Assignment 2 - Vehicle Stability Control

Vehicle Dynamics and Control
(RO47017)

Author:

Nicola Visentin (6354815)

May 23, 2025

Contents

Introduction

This assignment consists in the design and simulation of a Vehicle Stability Control (VSC) sytem for handling the lateral behavior of a car. When needed, a regulator should be able to apply a proper additional yaw moment dM_z to the vehicle in order to follow as much as possible the desired lateral behavior imposed by the driver through the steering wheel. Data and car parameters are reported in Appendix A: notice that our vehicle is designed to have an understeering behavior, as the understeering gradient K_{us} is larger than zero.

In section 1 the control strategy is explained in detail and two different approaches are proposed, along with their design and tuning: a PD and a LQR. Other components of the VSC system are also introduced (reference generator).
In section 2 the system is simulated to test the performances of the designed VSC. A Sine with Dwell (SwD) maneuver is performed at a high friction condition, starting from two different initial velocities V_0 (60 Km/h and 100 Km/h). The effectiveness of the controllers is evaluated by considering tracking capabilities and control effort.
Section 3 shows the same simulations of the previous section, but this time in presence of noise on the measurement of the yaw rate.
Finally, conclusions and reflections are drawn in section 4.

1 Vehicle Stability Control design

Ideally, we would like our car to perfectly follow the trajectory imposed by the steering wheel input provided by the driver. This is true at very low speeds in normal friction conditions, but in general, we also need to consider wheel lateral slip. In particular, three scenarios can arise:

- Neutral steering: when the rear and the front slip angles are equal, i.e. $\alpha_f = \alpha_r$. Here, we don't need to change the steering angle to follow the desired turn radius.

- Understeering: $\alpha_f > \alpha_r$. The actual yaw rate $\dot{\psi}$ with which the car is turning is lower than the desired one. We would like generate an additional contribution dM_z to increase the yaw moment M_z.

- Oversteering: $\alpha_f < \alpha_r$. The actual yaw rate $\dot{\psi}$ with which the car is turning is higher than the desired one. We would like to generate an opposite yaw moment dM_z to reach $\dot{\psi}_{ref}$.

Thus, to achieve a better tracking of the desired trajectory we need to generate a proper additional yaw moment contribution dM_z. This can be obtained by regulating the braking on the different wheels (details are not reported here).

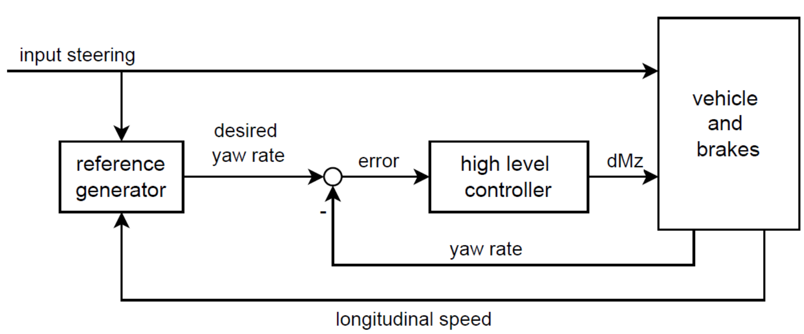


Figure 1: Simplified architecture of the VSC system.

The full simplified scheme of a VSC is shown in Figure 1. Since the controller acts on the error e between the desired yaw rate $\dot{\psi}_{ref}$ and the actual yaw rate $\dot{\psi}$, first of all we need to compute the reference. This is done by a reference generator that takes the input steering wheel command from the user and outputs $\dot{\psi}_{ref}$ (structure and design of this component are presented in subsection 1.1). The actual value of the yaw rate is obtained out of measurements, usually through an estimator. Finally, the role of the controller is to generate the proper "correcting" yaw moment dM_z, as already said. Two different controllers are developed in subsection 1.2 for that.

1.1 Reference generator

The goal of this component is to "convert" the steering wheel command from the driver (that causes a certain steering angle δ of the wheels) into the corresponding yaw rate $\dot{\psi}_{ref}$. To do so, a simple linear bicycle model [1] is used to compute the steady-state response of the vehicle $\dot{\psi}_{ss}$ to a certain δ:

$$\dot{\psi}_{ss} = G_{ss}^{yaw}\delta \ \ , \quad with \ G_{ss}^{yaw} = \frac{u}{L + K_{us}\ u^2/g}$$

Being u the longitudinal speed and K_{us} the understeering gradient. This approach is of course easy and fast, but also very simplified: we are considering linear behavior (not a good approximation at higher speeds), a steady-state cornering response (but in general δ can change rapidly in time) and we are using the bicycle model, which relies already on many other assumptions on its own.
For this reason, the produced output needs to be "corrected" to be more similar to the actual one. A saturation is first imposed on $\dot{\psi}_{ss}$, since we can intuitively imagine that the amount of steering we can obtain is limited by the "available" friction, which limits the lateral acceleration a_y. This limitation needs to be computed basing on the friction coefficient μ, so an estimator is required. Approximately, the maximum yaw rate we can impose is [2]:

$$|\dot{\psi}| < 0.85\ \frac{\mu g}{u} \tag{1}$$

Still, the saturated output $\dot{\psi}_{sat}$ is coming from a linear model. Thus, to make the response closer to the real one, we filter it using a second order transfer function [2]:

$$\dot{\Psi}_{ref} = \frac{\omega_n^2(1+\tau s)}{s^2 + 2\zeta\omega_n s + \omega_n^2}\ \dot{\Psi}_{sat}$$

The characteristic parameters τ, ζ and ω_n are fitted experimentally basing on Sine Swept Test. Notice that these parameters, that are reported in Appendix A, should be depending on the velocity, but in the assignment they are assumed to be constant for simplicity.

The entire structure of the reference generator block, as presented above, is shown in Figure 2. However, following the request of the assignment, the a_y-limiter saturation and the friction estimator blocks are omitted. Simulink model of the whole implementation is reported in Appendix B.

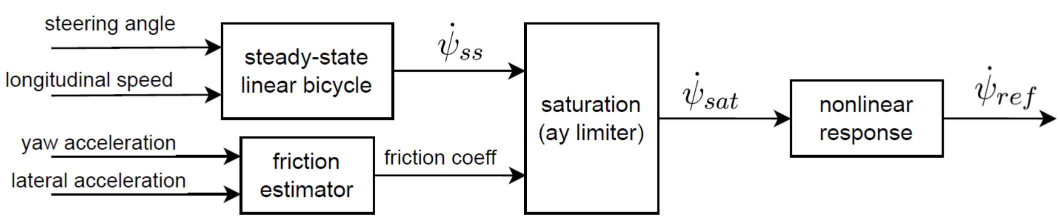


Figure 2: Reference generator block

1.2 Controller

Once we have the reference in terms of yaw rate $\dot{\psi}_{ref}$, we should try to track it by providing a certain additional yaw moment contribution dM_z to the vehicle, as already explained. Notice that the output of the controller saturates at 10000 Nm.

PD The simplest possible way to do so consists in using a PD feedback regulator. The control law is:

$$dM_z = K_p \cdot e + K_d \cdot \dot{e} \ , \quad \textit{with } e = \dot{\psi}_{ref} - \dot{\psi} \textit{ and } \dot{e} = \ddot{\psi}_{ref} - \ddot{\psi}$$

We assume we are directly able to measure the yaw rate. Since the lateral behavior of the car also depends on the speed, the gains are adjusted basing on the initial speed of the maneuver. After some trial-and-error, the values for K_p and K_d were chosen trying to accomplish a fair trade-off between tracking performance and control effort.

Velocity	$\mathbf{K_p}$	$\mathbf{K_d}$
$V_0 = 60$ Km/h	30000	20000
$V_0 = 100$ Km/h	20	100

Table 1: Gains for the PD controller

LQR A Linear Quadratic Regulator (LQR) is a more refined model based full-state controller. In particular, we will consider the infinite time LQR formulation, which is proven to be asymptotically stable at the origin under the assumptions of having a controllable LTI system, positive definite Q matrix and positive semidefinite R matrix.
The formulation of the problem is the following. We have a linear system defined by the bicycle model:

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{B}\ dM_z$$

where $\mathbf{x} = [v,\ \dot{\psi}]^T$ is the state (lateral velocity and yaw rate) and dM_z the input. Matrices A and $\mathbf{B}$ can be defined as (notice that our system is not time-invariant, as the system's matrices depend on the longitudinal velocity, that changes in time):

$$A = \begin{bmatrix} -\dfrac{C_{\alpha f} + C_{\alpha r}}{mu} & \dfrac{l_r C_{\alpha r} - l_f C_{\alpha f}}{mu} - u \\ \dfrac{l_r C_{\alpha r} - l_f C_{\alpha f}}{I_z u} & -\dfrac{l_r^2 C_{\alpha r} + l_f^2 C_{\alpha f}}{I_z u} \end{bmatrix} \quad \textit{and} \quad \mathbf{B} = \begin{bmatrix} \overset{0}{\dfrac{\cancel{1}}{\cancel{m}\ l_f}} \\ \dfrac{1}{I_z} \end{bmatrix}$$

where parameters $C_{\alpha f},\ C_{\alpha r},\ l_f,\ l_r,\ I_z,\ m$ are described in Appendix A and u is the longitudinal speed. Since we are interested in tracking $\dot{\psi}$, we could define:

$$y = \mathbf{C}\mathbf{x} \ , \quad \textit{with } C\text{=}[0,\ 1]$$

And develop an output tracking problem as described in [3], by finding the reference state $\mathbf{x_{ref}} = [v_{ref},\ \dot{\psi}_{ref}]^T$ such that:

$$\begin{cases} \mathbf{x_{ref}} = A\mathbf{x_{ref}} + \mathbf{B}\ dM_{z,ref} \\ y_{ref} = \mathbf{C}\mathbf{x_{ref}} \end{cases}$$

and then shifting coordinates and building a LQR on $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x_{ref}}$ (remember that an LQR stabilizes the system at the origin). However this would require to compute v_{ref} and $dM_{z,ref}$, so, for simplicity, the LQR control law was defined in a simplified way as:

$$dM_z = \mathbf{K} \cdot \begin{bmatrix} v \\ \dot{\psi} - \dot{\psi}_{ref} \end{bmatrix}$$

where $\mathbf{K} = [K_1,\ K_2]$ is the LQR gain matrix computed as $\mathbf{K} = -R\mathbf{B}^T P$ being P the solution of the Algebraic Riccati Equation (ARE). The weight matrices[1] Q and R were chosen empirically as:

$$Q = \begin{bmatrix} \dfrac{100 \cdot u}{\dot{\psi}^2_{max}} & 0 \\ 0 & 0 \end{bmatrix} \quad and \quad R = \frac{u}{dM^2_{max}}$$

Notice that all quantities are normalized in order to take into account for the different order of magnitude of the different physical variables[2]. In particular, $\dot{\psi}_{max}$ was chosen as 1 rad/s (which is approximately the order of magnitude observed in the PD simulations) and dM_{max} as 10000 Nm (maximum allowable yaw moment according to the assignment requests). Finally, gain scheduling was performed as the weights depend on the longitudinal speed u.
The final values of the weights was chosen trying to ensure a good balance between tracking performance (increasing penalization on $\dot{\psi}$) and control effort. v has a null weight since we are only interested in tracking the yaw rate.

2 Simulation of Sine with Dwell maneuver

In this section simulation results are shown. The simulation consists in a car traveling at a constant initial speed V_0 and performing a Sine with Dwell maneuver starting at time $t_0 = 10$ s. The maneuver consists in giving a certain predefined steering input (Figure 3) and see "how well" the vehicle responds to that. To evaluate the performance of the VSC two things are accounted for:

- Control effort (saturation of the control input, "nervousness" of the controller, etc).
- Tracking performance.

In particular, in the second case the yaw velocity metric is computed as (t_0 and t_f are the start and the end of the maneuver):

$$yaw\ velocity\ metric = \frac{\int_{t_0}^{t_f} |\dot{\psi} - \dot{\psi}_{ref}| dt}{\int_{t_0}^{t_f} |\dot{\psi}_{ref}| dt}$$

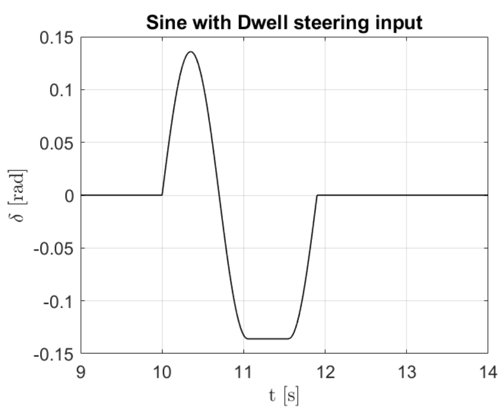


Figure 3: SwD steering angle input

Moreover, there are certain requirements that need to be satisfied in order to consider the maneuver "effective": the lateral displacement y of the vehicle at time $t = t_0 + 1.07$ must be greater than a certain threshold; the yaw rate $|\dot{\psi}|$ at time $t = t_f + 1$ must be lower than 35% of $|\dot{\psi}_{peak}|$; the yaw rate $|\dot{\psi}|$ at time $t = t_f + 1.75$ must be lower than 20% of $|\dot{\psi}_{peak}|$. $\dot{\psi}_{peak}$ is the lower observed value of the yaw rate. These "checks" are represented in the plots with the green (pass) and red (fail) lines.

First, the maneuver was simulated without any VSC control active, for both an initial speed of 60 Km/h and 100 Km/h: results are shown in Figure 4 and Figure 5. As we can see, for lower speed the car is still able to recover the turn, but for higher velocities this is not possible anymore.

[1] The theory behind LQR (cost function, optimality of the gains, convergence, convexity of the problem, ARE, etc) is not reported here, since it is out of the scope of this assignment.

[2] Remember that Q penalizes the state $[v,\ \dot{\psi}]^T$ while R the control input dM_z. We have a power 2 at the denominator as each weight multiplies the corresponding variable two times in the cost function (quadratic cost).

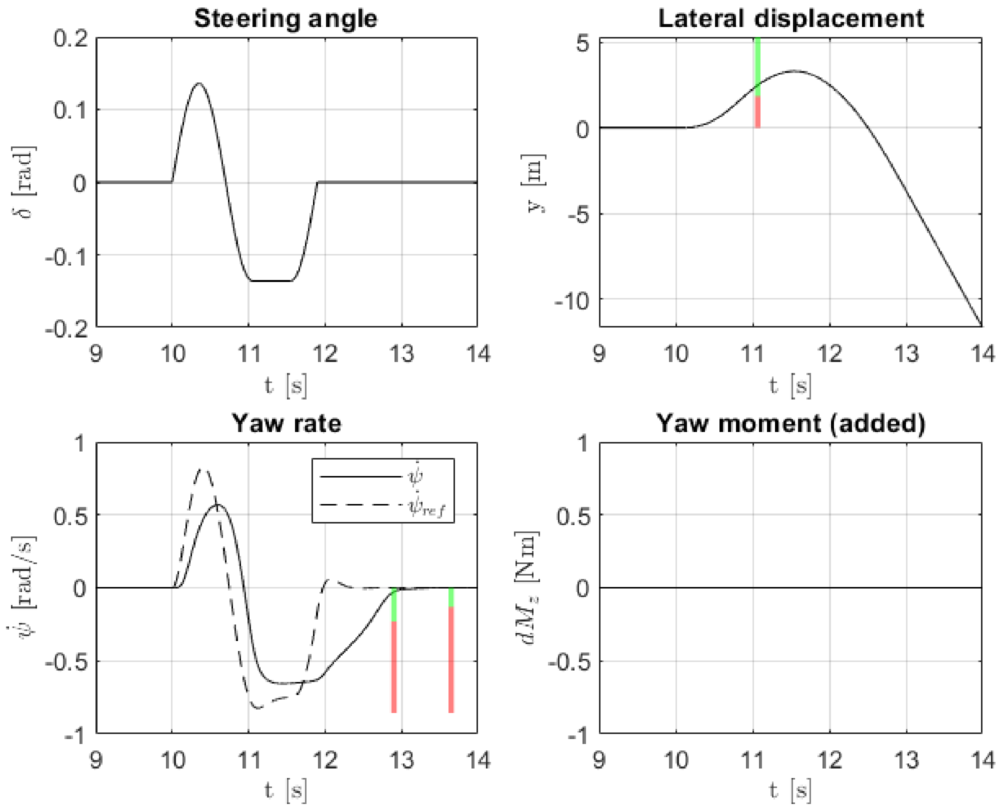


Figure 4: No VSC ($V_0 = 60\ Km/h$)

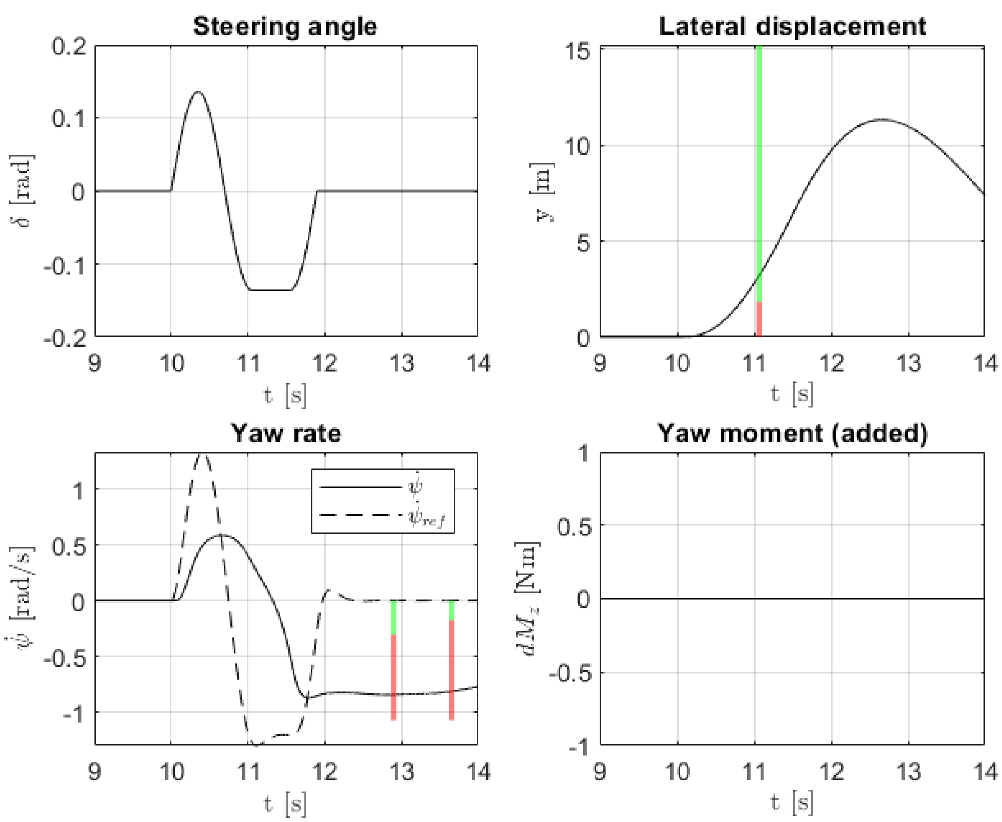


Figure 5: No VSC ($V_0 = 100\ Km/h$)

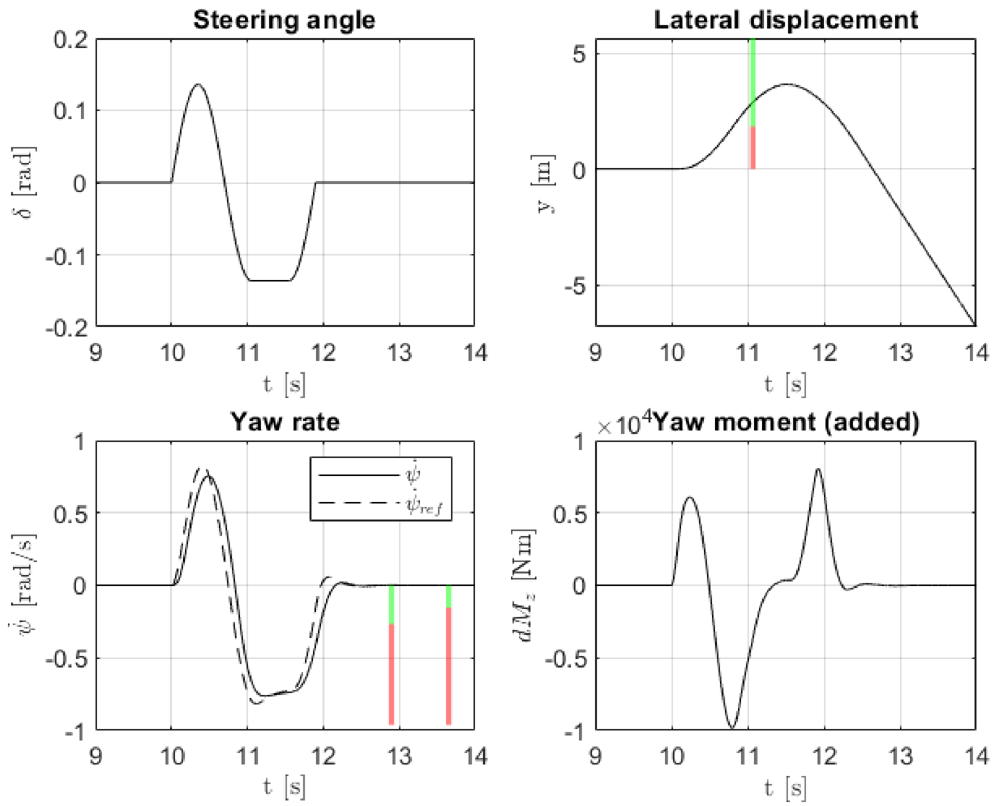


Figure 6: PD control ($V_0 = 60$ Km/h)

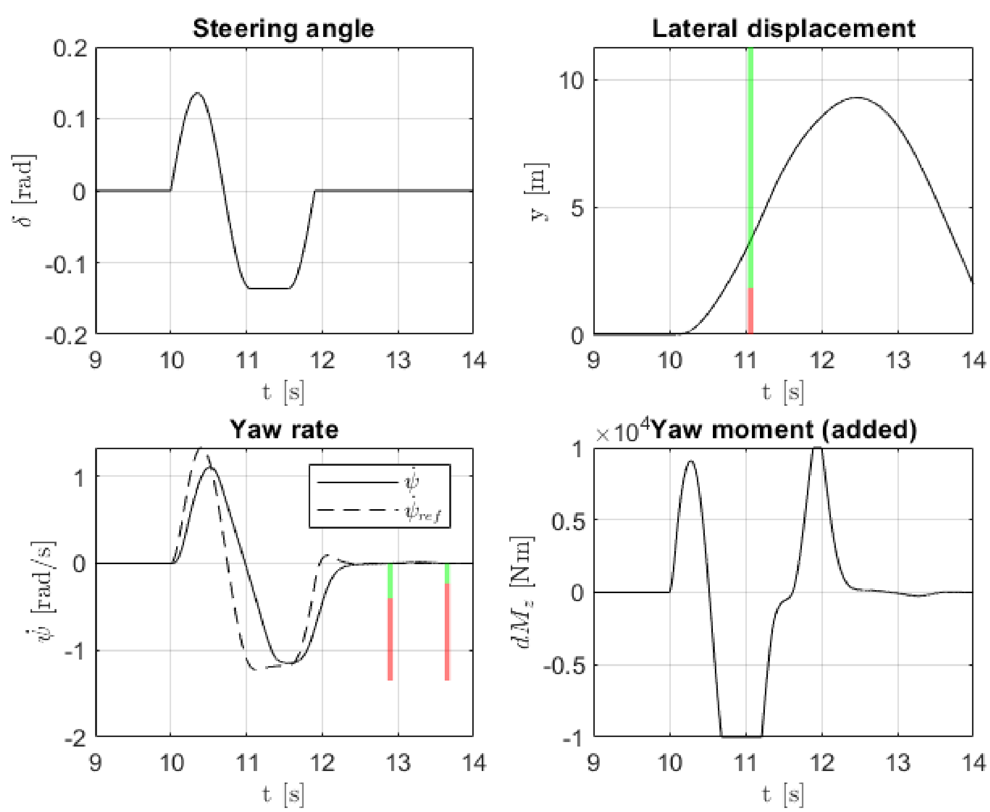


Figure 7: PD control ($V_0 = 100$ Km/h)

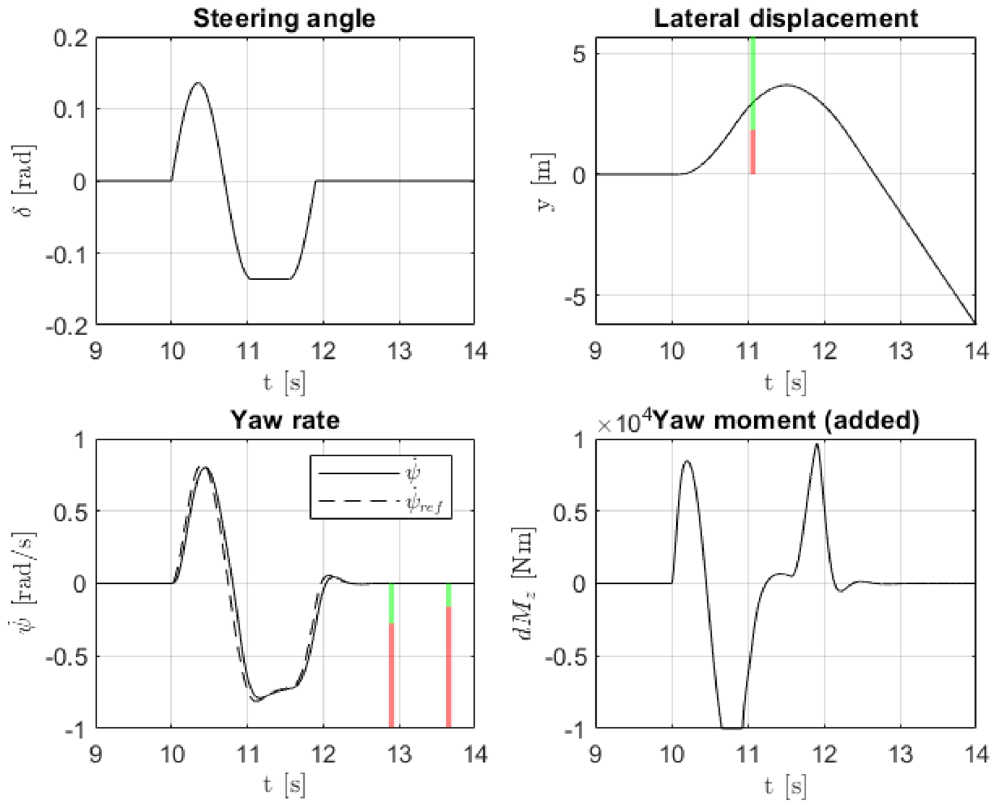


Figure 8: LQR control ($V_0 = 60$ Km/h)

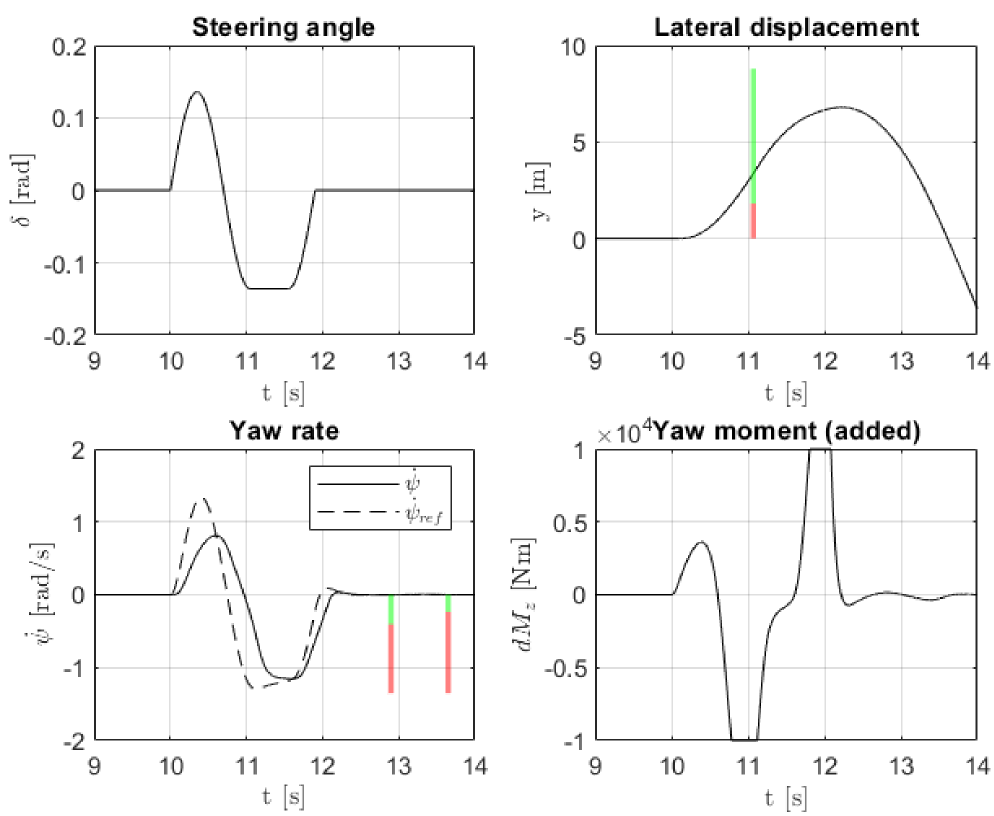


Figure 9: LQR control ($V_0 = 100$ Km/h)

Then VSC based on PD is turned on (Figure 6, Figure 7). Now the situation has improved a lot, and this is particularly evident from the better tracking of the reference yaw rate, in particular when the initial speed is 100 Km/h. The control effort is quite limited, with the controller not even reaching saturation for the 60 Km/h test.
Finally, LQR based VSC is simulated. From Figure 8 and Figure 9 we can see that the tracking performance has further improved with respect to PD, while the control effort remains almost the same.
Yaw velocity metric of all the tests is reported in Table 2 for comparison.

3 Sensor noise

Now the same simulations of section 2 are performed, but a noise of 1 rad/s is introduced on the yaw rate measurement. The corresponding plots are shown in the following pages (Figures from 10 to 13).
In general, the high frequency noise in the measurement causes of course the generation of a very noisy control input, that could result to be critical for the controller. However, the dynamics of the physical system acts as sort of a filter, and the real yaw rate (notice that the plots show the yaw rate *before* applying the noise) is not varying so quickly (impossible from a physical point of view).
Concerning PD, the situation appears more critical. This is somehow expected, as the derivative contribution always tends to amplify noise.
VSC based on LQR, instead, seems to be less sensitive to noise compared to PD, in particular when the initial velocity is higher.

Test	Yaw velocity metric	Control effort
no VSC (60 Km/h)	0.76	-
no VSC (100 Km/h)	2.50	-
PD (60 Km/h)	0.20	very limited
PD (100 Km/h)	0.45	limited
LQR (60 Km/h)	0.11	very limited (comparable with PD)
LQR (100 Km/h)	0.36	limited (comparable with PD)
PD with noise (60 Km/h)	0.21	very nervous
PD with noise (100 Km/h)	0.45	extremely nervous
LQR with noise (60 Km/h)	0.11	very nervous (comparable with PD)
LQR with noise (100 Km/h)	0.36	nervous (much less than PD)

Table 2: Performances of various controllers. In general, for yaw velocity metric, values lower than 0.20-0.25 are very good, values between 0.25 and 0.75 are acceptable and values above 0.75 are critical [2].

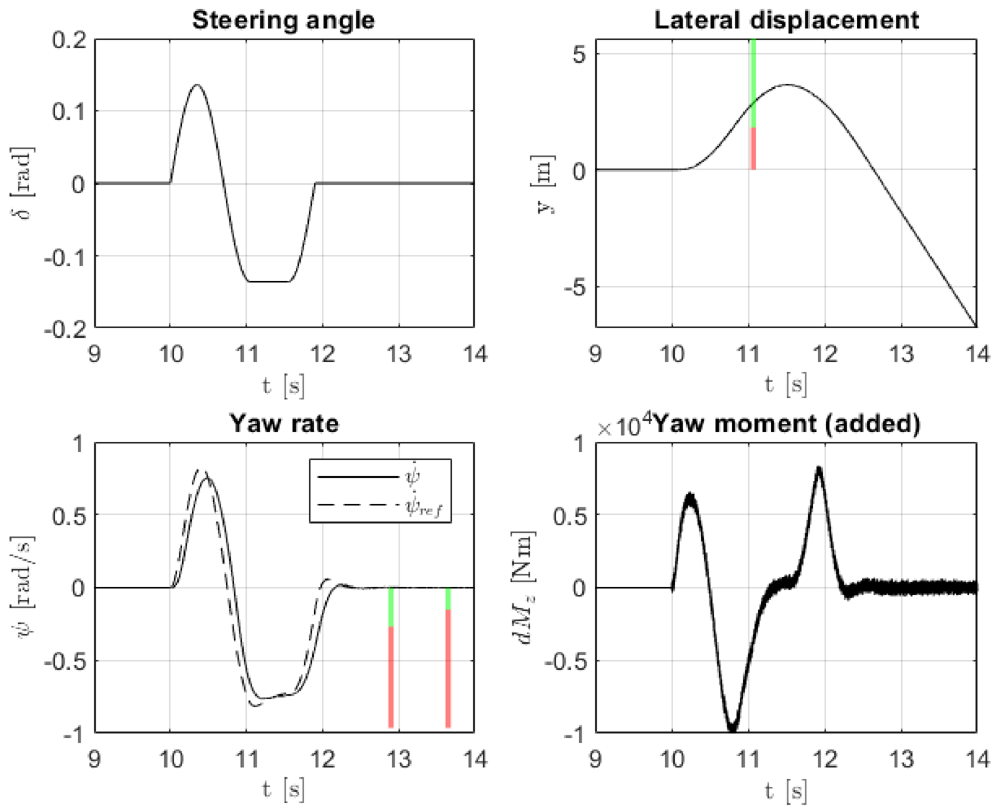


Figure 10: PD control with noisy measurement ($V_0 = 60$ Km/h)

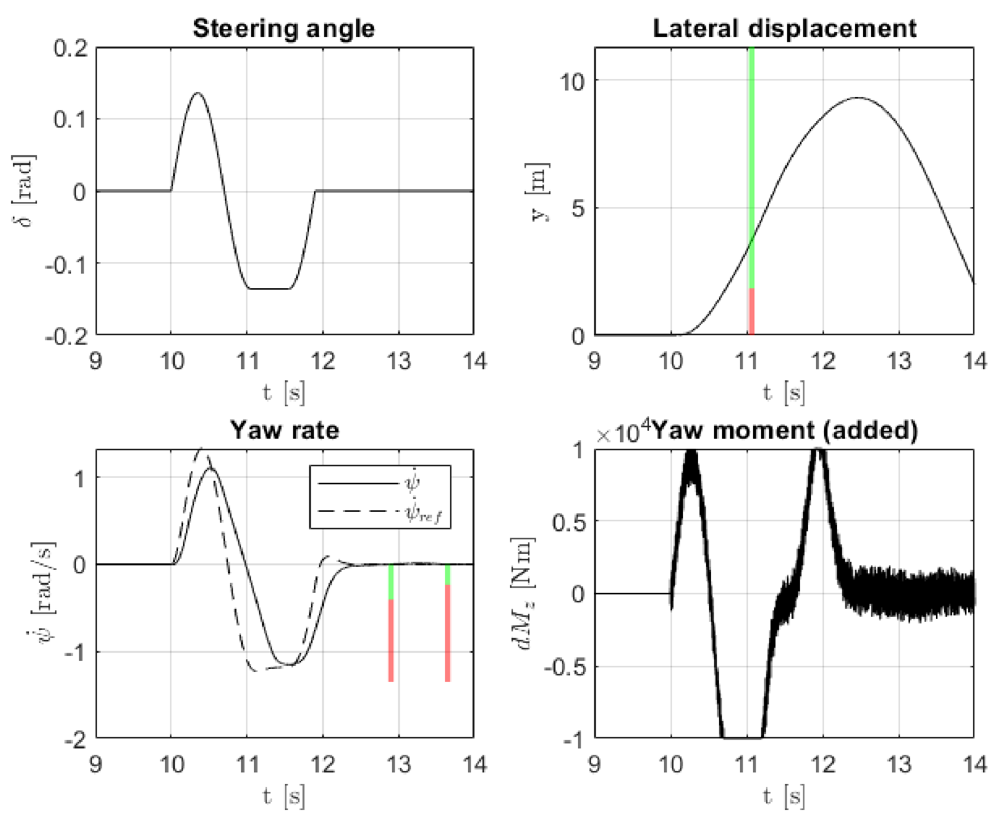


Figure 11: PD control with noisy measurement ($V_0 = 100$ Km/h)

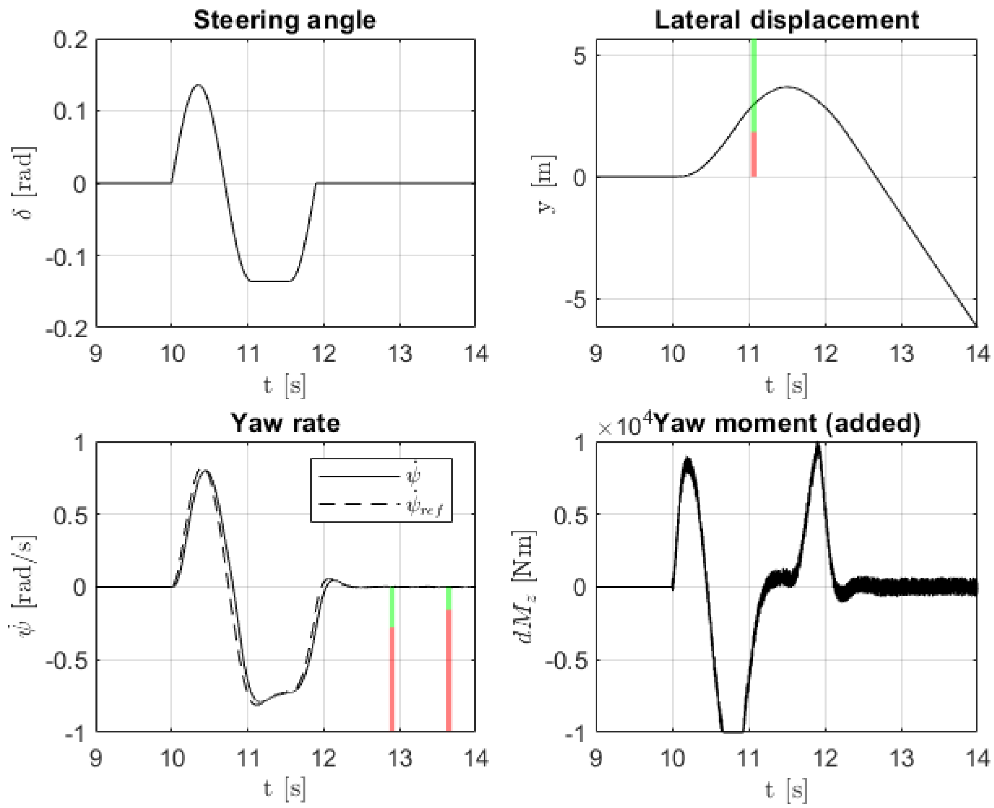


Figure 12: LQR control with noisy measurement ($V_0 = 60$ Km/h)

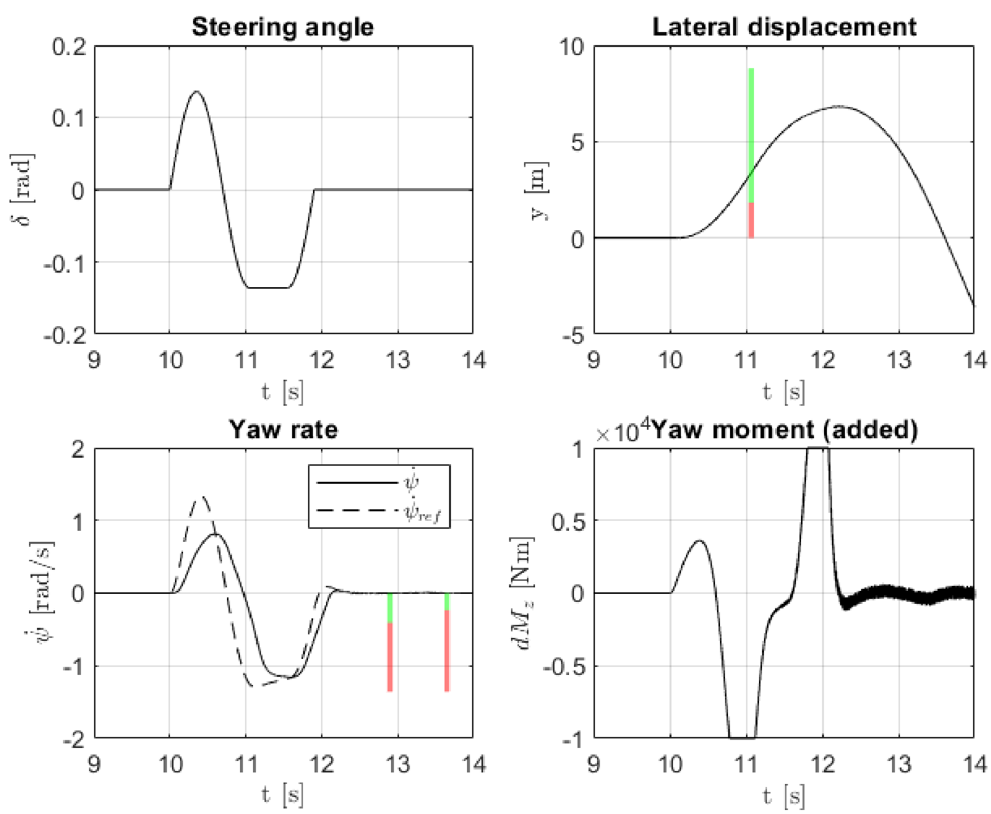


Figure 13: LQR control with noisy measurement ($V_0 = 100$ Km/h)

4 Self reflection

Implementing a simple VSC system in Matlab/Simulink was very useful to have a better insights on this topic and really helped me to understand how each component works in practice. The main challenge was probably related to tuning the controllers gains and weights in order to find a good trade off between performance and control effort, which took a bit of time.
Nevertheless the final result is quite satisfactory, and I think that would be interesting to further explore some aspects, like using a more advanced controller (MPC, SMC, etc), trying different tests with other velocities/steering inputs, or implementing a more complete VSC system (for example also including the a_y-limiter in the reference generator).

Conclusions

In this assignment a Vehicle Stability Control was implemented using Matlab/Simulink, focusing specifically on designing the reference generator and the controller. A simplified architecture was considered for the reference generator and two controllers were tested: PD and LQR. The system was then simulated on a Sine with Dwell maneuver with two different initial velocities, showing good results for all the cases.

In general, PD control is a little bit less performing, in particular when noise comes into play (due to the derivative contribution), but at the same time it is simpler and does not require a model of the system. LQR, on the other side, relies on optimality, and weights can be tuned to penalize the control effort/tracking performance in different ways.
Moreover, it has been observed that noise makes the control input too demanding in certain situations. This could be maybe mitigated by introducing a filter, even if in that case time delays will be introduced in the loop.

We also need to remember that some simplifications were made in this assignment, like assuming constant parameters for the 2^{nd} order transfer function in the reference generator or omitting some components, like estimators (e.g. for the friction coefficient) or the a_y-limiter.
Concerning this last point, some tests were actually made also using a simple implementation of the a_y-limiter as described in subsection 1.1. If the reader is interested, results are shown in Appendix C.

References

[1] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer Science & Business Media, 2nd edition, 2012.

[2] Barys Shyrokau. Ro47017 vehicle dynamics & control - vehicle stability control. *Lecture slides*, TU Delft, Brightspace, 2025. https://brightspace.tudelft.nl/d2l/le/content/682441/viewContent/3925786/View (Accessed: 21 May 2025).

[3] J. Rawlings and D. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.

A Appendix - Car parameters

Those are the provided car parameters and data:

Parameter	Value	Unit	Description
g	9.81	m/s^2	Gravitational acceleration
V_0	60 or 100	Km/h	Constant speed before the maneuver
m	1380	kg	Vehicle mass
I_z	2634.5	$kg{\cdot}m^2$	Moment of inertia about z-axis
L	2.79	m	Wheelbase
l_f	1.384	m	Distance from front axle to CoG
l_r	1.406	m	Distance from rear axle to CoG
i_{steer}	15.4	-	Steering ratio
m_f	695.4	kg	Front sprung mass
m_r	684.6	kg	Rear sprung mass
μ	1	-	Friction coefficient
$C_{\alpha f}$	120000	N/rad	Front axle cornering stiffness
$C_{\alpha r}$	190000	N/rad	Rear axle cornering stiffness
K_{us}	0.0022	N^{-1}	Understeer gradient
ω_n	11	rad/s	Yaw rate natural frequency
ζ	0.7	-	Yaw rate damping
τ	0.09	s	Yaw rate time constant

Table 3: Data and vehicle parameters

B Appendix - Simulink models

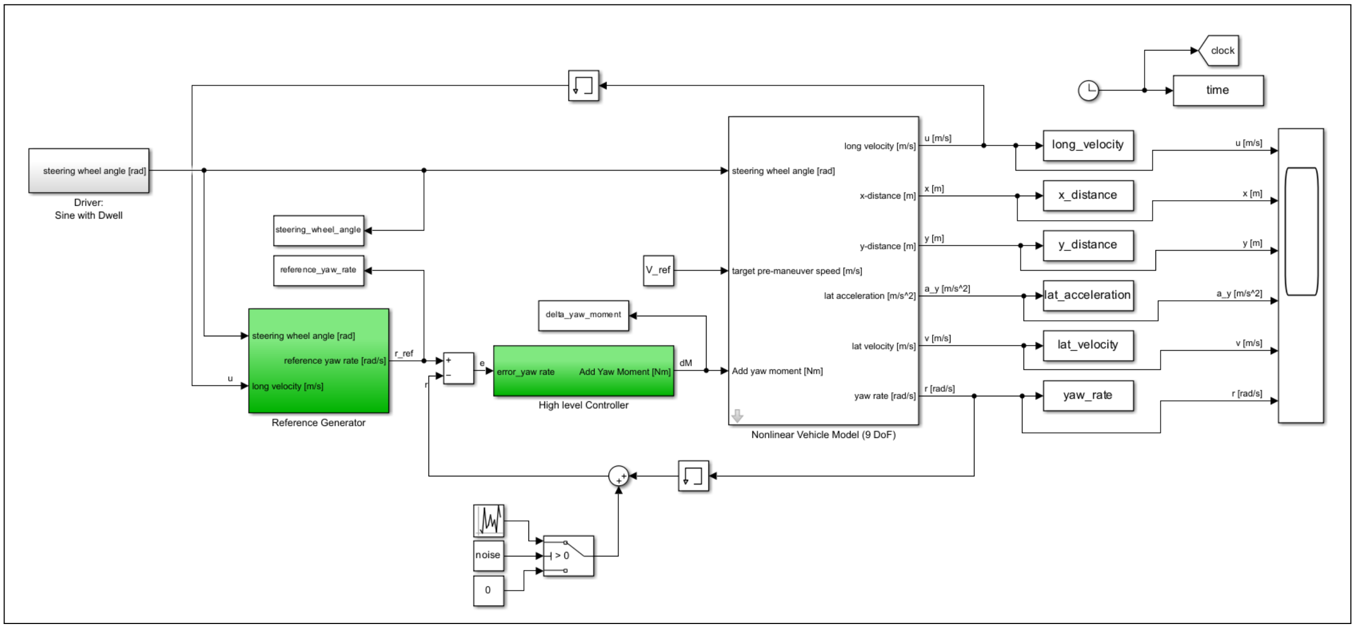


Figure 14: VSC system architecture implementation

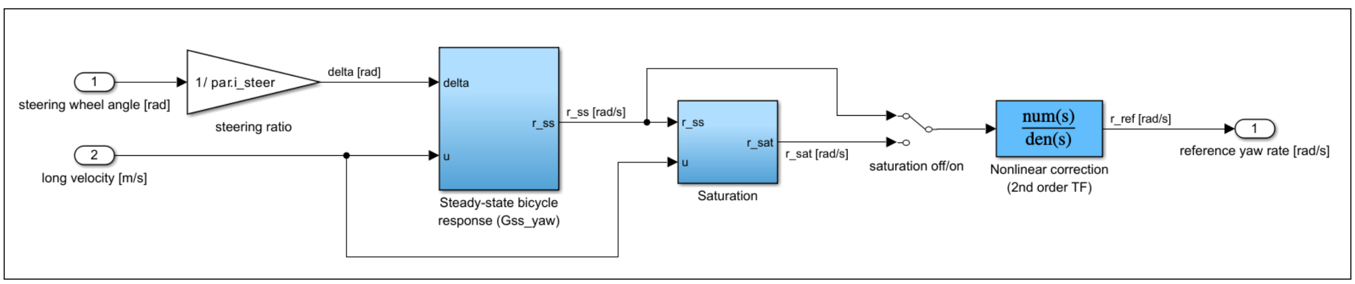
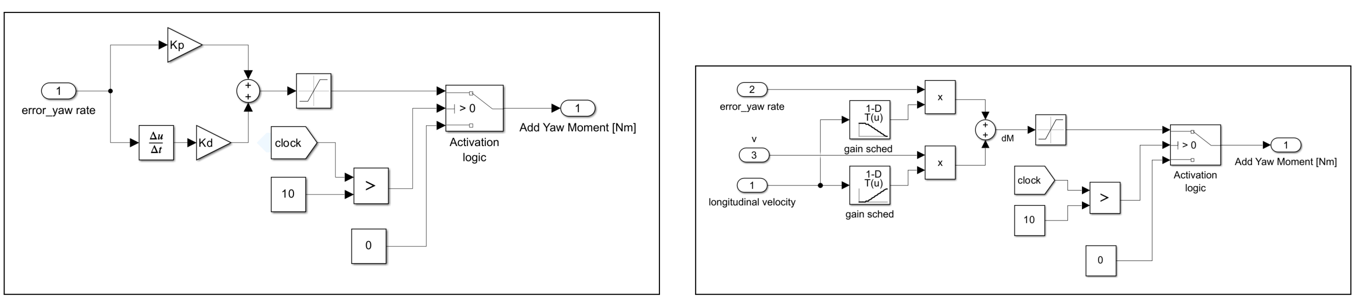


Figure 15: Reference genertor block



(a) PD Controller

(b) LQR Controller

Figure 16: High level controller block

C Appendix - Effect of the *ay-limiter* saturation block

Here some plots show the effect of the a_y-limiter component assuming a constant friction coefficient $\mu = 1$. As explained in subsection 1.1, its role is to limit the reference yaw rate by accounting for the available friction. Other limitations can be imposed, for example by looking at the vehicle sideslip angle β, but they are not considered here. Of course, this requires an estimation of those parameters, that are not directly measurable.

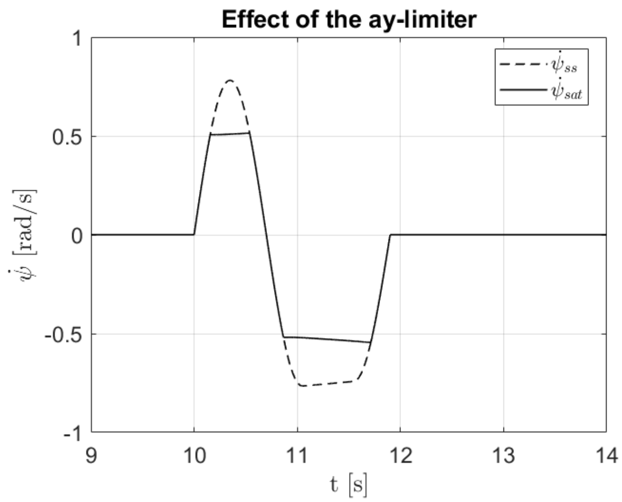
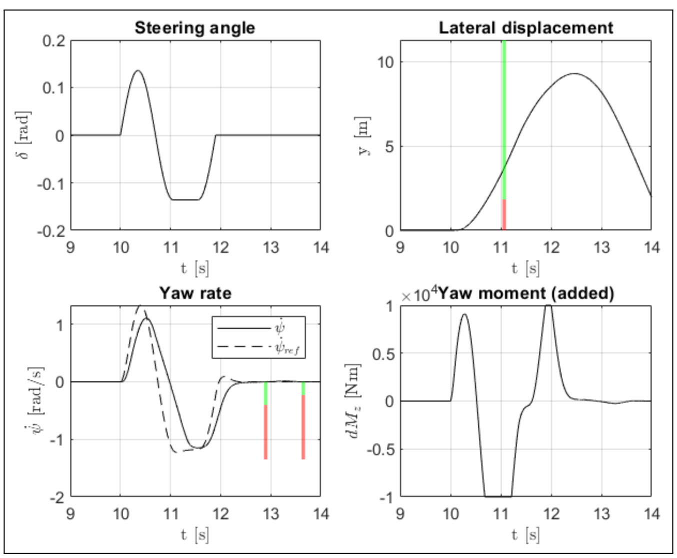
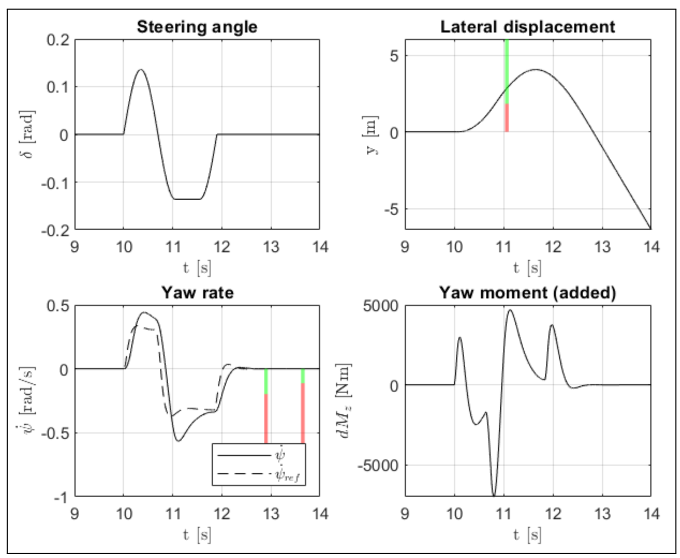


Figure 17: Saturation imposed by the limiter. Notice how the limit level depends on the longitudinal velocity, according to Equation (1)*.*



(a) No a_y-limiter



(b) With a_y-limiter

Figure 18: Simulations with PD controller and V_0 = 60 Km/h, with and without the limitation block. Notice in particular how the maximum lateral displacement y is much lower in the second case. Also, the control action is more limited.