Mini-Project - MPC cruise controller for a car on a highway

Report

Imane Jennane - 310900
Liandro Da Silva Monteiro - 352262

Professor: Colin Jones

January 10, 2025

1 Deliverable 2: Linearization

1.1 Deliverable 2.1: Analytical Expression

The dynamics of the system are given by:

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

where:

$$\beta = \arctan\left(\frac{\ell_r \tan(\delta)}{\ell_r+\ell_f}\right),$$

and:

$$F_{\text{motor}} = \frac{u_T P_{\max}}{V}, \quad F_{\text{drag}} = \frac{1}{2}\rho C_d A_f V^2, \quad F_{\text{roll}} = C_r m g.$$

The state vector is:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ V \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \delta \\ u_T \end{bmatrix}.$$

—

Matrix A: Partial Derivatives with Respect to $\mathbf{x}$

The matrix A is defined as:

$$A = \frac{\partial f}{\partial \mathbf{x}}$$

The components are:

$$\frac{\partial \dot{x}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & -V\sin(\theta+\beta) & \cos(\theta+\beta) \end{bmatrix},$$
$$\frac{\partial \dot{y}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & V\cos(\theta+\beta) & \sin(\theta+\beta) \end{bmatrix},$$
$$\frac{\partial \dot{\theta}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & \frac{\sin(\beta)}{\ell_r} \end{bmatrix},$$
$$\frac{\partial \dot{V}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & \frac{-u_T P_{\max}/V^2 - \rho C_d A_f V}{m} \end{bmatrix}.$$

Combining all terms:

$$A = \begin{bmatrix} 0 & 0 & -V\sin(\theta+\beta) & \cos(\theta+\beta) \\ 0 & 0 & V\cos(\theta+\beta) & \sin(\theta+\beta) \\ 0 & 0 & 0 & \frac{\sin(\beta)}{\ell_r} \\ 0 & 0 & 0 & \frac{-u_T P_{\max}/V^2 - \rho C_d A_f V}{m} \end{bmatrix}.$$

—

Matrix B: Partial Derivatives with Respect to $\mathbf{u}$

The matrix B is defined as:

$$B = \frac{\partial f}{\partial \mathbf{u}}$$

The components are:

$$\begin{aligned}
\frac{\partial \dot{x}}{\partial \mathbf{u}} &= \begin{bmatrix} -V \sin(\theta + \beta) \frac{\partial \beta}{\partial \delta} & 0 \end{bmatrix}, \\
\frac{\partial \dot{y}}{\partial \mathbf{u}} &= \begin{bmatrix} V \cos(\theta + \beta) \frac{\partial \beta}{\partial \delta} & 0 \end{bmatrix}, \\
\frac{\partial \dot{\theta}}{\partial \mathbf{u}} &= \begin{bmatrix} \frac{V}{\ell_r} \cos(\beta) \frac{\partial \beta}{\partial \delta} & 0 \end{bmatrix}, \\
\frac{\partial \dot{V}}{\partial \mathbf{u}} &= \begin{bmatrix} 0 & \frac{P_{\max}}{Vm} \end{bmatrix}.
\end{aligned}$$

Here:

$$\frac{\partial \beta}{\partial \delta} = \frac{\ell_r}{(\ell_r + \ell_f) \cos^2(\delta)}.$$

Combining all terms:

$$B = \begin{bmatrix} -V \sin(\theta + \beta) \frac{\ell_r}{(\ell_r + \ell_f) \cos^2(\delta)} & 0 \\ V \cos(\theta + \beta) \frac{\ell_r}{(\ell_r + \ell_f) \cos^2(\delta)} & 0 \\ \frac{V}{\ell_r} \cos(\beta) \frac{\ell_r}{(\ell_r + \ell_f) \cos^2(\delta)} & 0 \\ 0 & \frac{P_{\max}}{Vm} \end{bmatrix}.$$

Using the provided constants, we get the numerical matrices A and B explicited below:

- $V_s = 120/3.6 \approx 33.33\,\mathrm{m/s}$,
- $\ell_r = 1.56\,\mathrm{m}, \ell_f = 1.04\,\mathrm{m}$,
- $P_{\max} = 10^5\,\mathrm{W}, \rho = 1.205\,\mathrm{kg/m}^3$,
- $C_d = 0.267, A_f = 2.36\,\mathrm{m}^2, C_r = 0.01, m = 1800\,\mathrm{kg}$,

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 33.33 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2438 \end{bmatrix}.$$

$$B = \begin{bmatrix} 0 & 0 \\ 20 & 0 \\ 12.82 & 0 \\ 0 & 1.667 \end{bmatrix}.$$

Summary

The linearized dynamics of the car system are represented as:

$$\dot{x} = A(x - x_s) + B(u - u_s). \tag{1}$$

These matrices will serve as the foundation for the subsequent controller design.

1.2 Deliverable 2.2: Physical and Mechanical Intuition of Subsystem Separation

In this deliverable, we analyze the reasoning behind separating the system into independent subsystems. This separation simplifies the control design by decoupling the interactions between states and inputs. The car system is governed by the dynamics of position, velocity, and heading angle, while the control inputs include the steering angle (δ) and throttle (u_T). The system's states can be grouped based on how they are influenced by these inputs, as shown below:

- **Lateral Dynamics:** Steering angle δ primarily affects the yaw rate ($\dot{\theta}$) and lateral velocity (v_y), governing the states $y, v_y, \theta, \dot{\theta}$. Adjusting δ changes the car's orientation and lateral position without directly influencing longitudinal states.
- **Longitudinal Dynamics:** Throttle u_T controls the forward velocity and position, governing states x, v_x. This input adjusts the car's speed along the forward direction, without impacting lateral or yaw dynamics significantly.
- **Roll Dynamics:** Roll inputs (e.g., differential forces) operate independently and affect roll angle ϕ and angular velocity $\dot{\phi}$. These dynamics are decoupled from both lateral and longitudinal states under normal conditions.

The sparsity of the Jacobian matrices A and B reinforces this intuition. Each input predominantly influences a subset of states, enabling subsystem separation. For instance, the steering angle δ affects lateral states, while the throttle u_T influences longitudinal states.

Subsystem separation provides key advantages, such as:

- **Simplified Control Design:** Each subsystem can be controlled independently, reducing complexity.
- **Computational Efficiency:** By decoupling dynamics, smaller control problems are solved more efficiently.
- **Improved Stability:** Reduced interactions between subsystems mitigate potential instabilities caused by unmodeled dynamics.

However, under extreme conditions like high-speed maneuvers, coupling between subsystems may increase, necessitating coordination between controllers. Despite this, using the sparsity of the linearized model and understanding the physical dynamics enables effective subsystem separation, simplifying MPC design and improving system performance.

2 Deliverable 3: MPC Controllers Design for Each Sub-System

2.1 Design Procedure to Ensure Recursive Constraint Satisfaction

The design of the longitudinal and lateral MPC controllers was guided by the principle of ensuring recursive constraint satisfaction. This involves ensuring that at every time step, the optimization problem remains feasible, and the system respects the defined constraints on states and inputs.

Longitudinal Controller Design: The longitudinal controller governs the velocity of the car. The design steps were as follows:

- The system dynamics were modeled as a discrete-time state-space representation:
$$x_{k+1} = Ax_k + Bu_k,$$
where $x_k = V$ (velocity) and $u_k = u_T$ (normalized throttle).
- State constraints were not explicitly defined, but input constraints were imposed:
$$-1 \leq u_T \leq 1.$$
- The terminal set X_f was computed using the maximal invariant set approach. The terminal weight Q_f and feedback gain K were calculated using an LQR design:
$$Q_f = \text{solution of discrete-time Riccati equation (DARE)}, \quad K = -(R+B^\top QB)^{-1}B^\top QA.$$
- The terminal constraints ensure that the system remains within the feasible set at the end of the prediction horizon, maintaining recursive feasibility.

Lateral Controller Design: The lateral controller handles the car's lateral position and orientation. The design involved:

- Modeling the system as a discrete-time state-space representation:
$$x_{k+1} = Ax_k + Bu_k,$$
where $x_k = [y, \theta]^\top$ (lateral position and heading angle) and $u_k = \delta$ (steering angle).
- State and input constraints were defined as follows:
$$-0.5 \leq y \leq 3.5, \quad |\theta| \leq 0.0873, \quad |\delta| \leq 0.5236.$$
- Similar to the longitudinal controller, a terminal set X_f and terminal weight Q_f were computed to guarantee stability and feasibility.

2.2 Choice of Parameters Q, R, H

The parameters for the cost function were chosen to balance tracking performance, control effort, and computational complexity:

- Q (state weight): Higher weights were assigned to states critical to system performance. For example, in the lateral controller, higher weight was given to the lateral position y compared to the heading angle θ.
- R (input weight): Penalized excessive control effort. In the longitudinal controller, a smaller R was used to allow more aggressive adjustments to throttle for velocity tracking.
- H (horizon length): Set to 5 seconds for both controllers. This was a trade-off between computational efficiency and the ability to predict future behavior accurately.

2.3 Simulation Setup and Results

The controllers were tested using a nonlinear car model for a 15-second simulation with the following setup:

- **Initial State:** $x_0 = [0, 0, 0, 80/3.6]^\top$.
- **References:** Initial reference $[y, V] = [0, 80/3.6]$, step reference $[y, V] = [3, 120/3.6]$.
- **Prediction Horizon:** $H = 5$ seconds.

Results:

- **Longitudinal Controller:** Velocity tracking was accurate, with minimal throttle effort. Input constraints were respected.
- **Lateral Controller:** Lateral position and heading angle followed the references effectively. Both input and state constraints were satisfied.
- **Stability and Feasibility:** Terminal constraints ensured recursive feasibility and stability.

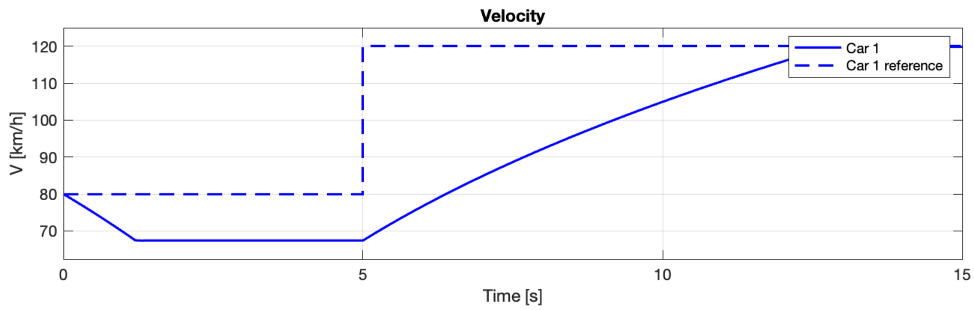

Figure 1: Velocity profile of the car over time, showing the actual velocity (solid line) tracking the reference velocity (dashed line).

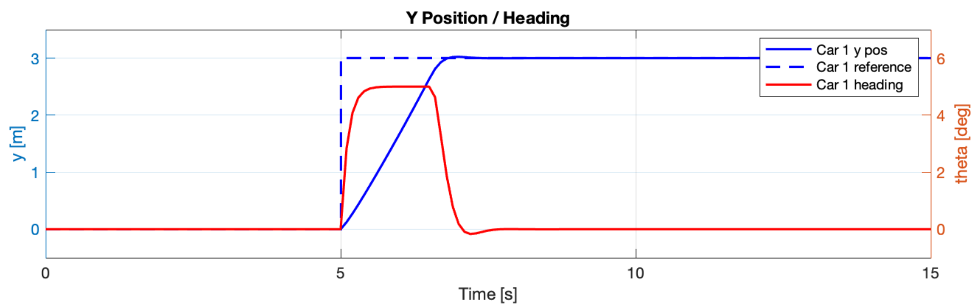

Figure 2: Lateral position (y) and heading angle (θ) evolution over time, with the car's actual states compared to the references.

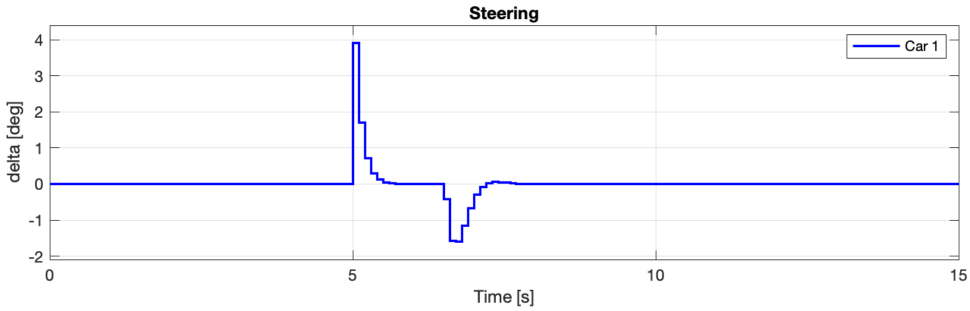

Figure 3: Throttle input (u_T) applied by the longitudinal controller, demonstrating constraint satisfaction and smooth adjustments.

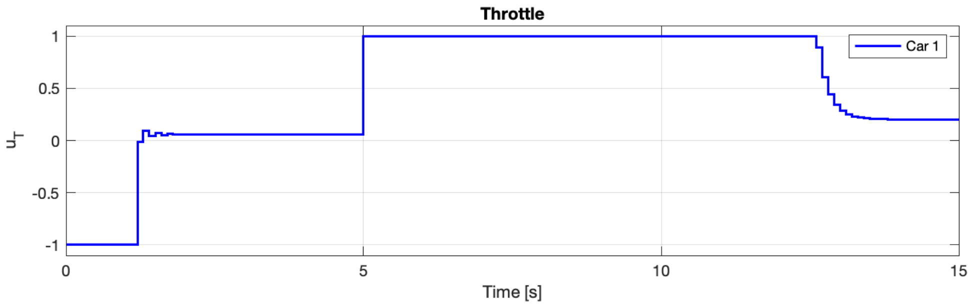

Figure 4: Steering angle (δ) applied by the lateral controller, illustrating control effort and constraint compliance.

2.4 Terminal Invariant Set

To ensure the stability and recursive feasibility of the designed controllers, a terminal invariant set was computed for the system. The terminal invariant set defines a region in the state space within which the system states remain invariant under the applied control law.

Figure 5 illustrates the terminal invariant set in the state space of relative position (Δx) and relative velocity (Δv).

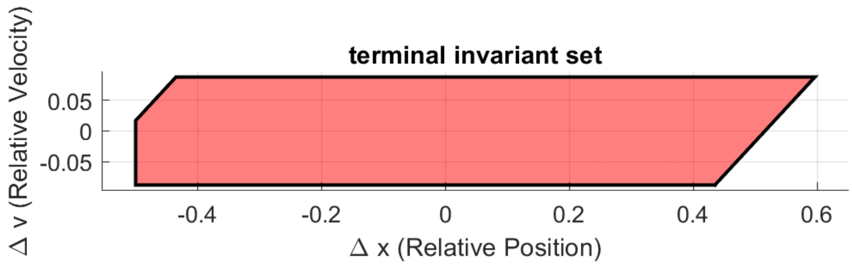

Figure 5: Terminal invariant set in the state space of relative position (Δx) and relative velocity (Δv). The shaded region represents the allowable state trajectories.

Key Features of the Terminal Invariant Set:

- The terminal invariant set ensures that at the end of the prediction horizon, the system state remains in a feasible region, guaranteeing stability and recursive feasibility.
- The shaded region indicates the set of allowable state trajectories that satisfy both the system dynamics and constraints.
- The boundaries of the terminal invariant set are defined to account for the system's physical and control limitations, ensuring that the terminal constraints are met.

By incorporating the terminal invariant set into the MPC design, we ensure that the controllers not only respect constraints during the trajectory but also achieve stability and feasibility at the terminal state.

3 Deliverable 4: Offset-Free MPC and Disturbance Estimation

This deliverable evaluates the capability of the longitudinal MPC controller to maintain offset-free tracking in the presence of disturbances. To achieve this, a disturbance estimator is designed and integrated into the MPC framework, ensuring robust performance even under challenging conditions.

3.1 Design of Longitudinal Disturbance Estimator

To address the effects of persistent disturbances, the longitudinal disturbance estimator augments the original system dynamics by incorporating a disturbance state. This enhancement enables the estimator to identify and compensate for disturbances during operation. The augmented system dynamics are represented as:

$$A_{\text{hat}} = \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix}, \quad B_{\text{hat}} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad C_{\text{hat}} = \begin{bmatrix} C & 0 \end{bmatrix}.$$

Here, A, B, C are the original longitudinal dynamics matrices, extended to model the disturbance. The observer estimates the state and disturbance using:

$$\hat{z}_{k+1} = A_{\text{hat}}\hat{z}_k + B_{\text{hat}}u_k + L(y_k - C_{\text{hat}}\hat{z}_k),$$

where $\hat{z}_k$ contains the system state and the disturbance estimate. The observer gain L is designed using pole placement to ensure stability and quick convergence, with poles selected at $[0.7, 0.8]$.

By prioritizing rapid convergence, the disturbance estimator ensures that the effects of external disturbances are quickly identified and mitigated. At the same time, the tuning process avoids excessive overshoot or noise amplification, preserving system stability.

3.2 Integration with Longitudinal MPC

The disturbance estimate is seamlessly integrated into the longitudinal MPC to ensure offset-free tracking. The prediction model is augmented to include the disturbance:

$$x_{k+1} = Ax_k + Bu_k + Bd_{\text{est}},$$

where d_{est} represents the estimated disturbance. By incorporating this estimate, the MPC dynamically adjusts control inputs to compensate for persistent disturbances.

The cost function for the MPC remains focused on minimizing deviations from the reference trajectory:

$$J = \sum_{k=0}^{N-1} \left[Q(x_k - x_{\text{ref}})^2 + R(u_k - u_{\text{ref}})^2 \right] + Q_f(x_N - x_{\text{ref}})^2,$$

where Q, R, and Q_f are weights that balance state tracking, control effort, and terminal stability. The MPC also enforces constraints on the control inputs ($-1 \leq u_T \leq 1$) and states to ensure safe and reliable operation.

This integration improves the controller's ability to maintain accurate tracking of the reference trajectory, even in the presence of model mismatches or external disturbances.

3.3 Simulation Results

The simulation evaluates the offset-free MPC in a nonlinear car model, starting with an initial state of $x_0 = [0, 0, 0, 80/3.6]^T$. The reference trajectory involves a step change from $[y, V] = [0, 80/3.6]$ to $[y, V] = [3, 120/3.6]$ after 2 seconds. During the simulation, a persistent longitudinal disturbance modeled as an external force is introduced.

Figure 6 demonstrates the controller's ability to track the velocity reference accurately. Although the disturbance initially causes a deviation, the disturbance estimator quickly identifies and compensates for it, ensuring the velocity returns smoothly to the desired trajectory.

The throttle input u_T, shown in Figure 8, reflects the controller's adjustments to counteract the disturbance. The input remains smooth and within constraints, demonstrating the controller's ability to maintain stable operation without abrupt changes.

Figure 9 highlights the disturbance estimator's performance, showing rapid convergence to the true disturbance value. This timely response allows the MPC to accurately compensate for the disturbance and maintain offset-free tracking.

Finally, Figure 7 presents the lateral dynamics, including the car's position and heading angle. These plots confirm that the longitudinal disturbance and its compensation do not interfere with the lateral behavior, validating the decoupled design of the longitudinal and lateral controllers.

Figure 8 shows the throttle input u_T applied by the controller. The adjustments demonstrate the controller's effort to counteract the disturbance while respecting the input constraints. Notably, the throttle input remains smooth, avoiding abrupt changes that could destabilize the system.

The disturbance estimation, as depicted in Figure 9, converges rapidly to the true disturbance value. This convergence ensures that the controller accurately compensates for the disturbance, maintaining offset-free tracking.

Figure 7 presents the lateral position and heading angle over time. These plots demonstrate that the longitudinal disturbance and its compensation do not affect the lateral dynamics, validating the decoupled design of the controllers.

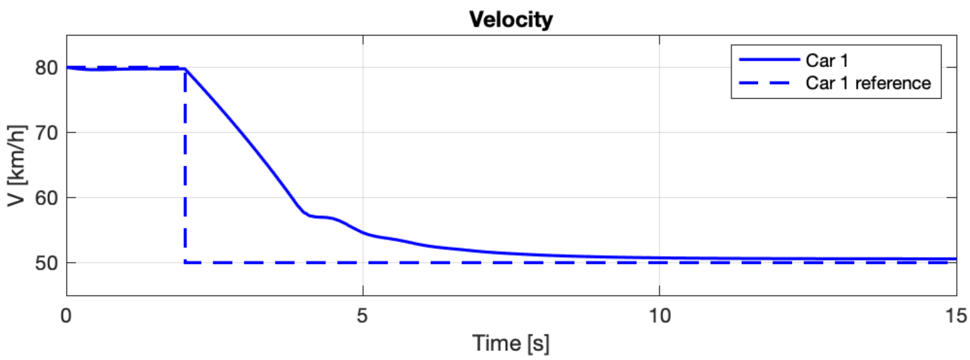

Figure 6: Velocity profile of the car over time, showing accurate tracking of the reference trajectory despite disturbances.

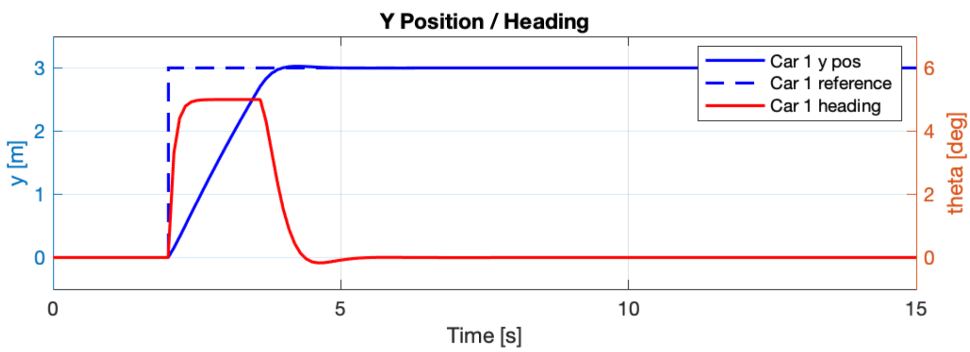

Figure 7: Lateral position and heading angle evolution, demonstrating unaffected lateral dynamics under longitudinal disturbances.

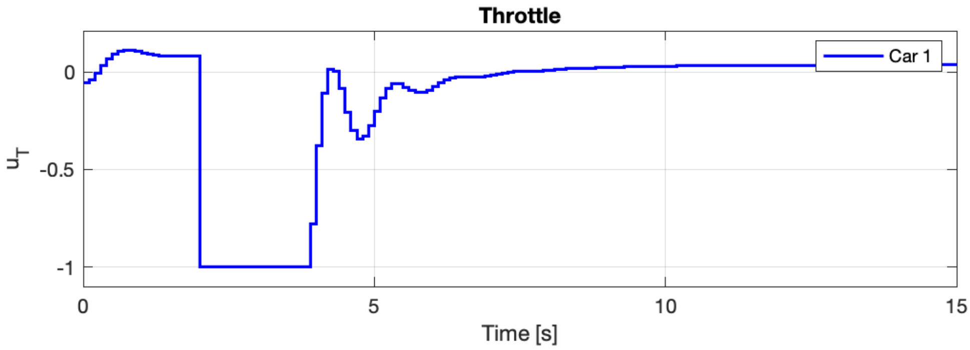

Figure 8: Throttle input trajectory showing adjustments to compensate for disturbances while respecting constraints.

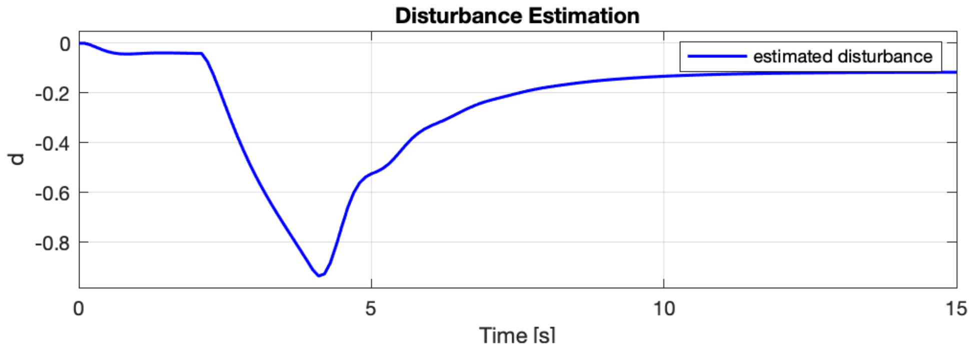

Figure 9: Disturbance estimation over time, converging to the true disturbance value for robust control.

4 Deliverable 5: Robust Tube MPC for Adaptive Cruise Control

4.1 Overview

In this deliverable, we design a Robust Tube MPC to handle adaptive cruise control. The objective is to:

- Control the relative dynamics between the ego car (car under control) and the lead car.
- Track the reference velocity for the ego car while maintaining a minimum distance of 6 meters behind the lead car.
- Account for uncertainty in the behavior of the lead car, specifically its throttle input u_T, which is modeled as a bounded disturbance.

4.2 Components calculated for the controller from `tube_mpc_sets.m`

The following components were calculated and saved in the file `tube_mpc_data.mat`:

- **Minimal Robust Invariant Set (E)**: Represents the set of deviations (error dynamics) from the tube center that are guaranteed to stay bounded under the disturbance. Computed iteratively to ensure recursive feasibility of the controller.

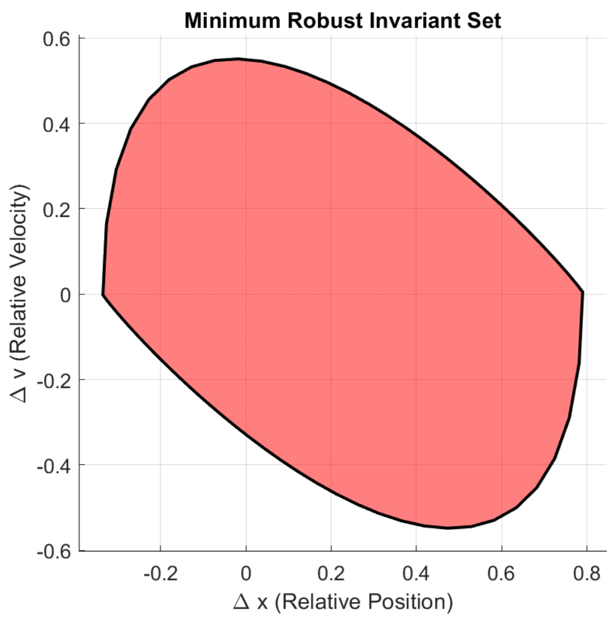

Figure 10: Minimal Robust Invariant Set (E) visualized in the relative state space.

- **Tightened Constraints (X_t and U_t)**:
  - $X_t = X \ominus E$: The tightened state constraints ensure that even with the maximal disturbance, the trajectory remains feasible.
  - $U_t = U \ominus KE$: The tightened input constraints account for the tracking feedback law $u = -Kx$ applied to keep the system within the tube.
- **Terminal Set (X_f)**: Ensures stability by forcing the system to converge to a robust terminal region within the prediction horizon. Computed using iterative methods over the tightened state constraints.

The terminal set guarantees that the point $(0, 0)$ is reachable, even under bounded disturbances, by constraining the trajectory to remain within a feasible region around the origin. This property is critical for ensuring recursive feasibility and stability of the controller.

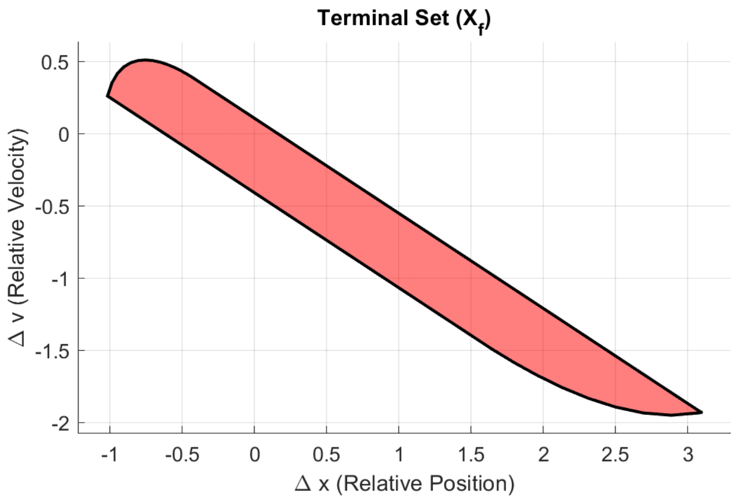

Figure 11: Terminal Set (X_f) ensuring stability within the prediction horizon.

- **Tracking Controller Gain (K)**: Designed using LQR to stabilize the error dynamics and keep the state close to the tube center.

4.3 Controller Design

4.3.1 Relative Dynamics

The relative longitudinal state Δ between the ego car and the lead car is defined as:

$$\Delta = \begin{bmatrix} \Delta x \\ \Delta V \end{bmatrix} = x_{\text{lead}} - x_{\text{ego}} - x_{\text{safe}},$$

where $x_{\text{safe}} = \begin{bmatrix} x_{\text{safe, pos}} \\ 0 \end{bmatrix}$ is a predefined safety distance. We chose $x_{\text{safe, pos}} = 8\,\text{m}$ to be close to the 6-meter minimum distance while allowing a margin for safety. This ensures that even if the lead car brakes suddenly, the ego car has enough time and space to react appropriately, maintaining a safe following distance.

The relative dynamics evolve according to:

$$\Delta^+ = A_d \Delta - B_d u_{\text{ego}} + B_d u_{\text{lead}}.$$

Here, the lead car's throttle u_{lead} is treated as a bounded disturbance:

$$u_{\text{lead}} \in [u_{T,s} - 0.5, u_{T,s} + 0.5].$$

4.3.2 Tightened Constraints

Using the computed minimal invariant set E, the tightened constraints ensure robust feasibility:

$$X_t = X \ominus E, \quad U_t = U \ominus KE.$$

These constraints guarantee that even under worst-case disturbances, the system will remain within the feasible region.

4.3.3 Objective Function

The Tube MPC objective function minimizes the deviation from the reference velocity V_{ref} while penalizing control effort:

$$J = \sum_{k=0}^{N-1} \left(\Delta_k^\top Q \Delta_k + u_{\text{ego},k}^\top R u_{\text{ego},k} \right) + \Delta_N^\top Q_f \Delta_N.$$

4.3.4 Terminal Constraints

The terminal set X_f ensures recursive feasibility and system stability by requiring the system state to converge to a robust terminal region.

4.4 Simulation Setup

Two scenarios were simulated to evaluate the performance of the Tube MPC:

- **Scenario 1**: The ego car starts at $x_{\text{ego}} = [0, 0, 0, 100/3.6]^\top$ with a reference velocity $V_{\text{ref}} = 120/3.6$. The lead car starts 15 meters ahead with a constant velocity of $100/3.6$.
- **Scenario 2**: The ego car starts at a higher initial velocity $115/3.6$ with the same reference velocity. The lead car starts 8 meters ahead and changes throttle dynamically.

In both scenarios, the robust Tube MPC ensures that the ego car maintains a safe distance while smoothly adjusting its velocity.

4.5 Simulation Results

The following plots illustrate the performance of the robust Tube MPC in two different scenarios: (1) with the lead car maintaining a fixed throttle and (2) with the lead car's throttle varying as a bounded disturbance ($u_T \pm 0.5$). These results validate the controller's ability to handle disturbances while ensuring safe and smooth operation.

4.5.1 Experiment 1: Fixed Throttle for Lead Car

In the first experiment, the lead car maintains a fixed throttle. The ego car adjusts its throttle to track the reference velocity while maintaining a safe distance from the lead car.

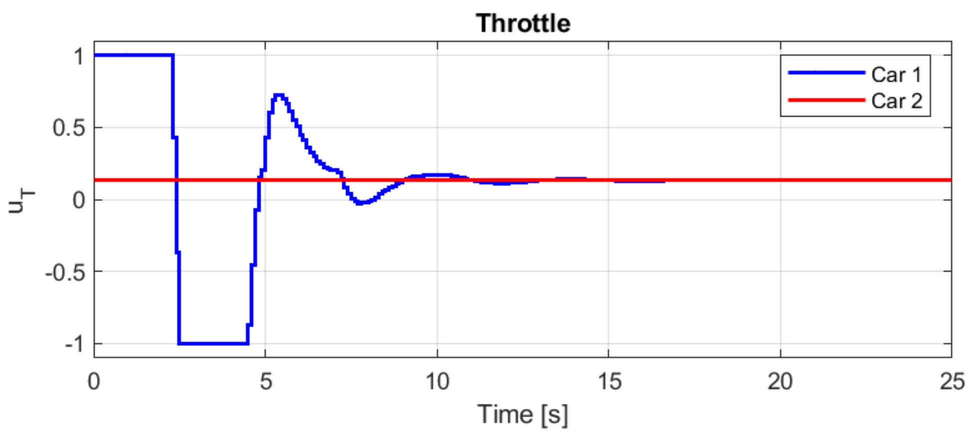

Figure 12: Throttle input (u_T).

As shown in Figure 12, the ego car's throttle adjusts dynamically to ensure compliance with the constraints.

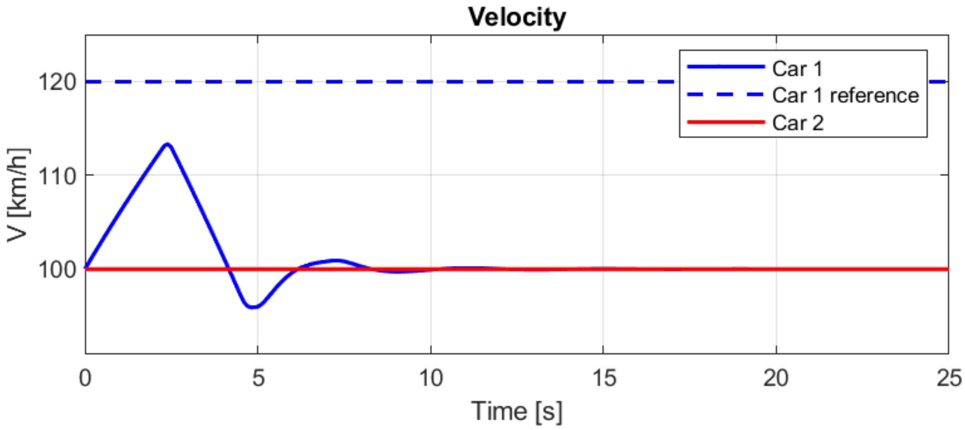

Figure 13: Velocity.

Figure 13 shows that the ego car initially accelerates to reach the reference velocity. However, upon detecting the minimum safe distance from the lead car, it slows down to maintain a constant distance.

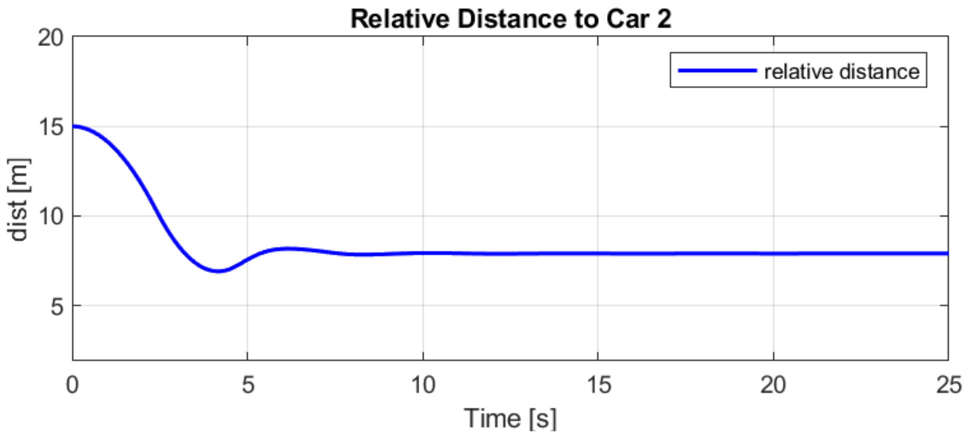

Figure 14: Distance between the ego car and the lead car.

Finally, Figure 14 confirms that the ego car maintains the minimum safe distance from the lead car throughout the simulation.

4.5.2 Experiment 2: Throttle Disturbance for Lead Car

In the second experiment, the lead car's throttle varies as a bounded disturbance ($u_T \pm 0.5$). The ego car adapts its throttle to maintain a safe distance while tracking the reference velocity.

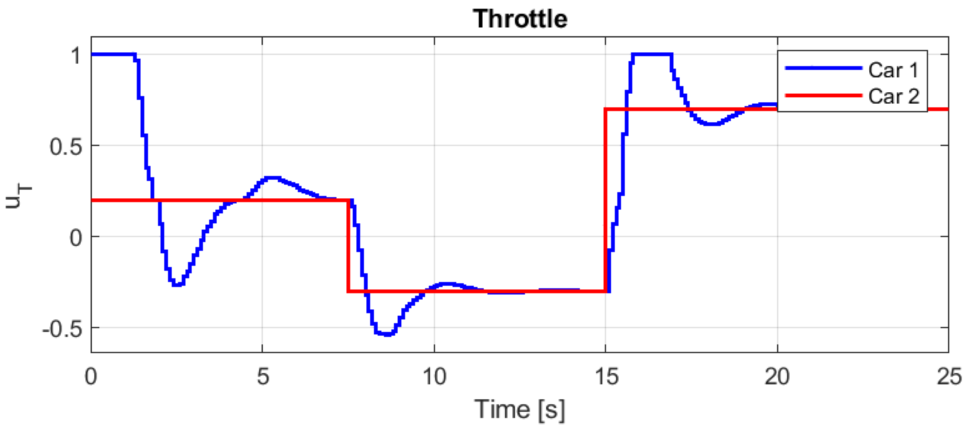

Figure 15: Throttle input (u_T).

In Figure 15, we observe that the ego car's throttle adjusts dynamically to account for the varying throttle of the lead car.

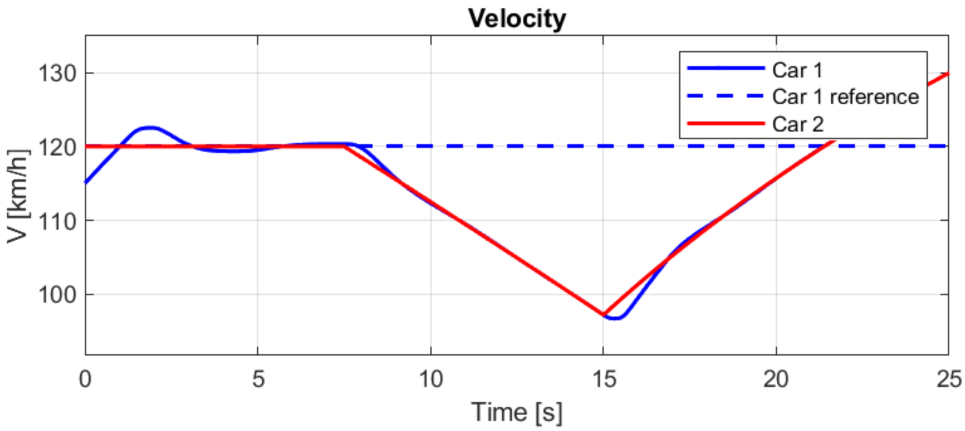

Figure 16: Velocity.

As shown in Figure 16, the ego car adjusts its velocity dynamically to handle the throttle changes of the lead car, ensuring smooth operation.

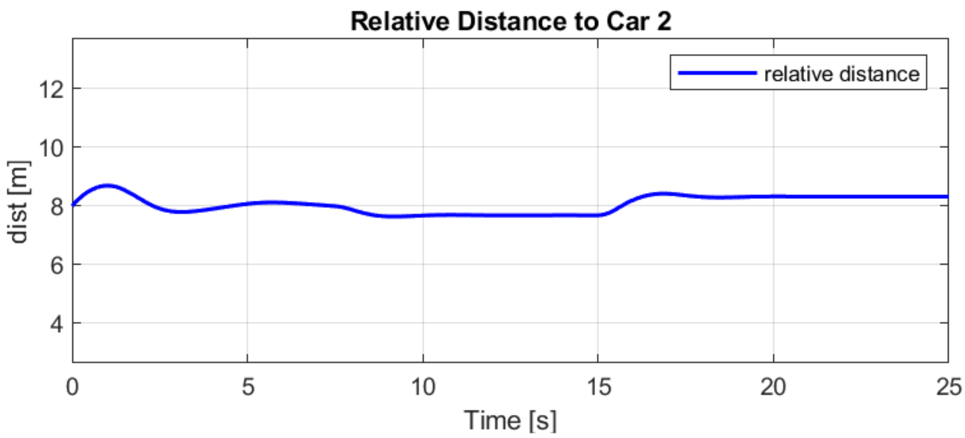

Figure 17: Distance between the ego car and the lead car.

Finally, Figure 17 confirms that the ego car maintains a safe distance from the lead car throughout the experiment, even with the varying throttle.

5 Deliverable 6: Robust Tube MPC for Adaptive Cruise Control

This section outlines the design, implementation, and evaluation of a Nonlinear Model Predictive Controller (NMPC) tailored for the car system. The NMPC predicts the system's future states over a finite horizon and computes optimal control inputs to minimize deviations from a reference trajectory, all while adhering to physical and operational constraints.

The system's continuous-time dynamics are discretized using the Runge-Kutta 4 (RK4) method, which ensures precise state predictions over the prediction horizon. A prediction horizon of 5 seconds was selected, providing a balance between computational feasibility and the ability to anticipate future system behavior accurately. The NMPC tracks the state variables $X = [x, y, \theta, V]^\top$, representing position, lateral displacement, heading angle, and velocity, while the control inputs $U = [\delta, u_T]^\top$ correspond to steering angle and throttle.

To ensure safe and efficient operation, the NMPC incorporates constraints: $-0.5 \leq y \leq 3.5$ for lateral position, $-5^\circ \leq \theta \leq 5^\circ$ for heading angle, $-30^\circ \leq \delta \leq 30^\circ$ for steering angle, and $-1 \leq u_T \leq 1$ for throttle input. These constraints prevent unsafe maneuvers while allowing sufficient control authority.

The cost function is designed to balance tracking accuracy and control effort:

$$J = \sum_{k=0}^{N-1} \left[Q_y (y_k - y_{\text{ref}})^2 + Q_V (V_k - V_{\text{ref}})^2 + R_\delta \delta_k^2 + R_{u_T} u_{T_k}^2 \right],$$

where $Q_y = 1$ emphasizes accurate lateral tracking, $Q_V = 0.1$ smooth velocity tracking, and $R_\delta = R_{u_T} = 0.01$ penalize aggressive control inputs. The CasADi framework and IPOPT solver were configured for efficient optimization, ensuring real-time feasibility.

The NMPC eliminates steady-state tracking error by considering nonlinear dynamics and optimizing control inputs directly. Its predictive capability enables smooth transitions and accurate tracking, even during abrupt reference changes.

5.1 Simulation Results

The NMPC was tested in a simulation environment with an initial state $x_0 = [0, 0, 0, 80/3.6]^\top$. Two reference trajectories were applied sequentially: $[y, V] = [0, 80/3.6]^\top$ initially, transitioning to $[y, V] = [3, 100/3.6]^\top$ after 2 seconds. The simulation results demonstrated the controller's ability to track references accurately while respecting constraints.

Velocity Tracking The velocity tracking is shown in Figure 18, highlighting smooth transitions and precise adherence to the reference trajectory. The NMPC effectively adjusts throttle inputs to maintain velocity within constraints.

Lateral Tracking The lateral position tracking in Figure 19 confirms minimal deviation from the desired trajectory, demonstrating the NMPC's ability to maintain stable lateral dynamics.

Control Inputs Figures 20 and 21 illustrate the control inputs. The throttle trajectory remains smooth, while the steering angle exhibits rapid yet controlled adjustments during transitions. Both inputs respect operational constraints, ensuring efficient control efforts.

The results validate the NMPC's effectiveness in handling nonlinear dynamics, enforcing constraints, and delivering robust performance under varying operating conditions.

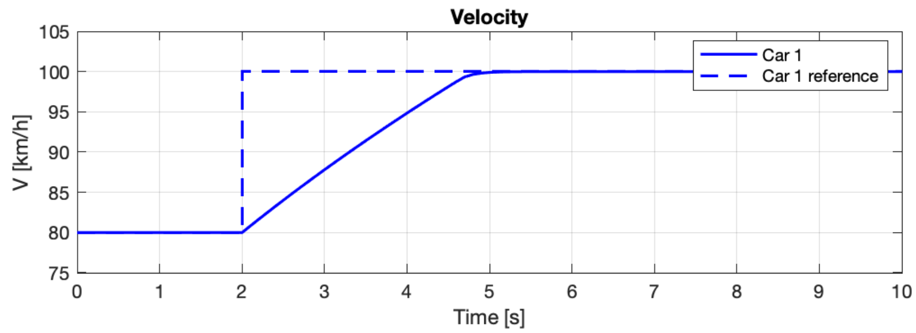

Figure 18: Velocity tracking over time, showing smooth transitions and accurate tracking of the reference velocity.

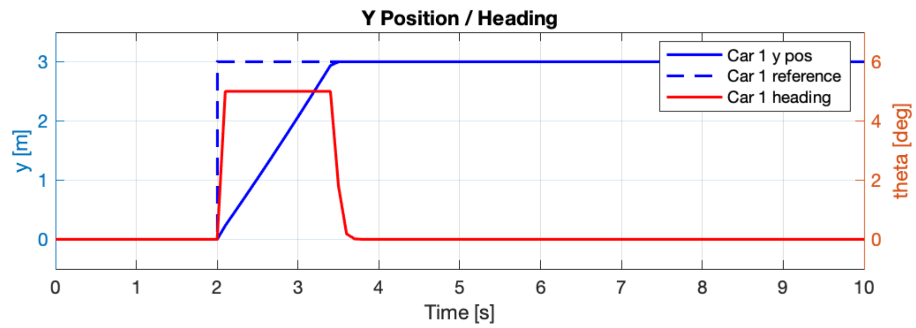

Figure 19: Lateral position tracking, demonstrating minimal deviation from the reference trajectory.

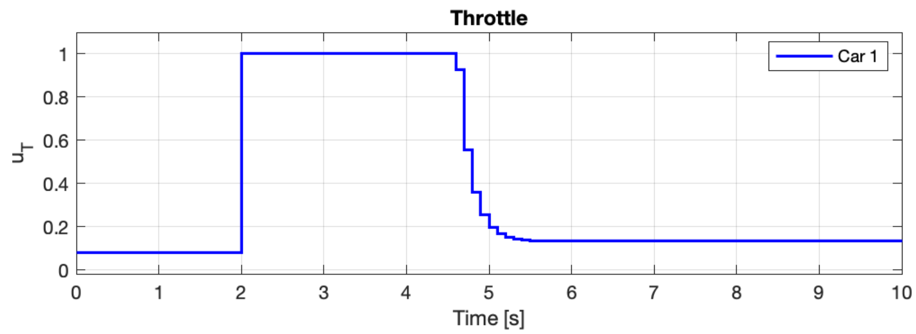

Figure 20: Throttle input trajectory, showing smooth adjustments within constraints.

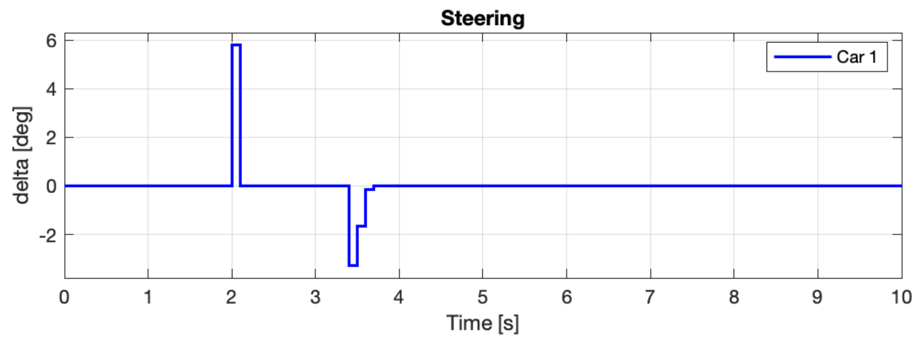

Figure 21: Steering angle trajectory, illustrating efficient control adjustments during transitions.