Assignment 1 - Wheel Slip Control

Vehicle Dynamics and Control
(RO47017)

Author:

Nicola Visentin (6354815)

May 12, 2025

Contents

Introduction

The assignment consists in the design and simulation of a controller that implements two different wheel slip control (WSC) strategies. The scenario is the following: a car is traveling at a constant speed on a wet asphalt road, when the driver performs an emergency braking. The wheels would block completely and would start slipping, causing the vehicle to loose control and stop after a very long distance. For this reason, the idea is to build a controller that "corrects" the braking action from the driver in order to regulate the slip of the wheel, trying to achieve the best possible braking performance.

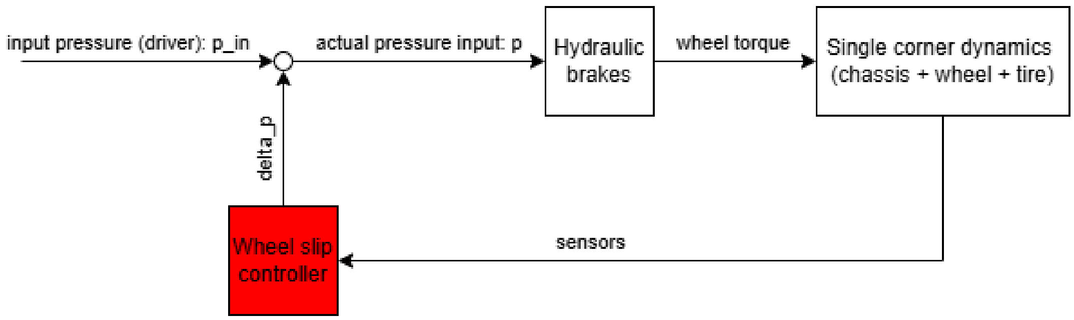


Figure 1: Schematic of the WSC logic.

This situation is modeled in Simulink as shown in Figure 1. The car is traveling with a constant speed of 120 Km/h on wet asphalt (friction coefficient assumed to be constant and equal to 0.6), and after 2 seconds the driver hits the brakes. The "driver's" input signal p_{in}, which represents the pressure to send to the brake actuator, is assumed to go from 0 bar to 100 bar in 1 second, linearly, and to stay at 100 bar for the whole duration of the emergency brake. This signal would completely block the wheels, as previously mentioned, so it is "corrected" by a certain Δp signal provided by the WSC controller. The "corrected" signal $p = p_{in} + \Delta p$ is finally sent to the hydraulic brakes, which have a certain dynamics and that generate a certain braking torque on the wheel. Then, basing on the dynamics of the vehicle (chassis, wheels and tyres), the car will react consequently. The braking maneuver is considered complete when the chassis speed is lower than 10 Km/h.
We are assuming we are able to measure chassis speed and acceleration and also angular speed and angular acceleration of the wheels. This information is used by the controller to compute the "correction" Δp.

This report is organized as follows: in section 1 and section 2 two different approaches for the design of the controller are presented and tested, namely a wheel slip control approach and a mixed slip-deceleration (MSD) approach. Then, noise is introduced in some measurements and the performances of the controllers are evaluated again. Finally, in section 4, the learning elements and the main challenges faced during the controller design are described and conclusions are drawn.

1 Wheel Slip Controller

Motivation In general, the idea for improving the braking performance consists in trying to maintain the longitudinal force between the wheel and the road as high as possible. This force depends on many factors in reality, and can be estimated by choosing a certain model for the tyre. In Figure 2 this force is plotted (using both Dugoff and Delft tyre models) versus the longitudinal slip κ, which is defined, during braking, as:

$$\kappa = \frac{v - \omega \cdot R_{eff}}{v} \quad , \quad v \textit{ chassis speed, } \omega \textit{ wheel angular speed, } R_{eff} \textit{ effective radius of the wheel} \tag{1}$$

As we can see, we don't want the wheels to be completely blocked ($|\kappa| = 1$), but we want to maintain a slip around $|\kappa| \approx 0.12$, which is able to provide the maximum force.

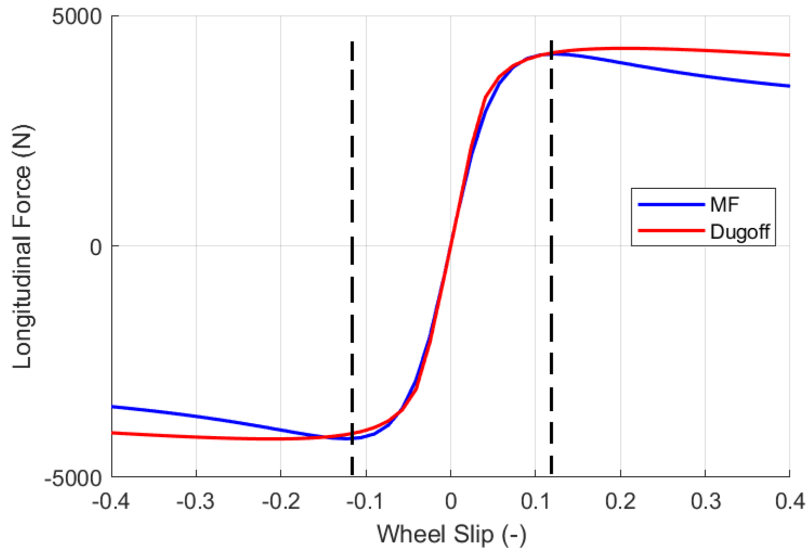


Figure 2: Longitudinal force vs wheel slip using both Dugoff and Delft "magic formula" tyre models. Normal force: 4000 *N, speed:* 60 *Km/h.*

Notice that the plot in Figure 2 actually depends on many parameters (speed, normal load, interaction with the lateral dynamics, etc), and for this reason the best operating point changes in time. However, different tests were made with different parameters, and the peak in the curve was always observed to be around 0.12, which should represent for this reason a good enough approximation.

Design Basing on what we just said, the idea is to design a PID controller that tracks a longitudinal slip $\kappa_{ref} = 0.12$. The regulator is defined as follows:

$$\Delta p = K_P \, e + K_I \int e \, dt + K_D \, \dot{e} \tag{2}$$

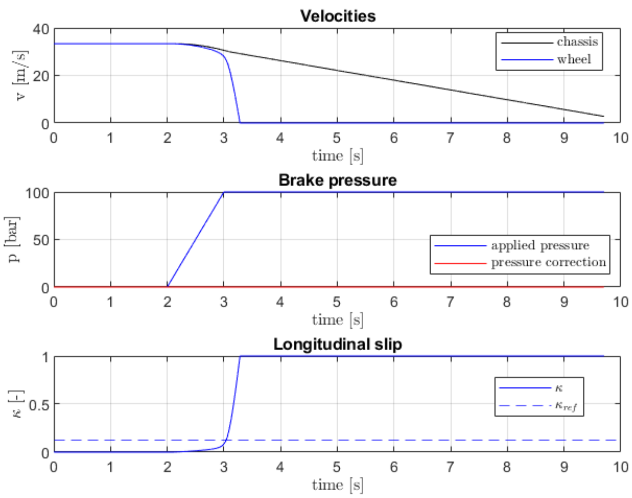
where $K_P, \; K_I, \; K_D$ are the proportional, integrative and derivative gains, while $e = \kappa_{ref} - \kappa$ is the error on the longitudinal slip, computed using data from the sensors and Equation 1. Moreover, a filter on the derivative contribution is added and also an anti-windup mechanism on the integral action. More details about this, along with the Simulink model, are reported in Appendix A. Since the behavior of the tyres is nonlinear, and, as we already explained, its interaction with the road also depends on the speed of the vehicle, gain scheduling was adopted to regulate the gains according to v.
The tuning of the parameters was made following a trial-and-error methodology, also considering that the system is very complex, with strong nonlinearities and a simple PID controller is not able to accurately take into account for this complexity, compared to other more refined controllers. At the end, the following parameters were chosen:

- K_P: linear dependency on the speed, ranging from $K_P = 500$ at $v = 120$ Km/h to $K_P = 100$ at $v = 10$ Km/h. When v is low, the denominator of Equation 1 tends to get bigger, thus we may want to have a less aggressive proportional gain.
- K_I: a simple threshold logic that imposes $K_I = 30$ when $v > 50$ Km/h and $K_I = 70$ otherwise.
- K_D: when used, it is imposed equal to 5 and constant.

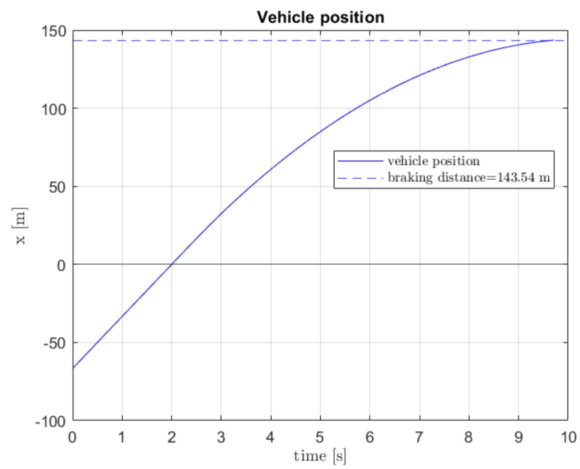
- Derivative filter: $N = 200$ (when K_D is used).
- Back-calculation coefficient for the anti-windup filter: $K_b = 0.8$.

Finally, to avoid interactions of the wheel slip control system with the user's braking input during normal driving conditions (due to numerical reasons or noise), an activation logic for the ABS system allows to activate the controller only during the emergency brake, either after 2 seconds of simulation (when the driver starts pressing the pedal), or when the longitudinal slip crosses a certain threshold.

Simulations First, the system was simulated without any ABS control active. Figure 3 shows the results. As we can see from Figure 3a, the wheel starts slowing down at $t = 2$ s and slips almost immediately. At $t \approx 3$ s, when the driver has pressed the pedal completely, the wheel is blocked ($\kappa = 1$). The braking distance in this case is $d = 143.54$ m.



(a) Chassis and wheel linear velocities (v and $\omega \cdot R_{eff}$), applied brake pressure p and controller output pressure Δp, longitudinal slip κ.



(b) Position of the vehicle. The breaking maneuver starts at $t = 2$ s. Braking distance is reported.

Figure 3: Simulation without ABS control.

Now a PI wheel slip controller is tested. The parameters are the ones reported above and in Appendix A, but no derivative gain is used in this case.

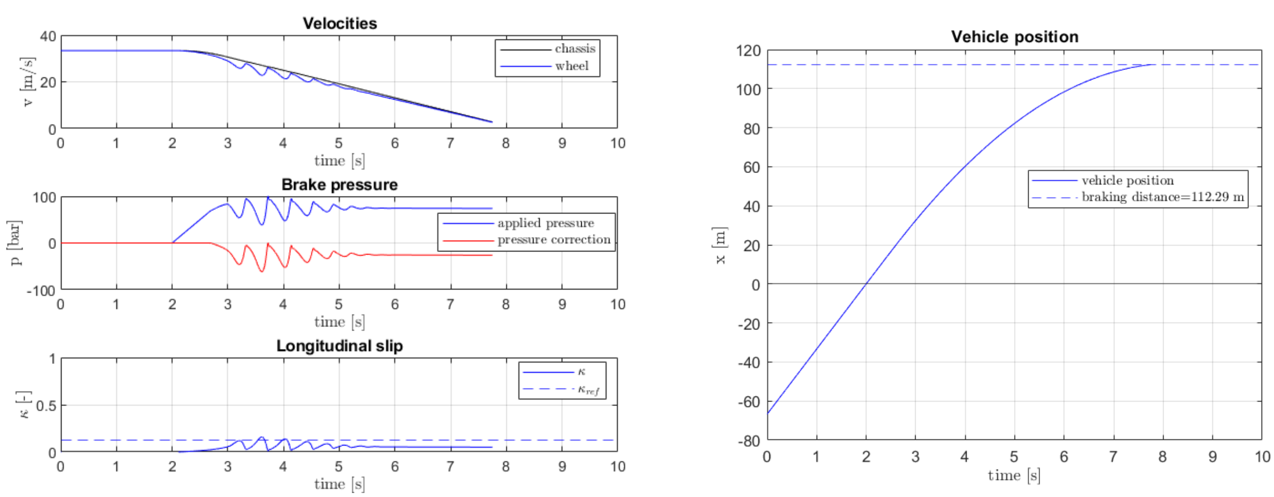


Figure 4: Simulation PI control.

As we can see in Figure 4, the braking distance is now way lower, $d = 112.29$ m, as the controller is trying to keep the longitudinal slip around the optimal value $\kappa \approx 0.12$. Notice that many different choices for the gains could have been taken, obtaining different performances. The final tuning was achieved trying to maintain a fair balance between tracking performances, braking distance and control effort. More considerations about this can be found in Appendix A. It's important to notice, instead, that the anti-windup filter is essential to avoid instability, as can be seen from Figure 5, where the simulation was performed with the exact same parameters, but without anti-windup.

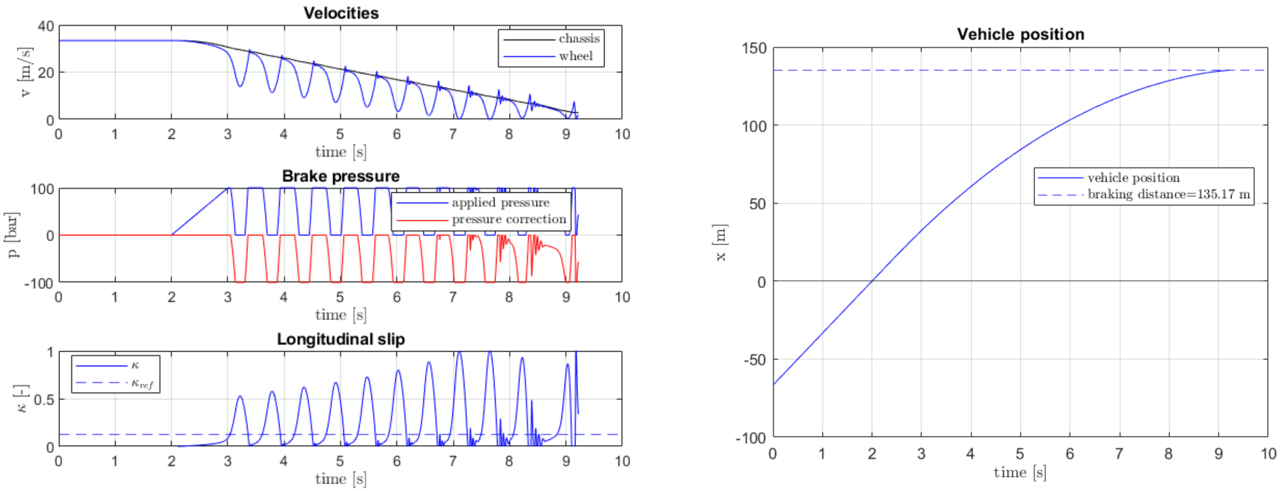


Figure 5: Simulation PI control without anti-windup filter.

Finally, a derivative gain $K_D = 5$ is introduced. It was observed that this can slightly improve the performances, but at the same time it makes the regulator more aggressive, less robust and more sensitive to noise. For this reason K_D must be carefully tuned, trying to avoid instability. The corresponding results are shown in Figure 6. Again, more considerations on the effect of the derivative action can be found in Appendix A.

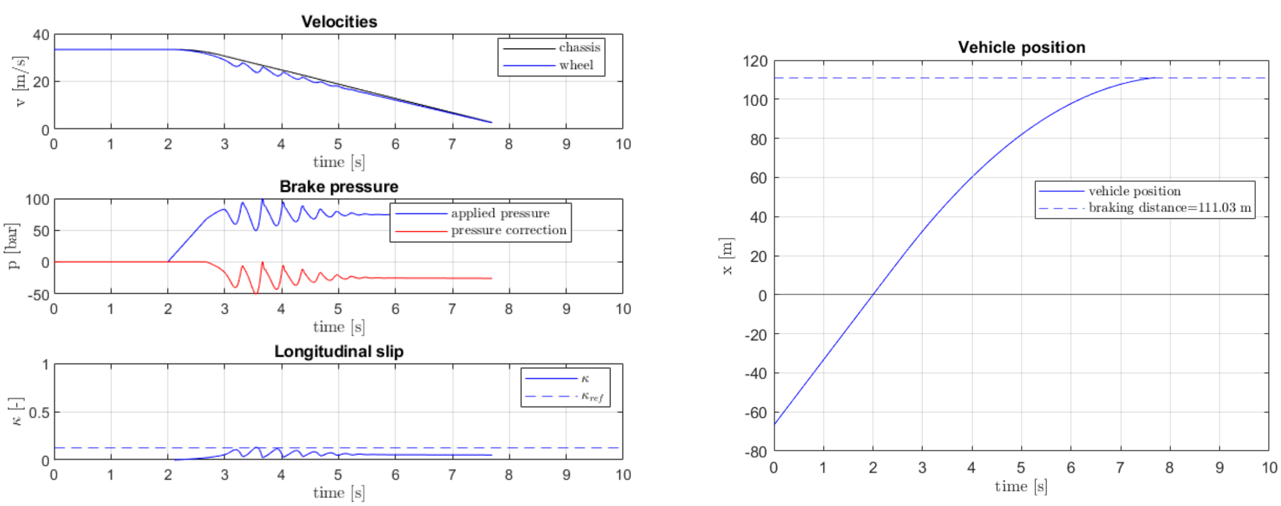


Figure 6: Simulation PID control.

To evaluate each tested control logic, four main factors were considered:

- Braking distance d.

- Control effort: looking at the plot of the applied pressure p (for example if it saturates a lot, if it presents very rapid oscillations, etc).
- Tracking of κ_{slip}.
- Comfort: can be evaluated by computing the Integral Time-weighted Average of the longitudinal jerk:

$$ITAE_{jerk} = \int_{t_0}^{t_f} t\ |j_x|\ dt, \quad \text{where } j_x \text{ is the longitudinal jerk [m/s}^3\text{]}$$

Results for the main tests are reported in Table 1.

Control strategy	**Braking dist. d**	**Control effort**	**Track. of $\kappa_{\mathbf{slip}}$**	**ITAE$_{\mathbf{jerk}}$**[1]
no ABS (Fig 3)	143.54 m	almost null	no	1
PI + anti-windup (Fig 4)	112.29 m	very limited	quite close	4.52
PID (Fig 6)	111.03 m	limited (for low K_D)	faster, smaller oscill.	3.00

Table 1: Performance of different wheel slip controllers.

2 Mixed-Slip-Deceleration Controller

Motivation In this case, the strategy still consists in tracking an optimal κ_{ref}, but now the speed of the wheel is also considered, trying to exploit its variation in time to "predict" the blocking of the wheel itself. The resulting controller is a mixed slip-deceleration controller (MSDC).

Design A PD controller is used to track a longitudinal slip $\kappa_{ref} = 0.12$ through the proportional term, while a derivative contribution involves the tangential speed of the wheel. The regulator is defined as follows:

$$\Delta p = K_P\ e_\kappa + K_D\ \frac{R_{eff}}{g}\ \dot{\omega} \tag{3}$$

where $K_P,\ K_D$ are the proportional and derivative gains, $e_\kappa = \kappa_{ref} - \kappa$ is the error on the longitudinal slip, computed using data from the sensors and Equation 1, and R_{eff} and ω the effective radius and the rotational speed of the wheel. Again, the Simulink model is reported in Appendix B. Gain scheduling is applied also in this case, and the final tuning (again, based on trial-and-error methodology) is:

- K_P: linear dependency on the speed, ranging from $K_P = 400$ at $v = 120$ Km/h to $K_P = 10$ at $v = 10$ Km/h.
- K_D: linear dependency on the speed, ranging from $K_D = 40$ at $v = 120$ Km/h to $K_D = 30$ at $v = 10$ Km/h.

This tuning was made in order to achieve similar braking distances with respect to the ones obtained with the previous controller presented in section 1.
Finally, to avoid interactions of the wheel slip control system with the user's braking input during normal driving conditions (due to numerical reasons or noise), an activation logic for the ABS system allows to activate the controller only during the emergency brake, either after 2 seconds of simulation (when the driver starts pressing the pedal), or when the longitudinal slip crosses a certain threshold.

[1] $ITAE_{jerk}$ is normalized with respect to the value obtained when no ABS is applied.

Simulations Tests were performed similarly to the ones for the wheel slip control (WSC) in section 1. Results are reported in Figure 7 and Table 2.

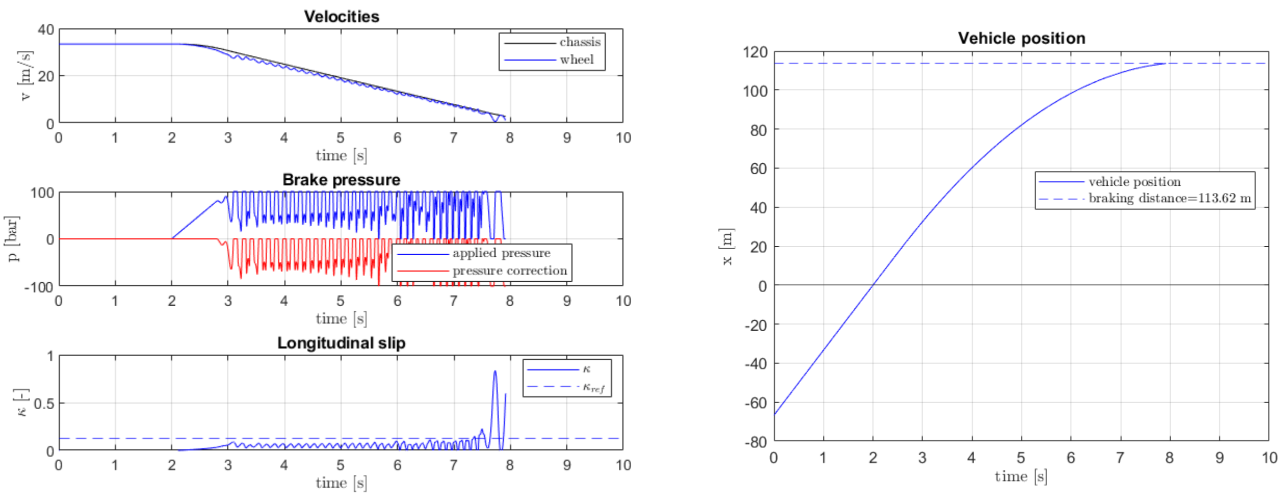


Figure 7: Simulation MSD control.

Control strategy	Braking dist. d	Control effort	Track. of $\kappa_{\mathbf{slip}}$	$\mathbf{ITAE_{jerk}}$[2]
MSDC (Fig 7)	113.62 m	quite high	similar to WSC, but unstable towards the end	30.49

Table 2: Performance of MSD controller.

As we can see from these results, the performance of the MSDC seems to be a bit worse with respect to the previous wheel slip controller. In particular, in order to achieve a braking distance which is comparable to the one for the PI controller in section 1, the gains need to be tuned in a more aggressive way. This also causes an higher control effort and a larger $ITAE_{jerk}$, that affects comfort.
The reason could also be the limited amount of time spent to tune this controller, compared to the PI WSC previously introduced. Moreover, a derivative filter is missing in this case.

3 Controllers performance with noise

In this section, both the controllers presented in section 1 and section 2 are tested in presence of noise in the sensors' measurements. In particular, following the motivations in Appendix A, the PI controller with anti-windup filter is chosen in the first case over the PID controller.
Random noise is introduced on the wheel slip κ (amplitude of 0.025) and on the angular acceleration of the wheel $\dot{\omega}$ (amplitude of 0.5 rad/s^2) as shown in Figure 12b and Figure 15b. The results of the simulations are shown in Figure 8 and Figure 9.
The input pressure signal is clearly more disturbed and, as expected, the performance is worse both in terms of braking distance and jerk. In particular, WS controller seems to be more sensitive to noise, compared to the MSD controller. The reason for that could be that the first approach is only relying on the computation of the wheel slip κ, while the second one is also accounting for the wheel deceleration. Thus, this may cause the WSC to be less robust in presence of uncertainties and disturbances. These considerations could drive the choice of the controller basing on the specific operative conditions.

[2] $ITAE_{jerk}$ is normalized with respect to the value obtained when no ABS is applied.

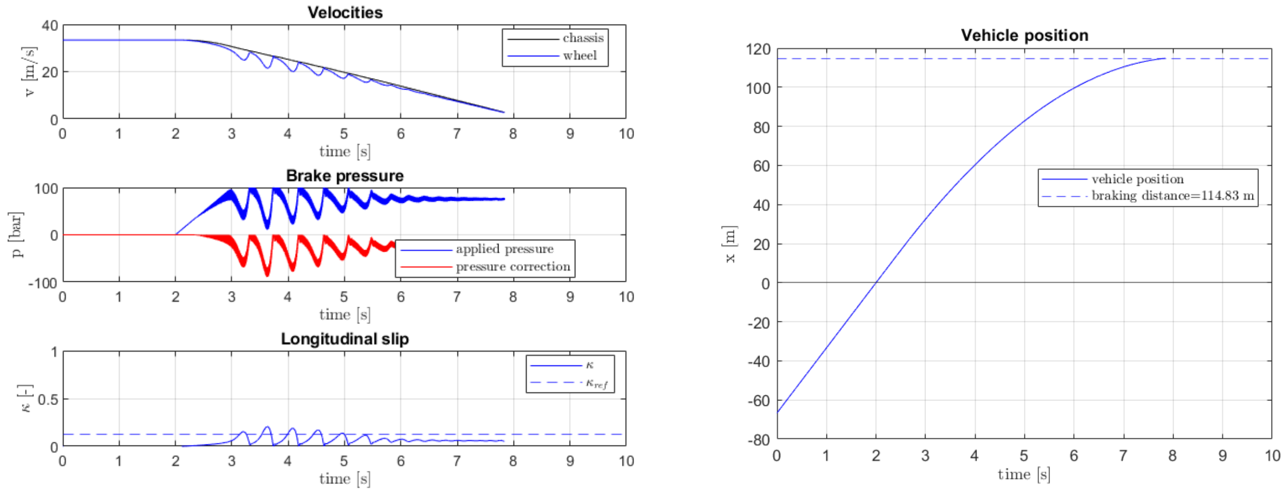


Figure 8: Simulation wheel slip PI controller with noise. Notice that the plot of the slip is accounting for the real κ, before introducing the noise.

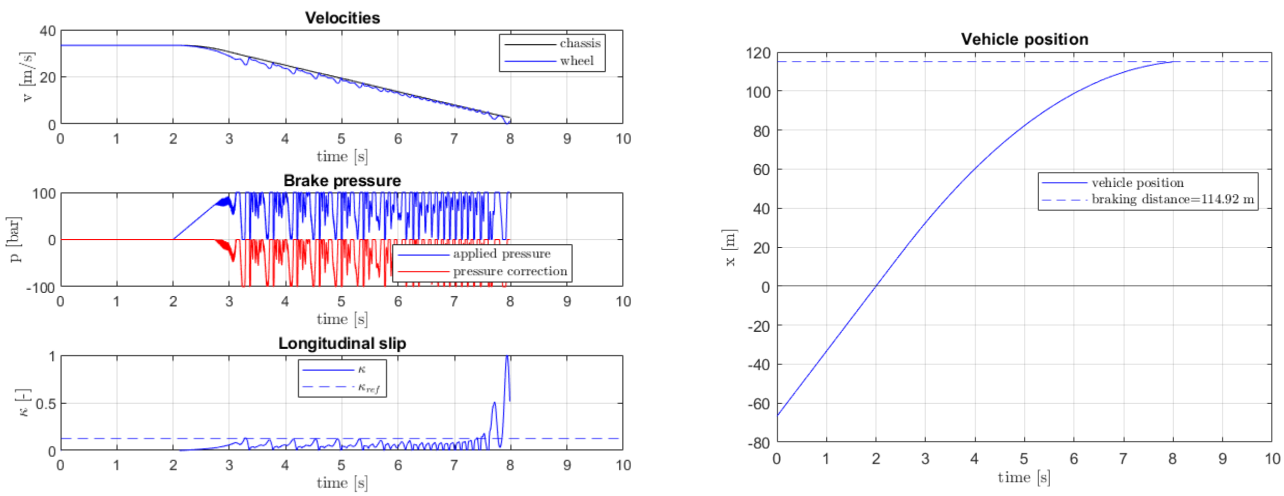


Figure 9: Simulation MSD controller with noise. Notice that the plot of the slip is accounting for the real κ, before introducing the noise.

To face the problem of noise in the sensors, a filter could be implemented on the signals. Just for the sake of completeness, this approach was tested in Simulink, using a low pass filter on the noisy slip estimation. Results are reported in Figure 10 and Figure 11, filtering in this case at 1000 Hz. For both controllers, the performances are in this case something in between the results obtained without noise and the ones obtained with noise and without filter.

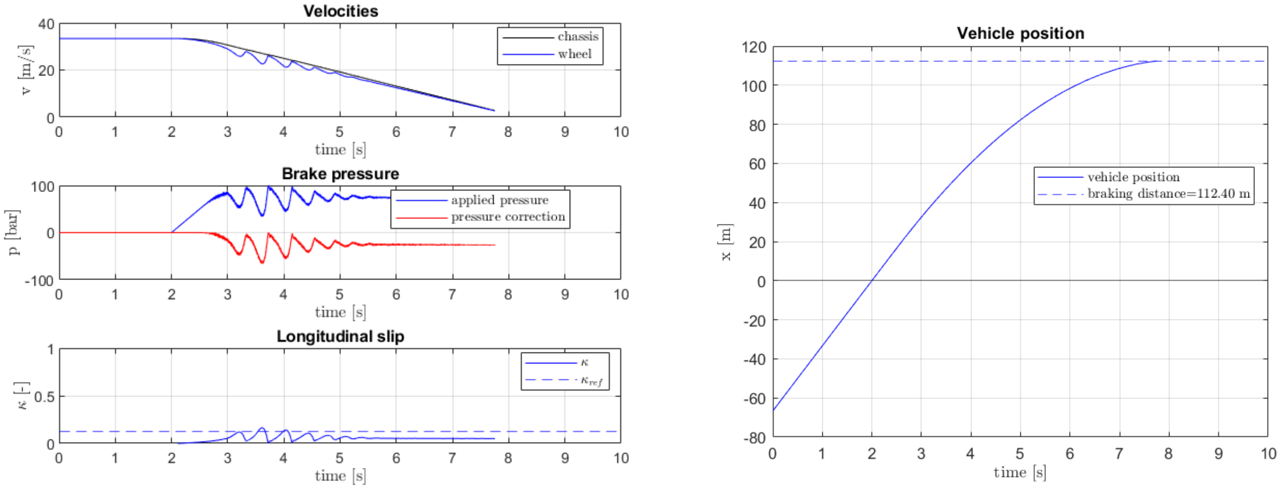


Figure 10: Simulation WS PI controller with noise and low pass filter (1000 Hz).

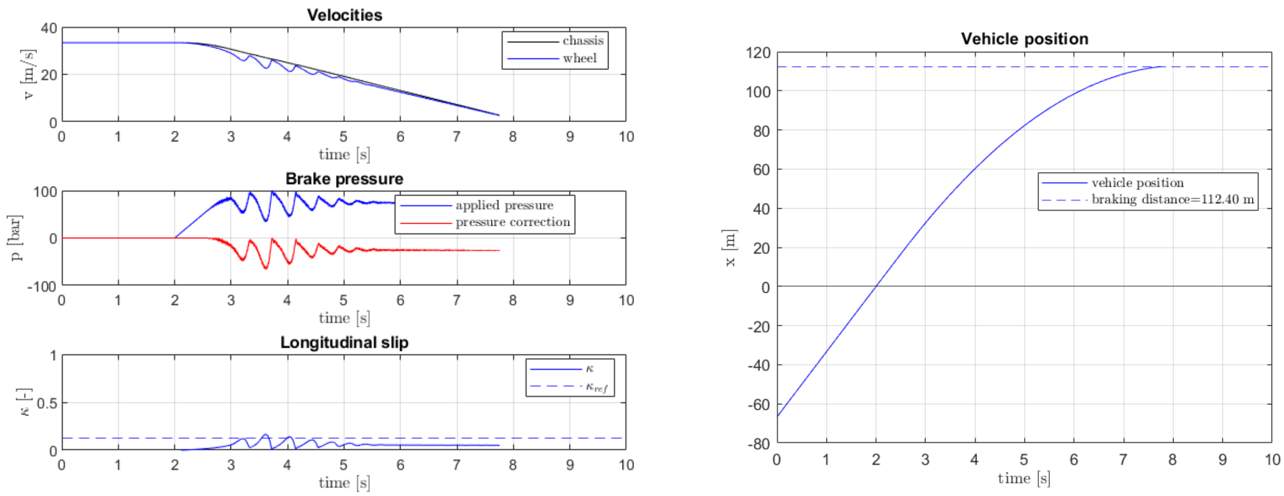


Figure 11: Simulation MSD controller with noise and low pass filter (1000 Hz).

Control strategy	**Braking dist. d**	**Control effort**	**Track. of $\kappa_{\mathbf{slip}}$**	**ITAE$_{\mathbf{jerk}}$**[3]
PI WSC (Fig 8)	114.83 m	high (noisy)	∼as before	10.09
MSDC (Fig 9)	114.92 m	high (noisy)	∼as before	35.75
PI WSC (filter) (Fig 10)	112.40 m	lower	∼as before	4.63
MSDC (filter) (Fig 11)	113.66 m	lower	∼as before	33.11

Table 3: Performance of different controllers with noise.

4 Self reflection and Conclusions

In this report two different approaches for wheel slip control in a simple emergency brake scenario have been considered. These approaches, namely WSC and MSDC, have been implemented using simple PI,

[3] $ITAE_{jerk}$ is normalized with respect to the value obtained when no ABS is applied.

PID or PD controllers, proving to be quite effective and consistent with the expectations in all the simulations.
A first important considerations is that, in a real scenario, the whole system dynamics (car, wheels, etc) is much more complex, strongly nonlinear and also depends on a lot of additional conditions (temperature, rain, road surface, load distribution, etc). In this assignment, many simplifying assumptions were made, for example neglecting the lateral motion, not considering all 4 the wheels, having a constant friction coefficient and so on.
In general we saw that the idea is to try to exploit the maximum longitudinal force possible by "imposing" a certain optimal κ (peak in Figure 2). However, this peak depends on many other parameters and changes continuously during the braking action. That's the reason why it is desired to actually have oscillations around this reference point κ_{ref} (in particular with simple controllers or when the uncertainties are high), which is what we observe in the simulations plots.
Finally, the main challenges faced during the project concern for sure the tuning of the regulator's parameters. In any case, such a simple error-based controller needs to accept some trade-offs between different performance indicators, and many parameters can be regulated to vary that. For sure, the choice of a more refined, model-based, state feedback controller, such as an MPC, for example, could have lead to better results overall. We also need to consider, though, that such a controller requires an higher computational power and also required a good knowledge of the system, which is, as already said, very complex. Additionally, an estimator (ex. extended Kalman filter), should be used to reconstruct the full state, but this also comes with an improvement of the filtering performances on noise and disturbances.

A Appendix - Wheel slip controller

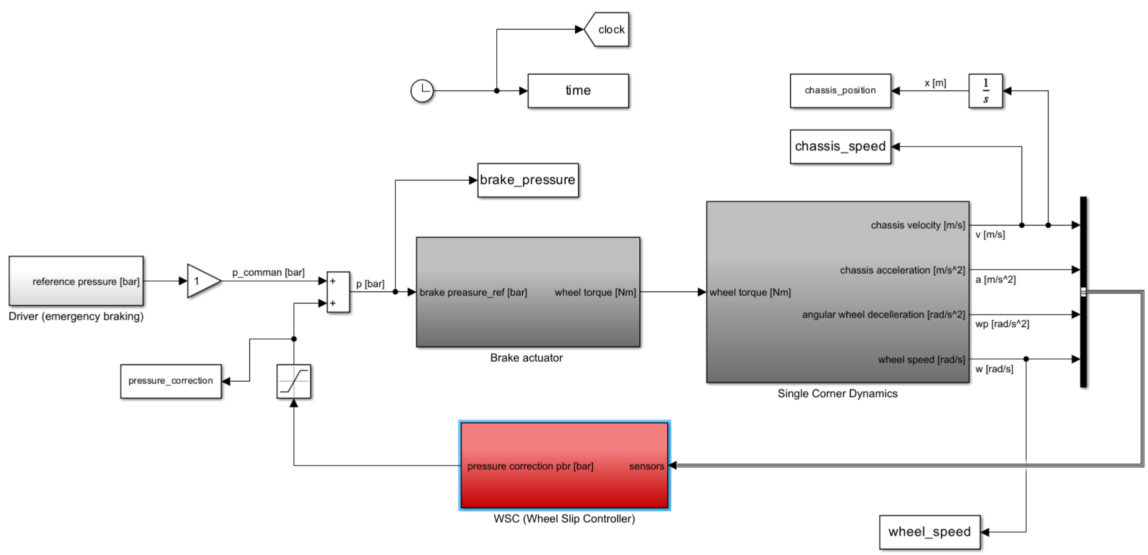
Design and architecture A basic PID controller for the slip error e can be defined as in Figure 13a, where the gains can be scheduled basing on the vehicle speed:

$$R(s) = K_P(v) + K_I(v)\,\frac{1}{s} + K_D(v)\,s\;, \quad \textit{(regulator in frequency domain)} \tag{4}$$

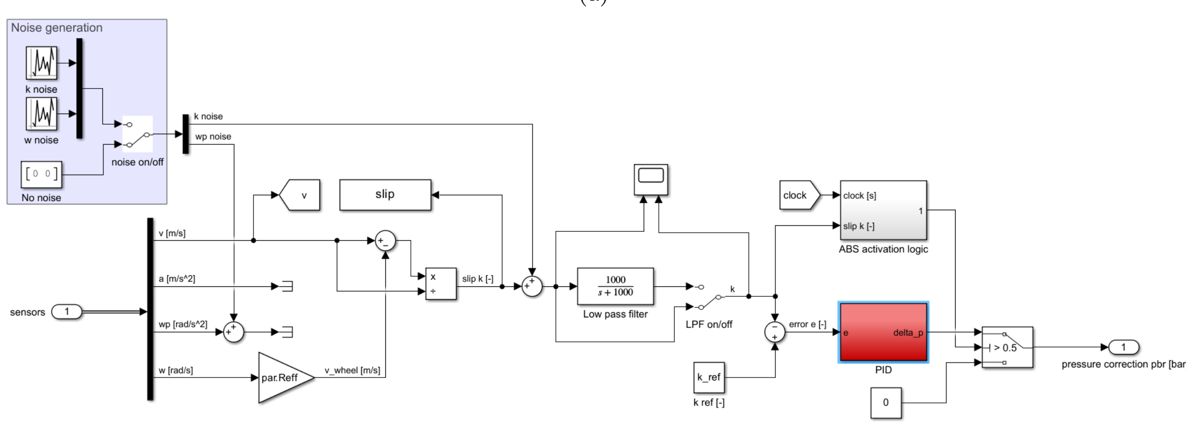
The derivative contribution is very sensitive to noise and tends to amplify it, thus a filter can be used to mitigate this behaviour, as in Figure 13b. Obviously this introduces a limitation of the derivative action itself, so N should be chosen so that it is not too low (killing K_D completely), nor too high (ineffective), and this is possible when the noise bandwidth is far enough from the system's one. The resulting controller is:

$$R(s) = K_P(v) + K_I(v)\,\frac{1}{s} + K_D(v)\,\frac{N}{1 + N/s} \tag{5}$$

Finally, the integral action is needed to avoid steady-state errors, but we want to prevent K_I from "accumulating too much error" when the controller reaches saturation, causing excessive oscillations and overshoot. Thus, an anti-windup filter is added to the system, in particular exploiting a back-calculation strategy, as shown in Figure 13c. The complete architecture of the Simulink model is shown in Figure 12.



(a)



(b)

Figure 12: (a) Simulink model. (b) Detail of the `WSC (Wheel Slip Controller)` block.

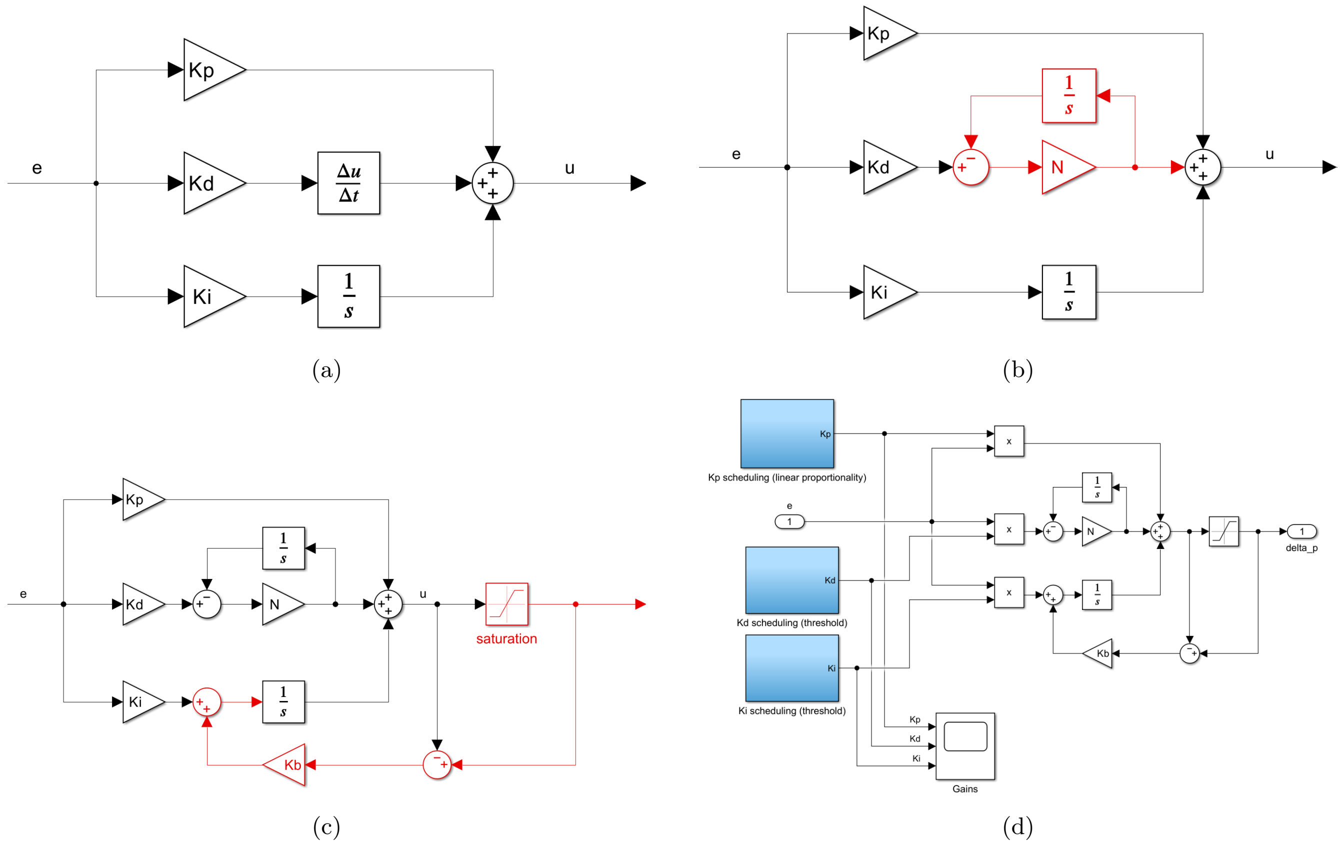


Figure 13: (a) Simple PID controller. (b) Filtered derivative. (c) Back-calculation anti-windup filter. (d) Complete Simulink model of the `PID` block, with gain scheduling.

The `ABS activation logic` block in Figure 12b allows for switching on/off the WSC manually or basing on when the emergency brake starts. Gain scheduling, as already presented in the report, is implemented as:

$$K_P = \frac{(K_{max} - K_{min})(v - v_{min})}{v_0 - v_{min}} \quad , \quad with\ K_{max} = 500,\ K_{min} = 100,\ v_0 = \frac{120}{3.6}\ m/s,\ v_{min} = \frac{10}{3.6}\ m/s \tag{6}$$

$$K_I = \begin{cases} 30, & if\ v > 50\ Km/h \\ 70, & if\ v \leq 50\ Km/h \end{cases} \tag{7}$$

$$K_D = 5 \tag{8}$$

Simulations In the following Table 4, some simulations results, in addition to the ones presented in Table 1, are reported to show the influence of various tuning parameters and their effects.
From these results, it's clear that K_D must be carefully chosen in order to avoid instability or an excessively aggressive controller that also penalizes comfort (high $ITAE_{jerk}$). In general, the derivative action does not seem to bring remarkable improvements with respect to the PI controller, so we can also think of omitting it. Moreover, the anti-windup filter is essential for the functioning of the control, as the controller tends to saturate too much without it.
Many other combinations of parameters are possible, such that, for example, more complex polynomial dependencies of the gains with respect to the vehicle speed. The reader is free to test them in the provided code to better see the effects of all the parameters.

Control strategy	Braking dist. d	Control effort	Track. of $\kappa_{\mathbf{slip}}$	$\mathbf{ITAE_{jerk}}$[4]
no ABS (Fig 3)	143.54 m	almost null	no	1
PI + anti-windup (Fig 4)	112.29 m	very limited	quite close	4.52
PID (Fig 6)	111.03 m	limited (for low K_D)	faster, smaller oscill.	3.00
PI no anti-windup (Fig 5)	135.17 m	high	very poor, unstable	123.63
PID with high K_D (Fig 14)	117.20 m	critical	aggressive, unstable	80.77

Table 4: Performance of different wheel slip controllers.

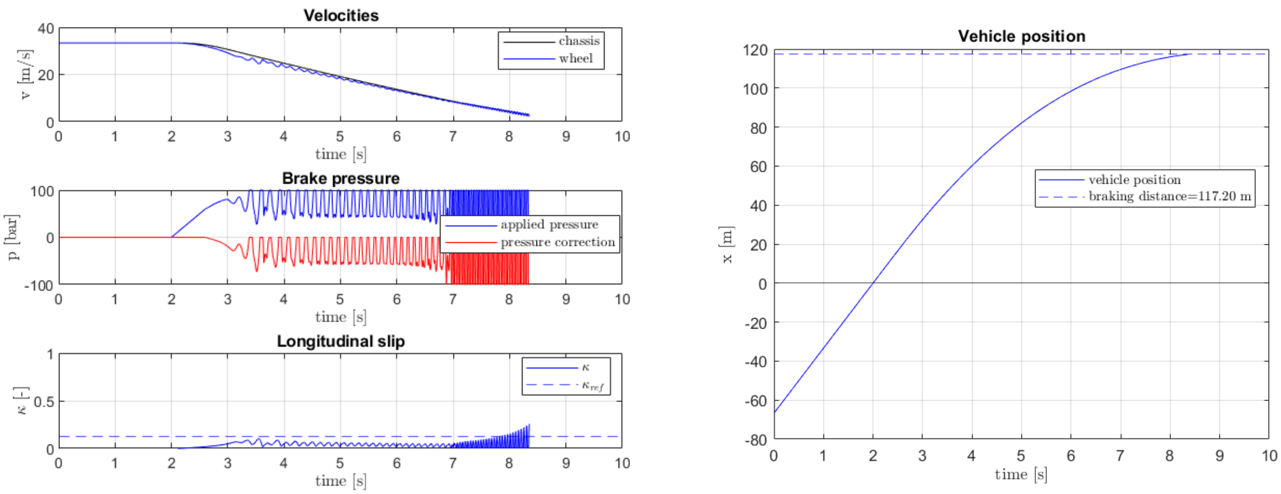
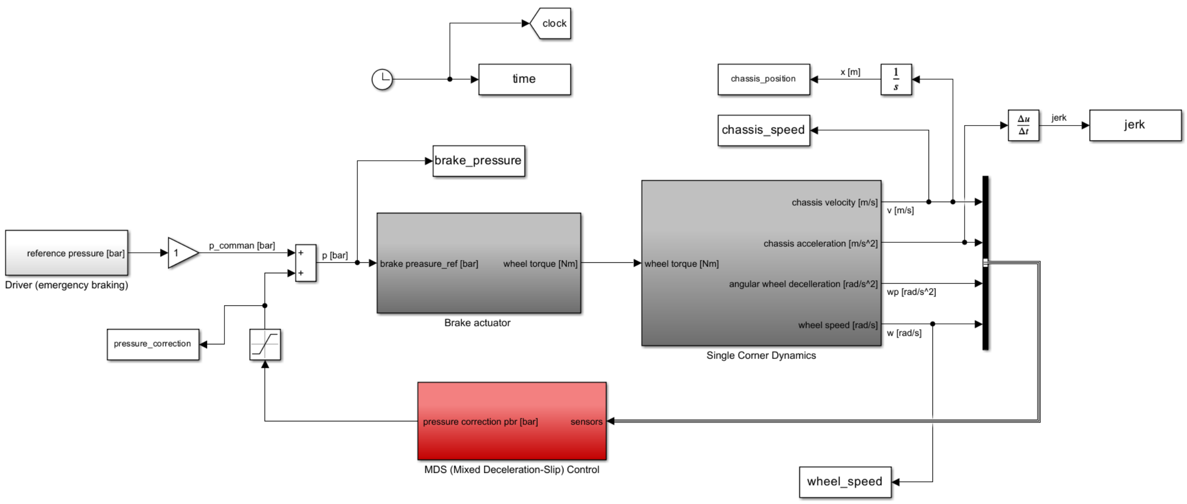


Figure 14: Simulation PID control with high derivative action ($K_D = 50$).

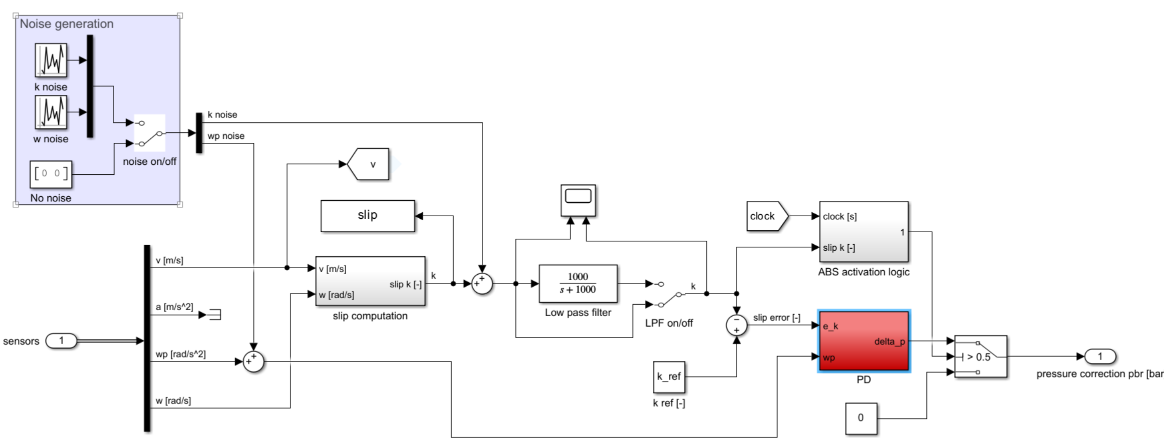
[4]$ITAE_{jerk}$ is normalized with respect to the value obtained when no ABS is applied.

B Appendix - Mixed slip-deceleration controller

Architecture The Simulink model of the system controlled with a MSDC is shown in Figure 15.



(a)



(b)

Figure 15: (a) Simulink model. (b) Detail of the `MDS (Mixed Slip-Deceleration) control` *block.*

The `ABS activation logic` block in Figure 12b allows for switching on/off the WSC manually or basing on when the emergency brake starts. Gain scheduling, as already presented in the report, is implemented as:

$$K_P = \frac{(K_{max} - K_{min})(v - v_{min})}{v_0 - v_{min}} \quad , \quad with\ K_{max} = 400,\ K_{min} = 10,\ v_0 = \frac{120}{3.6}\ m/s,\ v_{min} = \frac{10}{3.6}\ m/s \tag{9}$$

$$K_D = \frac{(K_{high} - K_{low})(v - v_{min})}{v_0 - v_{min}} \quad , \quad with\ K_{high} = 40,\ K_{low} = 30,\ v_0 = \frac{120}{3.6}\ m/s,\ v_{min} = \frac{10}{3.6}\ m/s \tag{10}$$