Middle East Technical University

Department of Electrical and Electronics Engineering

Final Project Report

Control System Design and Simulation

Part 1: Control System Design

Student Name: Aylar Roshan Nahad

Course: EE498 – SPECIAL TOPICS: CONTROL SYSTEM DESIGN AND SIMULATION

June 21, 2025

Contents

1 Overview and Introduction

The first part of the final project is dedicated to a controller design problem. The task is to design a controller for adaptive cruise control (ACC). The controller is based on the model predictive control (MPC) approach described in the paper "Model Predictive Adaptive Cruise Control" by E. Kural and B. Aksun Güvenç, presented at the 2010 IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, pp. 1455–1461, doi: 10.1109/ICSMC.2010.5642478. [2]

In the following sections, we introduce the design problem and necessary details on the theoretical and mathematical background. Subsequently, we go over the design procedure step by step. Afterwards, the design performance is analyzed through various simulations and relevant evaluation metrics. The report is concluded by a summary on the procedure and results.

2 System Model

In this section, we determine a suitable system model for the adaptive cruise control (ACC) system. First, we take a brief look at the longitudinal dynamics. While a PI controller is responsible for this subsystem, it can still provide valuable insight and intuition required for the design problem and MPC problem constraints. Following that, the high level inter-vehicle dynamics are represented. Based on this model structure, the MPC algorithm is employed to calculate desired acceleration and deceleration. [3]

2.1 Low Level Controller and Vehicle Longitudinal Dynamics

At the low level, the desired acceleration/deceleration values is transferred to an appropriate brake and throttle maneuver and a PI type controller is responsible for tracking the desired point. The sampling time for this controller is lower than the high level controller. Until the next change in desired reference value, this controller ensures the desired manuever is achieved. The model described for representing this dynamics consists of several subsystems:

- **Vehicle Dynamics**
  To represent the vehicles longitudinal and lateral dynamics, a nonlinear single track model augmented with a longitudinal model is used. [5]

- **Tire Force Calculation**
  Tire forces are calculated using the Dugoff tire model.

- **Powertrain Dynamics**
  All inertias related to rotating parts (engine, differential and transmission) are described using equivalent inertia, J_{eq}. Additionally, an automatic gear ratio is added by using a shifting logic at maximum power points of the engine where some hysteresis is being considered in order to prevent consecutive up-and down gear changes.

- **Engine Representation**
  The engine is represented as a look-up table where its inputs are throttle and engine speed and its output is the engine torque.

2.2 High Level Controller and Inter-vehicle Dynamics

At the high level, the MPC algorithm calculates the desired acceleration and deceleration values for the ongoing traffic scenario. The inter-vehicle dynamics are represented in the MPC structure. This model is linear as presented by Luo et al. [3]

Given that the lower level controller is designed well, its behavior is usually approximated by a first order system:

$$\tau \frac{d\ddot{s}(t)}{dt} + \ddot{s}(t) = u(t)$$

The discrete-time expression can be derived using difference approximation as:

$$a(k+1) = \left(1 - \frac{T_s}{\tau}\right) a(k) + \frac{T_s}{\tau} u(k)$$

Taking the acceleration of the preceding vehicle as a disturbance, we can form the state space model as given below. Additionally, in the table 1, all notation is explained.

$$x(k+1) = Ax(k) + Bu(k) + Ew(k)$$

$$y(k) = Cx(k) + F$$

where:

$$A = \begin{bmatrix} 1 & 0 & T & -0.5T^2 & 0 \\ 0 & 1 & 0 & T & 0 \\ 0 & 0 & 1 & -T & 0 \\ 0 & 0 & 0 & 1 - \frac{T}{\tau} & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{T}{\tau} \\ \frac{1}{\tau} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & -t_h & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$E = \begin{bmatrix} 0.5T^2 \\ 0 \\ T \\ 0 \\ 0 \end{bmatrix}, \quad F = \begin{bmatrix} -d_0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$x(k) \triangleq \begin{bmatrix} d & \upsilon_{ACC} & \upsilon_{rel} & a & \frac{\{a(k)-a(k-1)\}}{T} \end{bmatrix}$$

$$w(k) = a_p$$

$$y(k) = \begin{bmatrix} \Delta x & \upsilon_{rel} & a & \frac{\{a(k)-a(k-1)\}}{T} \end{bmatrix}$$

Table 1: Explanation of Notation

Symbol	**Description**
T	Sampling period of the control system
t_h	Time headway between vehicles
d_0	Safe distance between vehicles if the vehicles are stopped
τ	Time constant of the low level controller
d	Inter-vehicle distance
υ_{ACC}	Speed of the ACC equipped vehicle
υ_{rel}	Relative Speed
a	Acceleration of the ACC equipped vehicle
$\frac{\{a(k)-a(k-1)\}}{T}$	Jerk of the ACC equipped vehicle
a_p	Acceleration of the preceding vehicle
Δx	Relative Distance

From this point forward, we will be using this state space structure for our MPC design problem and define model constraints and goals accordingly.

3 System Features and suitability of MPC Algorithm

In this section, the possible motives for considering the MPC algorithm as a suitable approach for an adaptive cruise control scheme are discussed.

An excellent ACC system should have good track capabilities, address issues of fuel economy and consider the drivers longitudinal ride comfort. This makes it a multi-objective optimal control problem and a proper control algorithm needs to take the trade-off between these objectives into consideration.[4]

An MPC control scheme can introduce the system constraints within a quadratic programming problem and find the optimal control inputs that minimize the specified cost function. Taking cost function weights and constraints as design parameters, the MPC structure provides the option to optimize control inputs with consideration of design specifications such as level of comfort, power efficiency and tracking accuracy.

However, certain considerations must be made for implementation of an MPC-based controller. Since MPC is an online optimization procedure, it requires a high computational speed. Therefore, it sets a limit on the sampling time which must be considered while developing and tuning the design parameters.

Additionally, an MPC-based controller has many design parameters which require tuning. The upside is that it provides flexibility in its implementation to various scenarios. The challenge is that its proper tuning with the consideration of stability and system constraints might be more difficult compared to traditional controllers.

4 Controller Design

To proceed, we start with explaining the structure of the controller specified in the paper.[2] We will use the insights as an starting point for our own design scheme. Later on, we will adjust the structure according to our simulation results and design preferences.

4.1 Brief Explanation of the Controller Given in [2]

We begin by introducing the prediction model, which is used to predict the behavior of the system during the prediction horizon, N_p. Based on the information on the current state and future control inputs, this model extrapolates the behavior until the end of the horizon. The state space model is presented below and in the table 2, the notation is explained.

$$\hat{x}_p(k+N_p|k) = \tilde{A}x(k) + \tilde{B}U(k+N_c) + \tilde{E}W(k+N_p) + K_x(x(k) - \hat{x}_p(k|k-1)$$
$$\hat{y}_p(k+N_p|k) = \tilde{C}x(k) + \tilde{D}U(k+N_c) + \tilde{G}W(k+N_p) - \tilde{F} + K_y(y(k) - \hat{y}_p(k|k-1)$$

Table 2: Explanation of Notation

Symbol	Description
N_p	Prediction horizon
N_c	Control Horizon
$\hat{x}_p(k+N_p\|k)$	Predicted state vector
$\hat{y}_p(k+N_p\|k)$	Predicted output vector
$U(k+N_c)$	control input vector
$W(k+N_p)$	Disturbance vector (including the estimate of future disturbance values)
$\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{E}, \tilde{F}, \tilde{G}$	Expanded state matrices of the prediction model
K_x	Correction factor using state error
K_y	Correction factor using output error

Next, we focus on the cost function. We begin by only penalizing output state's tracking error. Therefore, the total cost function takes the shape:

$$J = \sum_{i=1}^{N_p} (\hat{y}_p(k+i|k) - y_{ref}(k+i))^T Q (\hat{y}_p(k+i|k) - y_{ref}(k+i)) + \sum_{i=1}^{N_c} u(k+i)^T R u(k+i)$$

As for the constraints, we consider the following:

$$\begin{aligned}
\text{Objective constraints:} \quad & d_0 \le d && \text{(fixed)} \\
\text{System constraints:} \quad & v_{min} \le v \le v_{max} && \text{(fixed)} \\
& a_{min} \le a \le a_{max} && \text{(design parameter)} \\
& j_{min} \le j \le j_{max} && \text{(design parameter)}
\end{aligned}$$

Now that we have defined both cost function and problem constraints, we can formulatize it as a quadratic programming (QP) problem which is solved in real time. The QP problem takes the form:

$$min_{U(N_c)} J(u(N_c), x(N_p)), \quad \text{subject to: } MU(N_c) \le \gamma$$

The QP problem is solved for each time step and only the first value of the control vector is applied to the system. Additionally, its control vector is used to predict the future step in the prediction model. Until the next step, this control signal is given to the low-level controller as the desired acceleration value.

In the paper, the numerical values for the control parameters are based on the previous works in the literature. [1] These are provided in table 3

Table 3: Control Parameters

Parameter	**Symbol**	**Numerical Values**
Prediction horizon	N_p	10
Control Horizon	N_c	5
Min. acceleration limit	a_{min}	-5m/s^2
Max. acceleration limit	a_{max}	2m/s^2
Min. jerk limit	j_{min}	-5m/s^3
Max. jerk limit	j_{max}	2m/s^3
Stopping distance	d_0	5m
Min. ACC vehicle speed	v_{min}	0 m/s
Max. ACC vehicle speed	v_{max}	30 m/s

Within the paper, the controller is tested for various scenarios such as:

- The ACC equipped vehicle cruises with a constant speed set by the driver (cruise control mode), and detects a preceding vehicle with a speed lower than its cruise control speed
- The preceding vehicle accelerates during a following condition
- A third vehicle is entered between the vehicles with the same speed and the ACC vehicle regulates its speed to adjust the inter-vehicle distance

Now that we have summarized the findings from the paper, we can define our own controller design formulation.

4.2 Controller Design Formulation

The design will be based on the system with no disturbances and assuming $w(k) = 0$. Later on, we will simulate the system with the inclusion of possible distubances, study their effect and if needed, adjust the controller.

The parameters for the ACC system are chosen according to previous works in the literature. [6] They are provided in table 4

Table 4: Parameters of the ACC System

Parameter	**Symbol**	**Numerical Values**
Sampling time	T	0.2 s
Time headway between vehicles	t_h	1.5 s
Time constant of the low level controller	τ	0.5 s

In our output equation, we eliminate the constant F vector since it is more intuitive to analyze the system without it. Therefore, the controller is designed (and tested) with respect to the state space as given below:

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$

where:

$$A = \begin{bmatrix} 1 & 0 & 0.2 & -0.02 & 0 \\ 0 & 1 & 0 & 0.2 & 0 \\ 0 & 0 & 1 & -0.2 & 0 \\ 0 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & -2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.4 \\ 2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & -1.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4.2.1 Setting Up Constraints

We reformulate the constraint on distance based on $y_1 = \Delta(x)$ as $0 \leq y_1$. Additionally, we also consider input constraints. Based on [3], we define it as:

$$u_{min} \leq u(k) \leq u_{max}, \quad u_{min} = -5.5m/s^2, u_{max} = 2.5m/s^2$$

We keep the rest of constraints and control parameters the same as given in 3 with the exception of taking $N_c = N_p = 10$.

To be able to isolate $\Delta(x) = y_1$, $v_{ACC} = x_2$, $a = x_4$ and $j = x_5$ for imposing their corresponding constraints, we also define matrices S_1, S_2, S_4 and S_5 as:

$$c_1 = \begin{bmatrix} 1 & -t_h & 0 & 0 & 0 \end{bmatrix} \quad \rightarrow \quad C_1 = \begin{bmatrix} c_1 & 0_{5\times1} & \cdots & 0_{5\times1} & 0_{5\times1} \\ \vdots & & & & \\ 0_{5\times1} & 0_{5\times1} & \cdots & 0_{5\times1} & c_1 \end{bmatrix}, \quad where: \quad \begin{bmatrix} \Delta(x)_{0|k} \\ \vdots \\ \Delta(x)_{10|k} \end{bmatrix} = C_1 \begin{bmatrix} x_{0|k} \\ \vdots \\ x_{10|k} \end{bmatrix}$$

Similar structures are used for S_2, S_4 and S_5.

We can define all the state constraints and input constraints as linear matrix operation now.

$$\begin{bmatrix} I \\ \hline -I \end{bmatrix} \begin{bmatrix} u_{0|k} \\ \vdots \\ u_{9|k} \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ \vdots \\ u_{max} \\ \hline -u_{min} \\ \vdots \\ -u_{min} \end{bmatrix} \quad \rightarrow F_1 U \leq V_1$$

$$S_1 X_k \leq \begin{bmatrix} d_0 \\ \vdots \\ d_0 \end{bmatrix} \quad \xrightarrow{X_k = GU_k + Hx_{0|k}} \quad S_1 G U_k \leq \begin{bmatrix} d_0 \\ \vdots \\ d_0 \end{bmatrix} - S_1 H x_{0|k} \quad \rightarrow \quad F_2 U \leq V_2$$

$$\begin{bmatrix} S_2 \\ \hline -S_2 \end{bmatrix} X_k \leq \begin{bmatrix} v_{max} \\ \vdots \\ v_{max} \\ \hline -v_{min} \\ \vdots \\ -v_{min} \end{bmatrix} \quad \xrightarrow{X_k = GU_k + Hx_{0|k}} \quad \begin{bmatrix} S_2 \\ \hline -S_2 \end{bmatrix} G U_k \leq \begin{bmatrix} v_{max} \\ \vdots \\ v_{max} \\ \hline -v_{min} \\ \vdots \\ -v_{min} \end{bmatrix} - \begin{bmatrix} S_2 \\ \hline -S_2 \end{bmatrix} H x_{0|k} \quad \rightarrow \quad F_3 U \leq V_3$$

Similar formulation as $F_3 U \leq V_3$ for imposing $a_{min} \leq x_{4,j|k} \leq a_{max}$ and $j_{min} \leq x_{5,j|k} \leq j_{max}$, resulting in $F_4 U \leq V_4$ and $F_5 U \leq V_5$ respectively.

4.2.2 Formulating Optimization as a QP Problem

Given the nature of our design problem, it is sufficient to consider the output parameters in our cost function. We set up our reference vector as $y_r = \begin{bmatrix} d_0 & 0 & 0 & 0 \end{bmatrix}^T$. The goal of this controller is to minimize $(||y_{j|k} - r||_2)$'s and $||u_{j|k}||_2$'s. This objective can be formulated in the general MPC-based controller form.

$$min_{u_{0|k}, u_{1|k}, \cdots, u_{9|k}} \{J = \sum_{j=0}^{9} ((y_{j|k} - y_r)^T Q (y_{j|k} - y_r) + u_{j_k}^T R u_{j_k}) + (y_{10|k} - y_r) Q_f (y_{10|k} - y_r)$$

subject to:

$$x_{j+1|k} = Ax_{j|k} + Bu_{j|k}, \quad j = 0, 1, \cdots, 9, \quad x_{0|k} \quad given$$

$$y_{j|k} = Cx_{j|k}$$

$$0 \leq y_{1,j|k}$$

$$\upsilon_{min} \leq x_{2,j|k} \leq \upsilon_{max}$$

$$a_{min} \leq x_{4,j|k} \leq a_{max}$$

$$j_{min} \leq x_{5,j|k} \leq j_{max}$$

$$u_{min} \leq u_{j|k} \leq u_{max}$$

Given $N_p = N_c = 10$, we also define G and H matrices as:

$$X_k = GU_k + Hx_{0|k} \rightarrow G = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ A^9B & A^8B & A^7B & \cdots & B \end{bmatrix}, \quad H = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{10} \end{bmatrix}, \quad where: \quad X_k = GU_k + Hx_{0|k}$$

Based on our optimization problem, a brief derivation of the quadratic function's symmetric positive semi-definite matrix H and linear cost vector f parameters for the QP problem is provided below.

$$Y_k = CGU_k + CHx_{0|k}, \quad Y_r = \begin{bmatrix} y_r & y_r & \cdots & y_r \end{bmatrix}^T, \quad \underline{C} = blkdiag(C, C, ..., C)$$

$$\underline{Q} \triangleq blkdiag(Q, Q, ..., Q, Q_f, \quad \underline{R} \triangleq blkdiag(R, R, ..., R)$$

$$J = (Y_k - Y_r)^T \underline{Q}(Y_k - Y_r) + U^T \underline{R} U$$
$$= (\underline{C}GU_k + \underline{C}Hx_{0|k} - Y_r)^T \underline{Q}(\underline{C}GU_k + \underline{C}Hx_{0|k} - Y_r) + U^T \underline{R} U$$

$$J = U_k^T (G^T \underline{C}^T \underline{Q}\underline{C}G)U_k + [2(\underline{C}Hx_{0|k} - Y_r)^T \underline{Q}\underline{C}G]U_k + (\text{Independent parameters in terms of input } U_k)$$

$$\boldsymbol{H \triangleq G^T \underline{C}^T \underline{Q}\underline{C}G, \quad f \triangleq \underline{C}Hx_{0|k} - Y_r)^T \underline{Q}\underline{C}G}$$

We are now ready to formulate the QP problem that needs to be solved at each time step:
MATLAB's `quadprog` function has the structure:

$$min_{U_k} \frac{1}{2} U_k^T H U_k + f^T U_k, \quad s.t: \quad FU_k \leq V_k$$

Now, we can solve the on-line finite horizon constraint LQR problem at each step k and apply $u_{0|k}$. This completes the theoretical formulation part. From this point, we can work on adjusting the parameters to enhance the controller's performance in various scenarios.

4.3 Tuning Controller Parameters and Simulation

Our design parameters are Q, Q_f and R. These design parameters are tuned based on our simulation results for the two following scenarios:

- Following a leader vehicle at constant speed
- Following a leader vehicle performing speed changes

Additionally, our tuned controller is tested on the system with the inclusion of possible disturbances to analyze their effect.
Given $Q = \text{blkdiag}(q_1, q_2, q_3, q_4)$, the parameters penalize relative distance, relative velocity, acceleration and jerk respectively. This matrix is set to $Q = \text{blkdiag}(5, 10, 1, 1)$ for proper tracking while avoiding abrupt changes. Additionally, Q_f is set to be equal to Q. For a more aggressive controller,

Q_f can be chosen to have higher values than Q. Finally, R is set to $R = 0.001$ to provide proper input effort while still considering efficiency concerns.

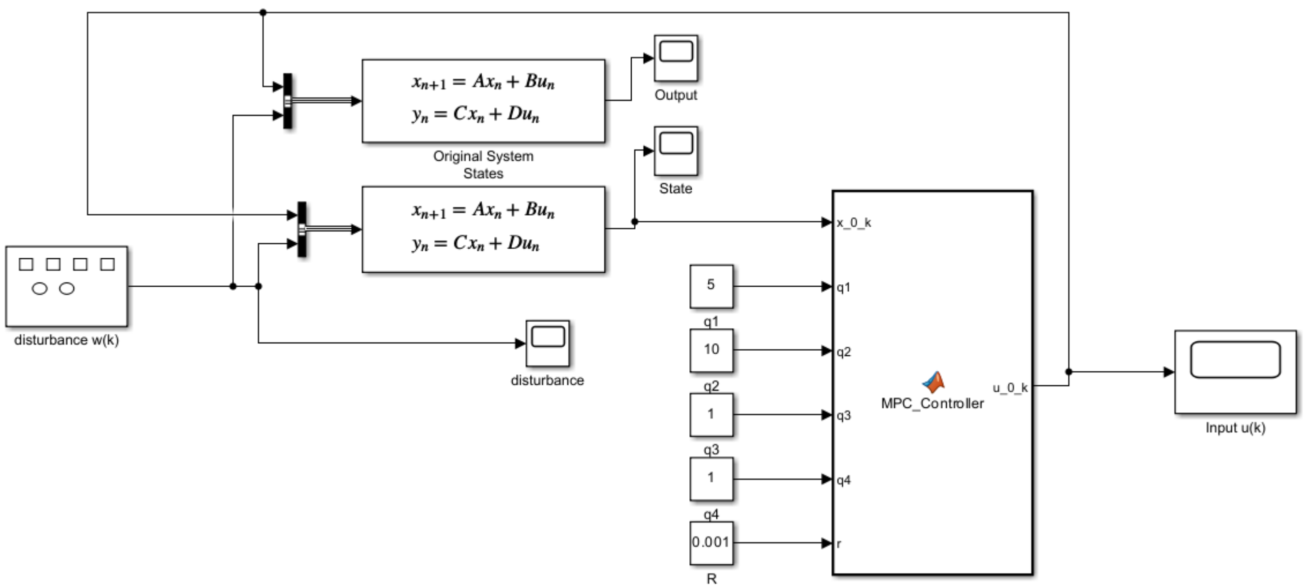
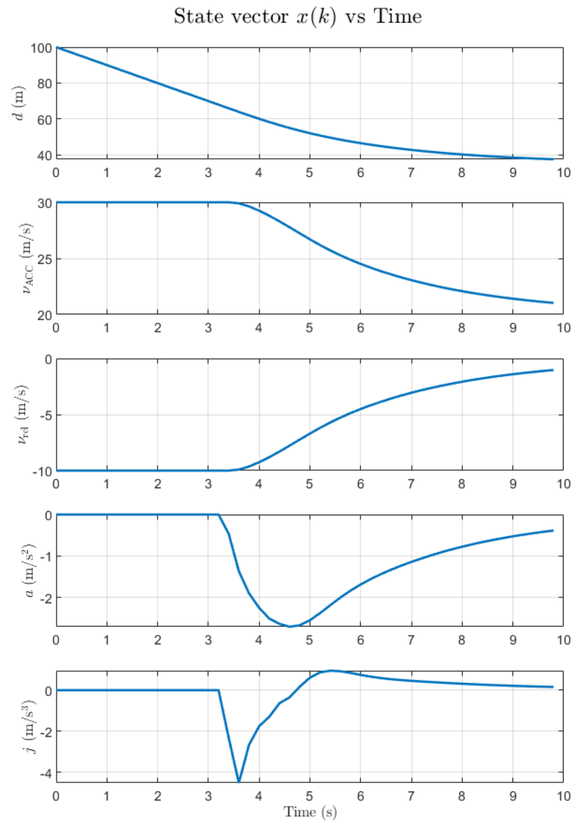


Figure 1: Schematic of the simulation. Disturbance included.

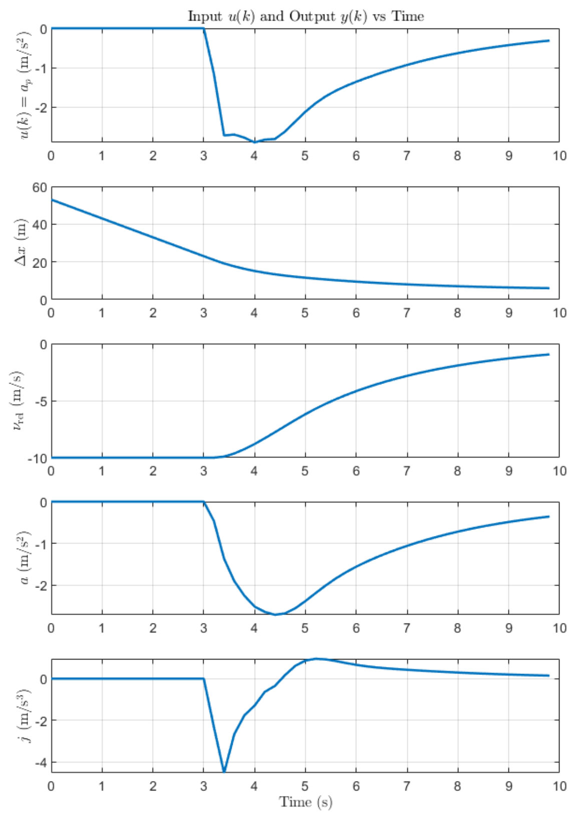
Following, responses for this controller choice are illustrated and results are analyzed.

Following a Leader Vehicle at Constant Speed

We set the initial condition as: $x_0 = [100,\ 30,\ -10,\ 0,\ 0]^T$. The controller performance is as given:



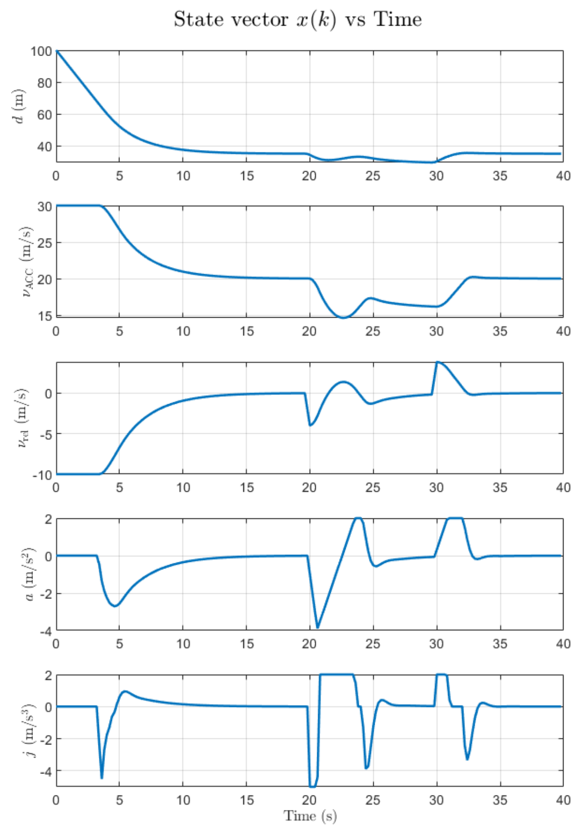
State response



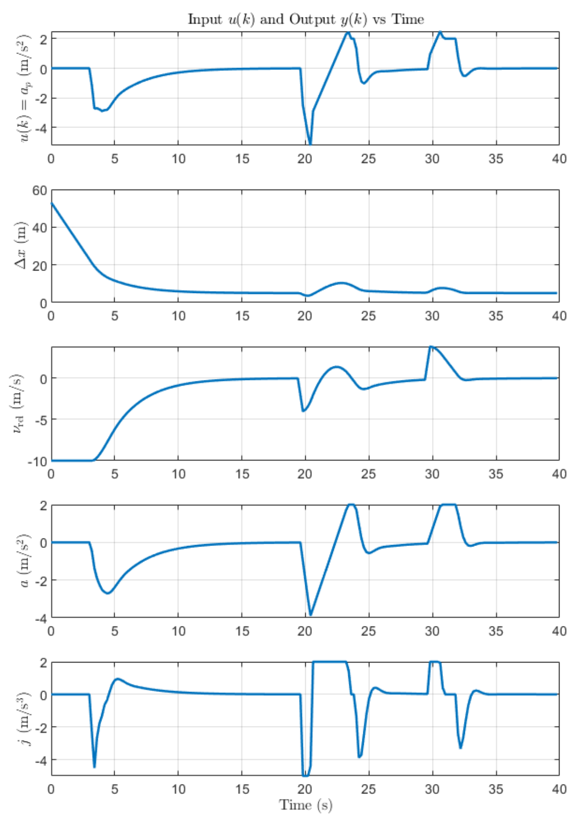
Input and Output signal

Following a Leader Vehicle Performing Speed Changes

We set the initial condition same as before and we introduce speed changes of leader vehicle via inclusion of nonzero constant $w(k)$ disturbance input during several short intervals. Results for the same controller are provided:



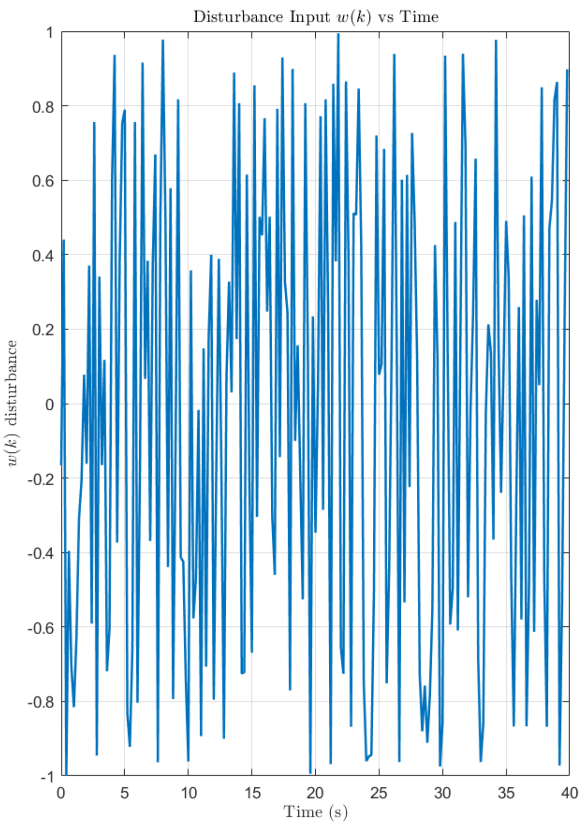
State response



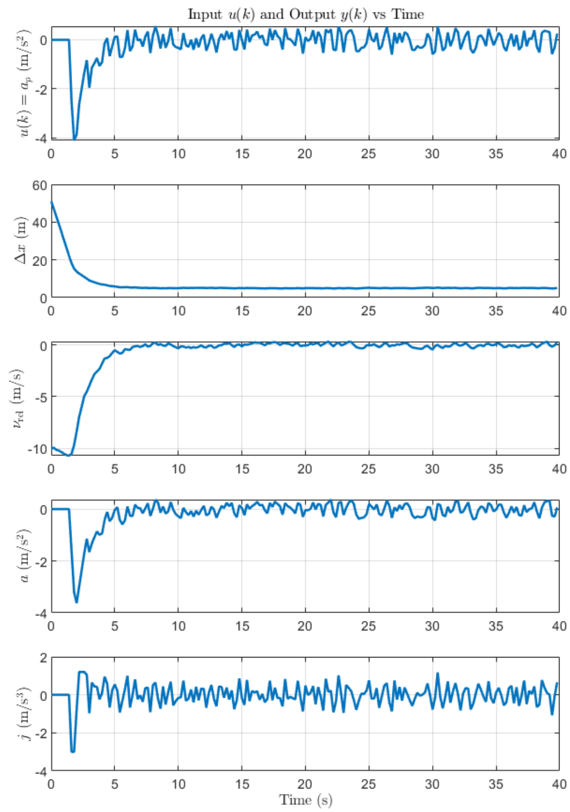
Input and Output signal

Possible Disturbances and Their Effects

We set the initial condition same as before and model disturbance $w(k)$ as a white noise signal. The response is as follows:



Disturbance



Input and Output signal

4.4 Main Observations

- Given the initial relative distance is set as 100m, the controller first minimized this value. Later on, it decreases the velocity smoothly to obtain zero relative velocity. This shows one of the advantages of using an MPC controller instead of classical controllers. Our two objectives (lowering speed to converge to zero relative velocity and decreasing relative distance) are contradictory. Via utilizing the flexibility that the controller gains provide, we can find a balance between these objectives. Additionally, we can increase the $\frac{Q_f}{Q}$ ratio to improve controller's tracking ability while keeping stability concerns in mind as well.

- It is worth mentioning that further tuning and quantifying these objectives are needed to fine-tune these gains for real-life usage. As an example, these gains might differ for various cases. A vehicle designed to maximize comfort might want to maintain a relatively low $\frac{Q_{2,y}}{Q_{y,3}}$ ratio to avoid sudden changes in speed while another vehicle in a high-traffic scenario might focus on fast distance tracking.

- Additionally, given that at each step, we are solving a constrained QP problem, the solution might not exist at all. In general, for close distances, the problem is likely not to have a solution. This is a significant problem that must be addressed. The simplest method would be to re-apply the same command input as previous step or increase control horizon. Other alternatives could be relaxing the constraints for certain cases or providing reference signal that interpolates smoothly to desired final state and changing the control problem into a servo-controller problem. This is often used in robot joint problems where time varying reference trajectory is given to the controller to smoothly move the joint to another location. This helps in avoiding sudden large controller effort.

- Another point worth mentioning is regarding the stability concerns of this controller. Since the state dynamics are not fully controllable, we cannot directly apply dual mode MPC controller approach. However, we can apply Kalman decomposition and find infinite gain for the fully controllable and reachable subspace of the system. Additionally, we can apply stabalizability analysis instead. These steps would be vital given the importance of having an stable ACC controller.

5 Conclusion

MPC Controllers are one of the most popular choices for multi-objective controllers. In this report, we discussed applicability of these controllers as adaptive cruise controllers (ACC). We analyzed an example ACC model and control scheme [2] and designed our controller such that it performs well for two different scenarios.

Based on our defined design specifications, we defined constraints and design specifications and tuned gain parameters accordingly. Furthermore, we tested its performance and observed its response to a given white noise disturbance. Finally, we analyzed its performance and suggested improvements that can be implemented in future work.

References

[1] J. J. Martinez and C. Canudas-de-Wit, "A Safe Longitudinal Control for Adaptive Cruise Control and Stop-and-Go Scenarios," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 2, pp. 246–258, Mar. 2007.

[2] E. Kural and B. Aksun Güvenç, "Model Predictive Adaptive Cruise Control," in *Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics*, Istanbul, Turkey, 2010, pp. 1455–1461, doi: 10.1109/ICSMC.2010.5642478.

[3] L. Luo, H. Liu, P. Li, and H. Wang, "Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel economy, safety, and car-following," *Journal of Zhejiang University-Science A*, vol. 11, no. 3, pp. 191–201, February 2010.

[4] T. Wang, "Application of Model Predictive Controller to Adaptive Cruise Control," in *Proceedings of the 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, Nanjing, China, 2022, pp. 1–6, doi: 10.1109/CVCI56766.2022.9964914.

[5] B. Aksun Güvenç and E. Kural, "Adaptive Cruise Control Simulator – A low-cost, multiple-driver-in-the-loop simulator," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 42–55, June 2006.

[6] K. S. Yi and J. T. Chung, "Nonlinear Brake Control for Vehicle CW/CA Systems," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 1, pp. 17–25, 2001, doi: 10.1109/3516.914387.