Robust Control of Mechatronic Systems
Lab 1: Robust Lateral Control of an Autonomous Vehicle

Students: TIRACH Raquel, SCHILD Lukas, and TRAN Gia Quoc Bao

Date: 10 January 2021

1 Introduction

This tutorial aims to understand reference generation for an autonomous car and apply, first state-feedback linear time-invariant (LTI) control, and then state-feedback linear parameter-varying (LPV) control to a practical problem: the lateral control of an autonomous vehicle. During this tutorial, both the kinematic and the dynamic models are used.

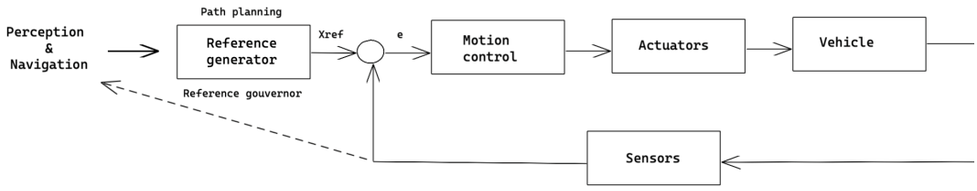

Figure 1: Lateral control of autonomous vehicles.

2 System modeling

2.1 Kinematic model

This is based on the assumption that the tire-slip of the vehicle with the ground is null, e.g., in low-speed systems such as mobile robots or vehicles in automatic parking tasks. The kinematic equations are:

$$\begin{cases} \dot{x} = v_x \cdot cos\psi \\ \dot{y} = v_x \cdot sin\psi \end{cases} \tag{1}$$

where x, y and ψ denote the position in meters (m) and the orientation in radians (rad), respectively, with respect to the global frame (x, y), and v_x and $\dot{\psi}$ denote the longitudinal velocity in m/s and the yaw rate in rad/s, respectively.

2.2 Dynamic model

The state-space representation of the vehicle lateral dynamics is:

$$\Sigma : \begin{cases} \begin{bmatrix} \dot{v}_y \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{C_f+C_r}{mv_x} & -v_x + \frac{-C_fL_f+C_rL_r}{mv_x} \\ \frac{-C_fL_f+C_rL_r}{I_zv_x} & -\frac{C_fL_f^2+C_rL_r^2}{I_zv_x} \end{bmatrix} \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \frac{C_f}{m} \\ \frac{C_fL_f}{I_z} \end{bmatrix} \delta =: Ax + Bu \\ y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} =: Cx \end{cases} \tag{2}$$

As we do not focus on building a state observer, we consider that both states are measured.

3 Reference generation

3.1 Kinematic model-based path planning

We suppose that the road curvature is known and given at every time instant and that the longitudinal velocity of the car is constant and known.

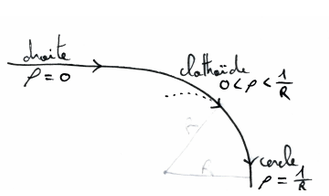

Figure 2: Road curvature.

Thanks to this data, we are able to calculate the yaw rate:

$$\dot{\psi} = \frac{v_x}{R} = \rho v_x \tag{3}$$

Once the yaw rate has been calculated, the centrifugal force can be given in terms of the road curvature since the lateral acceleration is null (only for a nominal small slip angle):

$$|F_c| = |m\dot{v_y}| = |mv_x\dot{\psi}| = |\frac{mv_x^2}{R}| \tag{4}$$

The lateral contact forces compensate the centrifugal ones.

3.2 Simulation

In MATLAB, the reference generator was simulated for a sampling period of $T_s = 2s$, a longitudinal velocity of $v_x = 10m/s$ and a road curvature of ρ_r =[0 0 0 0 0 0 0 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 0.010 0.009 0.008 0.007 0.006 0.005 0.004 0.003 0.002 0.001 0 0 0 0 0 0 0]. We use Eq.(3) to calculate the yaw velocity and then integrate it to get the yaw angle of the reference trajectory as follows:

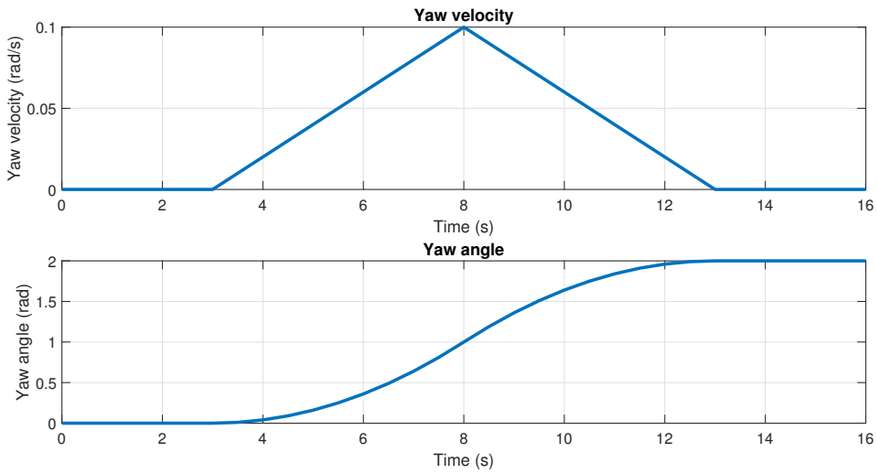

Figure 3: Reference yaw velocity and angle.

Using the kinematic model (Eq.(1)), we calculate the longitudinal and lateral velocities for this generated reference trajectory.

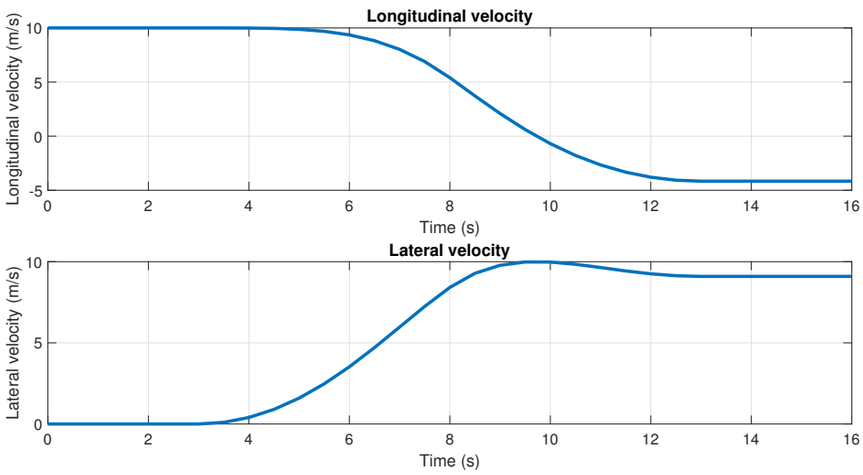

Figure 4: Reference longitudinal and lateral velocities.

For better visualization of the generated trajectory, we integrate the longitudinal and lateral velocities into (x, y) positions:

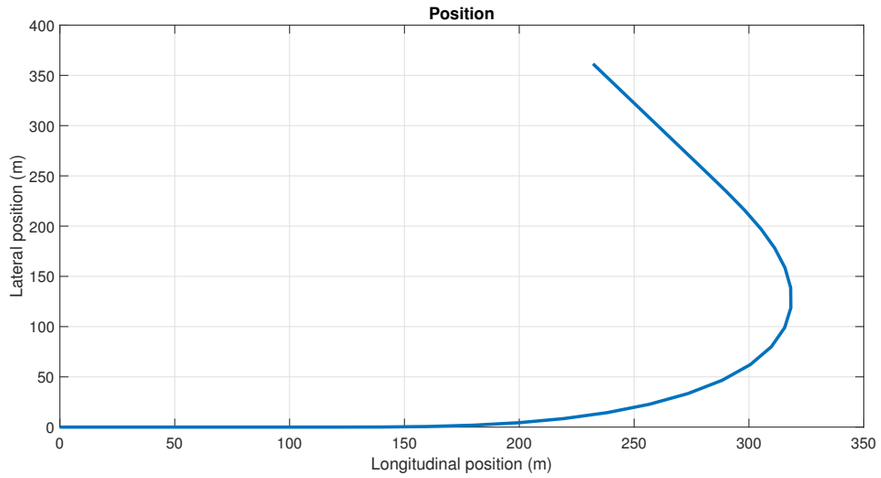

Figure 5: Reference trajectory.

Finally, Eq.(4) gives us the centrifugal force, which is compensated by the lateral contact forces. Fig.(6) shows that the centrifugal force does not exceed 1700N for this designed trajectory, which is below the maximum value beyond which the contact forces fail to compensate the centrifugal forces, and the car goes off-road. This maximum value can be estimated using the following compensation:

$$\frac{mv^2}{R} = \mu mg \tag{5}$$

The maximum force is approximately 7900N for a 1621kg car and a friction coefficient $\mu = 0.5$. Now that the reference has been generated, the autonomous vehicle's motion has to be controlled (see Fig.(1)).

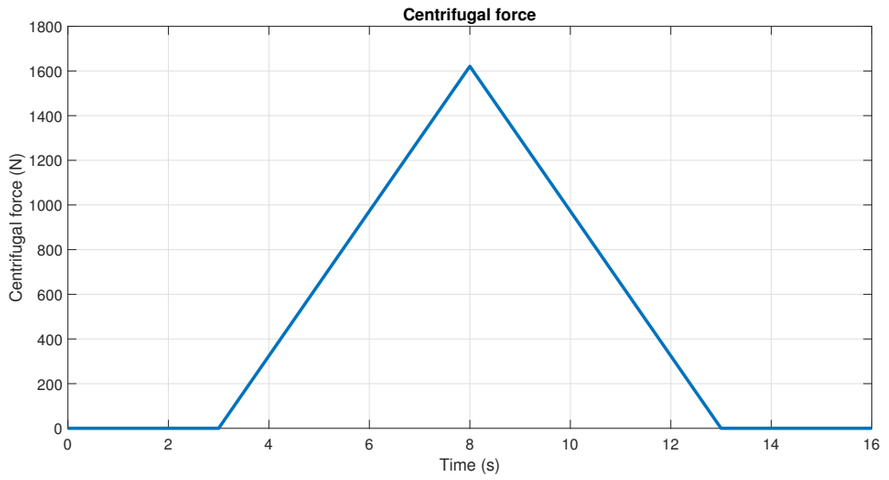

Figure 6: Centrifugal force.

4 Robust lateral control of an autonomous vehicle

In this part, we design several controllers in the two cases LTI and LPV.

4.1 LTI case

First, we consider the system to be linear time-invariant (LTI), i.e., the value of v_x, and all the other parameters are constant.

4.1.1 State-feedback LQR approach with integral action

Based on the dynamical bicycle model, we propose a state-feedback LQR controller with integral action so as to reject the effects of input disturbance. Let us first extend the system

by adding the integral term $z(t) = \int_0^t (r(\eta) - y(\eta))\, d\eta$ as:

$$\Sigma_e : \begin{cases} \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r + \begin{bmatrix} E \\ 0 \end{bmatrix} d \\ y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \end{cases} \tag{6}$$

where the matrices are as above and E is the gain of the disturbance d that we wish to reject, and r is the reference generated by motion planning. We are using $E = \begin{bmatrix} 0 \\ \frac{1}{I_z} \end{bmatrix}$, which means that the disturbance is a torque on the yaw rate.

For the LQR approach, the weighting functions Q and R were defined as follows:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{bmatrix} \text{ and } R = 0.01$$

We choose a small enough R in order to increase the controller's speed at the expense of cost, and Q has been chosen such that there is more weight on the integral term, i.e., we want to emphasize rejection of the effects of the input disturbance. The following figures shows the results of simulation using MATLAB:

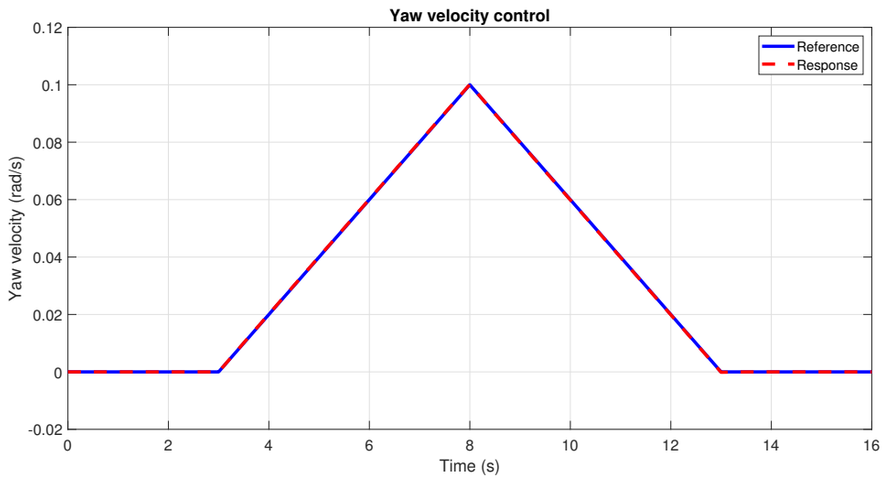

Figure 7: Yaw velocity in the case of state-feedback LQR.

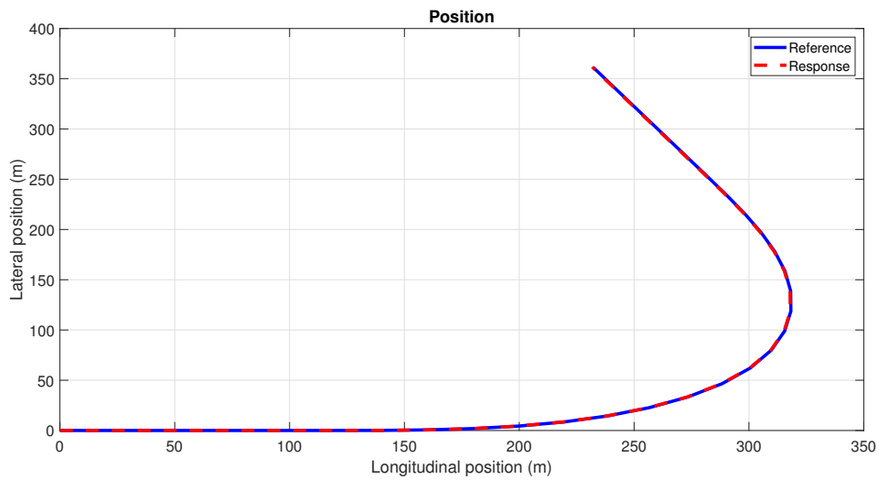

Figure 8: Trajectory in the case of state-feedback LQR.

Fig.(7) and Fig.(8) show that the LQR integral controller performs correctly. We then implement the $\mathcal{H}_\infty$ approach since we want our controlled system to be robust against disturbances and system modeling uncertainties.

4.1.2 State-feedback $\mathcal{H}_\infty$ approach

This part concerns the LTI system controlled using the state-feedback $\mathcal{H}_\infty$ approach.
To calculate the $\mathcal{H}_\infty$ controller, we use the `K = H_inf_control_design(A,B,E)` function which uses the bounded real lemma to compute a state-feedback $\mathcal{H}_\infty$ controller $u(k) = -Kx(k)$ for discrete-time linear systems written in the form $x(k+1) = A_d x(k) + B_d u(k) + E_d w(k)$ where $w(k)$ denotes a bounded input disturbance (this function requires the CVX optimization toolbox). For this, we do not use any frequency-domain templates on the response and the control input.
Before using this function, we discretize the model using the `c2d` function in MATLAB. To ensure a unit static gain that is neccessary for tracking the reference, the second part of the controller is $Gr(k)$ where $G = (C_d(I - A_d + B_d K)^{-1} B_d)^{-1}$ (we are calculating the gains with the discrete-time system).
In Simulink, we build a model described in the Appendix to compare between the LQR integral and state-feedback $\mathcal{H}_\infty$ control. When we have no input disturbance, the $\mathcal{H}_\infty$ one tracks the reference so fast that it overlapses the reference.

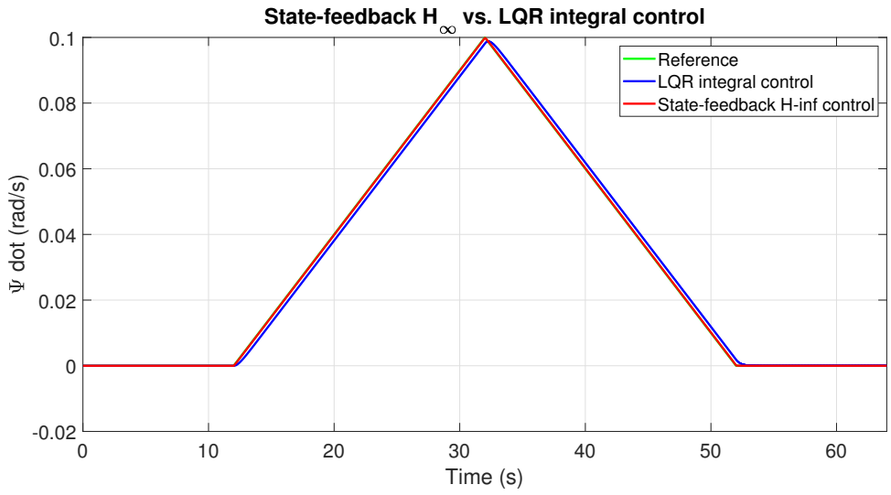

Figure 9: System response $\dot{\psi}$ for LQR integral and $\mathcal{H}_\infty$ control without input disturbances.

Besides, the control input is smoother, which shows that state-feedback $\mathcal{H}_\infty$ control gives an overall better time-domain performance.

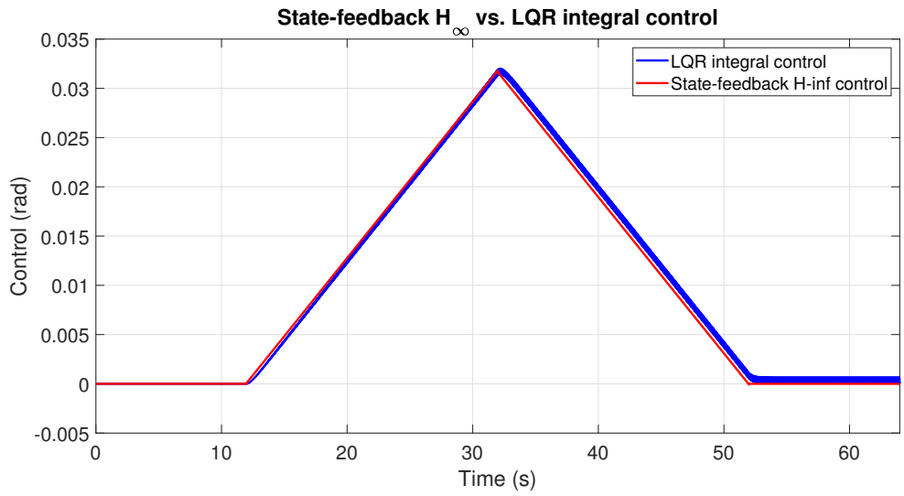

Figure 10: Control input for LQR integral and state-feedback $\mathcal{H}_\infty$ control without input disturbances.

We also see that even if the state-feedback $\mathcal{H}_\infty$ is faster than the LQR integral approach, it cannot reject input disturbances while LQR integral control can:

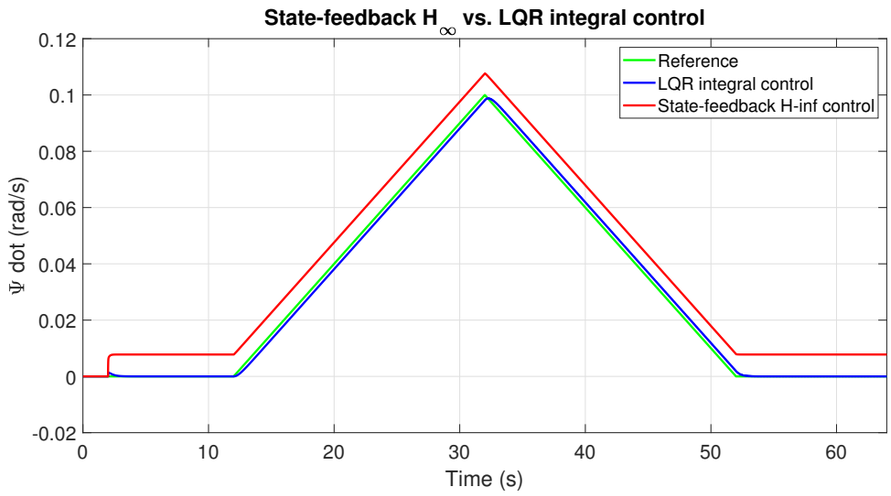

Figure 11: System response $\dot{\psi}$ for LQR integral and state-feedback $\mathcal{H}_\infty$ control with input disturbances.

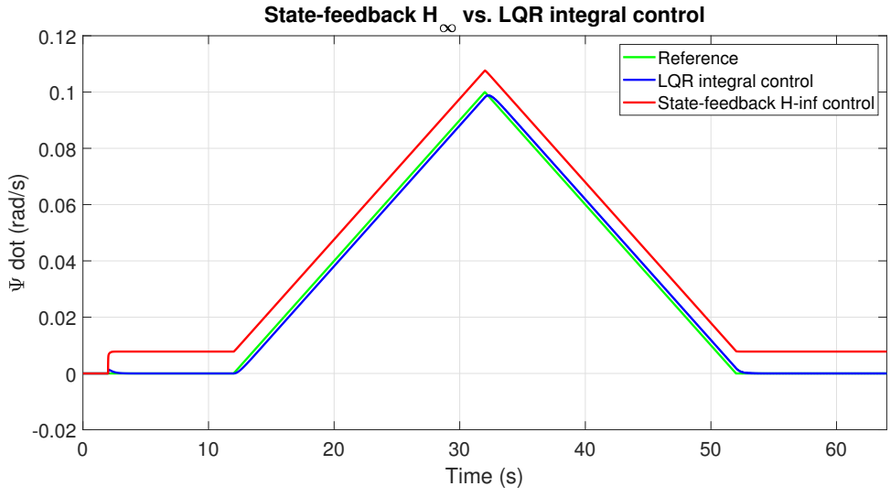

Figure 12: Control input for LQR integral and state-feedback $\mathcal{H}_\infty$ control with input disturbances.

Even if the $\mathcal{H}_\infty$ approach is faster and more robust, in an LTI case it can be preferable to use LQR integral control since is performs correctly and is less complicated.

4.2 LPV case

In this part, we treat v_x as a varying parameter and consider the system as an LPV one. Note that as $1/v_x$ appears in the state-space representation, we have two varying parameters which are $\rho_1 = v_x \in [v_{x,min}, v_{x,max}] \equiv [\underline{\rho_1}, \overline{\rho_1}]$ and $\rho_2 = 1/v_x \in [1/v_{x,max}, 1/v_{x,min}] \equiv [\underline{\rho_2}, \overline{\rho_2}]$. In our simulations, we take $v_{x,min} = 5m/s$ and $v_{x,max} = 15m/s$. In reality, it is obvious that the two cases $(\underline{\rho_1}, \underline{\rho_2})$ and $(\overline{\rho_1}, \overline{\rho_2})$ never happen. Doing this, we are being overly conservative, but it is necessary to ensure the theories of polytopic LPV systems.

4.2.1 Polytopic LPV model

The system is expressed in polytopic form as:

$$\Sigma(\rho) = \sum_{i=1}^{4} \alpha_i(\rho)\Sigma_i, \qquad \alpha_i(\rho) \geq 0, \qquad \sum_{i=1}^{4} \alpha_i(\rho) = 1. \tag{7}$$

where $\Sigma_1 = \Sigma(\underline{\rho_1}, \underline{\rho_2})$, $\Sigma_2 = \Sigma(\underline{\rho_1}, \overline{\rho_2})$, $\Sigma_3 = \Sigma(\overline{\rho_1}, \underline{\rho_2})$, and $\Sigma_4 = \Sigma(\overline{\rho_1}, \overline{\rho_2})$. We form a grid-based controller with four vertices.

$$u = -F(\rho) \cdot x + G(\rho) \cdot r \tag{8}$$

The controller is expressed in polytopic form as:

$$K(\rho) = \sum_{i=1}^{4} \alpha_i(\rho) K_i, \quad \alpha_i(\rho) \geq 0, \quad \sum_{i=1}^{4} \alpha_i(\rho) = 1. \tag{9}$$

where $K_1 = K(\underline{\rho_1}, \underline{\rho_2})$, $K_2 = K(\underline{\rho_1}, \overline{\rho_2})$, $K_3 = K(\overline{\rho_1}, \underline{\rho_2})$, and $K_4 = K(\overline{\rho_1}, \overline{\rho_2})$. The four corresponding interpolation functions are:

$$\alpha_1 = \frac{(\overline{\rho_1} - \rho_1)(\overline{\rho_2} - \rho_2)}{(\overline{\rho_1} - \underline{\rho_1})(\overline{\rho_2} - \underline{\rho_2})}, \alpha_2 = \frac{(\overline{\rho_1} - \rho_1)(\rho_2 - \underline{\rho_2})}{(\overline{\rho_1} - \underline{\rho_1})(\overline{\rho_2} - \underline{\rho_2})}, \alpha_3 = \frac{(\rho_1 - \underline{\rho_1})(\overline{\rho_2} - \rho_2)}{(\overline{\rho_1} - \underline{\rho_1})(\overline{\rho_2} - \underline{\rho_2})}, \alpha_4 = \frac{(\rho_1 - \underline{\rho_1})(\rho_2 - \underline{\rho_2})}{(\overline{\rho_1} - \underline{\rho_1})(\overline{\rho_2} - \underline{\rho_{c2}})} \tag{10}$$

4.2.2 LPV control

In order to calculate the four controllers K_1, K_2, K_3 and K_4, we modify the previously used `K = H_inf_control_design(A,B,E)` function such that it solves the following LMI sets U and Y_i minimizing γ (since we want to minimize the impact of the disturbance w_k):

$$\begin{bmatrix} -U & 0 & UA_i - Y_i B_i^T & U \\ \star & -\gamma^2 I & E^T & 0 \\ \star & \star & -U & 0 \\ \star & \star & \star & -Q^{-1} \end{bmatrix} \preceq 0, i \in \{1; 2; 3; 4\} \tag{11}$$

Then to ensure a unit static gain, $G_i = (C_i(I - A_i + B_i K_i)^{-1} B_i)^{-1}$. In this case, we also discretize our four systems in order to compute the four controllers.
The performance of this $\mathcal{H}_\infty$-LPV controller is tested with a different Simulink scheme (see Appendix). To test the effectiveness of using an LPV model, we let the parameter v_x vary from its minimum value of $5m/s$ to $13.8m/s$, from 20s to 64s.
The results are as follows, which shows that the yaw rate can track the reference:

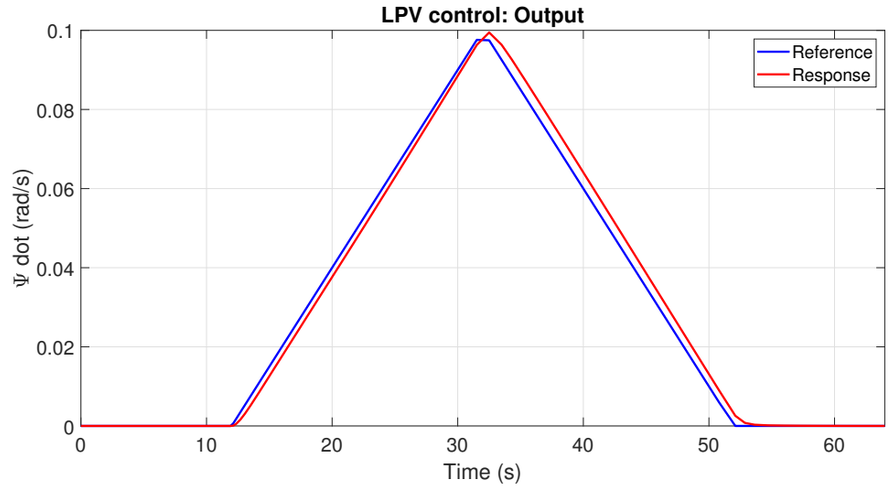

Figure 13: System response $\dot{\psi}$ with $\mathcal{H}_\infty$ controller for an LPV system.

with a sufficiently small control input:

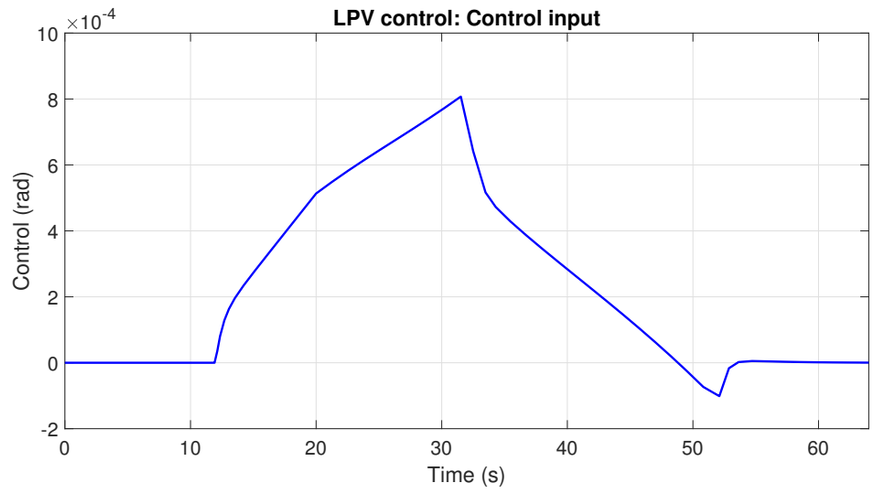

Figure 14: Control input for $\mathcal{H}_\infty$ controller for an LPV system.

And the vehicle positions can also track the desired trajectories.

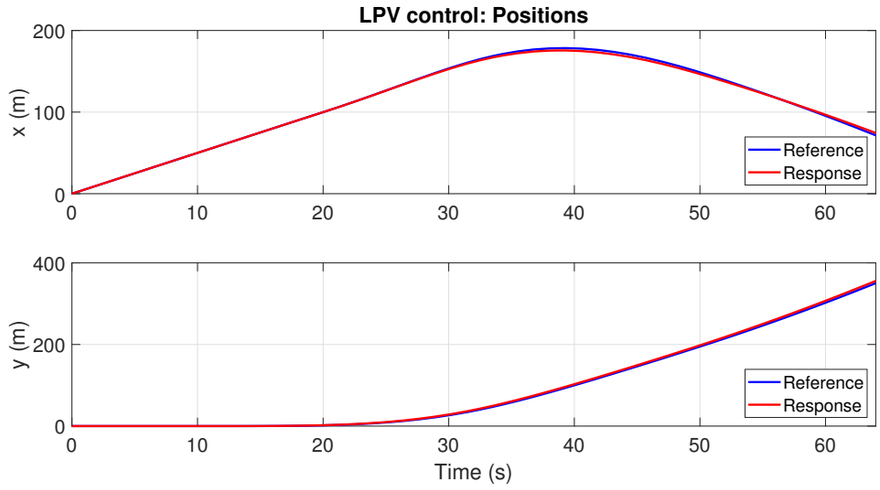

Figure 15: System response in positions x and y for $\mathcal{H}_\infty$ controller for an LPV system.

The centrifugal force is also not too large and it tracks the reference.

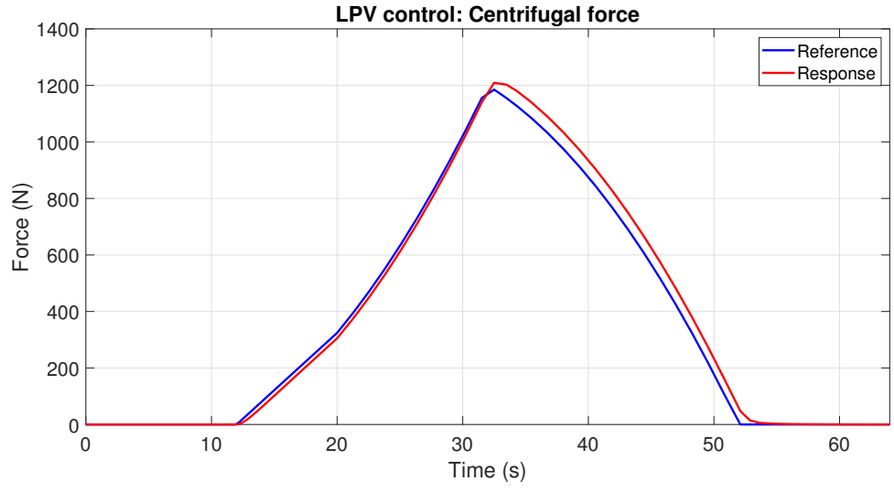

Figure 16: System response in centrifugal force for $\mathcal{H}_\infty$ controller for an LPV system.

Finally, we want to see how the interpolation functions $\alpha_i(\rho)$ work. Note that here as the two varying parameters ρ_1 and ρ_2 are not independent, i.e., $\rho_1\rho_2 = 1$, the vertices $(\underline{\rho_1}, \underline{\rho_2})$ and $(\overline{\rho_1}, \overline{\rho_2})$ never happen.

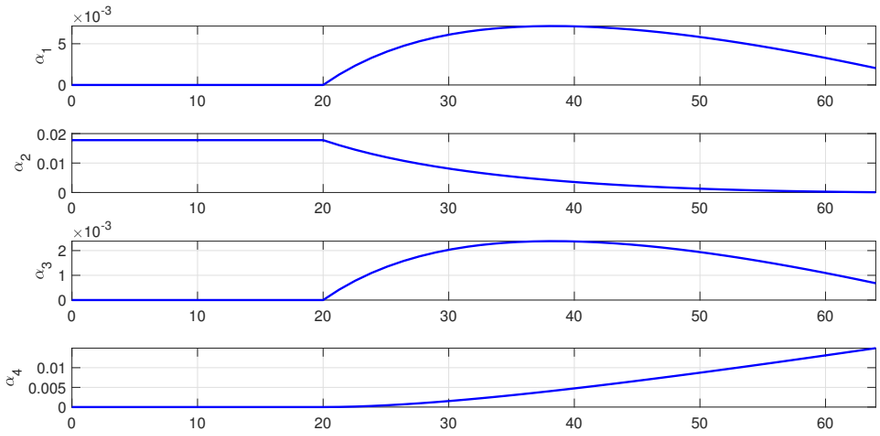

Figure 17: Interpolation functions for $\mathcal{H}_\infty$ controller for an LPV system.

Therefore, we conclude that LPV control works satisfactorily.

5 Conclusion

After this lab, the students understand better the robust approach in control, both in LTI and polytopic LPV forms. We know how to design controllers using LMIs and express the systems and controllers in LPV form.

6 Appendix: Simulink schemes and explanations

Here we put the Simulink schemes and some explanations. Simulation parameters are as follows: $C_f = 57117Nrad^{-1}$, $C_r = 81396Nrad^{-1}$, $L_f = 1.15m$, $L_r = 1.38m$, $I_z = 1975kgm^2$, $m = 1621kg$, and $g = 9.8m/s^2$.
First, for the LTI case, we compare the performance of two controllers using the following scheme. We build the `centrifugal` and `kinematic` blocks for calculating the centrifugal force and implementing the kinematic model, respectively. The input disturbance (if exists) is the same for both controllers.

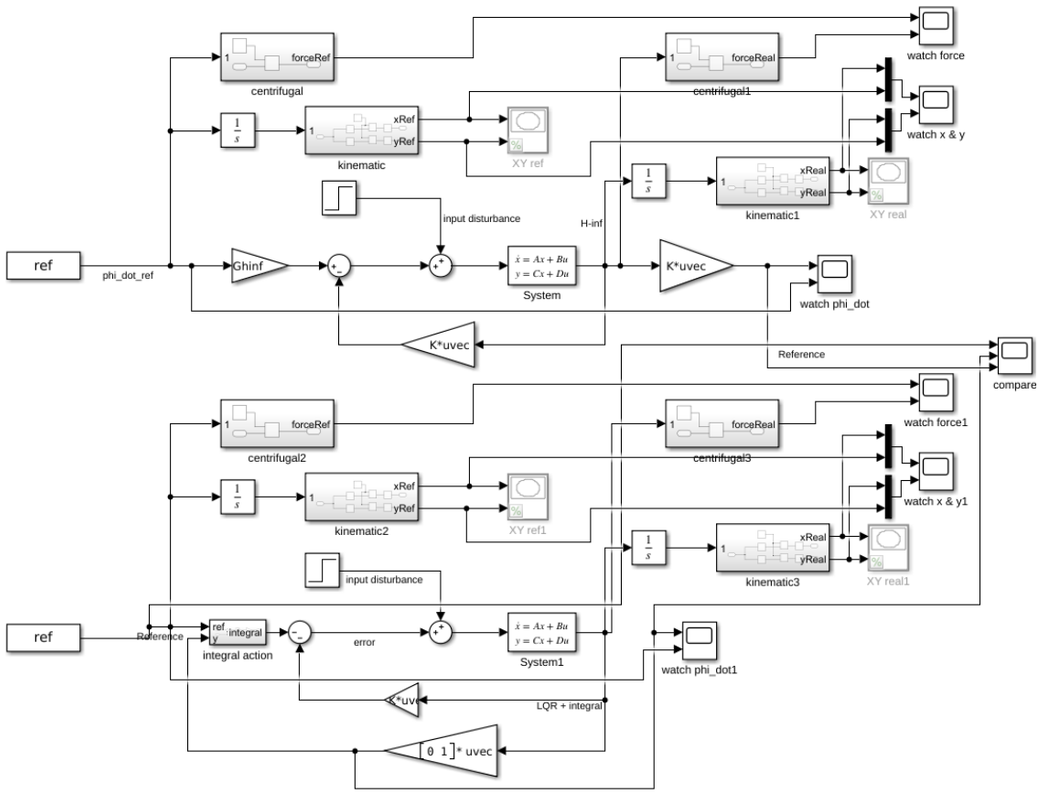

Figure 18: Simulink scheme for the LTI case (LQR integral and state-feedback $\mathcal{H}_\infty$ control).

For the LPV case, the matrix A of the system varies as described above, and so does the feed-forward and feedback gains. Note that v_x is varying and therefore it appears in the blocks for the centrifugal force and the kinematic model as well.

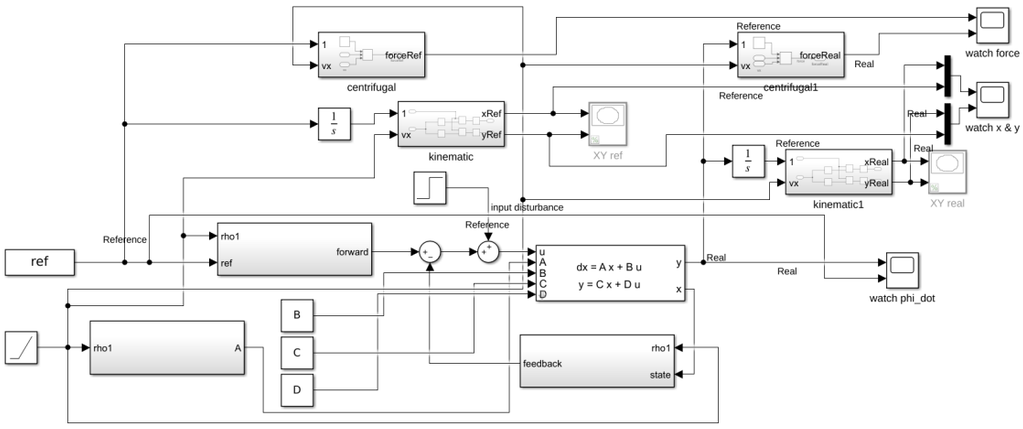

Figure 19: Simulink scheme for the LPV case.

The varying matrices are scheduled using the following block:

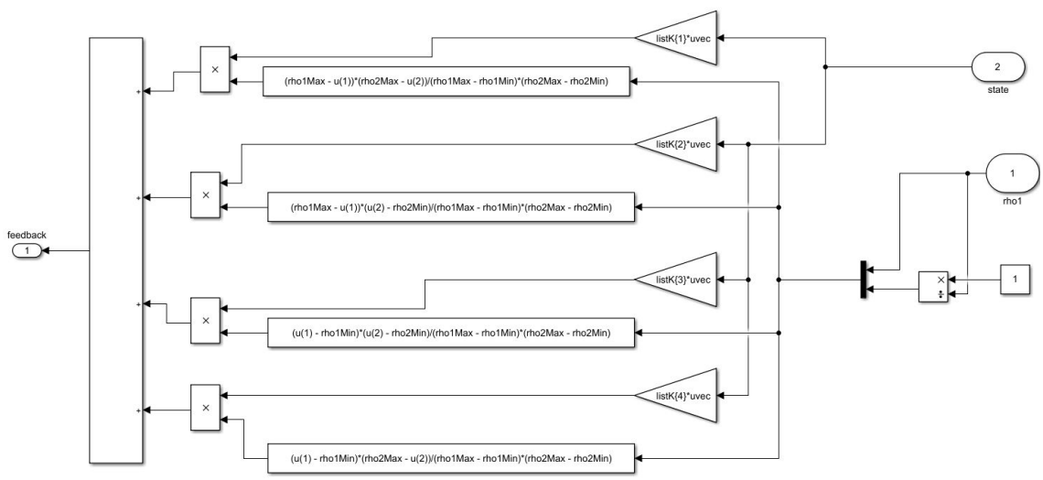

Figure 20: Interpolation functions for the LPV case.

In the block above, the interpolation functions are implemented and they give α_i from ρ.