



# Design Adaptive Cruise Control

Group CONFIDENTIAL

Full Name	Student ID
Alejandro Gomez Rivas	CONFIDENTIAL
CONFIDENTIAL	CONFIDENTIAL

Lecturers: CONFIDENTIAL & CONFIDENTIAL



Eindhoven, November 16, 2025

## Abstract

This report explains how to design a controller for an Adaptive Cruise Control application in a vehicle. The Adaptive Cruise Control is developed using MATLAB and Simulink. This page provides a brief summary of the report. The details can be found in the chapters.

### Chapter 1

In the introduction, the problem statement is defined (Chapter 1.1). The controller is also discussed in the problem statement. Requirements are necessary to design a controller (Chapter 1.2). Once the requirements are clear, the controller can be designed.

### Chapter 2

The Adaptive Cruise Control consists of two parts: the plant and the controller. The plant represents the vehicle, including a defined disturbance. The transfer function of the plant must be determined. Therefore, some assumptions are made regarding the parameters (Chapter 2.1). The air resistance is described by a quadratic formula. To simplify the model, it was decided to linearize the formula for the air resistance coefficients (Chapter 2.2). The transfer function of the plant is determined using a differential equation based on Newton's second law (Chapter 2.4). This is achieved with a free-body diagram (Chapter 2.3).

### Chapter 3

After defining the transfer function, the transfer function will be implemented in Simulink (Chapter 3.1). The outcome of the transfer function is the distance and the current velocity of the following vehicle. A desired distance is computed based on the velocity. This is done using the CBR (Central Bureau for Driving Licenses) rule of thumb (Chapter 3.3.1). An input value is needed to calculate the error. The input value is the distance between the two vehicles, which in practice, will be measured with a radar. A velocity profile of the leading vehicle is implemented in MATLAB to simulate this, where the velocity is integrated into a distance (Chapter 3.2). The system requires an error to be controlled. The error is computed by subtracting the desired distance of the following vehicle from the actual distance of the leading vehicle (Chapter 3.3.2). The controller used for the Adaptive Cruise Control is a PID controller. The values of the PID controller are determined through trial-and-error analysis, as finding these values mathematically is very challenging (Chapter 3.4). The Adaptive Cruise Control operates in discrete time. The model is in continuous time, so the components need to be converted to discrete time (Chapter 3.5).

### Chapter 4

All the different submodels for the Adaptive Cruise Control are created in Chapter 3. These submodels need to be connected into one complete model (Chapter 4.1). Once the model is ready, it can be simulated. During the simulation, various important figures are plotted, which are crucial for determining the correct values for the PID controller (Chapter 4.2). The desired distance and the actual distance are compared to evaluate the results. This step ensures that the results are sufficient and meet the requirements (Chapter 4.3). A conclusion regarding the PID values is drawn after analyzing the results (Chapter 4.4).

### Chapter 5

In this chapter, the conclusion about the process is written

### Chapter 6

In this chapter, the source references are shown.

### Chapter 7

This chapter examines the key learnings and insights from the project.

### Appendix

In the appendix, it is shown how the differential equation is solved.

# Contents

- 1 Introduction 3**
  - 1.1 Problem Statement . . . . . 3
  - 1.2 Requirements . . . . . 4
  
- 2 Determining The Plant Transfer Function 5**
  - 2.1 Defining Parameters . . . . . 5
  - 2.2 Linearization Air Resistance . . . . . 5
  - 2.3 Free-Body Diagram of the Plant . . . . . 6
  - 2.4 Determining Transfer Functions of the Plant . . . . . 6
  
- 3 Creating Simulink Model of the System 7**
  - 3.1 Implementation Transfer Function of the Plant and the Disturbance in the Simulink Model 7
  - 3.2 Implementation Velocity Profile 'Leading Vehicle' . . . . . 7
  - 3.3 Implementation Desired Following Distance and Error into Simulink . . . . . 8
  - 3.4 Implementation Controller into Simulink . . . . . 9
  - 3.5 Convert Model to Discrete time domain . . . . . 10
  
- 4 Tuning the Controller of the Simulink Model 11**
  - 4.1 Final Model in Simulink . . . . . 11
  - 4.2 Results of Simulink Model . . . . . 11
  - 4.3 Conclusion Simulink Model Results . . . . . 13
  - 4.4 Conclusion Tuning Controller . . . . . 14
  
- 5 Conclusion 15**
  
- 6 Key Learnings and Insights 16**
  
- 7 References 17**
  
- A Determining Differential Equation 18**

# 1 | Introduction

This section examines the general and specific problem statement. It also provides an overview of the requirements for the Adaptive Cruise Control system.

## 1.1 | Problem Statement

### 1.1.1 | General Problem Statement

Adaptive Cruise Control (ACC) enables a convoy of vehicles to follow each other on the road while maintaining a safe inter-vehicle distance. This system uses a control structure with two layers:

1. Upper-level Controller: Determines the target velocity for each vehicle to maintain safe distances. It accounts for the status of the current vehicle (e.g., acceleration, velocity, position) and the states of other vehicles in the convoy.
2. Lower-level Controller: Executes the desired velocity by adjusting the throttle angle and brake pressure within a specified time period.

The project aims to integrate these two layers into one system by designing a PID controller that can respond to real-world scenarios, prevent overshoot, and ensure vehicles maintain safe following distances and accelerations.

### 1.1.2 | Specific Problem Statement

#### PID Controller

A PID controller is a control mechanism consisting of three components: Proportional (P), Integral (I), and Derivative (D). Together, these components enable the system to achieve and maintain the desired output by continuously adjusting its input, such as throttle or brake, in response to the error signal—the difference between the desired value (setpoint) and the actual value (output).

- Proportional Control (P): Reacts to the current error to bring the system closer to the desired value. It determines how strongly the system reacts to the error.
- Integral Control (I): Accumulates past errors to eliminate steady-state error and ensure the system reaches the desired value.
- Derivative Control (D): Predicts the future trend of the error to reduce overshoot and oscillations, ensuring a smooth response.

Proper tuning of these components ensures the system is fast, stable, and accurate.

#### State-Feedback Controller

A state-feedback controller is a control system that uses the state variables of a plant (e.g., a vehicle) to compute the control input, enabling precise control of the system's behavior. The plant's state variables—such as position, velocity, and acceleration—are measured or estimated and fed back into the controller.

In Simulink, a state-feedback controller can be implemented to model, simulate, and control the dynamics of a convoy of vehicles, ensuring stability and desired performance.

#### Discrete-Time Systems

This project assumes the entire platoon system operates in discrete time, with a sampling period of 10 milliseconds. In Simulink, transitioning from continuous time to discrete time involves converting the system from the s-domain (continuous) to the z-domain (discrete).

This means the system updates its measurements and control actions at fixed intervals, defined by the 10-millisecond sampling period.

## 1.2 | Requirements

It is crucial to establish clear requirements for an Adaptive Cruise Control (ACC) system in advance to avoid unsafe situations. To present these requirements clearly, they are compiled into a requirements package.

To develop a well-functioning ACC, certain requirements must be met. Some of these are predefined, while others will be defined during the development process to create the best possible system.

### Requirements:

- **The controller must operate in discrete time with a sampling period of 10 milliseconds (sample frequency of 100Hz).** This ensures timely and precise updates of control actions.
- **There may be a maximum overshoot in the distance of 2.5%.** Although theoretically, it is possible to maintain an exact distance, practical limitations result in a slight overshoot. Therefore, a 2.5% maximum overshoot is acceptable.
- **Acceleration must be between  $-4 \text{ m/s}^2$  and  $4 \text{ m/s}^2$ .** This defines the limits for acceleration and deceleration, ensuring the system's behavior remains within safe and manageable bounds.
- **The car must maintain a minimum following time of 2 seconds behind the vehicle in front and adjust speed accordingly.** This requirement addresses both safety and comfort. Maintaining a time-based following distance, rather than a fixed distance, ensures the distance adjusts based on the speed of the vehicles.
- **At a speed of 0, a minimum distance of 4 meters must be maintained.** This ensures that the vehicles do not get too close to each other when at a standstill. Without a minimum distance, there could be the risk of the vehicles stopping just 1 cm apart, increasing the chance of minor collisions or damage.