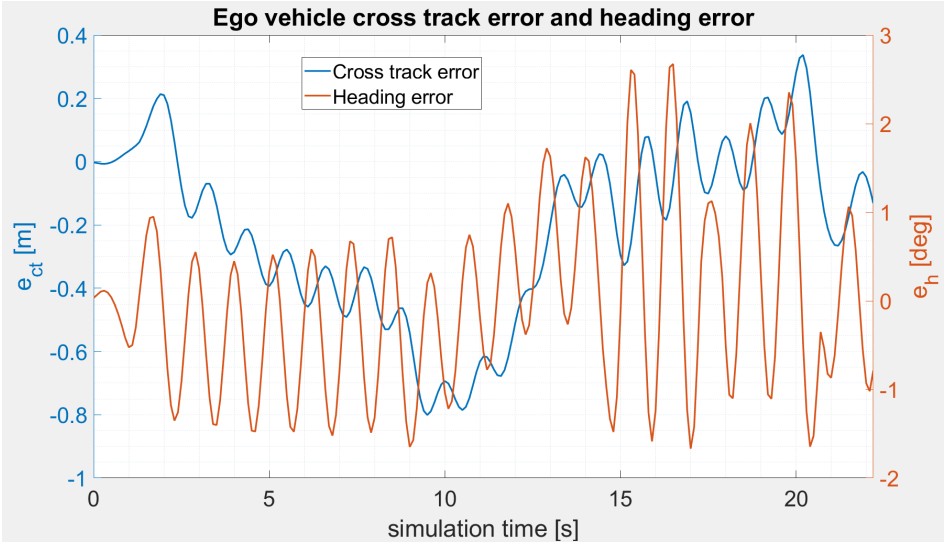


Figure 44: Cross-track and heading errors for the test bench simulation with safe distance = 1.7m - Stanley

Also in this case the value of cross track error never overcomes $1m$, meaning that the vehicle remains within the lane. Although, the behaviour of the controller is not completely satisfactory, showing highly oscillating errors and values of cross track error already close to the limit conditions (lane departure). This means that in a more severe condition, the system could easily fail. The explanation of these problems, as well as a possible solution, are reported in the Section 2.2.2.

2.2 Exercise 2: Stability robustness

In the following section, different road profiles are generated, varying the shape and the radius of curvature, to test the LKA controllers tuned in Part 1 (Section 2.1).

2.2.1 Road profile test generation

Starting from the provided function 'createDoubleCurveScenario', a new one has been developed, called 'createScenarioTrack', to obtain a closed path featuring constant curvature turns, with the possibility to change the value of such radius; the general shape of the track is made of:

- 4 straights;
- 2 left-right chicanes;

- 2 wide right turns;

The shape is represented below:

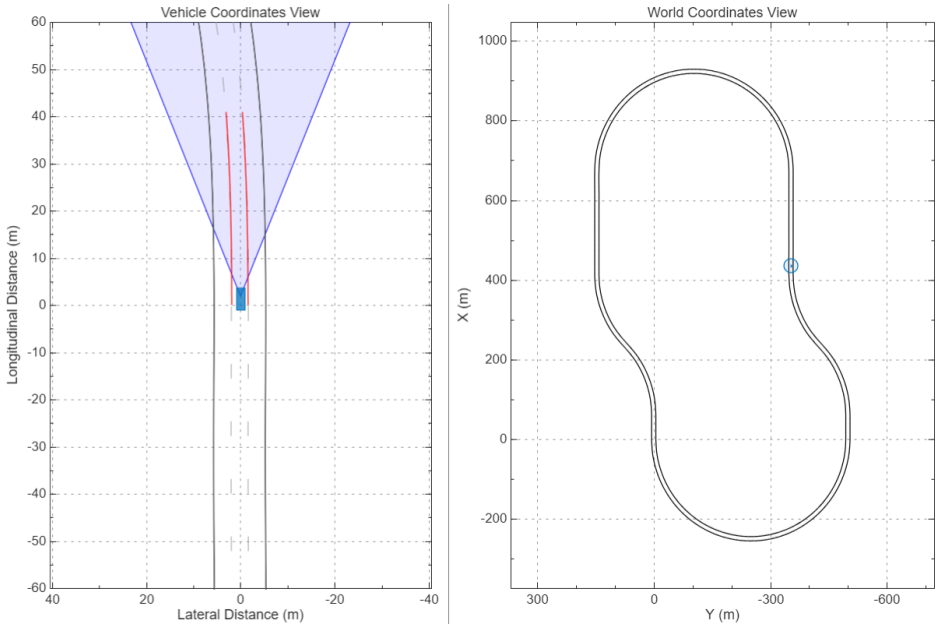


Figure 45: Bird's eye of the track used for controllers validation

With respect to the initial curvature value adopted in the test bench scenario (250m) three smaller radius values are investigated, standing for more severe driving conditions to manage for the LKA controllers: 200, 150, 100m.
Together with the road shape, the waypoints for the driver trajectory are defined introducing a deviation with respect to the lane center, as seen in Section 2.1.1.

2.2.2 Controller validation

PIDF

The different simulations carried out in this section are analysed tracking a single parameter that, for the sake of simplicity, is enough to determine if the system is able to maintain a safe behaviour that, throughout the simulation, is the maximum of the absolute value of the cross track error. At first, the simulations are performed with the LKA controller always engaged (safe distance $= 2m$) and subsequently the condition is partially relaxed. The test is using 4 different values of speed equal to $[80, 100, 120, 140]km/h$ and 4 values of road curvature equal to $[100, 150, 200, 250]m$; all the possible combinations of these scenarios are investigated. The result is a map of maximum cross-track error values:

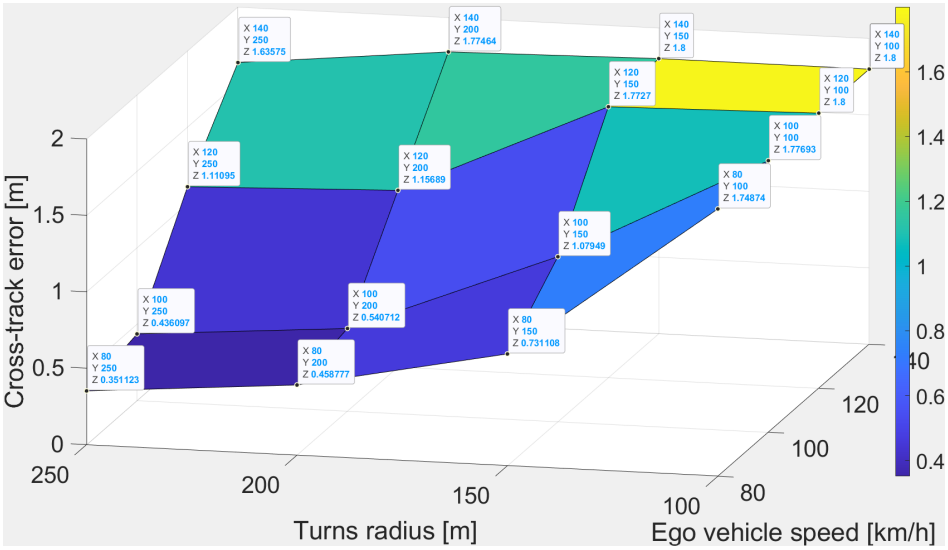


Figure 46: PIDF (safe distance=2m) maximum cross-track error for different driving scenarios

The values of the maps are all saturated to $1.8m$, which corresponds to half of the width of the lane: this means that the combinations of speed and curvature radius providing such a value of maximum cross-track error are leading the vehicle out of the lane. It is reminded that where the values of cross track error are lower than $1m$, it roughly means that the vehicle always remained inside the lane. The behaviour observed is quite straightforward, with a reasonably monotonic increase of the error both increasing the speed and reducing the curvature radius.

The same procedure is carried out reducing the value of safe distance to $1.7m$, leading to the following map:

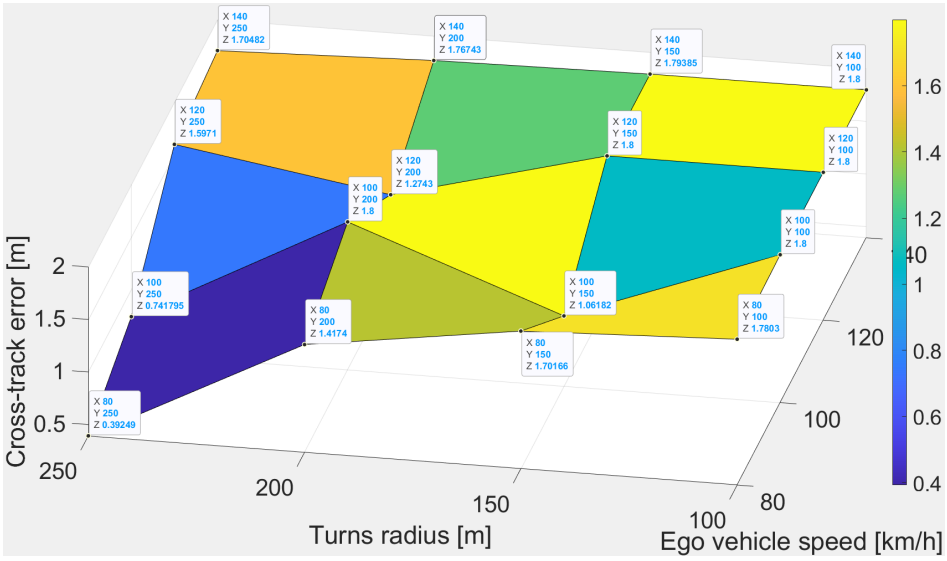


Figure 47: PIDF (safe distance=1.7m) maximum cross-track error tracking for different driving scenarios

In this case the obtained values of cross-track error are significantly higher than the previous one, meaning that the controller is not able to keep the vehicle totally inside the lane; however, this is partially justified by the fact that the action of the driver, when the system is disengaged, is introducing a disturbance to the trajectory, that the LKA tries to attenuate somehow.

Stanley

The same combinations of scenario used for the PIDF controller testing are simulated for the Stanley one as well, adopting a safe distance equal to $2m$; the results are shown below:

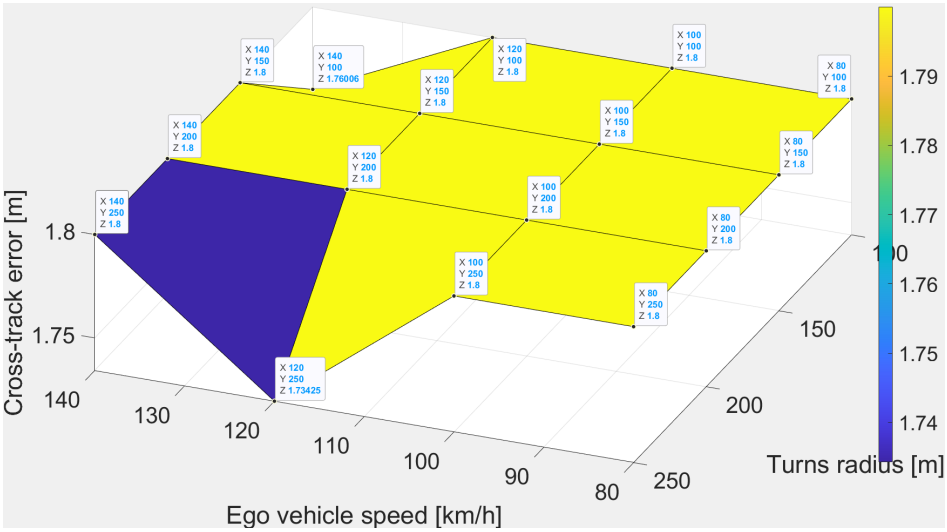


Figure 48: Stanley (safe distance=2m) maximum cross-track error for different driving scenarios

The resulting behaviour is pretty poor even when the controller is always active, indeed the vehicle exits from the track, so it would make no sense to go on with the investigation of the safe distance case at 1.7m. An explanation to this behaviour is provided in the following section.

2.2.3 Stanley controller analysis and improvement

During the tuning of the parameter K_s of the Stanley controller, an interesting behaviour was observed: for low values of this parameter (< 0.1) a satisfactory control of the heading error was obtained, on the contrary poor cross-track error management is achieved. To counteract this issue, the basic idea would be to increase the value of K_s. Although, for all the values > 0.1 that have been tested, a further problem arose: the correction of the cross-track error intrinsically induces an additional heading error. Being this error directly present in the output of the controller, it induces the two components of the output to fight against each other, providing an overall oscillatory behaviour, which degrades as K_s is increased. For this reason, probably because of the particular vehicle parameters considered, no satisfactory value of K_s for such formulation of the Stanley controller can be achieved.

An idea proposed in this project to solve the above mentioned issues is to introduce an additional gain for the heading error, so that the oscillations are damped; the modified ex-

pression is consequently:

$$\delta_0 = K_{s,h} \cdot e_h + atan(\frac{K_{s,ct} \cdot e_{ct}}{v_b + v_x})$$

The tuning of the controller is now realized by two gains; an improved controller behaviour is obtained selecting $K_{s,h} = 0.45$ and $K_{s,ct} = 0.55$.

To validate its enhanced performances, during the simulation the safe distance is set to $2m$ to keep the controller always activated. Analysing the results, the system behaviour is strongly improved, since the vehicle visibly undertakes the curves in a smoother way, by a graphical overview from the 'Bird's Eye View tool'. This enhancement is confirmed by means of a comparison between the cross-track and heading errors resulting from the original controller, with respect to the ones resulting from the improved ones. The two trends compared follow:

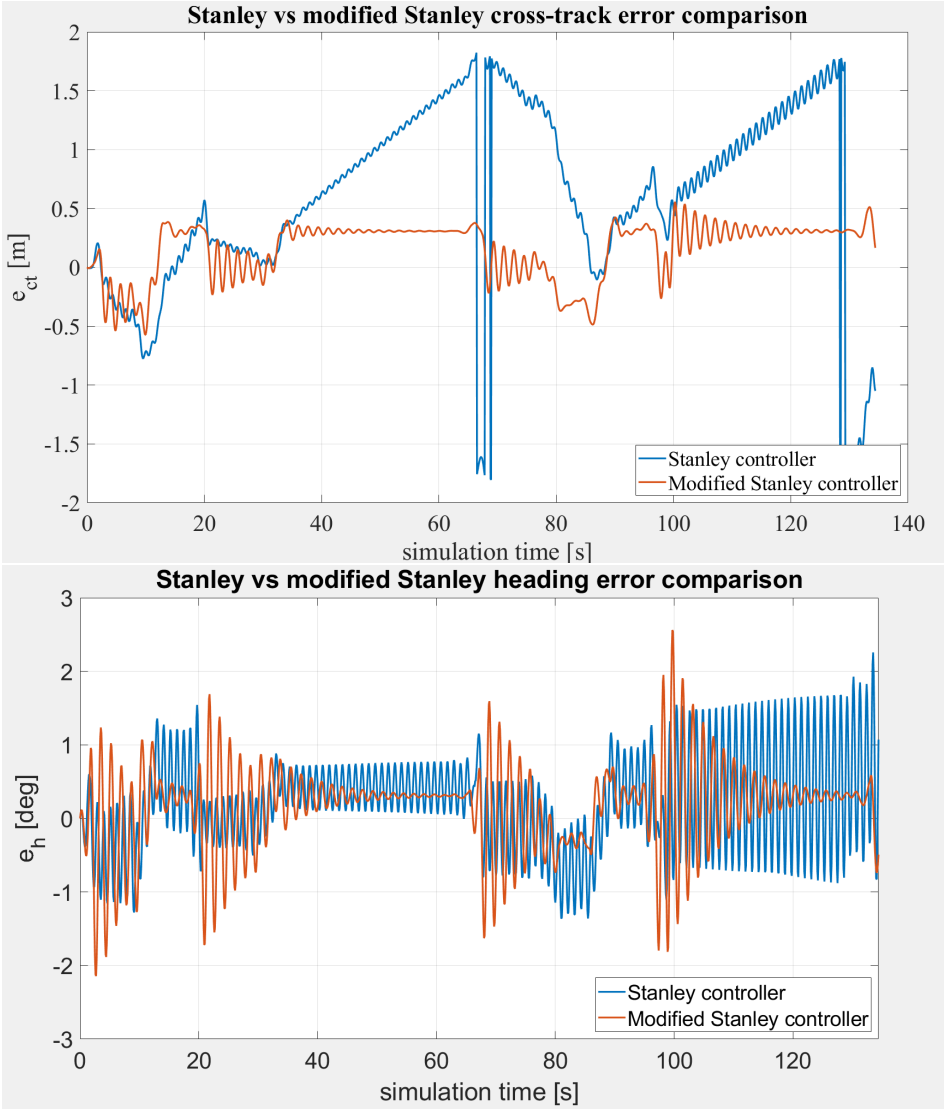


Figure 49: Comparison between Stanley and modified Stanley controller errors

From the behaviour of the cross-track error, it is evident how the introduction of the second gain is providing a significant improvement to the Stanley controller. The heading is not clearly reduced in terms magnitude, but the modification introduced a damping effect so that, when an error is introduced, its amplitude is progressively reduced, instead of being amplified, as in the other case.

The modified version of the Stanley controller is subsequently tested on the different tracks and for different speeds, as already done for the other controllers; the results are shown:

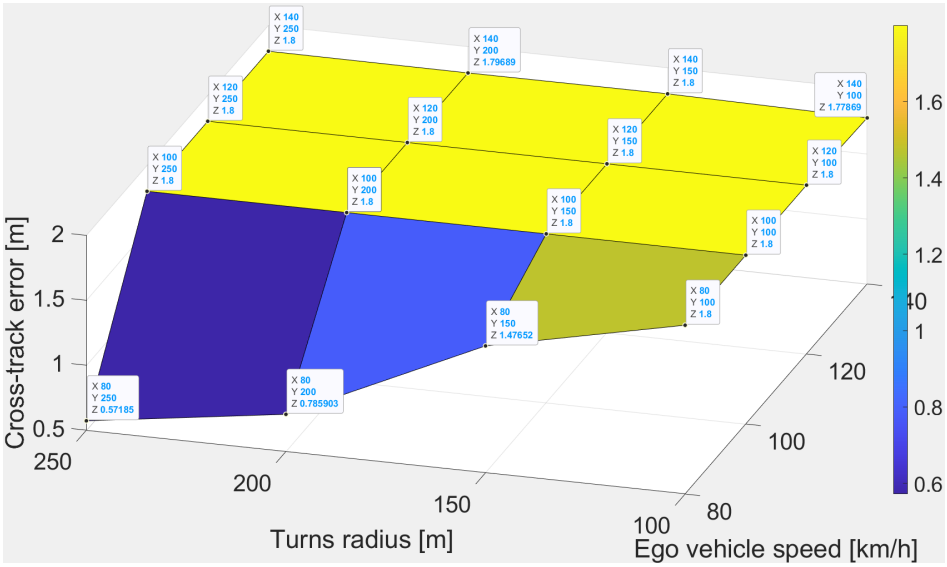


Figure 50: Modified Stanley (safe distance=2m) maximum cross-track error for different driving scenarios

An improvement can still be observed with respect to the basic version of the Stanley controller, since, at least for simulations at $80km/h$, the cross track error remains unsaturated for $250, 200$ and $150m$. However, the general behaviour remains pretty poor if compared to the PIDF controller, meaning that for the particular vehicle considered, the scheme of the Stanley controller is not adequate.

2.3 Exercise 3: Harsh boundary conditions

The last section is meant to show the improvements in terms of safety achieved by the LKA when it is enabled. The effect of the introduction of such a system on a vehicle is discussed.

2.3.1 Incoherent radii setting

As previewed in Section 2.1.1, when the LKA controller is disabled, the driver follows a trajectory determined by the interpolation of subsequent waypoints, thus his aim isn't anymore the lane maintenance, but to follow a real driving scenario where lane changes are present. In this case the radius of the turns followed by the driver are increased with respect to the track, in order to simulate a distracted driver and a comparison is carried out between the vehicle with the engaged LKA and the vehicle with the disengaged controller; both the PIDF controller and modified version of the Stanley one are considered for the comparison and activated accounting for a safe distance $= 1.7m$.

2.3.2 Driver influence on stability

The adopted evaluation parameters for the comparison are the cross-track and the heading errors. The results corresponding to the PIDF and the Stanley controllers are reported here:

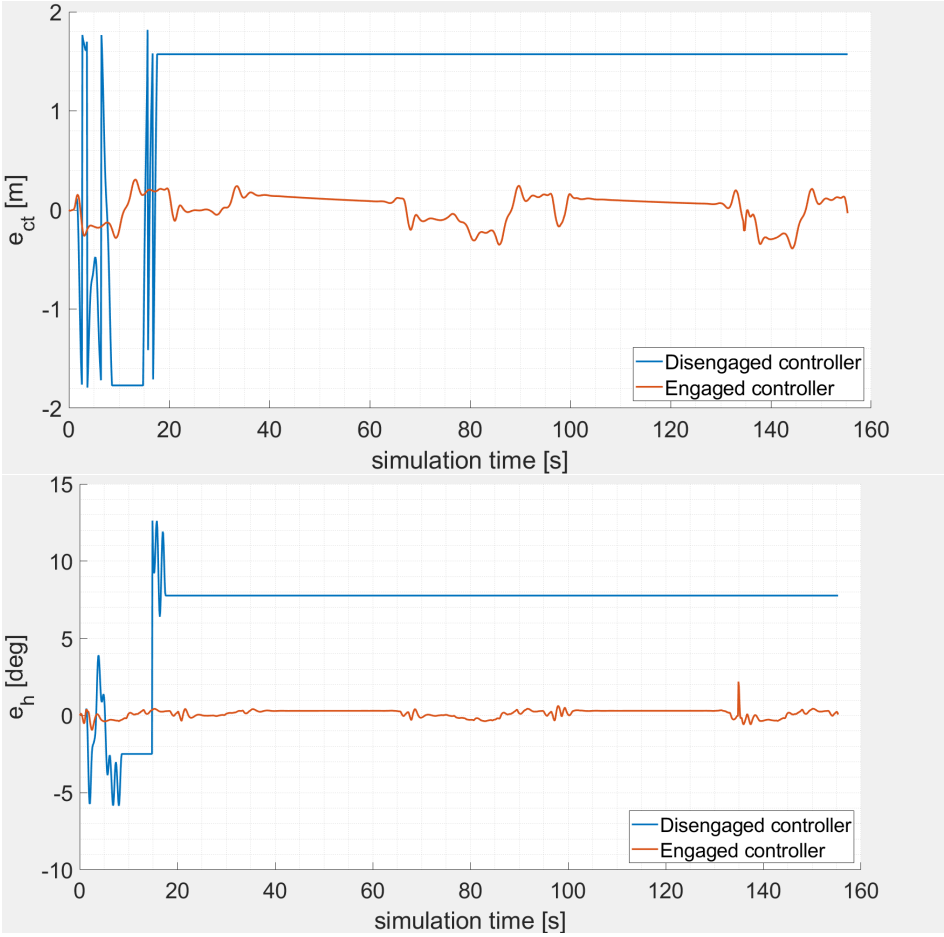


Figure 51: Comparison between errors of the vehicle with PIDF controller and of the vehicle without controls

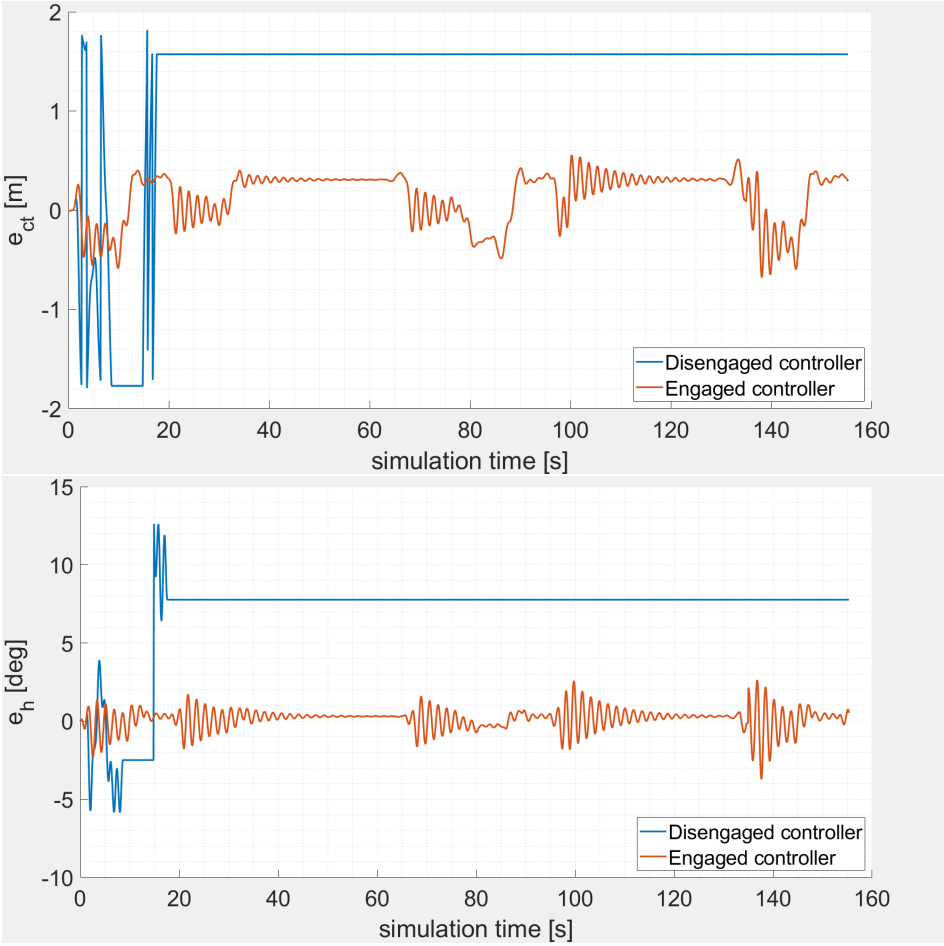


Figure 52: Comparison between errors of the vehicle with modified Stanley controller and of the vehicle without controls

The errors comparisons confirm how both the PIDF and the modified Stanley LKA systems tuned are an effective control solution for the passengers' safety, since the vehicle without any activated control presents a higher error with respect to the reference lane center.
Furthermore it's interesting to view the trajectory followed by the vehicle in the three different simulated scenarios:

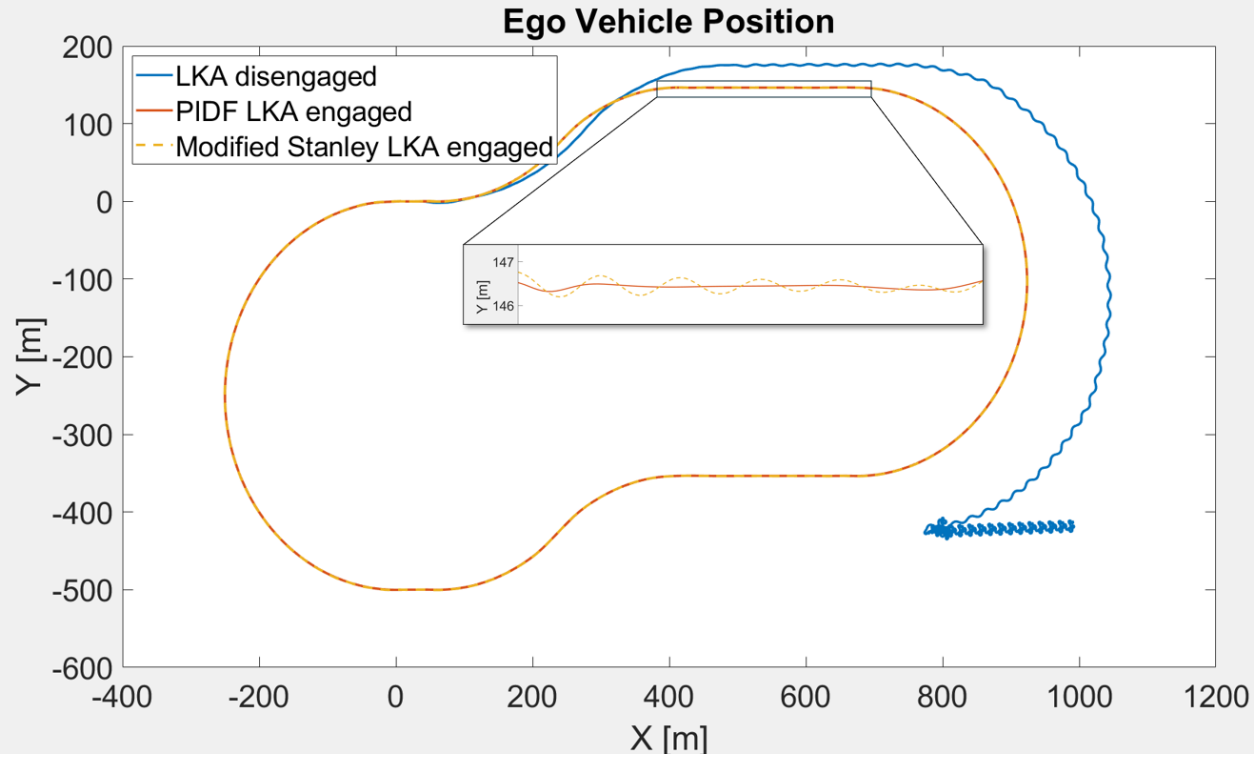


Figure 53: Trajectories comparison for the three different scenarios

As previewed in Section 2.2.3, considering the modified Stanley controller, the cross-track error oscillation, after a given excitation, dampens along the path. The graphical evidence of this behavior is highlighted by the zoom in the picture.
Finally, this chart clearly shows that the introduction of the LKA is able to avoid an announced accident, since the driver would clearly lead the vehicle out of the road, while the action of the controllers prevent this from happening.

2.4 Project 2 results

The main achievements and critical points of the Project 2 are summarized here:

- different control schemes can be implemented for the management of a LKA system; a PIDF and a Stanley controller have been compared. The performance of the controllers are strictly related to the vehicle parameters. For the particular vehicle used for this project, the PIDF based controller returns the best results in terms lateral and orientation errors management;

- the safe lateral distance is an important indicator for the overall system behaviour: the choice of a low value leaves a wide freedom range for manual driving, but making at

the same time more difficult for the controller to maintain the lane, risking to compromise the passenger's safety. The suitable value of $1.7m$ selected in the project could be considered excessive for a lane keeping system, but it is a necessary choice since problems on vehicle stability have been observed for lower values. However, it is still worth evidencing that these behaviours are influenced by the driver's actions as well: a different driver path could lead to a relaxation of this constraint.

- The analysis carried out varying the driving scenario with respect to the design one evidences that a simple controller like the ones considered is not able to provide a safe behaviour in all the conditions that could present in real applications. More complex approaches like gain scheduling or NMPC could be implemented for implementation in real applications.

Appendix

The complementary analysis of the results coming from the validation of the controllers in Section 2.2.2 consists in the detection of the possible failed simulations resulting in the vehicle spinning. As detection parameter the heading error is selected: when its values reaches 20 the vehicle is supposed to spin. Here the tables for each of the four simulations are reported:

	$R = 250m$	$R = 200m$	$R = 150m$	$R = 100m$
$V = 80km/h$	pass	pass	pass	pass
$V = 100km/h$	pass	pass	pass	pass
$V = 120km/h$	pass	pass	pass	pass
$V = 140km/h$	pass	pass	pass	pass

Table 2: PID (safe distance=2m) spin detection

	$R = 250m$	$R = 200m$	$R = 150m$	$R = 100m$
$V = 80km/h$	pass	pass	fail	pass
$V = 100km/h$	pass	fail	pass	pass
$V = 120km/h$	fail	pass	pass	pass
$V = 140km/h$	pass	fail	pass	fail

Table 3: PID (safe distance=1.7m) spin detection

	$R = 250m$	$R = 200m$	$R = 150m$	$R = 100m$
$V = 80km/h$	pass	pass	pass	fail
$V = 100km/h$	fail	fail	fail	fail
$V = 120km/h$	fail	fail	fail	fail
$V = 140km/h$	fail	fail	fail	fail

Table 4: Stanley (safe distance=2m) spin detection

	$R = 250m$	$R = 200m$	$R = 150m$	$R = 100m$
$V = 80km/h$	pass	pass	pass	pass
$V = 100km/h$	pass	pass	pass	fail
$V = 120km/h$	fail	fail	fail	fail
$V = 140km/h$	fail	fail	fail	fail

Table 5: Stanley (safe distance=1.7m) spin detection