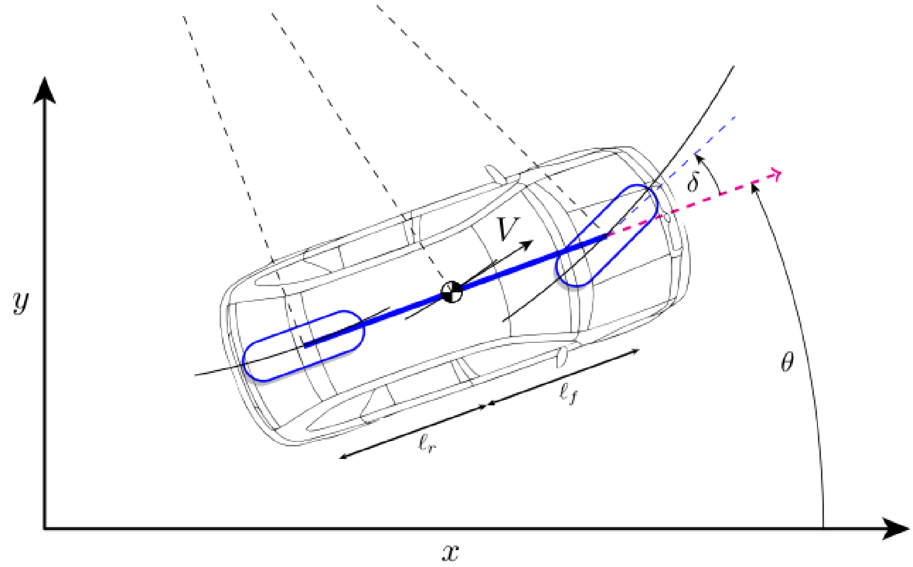
MINI-PROJECT REPORT

ME-425

MPC cruise controller for a car on a highway



Tomas Joaquin Garate Anderegg 313984
Matas Antanas Jones 313222
Louis Gilles Canoen 312526

January 9, 2025

Contents

1 System Dynamics

In this project, the aim is to develop a MPC controller for a simulated VW ID.3 car driving on a highway. The project leverages a four-state kinematic car model provided as part of the setup, encompassing the car's x and y position, heading and velocity. The objectives include designing both linear and nonlinear MPC controllers to achieve smooth and responsive driving behavior while adhering to physical and safety constraints such as lane boundaries and collision avoidance. The ultimate goal is to implement and test the controller for various maneuvers, including cruising, lane changes, and overtaking, ensuring comfort and safety for the passengers.

1.1 System definition

The system is defined by a four-state kinematic model. The state vector is defined as:

$$\mathbf{x} = [x,\, y,\, \theta,\, V]^T \tag{1}$$

The input vector is defined as:

$$\mathbf{u} = [\delta,\, u_T]^T \tag{2}$$

The complete state equations are:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} V\cos(\theta + \beta) \\ V\sin(\theta + \beta) \\ \frac{V}{\ell_r}\sin(\beta) \\ \frac{F_{\text{motor}} - F_{\text{drag}} - F_{\text{roll}}}{m} \end{bmatrix} \tag{3}$$

where:

$$\beta = \arctan\left(\frac{\ell_r\,\tan(\delta)}{\ell_r + \ell_f}\right), \quad F_{motor} = \frac{u_t\,P_{max}}{V}, \quad F_{drag} = \frac{1}{2}\,\rho\,C_d\,A_f\,V^2, \quad F_{roll} = C_r\,m\,g \tag{4}$$

2 Linearization

2.1 Deliverable 2.1

The linearized dynamics can be found through a Taylor series expansion up to first order:

$$f(\boldsymbol{x}, \boldsymbol{u}) \approx f(\boldsymbol{x}_s, \boldsymbol{u}_s) + A(\boldsymbol{x} - \boldsymbol{x}_s) + B(\boldsymbol{u} - \boldsymbol{u}_s) \tag{5}$$

At steady-state: $\boldsymbol{x}_s = (0, 0, 0, V_s)$ and $\boldsymbol{u}_s = (0, u_{T,s})$ where:

$$A = \frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\bigg|_{(\boldsymbol{x}_s, \boldsymbol{u}_s)} \quad \text{and} \quad B = \frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\bigg|_{(\boldsymbol{x}_s, \boldsymbol{u}_s)} \tag{6}$$

$$f(\mathbf{x_s}, \mathbf{u_s}) = \begin{bmatrix} V_s \\ 0 \\ 0 \\ \frac{1}{m} q(\frac{u_{T,s} P_{max}}{V_s} - \frac{1}{2}\rho C_d A_f V_s^2 - C_r mg) \end{bmatrix} \tag{7}$$

By taking the Jacobian with respect to x and u, the matrices A and B are obtained:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & V_s & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{m}(\frac{u_{T,s} P_{max}}{V_s^2} + \rho C_d A f V_s) \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 \\ \frac{l_r}{l_r + l_f} V_s & 0 \\ \frac{V_s}{l_r + l_f} & 0 \\ 0 & \frac{P_{max}}{m V_s} \end{bmatrix} \tag{8}$$

2.2 Deliverable 2.2

The longitudinal and lateral dynamics of the car can be treated independently due to the weak coupling between acceleration and steering. Longitudinal control, influenced by the throttle input u_T, directly affects the vehicle's speed and acceleration. In contrast, lateral control, determined by the steering angle δ, influences the vehicle's heading and lateral position. The forces considered in this analysis are assumed to be parallel to the vehicle's speed, with lateral forces from steering neglected. Additionally, assuming small heading angles θ, the vehicle's speed can be approximated as nearly horizontal, which further simplifies the analysis. Linearization around steady-state conditions confirms that cross-coupling between the subsystems is negligible, allowing for the design of separate controllers for each subsystem.

3 Design MPC (Deliverable 3.1)

3.1 Design procedure

First, the state and control constraints of the discretized problem are defined:

Longitudinal subsystem:

$$x^+ = x_s + A_d\,(x - x_s) + B_d\,(u - u_s) \tag{9}$$

where x_s (2×1) is the steady state of the x coordinate and the speed V, u_s (1×1) is the steady state input when V_{ref} is reached, A_d (2×2) is the discretized matrix of the state dynamics, B_d (2×1) is the discretized matrix of the dynamics of the control input. The state and control constraints that must be satisfied at each step in the horizon are:

$$\text{Control constraints:} \quad M\,u \leq m, \quad \text{where} \quad M = (1,\,-1)^T,\, m = (1,\,1)^T \tag{10}$$

Lateral subsystem:

$$x^{+} = x_s + A_d\,(x - x_s) + B_d\,(u - u_s) \tag{11}$$

where x_s (2×1) is the steady state of the y coordinate and the heading angle θ, u_s is zero as the steady-state steering angle δ is 0, A_d (2×2) is the discretized matrix of the state dynamics, B_d (2×1) is the discretized matrix of the dynamics of the steering angle δ . The state and control constraints that must be satisfied at each step in the horizon are:

$$\text{State constraints:} \quad F\,x \leq f, \quad \text{where} \quad F = \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}, \, f = \begin{pmatrix} 3.5 & 0.5 \\ 5^{\circ} & 5^{\circ} \end{pmatrix} \tag{12}$$

$$\text{Control constraints:} \quad M\,u \leq m, \quad \text{where} \quad M = (1,\, -1)^{T},\, m = (30^{\circ},\, 30^{\circ})^{T} \tag{13}$$

3.2 Choice of tuning parameters

The choice of the different tuning parameters depend on many factors such as the computing time for the horizon length H and on the car's reactiveness with respect to the state and input cost matrices Q and R (reaction too aggressive, presence of overshoots, ...). This is how each parameter affects the system:

3.2.1 Length of horizon H

The sampling time is defined as T_s and the number of steps of the horizon for the MPC controller N is simply defined as:

$$N = \frac{H}{T_s} \tag{14}$$

For a horizon length of $H = 10$ s and $T_s = 0.1$, the number of steps in the horizon would be equal to 100. The relationship between the computation time and the length of the horizon is approximately linear.

Since the problem is not too complex (no disturbance, no other car, ...), taking a short horizon doesn't affect the stability of the controller. As seen in Figures 1 and 2, no noticeable differences can been seen between a horizon of 0.2 s and 10 s (arbitrary values):

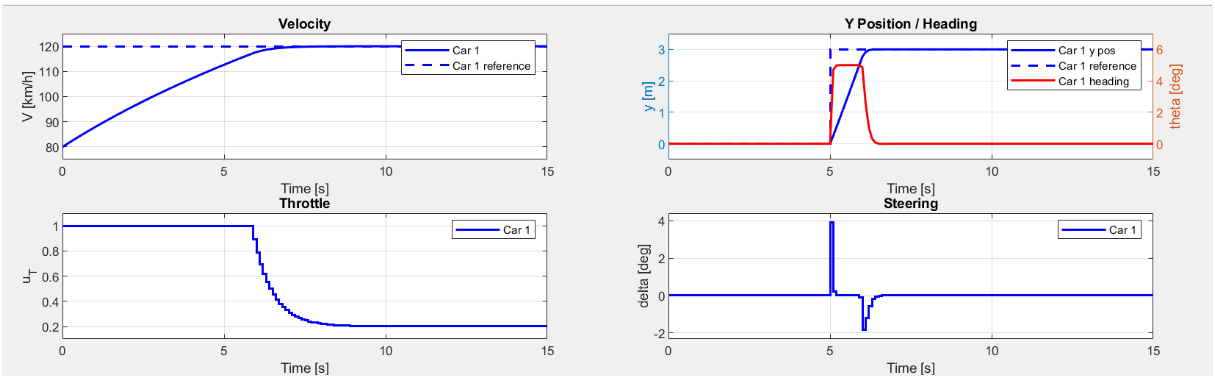


Figure 1: Car changing lane after 5 seconds ($H = 0.2$ s, $Q = I$ and $R = 1$).

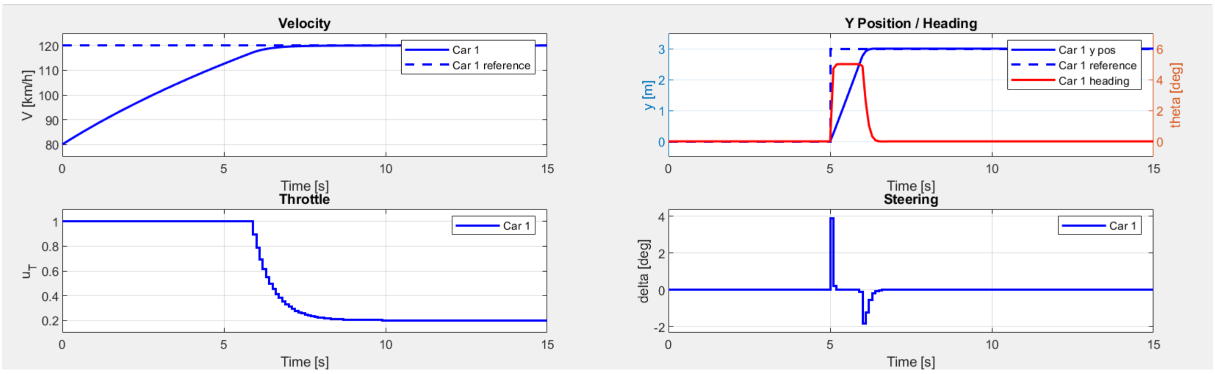


Figure 2: Car changing lane after 5 seconds ($H = 10$ s, $Q = I$ and $R = 1$).

The graphs in section §3 are the result of the MPC controller attempting to follow a first reference of 120 km/h and $y = 0$, starting from 80 km/h and $y = 0$. Then after 5 seconds, the reference changes to 120 km/h and $y = 3$.

3.2.2 State cost matrix Q and control cost matrix R

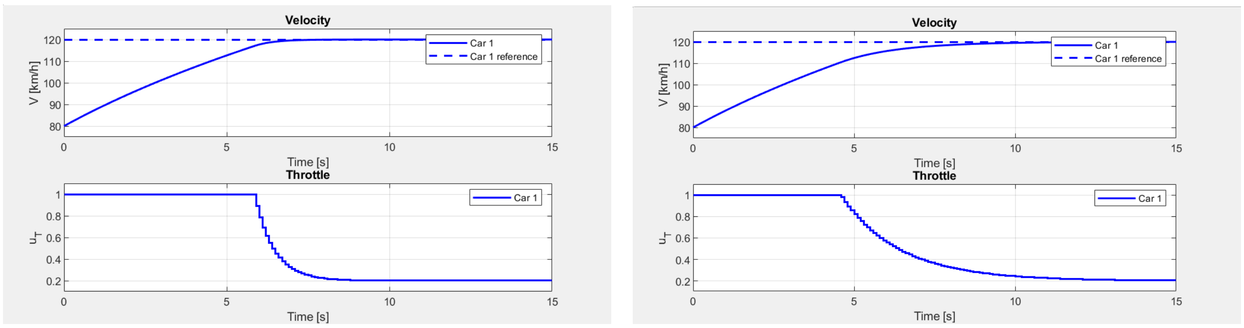
The matrix Q and R appear in the quadratic cost function to be optimized. It is defined as the minimum of the objective function:

$$J^*(x) = \min_{x,u} \sum_{i=0}^{N-1} (x_i - x_s)^T Q (x_i - x_s) + (u_i - u_s)^T R (u_i - u_s) + (x_N - x_s)^T Q_f (x_N - x_s) \quad (15)$$

where x_s is the steady state vector, u_s the steady state control input and Q_f the state cost matrix of the LQR controller used to compute the Nth stage cost. The Q and R matrices control the weights of the variables associated. This means that if the weight of the second state variable is larger than that of the first, the optimizer will prioritize minimizing the error of the second state. In the case of the lateral sub-controller, it means that if $Q_{(\theta,\theta)}$ is bigger than $Q_{(y,y)}$, the controller will more aggressively control θ instead of y.

The constraints given are a *settling time no more than 10 seconds when accelerating from 80 km/h to 120 km/h or 3 seconds doing a lane change.* Q and R can be tuned such that the car dynamics is smooth while satisfying the aforementioned constraints.

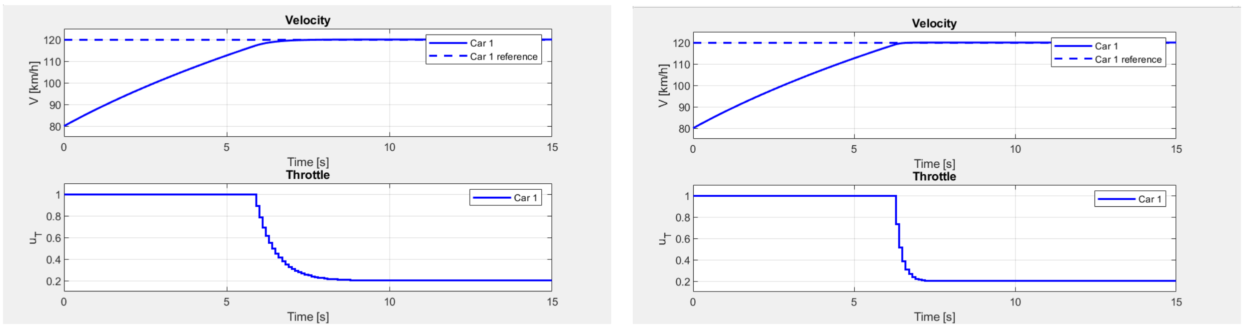
For the longitudinal sub-controller, R can be tuned such that the acceleration and deceleration are as smooth as possible while being able to accelerate from 80 km/h to 120 km/h in under 10 seconds. The horizontal speed only depends on the longitudinal sub-controller.



(a) $Q_{lon} = 1$, $R_{lon} = 1$, $T_{set} = 6.4$ s. (b) $Q_{lon} = 1$, $R_{lon} = 10$, $T_{set} = 8.4$ s.

Figure 3: Car accelerating from 80 to 120 km/h ($H = 10$ s).

As seen in Figure 3, the settling time (to be within 1% of the reference value) is $T_{set} = 8.4$ s for $R_{lon} = 10$ and $T_{set} = 6.4$ s for $R_{lon} = 1$. The maximum value of R_{lon} to satisfy the constraint is $R_{lon} = 19$ where the speed reaches 99% of the reference speed after 10 seconds. This behavior is expected because a higher value of R_{lon} penalizes the controller for keeping the throttle at high values for an extended period. As a result, the controller responds by releasing the accelerator sooner and more gradually compared to cases with lower values of R_{lon}.

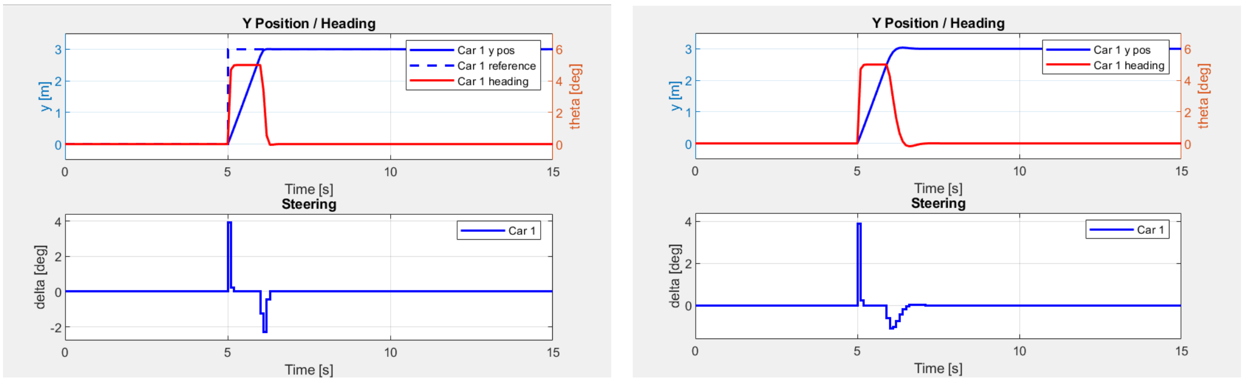


(a) $Q_{lon} = 1$, $R_{lon} = 1$, $T_{set} = 6.4$ s. (b) $Q_{lon} = 10$, $R_{lon} = 1$, $T_{set} = 6.2$ s.

Figure 4: Car accelerating from 80 to 120 km/h ($H = 10$ s).

The Q_{lon} matrix (which is a constant in the longitudinal sub-controller, as the x state is not controlled) can also be tuned, though the results are similar to tuning R_{lon} but with opposite effects. As seen in Figure 4, increasing Q_{lon} decreases the settling time by keeping the throttle at high values for longer, generating a more aggressive control.

Regarding the second constraint (lane changing in less than 3 seconds), Q_{lat} and R_{lat} have to be tuned accordingly. The lateral sub-controller controls the steering angle of the car depending on the state variables y and θ.



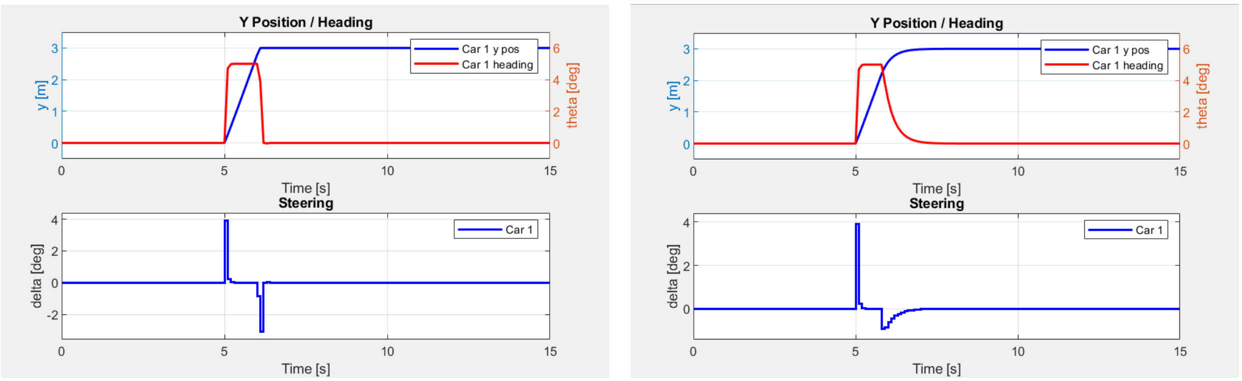
(a) $Q_{lat} = I$, $R_{lat} = 1$, $T_{set} = 1.1$ s. (b) $Q_{lat} = I$, $R_{lat} = 100$, $T_{set} = 1.5$ s.

Figure 5: Car changing lanes at 120 km/h ($H = 10$ s).

In Figure 5, there are two steering spikes per lane change. The first spikes in both graphs are identical since the y reference changes abruptly. However, the second spikes in both graphs are different due to the effect R.

However, the second spikes in both graphs differ because variations in R_{lat} influences the car dynamics. The figures show that a larger value of R_{lat} leads to smoother steering. Increasing R_{lat} leads to an overshoot in y and to and undershoot of θ. The steering becomes smoother but less dynamic.

With the lateral sub-controller, the weights of each state variable in Q_{lat} can be varied unlike for the longitudinal sub-controller where the speed is the only tunable state variable.



(a) $Q_{lat} = \text{diag}([100, 1])$, $R_{lat} = 1$, $T_{set} = 1.1$ s. (b) $Q_{lat} = \text{diag}([1, 100])$, $R_{lat} = 1$, $T_{set} = 1.8$ s

Figure 6: Car changing lanes at 120 km/h ($H = 10$ s).

The y reference value is reached faster and more aggressively when more weight is put on $Q_{(y,y)}$ but in return δ becomes sharper which could be unpleasant for the passengers of the car.

As seen in Figure 6, the undershoot in θ decreases and the steering becomes smoother when more weight is given to $Q_{(\theta,\theta)}$. However, the lane change is less dynamic and takes longer.

In order to satisfy the lane change constraint, the weight on $Q_{(\theta,\theta)}$ can be as big as 400 (y value within 1% of the reference value).

In conclusion, here are listed every tuning parameter and how they satisfy the constraints:

- Length of horizon $H = 10$ s ($N = 100$)
- Longitudinal control cost weight $R_{lon} = 1$
- Longitudinal state cost matrix $Q_{lon} = 1$
- Resulting settling time for speed (80 to 120 km/h) $T_{set} = 6.4$ s (< 10 s)
- Lateral control cost weight $R_{lat} = 1$
- Lateral state cost matrix $Q_{lat} = \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix}$
- Resulting settling time for y (lane change) $T_{set} = 1.8$ s (< 3 s)

These parameters were chosen as they offer a good compromise between dynamism and comfort as seen in Figure 7.

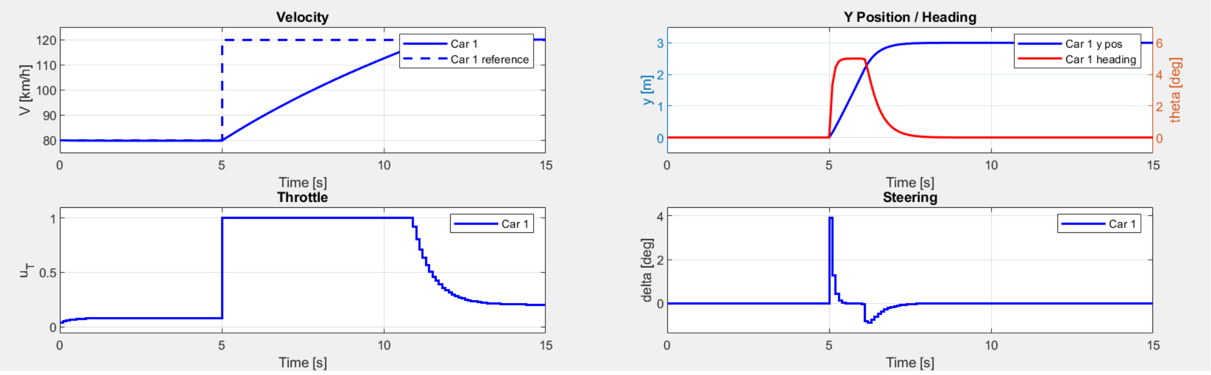
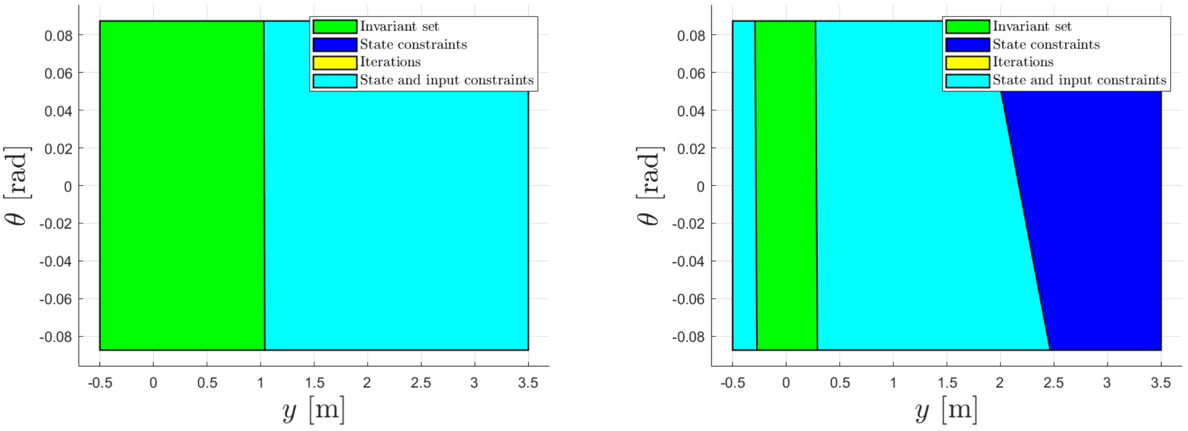


Figure 7: Closed-loop response of the car accelerating and changing lanes at t = 5 s.

3.3 Terminal invariant set

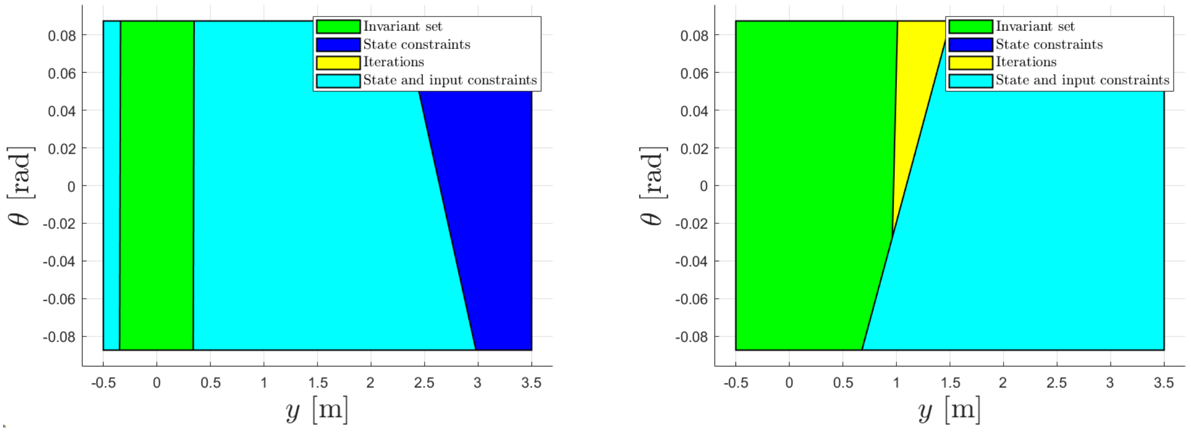
A larger terminal invariant set improves system feasibility. Increasing the weight of $Q_{(\theta,\theta)}$ expands the set as depicted in Figure 8, hence a large value of 100 for $Q_{(\theta,\theta)}$ was chosen. Whereas increasing the weight of $Q_{(y,y)}$ reduces the size of the set, reducing the controller feasibility. As seen in Figure 9, larger values of R_{lat} can also expand the terminal invariant set. However, they also increase the settling time, causing lane changes to take unnecessarily long. Thus the values of Q_{lat} and R_{lat} must be chosen as to get a good compromise between system dynamism and invariant set size.



(a) $Q_{lat} = \text{diag}([1,\ 100])$, $R_{lat} = 1$.

(b) $Q_{lat} = \text{diag}([100,\ 1])$, $R_{lat} = 1$.

Figure 8: Impact on terminal invariant set of Q_{lat}.



(a) $Q_{lat} = I$, $R_{lat} = 1$.

(b) $Q_{lat} = I$, $R_{lat} = 100$.

Figure 9: Impact on terminal invariant set of R_{lat}.

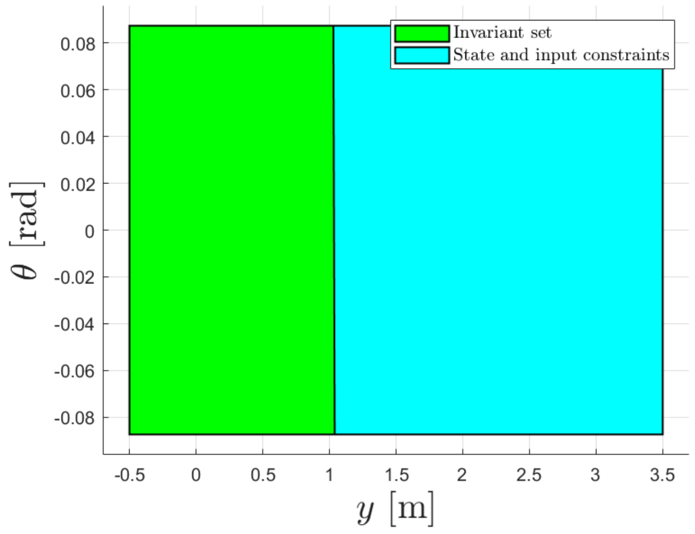


Figure 10: Terminal invariant set with final tuning parameters.

4 Offset-free tracking (Deliverable 4.1)

4.1 Tuning parameters

When including the state and disturbance estimators, the dynamics of the system need to be augmented. Notice that, the disturbance is only acting on the state and not the output ($C_d = 0$). The equation for the longitudinal sub-system becomes:

$$\begin{pmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{pmatrix} = \begin{pmatrix} A_d & \hat{B}_d \\ 0 & I \end{pmatrix} \begin{pmatrix} \hat{x}_k \\ \hat{d}_k \end{pmatrix} + \begin{pmatrix} B_d \\ 0 \end{pmatrix} \hat{u}_k + \begin{pmatrix} \hat{x}_s + A_d \, \hat{x}_s + B_d \, \hat{u}_s \\ 0 \end{pmatrix} + \begin{pmatrix} L_x \\ L_d \end{pmatrix} (C\hat{x}_k - y_k) \tag{16}$$

$$\text{or} \quad \hat{z}_{k+1} = \hat{x}_s + \hat{A}(\hat{z}_k - \hat{x}_s) + \hat{B}(u_k - \hat{u}_s) + L(\hat{C}\hat{z}_k - y_k) \tag{17}$$

where $\hat{x}_k$ is the current state estimate (V), $\hat{d}_k$ is the current disturbance estimate, $\hat{z}_k$ is the vector containing $\hat{x}_k$ and $\hat{d}_k$, $\hat{x}_s$ is the linearized speed, $\hat{u}_s$ is the control input u_T around which the system is linearized, L is the observer gain matrix and y is the measurement (the speed in this case).

By verification the observability matrix $\mathcal{O}$ constructed from A and C has full rank. Additionally, the 2×2 matrix $\mathcal{P}$ shown below also has full rank. Thus the two necessary conditions for an observable system are met. This means that the augmented system described in equation 16 is observable.

$$\mathcal{P} = \begin{pmatrix} A_d - I & B_d \\ C & 0 \end{pmatrix} \tag{18}$$

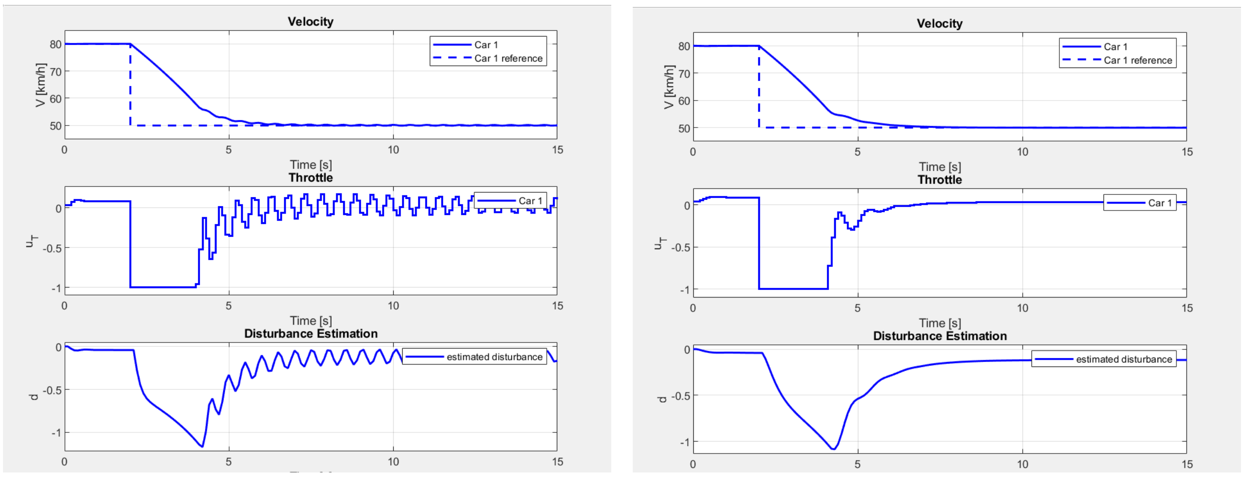
4.1.1 Pole placement

The observer gain matrix L can be tuned. The pole placement method was used to find the components of L. As the system is discretized, the poles need to be located inside the unit circle. Poles closer to the origin will help the system to converge faster. As seen in Figure 11, the augmented system requires slower poles to maintain stability. Faster poles will tend to create an oscillatory response if the disturbance d is not observable fast enough.

4.1.2 Other tuning parameters

As discussed in §3.2.2, the state and control cost matrices Q_{lon} and R_{lon} can be tuned. Increasing Q_{lon} will decrease the settling time but in exchange, the comfort of the passengers will be negatively impacted. This effect is even more visible now that state and disturbance estimators are introduced. Taking larger values of R_{lon} will help stabilize the system but will inevitably increase the settling time. Each final tunable parameter is listed under this paragraph:

- Pole placement (p1, p2) = (0.6, 0.7)



(a) Fast poles (0.3, 0.4).

(b) Slow poles (0.6, 0.7).

Figure 11: Impact of pole placement on the general stability ($R_{lon} = 1$, $Q_{lon} = 1$).

- Longitudinal state cost matrix $Q_{lon} = 1$
- Longitudinal control cost weight $R_{lon} = 1$
- Resulting settling time for speed with lane change (80 to 50 km/h) $T_s = 6.8$ s ($< 15\, s$)

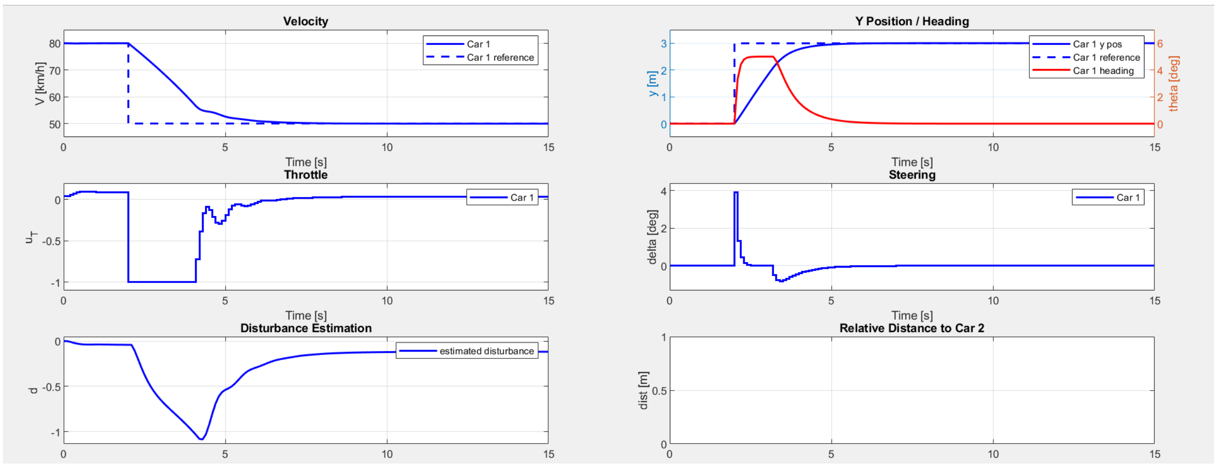


Figure 12: Closed-loop response of the car after adding offset-free tracking.

5 Robust tube MPC (Deliverable 5.1)

In order to compute the minimal invariant set $\mathcal{E}$, first the $\mathbb{W}$ polyhedron is defined, it represents the disturbance set of the lead car:

$$M\,\tilde{u}_T \leq m \quad \text{where} \quad M = (1, -1)^T, \quad m = (u_{T,s} + 0.5, u_{T,s} - 0.5)^T, \quad \tilde{u}_T \in \mathbb{W} \qquad (19)$$

The closed-loop dynamic matrix is defined as $A_{cl} = A_d$ - $B_d K$, where A_d and B_d are the discretized dynamic matrices and K is the control matrix found with LQR. The minimal invariant set $\mathcal{E}$ can be found using the algorithm from the lecture. Note that the input matrix B_d multiplies the disturbance $\tilde{u}_T$ in the $\boldsymbol{\Delta}$ equation 20 and thus multiplies $\tilde{u}_T$ in the algorithm.

$$\boldsymbol{\Delta}^+ = A_d\,\boldsymbol{\Delta} - B_d\,u_T + B_d\,\tilde{u}_T \tag{20}$$

To compute the tightened constraints $\tilde{\mathbb{X}}$ and $\tilde{\mathbb{U}}$, the Pontryagin difference is used:

$$\tilde{\mathbf{X}} = \mathbf{X} \ominus \mathcal{E} \quad \text{and} \quad \tilde{\mathbf{U}} = \mathbf{U} \ominus \mathcal{K}\mathcal{E} \tag{21}$$

The constraints must also be modified and they become:

$$\text{States:} \quad F_{\mathcal{E}}\,(\Delta_0 - z_0) \leq f_{\mathcal{E}}\,, \quad z_{k+1} = A\,z_k + B\,v_k\,, \quad F_{\tilde{X}}z \leq f_{\tilde{X}}\,, \quad F_{\mathcal{X}_f} z_N \leq f_{\mathcal{X}_f} \tag{22}$$

where, F and f are the inequality constraints matrices of their respective set, Δ_0 is the initial value of Δ, and z is the nominal state trajectory.

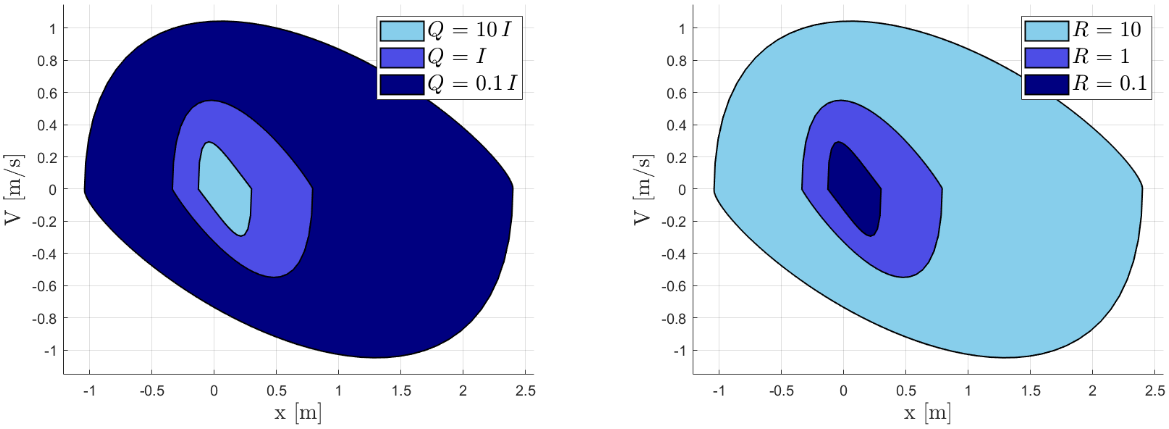
$$\text{Control input:} \quad M_{\tilde{U}}\,v \leq m_{\tilde{U}} \qquad \mu_{\text{tube}} = K(x_0 - z_0) + v_0 \tag{23}$$

where M and m are the inequality constraints matrices of their respective set, u is the nominal control input for z, and μ_{tube} is the tube MPC control law.

5.1 Tuning parameters

5.1.1 State and control cost matrices Q and R

Similarly to previous sections, Q and R appear in the objective function and helps compute the controller matrix K of the LQR controller used for the longitudinal sub-controller. They have an impact on the shape of the minimal invariant set $\mathcal{E}$.



(a) $\mathcal{E}$ with various Q values ($R = 1$).

(b) $\mathcal{E}$ with various R values ($Q = I$).

Figure 13: Impact of state and control cost parameters Q and R on the minimal robust invariant set $\mathcal{E}$.

As observed earlier, increasing R results in the car exhibiting a more sluggish behavior. This occurs because the MPC controller places a higher penalty on u_T when it deviates significantly from u_{ref}, making the car respond more slowly. Consequently, $\mathcal{E}$ expands, as the car needs to handle a wider range of scenarios.

Unlike previously, tuning the weight associated with the x coordinate is particularly important in this context, as it ensures the ego vehicle maintains a safe distance from the lead vehicle, denoted by x_{safe}. By increasing the weight on the x coordinate, the system places greater emphasis on preserving this safety distance, ensuring that the ego vehicle does not fall below the x_{safe} threshold. Increasing the weight associated with the speed is also beneficial, as it encourages the vehicle to more effectively reach the reference speed, while the distance between the two cars is maintained through the weight on the x coordinate. However, excessively increasing these weights can significantly reduce the range of feasible solutions, as $\mathcal{E}$ becomes more restrictive.

5.1.2 Minimal safe horizontal distance between two cars x_{safe}

When tuning x_{safe}, it is important not to decrease it too much, as this could shift $\tilde{\mathbb{X}}$ too far to the right, causing the origin to fall outside of $\mathcal{X}_f$. In such a case, $\mathcal{X}_f$ would become empty.

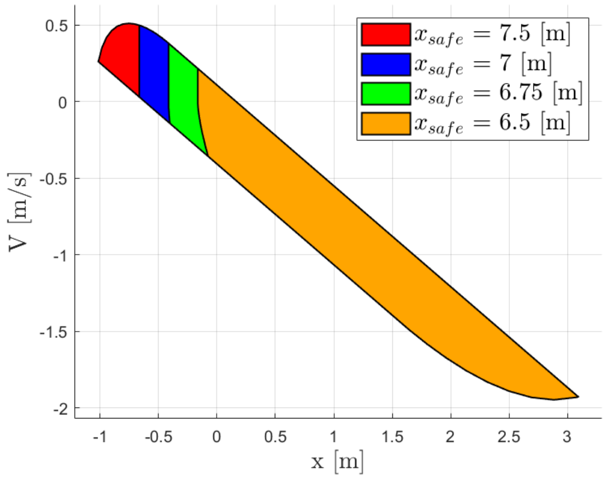


Figure 14: Impact of x_{safe} on the terminal set $\mathcal{X}_f$ ($Q = I$, $R = 1$). The origin would leave $\mathcal{X}_f$ if $x_{\text{safe}} \leq 6.3359$ m with the current controller K. Increasing x_{safe} beyond 7.5 m would not change the shape of $\mathcal{X}_f$ (beyond 7.3508 m to be exact).

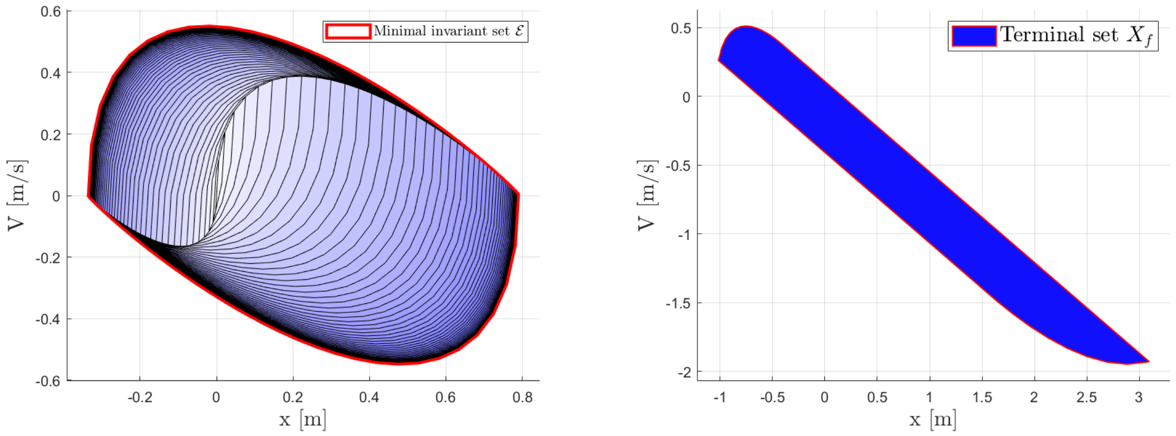
On the other hand, increasing x_{safe} too much will give an unfeasible problem to the controller (see Figure 14). This value is around 30.85 m ($Q = I$, $R = 1$ and $H = 10$ s) if the lead car starts 15 meters in front of the ego car with a speed of 100 km/h while the ego car has a speed of 120 km/h. The problem becomes unfeasible if the Nth step of the horizon does not respect the x_{safe} distance.

To compute the tightened constraints $\tilde{\mathbb{X}}$ and $\tilde{\mathbb{U}}$, the Pontryagin difference is used with $\mathcal{E}$:

$$\tilde{\mathbb{X}} : \Delta_x \geq 6.3359 - x_{safe} \quad \text{and} \quad \tilde{\mathbb{U}} : -0.5490 \leq u_T \leq 0.1436$$

The value of x_{safe} should be strictly bigger than 6.3359 m in order for $\tilde{\mathbb{X}}$ to contain the origin and for $\mathcal{X}_f$ to be non-empty ($Q = I$ and $R = 1$).

5.2 Results



(a) Minimal robust invariant set $\mathcal{E}$.

(b) Terminal invariant set $\mathcal{X}_f$.

Figure 15: Final minimal robust invariant set $\mathcal{E}$ and terminal set $\mathcal{X}_f$ with final tuning parameters.

Final tuning parameters:

$$H = 10\,s \quad \text{(unchanged)}, \quad Q = I, \quad R = 1 \quad \text{and} \quad x_{\text{safe}} = 10\,m$$

5.2.1 Case 1

In this first situation, the simulation starts with the following conditions: 1) the ego car moves at 100 km/h and would like to speed up at 120 km/h, 2) the lead car starts 15 m in front of it and moves at a steady-state speed of 100 km/h. The ego car is expected to speed up (or slow down) until the distance between the two car reaches x_{safe}.

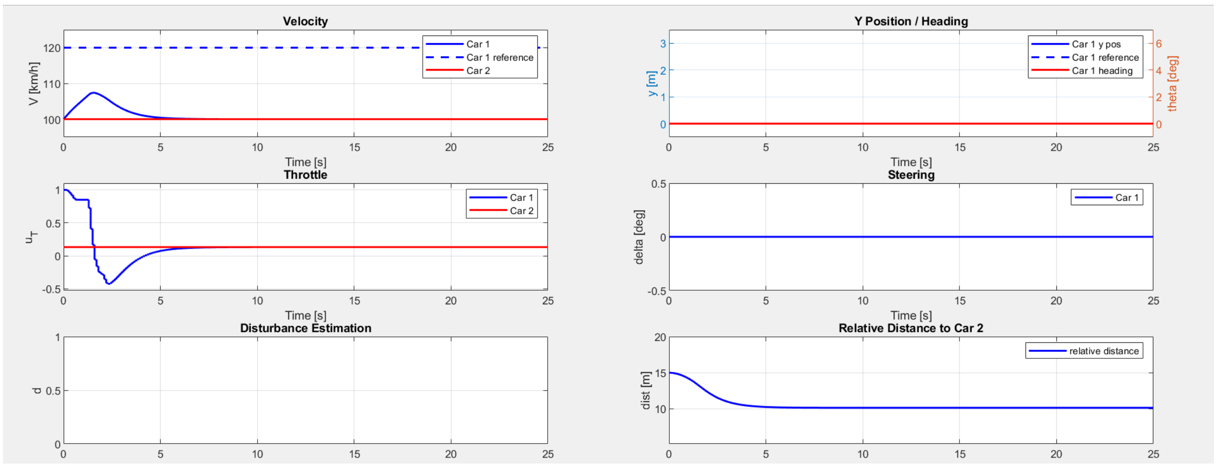


Figure 16: Closed-loop response of the robust tube MPC controller. We notice how the relative distance to the lead car stays at 10 m.

5.2.2 Case 2

In this second situation, the simulation starts with the following conditions: 1) the ego car moves at 115 km/h and would like to speed up at 120 km/h, 2) the lead car starts 8 m in front of it and moves at 120 km/h. The lead car then suddenly applies the brake ($u_T \approx -0.3$) at $t = 7.5$ s and finally applies the throttle again ($u_T \approx 0.7$) at $t = 15$ s.

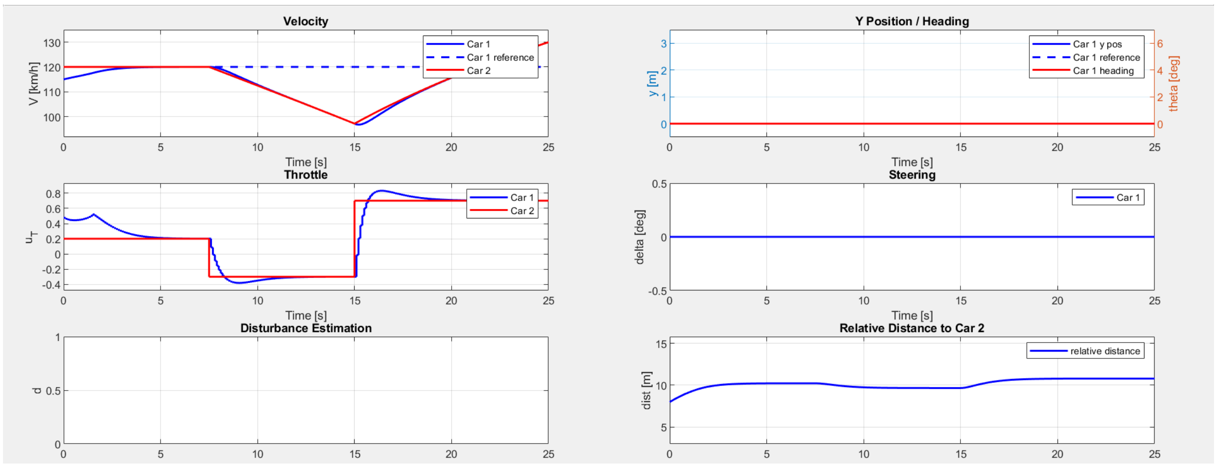


Figure 17: Closed-loop response of the robust tube MPC controller.

Interestingly, the relative distance to the lead car does not converge exactly to the x_{safe} value. This is because the ego car accumulates delay caused by the instantaneous dynamics of the lead car, and is unable to reach perfectly x_{safe} with the current controller. The controller determines that getting closer to x_{safe} is more costly than staying at a value close to it. As a result, it will have a relative distance to lead car equal to x_{safe} plus the accumulated delay. Choosing a more aggressive controller would encourage the controller to get closer to x_{safe} but in exchange, the ride quality would be negatively affected.

6 Nonlinear MPC

6.1 Deliverable 6.1

For non-linear MPC control, the system is not linearized around an arbitrary value anymore. The path of the future horizon steps are integrated over time using the Runge-Kutta 4 (RK4) integration algorithm. This method allows for better precision compared to using a first-order Taylor series approximation in exchange of more computation time.

This translates to a new constraint in the NMPC controller compared to linear MPC. The following constraints need to be satisfied for every step of the horizon:

$$x_{k+1} = x_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{24}$$

where, x_{k+1} is the next state step in the horizon, h is the sampling time and k_i are the intermediary slopes (computed from the f function in equation (2) in the project description).

Since the system is not divided into longitudinal and lateral subsystems anymore, the tuning parameters are combined. The sampling time h, the state cost matrix Q and the control cost matrix R become:

$$h = 0.1 \text{ s}, \quad Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

When using NMPC, steady-state tracking error is not expected as the RK4 algorithm provides greater accuracy when computing terms that are far from the reference point, compared to linearizing the system around a fixed reference speed. One key advantage of NMPC over linear MPC is that its reference point is dynamic and evolves over time, allowing the controller to adapt as the system progresses.

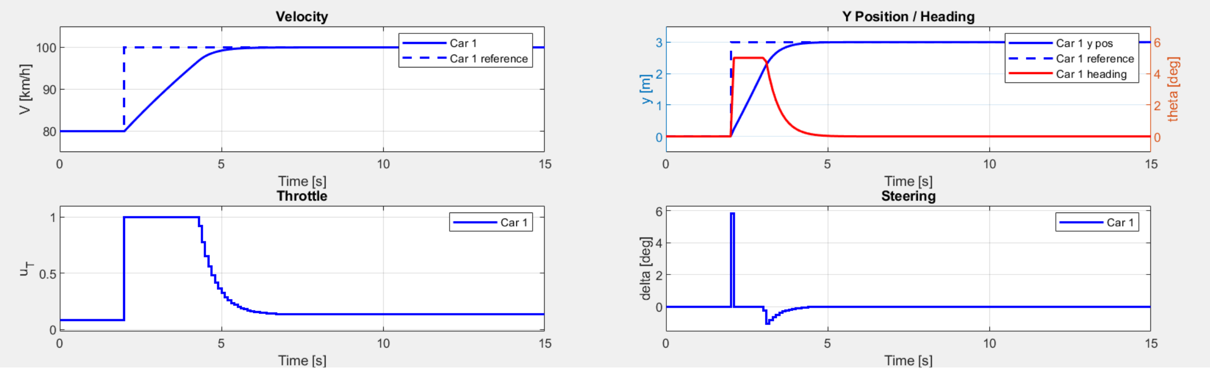


Figure 18: Closed-loop response of the car after adding the non-linear MPC controller.

With these parameters, the constraints are all satisfied:

- Resulting settling time for speed with lane change (80 to 100 km/h) $T_s = 2.9$ s ($< 5\,s$)
- Resulting settling time for y (lane change) $T_s = 2.4$ s ($< 5\,s$)

6.2 Deliverable 6.2

The first step in designing our NMPC control for overtaking is to define a new constraint to avoid collision. Like suggested, an ellipsoidal constraint is defined as:

$$(p - p_L)^T H (p - p_L) \geq 1 \tag{25}$$

where p is the vector containing the x and y coordinates of the ego car, p_L represents the same for the leading car, and H is a tunable matrix defined by:

$$H = \begin{pmatrix} \frac{1}{a^2} & 0 \\ 0 & \frac{1}{b^2} \end{pmatrix} \tag{26}$$

where a and b are the semi-major and semi-minor axis respectively of the ellipse. The values of a and b can be modified to respectively prevent the car from changing lanes too close to the leading car, and to ensure a safe lateral distance with the lead car.

The other usual tuning parameters (Q and R) must also be modified when taking into account this new ellipsoidal constraint. It is important to assign a weight of zero to $Q_{(x,x)}$, otherwise the optimizer will penalize the movement along x. The length of the horizon is also decreased from 10 s to 5 s in order to reduce computation time. this does not modify the precision. The final tuning parameters are listed below:

- Horizon length $H = 5$ s
- state cost matrix $Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$
- control input cost matrix $R = I$
- semi-major axis $a = 15$ m
- semi-minor axis $b = 3$ m

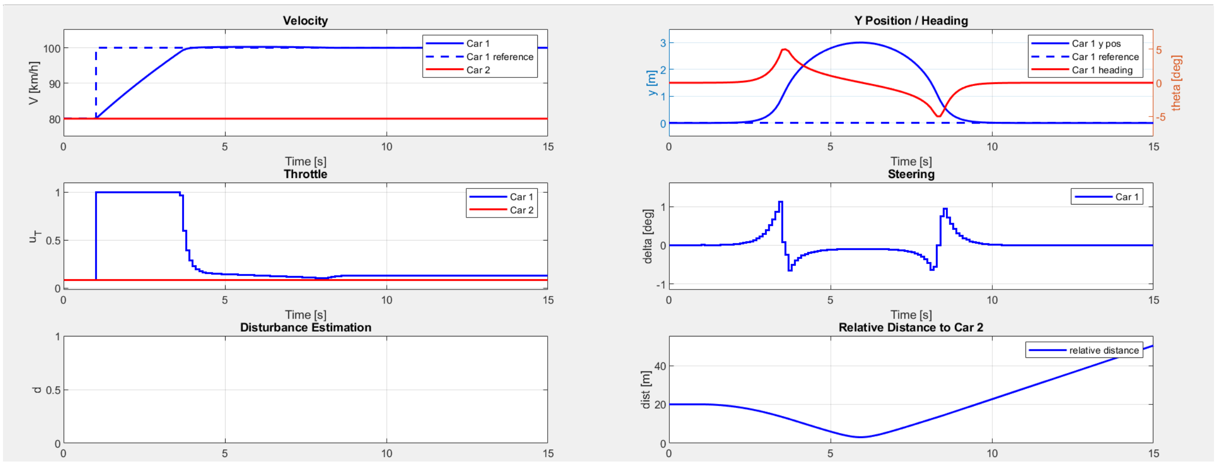


Figure 19: Closed-loop response of the car after adding the non-linear anti-collision MPC controller.

7 Conclusion

This project implemented MPC strategies to manage the longitudinal and lateral dynamics of a car on a highway. Linear and nonlinear MPC approaches were implemented and ensure a smooth, responsive, and safe driving behavior. Key parameters, such as cost matrices and horizon length, were tuned to optimize performance. Additionally, the integration of an observer ensured accurate state estimation and correctly rejects the disturbance introduced by the linearization of the MPC controller, further enhancing the controller's reliability. Robust tube MPC improved disturbance handling in a more complex scenario: caused by a second vehicle. Finally, nonlinear MPC was implemented as an alternative to linear MPC where the high precision of the controller allowed it to handle complex scenarios such as overtaking. The resulting controllers adhered to safety and comfort constraints, demonstrating the effectiveness of MPC in automotive applications.

8 Appendix

Finding A:

$$A = \begin{bmatrix} 0 & 0 & -Vsin(\theta+\beta) & cos(\theta+\beta) \\ 0 & 0 & Vcos(\theta+\beta) & sin(\theta+\beta) \\ 0 & 0 & 0 & \frac{sin(\beta)}{l_r} \\ 0 & 0 & 0 & \frac{u_T P_{max}}{V^2} - \rho C_d A f V \end{bmatrix} \Bigg|_{(\boldsymbol{x}_s, \boldsymbol{u}_s)}$$

$$A = \begin{bmatrix} 0 & 0 & -Vsin(\theta)cos(\beta) - Vsin(\beta)\cos(\theta) & cos(\theta)cos(\beta) - sin(\theta)sin(\beta)) \\ 0 & 0 & Vcos(\theta)cos(\beta) - Vsin(\beta)sin(\theta) & cos(\theta)sin(\beta) - cos(\beta)sin(\theta)) \\ 0 & 0 & 0 & \frac{sin(\beta)}{l_r} \\ 0 & 0 & 0 & \frac{u_T P_{max}}{V^2} - \rho C_d A f V \end{bmatrix} \Bigg|_{(\boldsymbol{x}_s, \boldsymbol{u}_s)}$$

Finding B:

$$\begin{bmatrix} -\frac{d\beta}{d\delta}Vsin(\theta+\beta) & 0 \\ \frac{d\beta}{d\delta}Vcos(\theta+\beta) & 0 \\ \frac{d\beta}{d\delta}\frac{Vcos(\beta)}{lr} & 0 \\ 0 & \frac{P_{max}}{V} \end{bmatrix}\Bigg|_{(\boldsymbol{x}_s,\boldsymbol{u}_s)} = \begin{bmatrix} -\frac{d\beta}{d\delta}(Vcos(\theta)sin(\beta)+Vcos(\beta)sin(\theta)) & 0 \\ \frac{d\beta}{d\delta}(Vcos(\theta)cos(\beta)-Vsin(\beta)sin(\theta)) & 0 \\ \frac{d\beta}{d\delta}\frac{Vcos(\beta)}{lr} & 0 \\ 0 & \frac{P_{max}}{V} \end{bmatrix}\Bigg|_{(\boldsymbol{x}_s,\boldsymbol{u}_s)}$$

where:

$$\frac{d\beta}{d\delta} = \frac{1}{1+\frac{l_r tan(\delta)}{l_r+l_f}^2} * \frac{d(\frac{l_r tan(\delta)}{l_r+l_f})}{d\delta} = \frac{1}{1+\frac{l_r tan(\delta)}{l_r+l_f}^2} * \frac{l_r}{l_r+l_f} * \frac{1}{cos(\delta)^2}$$

thus:

$$\frac{d\beta}{d\delta}|(\boldsymbol{x}_s,\boldsymbol{u}_s) = \frac{l_r}{l_r+l_f}$$