Model predictive control project report

Group BS: Luca Jiménez Glur, Geoffroy Renaut

January 8, 2025

Contents

1 System Dynamics

The goal of this project is to analyze the dynamics of a car using a simplified bicycle model, and implement various MPC controllers for its control. The state vector $\mathbf{x} = \begin{bmatrix} x & y & \theta & V \end{bmatrix}^\intercal$ contains the longitudinal position of the car x in meters, the lateral position of the car y in meters, the heading angle of the car θ in radians and the velocity of the car V in meters per second. The dynamics of the system being studied are given by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} V\cos(\theta+\beta) \\ V\sin(\theta+\beta) \\ \frac{V}{l_r}\sin(\beta) \\ \frac{F_{\text{motor}} - F_{\text{drag}} - F_{\text{roll}}}{m} \end{bmatrix} \quad , \quad \beta = \arctan(\frac{l_r \tan(\delta)}{l_r + l_f}) \quad , \quad \begin{cases} F_{\text{motor}} &= \frac{u_T P_{\max}}{V} \quad , \\ F_{\text{drag}} &= \frac{1}{2}\rho C_d A_f V^2 \quad , \\ F_{\text{roll}} &= C_r m g \quad . \end{cases}$$

The dynamics are controlled by two inputs, steering δ in radians and throttle u_T (no unit), which are contained in the input vector $\mathbf{u} = \begin{bmatrix} \delta & u_t \end{bmatrix}^\intercal$.

2 Linearization

2.1 Deliverable 2.1

We want to linearize the dynamics such that $f(\mathbf{x}, \mathbf{u}) \approx f_s(\mathbf{x}_s, \mathbf{u}_s) + \mathbf{A}(\mathbf{x} - \mathbf{x}_s) + \mathbf{B}(\mathbf{u} - \mathbf{u}_s)$, around the steady state $(\mathbf{x_s}, \mathbf{u_s}) = (0, 0, 0, V_s, 0, u_{T,s})$, where $\mathbf{A}$ and $\mathbf{B}$ are the jacobian matrices of f with respect to $\mathbf{x}$ and $\mathbf{u}$ in that order. Here, we show the steps followed to derive $\mathbf{A}$ and $\mathbf{B}$ analytically.

$$\beta|_{(\mathbf{x_s}, \mathbf{u_s})} = \arctan(\frac{l_r \tan(0)}{l_r + l_f}) = 0$$

$$\mathbf{A} = \left.\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\right|_{(\mathbf{x}_s, \mathbf{u}_s)} = \left.\begin{bmatrix} 0 & 0 & -V\sin(\theta+\beta) & \cos(\theta+\beta) \\ 0 & 0 & V\cos(\theta+\beta) & \sin(\theta+\beta) \\ 0 & 0 & 0 & \frac{\sin(\beta)}{l_r} \\ 0 & 0 & 0 & \frac{-\rho C_d A_f V - \frac{P_{\max} u_T}{V^2}}{m} \end{bmatrix}\right|_{(\mathbf{x}_s, \mathbf{u}_s)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & V_s & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-\rho C_d A_f V_s - \frac{P_{\max} u_{T,s}}{V_s^2}}{m} \end{bmatrix} \tag{1}$$

$$\left.\frac{\partial \beta}{\partial \delta}\right|_{\delta=0} = \left.\frac{1}{\frac{l_r + l_f}{l_r}\cos(\delta)^2 + \frac{l_r}{l_r + l_f}\sin(\delta)^2}\right|_{\delta=0} = \frac{l_r}{l_r + l_f}$$

$$\mathbf{B} = \left.\frac{\partial f(\mathbf{x}, \mathbf{u}))}{\partial \mathbf{u}}\right|_{(\mathbf{x}_s, \mathbf{u}_s)} = \left.\begin{bmatrix} -V\sin(\theta+\beta)\frac{\partial\beta}{\partial\delta} & 0 \\ V\cos(\theta+\beta)\frac{\partial\beta}{\partial\delta} & 0 \\ \frac{V}{l_r}\cos(\beta)\frac{\partial\beta}{\partial\delta} & 0 \\ 0 & \frac{P_{\max}}{mV} \end{bmatrix}\right|_{(\mathbf{x}_s, \mathbf{u}_s)} = \begin{bmatrix} 0 & 0 \\ \frac{l_r V_s}{l_f + l_r} & 0 \\ \frac{V_s}{l_f + l_r} & 0 \\ 0 & \frac{P_{\max}}{mV_s} \end{bmatrix} \tag{2}$$

2.2 Deliverable 2.2

The system is linearized around the quasi-steady-stade $(\mathbf{x_s}, \mathbf{u_s}) = (0, 0, 0, V_s, 0, u_{T,s})$. In this state, the longitudinal (x) behavior of the system is independent of the lateral (y) behavior. In fact, according to the computed jacobian, $\dot{x}$ depends only on V, and $\dot{V}$ depends only on V and u. Similarly, $\dot{y}$ depends only on θ and δ; and $\dot{\theta}$ depends only on δ. We can thus look at two separate subsystems: one enclosing $\dot{x}$ and $\dot{V}$, the other enclosing $\dot{y}$ and $\dot{\theta}$. The separation of systems can be visualized as follows:

$$\mathbf{A}(\mathbf{x}-\mathbf{x}_s)+\mathbf{B}(\mathbf{u}-\mathbf{u}_s)=\begin{bmatrix}0 & 0 & 0 & 1\\ 0 & 0 & V_s & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{-\rho C_d A_f V_s-\frac{P_{\max}u_{T,s}}{V_s^2}}{m}\end{bmatrix}\begin{bmatrix}x\\ y\\ \theta\\ V-V_s\end{bmatrix}+\begin{bmatrix}0 & 0\\ \frac{l_r V_s}{l_f+l_r} & 0\\ \frac{V_s}{l_f+l_r} & 0\\ 0 & \frac{P_{\max}}{mV_s}\end{bmatrix}\begin{bmatrix}\delta\\ u_T-u_{T,s}\end{bmatrix}$$

We observe that the blue group and the red group do not share cross-terms. Furthermore, all the terms that are not blue or red are 0, thus they can be ignored. The two independent subsystems are:

$$\begin{bmatrix}\dot{x}\\ \dot{V}\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & \frac{-\rho C_d A_f V_s-\frac{P_{\max}u_{T,s}}{V_s^2}}{m}\end{bmatrix}\begin{bmatrix}x\\ V\end{bmatrix}+\begin{bmatrix}0\\ \frac{P_{\max}}{mV_s}\end{bmatrix}u \tag{3}$$

$$\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}0 & V_s\\ 0 & 0\end{bmatrix}\begin{bmatrix}y\\ \theta\end{bmatrix}+\begin{bmatrix}\frac{l_r V_s}{l_f+l_r}\\ \frac{V_s}{l_f+l_r}\end{bmatrix}\delta \tag{4}$$

Intuitively, on a car, when using the steering wheel at a small angle, one would not influence the speed in the x direction and so would the initial angle of the car if it is small enough. But it would change the speed in the y direction and the angular speed of the vehicle.

On the other hand, when changing the throttle, the speed in the x direction would significantly change and so would the acceleration, but the position in the y direction should stay the same.

3 Design MPC controllers for each sub-system

3.1 Deliverable 3.1

3.1.1 Explanation of design procedure that ensures recursive constraint satisfaction

The longitudinal system has no constraints for the longitudinal states x and V. The lateral system sets the following constraints for the lateral states y and θ:

$$\begin{aligned}-0.5 \leq & y \leq 3.5 \quad [\text{m}],\\ -0.0873 \leq & \theta \leq 0.0873 \quad [\text{rad}].\end{aligned}$$

To ensure that these constraints are satisfied recursively, we tune the Q and R matrices such that the discrete LQR controller K produces a closed-loop matrix $A_{\text{cl}} = A + BK$ that yields a maximal invariant set in the state-space (y, θ) which remains within the bounds $[-0.5, 0.5] \times [-0.0873, 0.0873]$. The reason for this is that when tracking the lateral position to the minimal reference $y = 0$ m, the car cannot go further down than $y = -0.5$ m (-0.5 m from the reference) without violating the state constraints, and when tracking to the maximal reference $y = 3$, the car cannot go further up than $y = 3.5$ m ($+0.5$ m from the reference) without violating the state constraints. Putting these two values together for the terminal set gives the bounds $[-0.5, 0.5]$ for the y coordinate in the terminal set to ensure that when shifting the terminal set with respect to the y reference, it will remain within the y state constraint set. The heading angle θ always remained very small compared to the maximal and minimal values of $\pm 30°$, and because tracking was performed on y and not θ, the final θ in the prediction horizon should be approaching 0. Therefore, conducting a similar analysis for the variable θ was not essential. For the terminal set to be within the specified bounds, it was sufficient to let the matrices $Q = I_2$ and $R = 1$, so we kept these simple values.
The steady state target is computed by solving the linear system equation :

$$\begin{bmatrix}1-A & -B\\ 1 & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}_{\text{ref,lon}}\\ \mathbf{u}_{\text{ref,lon}}\end{bmatrix}=\begin{bmatrix}-A\cdot x_s - B\cdot u_s + x_s\\ r\end{bmatrix}$$

$$\begin{bmatrix} I_2 - A & -B \\ \begin{bmatrix} 1 & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\text{ref,lat}} \\ \mathbf{u}_{\text{ref,lat}} \end{bmatrix} = \begin{bmatrix} -A \cdot x_s - B \cdot u_s + x_s \\ r \end{bmatrix}$$

where r is the reference speed $MpcControl\_lon$ or position for $MpcControl\_lat$.

3.1.2 Explanation of choice of tuning parameters

Choice of H :
A long horizon improves performance by accounting for future behavior but increases computation. A trade-off is chosen based on system dynamics and computational capability. As we do not need to achieve real-time, a relatively large H is acceptable, here we have chosen

$$H = 15$$

as this is also the simulation duration.
Lateral control :

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad R = 1$$

Longitudinal control :

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad R = 1$$

Q penalizes the state deviation. Diagonal weights are chosen to emphasize critical states (e.g., is high for velocity tracking). R penalizes the control effort to ensure smooth and efficient control inputs. We have chosen the hyper-parameters Q and R to use the most simple values (ones everywhere) because these values do not cause a violation of the constraints to go from 80km/h to 120km/h in less than 10 seconds and changing lane in less than 3 seconds, as it can be seen in the plots of part 3.1.4.

3.1.3 Terminal invariant set

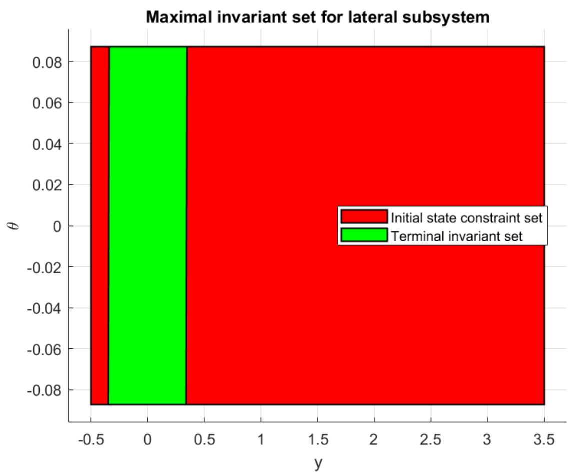


Figure 1: Terminal invariant set for the lateral sub-system

3.1.4 Closed loop plots

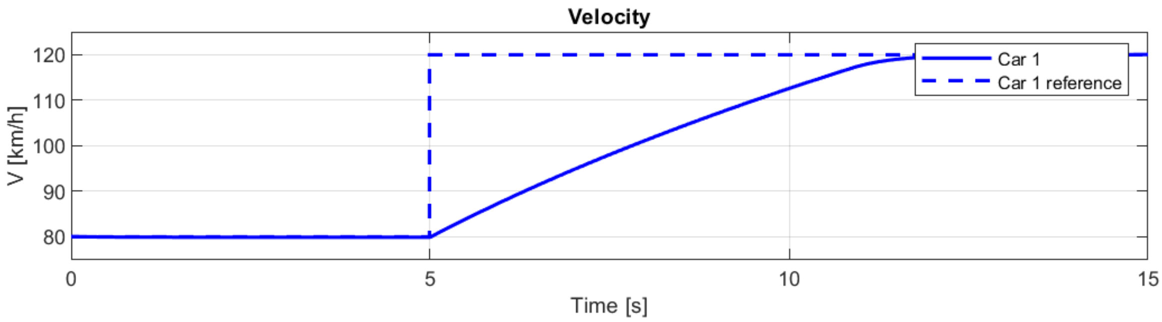


Figure 2: Velocity

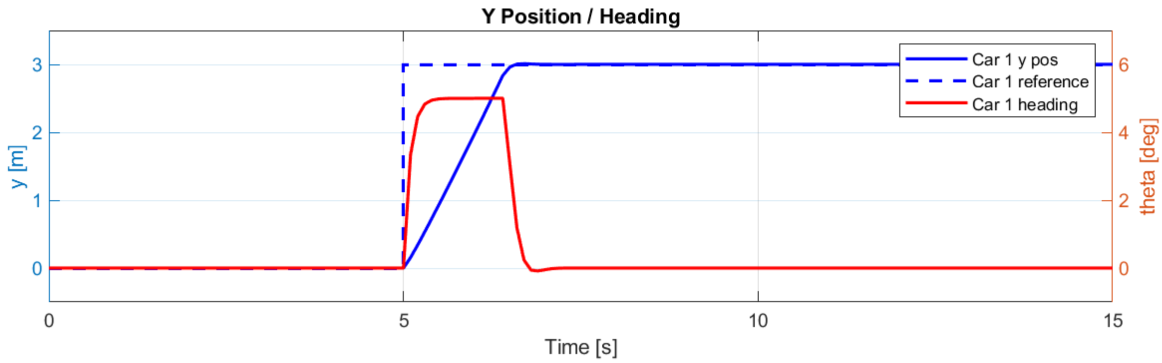


Figure 3: Y-position and θ-orientation

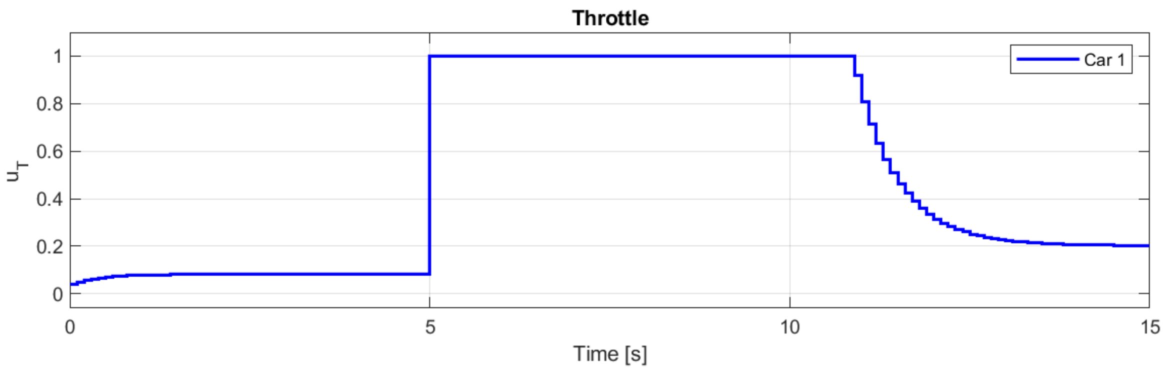


Figure 4: Throttle

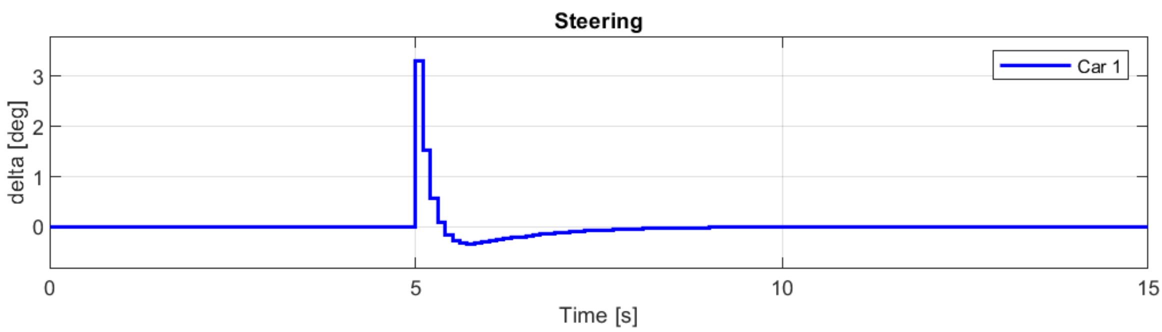


Figure 5: Steering

4 Offset-free tracking

4.1 Deliverable 4.1

4.1.1 Design procedure and choice of parameters

One first estimates the disturbance in *LonEstimator*. In order to do this, a new state vector including the velocity V and the disturbance d as an additional state variable is created:

$$\mathbf{z} = \begin{bmatrix} V \\ d \end{bmatrix}.$$

At each step, the next estimation is computed with :

$$\hat{\mathbf{z}}_{\text{next}} = \hat{\mathbf{x}}_s + \hat{\mathbf{A}}(\hat{\mathbf{z}} - \hat{\mathbf{x}}_s) + \hat{\mathbf{B}}(\mathbf{u} - \hat{\mathbf{u}}_s) + L(\hat{\mathbf{C}}(\hat{\mathbf{z}}) - y)$$

where:

- $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ are augmented matrices derived from system linearization :

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{d(2,2)} & \mathbf{B}_{d(2,1)} \\ 0 & 1 \end{bmatrix}, \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_{d(2,1)} \\ 0 \end{bmatrix}, \hat{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_{d(2,2)} & 0 \end{bmatrix}$$

- $\mathbf{L}$ is the observer gain, chosen via pole placement (`place` function with poles 0.5 and 0.6).

The obtained disturbance estimate $\hat{d}$ is then included in the prediction model of the MPC, note that X and U use the Δ formulation:

$$X_{k+1} = AX_k + BU_k + B\hat{d}.$$

This allows the controller to anticipate the effect of disturbances and adjust the control input accordingly.

The disturbance estimate $\hat{d}$ is explicitly accounted for when calculating the steady-state velocity and input:

$$\begin{bmatrix} (1-A) & -B \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{\text{ref,lon}} \\ u_{\text{ref,lon}} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}x_s - \mathbf{B}u_s + \mathbf{B}\hat{d} + x_s \\ r \end{bmatrix}.$$

This ensures that the reference V_{ref} can be achieved despite disturbances.

4.1.2 Simulation

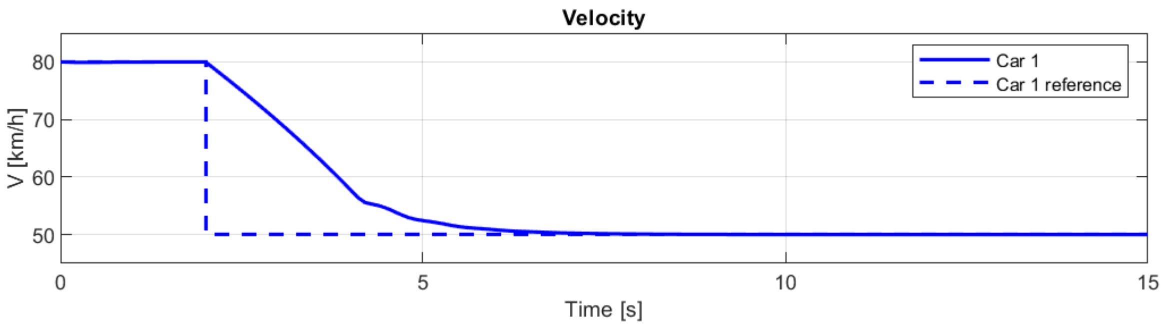


Figure 6: Velocity

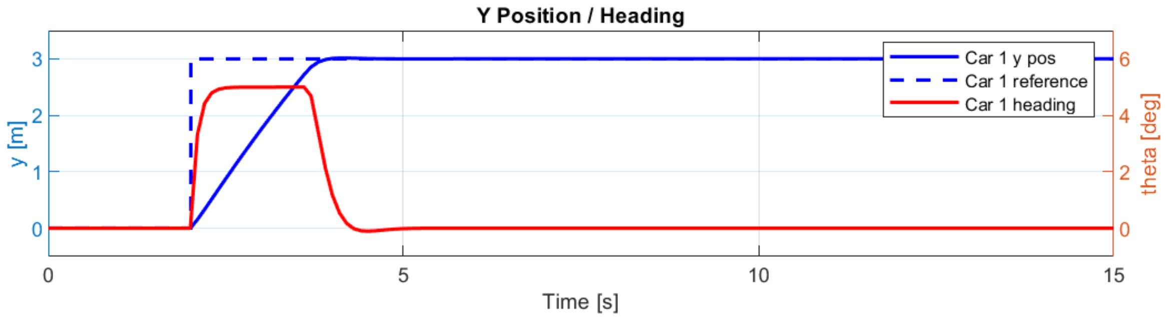


Figure 7: Y-position and θ-orientation

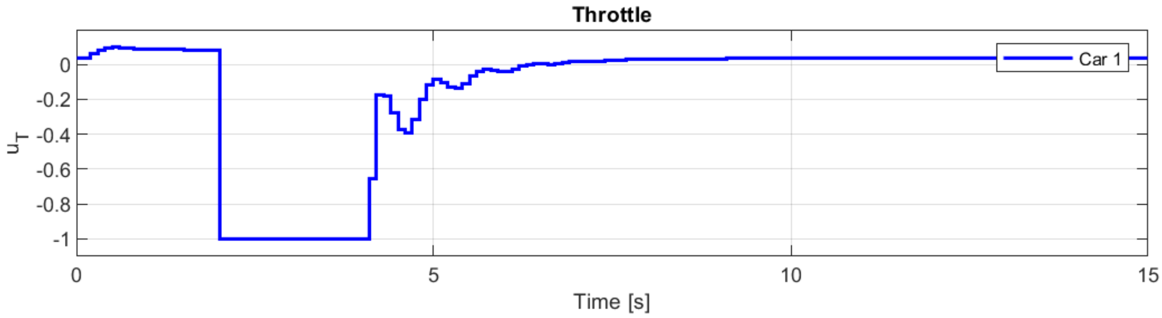


Figure 8: Throttle

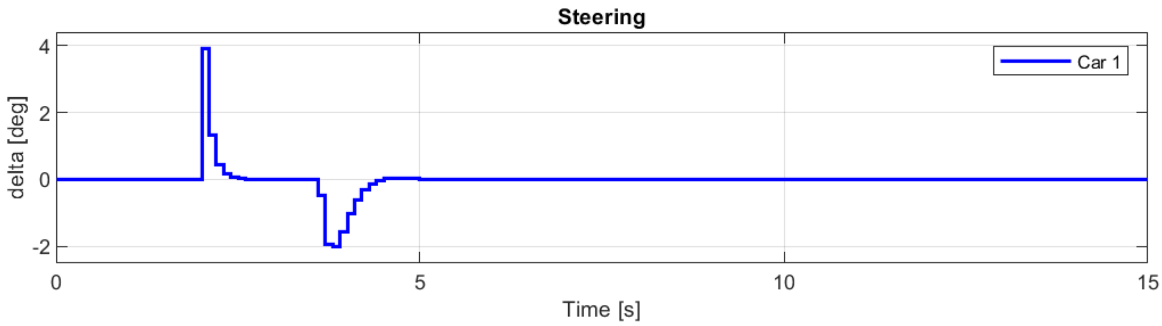


Figure 9: Steering

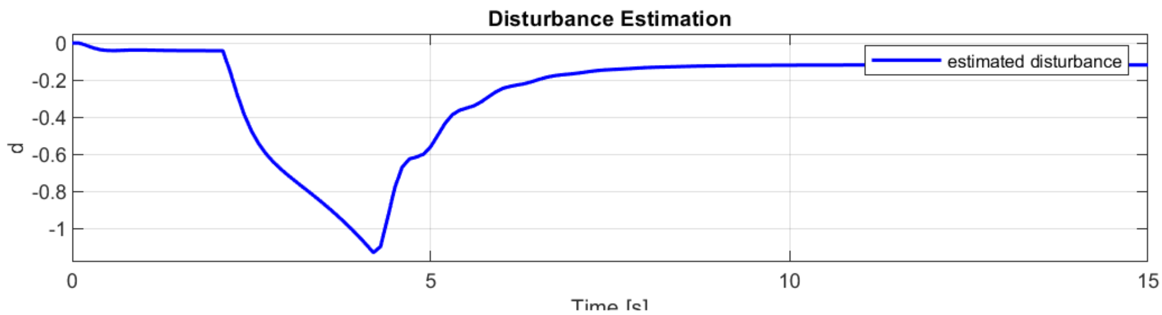


Figure 10: Estimated disturbance

5 Tube MPC

The goal of this section is to implement a robust Tube MPC controller to follow the other car in front of us. This means we assume the control policy $\mu_i(x)$ follows the structure $\mu_i(x) = v_i + K(x - x_i)$. The states we want to minimize are the relative dynamics $\mathbf{\Delta} = \tilde{\mathbf{x}} - \mathbf{x} - \mathbf{x}_{\text{safe}}$, i.e. the difference in longitudinal position and also the difference in velocity between the other car and our car, to which we subtract a safe distance for security reasons.

5.1 Deliverable 5.1

5.1.1 Design procedure, choice of tuning and plots of minimal invariant set and terminal set

The plots of the minimal robust invariant set $\mathcal{E}$ and the maximal robust positive invariant set $\mathcal{X}_f$ computed in `tube_mpc_sets.m` are visible in figures 11 and 12. Practically, $\mathcal{E}$ represents the maximal extent of the influence that the noise on the input $\tilde{u}_T$ will have on the state $\mathbf{\Delta}$ centered around the origin, given the constraints for the noise

$$\tilde{u}_T \in \mathbb{W} = [u_{T,s} - 0.5, u_{T,s} + 0.5]$$

and the closed loop matrix $\mathbf{A}_{\text{cl}} = \mathbf{A}_d - \mathbf{B}_d K$, with K being the discrete LQR controller computed using the discrete matrices $\mathbf{A}_d$ and $-\mathbf{B}_d$ from the longitudinal subsystem, and setting $Q = 10I_2$ and $R = 1$. The maximal robust positive invariant set represents the set inside which the state will be able to remain indefinitely given the *tightened* state and input constraints, $\tilde{\mathbb{X}}$ and $\tilde{\mathbb{U}}$, as well as the controller K and the closed loop matrix A_{cl}. The parameters we had to tune were the matrices Q and R, and also $x_{\text{safe,pos}}$. Through trial and error, we set $x_{\text{safe,pos}} = 6.5$ m, which ensured our car never went closer than 6 m from the other car no matter the accelerations of the other car within the throttle constraints. We set $Q = 10I_2$ and $R = 1$, because it allowed for more reactive dynamics than $Q = I_2$, hence the car in front could be followed more precisely.

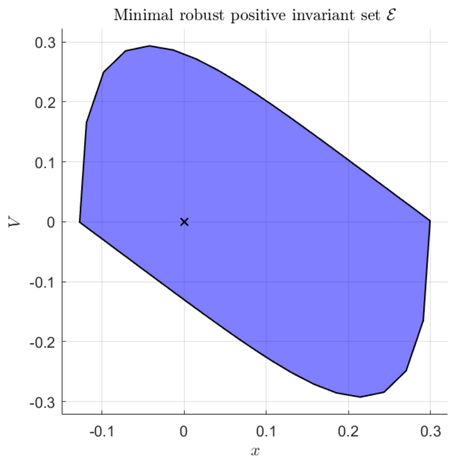


Figure 11: Minimal invariant set $\mathcal{E}$

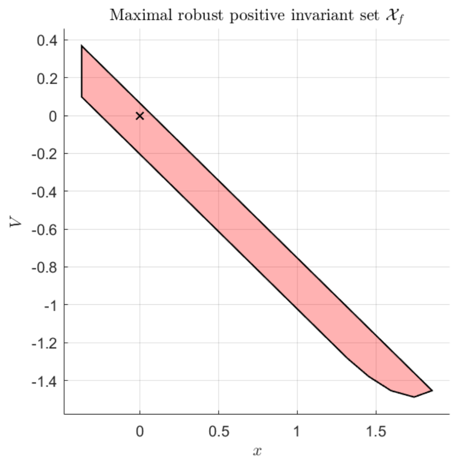
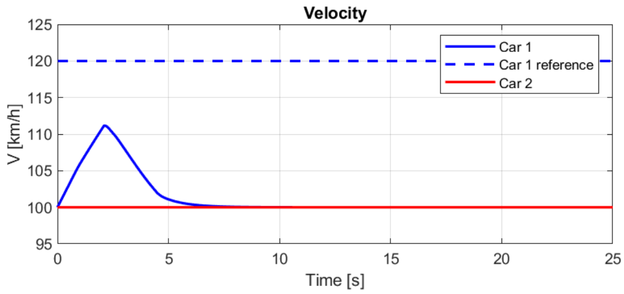


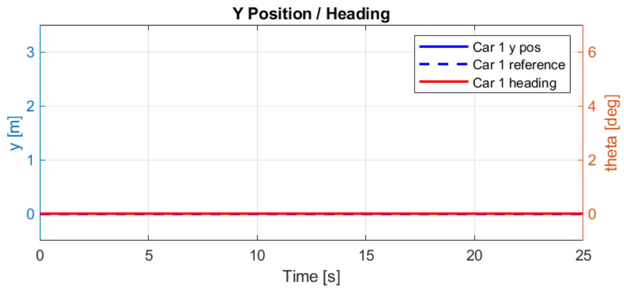
Figure 12: Terminal set $\mathcal{X}_f$

5.1.2 Robust controller test 1

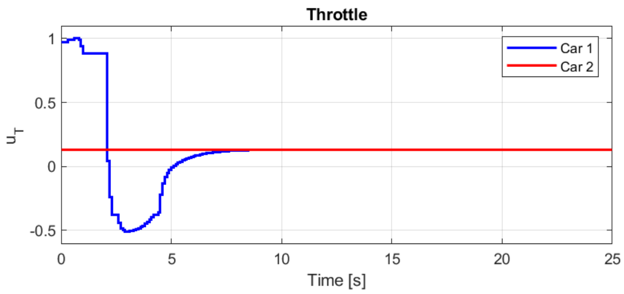
The results of the first test are visible in figure 13.



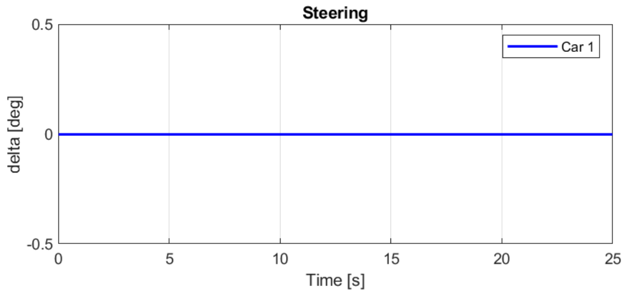
(a) Velocity for Trial 1



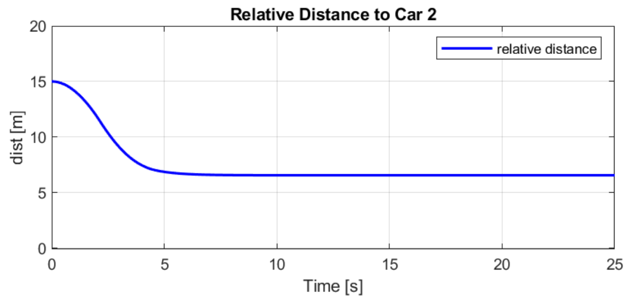
(b) Lateral Position and Heading for Trial 1



(c) Throttle Input for Trial 1



(d) Steering Input for Trial 1

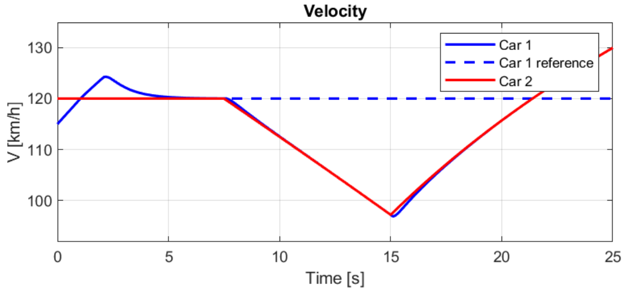


(e) Relative Distance for Trial 1

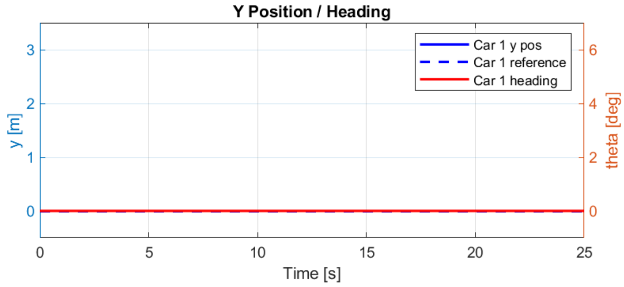
Figure 13: Robust controller working when starting from `myCar.x0 = [0 0 0 100/3.6]'` with `myCar.ref = [0 120/3.6]'` and the other car starting from `otherCar.x0 = [15 0 0 100/3.6]'` with `otherCar.u = car.u const(100/3.6)`.

5.1.3 Robust controller test 2

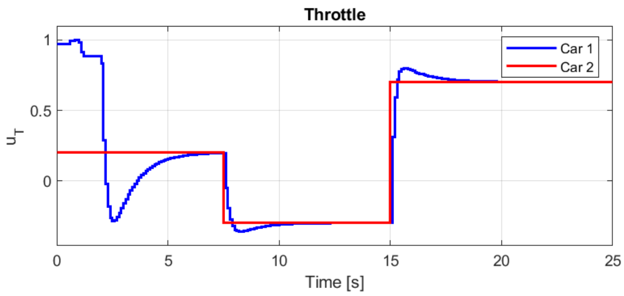
The results of the second test are visible in figure 14.



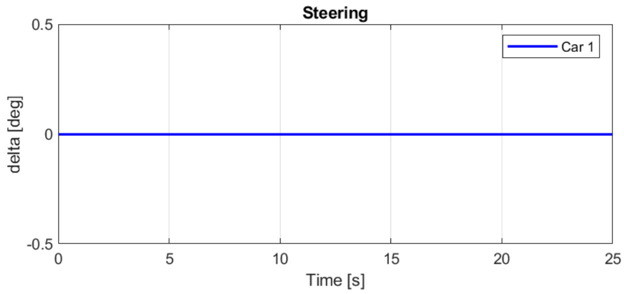
(a) Velocity for Trial 2



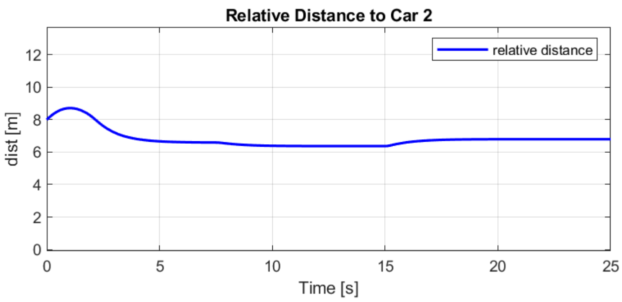
(b) Lateral Position and Heading for Trial 2



(c) Throttle Input for Trial 2



(d) Steering Input for Trial 2



(e) Relative Distance for Trial 2

Figure 14: Robust controller working when starting from `myCar.x0 = [0 0 0 115/3.6]'` with `myCar.ref = [0 120/3.6]'` and the other car starting from `otherCar.x0 = [8 0 0 120/3.6]'` with `otherCar.u = car.u_fwd_ref()` and `otherCar.ref = car.ref_robust()`.

6 Nonlinear MPC

6.1 Deliverable 6.1: Tracking NMPC controller

6.1.1 Design procedure and choice of tuning parameters

1. **Model Discretization**: The continuous-time dynamics of the car are discretized using the Runge-Kutta 4th order (RK4) method to generate a discrete-time system f_{discrete}.

2. **Cost Function**: The cost function is designed to minimize:

   - Speed error ($V_{\text{err}} = v - v_{\text{ref}}$) and lateral error ($y_{\text{err}} = y - y_{\text{ref}}$). This in meant to reach the reference state. A weight of 1 has been used in order to emphasize the importance of getting to the right state in a reduced time.
   - Steering angle (δ), throttle (U_t) and heading angle (θ) to minimize the effort and smooth the trajectory. A weight of 100 has been used for both the angles and a weight of 2 has

been used for the throttle in order to respect the constraint of accelerating to 100 km/h in less than 5 seconds while smoothing the steering and acceleration.

3. **Constraints**:

   - **State Constraints**: Include bounds on position ($-0.5 \leq y \leq 3.5$) and heading angle ($-0.0872 \leq \theta \leq 0.0872$).
   - **Control Constraints**: Throttle ($-1 \leq$ throttle ≤ 1) and steering ($-0.5236 \leq$ steering ≤ 0.5236) are bounded to ensure feasible inputs.

4. **Dynamics**: The system dynamics are enforced via equality constraints: $X_{k+1} = f_{\text{discrete}}(X_k, U_k)$.

6.1.2 Steady-state tracking error

As the steady-state tracking error in part 3 was due to the linearization of the system dynamics, the NMPC controller should not repeat this error because this time the controller uses the 'real' dynamics that are correct at any point and not only where the system has been linearized. This is why the steady state error should be near zero for the NMPC-controller. This behavior is indeed observed in part 6.1.3.

6.1.3 Performance of tracking NMPC controller

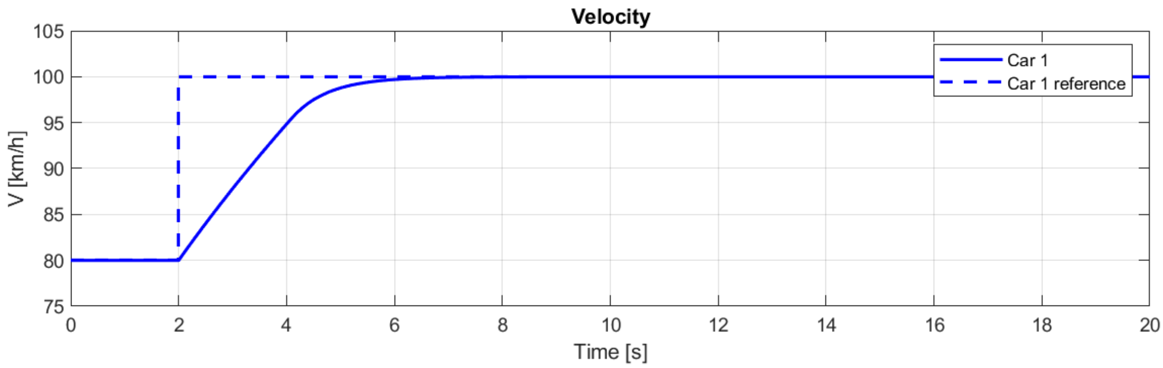


Figure 15: Velocity

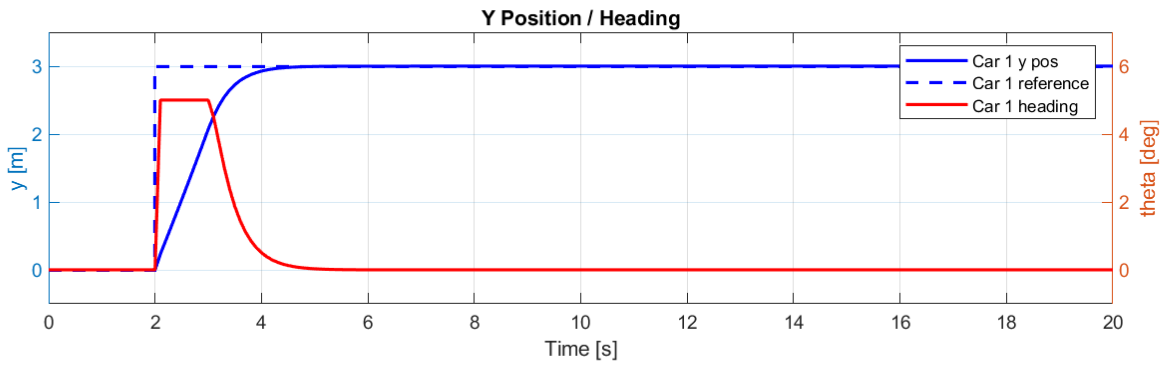


Figure 16: Y-position and θ-orientation

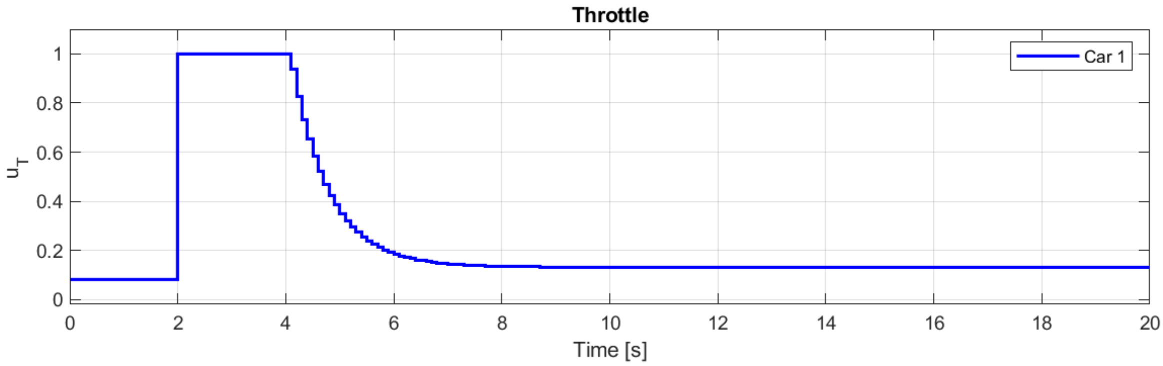


Figure 17: Throttle

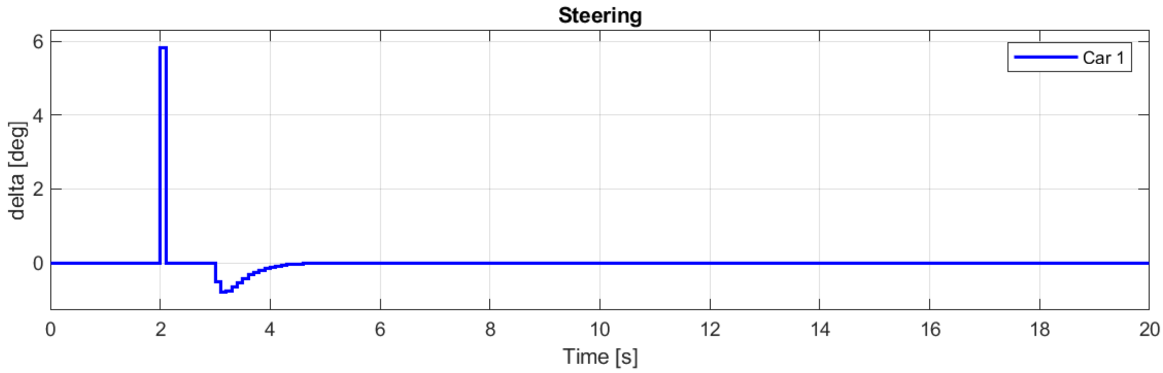


Figure 18: Steering

6.2 Deliverable 6.2: NMPC controller for overtaking

6.2.1 Design procedure and choice of tuning parameters

The controller used in this part is very similar to the one used in part 6.1.3. However, a few changes were made; an ellipsoidal constraint was added, with $p = [X, Y]$ the position of the ego-car and p_{L} the position of the other car such that $(p - p_{\mathrm{L}})^T H (p - p_{\mathrm{L}}) \geq 1$. In order to choose the elements inside the diagonal matrix H, the following calculations were made :
When the two cars are at the same X-position, the ego-car should be at the middle of the second lane.

$$\begin{bmatrix} 0 & 3 \end{bmatrix}^T \ H \begin{bmatrix} 0 & 3 \end{bmatrix} = 1 \Rightarrow 3 \times H_{2,2} \times 3 = 1 \Rightarrow H_{2,2} = 0.11$$

In order to choose the element $H_{1,1}$, a security distance way larger than 4.3m was chosen so that the cars would not come too close from each other.
Also, another change that was made is a rebalance of the weights of the cost, this time, all the weights are set to 10 except the y-error which is set at 1. Those weights have been tuned to avoid oscillatory behaviors that happened if the weights used in part 6.1.1 were kept.

6.2.2 Performance of NMPC controller for overtaking

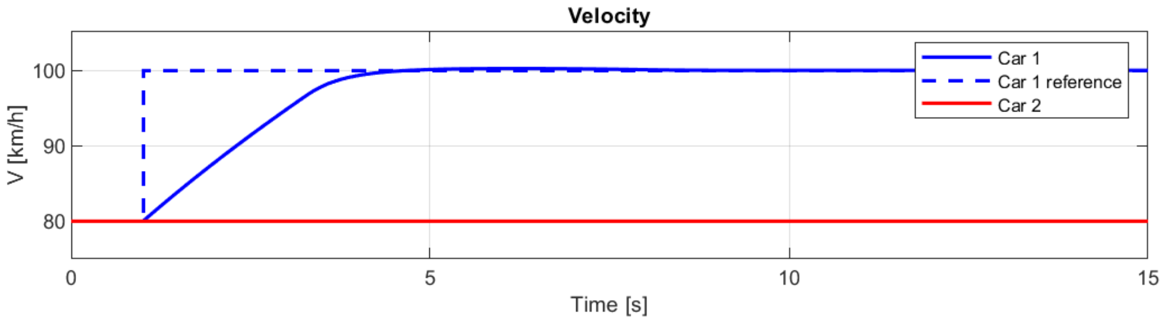


Figure 19: Velocity

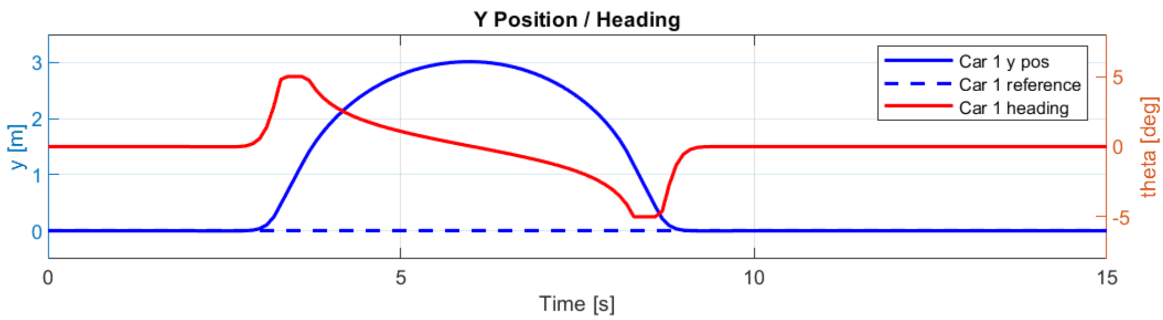


Figure 20: Y-position and θ-orientation

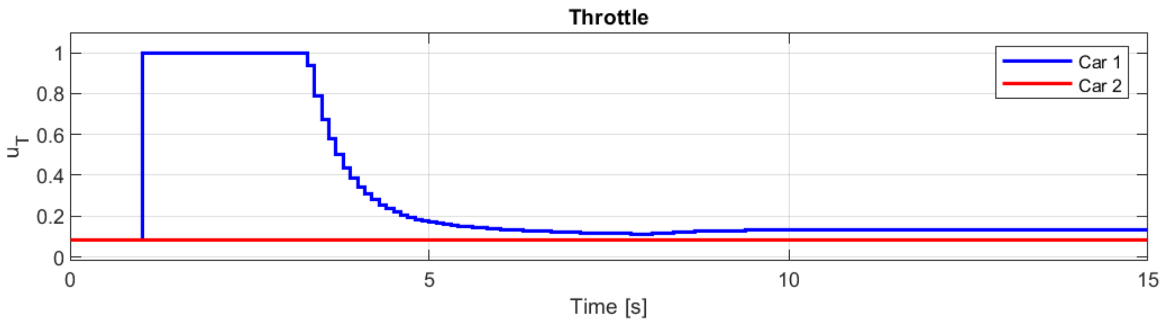


Figure 21: Throttle

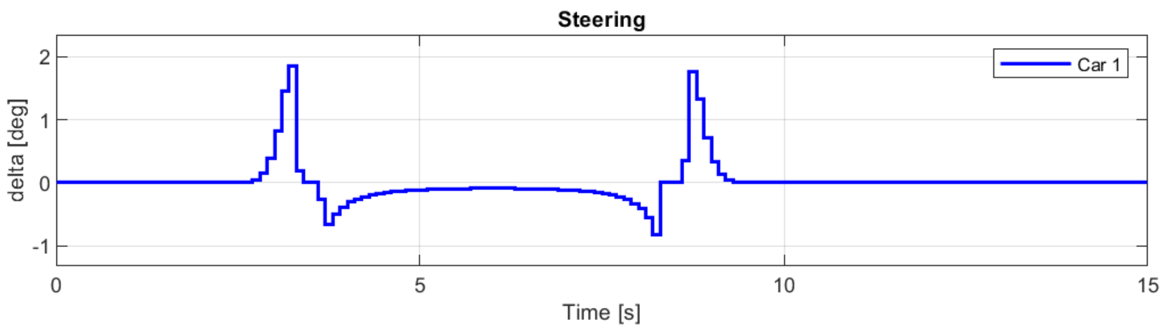


Figure 22: Steering

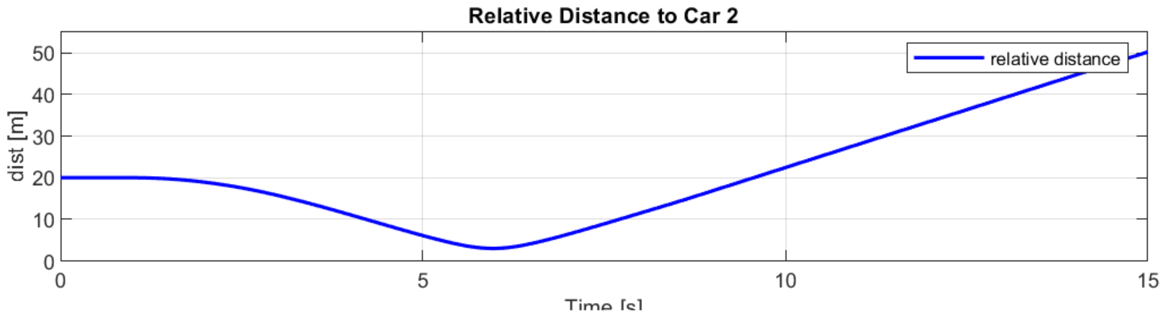


Figure 23: Relative distance of the two cars