



Adaptive Cruise Control Design

Master's Degree in Automotive Engineering

Loris Fonseca: s342696@studenti.polito.it

Nikoloz Kapanadze: s342649@studenti.polito.it

Maurizio Pio Vergara: s346643@studenti.polito.it

Giovanni Castagna: s333934@studenti.polito.it

a.y 2024/2025

Contents

1	Introduction	3
2	Upper level controller	3
2.1	Effect of controller parameters on vehicle response	7
2.2	Sensitivity analysis on the Ego vehicle transfer function	11
2.3	Response analysis to a random leading vehicle speed profile	12
3	Upper level controller - Platoon stability	13
3.1	Analysis of the plateaux behavior	16
3.2	Controller failure analysis	17
4	Lower level controller	19
4.1	Effect of controller parameters on vehicle response	22
4.2	Response analysis to a random leading vehicle speed profile	23
5	Lower level controller - Platoon stability	25
5.1	Impact of reduced string size on stability	28
5.2	High-speed profile tracking	29
6	Conclusions	30

1 Introduction

Adaptive Cruise Control (ACC) is designed to automatically adjust a vehicle's speed to maintain a safe distance from a lead vehicle. This project focuses on the development and evaluation of an ACC system using a constant time-gap policy, implemented through a Proportional-Derivative (PD) controller. The initial phase involves the use of a simplified vehicle model to analyze the controller's response to various lead vehicle speed profiles. Key parameters such as time gap and control gain are tuned to optimize performance while ensuring compliance with safety constraints outlined in ISO 15622:2018. The study then extends to a complete vehicle model, incorporating realistic dynamics and actuator behavior. Sensitivity analyses are performed to determine the impact of parameter variations on system safety and responsiveness. Additionally, the project examines platoon stability to assess the controller's performance in multi-vehicle scenarios. The objective is to evaluate the feasibility, safety, and effectiveness of the ACC controller under different conditions and configurations.

2 Upper level controller

The first section of this project focuses on the design and implementation of an Adaptive Cruise Control system in Matlab/Simulink, using a Proportional-Derivative controller based on a constant time gap policy. The system to be controlled consists of two vehicles: a leading vehicle, which sets a given speed and acceleration profile, and a following vehicle, which implements the ACC strategy. This initial analysis is limited to evaluating the individual stability of the following vehicle, without considering the overall platoon stability of a larger fleet. The control logic adopted is the Constant Time Gap (CTG) policy, where the controller aims to maintain a specified time gap between the vehicles.

The CTG control law used in this project is defined by the following PD equation:

$$a_i = -\frac{1}{h}(\lambda \delta_i + \dot{\epsilon}_i) \quad (1)$$

where:

- h is the desired time gap between the two vehicles,
- $\epsilon_i = x_i - x_{i-1}$ is the relative distance between the follower and the leader,
- δ_i is the spacing error defined as $\delta_i = \epsilon_i + L_{des}$,
- L_{des} is the desired spacing, given by $L_{des} = h \cdot \dot{x}_i$,
- λ is a tunable control parameter used to achieve the desired system behavior.

The following figures illustrate the Simulink implementation of the proposed control scheme.

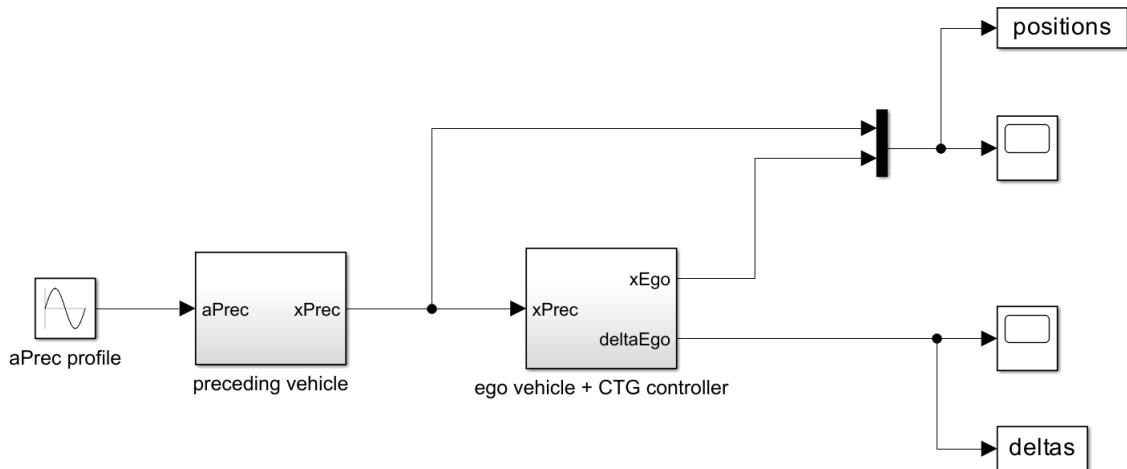


Figure 1: Primary control architecture of the leader and follower vehicles

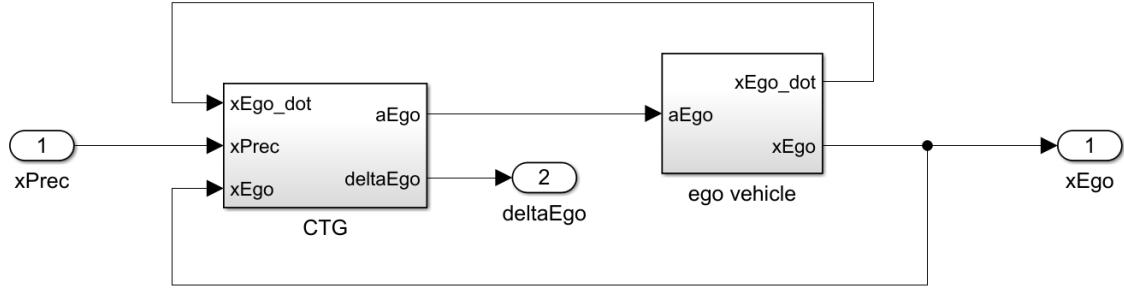


Figure 2: Control block of the ego vehicle and the CTG controller

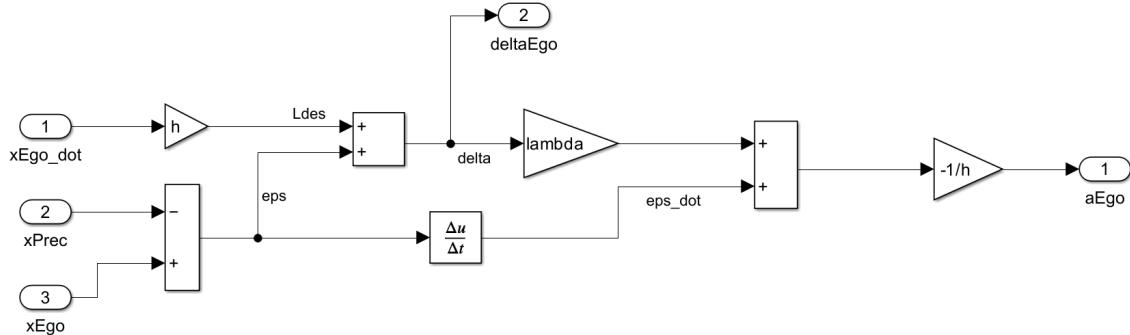


Figure 3: CTG block implementation

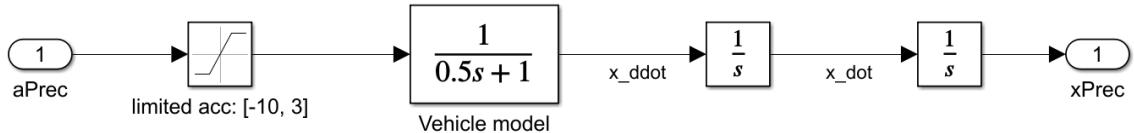


Figure 4: Vehicle model block implementation

To be noticed that, as shown in Figure 4, the vehicle model used to describe the longitudinal dynamics is represented by a simple first-order transfer function. This approach is applicable when a low-level controller is available to convert the desired acceleration into appropriate throttle and brake commands. The transfer function introduces a delay in the control loop, reflecting the vehicle's responsiveness to external inputs. This delay is characterized by the parameter τ , which is assumed to be $\tau = 0.5$ in the reported block diagram. Moreover, the vehicle model includes a saturation operator on the input acceleration to prevent unfeasible or unrealistic acceleration values.

The following charts illustrate the longitudinal positions of the leading and following vehicles, as well as the spacing error of the controlled vehicle, as function of the simulation time. These results were obtained through a trial-and-error procedure used to tune the parameters λ and the time gap h , with the goal of improving the vehicle's response to the speed profile of the lead vehicle, which is modeled as a harmonic function in this simulation. The final values of these parameters, which ensure proper functioning of the ACC, are listed below:

$$\begin{cases} \lambda = 0.5 \\ h = 2.7 \end{cases}$$

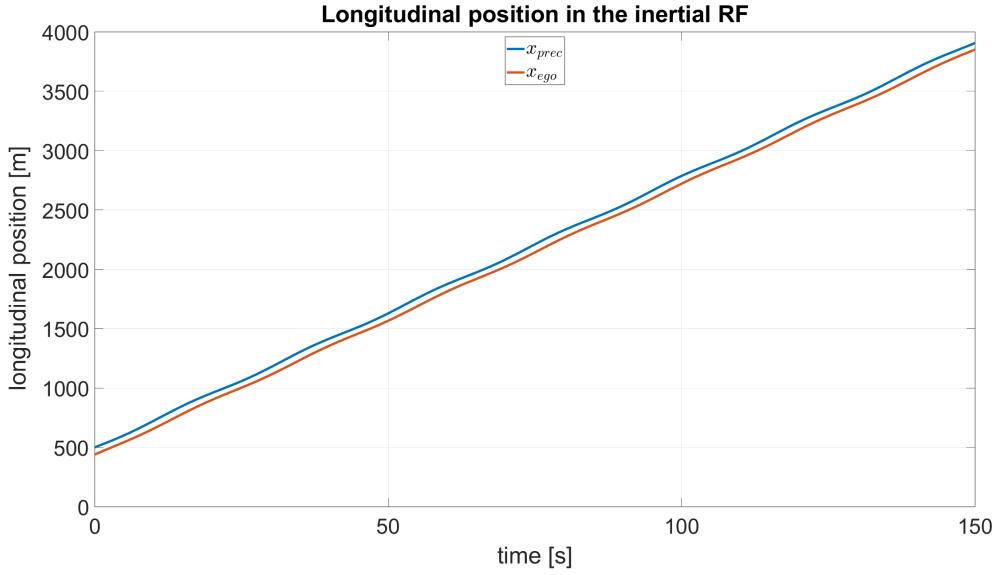


Figure 5: Vehicles longitudinal position evolution

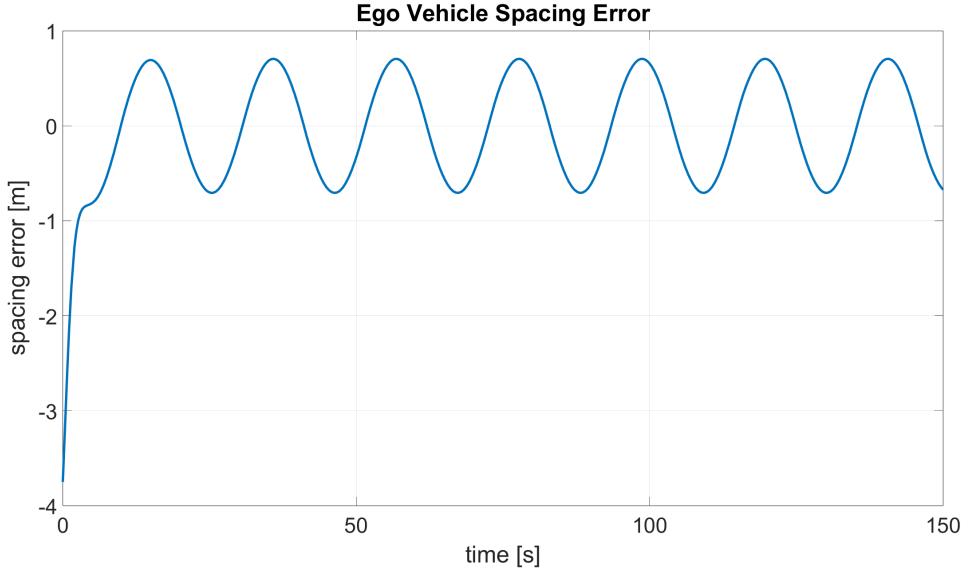


Figure 6: Ego vehicle spacing error

The proper functioning of the control system can be evaluated by analyzing the time evolution of the spacing error. Initially, due to the imposed starting conditions, the spacing error exhibits a transient phase with relatively large values. However, it quickly converges and begins oscillating around zero, with an amplitude not exceeding 1 meter. This level of error is acceptable, given the relatively large distance between the two vehicles. Indeed, as illustrated in the chart showing the longitudinal positions, the trajectories of the leading and following vehicles remain well separated at all times. This confirms that the follower vehicle maintains a safe distance and never risks colliding with the lead vehicle, thereby validating the effectiveness of the Adaptive Cruise Control system. This conclusion is further supported by the time gap plot shown in Figure 7. The plot confirms that the CTG strategy is correctly implemented, as the time gap oscillates around the desired target value.

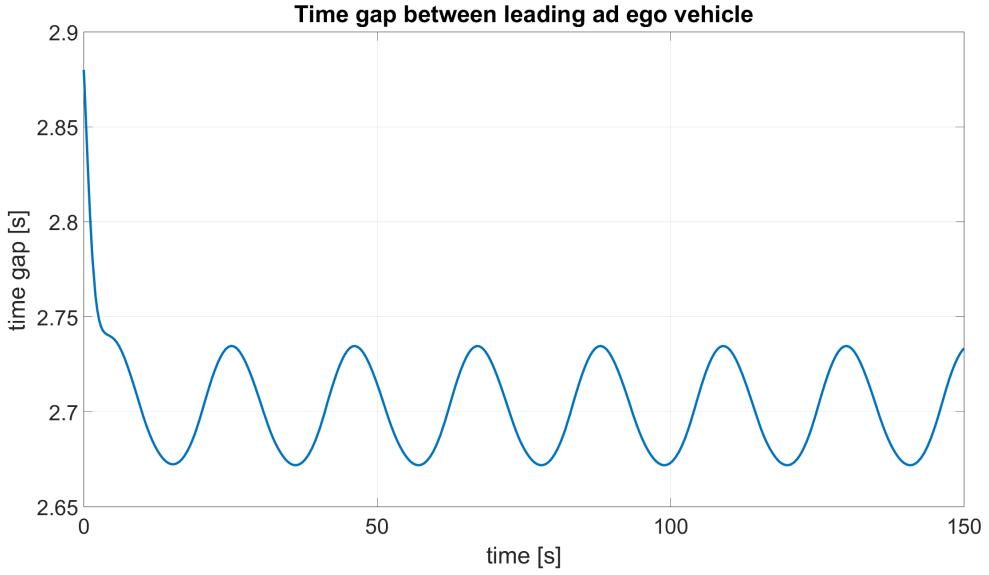


Figure 7: Time gap between leading and follower vehicle

Once the correct behavior of the controller has been verified, the final analysis focuses on its compliance with relevant legislation. In particular, the ISO 15622:2028 standard defines the performance requirements that Adaptive Cruise Control (ACC) systems must meet. The key parameters to evaluate in this context are the vehicle's acceleration/deceleration values and the deceleration rate. According to the ISO standard, the following limits must be satisfied:

- Maximum average deceleration: 3.5 m/s^2 , averaged over 2 seconds
- Maximum deceleration rate: 2.5 m/s^2 , averaged over 1 second
- Maximum automatic acceleration: 2.0 m/s^2

The charts below illustrate the evolution of these parameters over the simulation period. Figure 8 shows the follower vehicle's acceleration over time, including both the instantaneous acceleration and the 2-second moving average, along with the regulatory limit. It is evident that both the instantaneous and averaged deceleration values remain within the specified bounds, confirming that the controller observes the functional and regulatory requirements.

Similarly, Figure 9 displays the time evolution of the acceleration rate alongside its corresponding regulatory limit. The results confirm that the averaged deceleration rate consistently remains above the the imposed lower limit, further demonstrating compliance with the ISO standard.

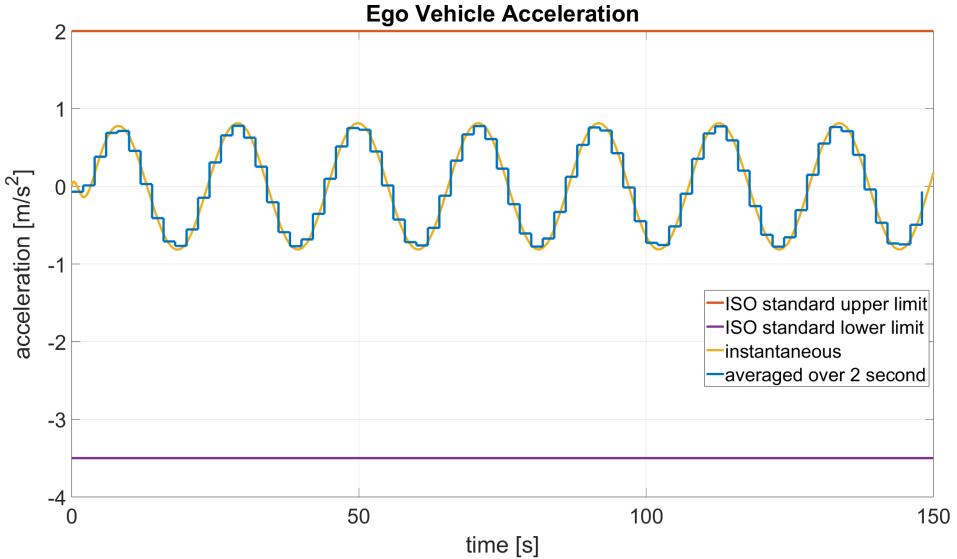


Figure 8: Ego vehicle acceleration

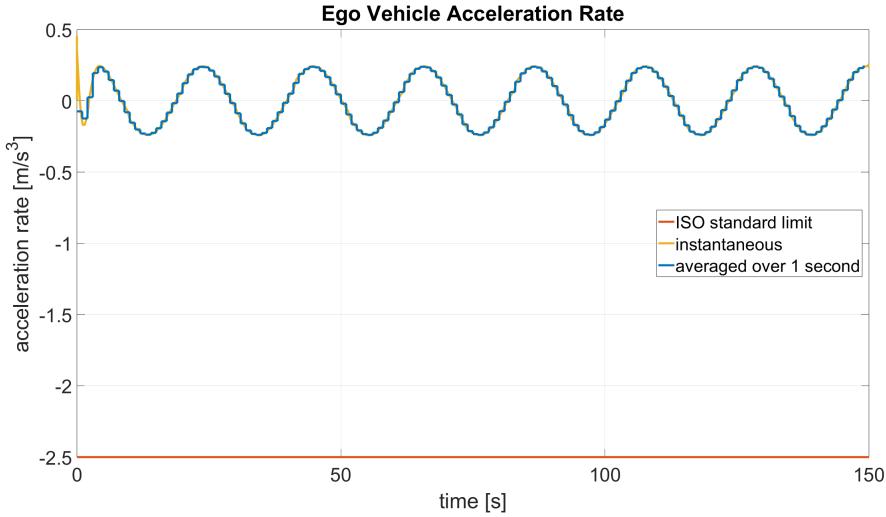


Figure 9: Ego vehicle acceleration rate

2.1 Effect of controller parameters on vehicle response

After verifying the proper functioning of the controller with the tuned values of λ and h , a sensitivity analysis can be performed to evaluate the controller's behavior under different parameter values and to determine the safety margins for λ and h that ensure compliance with relevant legislation. These two parameters significantly influence the response of the following vehicle, as they directly affect the control strategy and vehicle dynamics. Specifically, h , the time gap, must be sufficiently large to allow the vehicle enough time to react to the controller's input and avoid collisions with the lead vehicle. However, an excessively large time gap may impair the performance of onboard sensors, which may struggle to accurately detect the lead vehicle when it is too far away. Conversely, λ is a control parameter that regulates the aggressiveness of the ACC. A higher value of λ results in a more aggressive control strategy, characterized by quicker responses and higher acceleration and jerk levels. If the control becomes too aggressive, it may violate the ISO safety standards. On the other hand, a lower λ leads to a more conservative control behavior, potentially causing delayed reactions that prevent the follower vehicle from accurately tracking the speed of the lead vehicle, thereby compromising the proper functioning of the ACC system.

The following charts show the ego vehicle acceleration and acceleration rate for the following values of h and λ :

$$\begin{cases} \lambda = 10 \\ h = 2.7 \end{cases}$$

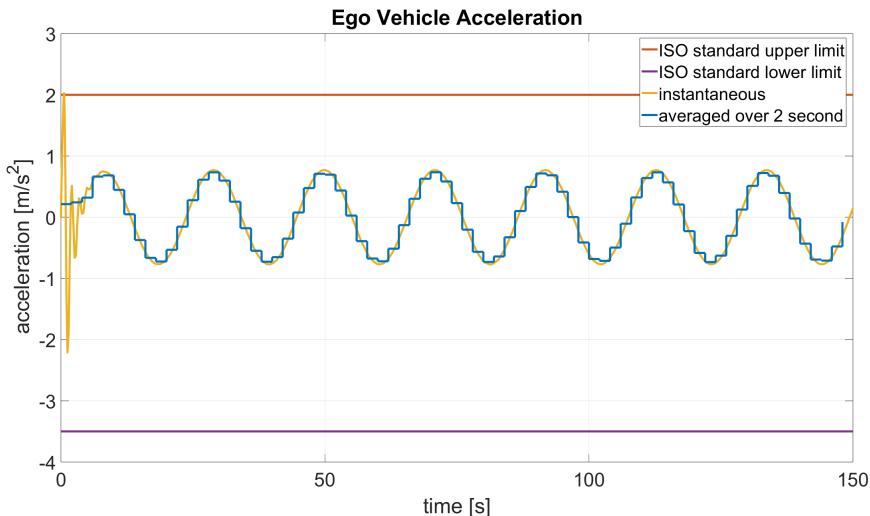


Figure 10: Ego vehicle acceleration with more aggressive controller

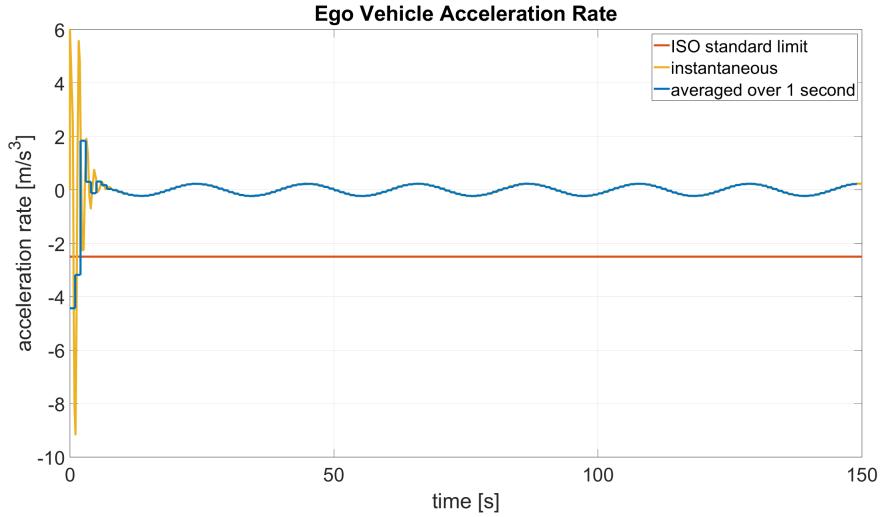


Figure 11: Ego vehicle acceleration rate with more aggressive controller

Compared to the previous case, the same time gap h is maintained, while the value of λ is increased. This allows the analysis to focus specifically on the influence of λ on vehicle dynamics. As previously discussed, a higher λ results in a more aggressive control strategy. This effect is clearly observable in the acceleration and jerk plots, where, at certain moments, the ISO limits are exceeded. This indicates that the control becomes overly aggressive, potentially compromising passenger comfort and violating regulatory safety standards, thereby making the system non-compliant with ISO requirements.

Similarly to the analysis conducted for the sensitivity with respect to λ , a sensitivity analysis is also performed to assess the influence of the time gap parameter h on both the controller performance and vehicle behavior. Two simulations are considered: one with a relatively large value of h , and the other with a small value. Both simulations are carried out using the same value of λ , specifically the one corresponding to the properly tuned controller.

The parameters used in the first simulation are reported below, followed by the corresponding plots illustrating the results:

$$\begin{cases} \lambda = 0.5 \\ h = 7 \end{cases}$$

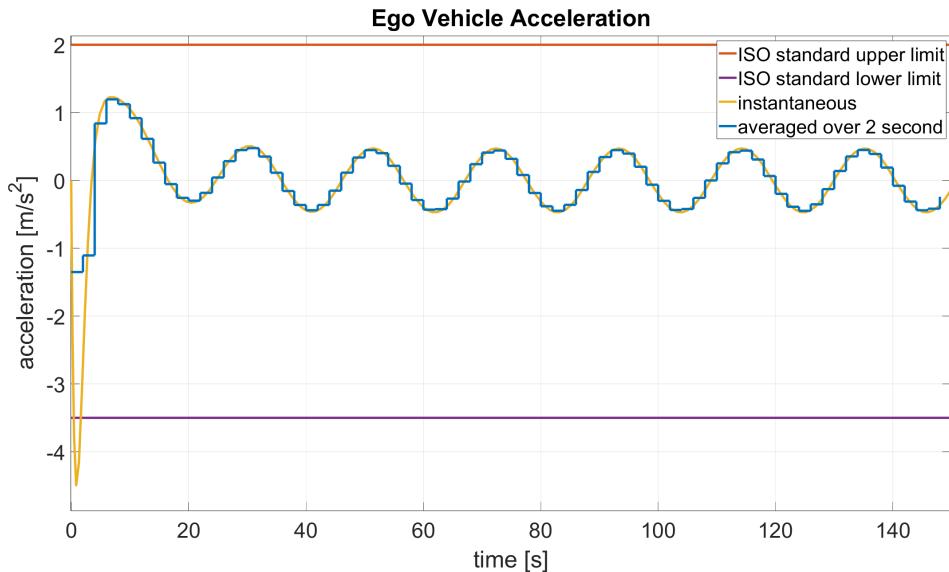


Figure 12: Ego vehicle acceleration with a large time gap

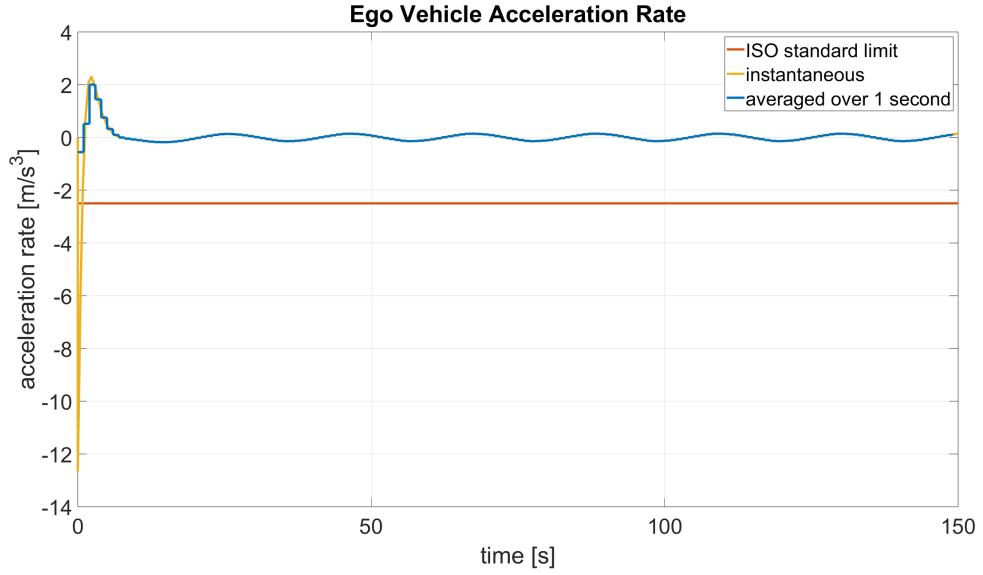


Figure 13: Ego vehicle acceleration rate for a large time gap

As clearly observed from the charts, both the acceleration and the acceleration rate of the ego vehicle exceed the limits prescribed by the ISO standard. However, this behavior is primarily attributed to the influence of the initial conditions. Specifically, as illustrated in Figure 14, the initial distance, and consequently the initial time gap between the leading vehicle and the follower, is significantly smaller than the target time gap enforced by the controller. As a result, at the beginning of the simulation, the controller responds with high deceleration commands in order to rapidly increase the time gap to the desired value.

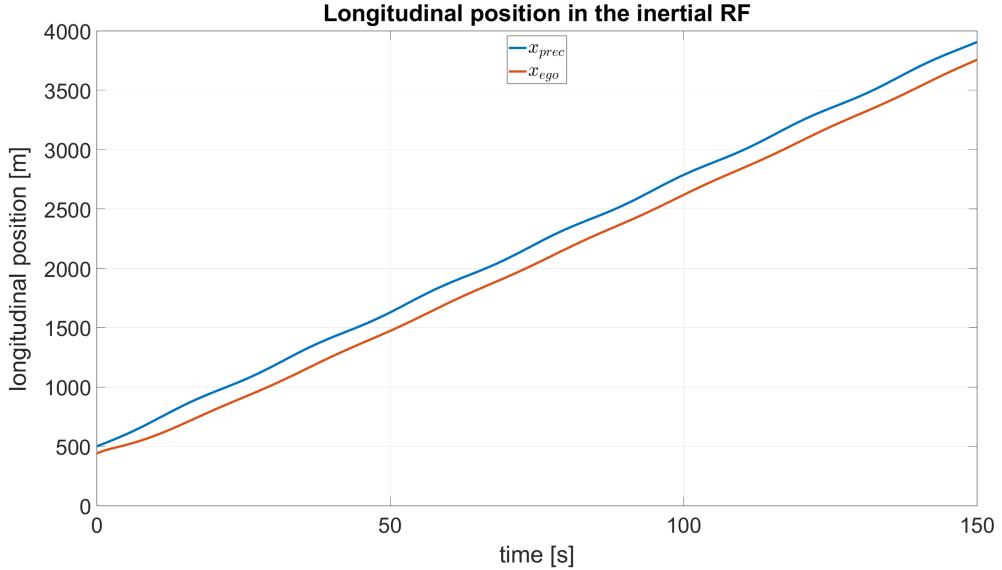


Figure 14: Vehicles longitudinal positions for a large time gap

This issue can be addressed by setting the initial position of the follower vehicle to match the target time gap defined by the controller. However, the time gap should not be set too large. At greater distances, the sensors may fail to detect the lead vehicle accurately, resulting in errors in estimating both the relative position and the time gap between the vehicles.

A similar problem arises when the time gap is set too low. The following charts display the ego vehicle's acceleration and acceleration rate obtained with the control parameters reported below:

$$\begin{cases} \lambda = 0.5 \\ h = 1 \end{cases}$$

As with a large time gap, the ISO standard limits are exceeded even with a small time gap. The key difference between the two cases lies in the vehicle's behavior at the beginning of the

simulation. With a low time gap, the initial distance between the vehicles exceeds the target value for the given speed. To reduce this gap, the controller commands a high positive acceleration, attempting to close the distance quickly. This contrasts with the case of a large time gap, where the vehicle initially decelerates to increase the distance from the leading vehicle.

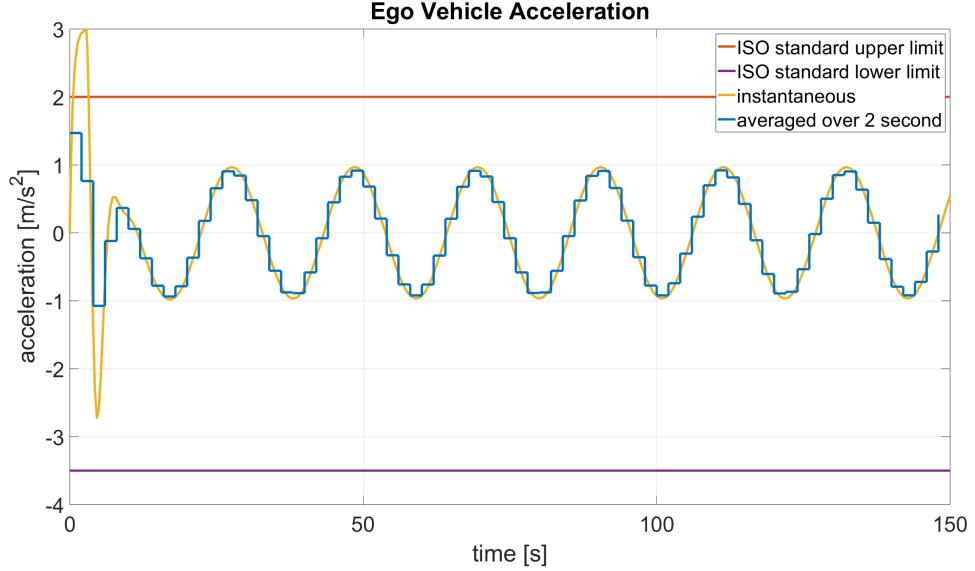


Figure 15: Ego vehicle acceleration with a low time gap

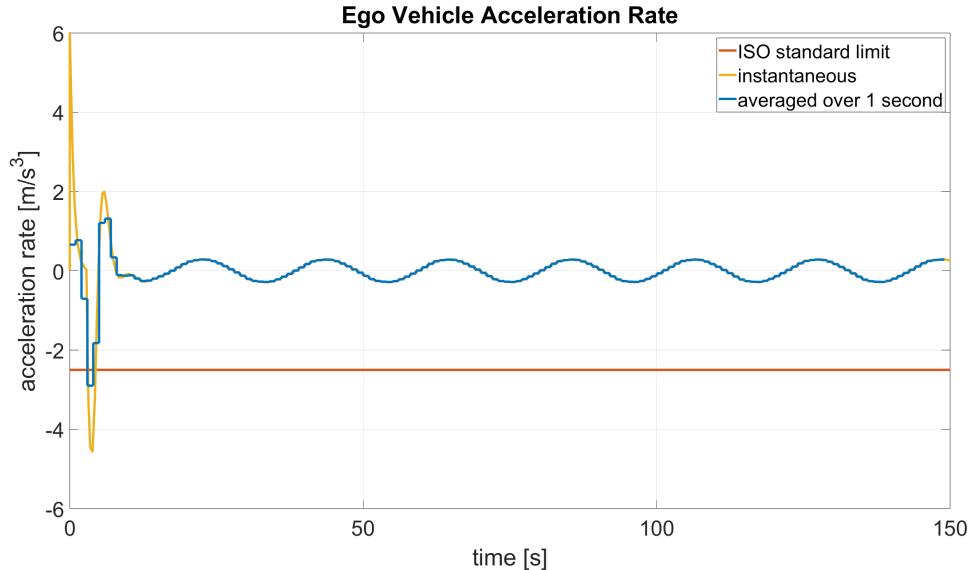


Figure 16: Ego vehicle acceleration rate for a low time gap

This behavior is further illustrated in Figure 17, which shows the longitudinal positions of both vehicles. It is evident that the relative distance is large at the start and decreases throughout the simulation. Comparing this with Figure 14, the impact of the time gap value is clear: the follower vehicle remains much closer in the low time gap scenario, while it stays farther back with a large time gap.

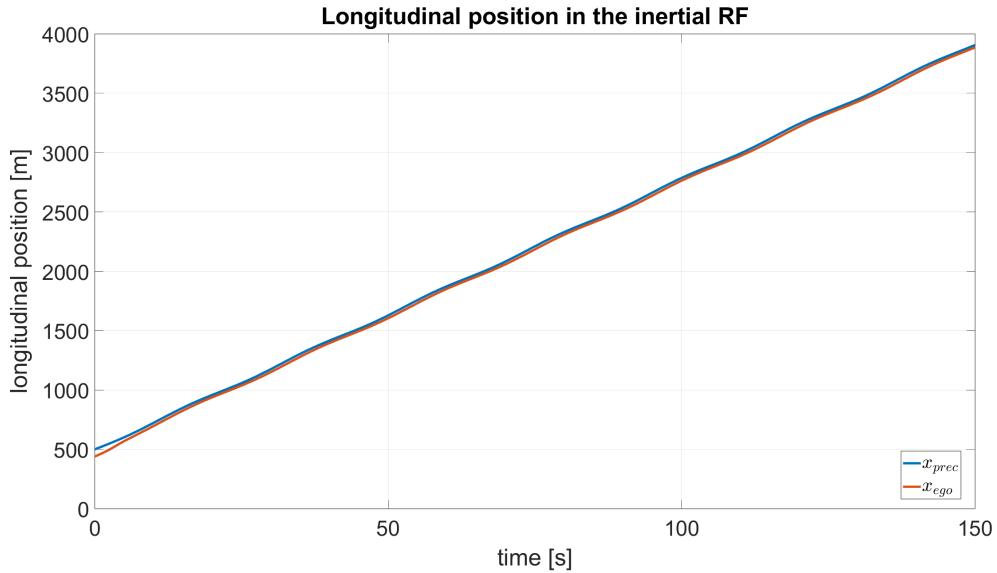


Figure 17: Vehicles longitudinal positions for a low time gap

Based on the analysis of λ and h it is therefore possible to identify the limits of these parameters for which the ACC works safely. Through a trial and error procedure the parameters' intervals found are:

$$\begin{cases} \lambda = 0.05 \div 2.15 \\ h = 2.24 \div 3.12 \end{cases}$$

It is important to note that these limits are not independent of each other. In fact, changing the value of h directly affects the safe range of λ , and vice versa. For example, if the time gap is set to a low value, meaning the follower vehicle is very close to the lead vehicle, the safe range for λ shifts toward higher values. This is because, with a shorter distance between the vehicles, the controller must respond more quickly to any changes in the lead vehicle's speed.

2.2 Sensitivity analysis on the Ego vehicle transfer function

As previously mentioned, in this first simulation, the vehicle is modeled as a first-order transfer function characterized by the parameter τ , as illustrated in Figure 4. It is important to analyze how τ affects the vehicle's response and the admissible ranges of the controller tuning parameters.

The following chart presents the spacing error for $\tau = 5$. A clear degradation in vehicle response performance is observed: the spacing error exhibits oscillations with an amplitude of up to 10 meters, making the ACC system unsafe due to the increased risk of collision with the lead vehicle. This behavior results from the excessive delay introduced by such a high value of τ . With a slower longitudinal dynamic, the vehicle takes significantly longer to react to speed changes. One potential solution to mitigate this effect is to increase the controller's aggressiveness by raising the value of λ . Consequently, when the follower vehicle exhibits slower longitudinal dynamics, the safe operating interval for λ shifts toward higher values. Furthermore, the slower response of the vehicle to the controller input requires setting a larger desired time gap, and thus a greater distance between vehicles, to ensure that the following vehicle does not collide with the one ahead.

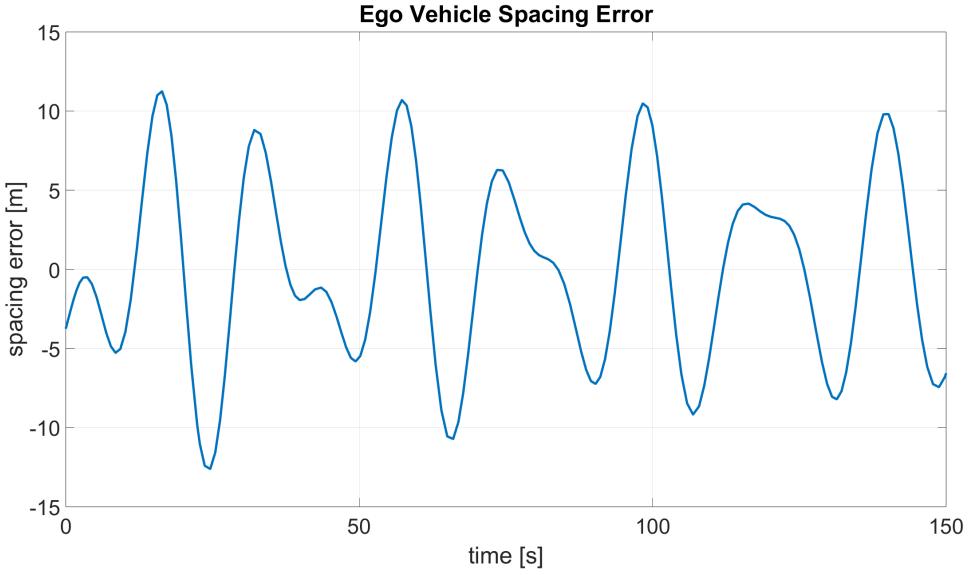


Figure 18: Ego vehicle spacing error for a large τ value

Conversely, with a lower value of τ , for instance, $\tau = 0.1$, which corresponds to a faster vehicle response to control inputs, the spacing error is reduced, as shown in Figure 19. This reduction makes it feasible to decrease the time gap between vehicles and to reduce the controller's aggressiveness, thereby shifting the safe intervals for both λ and h towards lower values.

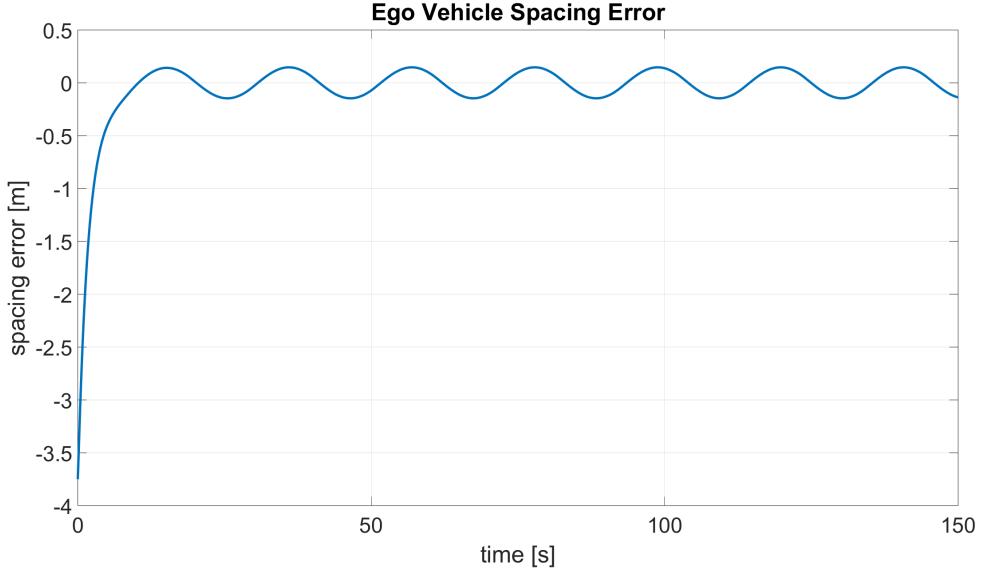


Figure 19: Ego vehicle spacing error for a low τ value

2.3 Response analysis to a random leading vehicle speed profile

The final analysis evaluates the controller's performance when the leading vehicle follows a random acceleration profile, in contrast to the sinusoidal input assumed in the previous sections. As illustrated in the following charts, the spacing error converges to zero and the time gap approaches the desired value. These results confirm the effectiveness of the controller.

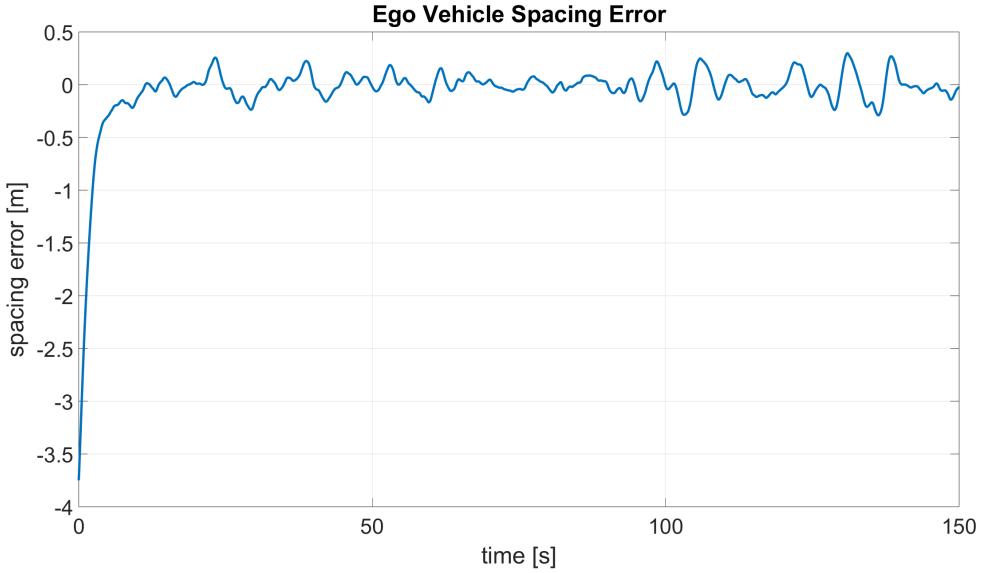


Figure 20: Ego vehicle spacing error for a random leading vehicle acceleration profile

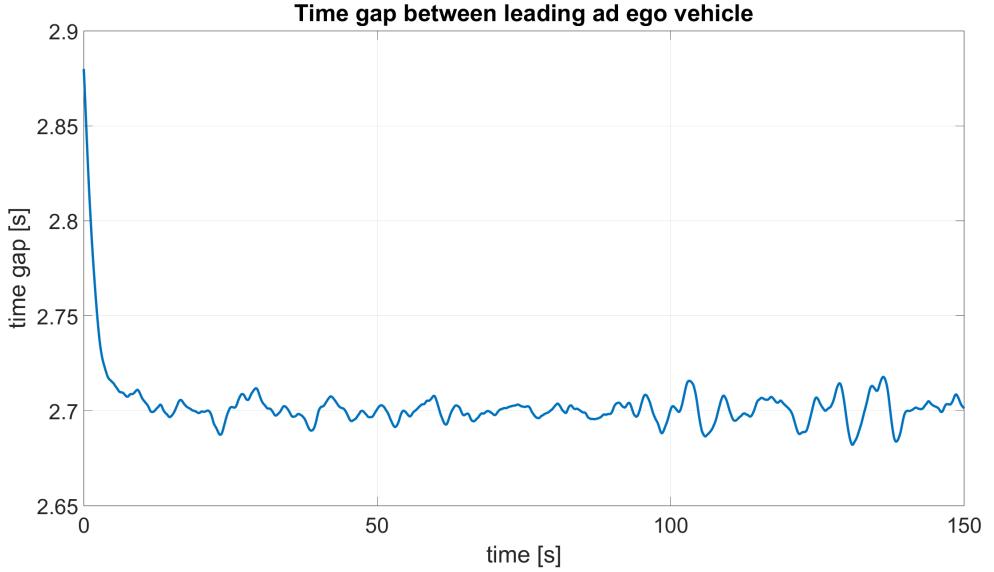


Figure 21: Time gap for a random leading vehicle acceleration profile

3 Upper level controller - Platoon stability

The second section of this document aims to verify the platoon stability of the controller designed in Section 2. To this end, a simulation is implemented involving a platoon of six follower vehicles, each equipped with the previously developed CTG controller. The primary control architecture of the six-vehicle platoon is illustrated in Figure 22.

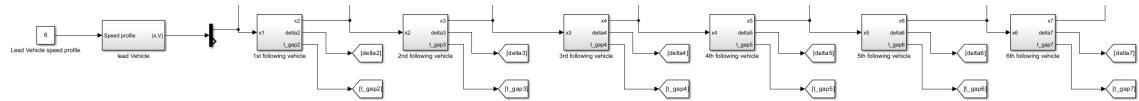


Figure 22: Primary control architecture of a six-vehicle platoon

The following chart presents the main results relevant to verifying platoon stability, specifically the longitudinal positions of the vehicles, the spacing error, and the time gap between them. A preliminary analysis indicates that the controller performs effectively, maintaining platoon stability,

preventing collisions, and achieving the desired time gap between vehicles. In particular, the spacing error converges and begins to oscillate around zero, while the time gap between each pair of successive vehicles stabilizes at the target value, $h = 2.7$ s. Similarly, after a transient phase of approximately 150 seconds, the longitudinal positions of the vehicles align, demonstrating that each vehicle is able to follow the one ahead by adjusting its speed accordingly.

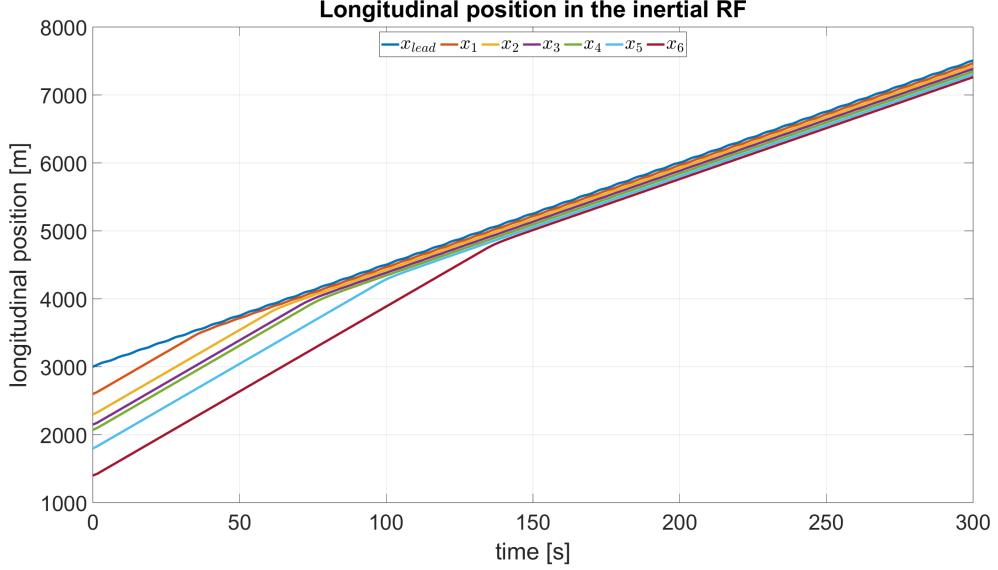


Figure 23: Vehicles longitudinal position evolution - Platoon

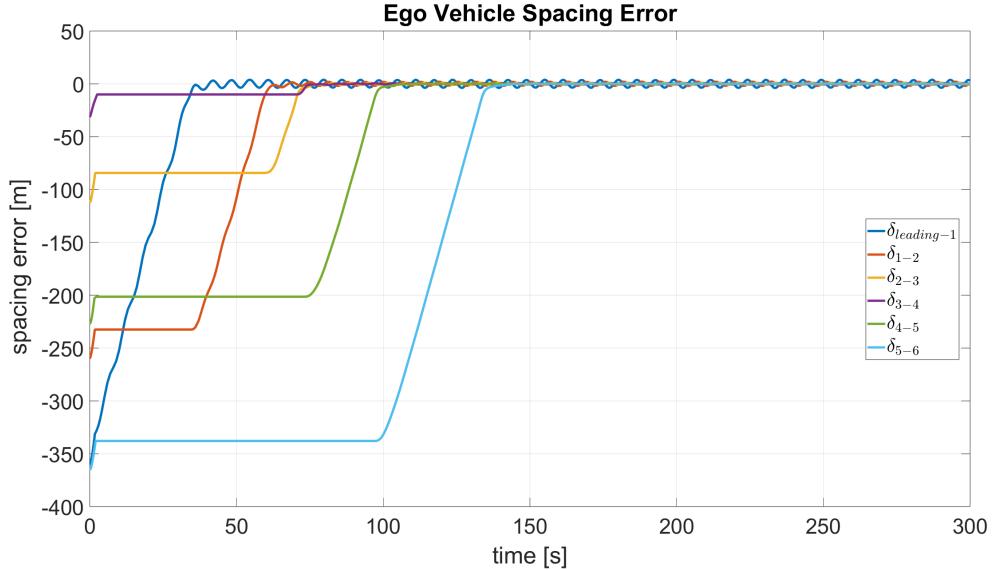


Figure 24: Spacing error between vehicles - Platoon

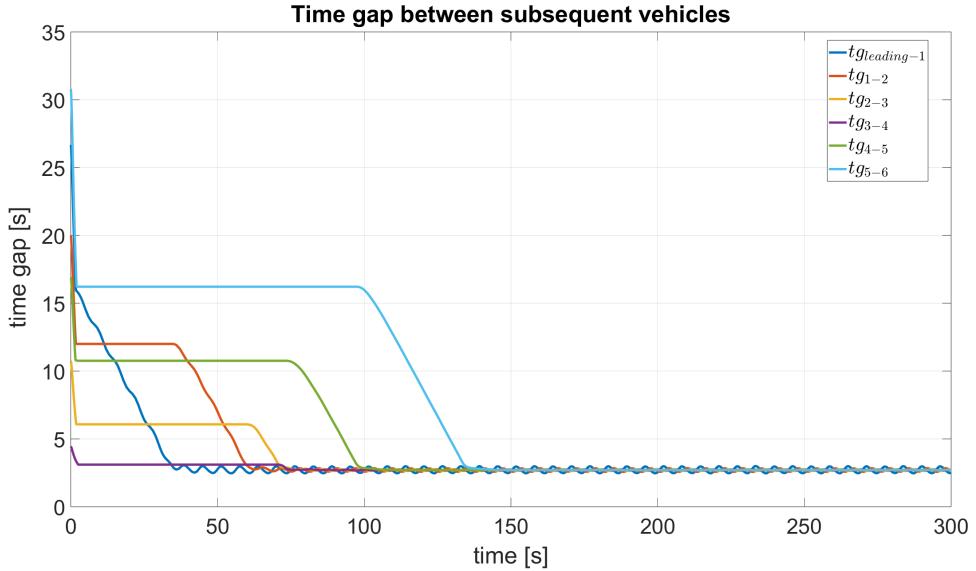


Figure 25: Time gap between vehicles - Platoon

However, this initial verification only confirms the convergence of each individual vehicle to the desired time gap and to a zero spacing error. It therefore demonstrates individual stability, but provides no information regarding string stability. To completely assess the overall stability of the vehicle string, a more in-depth analysis of the spacing error is needed. In particular, it is important to examine the steady-state behavior of the spacing errors across the platoon to verify that the amplitude of the spacing error does not grow as it propagates from one vehicle to the next. Indeed, even if individual stability is achieved, an increasing spacing error along the string could result in large oscillations for the last vehicles in the platoon, especially in long formations, potentially leading to unsafe conditions or even collisions. To address this, a zoomed-in analysis of the spacing errors is performed, as shown in Figure 26. As can be clearly observed, the spacing error does not increase with the number of vehicles; on the contrary, it gradually decreases, thereby demonstrating string stability. This behavior also indicates that oscillations in relative distance between vehicles are damped as they propagate. Consequently, this leads to smaller oscillations in the vehicle speeds as well, as shown in Figure 27, resulting in smoother acceleration and speed profiles throughout the platoon.

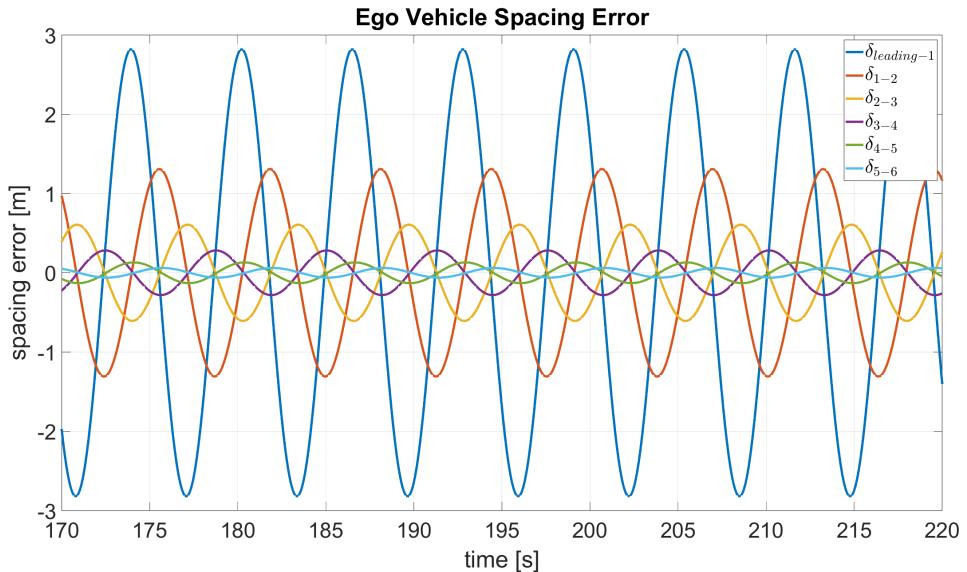


Figure 26: Zoom-in view of the spacing error between vehicles - Platoon

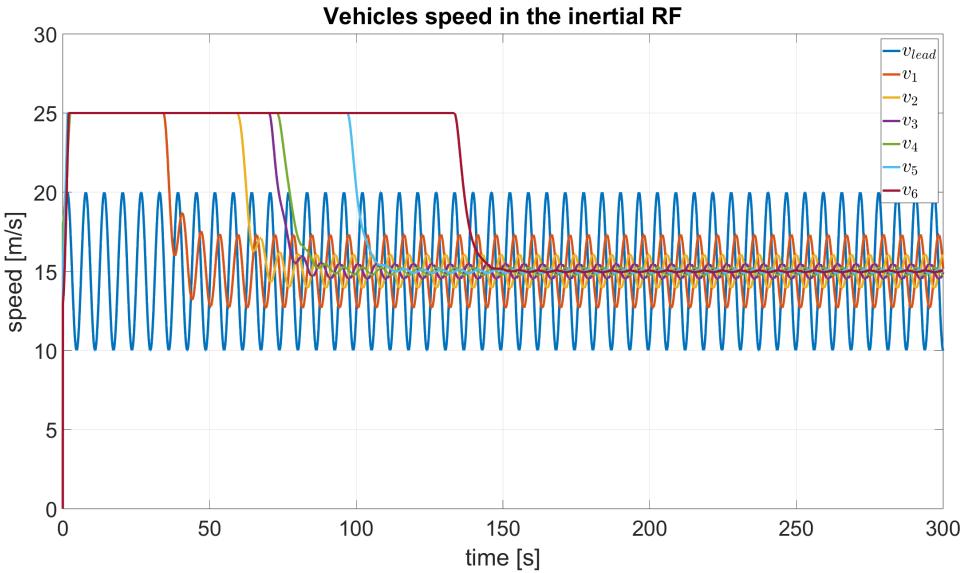


Figure 27: Vehicles speed - Platoon

3.1 Analysis of the plateaux behavior

As shown in Figure 24, the spacing error functions exhibit a plateau at the beginning of the simulation, meaning that the controller is unable to reduce the excessive time gap between the vehicles immediately. In this phase, the vehicles are moving at the same constant speed, as also illustrated in Figure 27. This plateau is not the result of improper controller behavior, but rather a consequence of its design. Specifically, the controller incorporates saturation limits on both speed and acceleration to prevent aggressive responses and to account for the physical constraints of the simulated vehicle. As a result, even if the controller demands a speed greater than the vehicle's maximum one, the saturation block caps the command at the predefined limit of 25 m/s. Being the initial relative distances between vehicles significantly larger than the desired time gap, all vehicle controllers set the maximum allowable speed in order to close the gap as quickly as possible. This leads to a condition in which all vehicles travel at their maximum speed, causing the spacing error to remain constant during this phase until the gap is recovered. This behavior is also reflected in the longitudinal position plot, where vehicle trajectories display a linear trend, which is characteristic of a constant speed motion.

It's important to note that this plateau occurs because the simulation uses identical vehicles with the same saturation constraints. If the vehicles had different saturation values, their speeds would differ, leading to varying spacing errors and differing slopes in the position trajectories.

To eliminate the plateau effect, two options can be considered. The first is to remove the saturation blocks. Without these constraints, the controller can set whatever acceleration or speed value necessary to minimize the spacing error as quickly as possible. As shown in Figure 28, this leads to convergence within a few seconds, much less than the roughly 150 seconds needed in the case with saturation. However, this solution is not practical. As previously discussed, it would negatively impact passenger comfort and neglect the vehicle's physical limitations. This is evident in Figure 29, where speeds ramp up to 100 m/s, clearly unrealistic for real-world applications. Therefore, a viable solution could be to adjust the initial positions of the vehicles according to the desired time gap, placing the vehicles closer together at the start of the simulation.

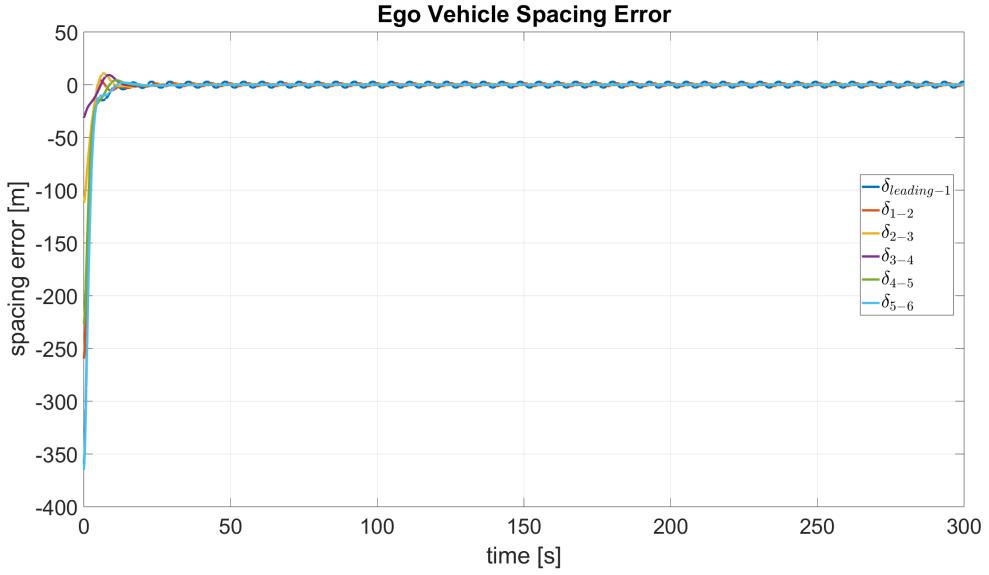


Figure 28: Spacing error between vehicles without saturation - Platoon

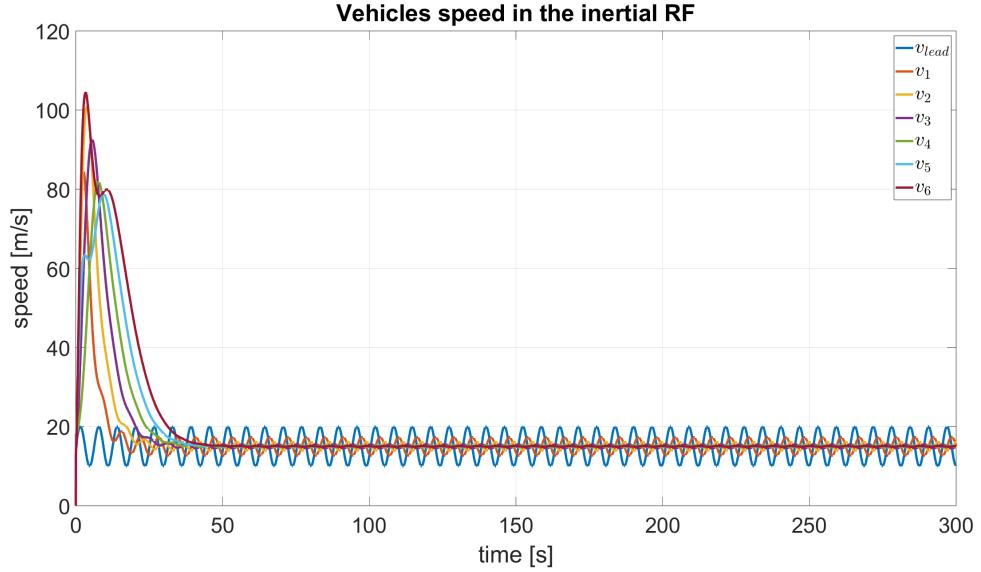


Figure 29: Vehicles speed without saturation - Platoon

3.2 Controller failure analysis

In this section, the controller failure scenario is analyzed to identify the critical issues and assess their potential impact on the real-world implementation of the designed ACC system. The chart below illustrates the longitudinal positions of the vehicles over time. It can be observed that the reference trajectory of the leading vehicle consists of a combination of ramp signals with varying slopes. A closer inspection reveals that the trajectories of the leading vehicle and the first following vehicle intersect, indicating a collision between the two. This critical event is highlighted in Figure 31.

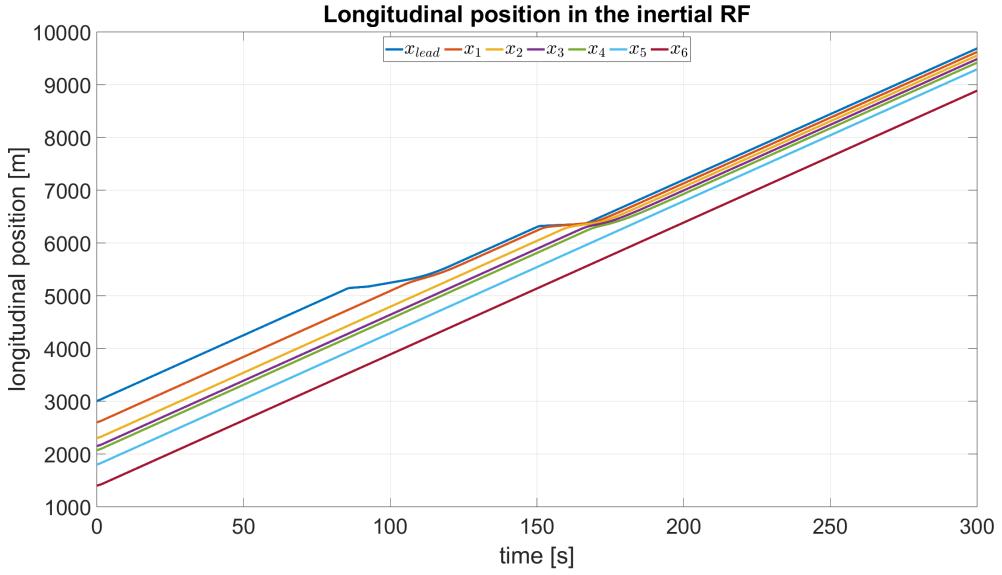


Figure 30: Vehicles longitudinal position evolution - Controller failure

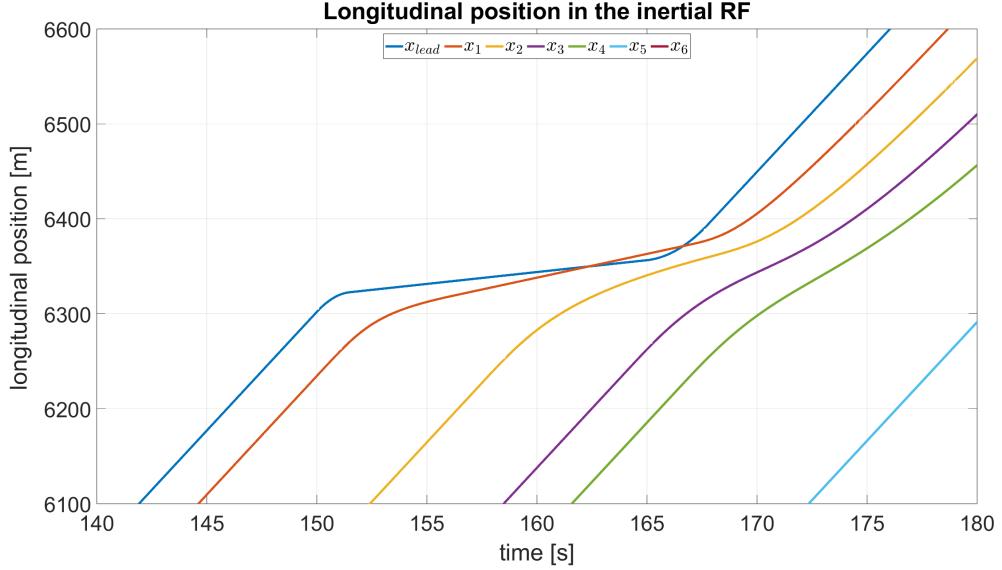


Figure 31: Zoom-in view of the vehicles longitudinal position evolution - Controller failure

It is evident that the collision between the leading vehicle and the first follower is caused by a sudden and significant reduction in the speed of the leading vehicle. This is clearly observable through the sharp decrease in the slope of its longitudinal position curve. The large deceleration imposed by the leading vehicle overwhelms the controller of the first follower vehicle, which is not capable of responding quickly enough to the abrupt change in speed, ultimately resulting in a crash. To better assess the rapidity of this speed change, the acceleration profiles of the vehicles in the platoon are analyzed. It becomes immediately apparent that, just before the collision occurs, the leading vehicle undergoes a pronounced deceleration phase, reaching values as high as 15 m/s^2 . However, only the first follower vehicle ends up colliding. The acceleration curves of the remaining follower vehicles show that the deceleration signal becomes progressively damped as it propagates through the platoon. This attenuation allows the controllers of the other vehicles to react more effectively, preventing further collisions.

It is also worth noting that the collision at this specific time is not the first critical event in the scenario. Around $t = 80 \text{ s}$, the leading vehicle again experiences a sharp deceleration. However, in that instance, the first follower vehicle manages to avoid a crash, primarily due to the greater distance separating it from the leading vehicle compared to the distance at the moment of the actual collision.

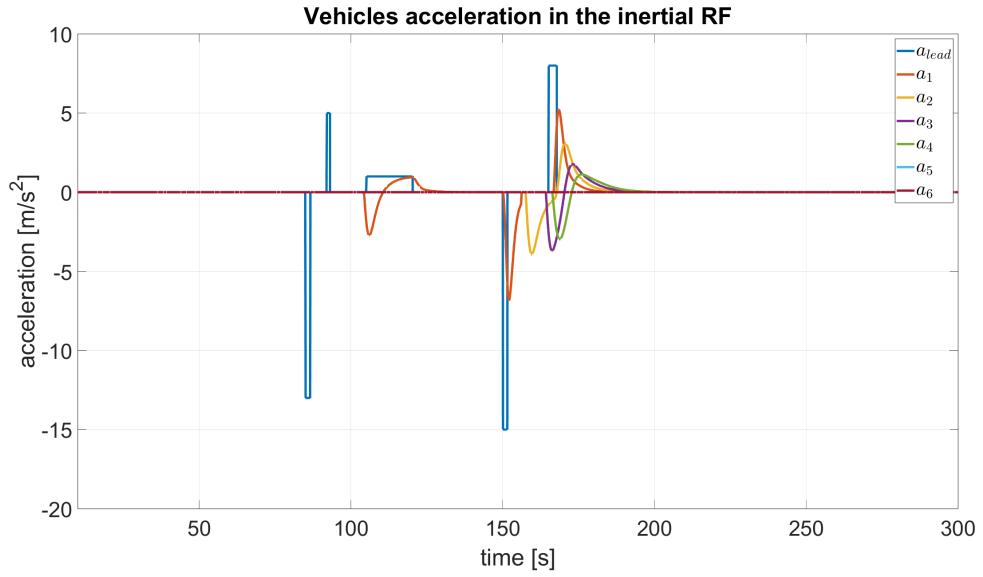


Figure 32: Vehicles acceleration - Controller failure

Although this behavior might appear to compromise the validity of the controller, it is important to note that the extreme deceleration imposed by the leading vehicle does not reflect realistic driving conditions. Under normal operation, the maximum deceleration a vehicle is typically subjected to is approximately $7\text{-}8 \text{ m/s}^2$. Therefore, the scenario analyzed represents a highly unrealistic stress test rather than a common real-world situation.

4 Lower level controller

Once the stability of the CTG controller, both at the individual vehicle level and within a platoon, has been assessed, the controller previously designed can be leveraged to further develop the Adaptive Cruise Control. It is important to note that the CTG controller operates as a high-level controller, as it solely determines the desired acceleration command sent to the vehicle model. In the previous sections, the vehicle model used, as shown in Figure 4 was a simplified representation implemented as a transfer function. This function introduces a delay, emulating the vehicle's response time to acceleration commands, without explicitly modeling the underlying mechanisms, namely, the throttle and brake commands, required to achieve the desired acceleration.

The following section focuses on the implementation of the low-level controller, which is responsible for computing the appropriate throttle and brake commands to achieve the acceleration target set by the CTG controller. Figure 33 shows the Simulink block diagram developed for this purpose. A key addition in this implementation is the low-level control block, which receives as input the error between the reference acceleration and the actual vehicle acceleration, and outputs the throttle and brake commands, h_a and h_b , respectively. Another significant difference compared to the Simulink block diagram used in the previous sections lies in the vehicle model itself. In this simulation, the model is considerably more complex than the previous one. The current vehicle model incorporates the full longitudinal dynamics of the vehicle, including wheel dynamics and gear shifting logic. Furthermore, the inputs to the vehicle model are no longer acceleration commands. Instead, the model now receives the throttle and brake signals directly, allowing for a more realistic simulation of the vehicle's response.

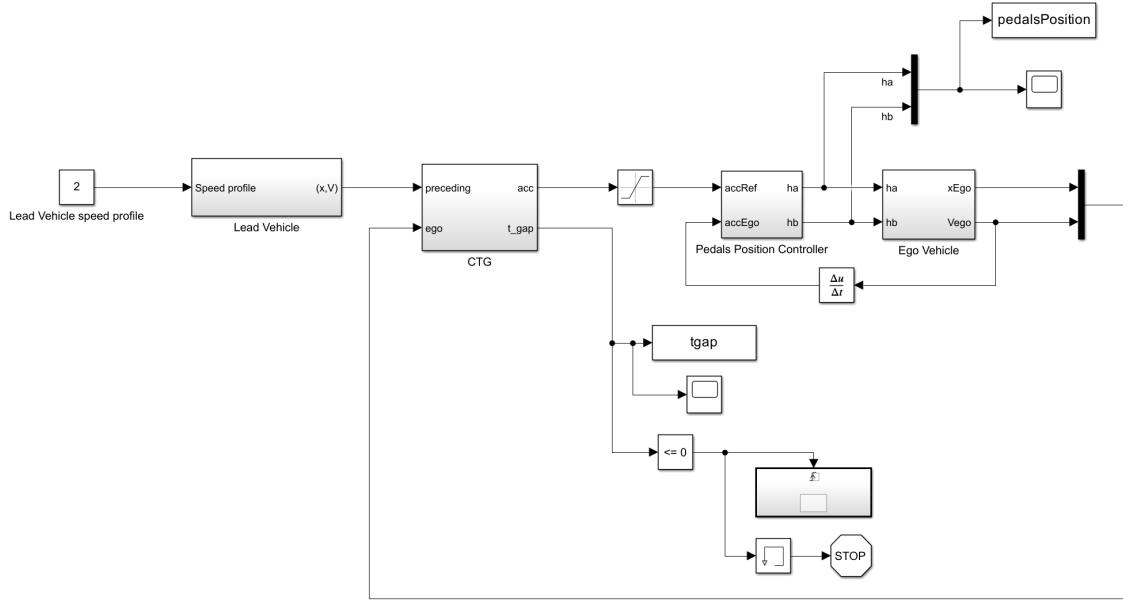


Figure 33: Simulink control block diagram of the complete ACC

The figure below illustrates in detail the pedal position controller block. This component is based on a PI controller, which guarantees null error at steady state, with output saturation between -1 and 1, ensuring a normalized control signal. The input to the controller is the acceleration error, i.e., the difference between the reference and actual acceleration. In addition, the software dynamics are modeled by introducing a delay within the control loop, simulating the real-time execution constraints of embedded systems. The output of this first stage is then split into two separate commands: throttle and brake. Specifically, positive control signals correspond to an acceleration command (h_a), while negative values indicate a braking request (h_b). Since the vehicle model equations are implemented assuming positive values for both throttle and brake inputs, an absolute value block is applied to h_b . This ensures that the final output of the low-level controller consists of two control signals, h_a and h_b , that are both non-negative and compliant with the structure of the adopted vehicle model.

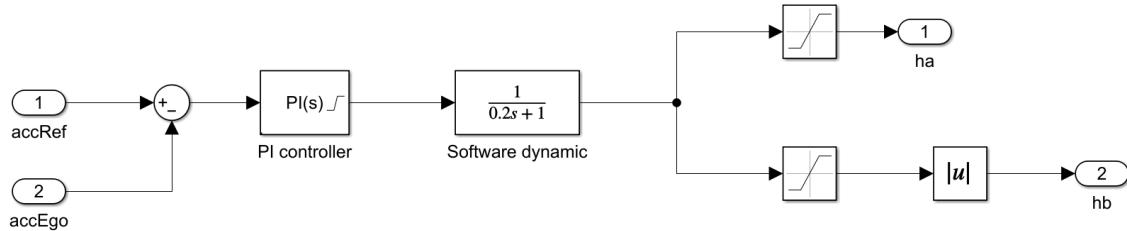


Figure 34: Pedals position controller block

The PI controller is tuned to avoid introducing an additional pole in the loop transfer function. This approach helps to preserve phase margin and system stability, while maintaining a fast and robust response. To ensure this condition, the numerator of the PI controller's transfer function has to cancel out the pole introduced by the software dynamics. Specifically, this results in the condition:

$$k_i = \frac{k_p}{\tau_i}$$

where k_p and k_i are the proportional and integral gains of the PI controller, respectively, and τ_i represents the delay introduced by the software dynamics. In this simulation, τ_i is set to 0.2. The specific values of the PI controller coefficients used in this simulation are provided below.

$$\begin{cases} k_p = 1/3 \\ k_i = 1/3/0.2 \end{cases}$$

To analyze the behavior of the low-level controller, it is useful to compare the results obtained

previously, using only the upper-level controller, with those obtained now, using the same values for the tuning parameters, which are set as follows:

$$\begin{cases} \lambda = 0.5 \\ h = 2.7 \end{cases}$$

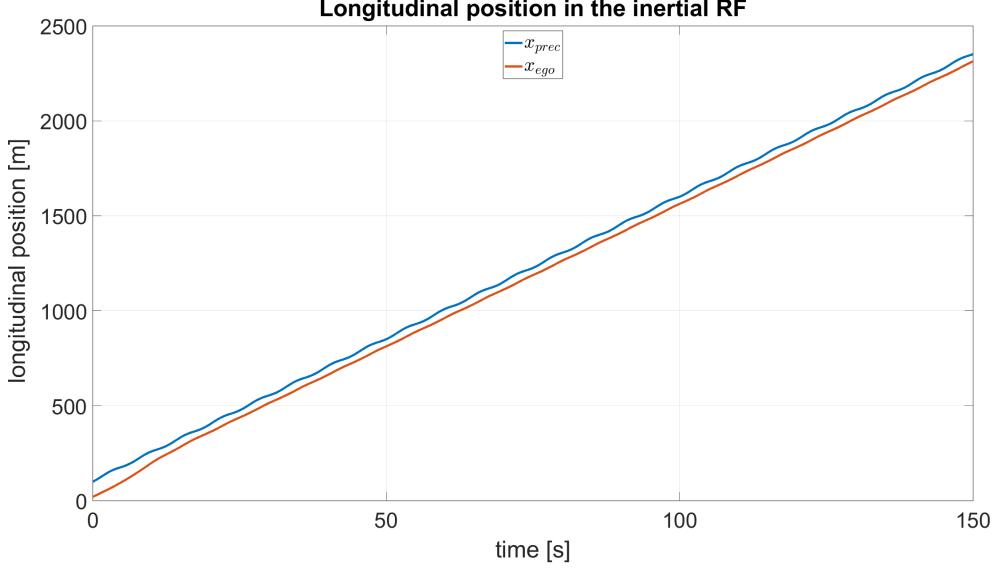


Figure 35: Vehicles longitudinal position evolution - Lower level controller

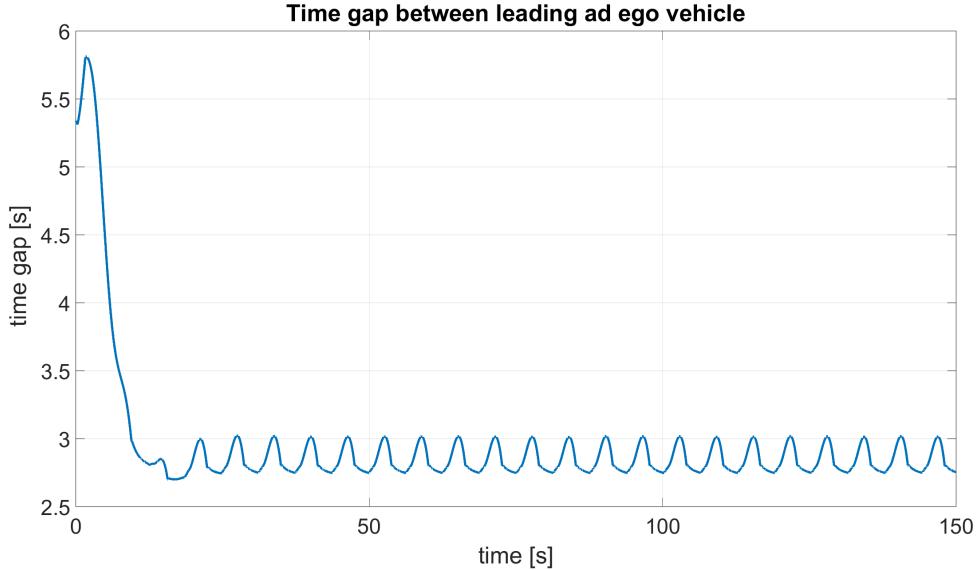


Figure 36: Time gap between leading and followe vehicle

As clearly shown by the previously plotted charts, using the same tuning parameters λ and h results in a good control performance even when the lower-level controller is included in the simulation. Specifically, the time gap converges to the desired value, indicating that the follower vehicle successfully maintains a safe and controlled distance from the lead vehicle. This is further confirmed by the plot showing the evolution of the longitudinal position over time.

In the following chart, the throttle and brake commands are plotted to assess the behavior of the lower-level controller and evaluate its response in terms of the evolution of control actions over time.

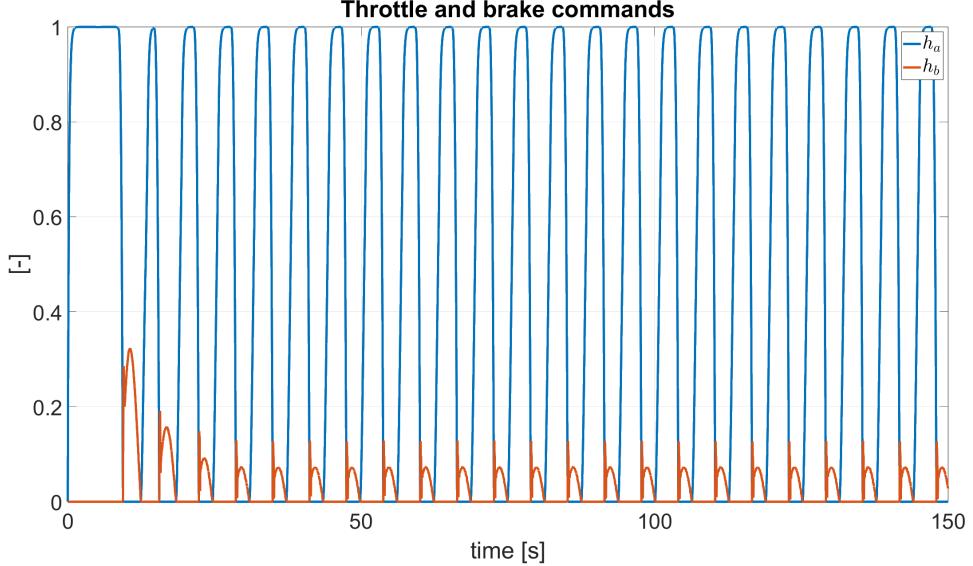


Figure 37: Throttle and brake commands

The first key insight to verify the correct implementation of the lower-level controller concerns the non-simultaneous activation of the throttle and brake commands. As shown in the chart, at each time instant, only one of the two commands is active. This behavior confirms that the controller avoids applying both throttle and brake at the same time, which would be physically inconsistent and inefficient. Furthermore, a clear periodic alternation between throttle and brake commands can be observed. This alternation aligns with the evolution of the leading vehicle's speed profile, which in this simulation is defined by a sinusoidal variation with an amplitude of 15 m/s, a very large and unrealistic value that would not occur under real-world conditions. Although this oscillating behavior may suggest discomfort for passengers due to frequent acceleration and deceleration phases, it actually demonstrates the controller's responsiveness and effectiveness in tracking the reference signal, thereby validating the implemented model. It is also worth noting that the throttle signal ranges from 0 to 1, indicating the full span of throttle pedal input. In contrast, the brake signal, in steady-state conditions, never exceeds 0.2. This behavior might seem counterintuitive, given that the speed variation is symmetric in both positive and negative directions. However, this asymmetry is explained by the fact that the longitudinal vehicle model also includes aerodynamic drag and rolling resistance forces, which naturally assist in decelerating the vehicle. As a result, less braking effort is required, which justifies the lower brake input levels and the corresponding value of the brake gain parameter, h_b .

4.1 Effect of controller parameters on vehicle response

In this section, the influence of the parameters λ and h on the controller behavior is analyzed. Since the main difference compared to the simulations discussed in Section 2 lies in the lower-level controller, to avoid repeating previously explained concepts and to focus only on new insights, the only chart analyzed here is the one representing the throttle and brake commands. Figure 38 shows these two signals for a very small time gap value. Compared to the previous case, where the tuning parameters were set to optimal values, it is evident that the brake is used more intensively. This behavior is consistent with the system's dynamics: a smaller time gap implies that the follower vehicle remains closer to the leading vehicle. Consequently, to prevent a collision, the follower vehicle's controller must respond more aggressively to changes in the leader's speed, resulting in stronger brake commands. A similar trend is observed in the simulation with a large value of λ , as shown in Figure 39. In this case, the controller is more reactive, issuing more aggressive commands in response to speed variations. This increased reactivity again leads to greater brake pedal usage.

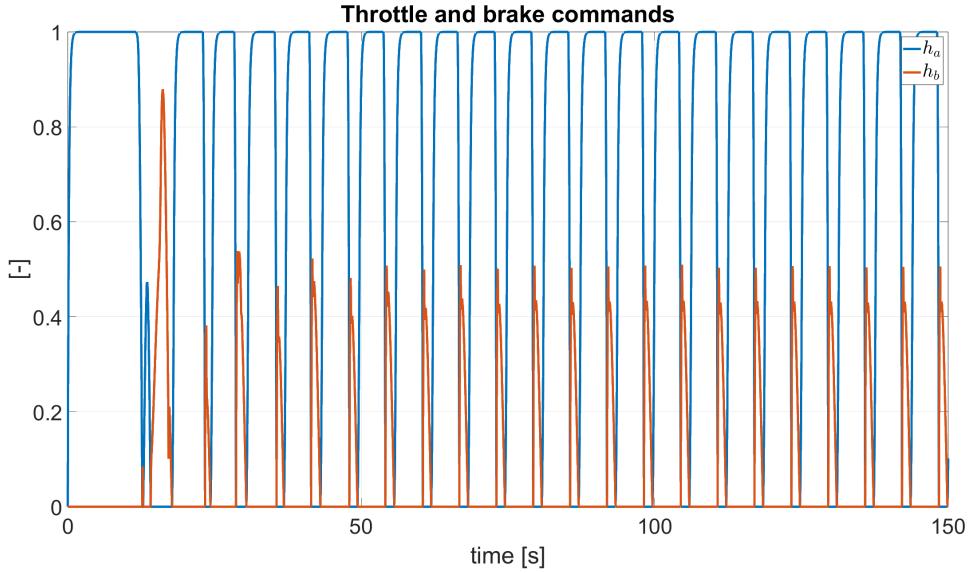


Figure 38: Throttle and brake commands - small time gap

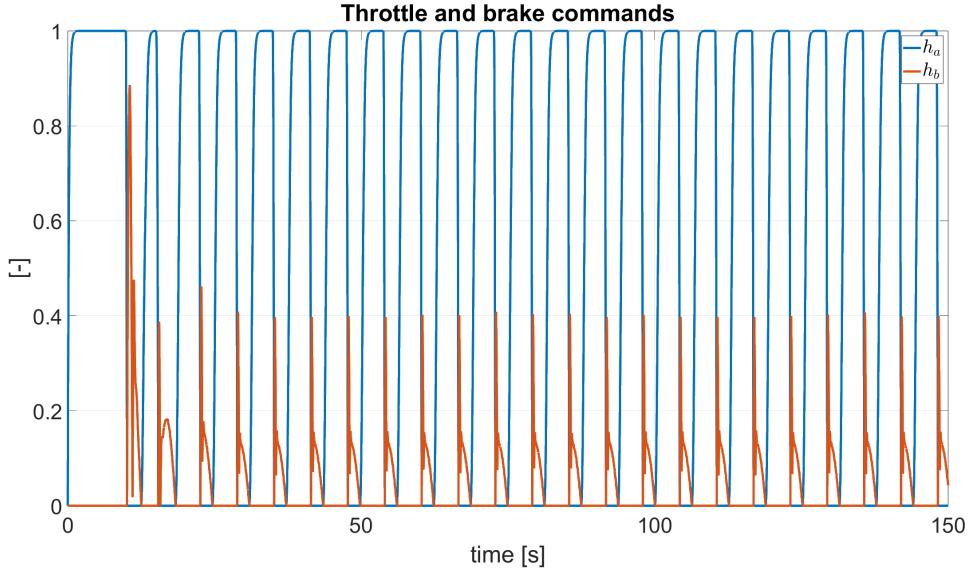


Figure 39: Throttle and brake commands - more aggressive controller

It is important to note that, although varying both parameters leads to the same outcome, the underlying physical reasons for this behavior are fundamentally different. In the first case, the increase in the brake command results from a physical limitation of the simulated system: the braking action is triggered to prevent the follower vehicle from colliding with the lead vehicle. In contrast, in the second case, the rise in the brake command is driven by a controller parameter and is not directly tied to a real-world condition. That is, there is no actual risk of collision, and the controller's response does not reflect a reaction to a hazardous situation. Instead, it reflects a design choice rather than a physical necessity, and the controller's actions are not driven by a real-life dangerous situation.

4.2 Response analysis to a random leading vehicle speed profile

As previously mentioned, the speed profile of the leading vehicle used in the earlier simulation does not reflect typical real-world driving conditions. Therefore, in this section, a more realistic speed profile for the leading vehicle is adopted in order to evaluate the performance of the lower-level controller under conditions that better resemble real-world scenarios. Specifically, the speed profile of the leading vehicle is modeled as a random signal with limited variance. The main results are presented in the following charts.

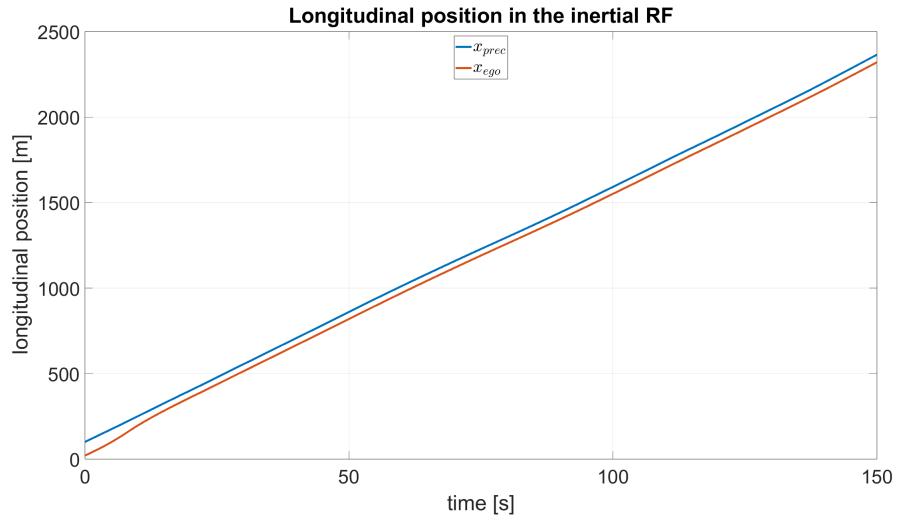


Figure 40: Vehicles longitudinal position evolution - Random speed profile

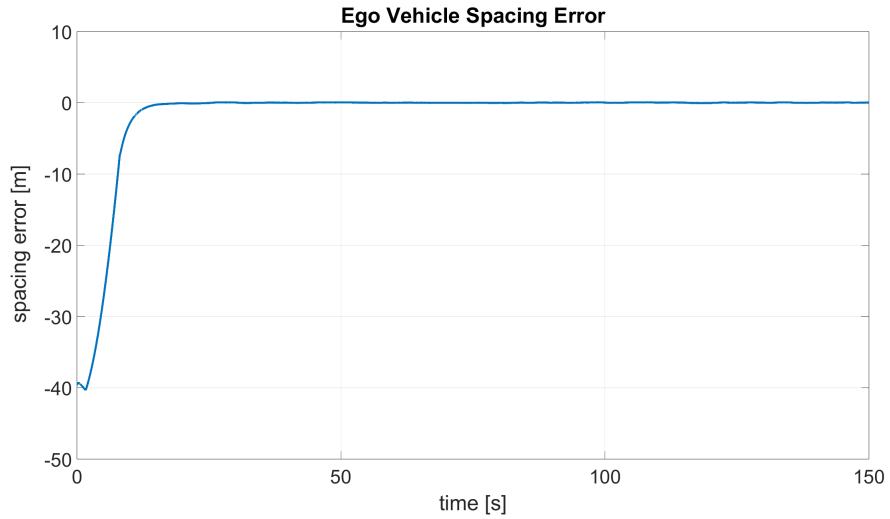


Figure 41: Spacing error - Random speed profile

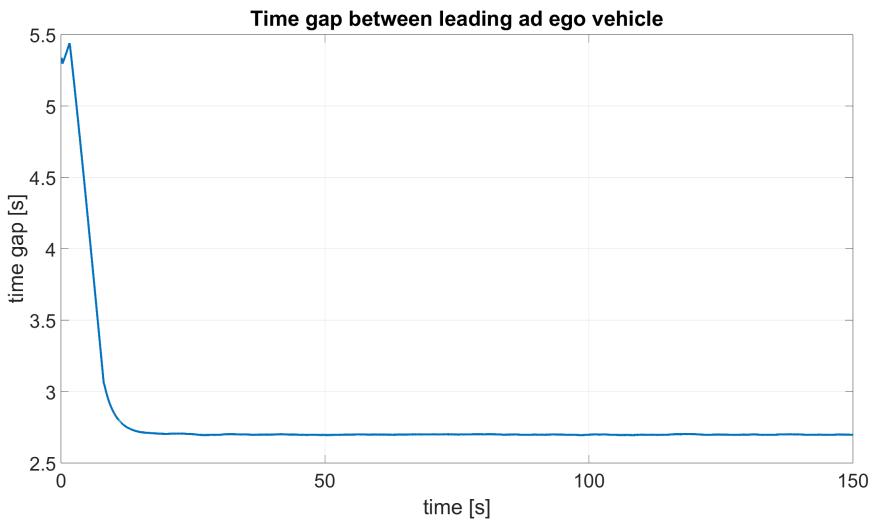


Figure 42: Time gap - Random speed profile

The charts clearly demonstrate the effectiveness of the controller in handling the random speed profile of the leading vehicle. Both the spacing error and the time gap converge to their desired

values, zero for the spacing error and 2.7 seconds for the time gap. This performance is further confirmed by the longitudinal position plot, which shows that the distance between the vehicles remains nearly constant throughout the entire simulation.

As previously discussed, the analysis of the pedal commands provides additional insight into the behavior of the lower-level controller. The corresponding plot is shown in Figure 43. Unlike the results obtained with the sinusoidal speed profile, this chart reveals a remarkable different behavior. Aside from the initial transient phase, primarily influenced by initial conditions, the vehicle's speed is regulated only through throttle adjustments, with the brake command remaining steadily at zero. This behavior is consistent with real-world driving conditions, where resistance forces such as aerodynamic drag and rolling resistance contribute to deceleration. As a result, slight reductions in throttle level variations are often sufficient to decrease vehicle speed, eliminating the need for active braking. Naturally, this is only feasible when the required deceleration is small, a condition commonly found in low-traffic scenarios and highway environments, where drivers are most likely to engage Adaptive Cruise Control. Therefore, it can be reasonably concluded that this final simulation represents the most realistic operating condition for the controller across the majority of its usage time.

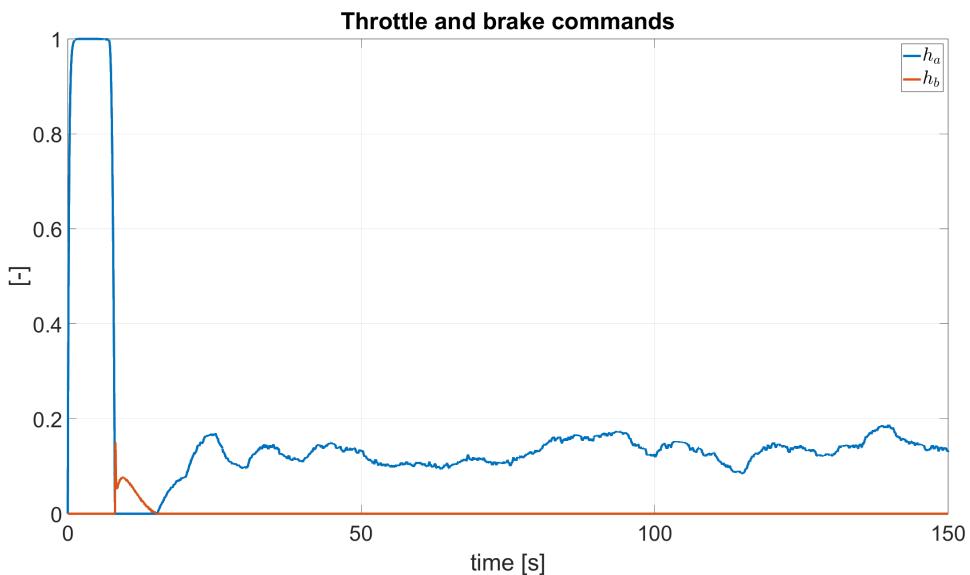


Figure 43: Throttle and brake commands - Random speed profile

5 Lower level controller - Platoon stability

Once the individual stability of the lower-level controller has been verified, it is necessary to assess whether stability can also be maintained within a platoon. To this end, the same simulation described in Section 3 is employed, with the addition of another vehicle at the end of the string, implementing the designed lower-level controller. The following plots, showing the longitudinal position, time gap, and spacing error, demonstrate that each vehicle in the platoon is able to achieve individual stability.

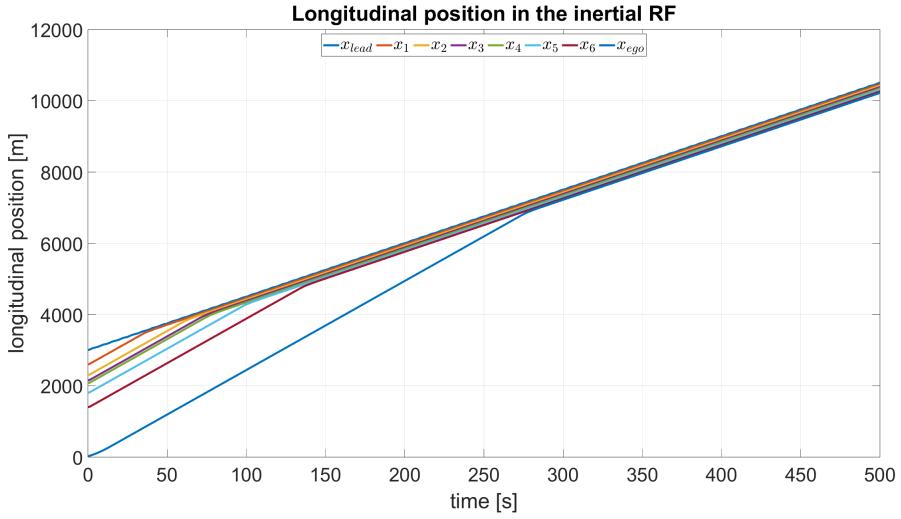


Figure 44: Vehicles longitudinal position evolution - Lower level controller, platoon stability

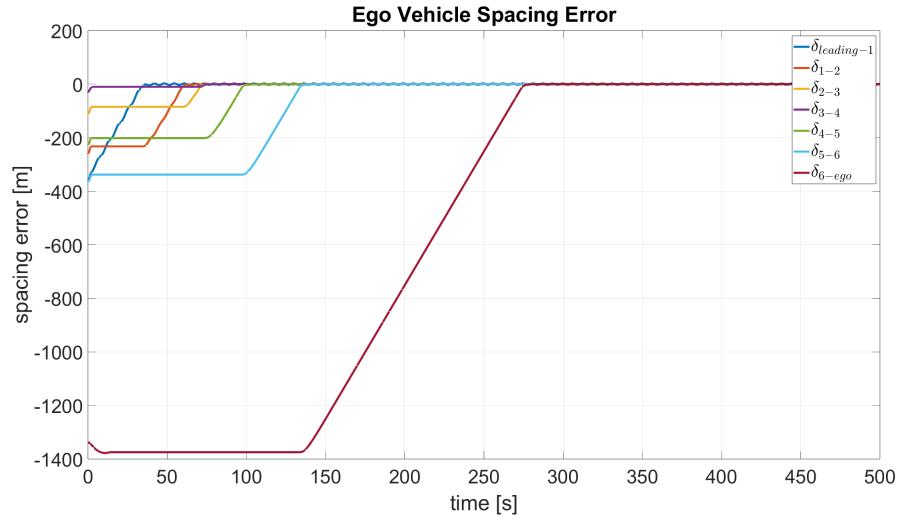


Figure 45: Spacing error - Lower level controller, platoon stability

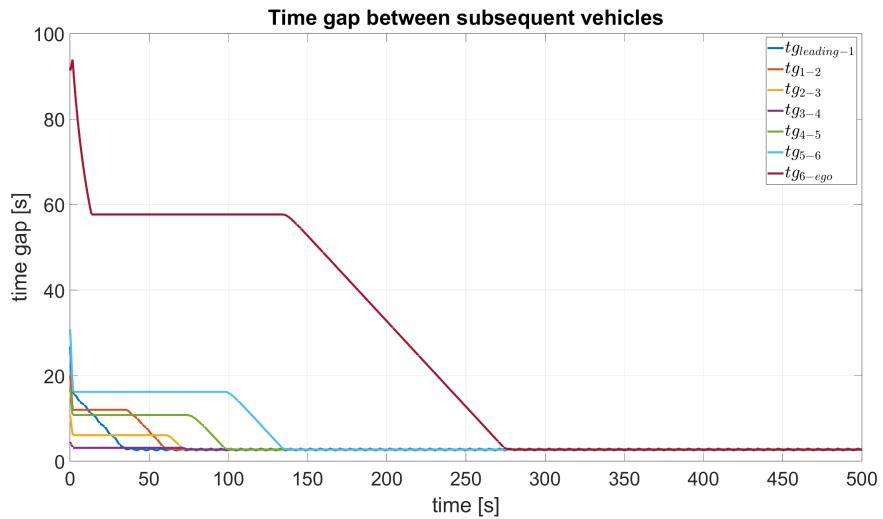


Figure 46: Time gap - Lower level controller, platoon stability

To properly assess platoon stability, it is crucial to examine the spacing error. Indeed, platoon stability is guaranteed only if the spacing error does not amplify as it propagates along the vehicle

string. To verify this condition, a detailed analysis of the spacing errors at steady state is required. Figure 47 presents a zoomed-in view of the spacing error. It can be observed that the error amplitude decreases as it propagates through the platoon. This last analysis provides an important additional insight: the lower-level controller preserves both individual and platoon stability.

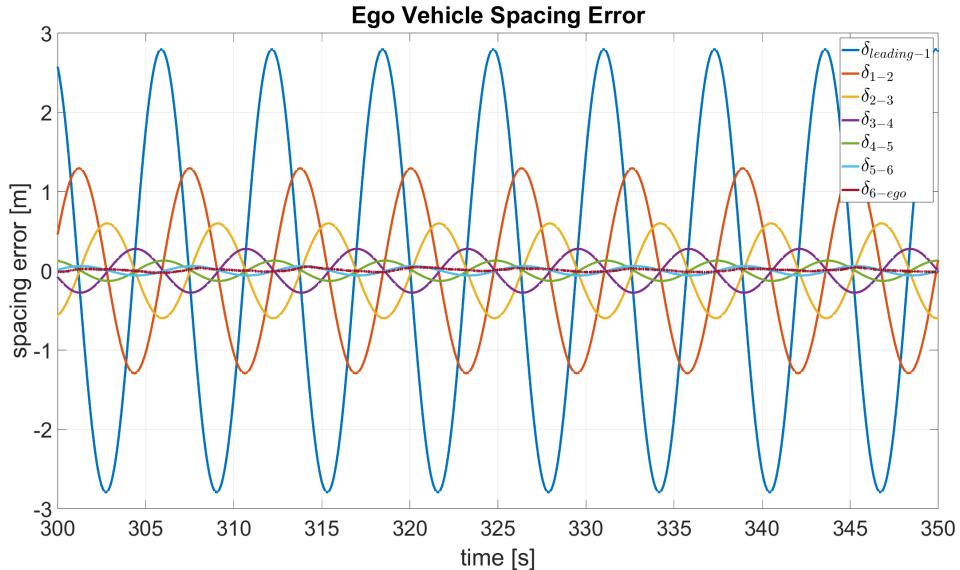


Figure 47: Zoom-in view of the spacing error - Lower level controller, platoon stability

The final chart illustrates the evolution of the throttle and brake commands. Two distinct phases can be identified. The first phase, which ends around $t = 270$, is characterized by sustained full-throttle operation. The second phase displays a smaller, slightly oscillating acceleration command. The wide-open throttle condition in the first phase is due to the significant distance between the last vehicle, implementing the lower-level controller, and the vehicle ahead. Since this distance exceeds the one corresponding to the desired time gap, the controller applies maximum throttle to close the gap as quickly as possible. Once the gap is reduced to the desired value, the throttle command decreases substantially.

A key insight from this analysis is the contrast between this behavior and what observed in the simulation corresponding to Figure 38, where the first vehicle directly follows a sinusoidal speed profile. In that earlier case, a sharp alternation between full throttle and braking is evident. In contrast, in the current platoon scenario, the throttle input is smoother and the brake is not engaged at all. This difference is a direct result of platoon stability: as spacing errors reduce along the vehicle string, the speed and acceleration profiles become progressively smoother and less intense. In essence, the platoon acts as a filter, damping the oscillations of the leading vehicle's speed profile. Therefore, in a long vehicle string like the one considered in the current simulation, the speed profile that reaches the last vehicle, the one it must follow, is significantly smoothed out compared to the original, oscillatory profile. This explains why the throttle and brake commands in this case are much more moderate compared to the previous scenario, where the vehicle had to respond directly to the aggressive speed changes of the leader.

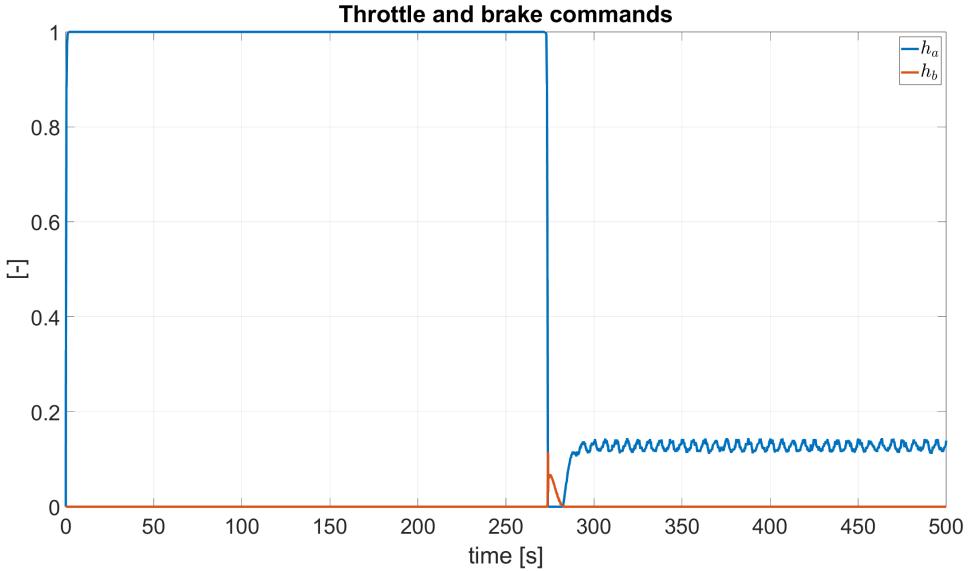


Figure 48: Throttle and brake commands - Platoon stability

5.1 Impact of reduced string size on stability

To gain a better understanding of the filtering effect that the string exerts on speed oscillations, a new simulation is analyzed in this section. In contrast to the previous case, which involved six string vehicles, this simulation considers only two string vehicles, in addition to the leading and ego vehicles. From the zoomed-in view of the spacing error chart, reported in Figure 49, it is evident that both individual and platoon stability are maintained. However, the ego vehicle exhibits a larger amplitude in spacing error, which can be attributed to the reduced filtering effect resulting from the smaller number of vehicles in the string. This diminished damping effect is also reflected in the throttle and brake commands, as shown in Figure 50. Unlike the previously analyzed case, where braking was never required for speed control, the current scenario shows an alternation between throttle and brake usage. The throttle reaches up to approximately 30% of WOT, while the brake command remains limited to very small values.

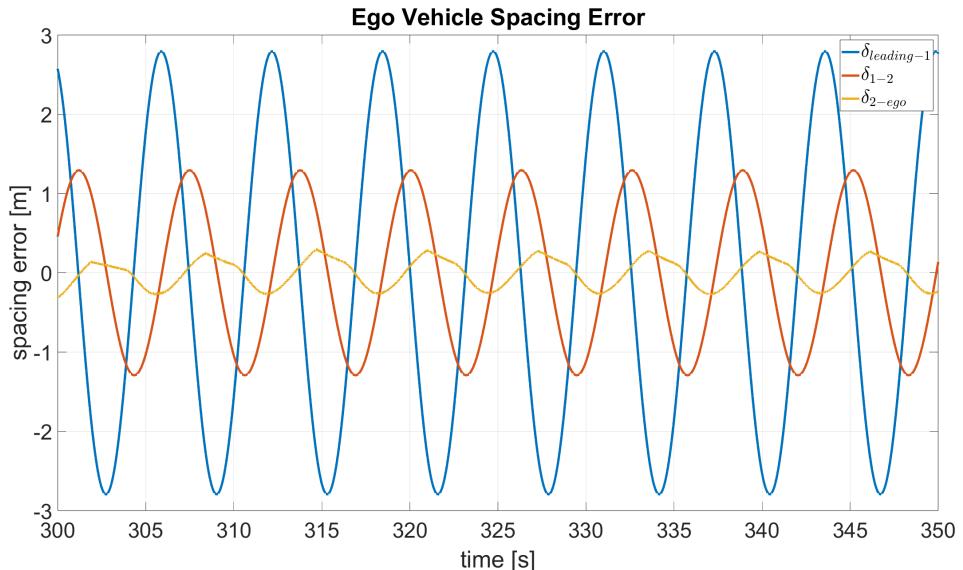


Figure 49: Zoom-in view of the spacing error - Lower level controller, 2 string vehicles

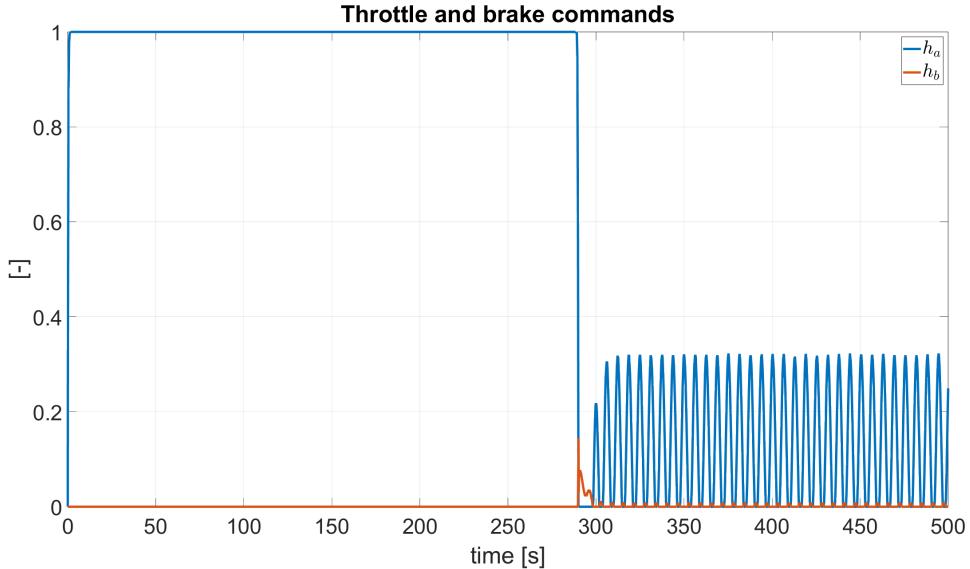


Figure 50: Throttle and brake commands - Two string vehicles

5.2 High-speed profile tracking

This section analyzes the results obtained using the same leading vehicle speed profile as in Section 3.2, with a string composed of six vehicles. The evolution of the vehicles' longitudinal positions shows that the ego vehicle is unable to catch up to and follow the leading vehicle's speed. This is further confirmed by the spacing error between the ego vehicle and the one ahead, which never converges to zero. As in the scenario discussed in Section 3.2, the ego vehicle's limited acceleration capability, due to powertrain constraints, prevents it from closing the gap with the rest of the platoon when the leading vehicle follows a high-speed profile. This condition results in the powertrain operating continuously at Wide Open Throttle, as illustrated in Figure 53. In this figure, it can be observed that the brake command remains at zero throughout the simulation, while the throttle command consistently reaches its maximum value.

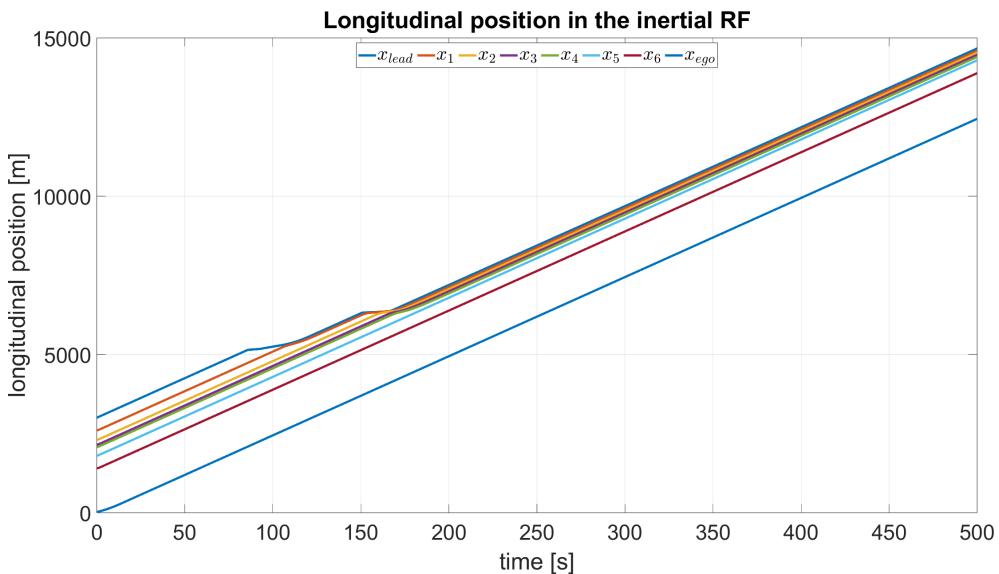


Figure 51: Vehicles longitudinal position evolution - Lower level controller, high speed profile

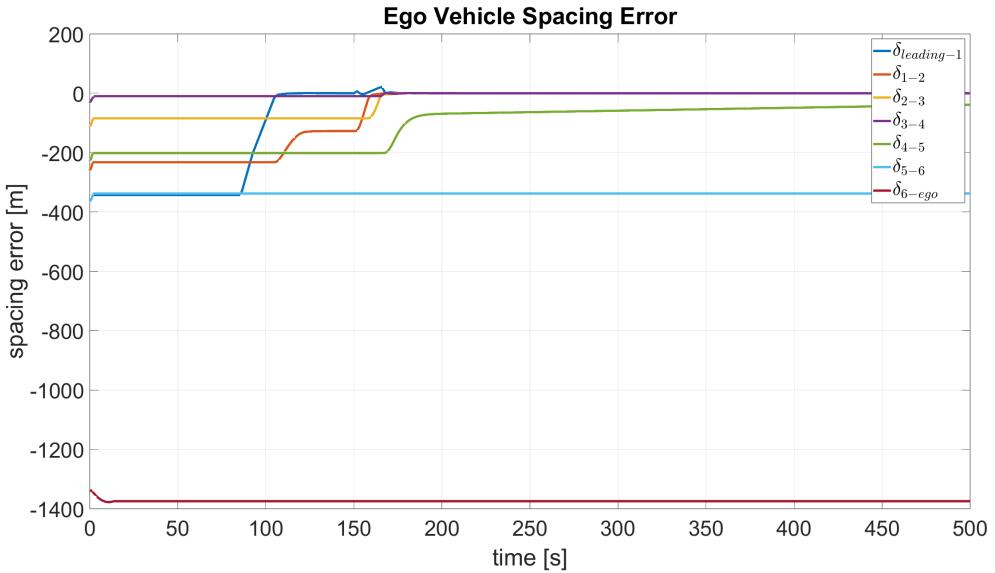


Figure 52: Spacing error - Lower level controller, high speed profile

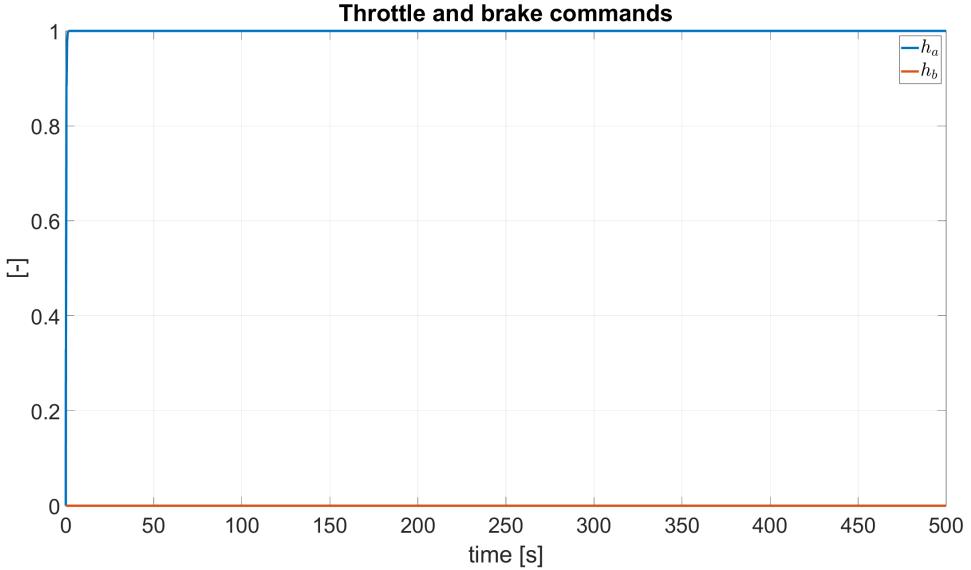


Figure 53: Throttle and brake commands - Lower level controller, high speed profile

6 Conclusions

The analysis carried out in this project demonstrates the feasibility and effectiveness of an Adaptive Cruise Control system based on a Constant Time Gap policy implemented through a PD controller. Initial simulations with a simplified vehicle model verified the controller's ability to maintain safe inter-vehicle distances while complying with ISO 15622:2018 standards. Sensitivity analyses on control parameters and vehicle dynamics identified safe tuning intervals for λ and h , highlighting their interdependence. Further evaluations under both harmonic and random leading vehicle speed profiles confirmed the robustness of the controller. Extension to multi-vehicle platoons proved individual and string stability, with oscillations attenuating along the vehicle string, ensuring smooth and safe traffic flow. The integration of a lower-level controller introduced realistic actuator behavior and preserved performance and stability under more complex dynamics. Even in stress-test conditions with extreme speed profiles or sudden decelerations, the controller generally maintained safety, although physical limitations could lead to failures under unrealistic scenarios. Overall, the designed ACC system demonstrated strong performance across a range of realistic driving conditions and configurations, making it suitable for real-world automotive applications.