Bachelor's Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering

Adaptive Cruise Control response optimization with respect to riding comfort

Vít Fanta

Supervisor: Ing. Tomáš Haniš, Ph.D.
Field of study: Cybernetics and Robotics
May 2021

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Fanta Vít**   Personal ID number: **474463**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Adaptive Cruise Control response optimization with respect to riding comfort

Bachelor’s thesis title in Czech:

Optimalizace odezvy adaptivního tempomatu z hlediska komfortu posádky

Guidelines:

The passenger comfort criterion is an important aspect of the Driver Assistance Systems development process. Adaptive cruise control functionality is one of the building blocks for autonomous technology. The goal of the thesis is to address the comfort criteria for extreme situations like cut-in maneuver. The thesis will consist of the following points:

1. Get familiar with vehicle dynamics and adaptive cruise control.
2. Development of passenger comfort optimization tool for cut-in maneuver scenario.
3. Modification of adaptive cruise control algorithm to accommodate passenger comfort aspect
4. Verification of developed control system

Bibliography / sources:

[1] Rajamani R. - Vehicle Dynamics and Control, ISBN 978-1-4614-1432-2, Springer, London
[2] Robert Bosch GmbH - Bosch automotive handbook - Plochingen, Germany : Robet Bosch GmbH ; Cambridge, Mass. : Bentley Publishers
[3] Winner H. (2012) Adaptive Cruise Control. In: Eskandarian A. (eds) Handbook of Intelligent Vehicles. ISBN 978-0-85729-084-7 Springer, London.

Name and workplace of bachelor’s thesis supervisor:

Ing. Tomáš Haniš, Ph.D., Department of Control Engineering, FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **20.01.2021**   Deadline for bachelor thesis submission: **21.05.2021**

Assignment valid until:
by the end of summer semester 2021/2022

Ing. Tomáš Haniš, Ph.D. Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.
Date of assignment receipt   Student’s signature

Acknowledgements

First of all, I would like to express gratitude to my family for their great support during my studies.

I want to thank my supervisor, Ing. Tomáš Haniš, Ph.D., for weekly consultations which were immensely helpful, and to Ing. Petr Liškář for his enlightening remarks about the questionnaire and the experiment design.

Also, I would like to thank Ing. Zdeněk Houf and my fellow students Matouš Pikous and Martin Rektoris for their assistance in the Strojovna lab on numerous occasions.

Finally, special thanks go to my friend Bc. Dominik Hodan and my mum, an English teacher, for proofreading and correcting the text of this thesis.

Declaration

I declare that I have written the presented thesis on my own and that I have cited all of the used information sources in accordance with the Methodical instructions about ethical principles for writing an academic thesis.

In Prague, 21st May, 2021

Abstract

The goal of this thesis is to design a part of an adaptive cruise control system to maintain riding comfort during a cut-in maneuver. Its functionality is then verified on a set of experiments with the use of a 6-DoF motion platform and the help of volunteers.

The developed optimisation tool takes multiple parameters of the Ego vehicle – Target vehicle system into account (relative distance, relative velocity etc.) and provides a reference acceleration for Ego vehicle to follow. This reference is calculated using linear programming and takes the passengers' comfort into consideration by introducing limits on longitudinal acceleration and jerk. This approach is compared to a different control system with no comfort consideration.

The results from the experiments proved that the developed tool ensures more riding comfort during a cut-in maneuver than the referential control system. The algorithm will be available for further use during future research at the Department of Control Engineering at FEE, CTU Prague.

Keywords: Cruise control, adaptive cruise control, riding comfort optimisation, comfort evaluation, cut-in control.

Supervisor: Ing. Tomáš Haniš, Ph.D.

Abstrakt

Cílem této práce je navrhnout část adaptivního tempomatu tak, aby zachoval jízdní komfort, pokud se před daný vůz nečekaně zařadí jiné vozidlo. Funkce tohoto systému je poté ověřena několika experimenty, při kterých byla dobrovolníkům navozena tato situace na pohybové plošiny se šesti stupni volnosti.

Vstupem do navrženého optimalizačního nástroje jsou parametry popisující soustavu obou vozidel (relativní vzdálenost, relativní rychlost, atd.) a výstupem je pak referenční zrychlení pro vlastní vozidlo. Tato reference je spočtena pomocí lineárního programování a zachovává komfort pasažérů pomocí omezení podélného zrychlení a ryvu. Popsaný přístup je poté srovnán s jiným řídicím systémem, který nebyl navržen s ohledem na pohodlí pasažérů.

Výsledky získané z experimentů potvrdily, že navržený nástroj poskytuje větší míru jízdního komfortu při popsaném manévru než referenční řídicí systém. Algoritmus bude k dispozici pro další výzkum na Katedře řídicí techniky na FEL, ČVUT v Praze.

Klíčová slova: Tempomat, adaptivní tempomat, optimalizace jízdního komfortu, vyhodnocení komfortu, řízení při těsném zařazení jiného vozidla.

Překlad názvu: Optimalizace odezvy adaptivního tempomatu z hlediska komfortu posádky

Contents

Introduction **1**

1 Cruise Control Architectures **3**

1.1 Cruise Control 4

1.2 Adaptive Cruise Control 5

2 Riding Comfort Perception and Optimisation **7**

2.1 Human Riding Comfort Perception 7

2.2 Control Objectives with Respect to Comfort 8

2.3 Problem Statement 8

2.4 Finite Difference Method 9

2.5 Boundary Conditions and Constraints 10

2.6 Linear Program Formulation . . . 10

2.7 MATLAB Implementation 12

3 Verification Process Overview **15**

4 Verification Process Development **17**

4.1 MATLAB 17

4.1.1 Automated Driving Toolbox . 17

4.1.2 Acceleration regulator 19

4.1.3 Simulink 22

4.2 vJoy 22

4.3 Live for Speed 23

4.4 Motion Platform 24

5 Experiment Methodology **29**

6 Results **33**

Conclusion **37**

References **39**

Introduction

According to [1], approximately 94 % of all motor vehicle crashes in the USA are caused by driver errors. This statistic can be reduced by introducing driver assistance systems which help to relieve the driver's workload, allowing the driver to be more observant about the surroundings. One of such systems is Adaptive Cruise Control (ACC).

Besides the main functional and safety requirements of ACC (i.e. velocity and headway control), the comfort aspect is becoming significant. One of the reasons of the shift of focus is the increasing relevance of autonomous vehicles. Higher comfort of autonomous and highly automatic systems can help persuade public opinion to acknowledge advantages and convenience of such vehicles.

The comfort of passengers is a subjective notion which is commonly evaluated by professional drivers during ACC development [2]. This approach has several disadvantages: each test driver has a different subjective comfort perception, which can even vary in time. Because of these drawbacks, efforts have been made to objectify riding comfort and to predict human comfort levels based on measured motional data [2, 3]. Longitudinal acceleration and its derivative, jerk, are key factors influencing riding comfort. More specifically, high values of longitudinal jerk and longitudinal acceleration are considered uncomfortable for vehicle passengers.

This thesis describes the development of a part of an ACC focused on maintaining passenger comfort during a specific maneuver. The comfort aspect is then verified on a series of experiments with the use of a 6-DoF motion platform, MATLAB's Automated Driving Toolbox and the racing simulator Live for Speed.

In chapter 1, the architectures of cruise control and adaptive cruise control systems are introduced along with the nomenclature used throughout this thesis. The concept of $R - \dot{R}$ diagrams is also presented. Chapter 2 focuses on the design of the optimisation algorithm which takes riding comfort of passengers during the cut-in maneuver into account.

Chapter 3 provides a short overview of the verification procedure design. This design is explained in greater detail in chapter 4 including the software and hardware tools: Automated Driving Toolbox, Simulink, Live for Speed and the motion platform. The final chapters 5 and 6 present the verification method and discuss the obtained results.

Chapter 1

Cruise Control Architectures

The purpose of this chapter is to describe the architecture of cruise control (CC) and adaptive cruise control (ACC) systems. In the further text, the term Ego vehicle (or just Ego) will be used to specify the vehicle equipped with the CC or ACC system in focus. The term Target vehicle (or just Target) will refer to the vehicle directly in front of Ego vehicle (see figure 1.1). Also, a number of physical quantities will be used to describe motion of these vehicles. They are compactly presented in table 1.1.

A useful way to represent the current state of the Ego – Target system is a range – range rate ($R - \dot{R}$) diagram [4]. It is a parametric plot (with respect to time) such that the Ego – Target relative distance (R) is plotted on the horizontal axis and the Ego – Target relative velocity ($\dot{R}$) is plotted on the vertical axis. Let R be defined as the measured distance between Ego and Target and

$$\dot{R} = v_E - v_T. \tag{1.1}$$

Using this definition, $\dot{R} > 0$ when Ego vehicle is moving faster than Target vehicle, and vice-versa. The relative distance R should be strictly positive because $R = 0$ would imply a physical contact of the two vehicles. An example of the $R - \dot{R}$ diagram is shown in figure 1.2.

v_E	v_T	a_E	a_T	j_E
–	–	$\mathrm{d}v_E/\mathrm{d}t$	$\mathrm{d}v_T/\mathrm{d}t$	$\mathrm{d}a_E/\mathrm{d}t$
Ego velocity	Target velocity	Ego acceleration	Target acceleration	Ego jerk

Table 1.1: An overview of the used quantities.

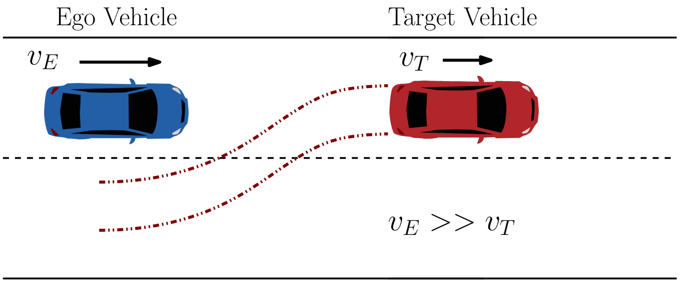


Figure 1.1: A depiction of a cut-in maneuver (the car sprite originally from [5]).

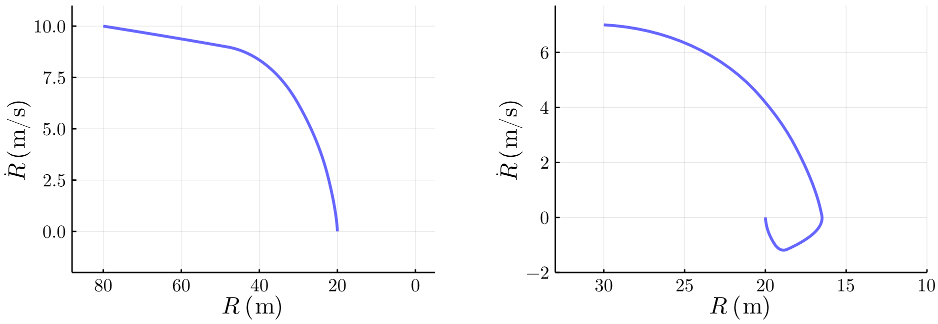


Figure 1.2: Examples of $R - \dot{R}$ diagrams describing Ego vehicle's response during cut-in maneuvers for different initial conditions. The final point of $\dot{R} = 0\,\mathrm{m/s}$, $R = 20\,\mathrm{m}$ represents no relative motion between Ego and Target while keeping the distance of 20 m.

1.1 Cruise Control

A cruise control system is able to control v_E according to the driver's setting (provided that v_E is greater than the functional minimum) [6]. This is particularly useful when driving on a motorway as the necessity to divert from the cruising speed is much less common than in e.g. urban areas. Furthermore, motorway turns are only mildly curved, which makes the task of velocity control easier.

A basic two-level architecture of a CC is presented in figure 1.3. The upper level controller calculates the acceleration needed to satisfy the reference given by the input parameters (desired velocity, current v_E etc.). The lower level controller maps the computed acceleration to actuator inputs of Ego vehicle, which typically include throttle and brake levels. The upper level controller is often of classical nature (e.g. a PI regulator) whereas the lower level controller has to take the vehicle dynamics into account and its design can be more complex [4, 7].

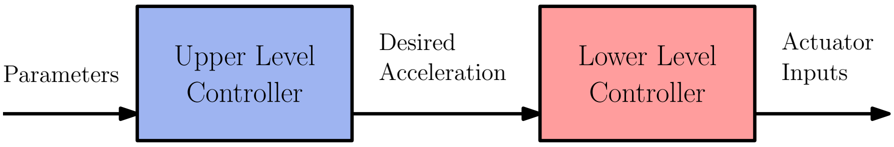


Figure 1.3: An overview of a cruise control architecture.

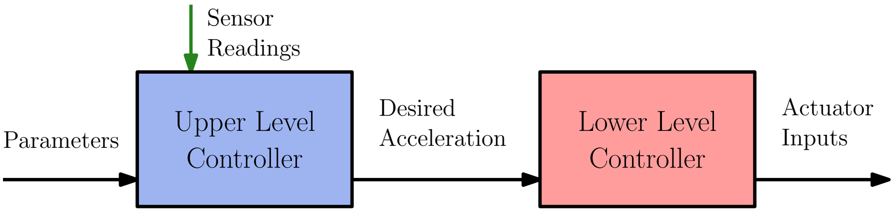


Figure 1.4: An overview of an adaptive cruise control architecture.

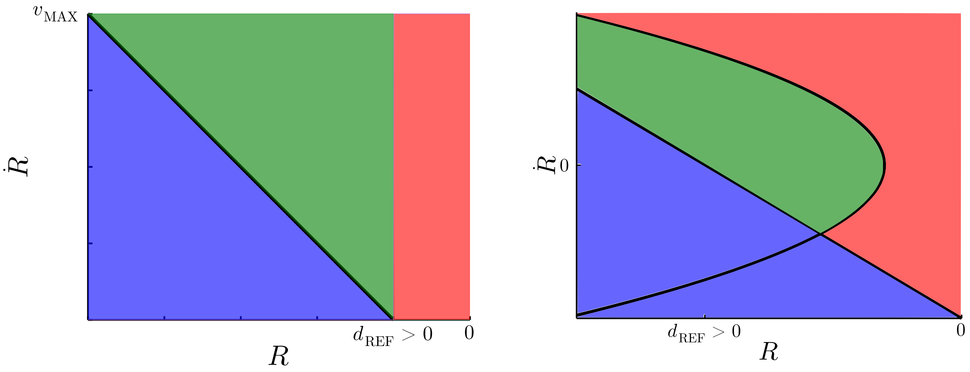
1.2 Adaptive Cruise Control

A standard cruise control system as described in the previous section has a major drawback. The driver is expected to take control during circumstances (such as a cut-in maneuver, see figure 1.1) which prevent the correct function of the CC. Adaptive cruise control systems attempt to solve this issue by utilising on-board sensors (e.g. radar or lidar) mounted on the front of Ego vehicle. This enables the system to react to the presence of Target vehicle and its velocity v_T development [8].

The control architecture of ACC is modified to use additional information provided by sensors (see figure 1.4). ACC can then reduce v_E in the case of Target vehicle being slower to prevent a crash. Moreover, the system can maintain v_E to be the same as v_T or switch to the regular cruise control in the event of Target vehicle going faster than the current Ego reference velocity set by the driver. The system must therefore contain an internal logic which can switch seamlessly between the velocity control (regular CC) and the CTG[*] control [4, 8].

The $R - \dot{R}$ diagram can be used to design a set of rules which determine which type of control strategy should ACC follow. More specifically, if the $R - \dot{R}$ region is divided into several sections which represent different control modes, the position of the current state in the $R - \dot{R}$ space determines the correct mode. Two examples of such $R - \dot{R}$ planes are presented in figures 1.5. ACC system also includes transitional controllers which are employed during ACC mode switching process. They improve passenger experience by ensuring a smooth transition between the modes [4, 8].

[*] Constant time-gap policy: used for following Target vehicle in such a way that the spacing depends linearly on the current v_E. Should multiple subsequent vehicles use the same ACC system, this policy will be more stable than the constant spacing policy [4, 7].



(a) : The division of the $R - \dot{R}$ space in [7].

(b) : The division of the $R - \dot{R}$ space in [4].

Figure 1.5: Different ACC $R - \dot{R}$ control strategy selection approaches. The blue area represents a velocity control (regular CC), the green represents the area of spacing control and the red represents the area of emergency braking, which is usually not a part of ACC. The straight lines represent a linear relation between R and $\dot{R}$, the parabola is a curve of constant deceleration. The positive direction of the R axis is to the left [4, 6].

There are numerous functional requirements to which a commercial ACC must adhere. The requirements depend on the current mode of ACC and include minimal swinging, longitudinal jerk during regular CC, damping of swinging when following Target vehicle and also transparent maneuvering for easier assessment by the driver during all circumstances [8]. More detailed functional and performance requirements are stated in the currently valid standard issued by International Organization for Standardization [9].

This bachelor thesis focuses on the design of a transitional controller which is responsible for an adequate reaction during a cut-in maneuver. Referring to figures 1.5, the situation corresponds to a sudden change from the regular CC (no Target vehicle is registered) to the green area representing the spacing control. Later chapters are dedicated to the design of a transitional controller which ensures smoother transition to the spacing control.

Chapter 2

Riding Comfort Perception and Optimisation

2.1 Human Riding Comfort Perception

The purpose of this section is to give a brief overview of how a person perceives comfort and discomfort on board of a motorised vehicle. Human beings possess a pair of mechanical sensory organs capable of sensing angular and linear acceleration: two vestibular apparatuses are a part of the inner ear. Each of them contains three ring-like structures which can perceive angular accelerations in their respective axes. Linear accelerations (vertical and horizontal) are sensed with the use of two structures: utricle and saccule. Velocity can be sensed only via a visual sensory input [10].

Riding comfort perception is a subjective evaluation of the biological sensory inputs by an individual. Although the personal experience of a particular ride can differ, there are general gauges which can objectify the comfort assessment to a certain extent. The functional requirements of an ACC state in [8] introduce upper and lower bounds on admissible values of Ego acceleration and Ego jerk to ensure more comfort. Intuitively, these restrictions are justified by the fact that a human can perceive acceleration (and also its derivatives) and by experience with driving and being aboard public transport vehicles. A recent experimental study showed that there exists a positive correlation between the riding comfort perception and longitudinal acceleration and its derivative (jerk). Large values (positive and negative) of longitudinal acceleration and jerk were found to induce discomfort among passengers [3].

2.2 Control Objectives with Respect to Comfort

In the previous section, it has been established that acceleration and jerk are quantities related to human riding comfort perception. More specifically, large magnitudes (positive and negative) are one of the key actors of inducing discomfort. In order to achieve better overall passenger experience, especially during a cut-in situation, an algorithm has been devised as a key part of this thesis. The description of the algorithm is the topic of this chapter.

In the cut-in situation, for which the algorithm is designed, another vehicle, Target vehicle, changes a lane not far from the front of Ego vehicle and is registered by Ego vehicle's on-board radar system (see figure 1.1). This maneuver disrupts the regular CC function as Target vehicle usually has a lower longitudinal velocity than the Ego vehicle, i.e.

$$v_E > v_T. \tag{2.1}$$

Real life examples of the setting include inconsiderate lane changes of a slower vehicle during joining a motorway.

The algorithm should take multiple initial parameters along with the desired final state into account. The parameters include measured quantities of the Target – Ego system, such as relative distance, Ego velocity and Target velocity. The final state can be described by the desired relative acceleration and the relative distance between the vehicles. The outputs computed by the algorithm should then provide a reference for the Ego vehicle's CC to follow. The vehicles in the final state should have the same velocity (zero relative velocity) and be a set distance apart.

2.3 Problem Statement

To formulate an optimisation problem, an objective function f which is to be minimised is needed. In this case, f will be a function of Ego vehicle acceleration and jerk because of the reasons stated previously. At first, it is not obvious how to choose f, but after a series of tests, it was found that the following function provided satisfactory results:

$$f\left(a(t), j(t)\right) = \max_t |j(t)| + \int\limits_0^t |j(t')|\mathrm{d}t' + \frac{1}{t}\int\limits_0^t |a(t')|\mathrm{d}t' = j_\mathrm{M} + \Sigma_j + \Sigma_a, \tag{2.2}$$

where $j(t)$ is Ego jerk and $a(t)$ is Ego acceleration. Quantities j_{M} (maximum absolute value of Ego jerk), Σ_j (area under the absolute value of Ego jerk) and Σ_a (area under the absolute value of Ego acceleration, divided by the time horizon to assure SI unit compatibility) are defined respectively by (2.2) for clarity.

Because the function (2.2) will be minimised with the use of the finite difference method (see the next section), the final objective function can be rewritten* from (2.2) as

$$F(\mathbf{a}, \mathbf{j}) = \max_{n=1}^{N} |\mathbf{j}(n)| + \sum_{n=1}^{N} |\mathbf{j}(n)| + T_S^{-1} \sum_{n=1}^{N} |\mathbf{a}(n)|. \tag{2.3}$$

The definition of the function F contains new parameters: T_S is the sampling period and N is the chosen resolution of a time interval $[0, T]$, $T = N \cdot T_S$. The time horizon T and the resolution N are parameters of the function F.

2.4 Finite Difference Method

Because the objective function F (2.3) is defined in terms of discrete timesteps, it is necessary to use a suitable discrete approximation to the continuous derivative. In total, two different approximations are used: the central difference and the forward difference.

The central difference of a vector $\mathbf{v} \in \mathbb{R}^n$ is defined as

$$\Delta_{\mathrm{C}} \mathbf{v}(n) = \frac{\mathbf{v}(n+1) - \mathbf{v}(n-1)}{2T_S}, \tag{2.4}$$

where T_S is the sampling period. The main advantage of using the central difference as the approximation of the first derivative is that its error is second order accurate. More formally:

$$\Delta_{\mathrm{C}} \mathbf{v} - \frac{\mathrm{d}v}{\mathrm{d}t} = O(T_S^2). \tag{2.5}$$

The central difference (2.4) has the undesirable property of being equal to zero, even though the values of a signal differ at subsequent timesteps. This can be circumvented by using the forward difference, defined as

$$\Delta_{\mathrm{F}} \mathbf{v}(n) = \frac{\mathbf{v}(n+1) - \mathbf{v}(n)}{T_S}, \tag{2.6}$$

*Continuous functions are replaced with vectors, whose elements represent values at the respective timestamps. Also, MATLAB notation for vectors is used, so vector elements are indexed with 1-based indices in parentheses.

at the cost of the error now being of the first order of T_S. Higher order differences can be obtained by repeated applications of definitions (2.4) and (2.6).

2.5 Boundary Conditions and Constraints

The algorithm requires a number of constraints and initial/final conditions in order to find a suitable solution. These are presented in this section.

Target vehicle is assumed to be in a longitudinal motion with constant acceleration:

$$v_T(t) = v_T^0 + t \cdot a_T. \tag{2.7}$$

To partially circumvent this assumption, it is possible to rerun the algorithm multiple times with different parameters to account for the varying Target vehicle acceleration. The initial Target velocity v_T^0 and the Target acceleration a_T are the first of required parameters. Similarly, initial conditions for Ego vehicle are also needed. Apart from the initial Ego acceleration a_E^0, the remaining initial parameters are specified in relative quantities: the initial relative velocity v_R^0 and the initial relative distance d_R^0. The relative parametrization is more convenient as the on-board radar directly measures relative quantities.

The remaining constraints include limitations of Ego vehicle's capabilities (the maximum acceleration a_E^{MAX} and the maximum deceleration a_E^{MIN}) and the final desired relative distance d_{REF}. The minimal relative distance parameter d_R^{MIN} to ensure safety can also be included. However, it is possible that under some initial conditions (high initial relative velocity and close initial proximity of the vehicles) the algorithm does not find a solution and it is necessary to relax the requirements.

All of the required parameters, including the ones from previous sections, are presented in table 2.1.

2.6 Linear Program Formulation

Linear programming is used to minimise function F (2.3). This is possible because of the discretization of the problem in time, and thus the dynamical relationships between variables, boundary conditions and other constraints

Symbol	Meaning
T	Time horizon to finish the maneuver.
N	Resolution of the time interval $[0, T]$; used for discretization.
v_T^0	Initial Target vehicle velocity.
a_T	Target vehicle acceleration; assumed to be constant for $t \in [0, T]$.
a_E^0	Initial Ego vehicle acceleration.
a_E^{MAX}	Maximum Ego acceleration.
a_E^{MIN}	Maximum Ego deceleration.
v_R^0	Initial relative velocity.
d_R^0	Initial relative distance.
d_{REF}	Reference relative distance; final spacing between Ego and Target.
d_R^{MIN}	Minimal relative distance; for safety, might be relaxed.

Table 2.1: Parameters required by the algorithm.

can be rewritten as matrix linear equations. This section describes how to perform such manipulations.

It is possible to solve minimisation problems defined in the ∞-norm of the form

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_\infty = \min_{\mathbf{x}\in\mathbb{R}^n} \max_{k=1}^{m} |\mathbf{a}_k^T \mathbf{x} - b_k|, \tag{2.8}$$

where $\mathbf{b} \in \mathbb{R}^m$ is a given vector, $\mathbf{A} \in \mathbb{R}^{n\times m}$ is a given matrix and $\mathbf{a}_k$ are its rows. The problem (2.8) can be rewritten[†] as

$$\min_{z\in\mathbb{R}} z \quad \text{under conditions} \quad \mathbf{x} \in \mathbb{R}^n, -z\mathbf{1} \leq \mathbf{A}\mathbf{x} - \mathbf{b} \leq z\mathbf{1}. \tag{2.9}$$

The two vector inequalities in (2.9) can be stated as a single inequality, which is later required by the `linprog` function:

$$-z\mathbf{1} \leq \mathbf{A}\mathbf{x} - \mathbf{b} \leq z\mathbf{1} \quad \text{is equivalent to} \quad \begin{bmatrix} -\mathbf{A} & -\mathbf{1} \\ \mathbf{A} & -\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ z \end{bmatrix} \leq \begin{bmatrix} -\mathbf{b} \\ \mathbf{b} \end{bmatrix}. \tag{2.10}$$

In addition to the ∞-norm minimisation, 1-norm minimisation

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1 = \min_{\mathbf{x}\in\mathbb{R}^n} \sum_{k=1}^{m} |\mathbf{a}_k^T \mathbf{x} - b_k| \tag{2.11}$$

is also expressible as a linear programming problem:

$$\min_{\mathbf{z}\in\mathbb{R}^m} \mathbf{z}^T\mathbf{1} \quad \text{under conditions} \quad \mathbf{x} \in \mathbb{R}^n, -\mathbf{z} \leq \mathbf{A}\mathbf{x} - \mathbf{b} \leq \mathbf{z}. \tag{2.12}$$

[†] The symbol $\mathbf{1}$ represents a vector of ones of the appropriate dimension.

As in the previous case, the two vector inequalities in (2.12) can be stated as a single inequality:

$$-\mathbf{z} \leq \mathbf{Ax} - \mathbf{b} \leq \mathbf{z} \quad \text{is equivalent to} \quad \begin{bmatrix} -\mathbf{A} & -\mathbf{I} \\ \mathbf{A} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \leq \begin{bmatrix} -\mathbf{b} \\ \mathbf{b} \end{bmatrix}, \tag{2.13}$$

where $\mathbf{I} \in \mathbb{R}^{m \times m}$ is the identity matrix. It is worth noting that this reduction of an inequality is possible thanks to introducing new variables which are not present in the original problem.

Discrete approximations of the derivative (formulas (2.4) and (2.6)) can be used to relate two vectors with a matrix equation. Under the assumption that vector $\mathbf{a} \in \mathbb{R}^{n-2}$ is defined as the central difference of vector $\mathbf{v} \in \mathbb{R}^n$, it is possible to express its components as

$$\mathbf{a} = \frac{1}{2T_S} \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & & 0 & -1 & 0 & 1 \end{bmatrix} \mathbf{v}. \tag{2.14}$$

Similarly, the forward difference $\mathbf{a} \in \mathbb{R}^{n-1}$ of a vector $\mathbf{v} \in \mathbb{R}^n$ can be written in the matrix form

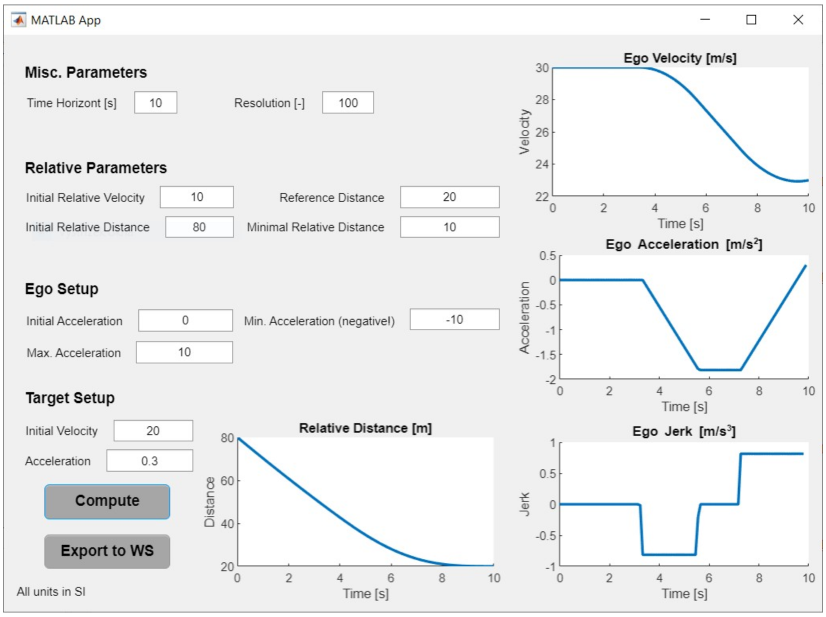
$$\mathbf{a} = \frac{1}{T_S} \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & \cdots & & 0 & -1 & 1 \end{bmatrix} \mathbf{v}. \tag{2.15}$$

The second difference can be obtained using (2.14) and (2.15) repeatedly. As an example, here is the second central difference $\mathbf{j} \in \mathbb{R}^{n-2}$ of vector $\mathbf{v} \in \mathbb{R}^n$:

$$\mathbf{j} = \frac{1}{T_S^2} \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & & 0 & 1 & -2 & 1 \end{bmatrix} \mathbf{v}. \tag{2.16}$$

2.7 MATLAB Implementation

The algorithm which takes aforementioned requirements and approaches into account has been comprised into a MATLAB application. Its GUI is presented in figure 2.1.



MATLAB App

Misc. Parameters

Time Horizont [s]: 10

Resolution [-]: 100

Relative Parameters

Initial Relative Velocity: 10

Reference Distance: 20

Initial Relative Distance: 80

Minimal Relative Distance: 10

Ego Setup

Initial Acceleration: 0

Min. Acceleration (negative!): -10

Max. Acceleration: 10

Target Setup

Initial Velocity: 20

Acceleration: 0.3

All units in SI

Figure 2.1: The graphical user interface of the MATLAB application. The parameters can be specified (in SI units) on the left side; the results are presented in the plots on the right. The optimisation can be rerun with the `Compute` button; the v_E and a_E timeseries can be exported to MATLAB's workspace with the `Export to WS` button.

The core optimisation is handled by MATLAB's `linprog` function, which is capable of solving the following optimisation task:

$$\min \mathbf{f}^T\mathbf{x} \quad \text{under conditions} \quad \begin{array}{c} \mathbf{x} \in \mathbb{R}^n, \mathbf{A}_{=}\mathbf{x} = \mathbf{b}_{=}, \\ \mathbf{A}_{\leq}\mathbf{x} \leq \mathbf{b}_{\leq}, \\ \mathbf{u} \leq \mathbf{x} \leq \mathbf{v}. \end{array} \tag{2.17}$$

The variable $\mathbf{x}$ has the following structure:

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_R \\ \mathbf{d}_R \\ z \\ \mathbf{z}_A \\ \mathbf{z}_J \end{bmatrix}, \tag{2.18}$$

where $\mathbf{v}_R$ is the relative velocity vector, $\mathbf{d}_R$ is the relative distance vector, z is the maximum value of Ego jerk, $\mathbf{z}_A$ is the vector of unsigned areas under Ego acceleration and $\mathbf{z}_J$ is the vector of unsigned areas under Ego jerk.

The matrices $\mathbf{A}_{=}$, $\mathbf{A}_{\leq}$ and the bounds $\mathbf{u}$, $\mathbf{v}$ in (2.17) are used to enforce the requirements described in table 2.1 and to describe relationships between the variables themselves. The vector $\mathbf{v}_R$ is obtained as the first forward difference of the vector $\mathbf{d}_R$ with the use of (2.15). The remaining variables z, $\mathbf{z}_A$ and $\mathbf{z}_J$ are obtained by the repeated use of (2.15) and (2.10) or (2.13) respectively.

Once the `linprog` function computes the resulting vector $\mathbf{x}$, the $\mathbf{v}_R$ part can be used as a reference for a CC system after a preprocessing phase to obtain $\mathbf{v}_E$ or $\mathbf{a}_E$. An example output for the default parameter values (visible in figure 2.1) is shown in figure 2.2. The optimisation should be performed repeatedly to compensate for probable changes in the conditions during the maneuver (e.g. variable acceleration of Target vehicle).

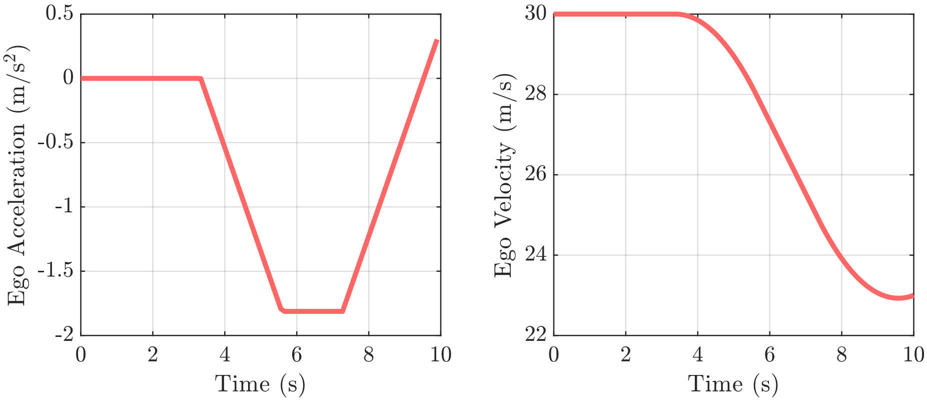


Figure 2.2: An example output of the MATLAB application for the default parameter values. The corresponding $R - \dot{R}$ diagram is shown in figure 1.2 on the left.

Chapter 3

Verification Process Overview

One of the objectives of the thesis is to verify the passenger comfort improving aspect of the algorithm described in the previous chapters. This section describes how this goal is fulfilled.

The devised algorithm outputs reference Ego acceleration which is to be physically tested and compared against the acceleration reference provided by ACC in [7]. The physical testing is handled using a motion platform. To prove the fact that reference accelerations are realisable, a closed loop system is created consisting of an acceleration controller and the software racing simulator Live for Speed (LFS) [11]. The complete verification process layout is presented in figure 3.1.

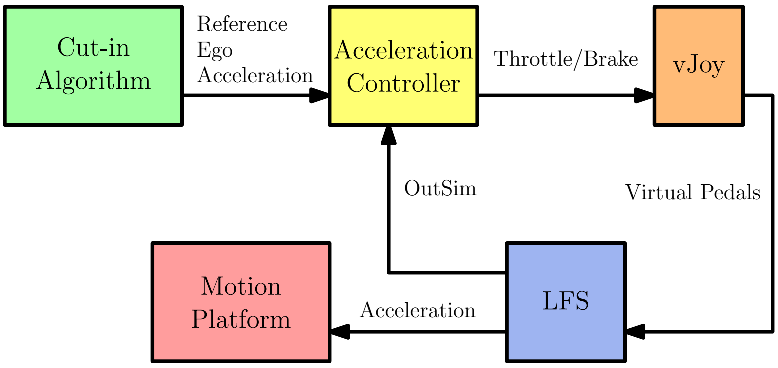


Figure 3.1: The layout of the reference acceleration → motion platform loop.

The reference acceleration is fed to LFS via the acceleration controller. The responses from LFS are recorded and later fed to the motion platform. The motion platform movements are synchronised with a visualisation shown on a screen inside of the motion platform. The visualisation is provided by MATLAB and simulates the front view of Ego vehicle (see figure 3.2). Together, they create a visual and kinetic illusion of a vehicle movement.

A total number of four scenarios of a cut-in maneuver have been prepared in MATLAB's Driving Scenario Designer, two of which use the acceleration calculated using the algorithm described earlier as the reference, and the remaining two use the acceleration reference provided by ACC in [7]. Participants of verification experiments have filled out a questionnaire after each "ride". Results from the questionnaire provide evidence for objective comparison of the riding comfort criterion between the two algorithms.

Figure 3.2: An example of the simulation front view.

Chapter 4

Verification Process Development

This chapter describes individual software and hardware parts during the verification process. These are also presented in figure 3.1 with the exception of MATLAB.

4.1 MATLAB

4.1.1 Automated Driving Toolbox

MATLAB's Automated Driving Toolbox contains means for designing and implementing various algorithms for smart vehicle solutions [12]. More specifically, Driving Scenario Designer app has been used to create the experimental scenarios. Finished scenarios can then be run either in the so-called cuboid environment (vehicles and pedestrians are represented by simple stereometric shapes like cuboids and cylinders) or in the Unreal Engine virtual environment. The latter option has the advantage of a more realistic visual presentation, and has been therefore used during the experiments.

Scenarios are used to show vehicle movement along a set path in a set scene. A scene can be created manually by adding individual roads to form a desired road grid. This is useful for simple tests, but when such a scene is rendered with the Unreal Engine, the resulting impression is very bare as there are only cars on several roads on an infinite grassy plane. To solve this issue, the toolbox comes pre-equipped with a number of 3D scenes which contain more sophisticated structures like buildings and intersections with traffic

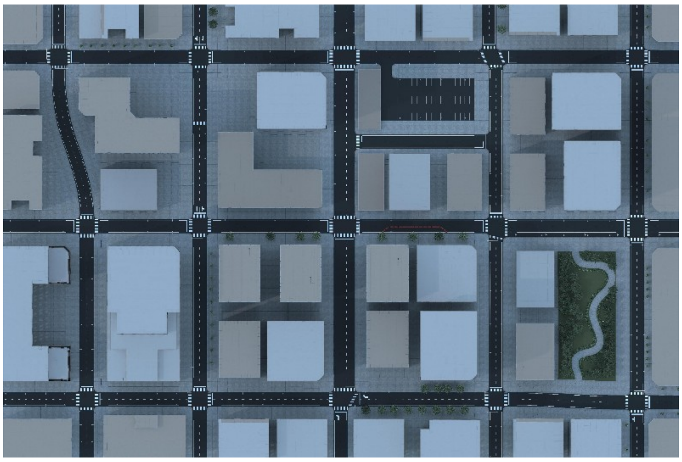


Figure 4.1: A bird's eye view of the *US City Block* 3D scene.

lights. The *US City Block* scene have been selected to host the experimental scenarios. A bird's eye view of the entire scene is presented in figure 4.1.

Car models can be inserted into simulation with the help of Driving Scenario Designer. Their movements can then be customised by placing a series of waypoints on the scene. For each of the waypoints, velocity, as well as waiting time at that waypoint can be specified. This is useful for stopping cars at traffic lights in the simulation. Velocity between waypoints is linearly interpolated for each segment. The trajectory is computed by the app as a piecewise clothoid curve [13]. Because Driving Scenario Designer cannot visualise a 3D scene layout, an intermediary road grid has been created to ease the scenario design process. Individual waypoints are placed onto this grid to construct the scenarios. Besides Ego and Target vehicles, additional car models have been added to prevent respondents' fixation to Target vehicle. Grids with waypoints of the four created scenarios are presented in figures 4.6.

The key difference between the scenarios are the parameters regarding the cut-in maneuver. These include initial Ego velocity v_E^0, initial Target velocity v_T^0, Target acceleration a_T, initial relative distance d_R^0 and initial relative velocity v_R^0. Their values are shown in table 4.1. Scenarios CO-hi and R-hi have more challenging initial condition (higher initial relative velocity) compared to scenarios CO-lo and R-lo. During the cut-in maneuver, Scenarios CO-hi and CO-lo are controlled by the Cut-in Optimised algorithm developed for this thesis, while scenarios R-hi and R-lo are controlled by the Reference ACC system from [7] (hence the naming).

Scenario \ Parameter	v_E^0 (m/s)	v_T^0 (m/s)	a_T (m/s^2)	d_R^0 (m)	v_R^0 (m/s)
CO-hi, R-hi	14	8	0.3	22	6
CO-lo, R-lo	11	9	0.1	20	2

Table 4.1: Parameters of the created scenarios.

4.1.2 Acceleration regulator

According to ACC structure in figure 1.4, the lower level controller adjusts the actuator inputs according to the reference acceleration from the upper level controller. A combination of a feedforward (FF) and a proportional integral (PI) control is used. The PI part has been tuned iteratively, while the FF component has been designed to approximately invert the transfer function from the throttle/brake ($u_{\text{throttle}}/u_{\text{brake}}$) inputs to vehicle acceleration. This transfer function is assumed to be linear[*]:

$$a^+ = K_T \cdot u_{\text{throttle}} + c_1, \quad u_{\text{throttle}} \in [0,1], \tag{4.1a}$$

$$-a^- = K_B \cdot u_{\text{brake}} + c_2, \quad u_{\text{brake}} \in [0,1], \tag{4.1b}$$

where $a^+ = \max(a_E, 0)$ is the positive part of Ego acceleration and $a^- = \max(-a_E, 0)$ is the negative part of Ego acceleration. The constants K_T, K_B, c_1 and c_2 have been identified by the least squares interpolation of points sampled from the appropriate characteristics for the second gear using Live for Speed and vJoy (these are described further in the text). The final parameter values are shown in table 4.2. The sampled points along with the interpolated lines are shown in figure 4.2.

Constant	K_T	K_B	c_1	c_2
Value	$4.04\,\mathrm{m\,s^{-2}}$	$-11.42\,\mathrm{m\,s^{-2}}$	$-0.7505\,\mathrm{m\,s^{-2}}$	$0.3751\,\mathrm{m\,s^{-2}}$

Table 4.2: Identified values for the parameters from (4.1).

After the parameter identification, the feedforward term was obtained by rearranging the equations (4.1) to find throttle/break levels based on the input acceleration. The yielded relations are

$$u_{\text{throttle}} = \begin{cases} K_T^{-1} a, & a \geq 0\,\mathrm{m\,s^{-2}}, \\ 0, & a < 0\,\mathrm{m\,s^{-2}}, \end{cases} \tag{4.2a}$$

$$u_{\text{brake}} = \begin{cases} 0, & a \geq 0\,\mathrm{m\,s^{-2}}, \\ K_B^{-1} a, & a < 0\,\mathrm{m\,s^{-2}}. \end{cases} \tag{4.2b}$$

[*] This is valid under the conditions of no gear changes, small velocity changes and no slip.

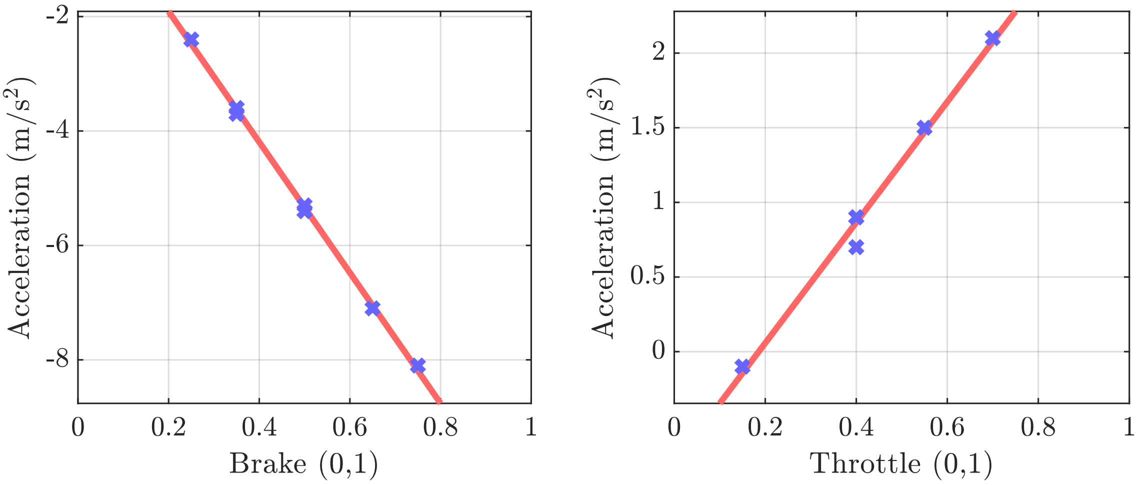


Figure 4.2: The measured points of the pedal characteristics along with the interpolated lines.

The offset terms c_1 and c_2 have been omitted in (4.2) to avoid discontinuities, which would worsen the regulator responses.

The PI part of the acceleration controller have been tuned manually. Its transfer function is

$$C(s) = \frac{0.06s + 1.5}{s}. \tag{4.3}$$

The feedback controller's effort e is limited to the interval $[-1, 1]$ and is mapped to the throttle and brake inputs:

$$u_{\text{throttle}} = \begin{cases} e, & e \geq 0, \\ 0, & e < 0, \end{cases} \tag{4.4a}$$

$$u_{\text{brake}} = \begin{cases} 0, & e \geq 0, \\ -e, & e < 0. \end{cases} \tag{4.4b}$$

The effort e is symmetric with respect to the input acceleration which contrasts with the feedforward approach. During the design process, e was also scaled. The u_{throttle} output was scaled by the factor of $-K_B/K_T$, but it resulted in greater oscillations in responses. Therefore, the feedback regulator has been left in its symmetric form.

The final form of the acceleration regulator is presented in figure 4.3. The acceleration references and the acceleration regulator/LFS responses are plotted in figures 4.4 and 4.5. The worse behaviour in the neighbourhood of $0\,\mathrm{m/s^2}$ is caused by the friction and other dynamics which the feedforward controller fails to capture. ACC from [7] provides controller effort which is an order of magnitude higher than the effort the developed algorithm provides. This fact results in suboptimal acceleration control as the car wheels lock up during maximum braking, which violates the assumptions under which the acceleration controller is designed: no gear changes, small velocity changes and no slip.

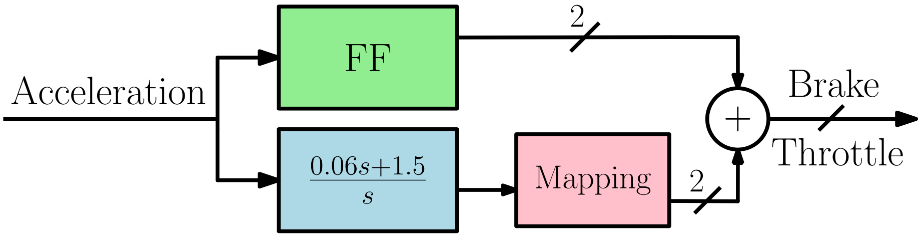


Figure 4.3: The layout of the acceleration regulator. The mapping refers to equations (4.4). The marked signals have two components (brake and throttle).

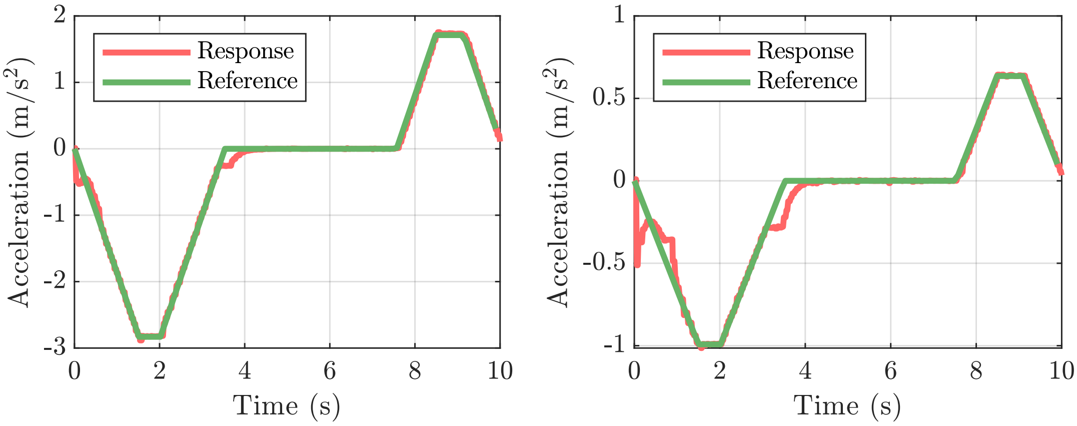


Figure 4.4: The acceleration reference and the LFS response for the conditions of scenario CO-hi (on the left) and scenario R-hi (on the right).

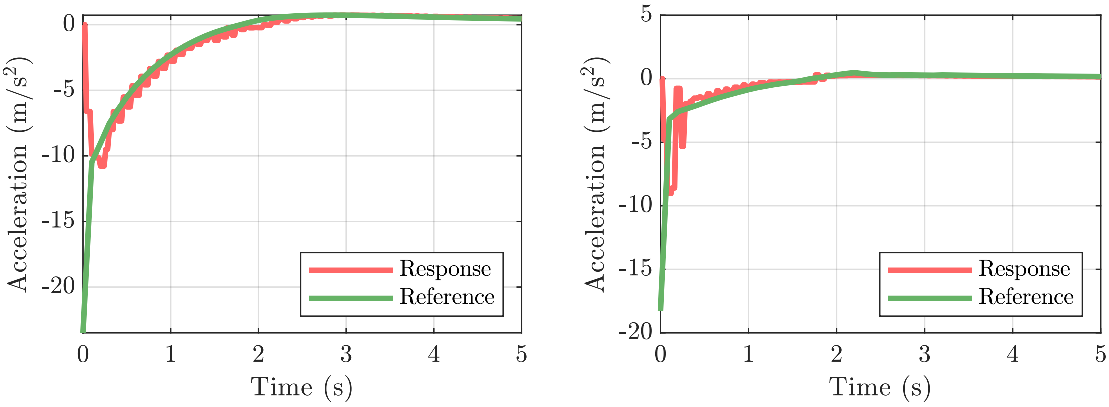


Figure 4.5: The acceleration reference and the LFS response for the conditions of scenario R-hi (on the left) and scenario R-lo (on the right).

4.1.3 Simulink

MathWorks' Simulink is used to read the scenario files and to send motion packets to the platform. Because the acceleration which is to be fed to the motion platform (see figure 3.1) is only valid for 10 seconds at most (the time horizon parameter of the algorithm), it is computed in a different way for the rest of the scenarios' run time.

The scenario files contain information about movements of the simulated vehicles: the velocity vector, the angular velocity vector and the orientation angles about the principal axes. The ω_z component of the angular velocity vector can be used to approximate lateral acceleration:

$$a_{\text{lat}} \approx R\omega_z^2, \tag{4.5}$$

where R is the radius of curvature. The value of R has been later specified to 1.4 m after testing simulations with the motion platform. The longitudinal acceleration is approximated with washout filtering of the v_x velocity component from the scenario file. A washout filter is employed to prevent sudden changes in longitudinal acceleration. Its transfer function is

$$G(s) = \frac{s}{0.7s + 1}. \tag{4.6}$$

Simulink files are included in the attachment to this thesis.

4.2 vJoy

The vJoy driver is able to mimic the presence of a joystick-like device to the operating system. For the purpose of this thesis, the mimicked hardware are pedals and a steering wheel. vJoy is used as an intermediary between the output of the acceleration regulator (in Simulink) and the Live for Speed simulator.

The driver can be downloaded from the website [14]. It also comes with several applications to configure virtual devices, monitor their current state and, most importantly, feed the virtual devices with a custom input. The vJoy SDK contains interface for C/C++ and C# programs to interact with vJoy devices. While these SDK tools are used in [15, 16], a different approach has been chosen for this thesis. A Python vJoy interface from the repository [17] is used for a more lightweight implementation. The Python script receives a UDP packet with three floats (throttle, brake and heading levels). The virtual axes of pedals and the steering wheel are then set according to these values. The whole framework is presented in figure 4.7.

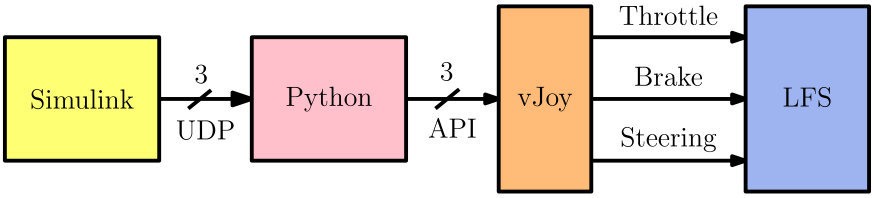


Figure 4.7: A diagram of the vJoy framework. The throttle signal has been kept at 0.

Figure 4.8: A screenshot from Live for Speed. The acceleration responses were captured during rides on the visible straight track.

4.3 Live for Speed

Live for Speed is a software racing simulator, steadily developed since 2002. Its advanced vehicle model and the ability to access the current state of the simulation make it convenient software to utilise for the task. Also, it has been previously used in [15, 16] with success.

The LFS's configuration file `cfg.txt` can be modified to periodically send UDP packets containing the simulation state to a specified IP address. The structure of the packets can be customised to include only parts of the complete simulation state. For the needs of this thesis, the following quantities are extracted from the simulation: position†, velocity†, angular velocity, acceleration† and angles of rotation with respect to principal axes (roll, pitch and yaw).

†Specified in global coordinates of the current simulation map.

To capture simulated acceleration responses, the simulated vehicle first has to adhere to the initial conditions of the algorithm (initial Ego velocity). To reach this state, a separate simulation has been created: a PI velocity controller active during the first 12 s of the simulation before the acceleration controller takes over. The acceleration responses are recorded and later fed to the motion platform at the appropriate moment of the cut-in maneuver. For the duration of the capture simulation, the LFS vehicle has been in the second gear to match the conditions under which the acceleration controller is designed. A screenshot taken during the acceleration capture process is shown in figure 4.8.

4.4 Motion Platform

A 6-degree-of-freedom (DoF) motion platform has been used for the purposes of the simulation of vehicle dynamics. It has originally served as a multi-functional flight simulator [18], however, its motion can be controlled by a customised program.

The motion is controlled directly from Simulink with the use of the Simulink Real-Time add-on. Every 200 ms, a UDP (User Datagram Protocol) packet containing a total of nine control signals is sent to the platform. These signals include roll, pitch and yaw orientation angles, Cartesian coordinates of the acceleration vector and Cartesian coordinates of the angular velocity vector, all of which are floating-point numbers. The UDP packets are then processed by the internal controller of the platform [16].

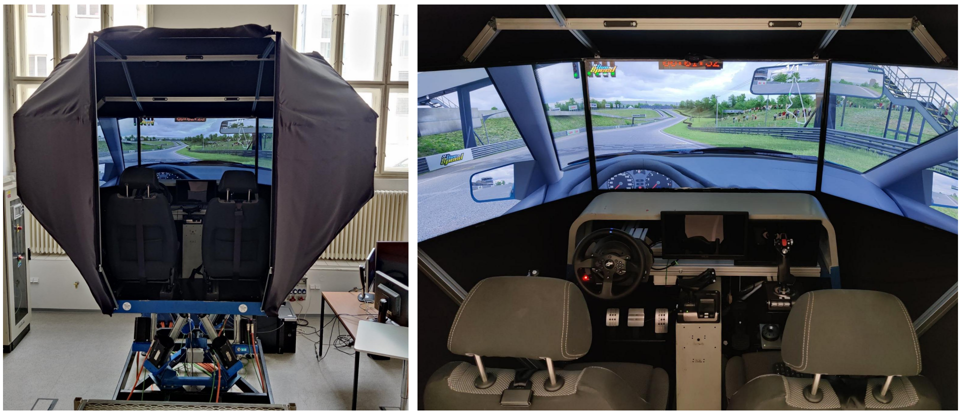


Figure 4.9: The 6-DoF motion platform used for the algorithm verification.

In order to maintain the safety of passengers, limits on the control inputs have been introduced. Because only the longitudinal dynamics are of interest, the angular velocity inputs and orientation angle inputs are scaled down and clipped as in [15]. The acceleration inputs are not scaled down, but are limited to $\pm 3\,\mathrm{m\,s^{-2}}$. These restrictions were taken into account during the scenario design; computed Ego acceleration reference which would exceed the mentioned limit would not be simulated properly and the yielded results would be inconclusive. The acceleration restriction did not limit the choice of initial parameters for the design of the scenarios as more extreme situations do not usually occur in a real city-like environment, which the visualisation is supposed to mimic. The overview of the scaling factors and clipping values is presented in table 4.3. Finally, the x and y components of the linear acceleration are swapped to accommodate the platform's coordinate system.

Quantity / **Scale/clip**	Angular vel.	Linear accel.	Orientation
Scaling factor	0.1	1	1
Clipping value	$2\,°/\mathrm{s}$	$3\,\mathrm{m\,s^{-2}}$	25°

Table 4.3: Scaling.

To capture the actual longitudinal acceleration simulated by the motion platform, an integrated smartphone accelerator has been utilised. The captured responses are shown in figures 4.10 and 4.11. The reference is not precisely followed because of the time delay and the response time of the platform. It is also possible that even though the capture was triggered remotely from Simulink at the start of the simulation, the synchronisation of the signals is not optimal. However, the on-board sensation is very distinguishable for each of the scenarios.

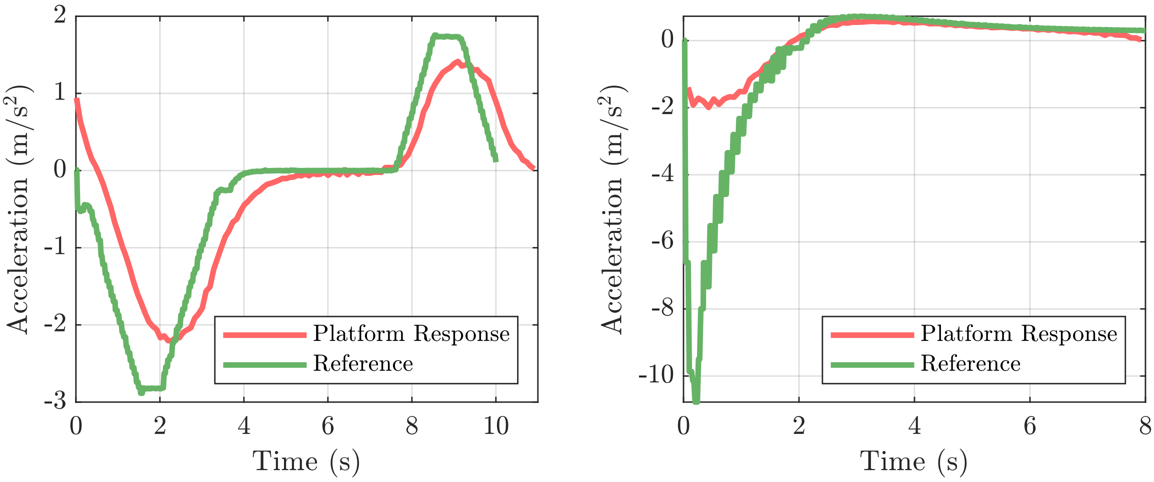


Figure 4.10: Recorded motion platform responses and their references for scenario CO-hi (on the left) and scenario R-hi (on the right).

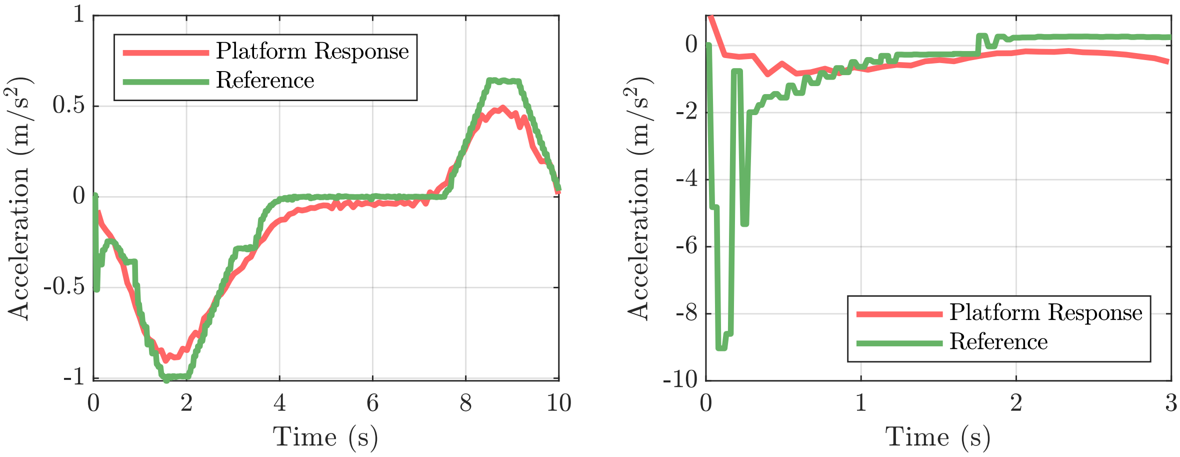
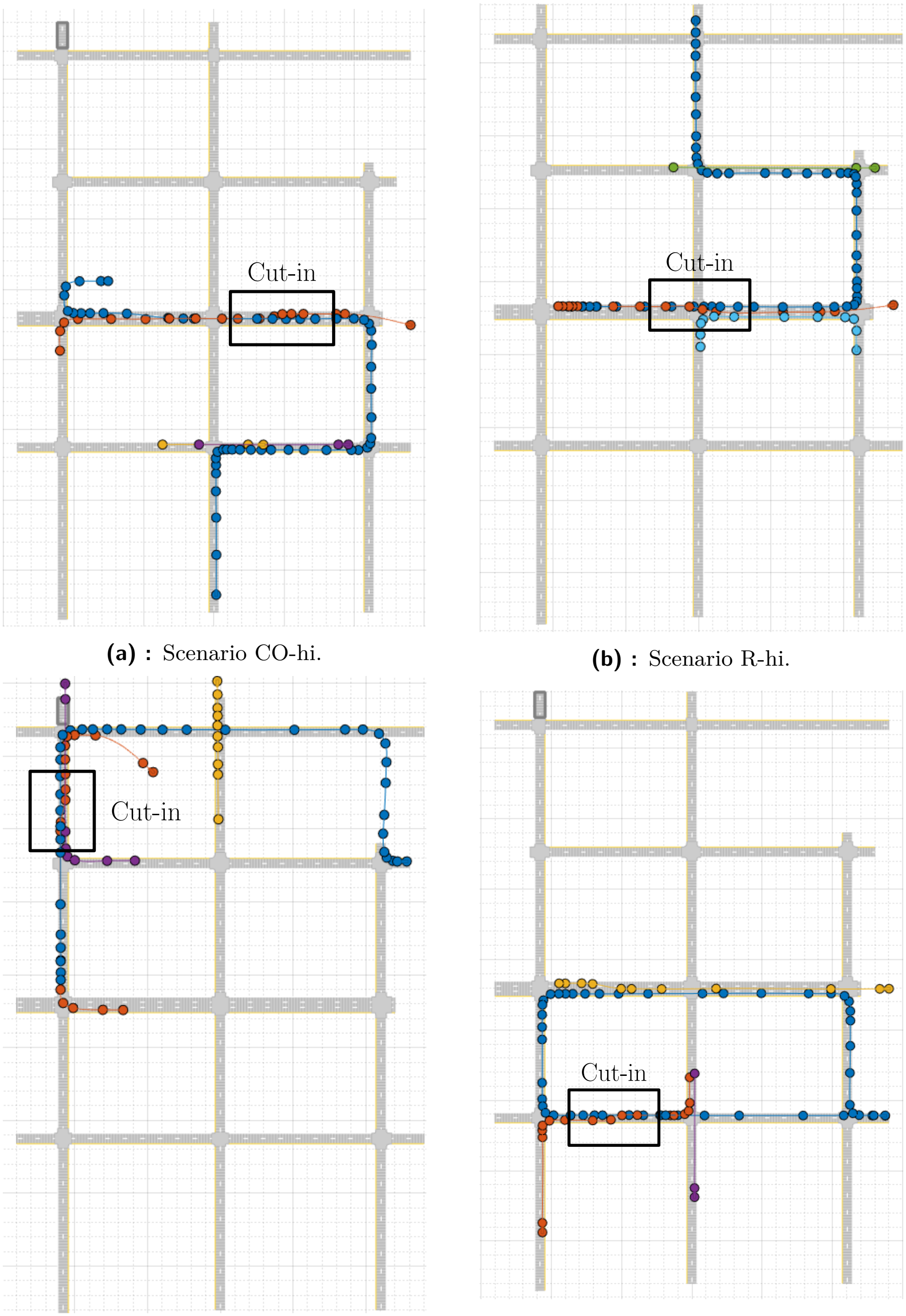


Figure 4.11: Recorded motion platform responses and their references for scenario CO-lo (on the left) and scenario R-lo (on the right).



(a) : Scenario CO-hi.

(b) : Scenario R-hi.

(c) : Scenario CO-lo.

(d) : Scenario R-lo.

Figure 4.6: Waypoint visualisations of the designed scenarios. The axes are swapped compared to figure 4.1 because Driving Scenario Designer plots the Y axis in the horizontal direction. The blue coloured waypoints show the trajectory of Ego vehicle, the orange coloured waypoints the trajectory of Target vehicle. The other colours are of the additional vehicles.

Chapter 5

Experiment Methodology

The purpose of the created scenarios is to prove the improvement of the riding comfort during scenarios CO-hi and CO-lo compared to scenarios R-hi and R-lo. This has been done by subjecting volunteers to the created scenarios aboard the motion platform. The feedback was gathered through a questionnaire after each of the scenarios. The detailed verification procedure is the subject of this section.

The number of volunteers has been affected by the COVID-19 outbreak and the limited access to the Strojovna lab (where the motion platform is stationed) as one of the pandemic's consequences. Therefore, the volunteers have been gathered among other students and employees of the Czech Technical University in Prague.

The verification procedure has been initiated by an introductory questionnaire to find out personal information about each participant: age, sex, information about the participants' driving licence if they possess one, frequency of their driving and their usage of CC and ACC systems. Participants have then been instructed to fasten their seat belt and informed that their feedback will be used to help the development of a navigation and a control system of an autonomous vehicle.

Each participant has been asked to count the number of intersections with traffic lights for each scenario. This served as a distraction from the main task – riding comfort evaluation during a cut-in maneuver – which has not been revealed to the participants. The order of scenario execution has been chosen randomly to prevent a bias from getting accustomed to the motion platform's movements during the first played scenario.

A separate questionnaire was given to the participant after each of the four scenarios. Its Czech version is presented in figure 5.1 and the English version in figure 5.2. The structure is adapted from [15] with modifications to focus on passenger's riding comfort rather than vehicle handling quality. A numerical rating has been assigned to the verbal "endpoints" of the questionnaire to help transform the subjective semantics of the answers into an objective form. Lower values of the rating imply higher riding comfort of the passenger, whereas high values of the rating imply lower riding comfort of the passenger. During the questionnaire design, the "endpoint" sentences without questions were given to four people to order them from the most positive to the most negative to see if the rating enumeration is reasonable. Apart from an occasional swap of two "neighbouring" sentences the order was consistent in their answers.

After having executed all four scenarios, a short verbal interview with the participants was initiated. The precise questions have been chosen for each participant individually, however, they followed a set pattern:

- "Have you noticed a recurring traffic situation in the four scenarios?"
- "Was the autopilot's reaction to the situation different in different scenarios?"
- "Which reaction do you consider better/worse and why?"

This dialogue concluded the verification session.

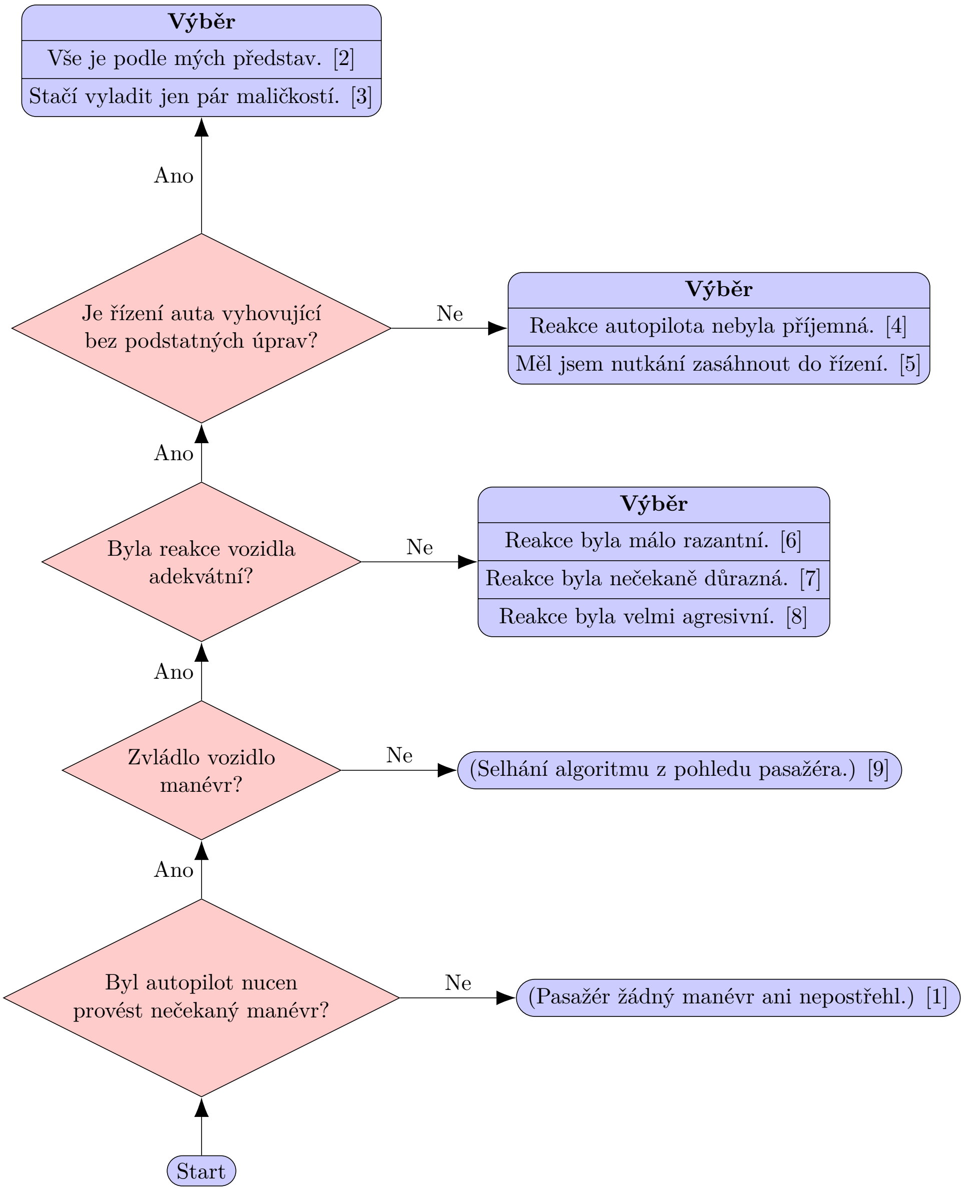

Výběr
Vše je podle mých představ. [2]
Stačí vyladit jen pár maličkostí. [3]
Ano
Je řízení auta vyhovující bez podstatných úprav?
Ne
Výběr
Reakce autopilota nebyla příjemná. [4]
Měl jsem nutkání zasáhnout do řízení. [5]
Ano
Byla reakce vozidla adekvátní?
Ne
Výběr
Reakce byla málo razantní. [6]
Reakce byla nečekaně důrazná. [7]
Reakce byla velmi agresivní. [8]
Ano
Zvládlo vozidlo manévr?
Ne
(Selhání algoritmu z pohledu pasažéra.) [9]
Ano
Byl autopilot nucen provést nečekaný manévr?
Ne
(Pasažér žádný manévr ani nepostřehl.) [1]
Start

Figure 5.1: The Czech version of the questionnaire. The rating is specified in the brackets (lower is better).

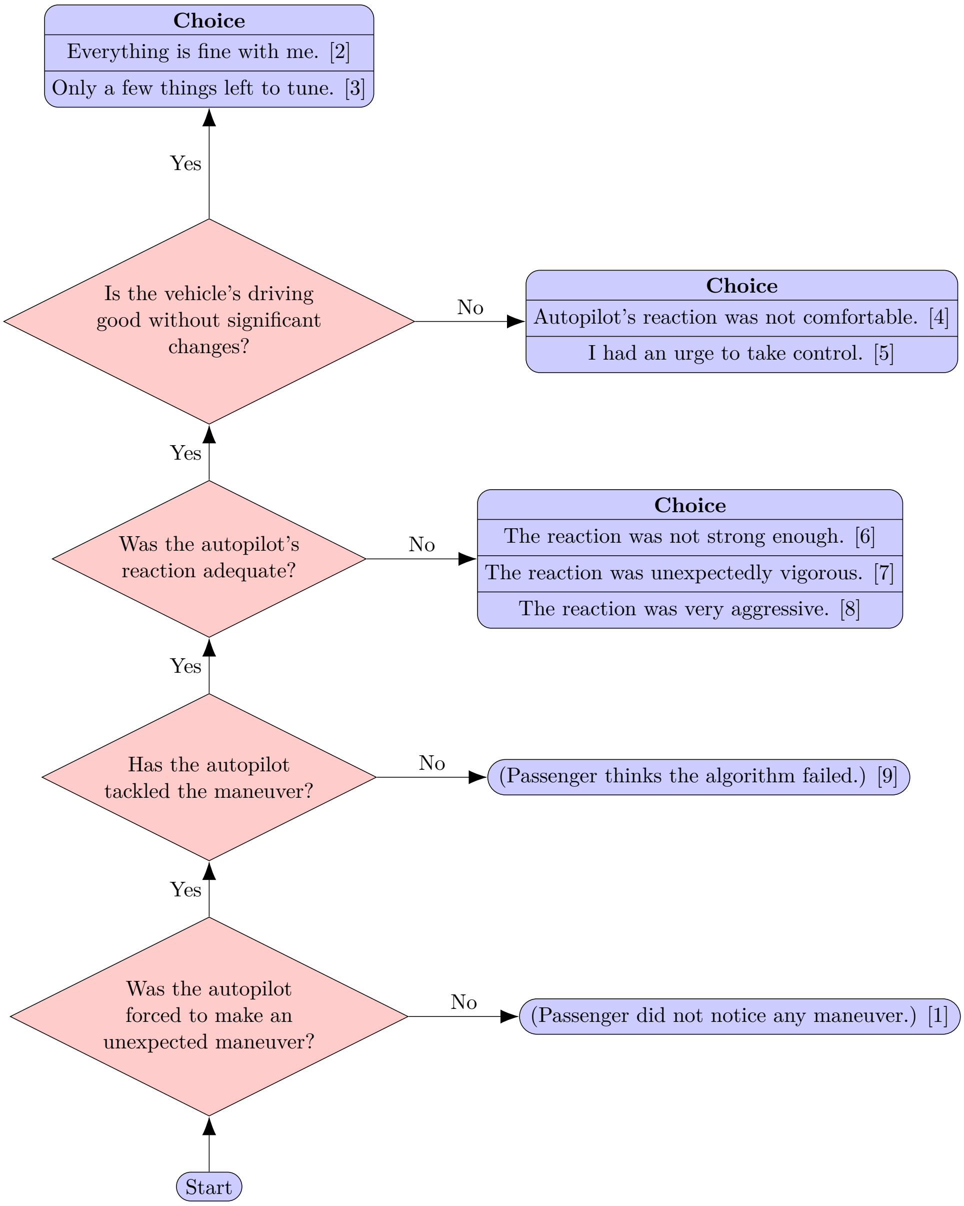

Choice
Everything is fine with me. [2]
Only a few things left to tune. [3]
Yes
Is the vehicle's driving good without significant changes?
No
Choice
Autopilot's reaction was not comfortable. [4]
I had an urge to take control. [5]
Yes
Was the autopilot's reaction adequate?
No
Choice
The reaction was not strong enough. [6]
The reaction was unexpectedly vigorous. [7]
The reaction was very aggressive. [8]
Yes
Has the autopilot tackled the maneuver?
No
(Passenger thinks the algorithm failed.) [9]
Yes
Was the autopilot forced to make an unexpected maneuver?
No
(Passenger did not notice any maneuver.) [1]
Start

Figure 5.2: The English version of the questionnaire. The rating is specified in the brackets (lower is better).

Chapter 6

Results

A total of 27 volunteers have been gathered to attend the experiments. The information gathered from the introductory questionnaire are presented in figures 6.1 and 6.2. The primary hypothesis is that the average rating for each of the scenarios will be different. More specifically, the order of the scenarios with respect to the average rating is presumed to be CO-lo, CO-hi, R-lo, R-hi from the lowest rating to the highest rating. This ordering is based upon the initial subjective impressions acquired during the scenario design process.

The rating x for each scenario is assumed to have normal distribution in the form

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \tag{6.1}$$

where μ is the mean and σ the standard deviation. The μ and σ parameters are found by fitting the obtained data using the Julia's `fit_mle` function implementing the maximum likelihood estimation [19]. It is acknowledged that the gathered sample of 27 volunteers is not statistically significant. The found distributions are only a rough approximation of what the true probability density is. The key parameters are the rating means μ for the scenarios. The fitted distributions are shown in figure 6.3a and the found parameters in the first row of table 6.1.

During the verbal interviews, it was noticed that participants without a driving licence were either struggling to describe the differences between the scenarios, or even failing to notice any differences between them. For this reason, the distribution fitting process was repeated with only the responses from the volunteers with a driving licence taken into account. The results are presented in figure 6.3b and the second row of table 6.1.

After about a half of the total number of volunteers had undertaken the experiments, it became apparent that there was a flaw in the wording of the first question of the questionnaire. Thus, a great number of participants responded negatively, which caused the questionnaire to end with the best rating of 1 as the result. These responses had the effect of greatly lowering the average of the scenario ratings. In the endeavour to partly circumvent the issue, other distributions were fitted using only the questionnaires whose rating was above 1. The distributions are shown in figure 6.3c and the third row of table 6.1

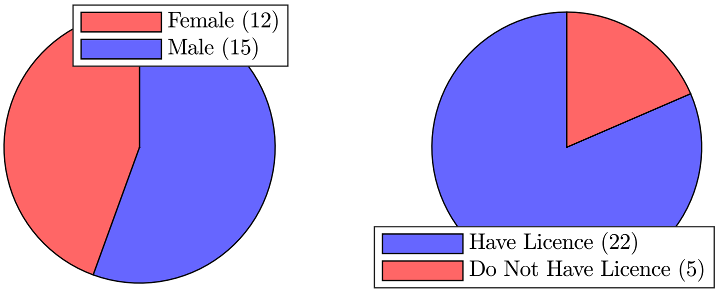


Figure 6.1: Diagrams showing the personal information about the volunteers regarding their sex and the possession of a driving licence.

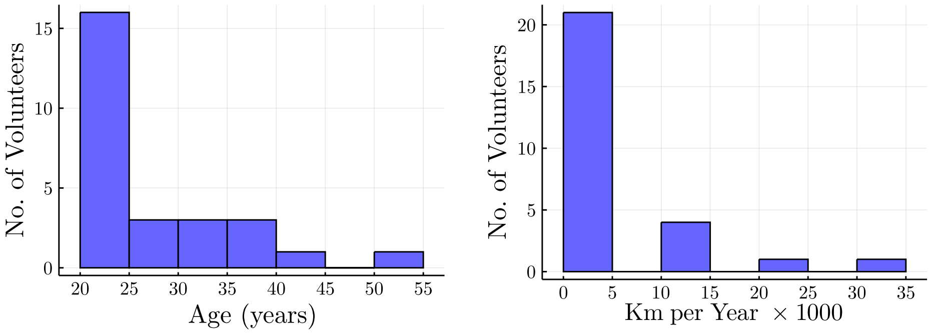
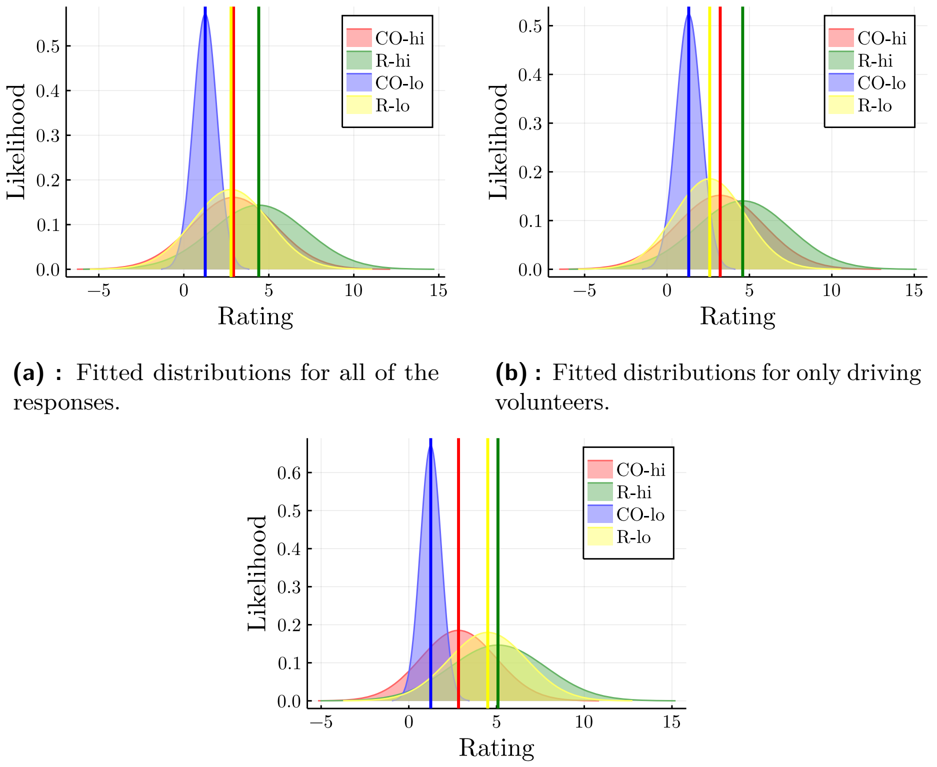


Figure 6.2: Histograms of the age of the volunteers and their yearly driven distance. Volunteers without a driving licence have driven 0 km per year.

(μ, σ)	CO-hi	R-hi	CO-lo	R-lo
All	(2.93, 2.48)	(4.41, 2.78)	(1.26, 0.70)	(2.78, 2.23)
Drivers only	(3.22, 2.63)	(4.60, 2.84)	(1.32, 0.76)	(2.60, 2.15)
Upper ratings	(2.83, 2.15)	(5.08, 2.72)	(1.25, 0.60)	(4.50, 2.22)

Table 6.1: Found means and standard deviations for all samples, samples from only the passengers who possess a driving licence and samples above the rating of 1.

The extension of the verification process in the form of a verbal interview proved to be very valuable. Even the participants whose answers in the questionnaire did not fully reflect their experience were able to share them verbally. In total, only two people mentioned they were not able to distinguish



(a) : Fitted distributions for all of the responses.

(b) : Fitted distributions for only driving volunteers.

(c) : Fitted distributions for ratings different from 1.

Figure 6.3: The fitted normal distributions for the scenarios CO-hi, R-hi, CO-lo and R-lo. The vertical lines show the mean values of the distribution of the corresponding colour.

the scenarios while scenario R-hi was widely considered the least comfortable. Another interesting finding was that the cut-in maneuver seemed to retroactively affect the overall impression of the entire scenario. The participants claimed the "autopilot" in scenario R-hi was driving more aggressively for the whole duration even though the only part that was different from the other scenarios was the reaction to the cut-in maneuver.

A number of correlation coefficients have been calculated to check for any unexpected relationships in the collected data. The coefficients are shown in table 6.2. Their absolute values are generally less than 1, which would imply a linear statistical relationship. The only coefficients that are greater than 0.50 are the correlation between the participant's driven kilometres per year and rating R-hi and the correlation between rating R-hi and rating R-lo. The correlations further support the fact that the volunteers with a driving licence proved to be more observant during the experiments.

Correlations	km/year	Rtg. CO-hi	Rtg. R-hi	Rtg. CO-lo	Rtg. R-lo
Age	0.26	-0.02	0.11	-0.11	0.07
km/year	–	0.32	0.55	0.08	0.15
Rtg. CO-hi	–	–	0.31	-0.05	-0.07
Rtg. R-hi	–	–	–	0.35	0.54
Rtg. CO-lo	–	–	–	–	0.42

Table 6.2: Correlations between the age of the volunteers, their yearly driven distance and their rating (abbreviated as rtg.) of the four created scenarios obtained from the questionnaire.

The participants with a driving licence were also very keen on sharing their idea of a comfortable ride. According to their remarks, the key lies in the close observation of the surrounding traffic situation and planning ahead. Under these conditions, the driver is able to drive with mild and continuous pressing of the pedals. This translates as minimising the longitudinal jerk.

Despite the faulty wording of the first question, the distributions in figure 6.3a show that scenario CO-lo was considered the most comfortable and scenario R-hi the least comfortable. This monotony is also preserved in the fitted distributions in figures 6.3b and 6.3c. The average ratings of scenarios CO-hi and CO-lo are sufficiently distanced only in figure 6.3c, where the originally predicted ordering CO-lo, CO-hi, R-lo, R-hi is also visible. Together, it is believed that the findings give enough evidence to claim the developed optimisation algorithm can provide more comfort during a cut-in maneuver.

Conclusion

The goals of this thesis were to develop an algorithm for ACC response optimisation during a cut-in maneuver and its verification on a set of experiments with the use of a 6-DoF motion platform and with the help of volunteers. These objectives have been fulfilled: ACC optimisation algorithm has been implemented in a MATLAB application and a total of 27 participants helped to verify its performance. The number of volunteers was limited due to the COVID-19 outbreak.

The optimisation tool relies on linear programming and computes the reference velocity or acceleration for ACC controlled vehicle to follow during a cut-in maneuver. The comfort aspect is maintained by introducing a cost function which negatively reflects high longitudinal acceleration and longitudinal jerk values. The calculated reference signals are then fed to the software driving simulator Live for Speed to obtain realistic responses. These are used to control the motion of the 6-DoF platform.

A total of four riding scenarios have been created in the Automated Driving Toolbox, two of which are controlled by the developed algorithm and the other two by a different ACC without riding comfort optimisation in place. The volunteers filled out a questionnaire regarding their riding comfort after each ride. An additional verbal interview concluded the testing session.

The results obtained from the questionnaire have provided enough evidence that the developed algorithm offers better riding comfort during a cut-in maneuver than the referential ACC system. This conclusion is justified even though the wording of one of the questions has been found to be flawed. Only two people reported they were feeling comfortable during all of the scenarios, while the other verbal responses were clear that the developed algorithm had performed better.

References

1. SINGH, Santokh. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*. NHTSA's National Center for Statistics and Analysis, 2018. Available also from: `https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812506`.
2. STRANDEMAR, Katrin; KUNGLIGA TEKNISKA HÖGSKOLAN; INSTITUTIONEN FÖR SIGNALER, sensorer och system. *On objective measures for ride comfort evaluation*. Stockholm, 2005. PhD thesis. ISBN: 9789171782045 OCLC: 185156136.
3. WANG, Chang; ZHAO, Xia; FU, Rui; LI, Zhen. Research on the Comfort of Vehicle Passengers Considering the Vehicle Motion State and Passenger Physiological Characteristics: Improving the Passenger Comfort of Autonomous Vehicles. *International Journal of Environmental Research and Public Health* [online]. 2020, vol. 17, no. 18, p. 6821 [visited on 2021-04-15]. ISSN 1660-4601. Available from DOI: `10.3390/ijerph17186821`.
4. RAJAMANI, Rajesh. *Vehicle dynamics and control*. 2. ed. New York, NY: Springer, 2012. Mechanical engineering series. ISBN 978-1-4614-1433-9 978-1-4614-1432-2. OCLC: 774382146.
5. *How to Parallel Park* [online] [visited on 2021-05-04]. Available from: `https://www.dtc-wsuv.org/pmeiners14/howto/`.
6. GMBH, Robert Bosch (ed.). *Automotive handbook*. 6th rev. and extended ed. Plochingen : London: Robert Bosch ; Professional Engineering Pub, 2004. ISBN 978-1-86058-474-9. OCLC: ocm57599060.
7. ŠVANCAR, Jan. *Adaptive cruise control algorithm development*. Prague, 2020. Bachelor Project. České vysoké učení technické v Praze.
8. WINNER, Hermann (ed.). Adaptive Cruise Control. In: *Handbook of intelligent vehicles*. London ; New York: Springer, 2012. Springer reference. ISBN 978-0-85729-084-7. OCLC: ocn670283732.

9. *ISO - ISO 15622:2018 - Intelligent transport systems — Adaptive cruise control systems — Performance requirements and test procedures* [online] [visited on 2021-04-24]. Available from: `https://www.iso.org/standard/71515.html`.

10. *Sensory Perception – Anatomy and Physiology* [online] [visited on 2021-04-28]. Available from: `https://opentextbc.ca/anatomyandphysiologyopenstax/chapter/sensory-perception/`.

11. *Online Racing Simulator - Live for Speed* [online] [visited on 2021-04-14]. Available from: `https://www.lfs.net/`.

12. *Automated Driving Toolbox - MATLAB* [online] [visited on 2021-04-14]. Available from: `https://www.mathworks.com/products/automated-driving.html`.

13. *Create actor or vehicle trajectory in driving scenario - MATLAB trajectory* [online] [visited on 2021-05-08]. Available from: `https://www.mathworks.com/help/driving/ref/drivingscenario.trajectory.html?s_tid=doc_ta`.

14. EIZIKOVICH, Shaul. *vJoy* [online] [visited on 2021-05-10]. Available from: `http://vjoystick.sourceforge.net/site/`.

15. KONOPISKÝ, Adam. *Objective evaluation of a vehicle handling quality perception*. Prague, 2020. Bachelor Project. České vysoké učení technické v Praze.

16. TWARDZIG, Tomáš. *Verification platform development for vehicle control system validation*. Prague, 2020. Bachelor Project. České vysoké učení technické v Praze.

17. SAUNDERS, Matt. *tidzo/pyvjoy* [online]. 2021 [visited on 2021-05-10]. Available from: `https://github.com/tidzo/pyvjoy`. original-date: 2016-10-17T21:05:24Z.

18. *Multifunkční letecký simulátor* [online] [visited on 2021-04-09]. Available from: `http://www.pragolet.cz/projekty-multifunkcni-letecky-simulator-detail-10`.

19. *Distribution Fitting · Distributions.jl* [online] [visited on 2021-05-16]. Available from: `https://juliastats.org/Distributions.jl/latest/fit/`.