Fakultät für Maschinenwesen
Lehrstuhl für Angewandte Mechanik

Urban Motion Cueing Algorithms –
Trajectory Optimization for Driving Simulators

Felix J. Ellensohn

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Florian Holzapfel

Prüfer der Dissertation:

1. Prof. dr.ir. Daniel J. Rixen
2. Prof. Richard Romano, Ph.D.

Die Dissertation wurde am 4. September 2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 11. Dezember 2019 angenommen.

Acknowledgement

This dissertation was written at the Chair of Applied Mechanics at the Technical University of Munich between October 2016 and November 2019. Most of the research results are based on the industry funded project “Urban Motion Cueing Algorithms”. Without the support of numerous people, the present work would not have been possible.

My special thanks go to my supervisor Prof. dr. ir. Daniel Rixen. His trust and valuable advice gave me the freedom to work on new ideas and to pursue my interests. I would like to emphasize the positive working atmosphere at the chair, which enhances motivating working conditions. Proofreading of the dissertation and executing the associated examination require a lot of effort. Therefore, I would like to acknowledge Professor Romano for serving on my thesis defense committee and Prof. Dr.-Ing. Holzapfel for being the chair of the examination.

The project “Urban Motion Cueing Algorithms” would not have been possible without the financial support and the technical facilities of the BMW Group. I warmly thank Dr.-Ing. Markus Schwienbacher who requested the industrial project and elaborated such an exciting research project. He arose my enthusiasm in the fascinating field of driving simulation. Dr. ir. Joost Venrooij and Dr.-Ing. Markus Schwienbacher supported the project. I learned so much from your long experience in driving simulation and research. Thank you very much for your valuable advice and support!

I am grateful for having had the chance to work with the robotics research group at the Chair of Applied Mechanics. Thank you for the enlightening discussions in the fields of robotics! Cooperating with Dr. Daniel McGehee, Omar Ahmad and Dr. Chris Schwarz at the University of Iowa was a valuable enrichment. Thank you very much for providing the NADS driving simulator in order to test our motion cueing algorithms! Performing driving simulation studies is essential in the evaluation of the motion cueing quality. I am especially thankful to all participants for taking the time and the efforts in the driving simulation studies. Furthermore, I warmly thank the BMW Group for providing the driving simulators and for assisting the experimental studies. I would like to thank the consulting service of the TUM for revising the statistical analysis. I also would like to express my gratitude to Dr.-Ing. Markus Schwienbacher, Christina Insam, Dimitar Hristakiev and Jonas Wittmann for proofreading my dissertation and for giving helpful comments. I owe special thanks to Dr.-Ing. habil. Thomas Thümmel, Manuella Müller-Philipp, Daniela Priller, Georg Mayr, Georg König and Simon Gerer. You play an important role in the well-functioning infrastructure of the chair.

Finally, I would like to deeply thank all colleagues of the chair and all students I have supervised. The success of my dissertation and the research project was largely possible due to your motivation and efforts.

Munich, January 28, 2020 *Felix Ellensohn*

Abstract

Immersing humans to virtual driving environments involves a realistic reproduction of the motion simulation. This thesis covers methods for the trajectory generation of driving simulators which aim to increase the motion fidelity. An essential task of these motion cueing algorithms (MCAs) is to generate realistic motions by adhering to the motion system's physical limits. Simulations and experiments were performed with a nine degrees of freedom driving simulator located at the *BMW Group, Research, New Technologies, Innovations*. A special focus is put on optimization-based MCAs which minimize deviations between desired vehicle motions and actual simulator motions over a future time horizon. For real-time interactions, allowing an active drive, model predictive control techniques are used. An MPC-scheme solves the optimal control problem at discrete time steps to obtain an optimal motion for the redundant motion system. As a prediction, the current vehicle motions are kept constant over the time horizon. For open-loop interactions, where the driver sits passively in the simulator and is being driven through a pre-recorded driving manoeuvre, an ideal driving dynamics prediction over the complete manoeuvre is applied to obtain an optimal trajectory for the pre-defined manoeuvre. Experimental studies with participants demonstrate an increase in motion cueing quality of these passive algorithms compared to real-time capable MCAs. The thesis proposes a method to make use of this optimal trajectory in an active drive in order to improve the motion system's workspace exploitation. Experimental studies show the potential of the presented MCAs and give good evidence to increase the motion fidelity compared to state-of-the-art algorithms. Hereby, various rating systems, such as the continuous rating, post-hoc rating and section-wise post-hoc rating, are applied which aim to measure deviations between expected vehicle motions and actual simulator motions.

KEYWORDS: Motion cueing algorithms, trajectory optimization, model predictive control, prediction, driving simulation, motion cueing rating methods

Contents

List of Abbreviations

BVP Boundary Value Problem

CRP Cockpit Reference Point

DoF Degree of Freedom

DiM Driver-in-Motion

FoR Frame of Reference

HP High-Pass

IVP Initial Value Problem

LP Low-Pass

MC Motion Cue

MCA Motion Cueing Algorithm

MPC Model Predictive Control

OCP Optimal Control Problem

ODE Ordinary Differential Equation

OMI Objective Motion Incongruence

PMI Perceived Motion Incongruence

QP Quadratic Programming

RM Rating Model

rms root-mean-square

SPR Section-wise Post-hoc Rating

TCP Tool Centre Point

Chapter 1

Introduction

Content of this thesis have been published in the following author's publications: [47–56].

Due to the increasing performance of computers, employing simulation tools is becoming more and more important in research and development. Nowadays, applying simulation environments is common practice in the design of components or the control of dynamic systems, to name just a few examples. It offers advantages to replace practical experiments by a computer-based calculation in order to safe time and to reduce cost. Apart from these general aspects, the feasibility of practical experiments, using current measurement techniques, is often restricted. For example, physically measuring certain states of a system, can be associated with large efforts. A further complicating factor is that some states are not directly measurable, but can only be estimated from other states. A model of the system may therefore be a common approach to approximate the desired system states rather than conducting cumbersome experiments.

Driving simulation is another application area, where driving simulators emulate a real vehicle drive in a virtual environment. Based on the driver inputs, simulation models calculate the states of the virtual scenery. The simulation hardware deploys the models' outputs to give feedback to the driver in form of visuals, acoustics and motions.

In recent years, employing driving simulators in research and development has increased significantly. Along with the advantages mentioned above, driving simulation offers reproducible and safe test conditions. These specifications play an important role, for instance, in investigating the driving behaviour and/or in testing safety-critical functions. Driving simulation opens new doors to relocate practical studies from a real vehicle to a virtual environment. Though, this requires that results of a driving simulation study have a comparable significance, saying that the scientific insights can be adapted to real vehicles [148].

This Chapter gives an introduction to the application fields of driving simulation. Based on this, the problem of using driving simulators for replicating urban scenarios is described. Finally, the contribution of this thesis and the structure of the work is briefly presented.

1.1 Driving Simulation

Immersing humans to virtual environments has its origin in flight simulation. In the late 1920s, the first pilot training device was developed which showed positive effects on pilots' performance. The dynamic flight simulator was pneumatically actuated and allowed a safe and improved pilot training environment in instrument flight. Due to progress in digital computer technology in the 1960s and 1970s, great improvements in the motion and vision

systems of flight simulators were accomplished. [3]
In the 1980s, the first vehicle manufactures, such as Volkswagen [43], Daimler [46] and VTI [126], began to employ concepts of flight simulators for use in vehicle development. Different requirement profiles between flight simulators and driving simulators represented major challenges in the adaptation process. Passenger aircraft exhibit relatively low dynamic requirements during a usual flight phase (neglecting the take-off and landing manoeuvre). Whereas, high dynamic manoeuvres may occur with road vehicles, for example, during sharp turns or abrupt stopping manoeuvres. Thus, driving simulators require high dynamic capabilities in the longitudinal, lateral and yaw direction whereas for flight simulators the vertical direction is more dominant. In order to quantify the accelerations which are to be reproduced during a virtual driving session, Figure 1.1 shows the results of an experimental study conducted by [14]. The authors measured the maximum acceleration ranges for different driver types with the distinction between average drivers and sportive drivers. Comparing these profiles to the average and maximum acceleration potential of a vehicle, indicates the high dynamics which are acting on the driver.

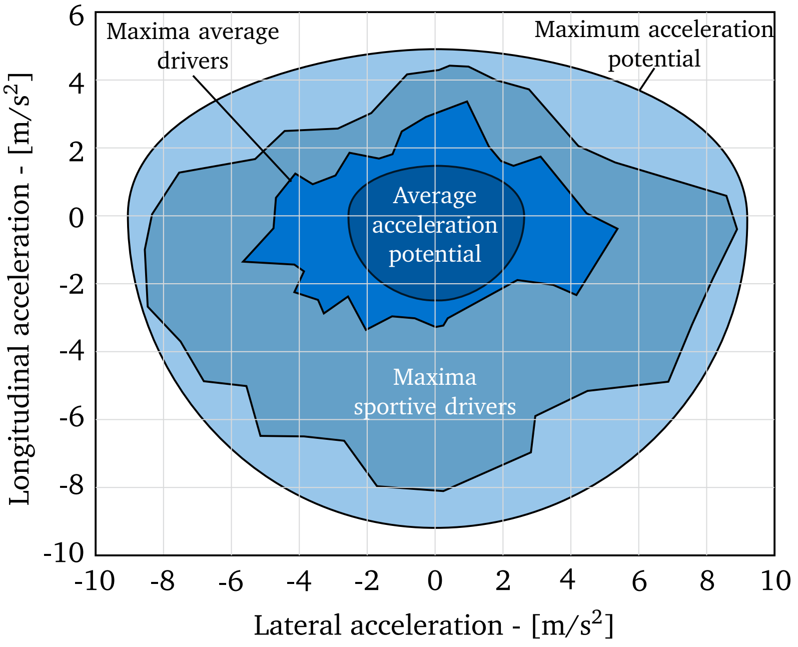


Figure 1.1: Ranges for maximum accelerations for average drivers and for sportive drivers compared to the average and maximum acceleration potential (Adapted from [14]).

Considering the application areas, flight simulators preferably have been used for pilot training. Tasks mainly involve the simulation of realistic operations and controls in the cockpit, whereas reproducing realistic flight motions is not the primary focus. In contrast to that, driving simulators are mainly employed in research and development, where the validity of the results are paramount. Motion incongruencies between a real vehicle drive and a simulator drive can considerably influence the purpose of the study. Reproducing a real vehicle drive more realistically leads to a higher immersion of the driver, and thus increases the driving experience. [17, 143]
Driving simulators are primarily applied in research areas which deal with questions about the driving behaviour during primary and secondary tasks or the interaction of the driver with the environment [19]. Using driving simulators enables an efficient way to include the human factor at an early stage of development; thus, avoiding time- and cost-intensive tests

with prototypes. Employing driving simulators instead of real vehicles can provide benefits which are described in [140] to:

1. *Safety*: Some experimental studies are too dangerous to conduct with real vehicles, such as driving behavioural tests under the influence of drugs or collision-avoidance tests.

2. *Equipment cost*: Changing the interior/cockpit or installing measurement techniques are often less expensive in simulator environments with standardized measurement interfaces than in real vehicles.

3. *Experimental control*: Repeatable and consistent test environments under freely definable weather, traffic and road conditions are crucial in driving behavioural analysis. Along with these aspects, weather plays an important role for using driving simulators. In some places, conducting experiments is not feasible because of bad weather conditions, such as rain and snow.

However, validating driving simulation studies is generally complex in order to show that the results are significantly transferable to real driving scenarios. Not all scenarios are replicable in a sufficient quality. For example, limitations of the visualization system regarding brightness (e.g. daylight, night drives) but also the limiting workspace capabilities (reproducing deceleration manoeuvres, turns, roundabouts, etc.) restrict the driving simulation capabilities.

Possible application areas for using driving simulators instead of real vehicles are for example mentioned in [7, 19, 67, 150, 152]:

- Determination of driver-related parameters (reactivity, fatigue, concentration, etc.)

- Development of driver assistance systems (tuning, controllability, evaluation, etc.)

- Testing of new operating concepts (human-machine-interfaces, driver distraction, handling, etc.)

- Design of chassis components (steering system, suspension system, etc.)

Evaluating In-Vehicle Information Systems (IVIS) and Advanced Driving Assistant Systems (ADAS) show valid results in driving simulators, test tracks and real environments [57, 184]. In order to immerse a driver to a virtual reality, acoustic, visual, somatosensory (e.g. haptic) and vestibular (perception by the equilibrium organ) stimuli can be provided by the driving simulator. Visual perception gives the driver an impression about the virtual environment of the driving manoeuvre. The immersion depends on the reality level of the simulation. The visual system usually consists of screens, projectors, virtual reality (VR) headsets, etc. . [182] Acoustic perception channels are stimulated with an integrated sound system to provide an appropriate background sound for the driver. In particular, the speed perception can be significantly improved by acoustic impressions including sounds for wind, engine and tyre rolling. [66] The interaction between the driver and the simulator takes place in a mockup, i.e. a vehicle body with a complete interior and cockpit. Typically, the same interfaces as in a real vehicle are present which include steering wheel, pedals, input devices, displays, etc. . [150]

Dynamic simulators contain a motion system for the representation of somatosensory and vestibular stimuli. To reproduce high frequent motions, shakers are often installed in the driver seat and/or the steering wheel. The motion system aims at providing the same angular velocities and translational accelerations to the driver as in a real vehicle. By contrast, static simulators are limited to visual and acoustic stimuli. [150]

Due to increasing demands on driving simulators, motion system structures are becoming more and more complex [61, 157, 190]. Redundant degrees of freedom are integrated in the motion system to increase the motion space. Distributing the redundant DoFs depending on the driving dynamics demands is a challenging task, since the kinematic dependencies between the subsystems as well as workspace restrictions have to be considered.

1.2 Problem Statement

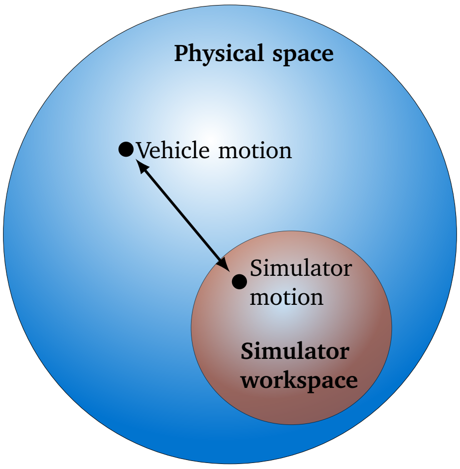


Figure 1.2: Schematic figure of the physical space of a vehicle and the workspace of a motion simulator.

Motion Cueing Algorithms (MCAs) compute the simulator's control inputs based on the desired motions of the simulated vehicle. The technical restrictions of the simulator's actuators constitute the main limitation in the execution of these input commands. The blue sphere in Figure 1.2 schematically represents the physical space of a real vehicle which is restricted by the vehicle's motion capabilities, such as the course of the road, traffic regulations, driving dynamics restrictions. The orange sphere depicts the workspace envelope of a motion simulator which is bounded by its kinematics and dynamics. Comparing both spheres, the workspace of the motion simulator is usually much smaller than the physical space of a real vehicle. To say it in other words, most often motion simulators cannot directly provide the demanded vehicle motions because of workspace restrictions. This thesis covers methods on how to transform vehicle motions to the simulator's workspace envelope under consideration of the actuator limitations. Discrepancies between desired vehicle motions and provided simulator motions can cause motion sickness which can lead to high drop-out rates during experiments with participants. Intending to improve the motion cueing quality, MCAs aim at reducing discrepancies between desired and provided motions by coping with the simulator's workspace capabilities.

These discrepancies highly depend on the driver commands and the choice of the course. Urban driving manoeuvres often involve large vehicle motions, which are difficult to reproduce by the driving simulator. Typical urban scenarios, such as roundabouts, sharp curves, lane-changes, etc., induce high lateral accelerations and yaw rates. Considerable longitudinal accelerations are acting on the driver during starting and stopping manoeuvres, which can occur in front of traffic lights, junctions or in traffic jams. Realizing these motions on a simulator often leads to large deviations from the demanded vehicle motions.

Since the simulator excursions are limited, significant improvements in motion capabilities can be achieved by expanding the simulator with redundant degrees of freedom (DoFs) by means of additional actuators. Aiming at increasing the motion cueing quality, several general questions arise:

1. How to solve the redundant DoFs?
   Motion perception can be divided into six main motion channels, three rotational motion channels and three translational motion channels. Based on the kinematic structure of the motion system, several DoFs of the driving simulator have an impact on a single motion channel of the driver.

2. Do simulator motions adhere to the workspace capabilities?
   Workspace capabilities have to be considered in the MCA.

3. How to evaluate the motion cueing quality?
   Motion cueing discrepancies have to be rated by simulator users.

In case the simulator driver is actively driving, the real-time capability of the MCA is another crucial point. The MCA framework has to provide simulator controls within hard response deadlines in the range of a few milliseconds. The main research platform for this work is the Driver-in-Motion-simulator (DiM, shown in Figure 1.3) at the *BMW Group, Research, New Technologies, Innovations*. The developed MCAs are applied and tested with this simulator but can be applied to other simulator kinematics.

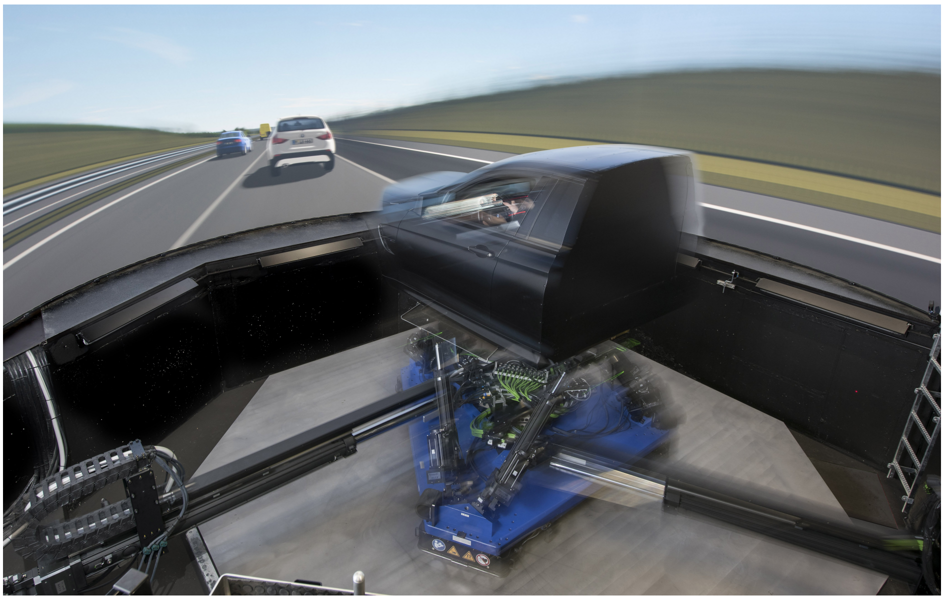


Figure 1.3: Nine DoFs driving simulator (DiM) consisting of tripod and hexapod.

1.3 Contribution of this Thesis

The first objective of the thesis is the development of MCA methods for a redundant driving simulator. These MCAs calculate control inputs for the motion system subject to the simulator's workspace capabilities. A driving dynamics model estimates the vehicle motion based on the driver commands. The vehicle motion is used as input for the MCAs. Two interaction types between simulator and driver are distinguished. For the first type, algorithms estimate the simulator motions in a pre-processing step for a pre-recorded manoeuvre so that the

driver sits passively in the simulator. The second interaction type constitute MCAs which are developed with the special effort to be real-time capable, meaning that the driver can actively drive in the simulator. A special focus is put on optimization-based MCAs which minimize deviations between desired vehicle motions and actual simulator motions over a future prediction horizon. Actuator limitations are included as constraints in the optimization.
The second objective is the experimental evaluation of the MCAs. Studies are performed which compare the developed MCAs to benchmark algorithms in order to determine the motion cueing potential. Published rating methods but also novel rating methods are used in the experiments to determine the motion cueing quality. Experimental results are analysed for significance.

The main contributions of the thesis are:

- Development of optimization-based MCAs that calculate an optimal motion for a redundant driving simulator. Two types are distinguished:
  - Real-time trajectory optimization using quadratic programming:
    - The optimization problem is included into a model predictive control algorithm
    - A linearized prediction model predicts the motions acting on the simulator driver and calculates the simulator states
    - The actuator constraints are considered by using a linearized, inverse kinematics model for parallel robots
    - Adhering to the hard response deadline within a few milliseconds of the simulation framework
  - Offline trajectory optimization using direct, multiple shooting methods:
    - Calculation of simulator motions for a pre-recorded manoeuvre
    - Consideration of the nonlinear kinematics of the redundant motion system to predict the motions acting on the simulator driver
    - Accounting for the nonlinear actuator constraints by using the inverse kinematics for parallel robots
- Development of an MCA that combines the benefits of real-time capable MCAs and precalculated simulator trajectories:
  - Merging an optimal trajectory of a pre-recorded manoeuvre with a real-time capable MCA
  - Quasi-optimal pre-positioning techniques in lateral and yaw directions
  - Heuristic pre-positioning techniques in longitudinal directions
- Experimental evaluation of the developed MCAs by:
  - Comparing the developed MCAs with state-of-the-art MCAs
  - Using the continuous rating method to measure perceived motion incongruencies in a passive drive
  - Using the section-wise post-hoc rating method to measure perceived motion incongruencies in an active drive
- Analysis of rating models which predict perceived motion incongruencies from objective motion incongruencies.

The thesis is structured as follows: Chapter 2 gives backgrounds on the considered driving simulator and describes basic preliminaries of the human motion perception. An overview of filter-based MCAs and pre-positioning techniques is part of Chapter 3. Chapter 4 presents an MCA which is based on an optimal control problem to calculate an optimal simulator trajectory for a pre-recorded manoeuvre. An optimization-based MCA, which uses a linear prediction model and linear inequality constraints, is part of Chapter 5. The quadratic programming problem is integrated into a model predictive control scheme. The algorithm was evaluated twice in an experimental study. In the first development stage, the algorithm was not ready for real-time applications. The experiment, described in Chapter 6, aimed at measuring the potential of the algorithm. In the second development stage, shown in Chapter 7, the algorithm was real-time capable. In the evaluation, the MCA was compared to a benchmark MCA and an optimal trajectory for a pre-recorded manoeuvre. Chapter 8 presents a novel motion cueing approach, called “Hybrid MCA”, which combines a filter-based MCA with a precalculated optimal trajectory for a pre-recorded manoeuvre. Chapter 9 summarizes the thesis and concludes with a discussion of the results.

Chapter 2

Motion Cueing Preliminaries

The following Chapter has partly been published in: [51, 52].

This Chapter gives an overview about the structure of motion systems and the human perception system which are essential components in the development of Motion Cueing Algorithms.

2.1 Motion Cueing Algorithms (MCAs)

In flight simulation and driving simulation, the term *motion cue* (MC) describes the immersion of pilots or drivers in virtual environments. A MC stimulates the respective sensory channels of the driver to replicate a real drive. A MC is generated by motion from a motion system's hardware which is commanded by the motion software. Both components are used to reproduce a real driving experience by using haptic, visual, acoustic and vestibular stimuli. Merging these sensory channels results in the human motion perception. The vestibular perception mainly triggers stimuli in response to velocity and orientation changes. These stimuli support the driver during his driving task and allow an immersion of the driver to the virtual world. Simulator motions give feedback about the strength and the direction of driving manoeuvres. Along with vestibular stimuli, other perception channels are stated in literature. Visual stimuli are a precise sensor to provide information about positions and velocities. Somatosensory stimuli and acoustic stimuli also play an important role to immerse the driver into the dynamic scenario. Vibrations, for instance, are perceived by haptic stimuli over the whole body. [6, 78]

Following [67], and in order to distinguish between the different perception channels, the term MC is associated with the vestibular perception system in the present work. The term *cue*, for distinction, is used for any stimuli type reproducing a driving dynamics task.

Taking these specifications into account, a *Motion Cueing Algorithm* (MCA) computes driving simulator motions subject to driver demands. These simulator motions are to replicate the driving dynamics of the virtual vehicle by providing MCs. An essential task consists in taking into account the simulator's workspace capabilities. Thus, a congruence must be made in providing a realistic driving experience but at the same time keep the simulator's states within the workspace envelope. Weaknesses of the human perception system are considered in order to exploit the motion system's workspace capabilities as far as possible.

2.2 Vehicle Kinematics

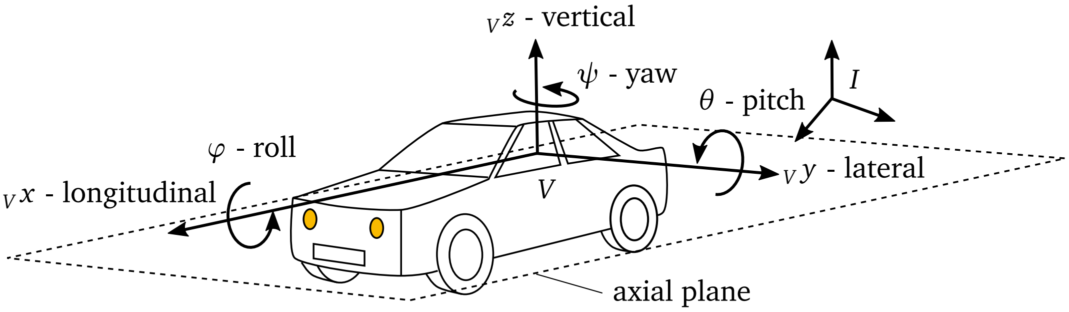


Figure 2.1: Vehicle coordinate system according to ISO 8855 [92].

In order to describe the kinematics of a virtual vehicle, two frames of reference (FoRs), V and I, are defined. The body fixed, relative FoR V is located at the driver's head. The orientation of the axes are chosen according to ISO 8855 [92] (cf. Figure 2.1). Its $_Vx$-axis (and the origin of the FoR V) lies in the vehicle's axial plane and is pointing in the positive longitudinal direction. The $_Vy$-axis is on the same axial plane and perpendicular to $_Vx$. The z-axis is orthogonal to the axial plane and is directed upwards. For the rotations, one distinguishes between roll (φ), pitch (θ) and yaw (ψ) rotations according to the axis. It is worth to mention that there are other conventions to define the coordinate system. In flight dynamics, the z-axis of the coordinate system usually points downwards [91, 142]. The notation of the relative kinematics, used throughout the work, can be found in Table 2.1. The transformation from the inertially fixed FoR I to the FoR V employs the Cardan's rotation sequence which is defined by

$$ {}_V\boldsymbol{R}_I(\boldsymbol{\beta}) = {}_I\boldsymbol{R}_V^T(\boldsymbol{\beta}) = \boldsymbol{R}_x(\varphi)\,\boldsymbol{R}_y(\theta)\,\boldsymbol{R}_z(\psi) \quad \text{with} \quad \boldsymbol{\beta} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}. \tag{2.1} $$

Rotations about the z-, y- and x- axis are defined by

$$ \boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad \text{and} $$
$$ \boldsymbol{R}_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\varphi) \end{bmatrix}. \tag{2.2} $$

Relevant equations for using the Cardan angles are listed in Appendix A.

2.3 Simulator Topologies

In recent years, engineers have developed new driving simulators with a variety of topologies. Dynamic simulators are associated to robotic systems which can exhibit a *serial* or *parallel topology* (cf. Figure 2.2). The *tool centre point* (TCP) of a parallel structure is linked with at least two actuators. The topology leads to a closed-loop kinematic chain between the fixed platform, the independent actuated joints and the motion platform. [22, 115]

Table 2.1: Notation of the relative kinematics.

Translational	
${}_K\boldsymbol{r}_{AB}$	*Vector* with respect to the FoR K with the start point A and the end point B.
Rotational	
${}_K\boldsymbol{R}_B$	*Rotation matrix* transforming a vector from FoR B to FoR K. *Example:* ${}_K\boldsymbol{r}_{AB} = {}_K\boldsymbol{R}_B\, {}_B\boldsymbol{r}_{AB}$.
${}_K\boldsymbol{\omega}_{NM}$	*Relative angular velocity* between the FoR N and the FoR M, denoted in the FoR K.

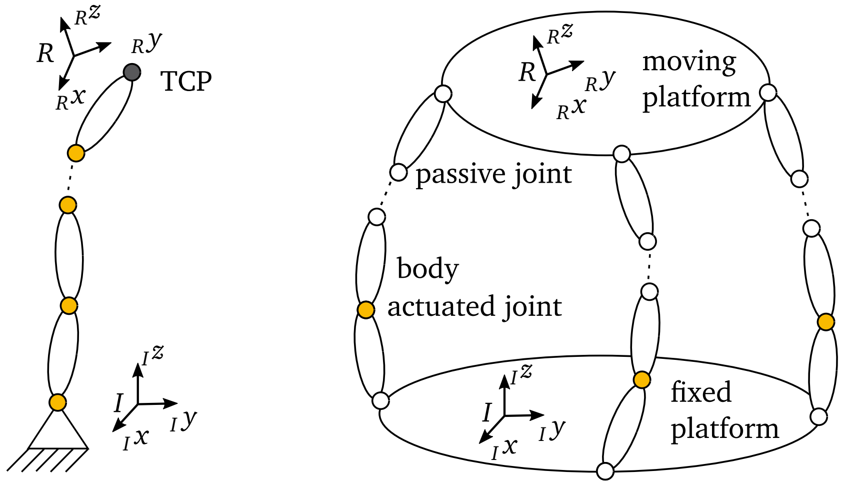


Figure 2.2: Schematic structures for a serial topology (left) and a parallel topology (right).

For serial topologies, the actuators are lined in rows and exhibit an open kinematic chain. The first actuator is connected to the surroundings and the last joint fixes the TCP of the manipulator. [160]

Both topologies find applications in driving simulation. A widely used parallel structure is the *hexapod* (also *Stewart/Gough Platform*) [76, 165]. It was developed by Gough/Stewart in the 1960s and exhibits six degrees of freedom (DoFs). A fixed base is connected to a motion platform by six translational actuators. The system is usually actuated by electric servo technology or hydraulics. Several driving simulators are based on this type, such as [79, 150, 167]. The compact design with high load capacity proves to be an advantage, in order to move complete vehicles with high dynamics. A disadvantage consists in the limited workspace capacities, which means that large accelerations can only be realized to a very limited temporal extent as otherwise position or velocity limits are reached. Thus, combinations of several parallel motion systems, which are modularly linked, are state-of-the-art. For example, a hexapod is fixed on top of an additional motion platform to increase the workspace. Most often, these platforms just exhibit motion space in a single plane, like a rail system or a tripod (three DoFs) [7, 33, 94, 170, 190] as the longitudinal, lateral and yaw motions are the main directions in driving simulation. These combinations result in *redundant motion systems* meaning that certain motions are reproducible by at least two parallel subsystems [159]. Research institutes, such as [162, 181], install rubber tires at the lower platform of parallel systems to additionally provide accelerations. These accelerations are restricted by the available area where the wheeled simulator is capable to move.

Motion simulators, based on serial robots, are described in [96, 156]. The simulators consist of industrial robots with a mockup mounted at the TCP. Advantages generally lie in larger workspace capacities in the vertical direction and in the rotational DoFs compared to parallel systems. This allows to reproduce manoeuvres with greater height differences and angles of attack. Thus, these simulator topologies are mainly used for flight simulation.

Throughout the thesis, the term simulator is associated to a motion system with a parallel kinematic chain. The algorithms proposed in this thesis can generally be applied to different parallel simulator topologies. The validation and evaluation of the algorithms is performed on a simulator at the BMW Group [77]. Throughout the thesis, the dynamic simulator is called Driver-in-Motion (DiM) simulator. The motion simulator consists of a tripod and a hexapod mounted on top of the tripod's motion platform (cf. Figure 1.3). The tripod is driven by three linear actuators in the horizontal plane to perform longitudinal, lateral and yaw motions. The tripod platform slides on a polished steel plate using a combination of air bearings and magnets to minimize friction. This combination effectively restricts the three DoFs motion to the plane. The hexapod has a smaller workspace but higher dynamic capabilities compared to the tripod. Therefore, particularly high frequency motions are performed by the hexapod, whereas low frequent motions are realized by the tripod. The system's workspace dimensions on position, velocity and acceleration level are listed in Table 2.2 . [51]

2.4 Simulator Kinematics

Referring to Figure 2.3, the *position vector* $\boldsymbol{r}_{IP}$ from the FoR I to the point P on the motion platform N can be described using vector notation from Table 2.1,

$$ {}_I\boldsymbol{r}_{IP} = {}_I\boldsymbol{r}_{IN} + {}_I\boldsymbol{R}_N(\boldsymbol{\beta})\, {}_N\boldsymbol{r}_{NP} \tag{2.3}$$

where ${}_I\boldsymbol{R}_N$ defines a rotation from the FoR N to the inertial FoR I with the angle parameters $\boldsymbol{\beta}$. The FoR I is spatially fixed to the surroundings and the FoR N is fixed to the motion platform. For the sake of simplicity, $\boldsymbol{\beta}$ is not written in the following formulations.

The absolute *velocity* of the point P can be expressed by a time derivative of equation (2.3)

Table 2.2: Tripod's and hexapod's workspace limitations on position, velocity and acceleration level [77].

Tripod					
x_T	± 0.80 m	$\dot{x}_T$	1.7 m/s	$\ddot{x}_T$	12 m/s^2
y_T	± 0.75 m	$\dot{y}_T$	1.5 m/s	$\ddot{y}_T$	10 m/s^2
ψ_T	± 25 °	$\dot{\psi}_T$	165 °/s	$\ddot{\psi}_T$	900 °/s^2
Hexapod					
x_H	± 0.28 m	$\dot{x}_H$	2.0 m/s	$\ddot{x}_H$	25 m/s^2
y_H	± 0.25 m	$\dot{y}_H$	1.7 m/s	$\ddot{y}_H$	25 m/s^2
z_H	± 0.22 m	$\dot{z}_H$	1.6 m/s	$\ddot{z}_H$	25 m/s^2
φ_H	± 20 °	$\dot{\varphi}_H$	135 °/s	$\ddot{\varphi}_H$	2500 °/s^2
θ_H	± 20 °	$\dot{\theta}_H$	130 °/s	$\ddot{\theta}_H$	2000 °/s^2
ψ_H	± 20 °	$\dot{\psi}_H$	135 °/s	$\ddot{\psi}_H$	3000 °/s^2

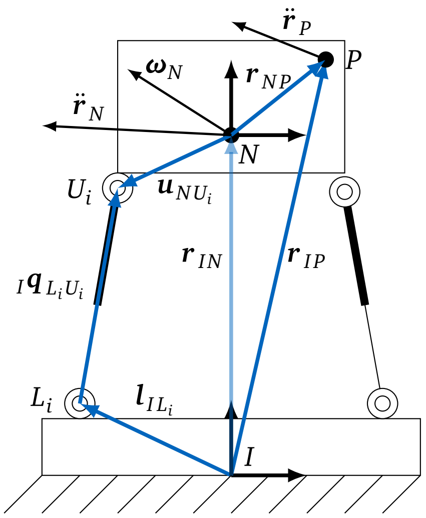


Figure 2.3: Relative kinematics of a rigid motion platform and kinematic chain of an actuator i of a parallel robot structure.

in the inertial FoR I

$$ {}_I\boldsymbol{v}_{IP} = \frac{\mathrm{d}_I\boldsymbol{r}_{IP}}{\mathrm{d}t} = {}_I\dot{\boldsymbol{r}}_{IN} + {}_I\dot{\boldsymbol{R}}_N\,{}_N\boldsymbol{r}_{NP} + {}_I\boldsymbol{R}_N\,{}_N\dot{\boldsymbol{r}}_{NP} \tag{2.4} $$

The time derivative of equation (2.4) yields the absolute *acceleration* of the point P

$$ {}_I\boldsymbol{a}_{IP} = \frac{\mathrm{d}_I\boldsymbol{v}_{IP}}{\mathrm{d}t} = {}_I\ddot{\boldsymbol{r}}_{IN} + {}_I\ddot{\boldsymbol{R}}_N\,{}_N\boldsymbol{r}_{NP} + 2\,{}_I\dot{\boldsymbol{R}}_N\,{}_N\dot{\boldsymbol{r}}_{NP} + {}_I\boldsymbol{R}_N\,{}_N\ddot{\boldsymbol{r}}_{NP}. \tag{2.5} $$

A left multiplication of equation (2.5) with the rotation matrix ${}_N\boldsymbol{R}_I$ leads to a transformation of the acceleration from the FoR I to the relative FoR N

$$ \begin{aligned} {}_N\boldsymbol{a}_{IP} = {}_N\boldsymbol{R}_I\,\frac{\mathrm{d}_I\boldsymbol{v}_{IP}}{\mathrm{d}t} &= {}_N\ddot{\boldsymbol{r}}_{IN} + {}_N\boldsymbol{R}_I\,{}_I\ddot{\boldsymbol{R}}_N\,{}_N\boldsymbol{r}_{NP} + 2\,{}_N\boldsymbol{R}_I\,{}_I\dot{\boldsymbol{R}}_N\,{}_N\dot{\boldsymbol{r}}_{NP} + {}_N\ddot{\boldsymbol{r}}_{NP} \\ &= \underbrace{{}_N\ddot{\boldsymbol{r}}_{IN}}_{(1)} + \underbrace{{}_N\dot{\tilde{\boldsymbol{\omega}}}\,{}_N\boldsymbol{r}_{NP}}_{(2)} + \underbrace{{}_N\tilde{\boldsymbol{\omega}}\,{}_N\tilde{\boldsymbol{\omega}}\,{}_N\boldsymbol{r}_{NP}}_{(3)} + \underbrace{2\,{}_N\tilde{\boldsymbol{\omega}}\,{}_N\dot{\boldsymbol{r}}_{NP}}_{(4)} + \underbrace{{}_N\ddot{\boldsymbol{r}}_{NP}}_{(5)} \end{aligned} \tag{2.6} $$

with the abbreviations:

$$ {}_N\boldsymbol{R}_I\,{}_I\ddot{\boldsymbol{R}}_N = {}_N\dot{\tilde{\boldsymbol{\omega}}} + {}_N\tilde{\boldsymbol{\omega}}\,{}_N\tilde{\boldsymbol{\omega}}, \qquad {}_N\boldsymbol{R}_I\,{}_I\dot{\boldsymbol{R}}_N = {}_N\tilde{\boldsymbol{\omega}} $$

and the tilde-operator $\tilde{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$

The terms in equation (2.6) are categorized in:

(1) absolute acceleration of the point N,

(2) tangential acceleration,

(3) centripetal acceleration,

(4) Coriolis acceleration and

(5) relative acceleration. [20, 134]

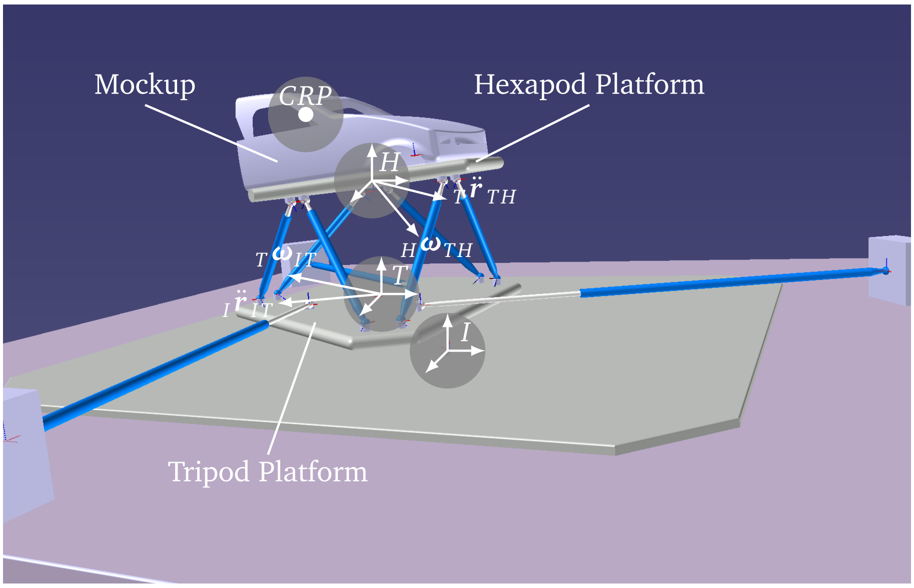

Mockup
CRP
Hexapod Platform
H
$_T\ddot{\boldsymbol{r}}_{TH}$
$_T\boldsymbol{\omega}_{IT}$
$_H\boldsymbol{\omega}_{TH}$
T
$_I\ddot{\boldsymbol{r}}_{IT}$
I
Tripod Platform

Figure 2.4: Schematic structure of the nine DoFs motion system. A coordinate system is introduced for each motion subsystem to describe the kinematics.

To describe the kinematics of the DiM simulator, three FoRs are introduced, as depicted in Figure 2.4. The inertial FoR I is earth-fixed, and the FoR T and FoR H are anchored at the middle position of the tripod's and hexapod's platform, respectively. The *CRP* point represents the *cockpit reference point* and is rigidly fixed to the cockpit, where the driver's head is assumed to be located.
The transformation from the FoR I to the FoR T is a rotation about the $_Iz$–axis

$$_T\boldsymbol{R}_I(\psi_{IT}) = \begin{bmatrix} \cos\psi_{IT} & \sin\psi_{IT} & 0 \\ -\sin\psi_{IT} & \cos\psi_{IT} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

with the rotational DoF $\boldsymbol{\beta}_{IT} = [0\;0\;\psi_{IT}]^T$ of the tripod. Cardan angles $\boldsymbol{\beta}_{TH} = [\varphi_{TH}\;\theta_{TH}\;\psi_{TH}]^T$ define a Cardan transformation from the FoR T to the FoR H

$$_H\boldsymbol{R}_T(\varphi_{TH}, \theta_{TH}, \psi_{TH}) = \boldsymbol{R}_x(\varphi_{TH})\,\boldsymbol{R}_y(\theta_{TH})\,\boldsymbol{R}_z(\psi_{TH}) \tag{2.8}$$

with the rotation matrices defined in equation (2.2). Relevant equations for using the Cardan angles are listed in Appendix A.

The resulting angular velocities $_H\boldsymbol{\omega}_{IH}$ are calculated by

$$\begin{aligned} _H\boldsymbol{\omega}_{IH} &= {}_H\boldsymbol{\omega}_{IT} + {}_H\boldsymbol{\omega}_{TH} \qquad (2.9) \\ &= \begin{bmatrix} \dot{\varphi}_{TH} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{R}_x(\varphi_{TH}) \begin{bmatrix} 0 \\ \dot{\theta}_{TH} \\ 0 \end{bmatrix} + \boldsymbol{R}_x(\varphi_{TH})\boldsymbol{R}_y(\theta_{TH}) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_{TH} + \dot{\psi}_{IT} \end{bmatrix} \\ &= {}_H\boldsymbol{J}_{\omega}(\varphi_{TH}, \theta_{TH}) \left(\dot{\boldsymbol{\beta}}_{IT} + \dot{\boldsymbol{\beta}}_{TH}\right) \qquad (2.10) \end{aligned}$$

with the Jacobian matrix $_H\boldsymbol{J}_{\omega}$.

The following Section describes the steps to estimate the resulting accelerations $_H\boldsymbol{a}_{CRP}$ that act on the driver at the CRP. Similar to the relative kinematics in equations (2.3)-(2.5), the position vector $\boldsymbol{r}_{I,CRP}$ from the FoR I to the CRP is described by

$$\begin{aligned} {}_{I}\boldsymbol{r}_{I,CRP} &= {}_{I}\boldsymbol{r}_{IT} + {}_{I}\boldsymbol{r}_{TH} + {}_{I}\boldsymbol{r}_{H,CRP} && (2.11)\\ &= {}_{I}\boldsymbol{r}_{IT} + {}_{I}\boldsymbol{R}_{T}\,{}_{T}\boldsymbol{r}_{TH} + {}_{I}\boldsymbol{R}_{H}\,{}_{H}\boldsymbol{r}_{H,CRP}. && (2.12)\end{aligned}$$

The velocity of the CRP is expressed by a time derivative of equation (2.12)

$${}_{I}\boldsymbol{v}_{I,CRP} = {}_{I}\dot{\boldsymbol{r}}_{IT} + {}_{I}\dot{\boldsymbol{R}}_{T}\,{}_{T}\boldsymbol{r}_{TH} + {}_{I}\boldsymbol{R}_{T}\,{}_{T}\dot{\boldsymbol{r}}_{TH} + {}_{I}\dot{\boldsymbol{R}}_{H}\,{}_{H}\boldsymbol{r}_{H,CRP}. \qquad (2.13)$$

The time derivative of equation (2.13) yields the acceleration at the CRP

$$\begin{aligned} {}_{I}\boldsymbol{a}_{I,CRP} &= {}_{I}\ddot{\boldsymbol{r}}_{I,CRP} + {}_{I}\boldsymbol{g} && (2.14)\\ &= {}_{I}\ddot{\boldsymbol{r}}_{IT} + {}_{I}\ddot{\boldsymbol{R}}_{T}\,{}_{T}\boldsymbol{r}_{TH} + 2\,{}_{I}\dot{\boldsymbol{R}}_{T}\,{}_{T}\dot{\boldsymbol{r}}_{TH} + {}_{I}\boldsymbol{R}_{T}\,{}_{T}\ddot{\boldsymbol{r}}_{TH} + {}_{I}\ddot{\boldsymbol{R}}_{H}\,{}_{H}\boldsymbol{r}_{H,CRP} + {}_{I}\boldsymbol{g} && (2.15)\end{aligned}$$

with the gravitational force ${}_{I}\boldsymbol{g} = [0,\ 0,\ g]^{T}$. A left multiplication of equation (2.15) with the rotation matrix ${}_{H}\boldsymbol{R}_{I}$ leads to the resulting accelerations ${}_{H}\boldsymbol{a}_{CRP}$ that act on the driver at the CRP

$$\begin{aligned} {}_{H}\boldsymbol{a}_{CRP} =& {}_{H}\boldsymbol{R}_{I}\,{}_{I}\boldsymbol{a}_{I,CRP} && (2.16)\\ =& {}_{H}\boldsymbol{R}_{I}\,{}_{I}\ddot{\boldsymbol{r}}_{IT} + {}_{H}\boldsymbol{R}_{T}\,{}_{T}\ddot{\boldsymbol{r}}_{TH} \\ & + 2\,{}_{H}\boldsymbol{R}_{T}\,{}_{T}\tilde{\boldsymbol{\omega}}_{IT}\,{}_{T}\dot{\boldsymbol{r}}_{TH} + {}_{H}\boldsymbol{R}_{T}\left({}_{T}\dot{\tilde{\boldsymbol{\omega}}}_{IT} + {}_{T}\tilde{\boldsymbol{\omega}}_{IT}\,{}_{T}\tilde{\boldsymbol{\omega}}_{IT}\right){}_{T}\boldsymbol{r}_{TH} \\ & + \left({}_{H}\dot{\tilde{\boldsymbol{\omega}}}_{IH} + {}_{H}\tilde{\boldsymbol{\omega}}_{IH}\,{}_{H}\tilde{\boldsymbol{\omega}}_{IH}\right){}_{H}\boldsymbol{r}_{H,CRP} + {}_{H}\boldsymbol{R}_{I}\,{}_{I}\boldsymbol{g} && (2.17)\end{aligned}$$

with the tripod's angular velocity ${}_{T}\boldsymbol{\omega}_{IT} = \dot{\boldsymbol{\beta}}_{IT}$. [52]
Throughout this work, the CRP is set to

$${}_{H}\boldsymbol{r}_{H,CRP} = [-0.435\ 0.2815\ 1.04]^{T}\ [\mathrm{m}]. \qquad (2.18)$$

2.5 Workspace and Actuator Space

The *workspace* of a manipulator defines the accessible space of the TCP with respect to its actuator limitations [159].
The TCP P of a manipulator can be expressed in *workspace coordinates* $\boldsymbol{w}_{S}$ and in *actuator coordinates* $\boldsymbol{q}_{S}$. Referring to Figure 2.3, workspace coordinates use global coordinates (translation $\boldsymbol{r}_{P}$ and orientation $\boldsymbol{\beta}$) to define the position and orientation of the point P. Actuator coordinates $\boldsymbol{q}_{S}$ are defined in relation to a respective actuator i and include its state. Using the direct or inverse kinematics, a transformation between workspace coordinates $\boldsymbol{w}_{S}$ and actuator coordinates $\boldsymbol{q}_{S}$ can be performed. [159]
The following Section describes the transformation between workspace and actuator space for parallel motion systems, where

- Inverse Kinematics: $\boldsymbol{w}_{S} \rightarrow \boldsymbol{q}_{S}$,
- Direct Kinematics: $\boldsymbol{q}_{S} \rightarrow \boldsymbol{w}_{S}$.

For the description of the serial kinematics, it is referred to [29, 154, 159].

2.5.1 Parallel Kinematics

The closed-loop vector chain, as shown in Figure 2.3 for a single actuator i, including the CRP (point P in Figure 2.3), is given by

$$ {}_I\boldsymbol{q}^{CRP}_{L_iU_i} = -{}_I\boldsymbol{l}_{IL_i} + {}_I\boldsymbol{r}_{CRP} + {}_I\boldsymbol{R}_N(\boldsymbol{\beta})\,({}_N\boldsymbol{u}_{NU_i} - {}_N\boldsymbol{r}_{N,CRP}) \tag{2.19} $$

where ${}_I\boldsymbol{l}_{IL_i}$ and ${}_N\boldsymbol{u}_{NU_i}$ are the lower and upper vector from the platform FoR to the Cardan joints. The parallel chain which directly connects the origins of the FoR I and the FoR N is [115]

$$ {}_I\boldsymbol{q}^{IN}_{L_iU_i} = -{}_I\boldsymbol{l}_{IL_i} + {}_I\boldsymbol{r}_{IN} + {}_I\boldsymbol{R}_N(\boldsymbol{\beta})\,{}_N\boldsymbol{u}_{NU_i}. \tag{2.20} $$

The workspace of a parallel motion system is composed of

$$ \boldsymbol{w}^{CRP}_S = [{}_I\boldsymbol{r}_{CRP},\ \boldsymbol{\beta}], \qquad \boldsymbol{w}^{IN}_S = [{}_I\boldsymbol{r}_{IN},\ \boldsymbol{\beta}], \tag{2.21} $$

depending on whether the CRP is considered. The parameters $\boldsymbol{l}_{IL_i}$, $\boldsymbol{u}_{NU_i}$ and $\boldsymbol{r}_{N,CRP}$ are known from the geometry of the motion system.

Inverse Kinematics

The *inverse kinematics* of a parallel structure can – in contrast to serial robot systems – be solved analytically. Herein, the absolute value of the vector ${}_I\boldsymbol{q}_{L_iU_i}$ yields the length of the actuator i as a nonlinear function of the generalized workspace coordinates $\boldsymbol{w}_S$ with

$$ q_{S,i} = \sqrt{{}_I\boldsymbol{q}_{L_iU_i}{}^T\ {}_I\boldsymbol{q}_{L_iU_i}} = ||{}_I\boldsymbol{q}_{L_iU_i}|| = f_i(\boldsymbol{w}_S). \tag{2.22} $$

The actuators' velocities are obtained by the time derivative of equation (2.22)

$$ \dot{q}_{S,i} = \frac{\mathrm{d}q_{S,i}}{\mathrm{d}t} = \boldsymbol{J}_i\ \dot{\boldsymbol{w}}_S. \tag{2.23} $$

The derivation of the matrix/vector form $\dot{q}_{S,i} = \boldsymbol{J}_i\ \dot{\boldsymbol{w}}_S$ in equation (2.23) is exemplarily shown using the time derivative of equation (2.20)

$$ \begin{aligned} {}_I\dot{\boldsymbol{q}}_{S,i} &= {}_I\dot{\boldsymbol{r}}_{IN} + {}_I\dot{\boldsymbol{R}}_N\ {}_N\boldsymbol{u}_{NU_i} \\ &= {}_I\dot{\boldsymbol{r}}_{IN} + \underbrace{{}_I\dot{\boldsymbol{R}}_N\ {}_N\boldsymbol{R}_I}_{{}_I\tilde{\boldsymbol{\omega}}}\ {}_I\boldsymbol{R}_N\ {}_N\boldsymbol{u}_{NU_i} \\ &= {}_I\dot{\boldsymbol{r}}_{IN} + {}_I\tilde{\boldsymbol{\omega}}\ {}_I\boldsymbol{R}_N\ {}_N\boldsymbol{u}_{NU_i} \end{aligned} \tag{2.24} $$

Introducing the actuator's unit vector ${}_I\bar{\boldsymbol{q}}_{S,i}$ yields the following relation

$$ {}_I\dot{\boldsymbol{q}}_{S,i} = {}_I\bar{\boldsymbol{q}}_{S,i}\ \dot{q}_{S,i} \qquad \text{with} \qquad \dot{q}_{S,i} = {}_I\bar{\boldsymbol{q}}^T_{S,i}\ {}_I\dot{\boldsymbol{q}}_{S,i}. \tag{2.25} $$

Inserting equation (2.24) in equation (2.25) yields

$$ \dot{q}_{S,i} = {}_I\bar{\boldsymbol{q}}^T_{S,i}\ {}_I\dot{\boldsymbol{r}}_{IN} + {}_I\bar{\boldsymbol{q}}^T_{S,i}({}_I\tilde{\boldsymbol{\omega}}\ {}_I\boldsymbol{R}_N\ {}_N\boldsymbol{u}_{NU_i}). \tag{2.26} $$

Transforming equation (2.26) to vector/matrix form, one obtains

$$ \dot{q}_{S,i} = \begin{bmatrix} {}_I\bar{\boldsymbol{q}}^T_{S,i} & {}_I\bar{\boldsymbol{q}}^T_{S,i}\ {}_I\tilde{\boldsymbol{u}}^T_{NU_i} \end{bmatrix} \begin{bmatrix} {}_I\dot{\boldsymbol{r}}_{IN} \\ {}_I\boldsymbol{\omega} \end{bmatrix} \tag{2.27} $$

with ${}_I\boldsymbol{u}_{NU_i} = \;{}_I\boldsymbol{R}_N\;{}_N\boldsymbol{u}_{NU_i}$ and ${}_I\tilde{\boldsymbol{\omega}}\;{}_I\boldsymbol{u}_{NU_i} = {}_I\tilde{\boldsymbol{u}}_{NU_i}^T\;{}_I\boldsymbol{\omega}$.
The extension of equation (2.27) to all actuators n yields the desired form

$$\underbrace{\begin{bmatrix} \dot{q}_{S,1} \\ \vdots \\ \dot{q}_{S,n} \end{bmatrix}}_{\dot{\boldsymbol{q}}_S} = \underbrace{\begin{bmatrix} {}_I\bar{\boldsymbol{q}}_{S,1}^T & {}_I\bar{\boldsymbol{q}}_{S,1}^T{}_I\tilde{\boldsymbol{u}}_{NU_1}^T \\ \vdots & \vdots \\ {}_I\bar{\boldsymbol{q}}_{S,n}^T & {}_I\bar{\boldsymbol{q}}_{S,n}^T{}_I\tilde{\boldsymbol{u}}_{NU_n}^T \end{bmatrix}}_{\boldsymbol{J}} \underbrace{\begin{bmatrix} {}_I\dot{\boldsymbol{r}}_{IN} \\ {}_I\boldsymbol{\omega} \end{bmatrix}}_{\dot{\boldsymbol{w}}_S} \tag{2.28}$$

with the Jacobian matrix $\boldsymbol{J}$.

The actuators' accelerations

$$\ddot{q}_{S,i} = \frac{\mathrm{d}^2 q_{S,i}}{\mathrm{d}^2 t} \qquad \text{for } \; i = \text{actuator number} \tag{2.29}$$

are the second time derivatives of equation (2.22). Analogously to the actuator velocities, equation (2.29) can be transformed to a vector/matrix representation. Therefore, equation (2.24) is derived with respect to time

$$\begin{aligned} {}_I\ddot{\boldsymbol{q}}_{S,i} &= {}_I\ddot{\boldsymbol{r}}_{IN} \;+\; {}_I\ddot{\boldsymbol{R}}_N\;{}_N\boldsymbol{u}_{NU_i} \\ &= \;{}_I\ddot{\boldsymbol{r}}_{IN} \;+\; {}_I\boldsymbol{R}_N\;{}_N\boldsymbol{R}_I\;{}_I\ddot{\boldsymbol{R}}_N\;{}_N\boldsymbol{R}_I\;{}_I\boldsymbol{R}_N\;{}_N\boldsymbol{u}_{NU_i} \\ &= \;{}_I\ddot{\boldsymbol{r}}_{IN} \;+\; {}_I\boldsymbol{R}_N\,({}_N\dot{\tilde{\boldsymbol{\omega}}} \;+\; {}_N\tilde{\boldsymbol{\omega}}\;{}_N\tilde{\boldsymbol{\omega}})\;{}_N\boldsymbol{R}_I\;{}_I\boldsymbol{u}_{NU_i} \\ &= \;{}_I\ddot{\boldsymbol{r}}_{IN} \;+\; ({}_I\dot{\tilde{\boldsymbol{\omega}}} \;+\; {}_I\tilde{\boldsymbol{\omega}}\;{}_I\tilde{\boldsymbol{\omega}})\;{}_I\boldsymbol{u}_{NU_i}. \end{aligned} \tag{2.30}$$

Applying the unit vector in the form $\ddot{q}_{S,i} = {}_I\bar{\boldsymbol{q}}_{S,i}^T\;{}_I\ddot{\boldsymbol{q}}_{S,i}$, yields

$$\ddot{q}_{S,i} = \;{}_I\bar{\boldsymbol{q}}_{S,i}^T\;{}_I\ddot{\boldsymbol{r}}_{IN} \;+\; {}_I\bar{\boldsymbol{q}}_{S,i}^T\;{}_I\dot{\tilde{\boldsymbol{\omega}}}\;{}_I\boldsymbol{u}_{NU_i} \;+_I\bar{\boldsymbol{q}}_{S,i}^T\;{}_I\tilde{\boldsymbol{\omega}}\;{}_I\tilde{\boldsymbol{\omega}}\;{}_I\boldsymbol{u}_{NU_i}. \tag{2.31}$$

Employing equation (2.31) to all actuators n yields

$$\underbrace{\begin{bmatrix} \ddot{q}_{S,1} \\ \vdots \\ \ddot{q}_{S,n} \end{bmatrix}}_{\ddot{q}_S} = \boldsymbol{J}\;\ddot{\boldsymbol{w}}_S \;+\; \begin{bmatrix} {}_I\bar{\boldsymbol{q}}_{S,1}^T\;{}_I\tilde{\boldsymbol{\omega}}\;{}_I\tilde{\boldsymbol{\omega}}\;{}_I\boldsymbol{u}_{NU_1} \\ \vdots \\ {}_I\bar{\boldsymbol{q}}_{S,n}^T\;{}_I\tilde{\boldsymbol{\omega}}\;{}_I\tilde{\boldsymbol{\omega}}\;{}_I\boldsymbol{u}_{NU_n} \end{bmatrix} \tag{2.32}$$

with the Jacobian matrix $\boldsymbol{J}$ from equation (2.28).

Direct Kinematics

The *direct kinematics* yields the solution of the nonlinear system of equations (2.22)

$$q_{S,i} = f_i(\boldsymbol{w}_S), \qquad \forall\; q_{S,i}. \tag{2.33}$$

The solution is in general not unique. Still, there are ways to deal with this problem by using numerical approaches. For a fast numerical analysis, it is important to get the best possible information of the current robot's pose as a starting point since the direct kinematics can have several solutions. On basis of the nonlinear system of equations

$$\boldsymbol{0} = \begin{bmatrix} q_{S,1} \;-\; f_1(\boldsymbol{w}_S) \\ \vdots \\ q_{S,n} \;-\; f_n(\boldsymbol{w}_S) \end{bmatrix} \;=\; \boldsymbol{q}_S \;-\; \boldsymbol{F}(\boldsymbol{w}_S) \;=\; \boldsymbol{G}(\boldsymbol{w}_S), \tag{2.34}$$

the Newton-Raphson method for root finding can be applied [13]. The vectors $\boldsymbol{q}_S$ and $\boldsymbol{F}$ in equation (2.34) include the parallel vector chains from equation (2.22) for all n actuators. For a workspace starting value $\boldsymbol{w}_{S_0}$ and a given actuator configuration $\boldsymbol{q}_S$, an iteration step k is expressed by

$$\begin{aligned} \boldsymbol{w}_S[k+1] &= \boldsymbol{w}_S[k] - \left(\frac{\partial \boldsymbol{G}(\boldsymbol{w}_S[k])}{\partial \boldsymbol{w}_S[k]}\right)^{-1} \boldsymbol{G}(\boldsymbol{w}_S[k]) \\ &= \boldsymbol{w}_S[k] - \boldsymbol{J}^{-1}(\boldsymbol{w}_S[k])\ \boldsymbol{G}(\boldsymbol{w}_S[k]) \end{aligned} \tag{2.35}$$

whereas $\boldsymbol{J}(\boldsymbol{w}_S[k]) = \frac{\partial \boldsymbol{G}(\boldsymbol{w}_S[k])}{\partial \boldsymbol{w}_S[k]}$ is the Jacobian matrix of the system, already derived in equation (2.28). The stopping criterion of the iterative algorithm is fulfilled for

$$||\underbrace{\boldsymbol{q}_S[k] - \boldsymbol{F}(\boldsymbol{w}_S[k])}_{\tau_{res}}|| < \varepsilon \tag{2.36}$$

where ε is a fixed error threshold. The residuum τ_{res} means that equation (2.34) is only satisfied to a certain extent. [115]

2.5.2 Singularities

Singular configurations of parallel manipulators can lead to uncontrollable operating configurations which can damage the parallel structure [146]. Singularities can be distinguished in *boundary singularities* (outstretched or retracted manipulator) and *internal singularities* (inside reachable workspace) [159].

In [75], three different kinds of singularities are described which are combinations of boundary and internal singularities. To derive and to better differentiate the three cases, equation (2.27) is transformed to

$$q_{S,i}\,\dot{q}_{S,i} = {}_I\boldsymbol{q}_{S,i}^T\,{}_I\dot{\boldsymbol{r}}_{IN} + {}_I\boldsymbol{q}_{S,i}^T\ {}_I\tilde{\boldsymbol{u}}_{NU_i}^T\ {}_I\boldsymbol{\omega} \qquad \text{with} \quad {}_I\bar{\boldsymbol{q}}_{S,i}^T = \frac{{}_I\boldsymbol{q}_{S,i}}{q_{S,i}} \tag{2.37}$$

Applying equation (2.37) to all actuators n yields a differential equation in the form

$$\boldsymbol{q}_{S,diag}\ \dot{\boldsymbol{q}}_S - \boldsymbol{J}^*\ \dot{\boldsymbol{w}}_S = \boldsymbol{0} \tag{2.38}$$

with the diagonal matrix of the actuator strokes $\boldsymbol{q}_{S,diag} = \mathrm{diag}(q_{S,1},\ldots,q_{S,n})$ and the modified Jacobian matrix

$$\boldsymbol{J}^* = \begin{bmatrix} {}_I\boldsymbol{q}_{S,1}^T & {}_I\boldsymbol{q}_{S,1}^T{}_I\tilde{\boldsymbol{u}}_{NU_1}^T \\ \vdots & \vdots \\ {}_I\boldsymbol{q}_{S,n}^T & {}_I\boldsymbol{q}_{S,n}^T{}_I\tilde{\boldsymbol{u}}_{NU_n}^T \end{bmatrix}. \tag{2.39}$$

The transformation of equation (2.27) yields the advantage to show the actuator strokes $q_{S,i}$, the actuator velocities $\dot{q}_{S,i}$ and the workspace coordinates $\boldsymbol{w}_S$ in a separate form which helps to distinguish between the following three singularities:

In the first case (boundary singularity), the matrix $\boldsymbol{q}_{S,diag}$ shows a drop in rank, which means that an actuated joint with a velocity vector unequal zero does not lead to a motion at the TCP. In other words, one can find non-zero actuator velocities $\dot{q}_{S,i}$, for which the workspace velocities $\dot{\boldsymbol{w}}_S$ will be zero. This is not possible for a non-redundant motion system. It can occur, for example, when an actuator $q_{S,i}$ reaches its minimum stroke $q_{S,i,min}$, as schematically depicted on the left plot in Figure 2.5. In the second configuration (internal singularity), the

matrix $\boldsymbol{J}^*$ shows a drop in rank, which means that the TCP is moveable although all actuators are locked. In other words, there are configurations where non-zero workspace velocities $\dot{\boldsymbol{w}}_S$ do not yield an actuator velocity $\dot{q}_{S,i}$. An example is shown in the right plot of Figure 2.5. In the third case, both matrices $\boldsymbol{q}_{S,diag}$ and $\boldsymbol{J}^*$ exhibit a drop in rank. In these configurations, the parallel structure can undergo finite motions although its actuators are locked or in which a finite actuator motion does not lead to motions of the TCP.
Boundary singularities have to be avoided by the MCA or a workspace supervision module which checks the adherence of the minimum and maximum actuator configurations. Internal singularities occur when

$$\det(\boldsymbol{J}^*) = 0 \tag{2.40}$$

which indicates a drop in rank of the matrix $\boldsymbol{J}^*$. The considered manipulators throughout the thesis do not exhibit internal singularities for which reason only workspace limitations on position, velocity and acceleration level of the motion systems are taken into account.

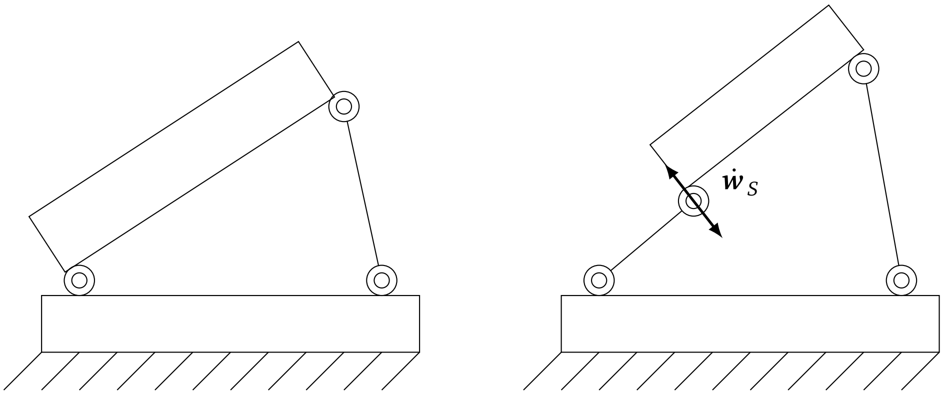


Figure 2.5: Schematic examples of singularities for parallel structures. Left: Boundary singularity. Right: Internal singularity.

2.6 Open-Loop and Closed-Loop Interaction Types

In this work, two types of control loops are distinguished in the field of driving simulation, which are schematically demonstrated in Figure 2.6. In the upper part of Figure 2.6, the closed-loop approach is shown, where the driver is actively driving and interacting with the driving simulator. In the closed-loop process, a driving dynamics model is used to model the dynamics of a virtual vehicle which is controlled by inputs from the driver. The MCA calculates the inputs for the simulator. The generated simulator motions are perceived by the driver which lead to further driver inputs. On the lower part of Figure 2.6, an open-loop structure is depicted, where the driver is located at the end of the process chain. Thus, the driver sits passively in the vehicle without any control possibilities and is driven through a pre-defined and pre-recorded manoeuvre. The simulator motions are estimated in an offline process by using pre-recorded driving dynamics data as input parameters for the open-loop MCA. [52]

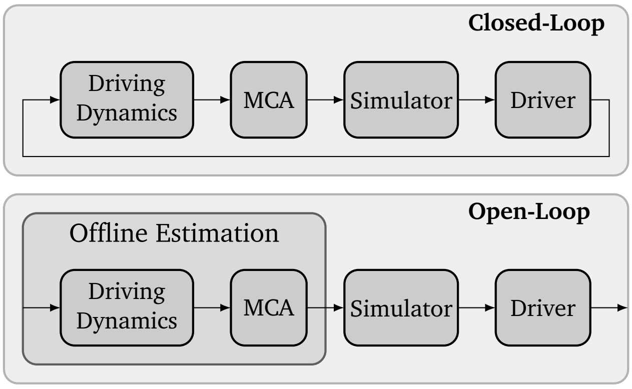


Figure 2.6: In a closed-loop simulation, the driver is actively driving. Whereas, in an open-loop simulation, the driver sits passively in the simulator mockup.

2.7 Motion Perception

Immersing humans to virtual environments, Motion Cueing intends to realize as many realistic vestibular stimuli to the driver as necessary. As workspace and technical capabilities are limited, it is important to be aware of the human perception channels. Thus, thresholds of the human perception system can be exploited and only perceivable cues can be provided by the simulator. Weaknesses of the perception system can be used to create an illusion of motions. The following Chapter describes functions of relevant perception organs. The information is taken up in perception model approaches which can be applied to an MCA.

2.7.1 Human Perception Systems

Perception of motion and position is based on different sensory channels which can be divided into:

- auditory perception,
- visual perception,
- somatosensory perception and
- vestibular perception.

It may occur that perceived stimuli from various sensory channels show a disparate, incomplete or erroneous behaviour. Thus, an integration process weights the stimuli according to the advantages of each perception system. Motion cues can exploit this behaviour by adjusting motion to the respective perception system. Severe incongruencies between the motion channels can however cause motion sickness. The following subsections discuss the different perception systems and introduce the vestibular organ in more detail as it represents the most important motion perception channel. [67, 183]

Auditory Perception

Auditory motion perception implies all vehicle and road noises such as sound from the engine, tyres rolling and wind. In [174], three different types of cues are described:

- intensity cues,

- binaural cues and
- Doppler effect.

Intensity cues focus on moving auditory emitters according to changes in sound pressure level. Binaural cues reproduce interaural time and level differences (ITD and ILD) at the receiving ear. [174] ITD specify the difference in arrival time of sound between the left and right ear. ILD implies the perceived level difference in sound between left and right ear. Both mechanisms are responsible for the localization of sound sources. [63] Doppler effect considers perceived frequency shifts between listener and sound source [174].

Visual Perception

In [183], it is described that exclusive visual stimuli are not a sufficient motion perception channel as humans need further information in terms of a fixed reference point or feedback from an additional perception system. Without any further information, the perception system cannot differentiate between own and observed motions. Velocity perception depends on the surroundings which means that velocity can be sensed more accurately in an environment with distinctive reference points. The visual system is especially sensitive to position and velocity perception whereas vestibular perception is primarily sensitive to acceleration perception. In [31], it is shown that the visual perception is more reliable compared to the vestibular one.

Somatosensory Perception

The somatosensory system gives feedback about position and motion of our body parts (proprioception), through the stimulation of skin (haptics), muscles, sinews and joints. Along with the vestibular system, the somatosensory system detects static and dynamic forces. The perceived stimuli allow conclusions to be drawn about accelerations which are working on the body. Mechanical changes of body parts induce these stimuli. Joint positions and passive stretches of muscles give feedback about body positions and length variations of body parts. Receptors under the skin surface perceive changes in pressure. [138] There are two different types of receptors. First, muscle spindles detect the muscles' length and the changing rate. Second, Golgi Tendon organs detect muscle contractions in reaction to external forces. [171] With regard to driving simulation, the somatosensory system detects pressure changes between back, buttocks, legs and vehicle seat as well as between hands and steering wheel. These stimuli give feedback about the vehicle acceleration. Thus, especially the interface between driver and vehicle is important. Additional artificial seat and steering wheel vibrations can simulate the oscillations of the vehicle. Actuators integrated in the seat can conduct forces on the driver, for example during turning manoeuvres. [83]

Vestibular System

The vestibular organ is located in the labyrinth of the inner ear and consists of two macula organs and three semicircular canals per body side. Figure 2.7 shows the labyrinth together with the cochlea. The semicircular canals are grouped into the posterior, anterior and horizontal canal. All five organs exhibit hair cells which are surrounded by a gelatinous mass. The viscous membrane is called cupula in the three semicircular canals and otolith membrane in the macula organs. Hair cells are able to transform mechanical stimuli in chemical and electrical signals. [86, 191]

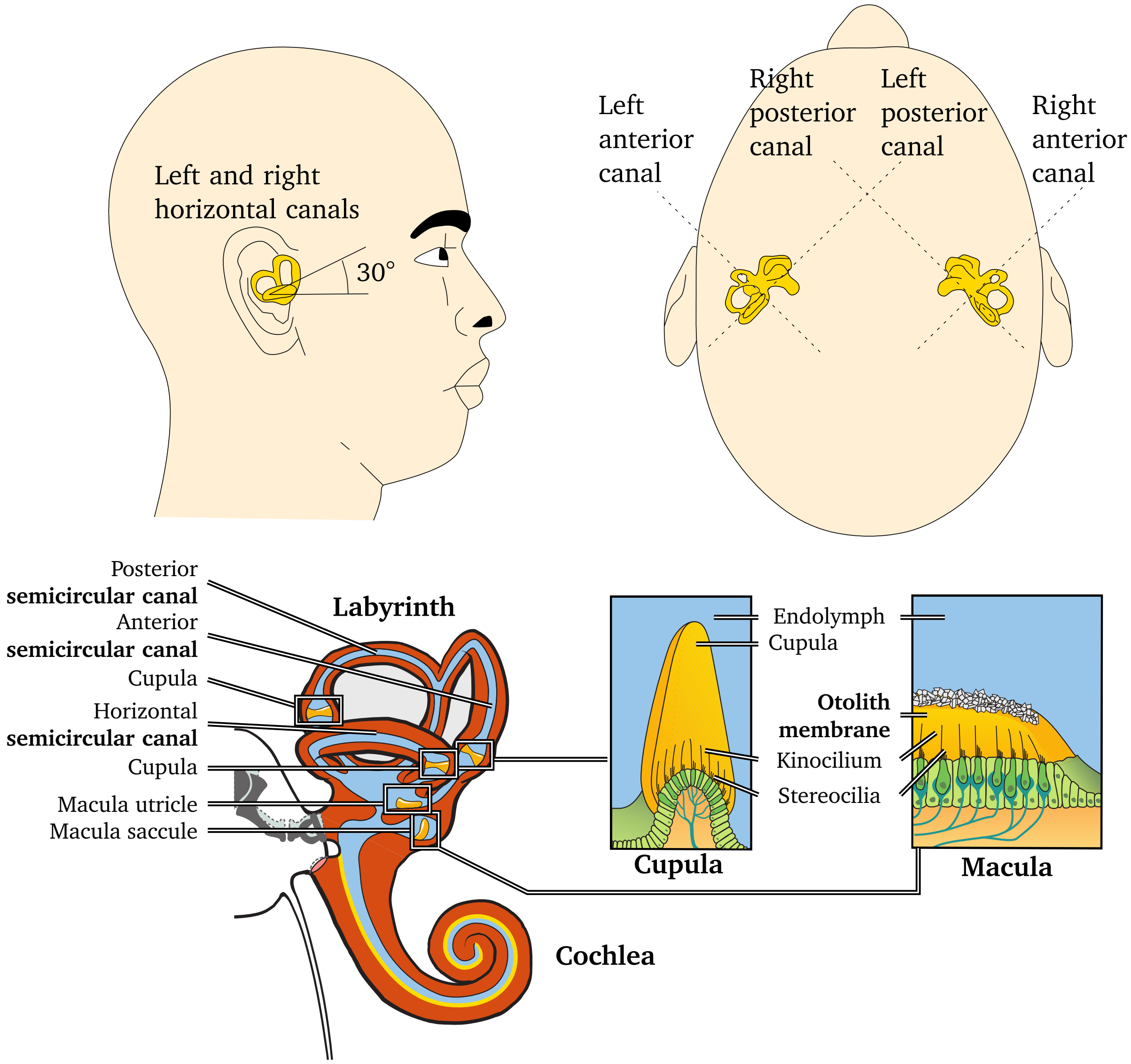


Figure 2.7: Upper figure: Location and orientation of the vestibular organs (based on [169]). Lower figure: Schematic diagram of the vestibular organ which can be classified into the labyrinth and the cochlea. The labyrinth can be subdivided into the cupula and the macula which are shown in detail on the right hand side (based on [191]).

The *semicircular canals* mainly perceive rotational accelerations. The canals are ordered perpendicular to each other in order to detect each of the three dimensions in space. The orientation of the canals to the head is shown in Figure 2.7. Both horizontal canals are tilted by about 30° to the earth-horizontal plane. Left and right anterior canals are inclined about 45° to the frontal plane of the head. The elliptical canals are filled by the fluid endolymph. At one point, the canals are interrupted by the cupula which is anchored to the bony canal wall. An applied rotational motion deflects the cupula in the opposite direction of the head motion. Deviations in inertia between the fluid endolymph and the cupula result in shearing of the sensory hairs. A hair cell exhibits several stereocilia and one kinocilium. In each cupula, the kinocilium is located on the same side next to all the stereocilia. Depending on the direction, a deflection of the stereocilia causes an excitation or inhibition of the hair cells. This motion alters the polarization in the hair cells which can be detected by afferent nerve fibres. [86, 168, 191]

The *otolith membrane*, which is on top of the macula, detects translational accelerations. The organ is divided into the utricle, which primarily perceives motions in the horizontal plane, and the saccule which mainly senses motions in the vertical plane. By this means, translational accelerations in all three directions of space can be detected with only two organs. The otoliths are located directly next to the inner ear with an inclination of about 20° to 30° to the earth-horizontal plane (see Figure 2.7). The structure of the otoliths can be divided into two layers. The otolith membrane is in touch with the fluid endolymph and consists of calcium carbonate crystals embedded in a gelatinous material. The lower gelatinous layer is in contact with the sensory hairs of the macula which is fixed to the skull. An applied acceleration leads to shearing of the sensory hairs whereby the hair cells are stimulated. The hair cells are excited or inhibited depending on the orientation of shearing. This inclination results in a polarization which is detected by the afferent nerve cells. [73, 86, 191]

The otolith system cannot distinguish between accelerations and tilts without any additional information. Figure 2.8 shows two heads of a human. The left one undergoes a linear acceleration $\boldsymbol{a}_x$ in x-direction whereas the right one is tilted by an angle θ around the y-axis. The applied specific force $\boldsymbol{f}_a$ of the left Figure results from the effective force $\boldsymbol{F}_e$ and the accelerated mass m

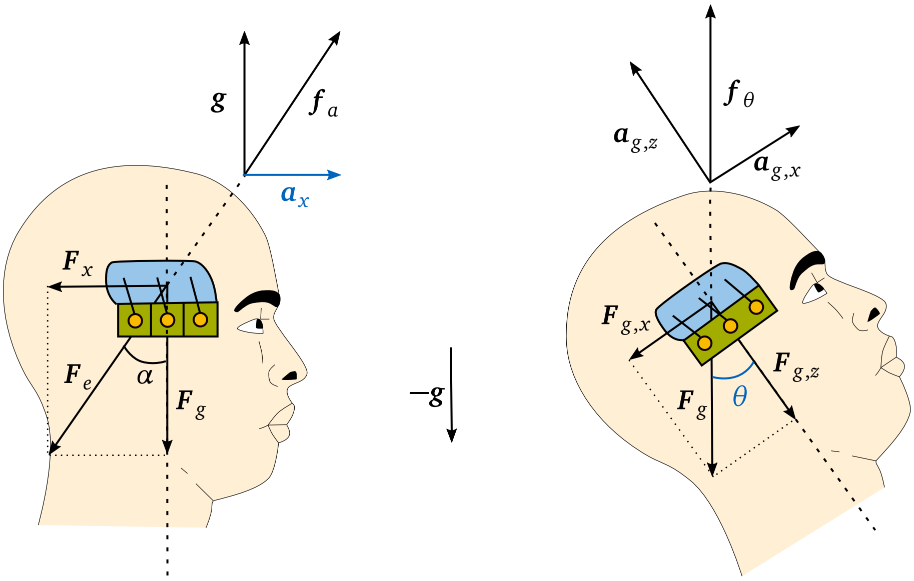


Figure 2.8: Schematic figure of the applied forces under gravity and acceleration. Left: Distribution of forces by an acceleration in x-direction. Right: Distribution of forces by tilting the body by an angle θ. (based on [67])

$$\boldsymbol{f}_a = -\frac{\boldsymbol{F}_e}{m}. \tag{2.41}$$

with

$$\boldsymbol{F}_e = \boldsymbol{F}_g + \boldsymbol{F}_x \tag{2.42}$$

is a superposition of the gravity force $\boldsymbol{F}_g = -m \cdot \boldsymbol{g}$ and the inertia force $\boldsymbol{F}_x = -m \cdot \boldsymbol{a}_x$ which results from a translational acceleration.

The effective force $\boldsymbol{F}_g$ of the right head, which is tilted and non-moving, is composed of the apparent gravity force $F_{g,z} = -m \cdot g \cdot \cos(\theta)$ as well as the apparent inertia force $F_{g,x} =$

$m \cdot g \cdot \sin(\theta)$. The otolith system interprets the rotated position as a translational acceleration. In literature, a maximum tilt angle of 20-30° and a maximum tilting rate of 3 $°/s$ is common before humans perceive the tilt motion [116, 144].
Throughout the thesis, the resulting translational accelerations ${}_H\boldsymbol{a}_{CRP}$ (cf. equation (2.17)) are called *specific forces* $\boldsymbol{f}$ and are described in the relative FoR H of the driver. The simulator reference motions $\boldsymbol{y}_S$ are composed of

$$\boldsymbol{y}_S = \begin{bmatrix} {}_H\boldsymbol{f}_{CRP} \\ {}_H\boldsymbol{\omega}_{CRP} \end{bmatrix} \tag{2.43}$$

with the specific forces ${}_H\boldsymbol{f}_{CRP}$ and the resulting angular velocities ${}_H\boldsymbol{\omega}_{CRP}$.

2.7.2 Perception Thresholds

Perception thresholds describe the minimum physical motions which are detected by the perception system. Driving simulation utilizes these thresholds through non-detectable motions by the driver. In literature, a variety of thresholds is present depending on the measuring methods, the experimental setup and the experimental conduction. In [139], a range for the translational acceleration threshold and the rotational velocity threshold is given to $0.014 - 0.25\,\mathrm{m/s^2}$ and $0.1 - 3.0°/\mathrm{s}$, respectively. Detailed analysis on perception thresholds is still subject of current research. [168]
The following items describe several, among others, effects found in literature which affect the measurement of perception thresholds:

- *Adaptation*: In [67], adaptation describes the process that humans habituate to motions. The author exemplifies the effect with a fast highway drive where the human body familiarizes with the high visual flow. In a subsequent slow urban drive, the perception system underestimates the speed as the perception system is still habituated to the high visual flow.

- *Absence of visualization:* In [82, 87], the perception threshold for angular accelerations is measured. The experiments were conducted in darkness.

- *Presence of visualization*: In [81], the perception of rotational motions is analysed in combination with visual stimuli (such as in a simulator). Results suggest that low scale rotations in combination with visual stimuli are initially interpreted as linear accelerations. Additional, distractive motion cues can enhance this interpretation by increasing the rotational perception threshold. In such environments, perception thresholds are mostly higher compared to experiments which are conducted in complete darkness [144].

- *Human expectation*: In [186], an experimental study shows that the rating of the perceived motion intensity differs whether the participants have seen the experimental setup before or not.

Table 2.3 and 2.4 lists some threshold values from literature. The list distinguishes between the kind of motion and whether the experiment was conducted with visualization or in darkness. The current thesis assumes a perception threshold for longitudinal and lateral accelerations of 0.17 $\mathrm{m/s^2}$ and a rotational velocity threshold of 3 $°/s$ according to [141]. The rotational acceleration threshold is not taken into account in this work.

Table 2.3: Comparison of translational acceleration thresholds in $[\mathrm{m/s^2}]$ for longitudinal, lateral and vertical direction.

Absence of Visualisation		Presence of Visualization			
Reference	$\ddot{x}$ / $\ddot{y}$ / $\ddot{z}$	**Reference**	$\ddot{x}$	$\ddot{y}$	$\ddot{z}$
Hosman and Van der Vaart (1978) [87]	0.04 - 0.085	Reid and Nahon (1985) [141]	0.17	0.17	0.28
Kingma (2005) [99]	0.065 - 0.085	Reymond and Kemeny (2000) [144]		0.05	
Zacharias (1978) [189]	0.02 - 0.3	Chapron and Colinot (2007) [33]		0.15	

Table 2.4: Comparison of rotational velocity thresholds $[°/s]$ and rotational acceleration thresholds $[°/s^2]$ for roll, pitch and yaw motions.

Absence of Visualization						
Reference	**Roll**		**Pitch**		**Yaw**	
	$\dot{\varphi}$	$\ddot{\varphi}$	$\dot{\theta}$	$\ddot{\theta}$	$\dot{\psi}$	$\ddot{\psi}$
Greig (1988) [80]	0.44		2.9			
Hosman and Van der Vaart (1978) [87]		0.022 -0.035		0.035-0.072		
Zacharias (1978) [189]	2.5 - 4		2.5 - 4		2.5 - 4	
Presence of Visualization						
Reference	**Roll**		**Pitch**		**Yaw**	
	$\dot{\varphi}$	$\ddot{\varphi}$	$\dot{\theta}$	$\ddot{\theta}$	$\dot{\psi}$	$\ddot{\psi}$
Gundry (1977) [84]	0.12				1.6 - 9	
Reymond and Kemeny (2000) [144]		0.3		0.3		0.3
Fortmüller and Meywerk (2005) [69]					2.5	
Nesti et al. (2012) [124]	6.3					
Reid and Nahon (1986) [141]	3.0		3.6		2.6	

2.7.3 Motion Sickness

Motion sickness can occur in virtual environments causing the following symptoms [71]:

- Mild symptoms: eyestrain, sleepiness, overstrain.
- Mid symptoms: sweating, headache, dizziness.
- Severe symptoms: unbalancing, nausea, vomitus.

The disease and the symptoms are comparable to seasickness. In literature [12, 119], a distinction is made between *motion sickness* which is provoked by uncomfortable motions in a boat or a vehicle and *simulator sickness* which is caused by missing or incorrect simulator motions. In both cases, stimuli of different perception channels are in conflict. In [105], it is shown that motion sickness depends on several factors, such as the mental rotation ability, gender, age, illness and simulator experience.

The following Section introduces several hypothesis to describe the causes of motion sickness.

The first approach is called *Sensory Conflict Theory* or *Cue Conflict Theory* and is the most accepted theory in literature. It can arise according to several scenarios: In the first scenario, the visual information (e.g. provided by a projector) about position, speed and acceleration does not correlate with the vestibular information (e.g. provided by the motion system). Thus, sensory conflicts occur between visual and vestibular percepts. The inconsistent, contrary stimuli can cause the above mentioned symptoms. Reasons for these perception discrepancies are all kinds of simulation errors, such as motion cueing errors, delays between driver command and motion execution, visualization errors, etc.. [105, 111, 119] In the second scenario, the sensory conflict is provoked by missing cues. An example is when the driver sits in a static simulator, and thus only observes motions by visual stimuli. Since this theory describes causes of the sickness, it cannot answer the question why sensory conflicts provoke simulator sickness. [74, 105]

A further hypotheses, known as *Poison* or *Intoxication Theory*, includes an approach to explain the occurrence of the symptom nausea. Herein, the human body interprets contrary stimuli in the perception system as poisoning. The presumed poison in the stomach is removed by vomiting to enhance survival. The theory gives no explanation why, for the same response, some people get sick and some not. [105, 118, 173]

In [145], an ecological approach, called *Postural Instability Theory*, is shown. It states that humans try to maintain within a postural stability in their surrounding environment. For example, humans perform a different walking pattern depending on whether they are walking on a solid ground or on an ice covered terrain. If humans perform the same walking pattern on ice as they do on a solid ground, they probably will fall down. In order to maintain a postural stability, the walking pattern changes depending on the terrain. On the contrary, postural instability occurs when humans loose their postural control. To apply this theory to simulators, motion sickness can occur in situations where participants do not maintain in posture stability. The longer participants feel an unstable situation, the stronger the likelihood and intensity of the symptoms. The authors mention four factors for causing a postural instability, such as: 1) weightlessness, 2) altered specificity, 3) changing relations between the gravitational force vector and the surroundings, and 4) low-frequency vibrations. It is assumed that motion sickness is provoked by altered specificities. Persons cannot develop postural control strategies for gaining postural stability as virtual environments create accelerations and rotations that are unrelated to the control of the body.

Another hypothesis [74] states that motion sickness is caused by increasing blood pressure and cardiac output. Simulator motions lead to an aberrant activation of neural pathways which function to maintain in a stable environment.

The *eye movement theory* is based on the vestibular nystagmus. The perception mechanism is a muscle reflex executed by the eyes and relates the visual system and the vestibular system. The reflex positions the eyes at a stationary point while moving the head. Hereby, the vestibular system retain control over the eyes. Each of the three vestibular canals interacts with one of the three muscles of an eye. By a movement of the head for example to the right side, the eyes slowly stir to the left side by focussing on one point. If the eyes reach the maximum position, they quickly move to the right side, followed again by a movement to the left side. The automatic stabilization of the eyes during head movements enables a stable view. Deviations in this process can cause headache, eye strain and lack of concentration. [23, 86, 105]

2.7.4 Perception Models

As mentioned in Section 2.7.1, the vestibular organ is divided into a semicircular and an otolith part. The semicircular canal mainly perceives rotational motions whereas the otolith organ primarily detects translational motions. Regarding this division, modelling of the vestibular functions is classified into two groups. Transfer functions relate the physical input, acting on the respective organ, with the perceived values. The following subsections outline some modelling approaches.

Semicircular Canal

In [164], the functions of the cupula are examined and two different ways for stimulation are described. First, investigations showed a deflection of the cupula induced by a rotational motion. Second, a deflection is measured, introduced by a stimulation with an electrically heated platinum. The deflection of the cupula increased shortly after terminating the stimulation and finally drops to the neutral position again. The analysis revealed the perception of rotational motion by the cupula and its transformation to an electrical signal.
In [175], the observations from [164] are used to describe the mechanism of the semicircular canal. The cupula-endolymph system is modelled as a heavily damped torsion pendulum. The differential equation with input angular velocity ω of the head and output angular deviation δ of the cupula in relation to the skull can be expressed in Laplace space as

$$\frac{\delta(s)}{\omega(s)} = \frac{K\,s}{(1+\tau_1 s)\,(1+\tau_2 s)} \tag{2.44}$$

with the time-constants τ_1, τ_2 and the static gain K. A suggestion for the time constants is made to $\tau_1 \approx 10\,\mathrm{s}$ and $\tau_2 \approx 0.1\,\mathrm{s}$ with a probable error of 20 - 25 %. In [189], it is noted that the time constants of the gain bandpass filter can be divided into a long time constant τ_L and a short time constant τ_S. The model assumes a linear relation between the cupula displacement δ and the perceived angular velocity $\hat{\omega}$. In addition, it is proposed to add a gain which is equal to the long time constant τ_L. The transfer function in equation (2.45) puts the applied angular velocity ω and the angular velocity $\hat{\omega}$, perceived by the hair cells, in relation

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{\tau_L\,s}{(1+\tau_L s)\,(1+\tau_S s)}. \tag{2.45}$$

Further investigations revealed that an exclusive model of the cupula is not sufficient as other mechanisms, such as the nystagmus and subjective responses, have to be taken into account as well. The authors in [188] showed deviations between the model in equation (2.44) and observations concerning the response to acceleration and velocity steps. Beside, there is a difference between the nystagmus estimated by eye-movement and the subjective perception of rotation.
To overcome these discrepancies a control-theory model is developed which includes an adaptation operator H_{adp}. The resulting model

$$H_{semi,1} = \frac{\hat{\omega}(s)}{\omega(s)} = \underbrace{\frac{\tau_A s}{1+\tau_A s}}_{H_{adp}} \cdot \frac{\tau_L\,s}{(1+\tau_L\,s)\,(1+\tau_S s)} \tag{2.46}$$

predicts the response to an angular velocity under exclusion of linear accelerations. Limitations of the model parameters include insufficient predictions for specific angular stimulations. Habituating to repeated stimulus patterns is also disregarded in the model. [188]
In [141], the authors used the model from equation (2.46) for their observations which distinguishes between pitch, roll and yaw rotations. The parameters for the different rotation axis are listed in Table 2.5.

Table 2.5: Parameters for model 2.46 in [141].

Parameters	Pitch	Roll	Yaw
τ_L - [s]	5.3	6.1	10.2
τ_S - [s]	0.1	0.1	0.1
τ_A - [s]	30	30	30

In [189], a lead sensitivity in vestibular perception of angular velocities is noted. To consider the lead behaviour, a second adaptation term $(1 + \tau_B\ s)$ is introduced with the time constant τ_B. Having regard to the previously introduced models, the authors in [168] proposed a model

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{\kappa \tau_L\, \tau_A\, s^2\, (1 + \tau_B\, s)}{(1 + \tau_L\, s)(1 + \tau_S\, s)(1 + \tau_A\, s)} \tag{2.47}$$

with the time constants $\tau_L = 5.73\,\mathrm{s}$, $\tau_S = 0.005\,\mathrm{s}$, $\tau_A = 80\,\mathrm{s}$, $\tau_B = 0.06\,\mathrm{s}$, $\kappa = 28.65$. The authors in [168] remark that numerical stability problems during an integration process can occur. A time step should be at least ten times smaller than the smallest time constant for numerical integration. For that reason, the authors in [168] propose a reduced order transfer function by neglecting the short time constant τ_S and the lead time constant τ_B to yield a reduced model

$$H_{semi,2} = \frac{\hat{\omega}(s)}{\omega(s)} = \frac{\tau_A\, \tau_L\, s^2}{(1 + \tau_L\, s)(1 + \tau_A\, s)} \tag{2.48}$$

with the time constants $\tau_L = 5.73\,\mathrm{s}$ and $\tau_A = 80\,\mathrm{s}$.
In [130], a similar model is proposed

$$H_{semi,3} = \frac{\hat{\omega}(s)}{\omega(s)} = \frac{\tau_L\, \tau_A\, s^2}{(1 + \tau_L\, s)\,(1 + \tau_A\, s)} \tag{2.49}$$

with the time constants $\tau_L = 18\,\mathrm{s}$ and $\tau_A = 30\,\mathrm{s}$.
In order to add the *perception threshold*, the work [141] proposes a block diagram as shown in Figure 2.9.

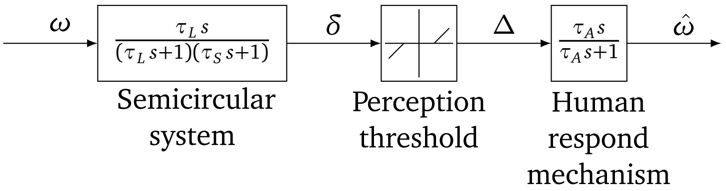


Figure 2.9: Block diagram of the semicircular perception system including the perception threshold [141].

The first block of Figure 2.9 refers to the cupula displacement model adapted from equation 2.46. The second block describes the perception threshold of humans (cf. Table 2.4).

The third block specifies the human response mechanism.
The first and the third block can be modelled in state space form. Adding the perception threshold leads to a nonlinear relation between the input and the perceived rotation rate.

Otolith Organ

Modelling of the otolith system can be made in a fashion similar to the semicircular system. Movements of the otolith membrane are assumed to be proportional to the hair cell deflections. An overdamped mass-spring-damper accelerometer functions as a model of the otolith system. [131]
In [113], a model is proposed

$$\frac{\hat{v}(s)}{v(s)} = \frac{\kappa \, \tau_L \, s}{(1+\tau_1 s)\,(1+\tau_2 s)} \tag{2.50}$$

with the time constants $\tau_1 = 10s$ and $\tau_2 = 0.66s$. The transfer function relates the translational input velocity v and the perceived velocity $\hat{v}$.
Using more modern measuring techniques, in [112], a lead term in response to a constant linear acceleration stimulus is added

$$H_{oto,1} = \frac{\hat{f}}{f} = \frac{\kappa\,(1+\tau_A s)}{(1+\tau_L s)\,(1+\tau_S s)} \tag{2.51}$$

with the time constants $\tau_A = 13.16s$, $\tau_L = 5.26s$, $\tau_S = 0.67s$, $\kappa = 0.4$, the perceived specific force $\hat{f}$ and the applied specific force f. The author in [113] mentions that the lead term could be referred to a more complex mechanical model, like a second mass-spring-damper combination, which models the movements of the macula with respect to the bony structure. Secondly, the lead term could be ascribed to the neurological processing of the otolith displacement signals.
In [131], a model for the response of the mechanical and afferent otolith system is developed which gives a relation between the input specific force f and the afferent firing rate (AFR - response to a neuron) as depicted in Figure 2.10. The second model $H(s)$, as shown in Figure 2.10, presumes that higher centres of the neuronal system process the AFR in order to obtain the perceived specific force $\hat{f}$.

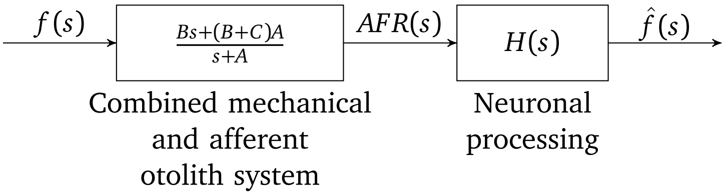


Figure 2.10: Block diagram of otolith model in [131].

In [168], updated model coefficients are proposed based on the model from equation (2.51). The specifications of the model in [168] are

$$H_{oto,2} = \frac{\hat{f}}{f} = \frac{\kappa\,(1+\tau_A s)}{(1+\tau_L s)\,(1+\tau_S s)} \tag{2.52}$$

with the time constants $\tau_A = 10s$, $\tau_L = 5s$, $\tau_S = 0.016s$ and $\kappa = 0.4$.
In [72], a simplified model is proposed

$$H_{oto,3} = \frac{\hat{f}}{f} = \frac{\kappa\,(1+\tau_A s)}{(1+\tau_L s)} \tag{2.53}$$

with time constants $\tau_A = 20s$, $\tau_L = 7.5s$ and $\kappa = 0.4$. Herein, the short time constant τ_S is neglected.

The otolith signal flow, including the perception threshold, is shown in Figure 2.11 and exhibits a similar structure to the semicircular system in Figure 2.11. Hereby, the first block describes the mechanical behaviour of the otolith organ. The second and third block model the perception threshold and the neural processing of the stimuli, respectively. Again, there is a nonlinear relation between input and output specific force.

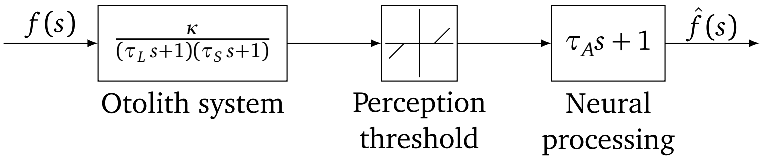


Figure 2.11: Block diagram of the otolith system including the otolith perception threshold [141].

2.7.5 Discussion

To compare the transmission behaviour of the various vestibular models, introduced in the previous Section, Figures 2.12 and 2.13 show the bode diagrams of $H_{semi,i}$ and $H_{oto,i}$ for the three models $i = 1\ldots3$, respectively. The magnitude plots of the semicircular and otolith models reveal similar characteristics for low to middle frequencies. Lower frequencies are less detected by the vestibular system. Secondly, the models' pass bands lie in comparable frequency ranges for the semicircular and otolith models, respectively. Though, the models' transmission behaviour differs for higher frequencies.
To observe the transmission behaviour of the vestibular organ for a real vehicle drive, the measured physical yaw angular velocity and the measured physical lateral accelerations of a roundabout crossing is shown in Figure 2.14 (blue plots). Using the perception models $H_{semi,i}$ and $H_{oto,i}$ for $i = 1\ldots3$, the perceived yaw angular velocity and the perceived lateral accelerations are calculated, respectively. The outputs of the transfer functions are included in the Figure 2.14. For short-time, constant values (yaw velocity $\approx 30°/s$; lateral acceleration $\approx$ 5m/s^2), the amplitude of the perceived values decreases over time which correlates with the high-pass characteristics of the vestibular models. The otolith model $H_{oto,1}$ shows a pronounced phase delay in the bode diagram and in the time domain.
Since the considered motion simulators throughout the work can only realize frequencies up to 30 Hz, there are no pronounced differences in the transmission behaviour between the models. Due to its low model order, the semicircular model $H_{semi,2}$ and otolith model $H_{oto,2}$ of Telban et al. [168] is taken for further developments which is also the applied perception model in state-of-the-art works, such as [9, 26, 28, 62].

2.8 Chapter Summary

This Chapter introduces the kinematics of the motion simulator which is considered throughout the thesis. Based on the simulator's structure, the specific forces and the resulting angular velocities are derived. The specific forces are the resulting accelerations working on the driver in the simulator. As the motion commands, estimated by an MCA, have to cope with the simulator limitations, the workspace and actuator space of a parallel motion system are described.

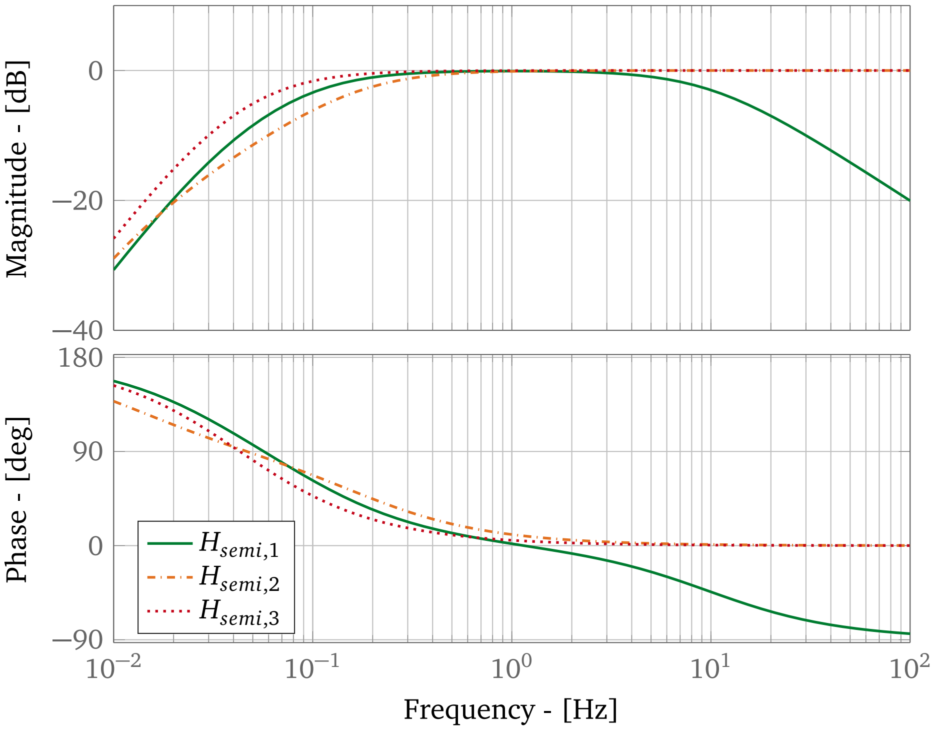


Figure 2.12: Bode plots of the semicircular models $H_{semi,1}$, $H_{semi,2}$ and $H_{semi,3}$.

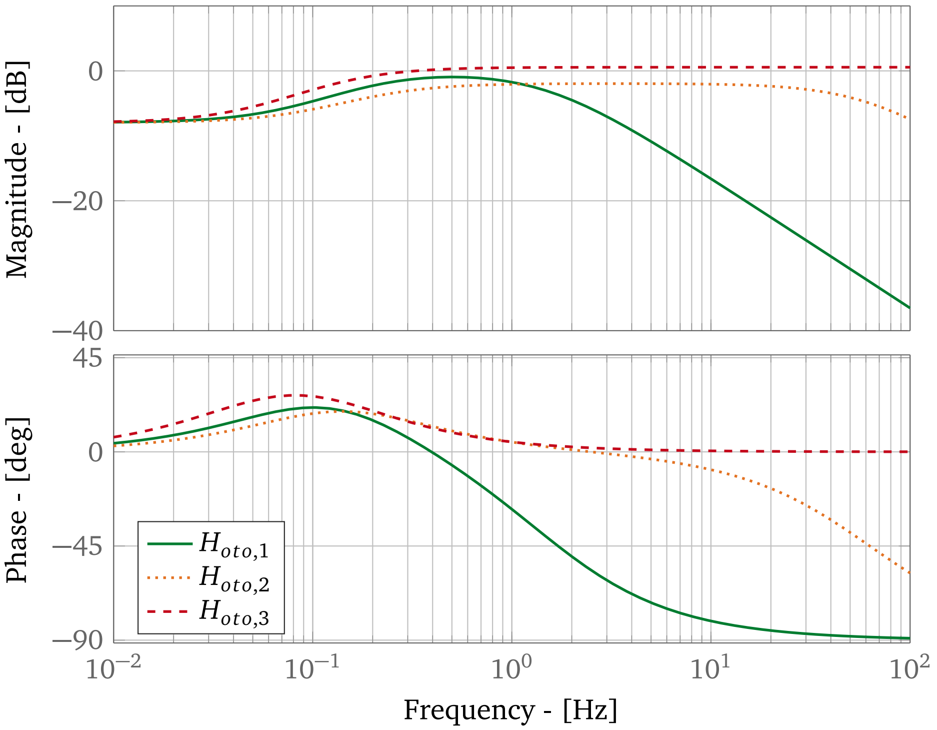


Figure 2.13: Bode plots of the otolith models $H_{oto,1}$, $H_{oto,2}$ and $H_{oto,3}$.

The inverse and direct kinematics transform motions between workspace and actuator space. The inverse kinematics can be solved analytically for a parallel robot, whereas the direct kinematics typically uses numerical schemes to solve a nonlinear system of equations.

In the second part of the chapter, the human motion perception system is described in order to

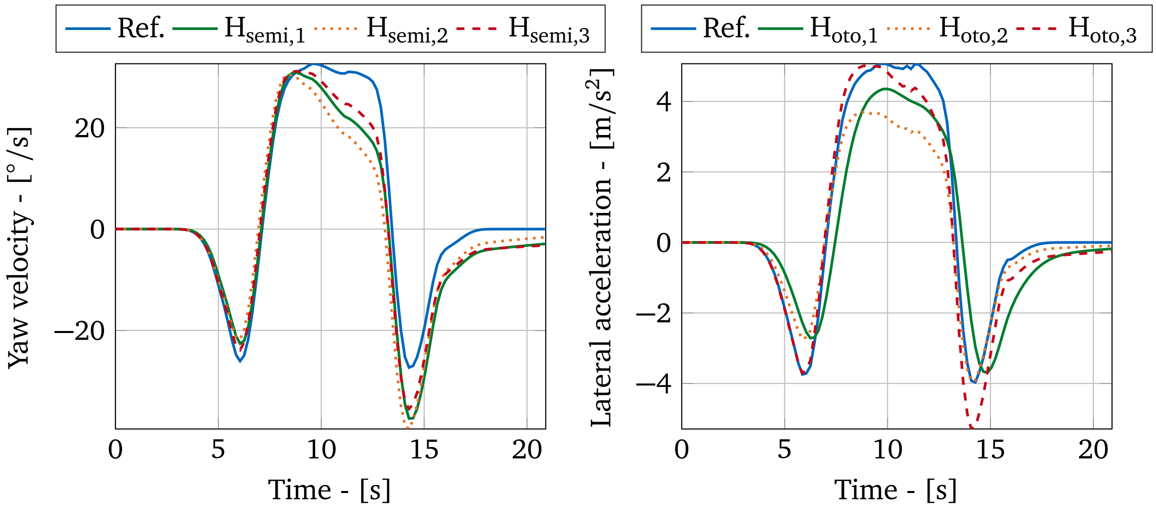


Figure 2.14: Measured yaw angular velocity (left) and lateral acceleration (right) of a roundabout crossing and outputs of the semicircular models $H_{semi,1}$, $H_{semi,2}$ and $H_{semi,3}$ and the otolith models $H_{oto,1}$, $H_{oto,2}$ and $H_{oto,3}$, respectively.

analyse the motions induced to the simulator driver. With regard to MCAs, only the vestibular system is considered, which consists of the otolith organ and the semicircular canals. The otolith organ mainly perceives translational motions and the semicircular canals primary perceive rotational motions. Vestibular models estimate the perceived motions in relation to the physical motions which are acting on the human. These models can be included in an MCA in order to consider the actual perceivable motions which are to be reproduced by a motion simulator.

Chapter 3

Filter-Based MCAs

The content of the following Chapter have been published in [47, 50, 52, 53].

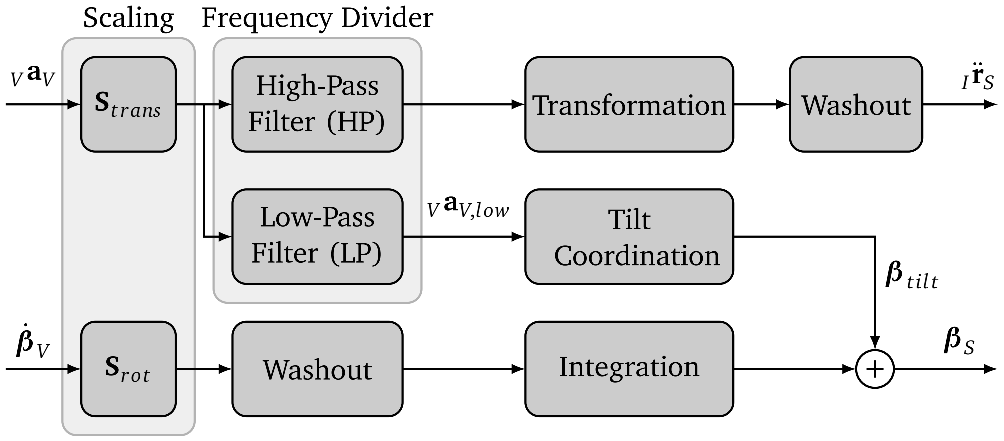


Figure 3.1: Structure of the filter-based, Classical Washout MCA.

As described in Figure 2.8, there are generally several ways to replicate horizontal motions with a simulator which exhibits more than one DoF. Translational motions can be replicated by translational accelerations and tilt motions. Filter-based MCAs split the motions by a frequency divider between high-frequency motions and low-frequency motions.
One of the earliest filter-based MCAs was most likely developed by [151] for a flight simulator and is known as the Classical Washout MCA. Even though a lot has changed since then, the basic operating principle of filter-based MCAs is still the same. Figure 3.1 shows a principal structure of a filter-based MCA. The algorithm mainly consists of linear transfer functions, such as high-pass and low-pass filters. Due to its simple, modular structure, the algorithm is still in use and finds application in flight and driving simulations. A major advantage is the relatively simple tuning process of the algorithm's parameters. Typical parameters and tuning sets are found in [78, 122, 142, 147]. Though, determining the cut-off frequency of the frequency splitters is a cumbersome task as it depends on the driving dynamics data and on the specificities of the simulator. The algorithm can be employed in a closed-loop and open-loop simulation. Actuator limitations are typically not directly included in a filter-based MCA, but can be integrated by an additional control algorithm. Inputs consist of the translational accelerations ${}_V\boldsymbol{a}_V$ and the time-derivatives of the vehicle's Cardan angles $\dot{\boldsymbol{\beta}}_V$. Outputs are the simulator's translational accelerations ${}_T\ddot{\boldsymbol{r}}_S$ and the rotational DoFs $\boldsymbol{\beta}_S$ of the motion system. The following Sections describe the principal functions of the blocks shown

in Figure 3.1. Along with this, also extensions of filter-based approaches are presented, such as optimal filters, a workspace supervision method and pre-positioning techniques.

3.1 Scaling

The scaling blocks $\boldsymbol{S}_{trans}$ and $\boldsymbol{S}_{rot}$ transform the driving dynamics inputs ${}_V\boldsymbol{a}_V$ and $\dot{\boldsymbol{\beta}}_V$ to reproducible ranges for the motion system. Thus, scaling values depend on the motion system's workspace capabilities and the maximum values of ${}_V\boldsymbol{a}_V$ and $\dot{\boldsymbol{\beta}}_V$. In [11, 67, 101, 136], studies show that scaling values in the range of $0.4 - 0.75$ are acceptable, whereas lower values lead to a reduced driving performance.

3.2 Frequency Divider

Long-lasting, low-frequency accelerations yield large platform deflections which can cause workspace overruns. Whereas translations in the high-frequency range are reproduced by translational simulator movements, accelerations in the low frequency range are emulated by tilt motions (cf. Figure 2.8). Due to the gravitational force, pitching and rolling simulator motions below the rotational perception threshold are perceived as translational accelerations [121]. This effect is used when tilting the motion platform in order to represent persistent longitudinal and lateral accelerations. Typically, these filter-based algorithms are designed for non-redundant motion systems, like hexapods [150]. High-pass (HP) and low-pass (LP) filters divide the translational accelerations into two different frequency ranges which are defined by

$$HP(s) = \frac{\kappa_{HP}\, s}{s + \omega_{HP}}, \qquad LP(s) = \frac{\kappa_{LP}}{\frac{s}{\omega_{LP}} + 1} \tag{3.1}$$

with the gains $\kappa_{HP}\,/\,\kappa_{LP}$. The factors ω_{HP}/ω_{LP} indicate the break frequency which separates the frequency ranges into a pass-band and a stop band. For the parametrization, complementary break frequencies

$$\omega_{HP} = \omega_{LP} \tag{3.2}$$

are proposed in [150]. Thus, no frequency ranges are lost since the initial signal is composed of the summation of the low-pass and high-pass filtered signal (for $\kappa_{HP} = \kappa_{LP}$). In [142], a transition between the break frequencies with the relation

$$2 \cdot \omega_{HP} = \omega_{LP} \tag{3.3}$$

is used. In [67, 68], some default filter parametrizations can be found for a three DoFs, six DoFs and eight DoFs motion system. Based on the analysis in [67, 150], complementary filters are used throughout this work. As the parametrization also depends on the inputs ${}_V\boldsymbol{a}_V$ and $\dot{\boldsymbol{\beta}}_V$, the filter parameters are chosen to maximally exploit the motion system's workspace capabilities for a pre-defined reference manoeuvre.

3.3 Tilt Coordination

Low frequency signals are reproduced by tilt coordination replicating long-lasting accelerations. As the otolith system cannot differentiate between translational and gravitational

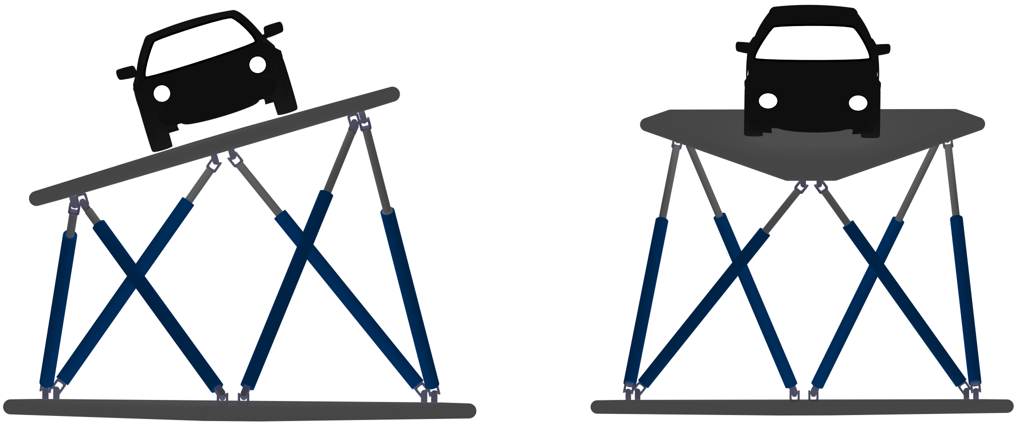


Figure 3.2: Left plot: Rotation about x-axis to replicate long-lasting, lateral accelerations. Right plot: Rotation about y-axis to replicate long-lasting, longitudinal accelerations.

accelerations, the platform is inclined by an angle to take advantage of the gravitational force (limited to approx. $20-30°$ [122, 141]), as depicted in Figure 2.8. Herein, the rotational platform velocity is preferably below the human perception threshold (often $3\,°/s$ [122, 141]). Thus, the perception system is not able to distinguish between the body inclination around the longitudinal or lateral axis and a translational acceleration. Figure 3.2 shows the tilting positions of a hexapod platform for replicating a lateral acceleration (left plot) and a longitudinal acceleration (right plot).
An approach to estimate the tilt-angle of a filter-based MCA is taken from [150]. The approach neglects coupling effects between the angles which are considered by the transformation between the inertial FoR I and the platform FoR N. For a general transformation ${}_N\boldsymbol{R}_I$, the long lasting specific force ${}_N\boldsymbol{f}_{LP}$ resulting from the gravitational force ${}_I\boldsymbol{g}$ is obtained by

$$ {}_N\boldsymbol{f}_{LP} = {}_N\boldsymbol{R}_I \; {}_I\boldsymbol{g}. \tag{3.4}$$

Applying these considerations to the tripod / hexapod system yields

$$ {}_H\boldsymbol{f}_{LP} = {}_H\boldsymbol{R}_I \left(\varphi_{TH,tilt}, \theta_{TH,tilt}, \psi_{IH,tilt}\right) \; {}_I\boldsymbol{g} \tag{3.5}$$

with

$$ {}_Hf_{LP,x} = -g \cdot \sin\big(\theta_{TH,tilt}\big) \tag{3.6}$$
$$ {}_Hf_{LP,y} = g \cdot \cos\big(\theta_{TH,tilt}\big) \cdot \sin\big(\varphi_{TH,tilt}\big) \tag{3.7}$$
$$ {}_Hf_{LP,z} = g \cdot \cos\big(\theta_{TH,tilt}\big) \cdot \cos\big(\varphi_{TH,tilt}\big). \tag{3.8}$$

The tilting angles $\theta_{TH,tilt}$ and $\varphi_{TH,tilt}$ are obtained by

$$ \varphi_{TH,tilt} = \arctan\big({}_Hf_{LP,y} / {}_Hf_{LP,z}\big) \approx \arctan\left(\frac{{}_Va_{V,low,y}}{g}\right) \tag{3.9}$$
$$ \theta_{TH,tilt} = -\arctan\big({}_Hf_{LP,x} / {}_Hf_{LP,z} \cdot \cos\big(\varphi_{TH,tilt}\big)\big) \approx -\arctan\left(\frac{{}_Va_{V,low,x}}{g}\right). \tag{3.10}$$

using a small angle approximation $\cos\big(\varphi_{TH,tilt}\big) \approx 1$ and the approximation ${}_Hf_{LP,z} \approx g$. The terms ${}_Va_{V,low,y}$ and ${}_Va_{V,low,x}$ indicate the scaled, low-pass filtered input signal ${}_V\boldsymbol{a}_V$ in ${}_Vy-$ and ${}_Vx-$direction, respectively (cf. Figure 3.1).

3.4 Classical Washout

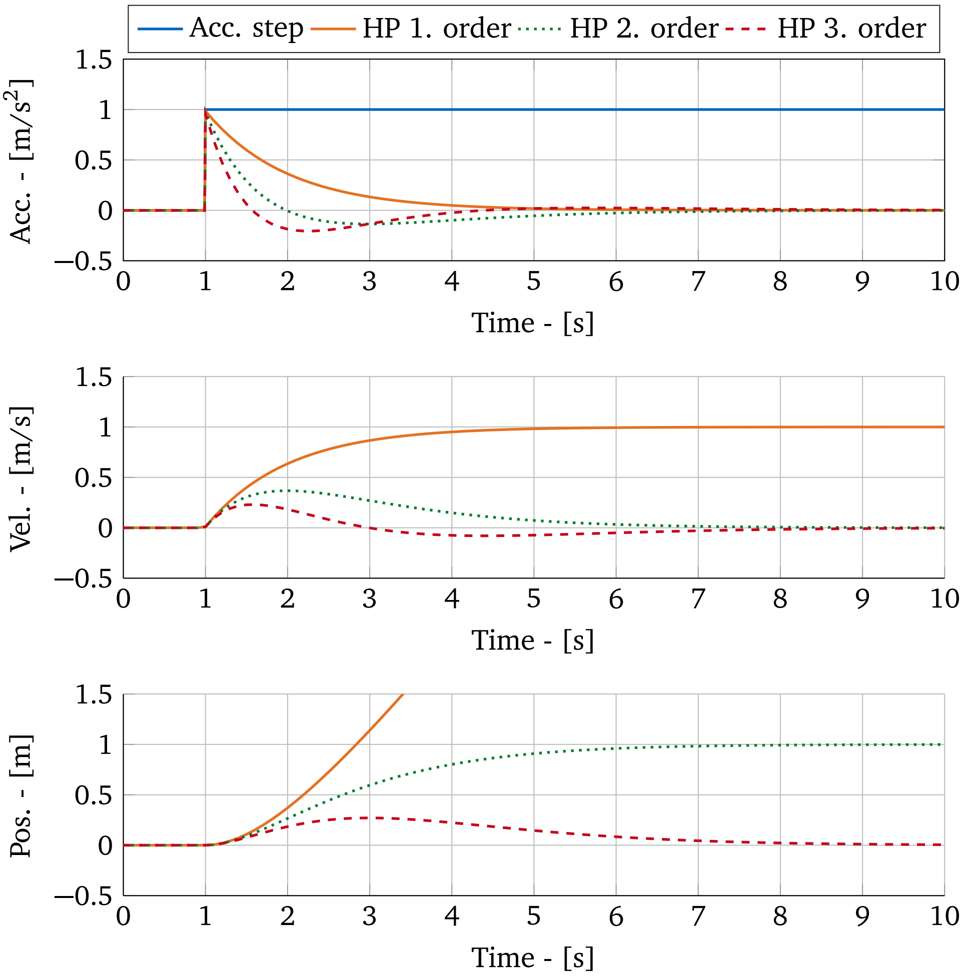


Figure 3.3: Upper plot: Step response applied to high-pass filters with order 1-3 on the acceleration (acc.) level. Middle and lower plots: Velocities (vel.) and positions (pos.) for the respective filter types.

First order high-pass filters of the frequency divider block (cf. Figure 3.1) are not sufficient to keep simulator motions within the workspace envelope as an additional acceleration is necessary to return the motion platform to the neutral position. In order to enable such a washout function, further higher order filters are integrated as washout blocks. Outputs of these washout blocks often result in perceivable contrary motions to the desired simulator motions in order to prevent workspace exceedance. In other words, these washout motions often have a negative impact on the driver perception as they yield deviations to the desired driver commands. The following paragraph exemplifies this functionality.
Figure 3.3 shows the step response on position, velocity and acceleration level for a uniform jump of the acceleration signal at the time step 1 s. First to third order filters are applied to the acceleration signal in the form

$$HP^{1^{st}order}(s) = \frac{\kappa_{HP} \cdot s}{s + \omega_{HP}} \tag{3.11}$$

$$HP^{2^{nd}order}(s) = \frac{\kappa_{HP} \cdot s^2}{s^2 + 2\,D\,\omega_{HP}\,s + \omega_{HP}^2} \tag{3.12}$$

$$HP^{3^{rd}order}(s) = \frac{\kappa_{HP} \cdot s^3}{s^3 + 3\,\omega_{HP}\,s^2 + 3\,\omega_{HP}^2\,s + \omega_{HP}^3} \tag{3.13}$$

with the damping coefficient $D = 1$, the scaling factor $\kappa_{HP} = 1$ and the break frequency ω_{HP} =rad/s.
The upper plot of Figure 3.3 shows the desired acceleration step (blue colour) to replicate the desired driver command. Since a constant, long-lasting, translational acceleration demand is usually not replicable on a motion system with workspace restrictions, the desired acceleration is modified through high-pass filters. On acceleration level, a first order filter is sufficient for the convergence of the unit jump to zero. The plot also reveals the deviations between the desired acceleration signal and the actual accelerations provided by the motion system. On velocity level (middle plot), the first order high-pass filtered signal goes to an asymptote so that the respective position signal diverges, demonstrating that a filter of at least second order is necessary for convergence. The lowest plot reveals that at least a third order filter is necessary to converge the position signal to zero, meaning that the motion platform returns to its neutral position. The observations can be explained by applying the limit theorem [129]. Herein, the output signal $y(t)$ is a function of the high-pass filter hp and the input $u(t)$ in time-domain

$$y(t) = \int_0^t hp(\tau)\,u(t-\tau)\mathrm{d}\tau \tag{3.14}$$

Transforming equation (3.14) to Laplace-space, the acceleration step $U(s) = 1/s$ is integrated twice to receive a response on position level. Applying the limit theorem on a third order high-pass filter $HP^{3^{rd}order}$ yields

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} s \cdot Y(s) = \lim_{s\to 0} s \cdot HP^{3^{rd}order}(s) \cdot \frac{1}{s^2} \cdot U(s) = \lim_{s\to 0} HP^{3^{rd}order}(s) \cdot \frac{1}{s^2} \longrightarrow 0 \tag{3.15}$$

Transferring this knowledge to a filter-based MCA from Figure 3.1, the frequency divider functions as a first order high-pass filter. In order to enable a washout function on position level, an additional second order filter (washout) is necessary to yield a third order transmission behaviour between input accelerations ${}_V\boldsymbol{a}_V$ and simulator positions ${}_I\boldsymbol{r}_S$. For the rotational inputs $\dot{\boldsymbol{\beta}}_V$, a second order washout filter is sufficient to return the platform angles $\boldsymbol{\beta}_S$ to zero.

3.5 Optimal Washout Filter

In order to differentiate optimal filters from optimization-based MCAs, optimal washout filters are shortly introduced in this Section. The structure of these filter-based MCAs is broadly reported in several works, such as [34, 89, 123, 161, 168]. In contrast to Classical Washout algorithms, the structure of optimal filters are based on a linear-quadratic-Gaussian control.
Figure 3.4 shows the structure of the algorithm which rests upon an optimization problem. The translational accelerations ${}_V\boldsymbol{a}_V$ and the Cardan angles' time-derivatives $\dot{\boldsymbol{\beta}}_V$ of the vehicle are the inputs ${}_V\boldsymbol{u}_V = [{}_V\boldsymbol{a}_V, \dot{\boldsymbol{\beta}}_V]$. The upper part of the scheme S_V estimates the driver's

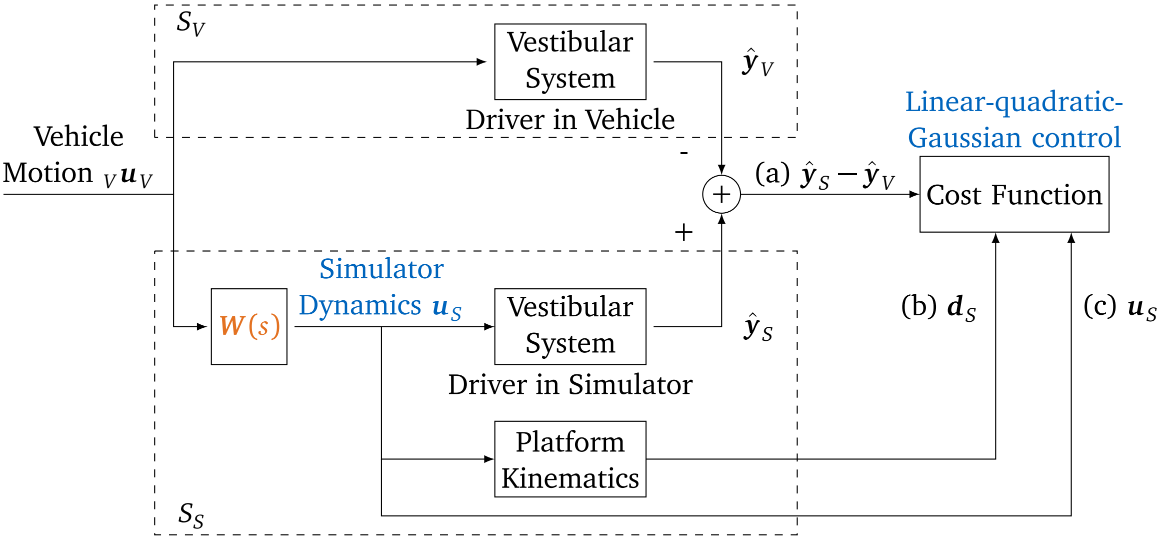


Figure 3.4: Structure of the offline optimization to estimate the optimal washout filters.

perception in a real vehicle where a human perceptional model is used (cf. Section 2.7.4). On the lower part of the scheme, a second system S_S provides an estimate of the driver's perception in the simulator $\hat{y}$ and the simulator motions $\boldsymbol{d}_S$, given by the simulator inputs $\boldsymbol{u}_S$. The simulator inputs $\boldsymbol{u}_S$ are unknown and the optimal control problem can be formulated as solving the simulator's inputs which minimizes a cost function without violating the simulator's dynamics. The quadratic cost function consists of three terms: (a) The difference between the motion a driver would perceive in a real vehicle $\hat{\boldsymbol{y}}_V$ and the motion actually felt by the driver in a simulator $\boldsymbol{y}_S$, (b) the platform kinematics $\boldsymbol{d}_S$ and (c) the simulator inputs $\boldsymbol{u}_S$. Symmetrical, positive-definite matrices provide a weighting between the optimization parameters influencing the used workspace and the motion cueing fidelity.
The optimal filter MCA [169] solves the continuous-time optimal control problem on basis of a linear-quadratic-Gaussian control (Figure 3.4) by applying the Riccati equation [15]. The method is executed in an offline calculation and generates a multidimensional filter $\boldsymbol{W}(s)$ transforming the input ${}_V\boldsymbol{u}_V$ into a feasible control vector $\boldsymbol{u}_S$ for the simulator providing tilt coordination and washout functions in the form

$$\boldsymbol{u}_S = \boldsymbol{W}(s)\,{}_V\boldsymbol{u}_V. \tag{3.16}$$

Due to the linear-quadratic-Gaussian control of the optimal filter MCA, future trajectories are not considered in the optimization problem for which reason pre-positioning of the platform is not performed. As the optimization problem does not include hard workspace constraints, violations of the simulator's technical restrictions have to be checked for each driving manoeuvre with a verification program.
Figure 3.5 exemplarily shows the closed-loop MCA for the longitudinal acceleration and the pitch rotation. The translational acceleration ${}_I a_{V,x}$ is represented in the FoR I. Similar to the Classical Washout MCA, the input signals are scaled by S_{trans} and S_{rot}. The washout functions and tilt coordination are provided by the matrix $\boldsymbol{W}(s)$ in the form

$$\boldsymbol{u}_S = \begin{bmatrix} \dot{\theta}_S \\ {}_I\ddot{r}_{S,x} \end{bmatrix} = \begin{bmatrix} W_{11}(s) & W_{12}(s) \\ W_{21}(s) & W_{22}(s) \end{bmatrix} \begin{bmatrix} S_{rot} \cdot \dot{\theta}_V \\ S_{trans} \cdot_I a_{V,x} \end{bmatrix} \tag{3.17}$$

The transfer function W_{21} is neglected in the algorithm. The coefficients of the transfer functions in $\boldsymbol{W}(s)$ are constant during a closed-loop driving session. The simulator's lateral,

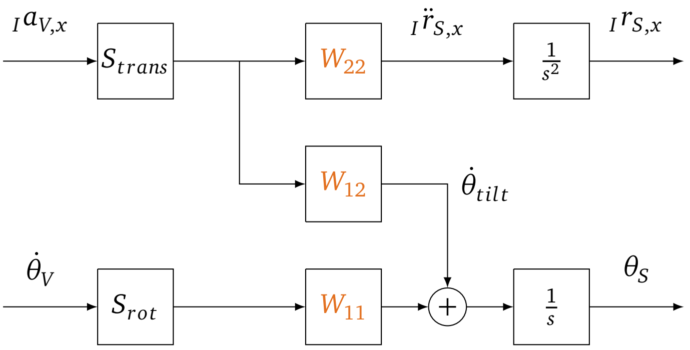


Figure 3.5: Blockdiagram of the optimal filter MCA.

vertical and rotational motions can be estimated in a similar way, which is shown in [168] in more detail.

3.6 Prepositioning

In order to increase the workspace for a specific manoeuvre, a pre-positioning technique can be used (e.g. [150]). Herein, the platform is moved in advance to future manoeuvre(s) (usually to the direction opposite to the direction of future manoeuvre(s)). For example, the motion platform is slowly moved to the right position when the virtual vehicle is on the right lane of a multilane road, such that, in case of a lane change to the left, there is more workspace available. The pre-positioning motion preferably is below the human perception threshold. Previous pre-positioning techniques are mainly used for longitudinal and lateral accelerations in combination with a filter-based MCA. The following Section presents two techniques from literature.

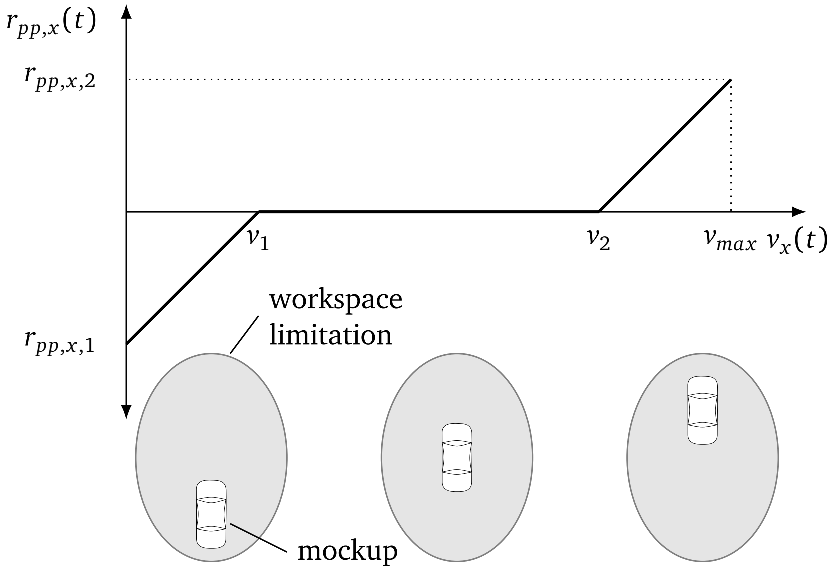


Figure 3.6: Predictive pre-positioning based on the vehicle speed (adopted from [150]).

In [150, 185], possibilities to predict upcoming driving manoeuvres for certain situations are stated. The following exemplary use-cases are described: for longitudinal dynamics, low-speed and high-speed scenarios are distinguished. Figure 3.6 depicts the longitudinal platform pre-positioning depending on the vehicle speed. In case the virtual vehicle is standing still, an acceleration manoeuvre is expected and the platform moves to the rear position (platform position $r_{pp,x,1}$ in Figure 3.6). By contrast, when the virtual vehicle runs at high speed, a deceleration manoeuvre is highly possible and the platform moves to the front position (platform position $r_{pp,x,2}$ in Figure 3.6). Linear functions enable a smooth transition between the front and rear pre-positioning covering the possible vehicle speed range.

In case the virtual vehicle is on the right lane of a multilane road, a lane change to the right is highly probable. Hereby, extreme cases, such as leaving the road, are neglected. In preparation for the predicted manoeuvre, the motion platform moves to the right position in order to increase the workspace capabilities for a future motion to the left.

These considerations are summed up as case-differentiations depending on the vehicle's speed. For the longitudinal pre-positioning (assumption: $v_x(t) \geq 0$), they are defined by

$$r_{pp,x} = \begin{cases} \frac{r_{pp,x,1}}{v_1} \cdot v_x(t) - r_{pp,x,1} & , 0 \leq v_x(t) \leq v_1 \\ 0 & , v_1 \leq v_x(t) \leq v_2 \\ (v_x(t) - v2) \cdot \frac{r_{pp,x,2}}{v_{max} - v_2} & , v_2 \leq v_x(t) \leq v_{max} \\ r_{pp,x,2} & , v_x(t) \geq v_{max} \end{cases} \tag{3.18}$$

and can be applied to the lateral dynamics in a similar manner.

In [85], a related approach to [150] for the longitudinal pre-positioning is shown. The authors considered variable maximum positive accelerations $a_{pos,max}$ and maximum negative accelerations $a_{neg,max}$ depending on the current vehicle speed and driving dynamics model as shown in Figure 3.7. The acceleration capabilities decrease at higher vehicle speeds and the acceleration capabilities depend on the vehicle model. For the estimation of the pre-positioning signal, an empirical function approximates the maximum accelerations $a_{pos,max}$ depending on the current vehicle speed and the vehicle model. Since the maximum vehicles deceleration $a_{neg,max}$ does not depend on the current vehicle speed, the vehicle model, $a_{neg,max}$ is considered constant. The averaged acceleration is estimated by

$$\bar{a}(v_x) = \frac{a_{pos,max}(v_x) + a_{neg,max}}{2}. \tag{3.19}$$

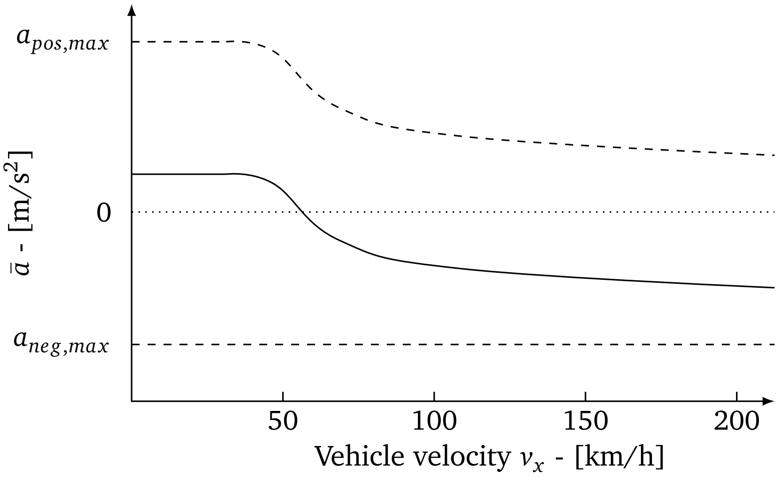


Figure 3.7: Averaged acceleration $\bar{a}$ based on the maximum positive accelerations $a_{pos,max}$ and maximum negative accelerations $a_{neg,max}$ (based on [85]).

The pre-positioning signal $r_{pp,x}$ is calculated by

$$r_{pp,x}(v_x) = \bar{a}(v_x) \cdot \frac{r_{pp,x,max}}{\max(|\bar{a}|)} \tag{3.20}$$

with the maximum averaged acceleration of $\bar{a}$ and the maximum pre-positioning $r_{pp,x,max}$. The pre-positioning can be integrated to an MCA by setting the washout's neutral position to the pre-positioning signal.

3.7 Filter-Based MCA for Redundant Motion Systems (MCA$_{\text{FILT}}$)

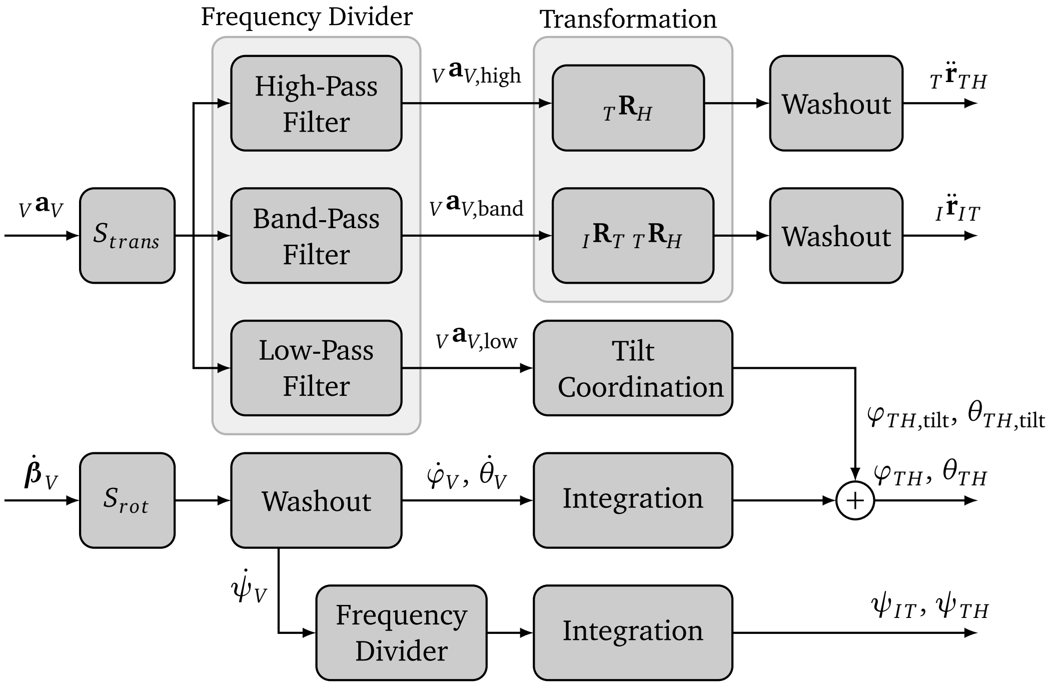

Frequency Divider
Transformation
High-Pass Filter
Band-Pass Filter
Low-Pass Filter
$_V\mathbf{a}_V$
S_{trans}
$_V\mathbf{a}_{V,\text{high}}$
$_V\mathbf{a}_{V,\text{band}}$
$_V\mathbf{a}_{V,\text{low}}$
$_T\mathbf{R}_H$
$_I\mathbf{R}_T\ _T\mathbf{R}_H$
Tilt Coordination
Washout
Washout
$_T\ddot{\mathbf{r}}_{TH}$
$_I\ddot{\mathbf{r}}_{IT}$
$\varphi_{TH,\text{tilt}}, \theta_{TH,\text{tilt}}$
$\dot{\boldsymbol{\beta}}_V$
S_{rot}
Washout
$\dot{\varphi}_V, \dot{\theta}_V$
Integration
+
$\varphi_{TH}, \theta_{TH}$
$\dot{\psi}_V$
Frequency Divider
Integration
ψ_{IT}, ψ_{TH}

Figure 3.8: Structure of the filter-based MCA solving the redundant DoFs with a frequency divider.

In [68, 94, 135, 149], approaches to run a filter-based algorithm on a redundant motion system is introduced by using a frequency splitter. Figure 3.8 shows the structure of a filter-based MCA (called MCA$_{\text{FILT}}$ throughout the work) for the redundant motion system introduced in Figure 1.3. The algorithm mainly differs from the general version in Figure 3.1 by the extended frequency divider. The block solves the redundancy in the DoFs and splits scaled translational accelerations into three signal domains. Low frequency signals $_V\boldsymbol{a}_{V,\text{low}}$ are performed by tilt coordination replicating long-lasting accelerations. Mid-frequency signals $_V\boldsymbol{a}_{V,\text{band}}$ are realized by the tripod because of its higher workspace and its lower dynamic capabilities compared to the hexapod. Thus, high-frequency signals $_V\boldsymbol{a}_{V,\text{high}}$ are performed by the hexapod. The redundant rotational yaw motion $\dot{\psi}_V$ is also split by a low/high-pass filter combination (this is simply indicated by a frequency divider block in the graph). Complementary filters are used which exhibit an equal break frequency for high/low-pass filter combinations as described in Section 3.2. Thus, the original signal is obtained when combining the filtered signals. Washout blocks are inserted, consisting in additional high-pass filters which ensure motions to the neutral platform position in order to take into account the workspace capabilities.

3.7.1 Workspace Restrictions

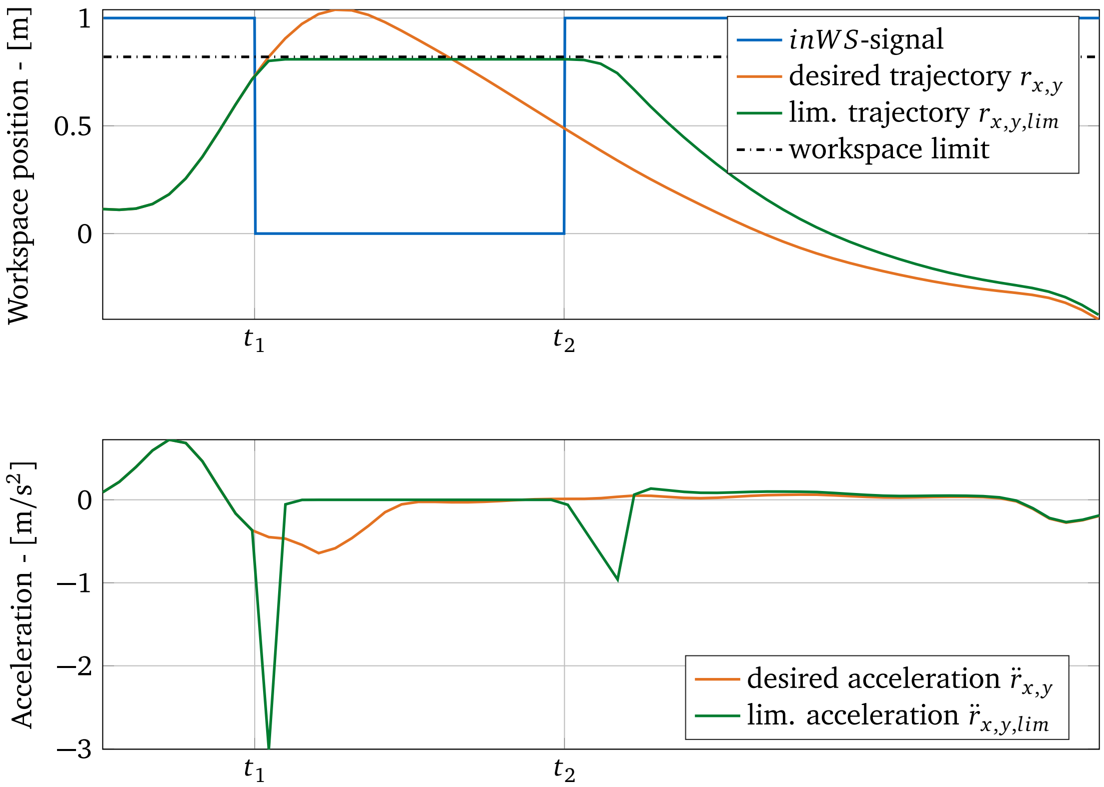


Figure 3.9: Example of a limitation procedure at the time instant t_1. After fulfilling the workspace restrictions and after expiration of the hold time t_{hold} (at the time instant t_2), the limited trajectory follows the desired trajectory (Upper plot: on position level. Lower plot: on acceleration level).

The MCA scheme in Figure 3.8 does not include workspace supervision, i.e. a monitoring function which checks whether the estimated platform motions are realizable or not. It is not guaranteed that the commanded hexapod and tripod motions stay within actuator limitations.

A workspace supervision module was developed for this purpose. It is the result of collaboration with Maximilian Spannagl [163]. As the platform motions are sent in workspace coordinates $\boldsymbol{w}_S$, the inverse kinematics, using equation (2.22), is applied to calculate the current actuator strokes q. In case an actuator limit exceeds, the workspace signal $inWS$ changes from 1 to 0.

The limitation function is performed by a cascade control [90], defining the desired workspace coordinates $\boldsymbol{w}_d$, $\dot{\boldsymbol{w}}_d$, $\ddot{\boldsymbol{w}}_d$ as inputs. If an actuator overshoot is detected, a stop trajectory halts the respective system by setting $\dot{\boldsymbol{w}}_d$, $\ddot{\boldsymbol{w}}_d$ to zero. The system is held at least for a time duration t_{hold} before the reactivation begins in order to avoid on/off effects in the $inWS$-signal. On reactivation of the motion cueing by the workspace supervision module, the desired platform motions estimated by the MCA are set as inputs again, leading to a fade into the desired trajectory.

Figure 3.9 exemplarily shows a limitation event. The supervision module detects a workspace overshoot at the time instant t_1 (signal $inWS$ changes from “1”- in workspace - to “0” - workspace exceedance). The desired trajectory $r_{x,y}$ is no longer feasible and the accelerations $\ddot{r}_{x,y}$ and the velocities $\dot{r}_{x,y}$ of the motion system are modified to the limitation trajectories $\ddot{r}_{x,y,lim}/\dot{r}_{x,y,lim}$. At the time instant t_2, the workspace limitations are fulfilled (and expiration of t_{hold}) and the limitation trajectories $\ddot{r}_{x,y,lim}/\dot{r}_{x,y,lim}$ tracks the desired trajectories $\dot{r}_{x,y}/\ddot{r}_{x,y}$ again.

3.7.2 Evaluation

Figure 3.10 shows a double lane change manoeuvre including a longitudinal acceleration phase and deceleration phase (at the beginning and at the end of the manoeuvre, respectively). In the upper two plots of Figure 3.10 (left: longitudinal, right: lateral), the frequency divider separates the scaled reference signal ${}_V\boldsymbol{a}_V$ (scaling factor: $k_x = 0.3$, $k_y = 0.6$) into the low frequency signal ${}_V\boldsymbol{a}_{V,\text{low}}$ for the tilt coordination, the middle-frequency signal for the tripod ${}_V\boldsymbol{a}_{V,\text{band}}$ and the high-frequency signal for the hexapod ${}_V\boldsymbol{a}_{V,\text{high}}$ (see Figure 3.8). These signals are not filtered by the washout function so that a summation of these three signals results in the scaled reference ${}_V\boldsymbol{a}_V$. Aside from this, the plots demonstrate the resulting acceleration ${}_H\boldsymbol{a}_S$ which is acting at the CRP of the simulator (after washout). The left plot focuses on the longitudinal and the right plot on the lateral direction. Deviations between the signals ${}_V\boldsymbol{a}_V$ and ${}_H\boldsymbol{a}_S$ are caused by the washout filter.

The lower two plots in Figure 3.10 (left: longitudinal, right: lateral) demonstrate the resulting motion of the tripod ${}_I\boldsymbol{r}_{IT}$ and hexapod ${}_T\boldsymbol{r}_{TH}$ as well as the tilting angles φ_{tilt} and θ_{tilt}. The same output variable names as in Figure 3.8 are used. The positions return to the neutral zero position as the signals pass through a washout filter.

3.8 Chapter Summary

The Chapter describes the main structure of filter-based MCAs for non-redundant and redundant motion systems. Linear transfer functions make a simple usage of these algorithms possible. Since the algorithms do not include the workspace restrictions of the motion system, the parametrization (scaling, washout) of the algorithm is mainly based on the most challenging manoeuvre. Considering the complete course, this tuning strategy yields a motion cueing that depends on the cueing of these challenging sections. In other words, in case the challenging manoeuvres make up only a small proportion of the complete course, the workspace exploitation of the low to medium challenging manoeuvres is probably quite low. In order to not exceed the workspace limitations, this Section proposes a workspace supervision module which modifies infeasible simulator inputs to feasible ones. However, a workspace intervention occurs with a high acceleration peak which strongly affects the motion cueing quality.

In order to stick to the workspace limitations of the motion system, washout functions move the platform to its neutral position. The estimated accelerations are often acting in opposite direction to the desired accelerations which leads to false cues. Prepositioning techniques enable possibilities to integrate additional driving dynamics information to the motion cueing in order to enhance the workspace exploitation. The pre-positioning procedure preferably has to be below the perception threshold, as it otherwise would lead to false cues.

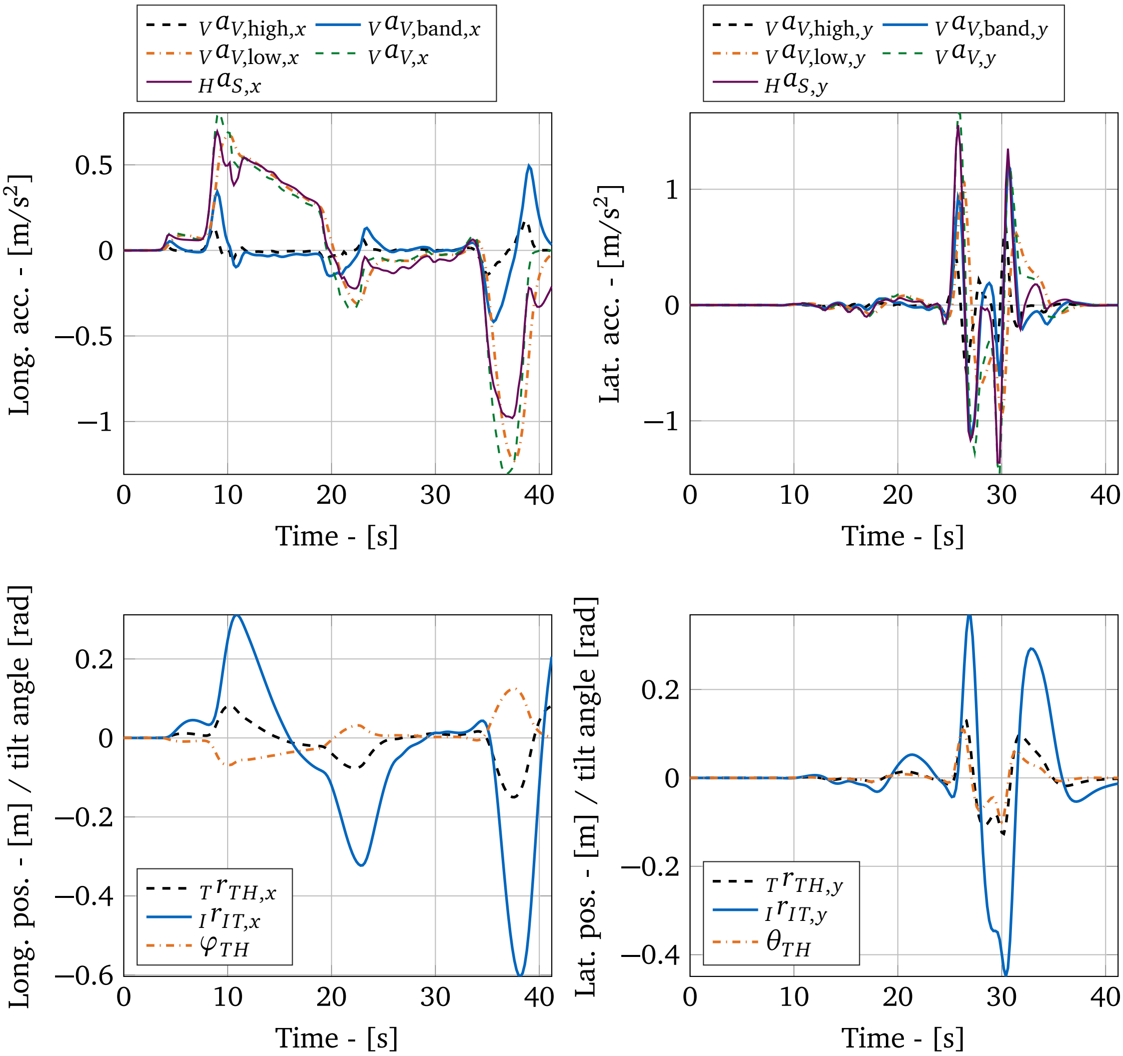


Figure 3.10: Top plots: The frequency divider separates the scaled reference signal $_V\boldsymbol{a}_V$ into the low frequency signal $_V\boldsymbol{a}_{V,\text{low}}$, the middle-frequency signal $_V\boldsymbol{a}_{V,\text{band}}$ and the high-frequency signal $_V\boldsymbol{a}_{V,\text{high}}$ as introduced in Figure 3.8. Lower plots: Resulting motions of tripod and hexapod and tilting angles.

Chapter 4

Open-Loop, Full-Time Prediction MCA (MCA$_{\mathrm{OFF}}$)

The content of the following Chapter have been published in [52].

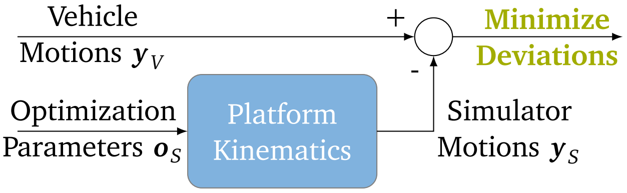

Vehicle
Motions $\boldsymbol{y}_V$
+
-
Minimize
Deviations
Optimization
Parameters $\boldsymbol{o}_S$
Platform
Kinematics
Simulator
Motions $\boldsymbol{y}_S$

Figure 4.1: Schematic scheme of the optimal control problem.

This Chapter presents an MCA which calculates the simulator motions based on an optimization scheme, in contrast to Chapter 3, where the simulator motions are mainly calculated by transfer functions. The open-loop, optimization-based MCA computes an optimal trajectory for a pre-defined manoeuvre in an offline calculation step. Herein, the pre-recorded driving dynamics data is taken as a reference over the complete manoeuvre in a single optimal control problem. The MCA is abbreviated throughout the work with MCA$_{\mathrm{OFF}}$.
The basic structure of the algorithm is shown in Figure 4.1. The upper input is composed of the vehicle's reference data

$$\boldsymbol{y}_V = \begin{bmatrix} {}_V\boldsymbol{f}_V^{ref} \\ {}_V\boldsymbol{\omega}_V^{ref} \end{bmatrix} = \begin{bmatrix} {}_V\boldsymbol{a}_V^{ref} \\ {}_V\boldsymbol{J}\,\dot{\boldsymbol{\beta}}_V^{ref} \end{bmatrix} \tag{4.1}$$

which includes the specific forces ${}_V\boldsymbol{f}_V^{ref}$ and angular velocities ${}_V\boldsymbol{\omega}_V^{ref}$ at the CRP, herein the index ref indicates the reference data base. The angular velocity is obtained by ${}_V\boldsymbol{\omega}_V^{ref} = {}_V\boldsymbol{J}\,\dot{\boldsymbol{\beta}}_V^{ref}$, where the Jacobian matrix ${}_V\boldsymbol{J}$ describes a transformation between the Cardan angles' derivatives $\dot{\boldsymbol{\beta}}_V^{ref}$ and the angular velocities ${}_V\boldsymbol{\omega}_V^{ref}$. The lower inputs in Figure 4.1 contain the optimization parameters $\boldsymbol{o}_S$, which are the motion system's states on position, velocity and acceleration level. The platform kinematics estimate the resulting motions $\boldsymbol{y}_S$ from equation (2.43) acting at the motion system's CRP containing the specific forces (cf. equation (2.17)) and angular velocities (cf. equation (2.10)). An optimal control problem minimizes deviations between the vehicle's reference data $\boldsymbol{y}_V$ and the simulator motions $\boldsymbol{y}_S$. In other words, the simulator motions $\boldsymbol{y}_S$ aims at tracking the vehicle reference motions $\boldsymbol{y}_V$. The minimization is constrained by the workspace restrictions of the motion system on position, velocity and acceleration level. Besides, the optimization problem considers the nonlinear kinematic chain of the motion system and includes the actuator limitations as constraints to exploit the workspace to a large extent.

The minimization is performed over the complete time horizon of the pre-defined manoeuvre and uses an ideal reference signal $\boldsymbol{y}_V$. Thus, depending on the chosen weighting parameters and the used simulator model, an optimal platform trajectory is given. For an open-loop application, real-time requirements do not have to be fulfilled. Thus, the difference between desired and provided percepts can be minimized further compared to a closed-loop application.

The following Sections describe the kinematics of the platform motions, the integration of the workspace restriction and finally introduces the considered optimization problem.

4.1 Related Work

In [36, 97, 137], an optimization-based MCA is proposed which exhibits a structure similar to the MCA$_{\text{OFF}}$. A cost function minimizes deviations between desired vehicle motions and actual simulator motions. Constraints bound the workspace within the limits of the motion system on actuator level. As the algorithm is used for open-loop simulations, the pre-recorded driving dynamics data were used for an ideal prediction strategy. In [36], the optimization-based algorithm is compared to a filter-based MCA in order to measure the motion cueing quality. For filter-based MCAs, only the current driving dynamics states are considered in the calculation of the simulator motions. Whereas, for optimization-based MCAs, the driving dynamics states over a future time horizon are taken into account in the simulator's motion calculation. Thus, pre-positioning is provided to exploit the simulator's workspace capabilities to a larger extent and to replicate the desired driver motions with a greater fidelity. Results show that the motion cueing quality of the optimization-based algorithm is higher, which points out the potential of these algorithms. One reason is the usage of predictions for upcoming driving manoeuvres. In [53, 95], it is shown that deviations between expected and actual motion percepts are reduced to a greater extent with increasing time horizon lengths. For the analysis, a perfect prediction is taken for pre-defined manoeuvres. A long time horizon enables an earlier pre-positioning of the motion platform yielding a better workspace exploitation.

4.2 Inverse Kinematics

The actuator limitations are included by integrating the inverse kinematics on position, velocity and acceleration level (see Section 2.5.1). Figure 2.3 shows the vector chain for a parallel robot, where the FoR I is fixed to the lower platform and the FoR N is anchored at the upper platform. The kinematic chain described in Figure 2.3 (see Section 2.4) represents either the kinematics of the actuation of the tripod (in which case N in Figure 2.4 is to be seen as T) or the actuation of the hexapod (in which case N in Figure 2.4 is to be seen as H). The aim of the inverse kinematics is to calculate the length of the actuators of the tripod $\boldsymbol{q}_t$ and the hexapod $\boldsymbol{q}_h$ (and the actuator velocities and actuator accelerations). Herein, the given positions, velocities and accelerations of the tripod's reference point ${}_I\boldsymbol{r}_{IT}$ (and orientation $\boldsymbol{\beta}_{IT}$) and the hexapod's reference point ${}_T\boldsymbol{r}_{TH}$ (and orientation $\boldsymbol{\beta}_{TH}$) are used. The actuator states will be employed later as constraints in the optimization. For the hexapod, the CRP has to be considered in the closed-loop vector chain of the parallel robot structure. The stroke of a single hexapod actuator $q_{h,i}$ is given by the quadratic norm

$$\begin{aligned} q_{h,i} &= ||\boldsymbol{q}_{L_iU_i}||_2 \\ &= ||-{}_T\boldsymbol{l}_{IL_i} + {}_T\boldsymbol{r}_{I,CRP} + {}_T\boldsymbol{R}_H\, ({}_H\boldsymbol{u}_{HU_i} - {}_H\boldsymbol{r}_{H,CRP})||_2 \end{aligned} \tag{4.2}$$

for $i = 1\dots 6$, where ${}_T\boldsymbol{l}_{IL_i}$ and ${}_H\boldsymbol{u}_{HU_i}$ describe the relative position of the lower and upper Cardan joint of actuator i [115]. These are known properties of the simulator.
The actuator stroke of the tripod is calculated by

$$q_{t,i} = ||-{}_I\boldsymbol{l}_{IL_i} + {}_I\boldsymbol{r}_{IT} + {}_I\boldsymbol{R}_T\ {}_T\boldsymbol{u}_{TU_i}||_2 \tag{4.3}$$

for $i = 1\dots 3$, where ${}_I\boldsymbol{l}_{IL_i}$ and ${}_T\boldsymbol{u}_{TU_i}$ describe the position of the lower and upper rotational joint of actuator i of the tripod [115].
The actuators' velocities

$$\dot{q}_{j,i} = \frac{\mathrm{d}q_{j,i}}{\mathrm{d}t} \qquad \text{for} \begin{cases} j = t,h \\ i = \text{actuator number} \end{cases} \tag{4.4}$$

and the actuators' accelerations

$$\ddot{q}_{j,i} = \frac{\mathrm{d}^2 q_{j,i}}{\mathrm{d}^2 t} \qquad \text{for} \begin{cases} j = t,h \\ i = \text{actuator number} \end{cases} \tag{4.5}$$

are expressed in equations (2.28) and (2.32), respectively. For the MCA$_{\mathrm{OFF}}$, the derivation with respect to time for the inverse kinematics in equations (4.2) and (4.3) is calculated in the symbolic toolbox CasADI which is a symbolic framework for algorithmic differentiation and numeric optimization [4]. CasADI offers an option to build a C-function of the derivative which can be directly integrated to the optimization solver. Thus, programming of the complicated results in equations (2.28) and (2.32), which are prone to errors, is not necessary.

4.3 Optimization Problem

The optimization problem is defined as

$$\begin{aligned} \min_{\boldsymbol{o}_S} \quad & J = \sum_{i=1}^{H_p} ||\boldsymbol{y}_S[i] - \boldsymbol{y}_V[i]||^2_{\boldsymbol{Q}} + \sum_{i=1}^{H_p} ||\boldsymbol{o}_S[i]||^2_{\boldsymbol{R}} \\ s.t.: & \\ \forall i = 1\dots H_p \quad & \boldsymbol{F}_S[i] = \boldsymbol{0} \\ \text{for } j = t,h \quad & \boldsymbol{o}_{S,min} \leq \boldsymbol{o}_S[i] \leq \boldsymbol{o}_{S,max} \\ & \boldsymbol{q}_{S,j,min} \leq \boldsymbol{q}_{S,j}[i] \leq \boldsymbol{q}_{S,j,max} \\ & \dot{\boldsymbol{q}}_{S,j,min} \leq \dot{\boldsymbol{q}}_{S,j}[i] \leq \dot{\boldsymbol{q}}_{S,j,max} \\ & \ddot{\boldsymbol{q}}_{S,j,min} \leq \ddot{\boldsymbol{q}}_{S,j}[i] \leq \ddot{\boldsymbol{q}}_{S,j,max} \end{aligned} \tag{4.6}$$

with the optimization parameters

$$\begin{aligned} \boldsymbol{o}_S = [\ & {}_T\boldsymbol{r}_{TH}, {}_T\dot{\boldsymbol{r}}_{TH}, {}_T\ddot{\boldsymbol{r}}_{TH}, \boldsymbol{\beta}_{TH}, \dot{\boldsymbol{\beta}}_{TH}, \\ & {}_I\boldsymbol{r}_{IT}, {}_I\dot{\boldsymbol{r}}_{IT}, {}_I\ddot{\boldsymbol{r}}_{IT}, \boldsymbol{\beta}_{IT}, \dot{\boldsymbol{\beta}}_{IT}\ ]. \end{aligned} \tag{4.7}$$

In this formulation the simulator dynamics are not considered, but limits related to the dynamics of the simulator are indirectly included by the constraints. The choice of the optimization parameters enables the calculation of the actuators' states, using the inverse kinematics. But also, regarding equations (2.10) and (2.17), the reference signal $\boldsymbol{y}_S$ can be obtained by $\boldsymbol{o}_S$. The objective function in equation (4.6) is divided into two parts. The first term tracks the reference signal $\boldsymbol{y}_S$ over the time horizon $H_p \cdot T_{samp}$ weighted by the matrix $\boldsymbol{Q}$. The second term penalizes the optimization parameters for deviations from the neutral position. The weighting matrix $\boldsymbol{R}$ defines the intensity of a “neutral push" to the middle of the motion system’s workspace (similar to the washout feature of the filter-based MCA described in Section 3.4). The hand-tuned weights are shown in Appendix D in Table D.4.

The parameters $\boldsymbol{o}_S$ can be used to determine the states of the actuators by the inverse kinematics, described in Section 4.2. The actuator states are included to the optimization as nonlinear inequality constraints. The term $\boldsymbol{q}_S$ contains all actuator strokes q_i of the respective motion subsystem. Analogously, the actuator velocities $\dot{q}_i$ and actuator accelerations $\ddot{q}_i$ are contained in $\dot{\boldsymbol{q}}_S$ and $\ddot{\boldsymbol{q}}_S$, respectively. The indices min and max indicate the limitations for the respective states. The optimal control problem is discretized with a multiple-shooting approach. The method discretizes the optimization parameters $\boldsymbol{o}_S$ over the complete time horizon. Thus, all time steps of the discretization are considered in the optimization problem at once. Different numerical optimization methods (including multiple-shooting) and optimization solvers are described in Appendix C. As a distinction to the formulation of the optimization problem in the appendix, the differential constraints in the form $\dot{x} = f(x)$ consist of the derivative relations of the optimization parameter $\boldsymbol{o}_S$. Time-differential relations between the optimization parameters are approximated as a double integrator. Positions, velocities and accelerations are described as relative parameters concerning their subsystem in the form

tripod:

$${}_I\dot{\boldsymbol{r}}_{IT}[i] = \frac{{}_I\boldsymbol{r}_{IT}[i+1] - {}_I\boldsymbol{r}_{IT}[i]}{T_{samp}}, \tag{4.8a}$$

$${}_I\ddot{\boldsymbol{r}}_{IT}[i] = \frac{{}_I\dot{\boldsymbol{r}}_{IT}[i+1] - {}_I\dot{\boldsymbol{r}}_{IT}[i]}{T_{samp}}, \tag{4.8b}$$

$$\dot{\boldsymbol{\beta}}_{IT}[i] = \frac{\boldsymbol{\beta}_{IT}[i+1] - \boldsymbol{\beta}_{IT}[i]}{T_{samp}}, \tag{4.8c}$$

hexapod:

$${}_T\dot{\boldsymbol{r}}_{TH}[i] = \frac{{}_T\boldsymbol{r}_{TH}[i+1] - {}_T\boldsymbol{r}_{TH}[i]}{T_{samp}}, \tag{4.8d}$$

$${}_T\ddot{\boldsymbol{r}}_{TH}[i] = \frac{{}_T\dot{\boldsymbol{r}}_{TH}[i+1] - {}_T\dot{\boldsymbol{r}}_{TH}[i]}{T_{samp}}, \tag{4.8e}$$

$$\dot{\boldsymbol{\beta}}_{TH}[i] = \frac{\boldsymbol{\beta}_{TH}[i+1] - \boldsymbol{\beta}_{TH}[i]}{T_{samp}}. \tag{4.8f}$$

Equality constraints $\boldsymbol{F}_S[i] = \boldsymbol{0}$ with the time-discretization steps $i \in [1 \ldots H_p]$ include the derivative relations between positions, velocities and accelerations, described in equations (4.8), of the motion systems $j = IT,\ TH$

$$\boldsymbol{F}_{S,j}[i] = \begin{bmatrix} \boldsymbol{r}_j[i+1] & - & \boldsymbol{r}_j[i] & - & T_{samp}\ \dot{\boldsymbol{r}}_j[i] \\ \dot{\boldsymbol{r}}_j[i+1] & - & \dot{\boldsymbol{r}}_j[i] & - & T_{samp}\ \ddot{\boldsymbol{r}}_j[i] \\ \boldsymbol{\beta}_j[i+1] & - & \boldsymbol{\beta}_j[i] & - & T_{samp}\ \dot{\boldsymbol{\beta}}_j[i] \end{bmatrix} = \boldsymbol{0} \tag{4.9}$$

which are discretized by an explicit Euler approach with the sample time T_{samp}. The simulator exhibits limitations not only due to the working range of its actuated legs (so at actuator level), but also due to the available space in the room to move the platform (hence in the workspace). These workspace limitations are a safety redundancy by the manufacturer and are included as constraints (summarized in Table 2.2) to the simulator control software. Therefore the optimization problem (4.6) also includes inequality constraints for the optimization parameters $\boldsymbol{o}_S$ in addition to those on the kinematics of the legs.

Actuator limitations are included for tripod and hexapod on position, velocity and acceleration level. Therefore, the inverse kinematics introduced in equations (4.2) and (4.3) are

considered in the optimization problem shown in equation (4.6).
The optimization problem is solved in a C++ environment using the software package *IPOPT* for large-scale nonlinear optimization [180].

4.4 Simulative Analysis of MCA$_{\mathrm{OFF}}$ and MCA$_{\mathrm{FILT}}$

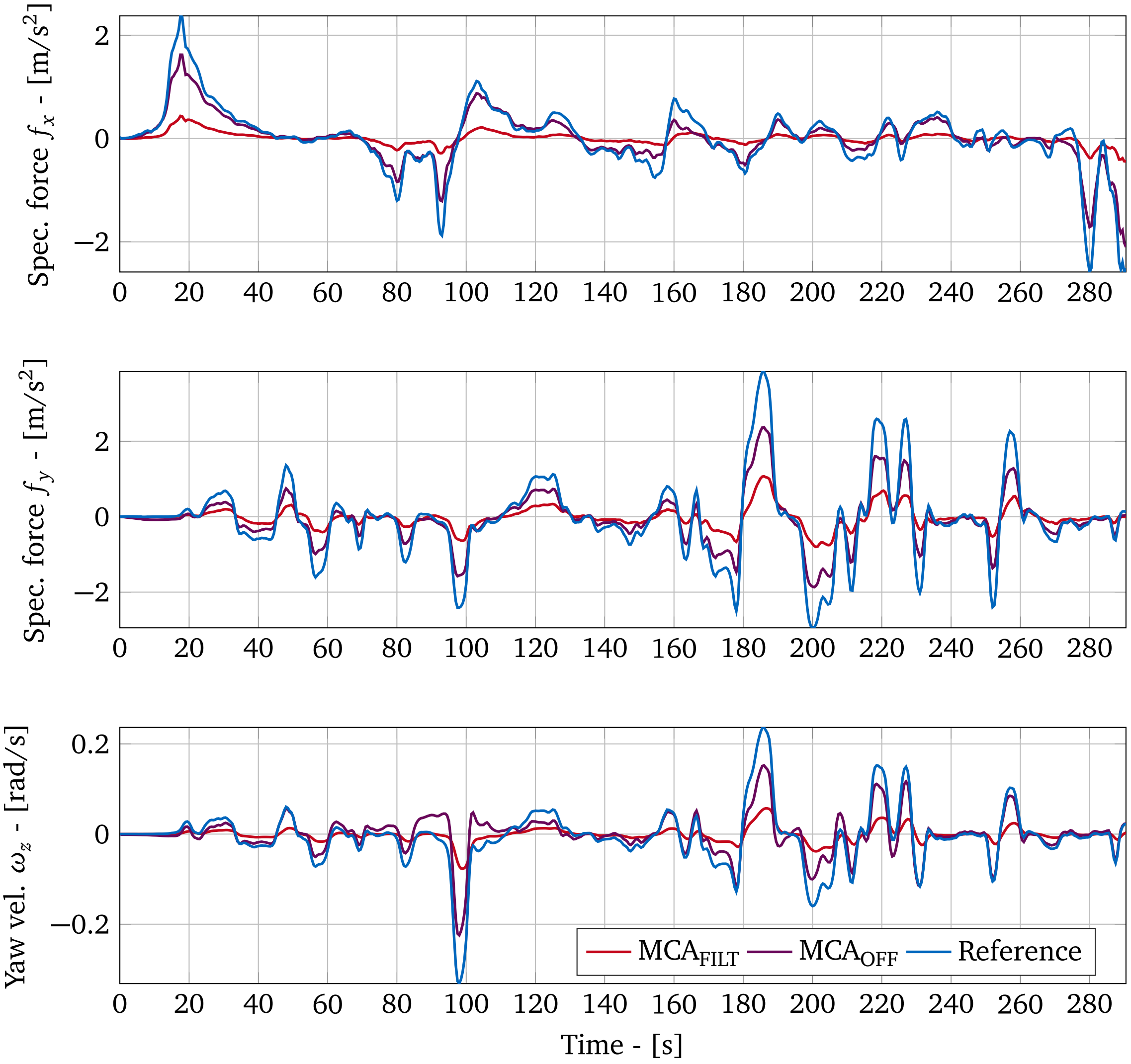


Figure 4.2: Resulting specific forces f_x, f_y and angular velocities ω_z for the MCA$_{\mathrm{FILT}}$ (input scaling factor = 0.3) and the MCA$_{\mathrm{OFF}}$ compared to the unscaled reference motions of a virtual vehicle.

Figure 4.2 shows the simulative motion cueing results for the manoeuvre described in Section 8.2 as a reference manoeuvre. The main motion channels are shown for the unscaled reference signal, a filter-based MCA$_{\mathrm{FILT}}$ and the optimization-based MCA$_{\mathrm{OFF}}$ in longitudinal, lateral and yaw direction at the CRP. For the MCA$_{\mathrm{FILT}}$, the input signal is scaled by a factor of 0.3 to maximally exploit the motion system's workspace. The plots reveal that better cueing is provided by the MCA$_{\mathrm{OFF}}$ compared to the MCA$_{\mathrm{FILT}}$. In [67, 78], these deviations are categorized in groups:

1. Missing cues: Not provided cues.

2. False cues: Cues without a stimuli demand.
3. Phase errors: Temporarily shifted stimuli.
4. Scaling errors: Different magnitude between provided and demanded stimuli with correct sign.
5. Sign errors: Opposite signs between provided and demanded stimuli.

Errors, such as the ones defined above, can generally be observed more significantly for the $\mathrm{MCA_{FILT}}$ than for the $\mathrm{MCA_{OFF}}$. To quantify the deviations between both algorithms, the root-mean-square (rms) error to the reference signal is shown in Table 4.1. The rms-values are higher for the $\mathrm{MCA_{FILT}}$ considering all six motion channels.

Table 4.1: Root-mean-square of the objective motion incongruencies for $\mathrm{MCA_{FILT}}$ and $\mathrm{MCA_{OFF}}$ to the reference motions of a virtual vehicle.

	Trans. acc. - [m/s^2]			Ang. vel. [rad/s]		
	x	y	z	x	y	z
$\mathrm{MCA_{FILT}}$	0.516	0.775	0.15	0.03	0.014	0.051
$\mathrm{MCA_{OFF}}$	0.196	0.394	0.15	0.02	0.012	0.029

4.5 Chapter Summary

This Chapter proposes an open-loop, optimization-based MCA which includes an optimal control problem to minimize deviations between desired and actual simulator motions. As the driving dynamics data was known a priori for a pre-defined manoeuvre, an ideal prediction over the complete manoeuvre was applied. In a comparison with the filter-based $\mathrm{MCA_{FILT}}$, the $\mathrm{MCA_{OFF}}$ shows considerably less motion incongruencies which indicates its potential in motion cueing quality towards closed-loop MCAs.

Chapter 5

Structure of Model Predictive Control (MPC)-Based MCA

The content of the following Chapter have been published in [51, 52, 56].

This Chapter presents the optimization-based MCA$_{\text{MPC}}$ which is similar to the MCA$_{\text{OFF}}$ (described in Chapter 4), but differs in the capability to run in a closed-loop simulation.
The architecture of the open-loop MCA$_{\text{OFF}}$ does not enable online applications, since it solves a single optimization problem to obtain the simulator motions over the complete prerecorded manoeuvre. Moreover, it uses a complex nonlinear description of the relative and inverse simulator kinematics.
The MCA$_{\text{MPC}}$ also uses a global optimization scheme to solve the nine redundant DoFs subject to the technical restrictions of the motion system, shown in Figure 1.3. However, the algorithm distinguishes from the MCA$_{\text{OFF}}$ in the following aspects:

- The global approach of the MCA$_{\text{MPC}}$ pursues the goal to be more efficient regarding the computational time of the optimization problem. Thus, a linearized prediction model is used in the optimization to calculate:
  - the relative kinematics (specific forces and angular velocities),
  - the inverse kinematics (actuator constraints) and
  - the predicted trajectories (simulator states).
- The optimization process of the MCA$_{\text{MPC}}$ is included into a model predictive control (MPC) scheme which yields the general advantages:
  - to run in real-time,
  - to cope with multivariable control problems,
  - to consider actuator limitations and
  - to allow operations closer to the constraints. [106]
- The MCA$_{\text{MPC}}$ includes a model of the vestibular system so that only perceivable motions are replicated by the simulator.
- In a closed-loop simulation, the future driving dynamics data is not known for generating the reference signal. Thus, a strategy to predict the future driving behaviour is necessary.

5.1 Related Work

Optimization-based MCAs are composed of an objective function which is minimized while subjected to constraints. For driving simulation, the objective function minimizes deviations between simulator motions and a reference trajectory describing the future motions in a virtual vehicle over a prediction horizon [5, 9, 41, 62]. The following Section gives a brief overview about various prediction strategies and ways to consider the simulator's technical restrictions.

Prediction: Optimization parameters aim to replicate the reference signal and ensure that the simulator constraints, such as actuator limitations, are not violated. Hence, a prediction model is necessary to forecast the future driving behaviour. However, in many practical cases, accurate information about future motions of the virtual vehicle is not available, for which reason it has been proposed to keep the current driving input constant over the prediction horizon [26]. The assumption is sufficiently valid for a short time horizon, for which reason in a later work [24], a *Look-Ahead* prediction is presented. Herein, the authors used the driving dynamics data of previously driven round courses for the reference prediction. The approach leads to an effective pre-positioning, as long as the driver behaviour does not vary strongly between the laps, as otherwise, an unreliable behaviour is used in the prediction. In [45], an online capable driver, environment and vehicle model was introduced which follows a two-dimensional road centreline. Because of the simplicity of the model, they were able to predict the driving behaviour for the upcoming ten seconds. Their results demonstrate that small improvements in the prediction method can result in considerable improvements in the motion cueing quality. These algorithms are based on an MPC-scheme, which enables closed-loop applications. An MPC-scheme solves an optimization problem at discrete time steps. The optimization includes the current driving dynamics state in the reference signal and sends the optimal motion inputs to the simulator.

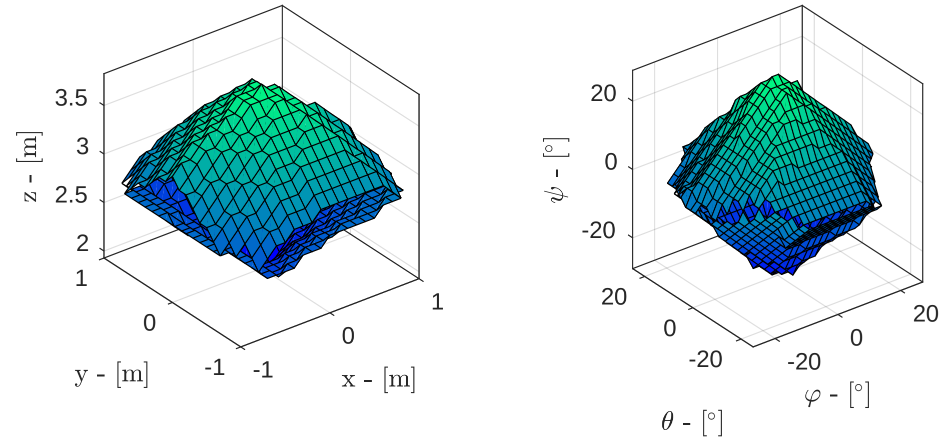


Figure 5.1: Left: translational workspace of a hexapod with $\varphi = \theta = \psi = 0°$. Right: rotational workspace with $x = y = z = 0\,\text{m}$.

Constraints: For a six DoFs hexapod simulator, an approach is to define the constraints on position, velocity and acceleration level in workspace coordinates. In [5, 9], the translational and rotational limitations are modeled as constraints in workspace coordinates that show a fast handling by the optimization solver. However, the approximation of the workspace displays relatively large deviations since the DoFs are coupled to each other. Workspace limitations are a function depending on the simulator's DoFs, meaning that variations of the

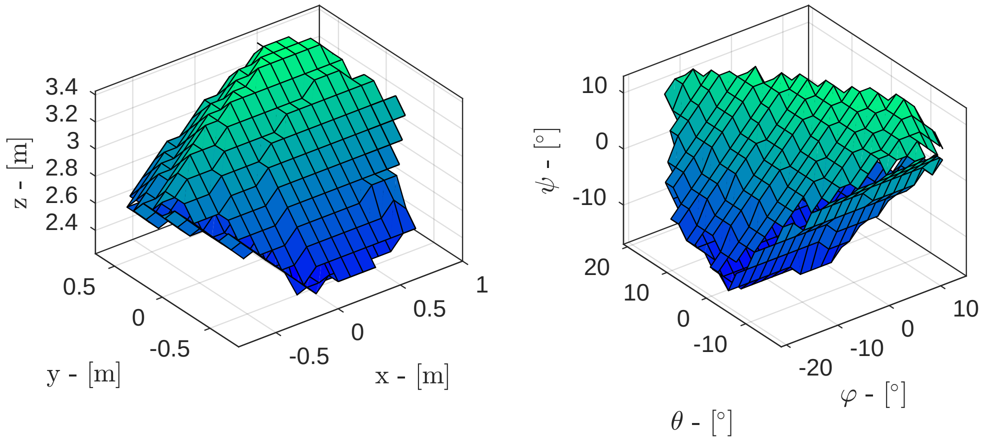


Figure 5.2: Left: the translation workspace of a hexapod with $\varphi = \theta = \psi = 10°$. Right: rotational workspace with $x = 0.5\,\text{m}$, $y = 0.5\,\text{m}$, $z = 0\,\text{m}$.

simulator configuration result in an adjustment of the workspace limitations. This fact applies for the position, velocity and acceleration level. To illustrate this characteristics, an example is shown in the following:

The left part of Figure 5.1 shows the approximated translational workspace of a hexapod, where the rotational DoFs are set to zero ($\varphi = \theta = \psi = 0°$). The right part of Figure 5.1 shows the approximated rotational workspace of the same hexapod structure in the neutral translational position ($x = y = z = 0\,\text{m}$). A grid search algorithm estimates the workspace envelope of the parallel robot, where the approximation error depends on the chosen grid size. The discrete, geometrical approach uses iteration loops to check compliance with the actuator length for each point in a three-dimensional set of points [102, 114]. In case different sets of dots in the workspace are kept constant as used in Figure 5.1, the workspace envelope varies. The left part of Figure 5.2 shows the approximated translational workspace ($\varphi = \theta = \psi = 10°$) and the right part the approximated rotational workspace ($x = 0.5\,\text{m}$, $y = 0.5\,\text{m}$, $z = 0\,\text{m}$). Comparing Figure 5.1 and Figure 5.2, a significant variation in the workspace envelope is apparent. Approximating the workspace envelope as an analytical function depending on a current platform configuration $\boldsymbol{w}_S$ is still part of current research. [114, 115]

Based on these observations, different approaches are reported in literature to consider the simulator restrictions in an optimization-based MCA.

In [41, 62], an MPC-based approach is proposed for an eight DoFs simulator, consisting of an $x-y-$ table and hexapod. The work separates longitudinal and lateral motions by a respective MPC scheme. Each MPC scheme exhibits two DoFs, comprising a translational motion executed by the $x-y-$ table and a rotational motion performed by the hexapod. Tilting the hexapod platform supports translational acceleration by using the gravitational force so that low frequency accelerations can be realized. The optimization problem proposed by those authors does not include hard workspace constraints. Instead, a washout filter is integrated to limit the output of the MPC scheme.

A nonlinear MPC scheme for a nine DoFs simulator introduced by [25, 27] considers the nonlinear relations between tripod and hexapod. Driving dynamics data is prefiltered to distribute translational motions between high frequency translational signals and low frequency tilting ones. Motion system's limitations are included as nonlinear constraints on actuator level. The inverse kinematics is hereby solved by transforming workspace coordinates to actuator ones. Constraining on actuator level was also realized for an eight DoFs simulator by [176].

Including nonlinear constraints is computational expensive and therefore, in [70], a linearization approach to approximate the hexapod's inverse kinematics is shown. A constant Jacobian matrix is used for the kinematic transformation from workspace velocities to actuator velocities. The linearized actuator velocities are integrated over time to receive the actuator strokes.
In [60], an MCA approach for a nine DoFs simulator is proposed consisting of $x-y-$ table, hexapod and yaw-rotation-table. Considering the nonlinear kinematics, the work divided the motion system up into independent subsystems and implemented MPC-schemes for each of these. Translational accelerations are split into horizontal ones that are performed by the $x-y-$ table and tilting ones, executed by the hexapod. Simulator limitations are added on workspace level.

5.2 Contribution

In this chapter, an MPC-based MCA is proposed that is comparable to the approach in [28], but differs in the description of the kinematics of the motion simulator. Whereas, in [28], the angular relations of the redundant motion system are neglected to estimate the driver's motion in the simulator, the present optimization scheme includes the kinematic chain to estimate the perceived driver's motion, composed of the motions of tripod and hexapod. As the kinematic chain exhibits nonlinearities, an approximation approach is shown to result in a linear prediction model. Thus, a convex Quadratic Programming (QP) problem based on a linear prediction model is solved which reduces the complexity of the optimization problem. Employing linear models in MPC is widespread as these models provide good results when operating in the area of the plant's operating point. By contrast, a nonlinear prediction model can yield a nonconvex problem, where the robustness and stability are much more difficult to obtain than for a QP-problem. Along with this, the higher complexity can lead to higher computational cost which can degrade the real-time capabilities. [32]
An additional distinction to the work [28] persists in the formulation of the optimization constraints. In [28], the limitations of the simulator are described as nonlinearities on actuator level. In the present work, the linearization approach based on [70] is applied to approximate the inverse kinematics in order to obtain linear inequality constraints for the QP-problem. The linearization is not only applied on a hexapod system but also on a tripod system. Based on [10], the prediction horizon is divided into three parts, considering input, constraint and prediction horizons to save further computation time.

5.3 Linearized State Space Model

Figure 5.3 shows the basic structure of the optimization problem. An external software estimates the driving dynamics states of the virtual vehicle subject to the driver inputs. The upper block S_V (index V for *Vehicle*) calculates the vehicle motions' states $\boldsymbol{u}_V$ comprising the translational accelerations ${}_V\boldsymbol{a}_V$ and the derivatives of the Cardan angles $\dot{\boldsymbol{\beta}}_V$. Hereby, V defines the relative FoR located at the driver's head as described in Figure 2.1. The *Driver* block estimates the specific forces ${}_V\boldsymbol{f}_V$ and the angular velocities ${}_V\boldsymbol{\omega}_V$ acting on the driver in the form

$$ {}_V\boldsymbol{f}_V = {}_V\boldsymbol{a}_V, \tag{5.1}$$

$$ {}_V\boldsymbol{\omega}_V = {}_V\boldsymbol{J}\ \dot{\boldsymbol{\beta}}_V. \tag{5.2}$$

The Jacobian matrix ${}_V\boldsymbol{J}$ describes a transformation between the Cardan angles' derivatives $\dot{\boldsymbol{\beta}}_V$ and the angular velocities $\boldsymbol{\omega}_V$.
The lower block S_S (index S for *Simulator*) represents the simulator system with the unknown simulator's input $\boldsymbol{u}_S$ taken as optimization parameters. The input

$$\boldsymbol{u}_S = [\dot{\varphi}_{TH}\ \dot{\theta}_{TH}\ \dot{\psi}_{TH}\ \dot{\psi}_{IT}\ {}_T\ddot{r}_{TH,x}\ {}_T\ddot{r}_{TH,y}\ {}_T\ddot{r}_{TH,z}\ {}_I\ddot{r}_{IT,x}\ {}_I\ddot{r}_{IT,y}]^T \qquad \in \mathbb{R}^{9\times 1} \tag{5.3}$$

consists of the Cardan angles' derivatives $\dot{\boldsymbol{\beta}}_{TH} = [\dot{\varphi}_{TH}\ \dot{\theta}_{TH}\ \dot{\psi}_{TH}]^T$ of the hexapod and of the tripod's yaw-rotation $\dot{\boldsymbol{\beta}}_{IT} = [0\ 0\ \dot{\psi}_{IT}]^T$ as well as the relative translational accelerations ${}_T\ddot{\boldsymbol{r}}_{TH} = [{}_T\ddot{r}_{TH,x}\ {}_T\ddot{r}_{TH,y}\ {}_T\ddot{r}_{TH,z}]^T$ and ${}_I\ddot{\boldsymbol{r}}_{IT} = [{}_I\ddot{r}_{IT,x}\ {}_I\ddot{r}_{IT,y}\ 0]^T$ with the FoR introduced in Figure 2.4. The input $\boldsymbol{u}_V$ is taken as a reference to minimize deviations between vehicle and simulator motions in an objective function. The following subsections describe the various blocks of Figure 5.3 in more detail.

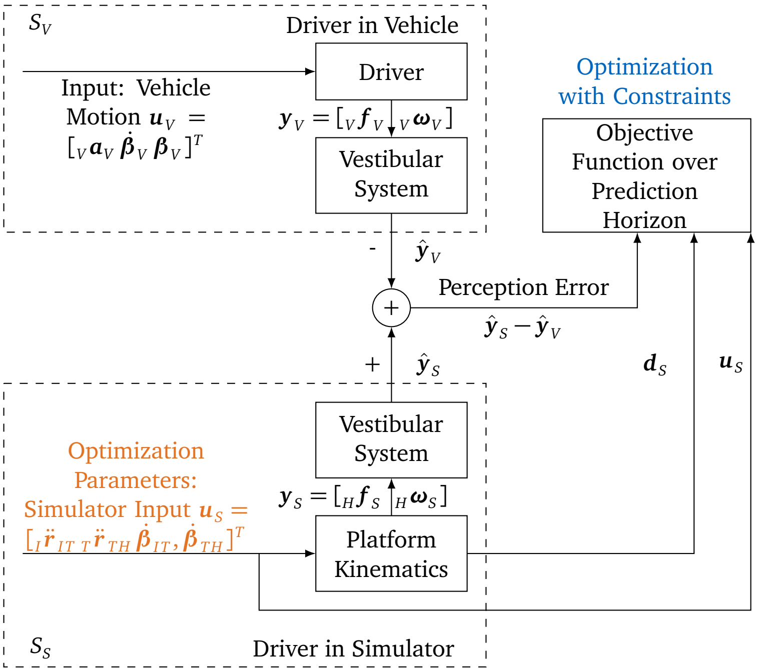

S_V
Driver in Vehicle
Driver
Input: Vehicle Motion $\boldsymbol{u}_V = [{}_V\boldsymbol{a}_V\ \dot{\boldsymbol{\beta}}_V\ \boldsymbol{\beta}_V]^T$
$\boldsymbol{y}_V = [{}_V\boldsymbol{f}_V\ {}_V\boldsymbol{\omega}_V]$
Vestibular System
Optimization with Constraints
Objective Function over Prediction Horizon
-
$\hat{\boldsymbol{y}}_V$
Perception Error
+
$\hat{\boldsymbol{y}}_S - \hat{\boldsymbol{y}}_V$
+
$\hat{\boldsymbol{y}}_S$
$\boldsymbol{d}_S$
$\boldsymbol{u}_S$
Vestibular System
Optimization Parameters: Simulator Input $\boldsymbol{u}_S = [{}_I\ddot{\boldsymbol{r}}_{IT}\ {}_T\ddot{\boldsymbol{r}}_{TH}\ \dot{\boldsymbol{\beta}}_{IT}, \dot{\boldsymbol{\beta}}_{TH}]^T$
$\boldsymbol{y}_S = [{}_H\boldsymbol{f}_S\ {}_H\boldsymbol{\omega}_S]$
Platform Kinematics
S_S
Driver in Simulator

Figure 5.3: Structure of the global optimization scheme with the optimization parameters $\boldsymbol{u}_S$ functioning to replicate the vehicle's motion $\boldsymbol{u}_V$.

5.3.1 Platform Kinematics

Simulator motions have to be described in the non-inertial H FoR so as to enable a comparison with the actual driving dynamics data ${}_V\boldsymbol{f}_V$ and ${}_V\boldsymbol{\omega}_V$. Herein, the specific force ${}_H\boldsymbol{f}_S$ is approximated with

$$\begin{aligned} {}_H\boldsymbol{f}_S &\approx {}_H\ddot{\boldsymbol{r}}_{IT} + {}_H\ddot{\boldsymbol{r}}_{TH} + {}_H\boldsymbol{g} \\ &= {}_H\boldsymbol{R}_I(\varphi_{TH}, \theta_{TH}, \psi_{IH})\ {}_I\ddot{\boldsymbol{r}}_{IT} + {}_H\boldsymbol{R}_T(\varphi_{TH}, \theta_{TH}, \psi_{TH})\ {}_T\ddot{\boldsymbol{r}}_{TH} + {}_H\boldsymbol{R}_I(\varphi_{TH}, \theta_{TH}, \psi_{IH})\ {}_I\boldsymbol{g} \end{aligned} \tag{5.4}$$

using the rotation matrices ${}_H\boldsymbol{R}_I(\varphi_{TH}, \theta_{TH}, \psi_{IH})$ and ${}_H\boldsymbol{R}_T(\varphi_{TH}, \theta_{TH}, \psi_{TH})$ from equations (2.8) and (2.7). The last term of equation (5.4) takes into account the gravitational force that influences translational accelerations by tilt motions. The gravitational force ${}_I\boldsymbol{g}$ is defined in the inertial FoR with ${}_I\boldsymbol{g} = [0\ 0\ g]^T$. Centripetal-, Coriolis- and rotational acceleration components, as introduced in equation (2.6), are neglected to further simplify the optimization

problem for potential real-time applications.
In order to estimate the error resulting from the approximation in equation (5.4), an analysis is performed. The specific force error ${}_{H}\bar{\bar{\boldsymbol{f}}}_{CRP}$ resulting from the Centripetal-, Coriolis- and rotational acceleration components is calculated in the form

$$\begin{aligned} {}_{H}\bar{\bar{\boldsymbol{f}}}_{CRP} = {}_{H}\boldsymbol{R}_{T}\, 2\, {}_{T}\tilde{\boldsymbol{\omega}}_{IT}\, {}_{T}\dot{\boldsymbol{r}}_{TH} + {}_{H}\boldsymbol{R}_{T} \cdot ({}_{T}\dot{\tilde{\boldsymbol{\omega}}}_{IT}\, {}_{T}\boldsymbol{r}_{TH} + {}_{T}\tilde{\boldsymbol{\omega}}_{IT}\, {}_{T}\tilde{\boldsymbol{\omega}}_{IT}\, {}_{T}\boldsymbol{r}_{TH}) \\ + {}_{H}\dot{\tilde{\boldsymbol{\omega}}}_{IH}\, {}_{H}\boldsymbol{r}_{H,CRP} + {}_{H}\tilde{\boldsymbol{\omega}}_{IH}\, {}_{H}\tilde{\boldsymbol{\omega}}_{IH}\, {}_{H}\boldsymbol{r}_{H,CRP}. \end{aligned} \tag{5.5}$$

These are the missing terms in equation (5.4) to estimate the specific force from equation (2.17). To determine the magnitude of the specific force error ${}_{H}\bar{\bar{\boldsymbol{f}}}_{CRP}$, the motion cueing data, calculated by the MCA$_{\text{MPC}}$, is taken. A roundabout manoeuvre is taken as a test-case for benchmarking the different algorithms, which exhibits high yaw rates and exploits the complete workspace of the hexapod and tripod. The maximum specific force error for the considered manoeuvre is ${}_{H}\bar{\bar{\boldsymbol{f}}}_{CRP,max} = [0.134\ 0.187\ 0.074]^{T}\ \text{m/s}^{2}$. This error is a rough indicator and higher values are possible. Still, assuming a perception threshold of $0.17\,\text{m/s}^{2}$ [141], the range of the error is reasonable to justify the approximations made in equation (5.4). Along with the dynamic motions of the simulator, the error depends on the geometric structure of the motion system. The hexapod of the considered motion system exhibits small translational motion space, yielding low values for ${}_{T}\boldsymbol{r}_{TH}$. The vector ${}_{H}\boldsymbol{r}_{H,CRP}$ depends on the design of the mockup and is constant throughout the simulation (see equation (2.18)).

In the following, the three rotational DoFs acting on the driver are described for the considered motion system. Observing that the third axis of the H and T FoR are always aligned, rotational velocities expressed in the H FoR are transformed by the Jacobian matrix ${}_{H}\boldsymbol{J}$

$${}_{H}\boldsymbol{\omega} = \begin{bmatrix} 1 & 0 & -s\theta_{TH} \\ 0 & c\varphi_{TH} & c\theta_{TH}s\varphi_{TH} \\ 0 & -s\varphi_{TH} & c\theta_{TH}c\varphi_{TH} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_{TH} \\ \dot{\theta}_{TH} \\ \dot{\psi}_{TH} + \dot{\psi}_{IT} \end{bmatrix} = {}_{H}\boldsymbol{J}\ \dot{\boldsymbol{\beta}}_{S}. \tag{5.6}$$

Inputs $\boldsymbol{u}_S$ are integrated in a state space model

$$\dot{\boldsymbol{d}}_S = \boldsymbol{A}_S\, \boldsymbol{d}_S + \boldsymbol{B}_S\, \boldsymbol{u}_S \tag{5.7}$$

to obtain

$$\begin{aligned} \boldsymbol{d}_S = [\varphi_{TH}\ \theta_{TH}\ \psi_{TH}\ \psi_{IT} \\ {}_{T}r_{TH,x}\ {}_{T}\dot{r}_{TH,x}\ {}_{T}r_{TH,y}\ {}_{T}\dot{r}_{TH,y}\ {}_{T}r_{TH,z}\ {}_{T}\dot{r}_{TH,z} \\ {}_{I}r_{IT,x}\ {}_{I}\dot{r}_{IT,x}\ {}_{I}r_{IT,y}\ {}_{I}\dot{r}_{IT,y}\ ]^{T} \qquad \in \mathbb{R}^{14\times1} \end{aligned} \tag{5.8}$$

with positions $\boldsymbol{r}$, velocities $\dot{\boldsymbol{r}}$ and angular configurations $\dot{\boldsymbol{\beta}}$ of the relative motion system needed later to express the constraints of the motion system. The matrices $\boldsymbol{A}_S$ and $\boldsymbol{B}_S$ include the derivative relations between $\boldsymbol{u}_S$ and $\boldsymbol{d}_S$ which are approximated by an explicit Euler discretization. For the sake of conciseness, the structure of the matrices $\boldsymbol{A}_S$ and $\boldsymbol{B}_S$ in 5.7 are described in a reduced formulation by considering just one arbitrary translational acceleration element $\ddot{w}_{S,i}$ out of $\boldsymbol{u}_S$ in the following way

$$\underbrace{\begin{bmatrix} \dot{w}_{S,i} \\ \ddot{w}_{S,i} \end{bmatrix}}_{\text{part of } \boldsymbol{d}_S} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\boldsymbol{A}^{*}_{S,i}} \begin{bmatrix} w_{S,i} \\ \dot{w}_{S,i} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\boldsymbol{B}^{*}_{S,i}} \underbrace{\ddot{w}_{S,i}}_{\text{part of } \boldsymbol{u}_S} \tag{5.9}$$

where $\boldsymbol{A}^{*}_{S,i}$ and $\boldsymbol{B}^{*}_{S,i}$ are parts of $\boldsymbol{A}_S$ and $\boldsymbol{B}_S$ from equation (5.7). Angular velocity elements $\dot{w}_{S,i}$ out of $\boldsymbol{u}_S$ are fed to a single integrator to obtain the respective angles.

5.3.2 Vestibular System

A model of the vestibular system is integrated to calculate the perceived motions of the simulator driver. Output signals of a vestibular system are indicated with the subscript '^'. The semicircular and otolith model of Telban et al. [168] is used for rotational and translational inputs, which is already introduced in equations (2.48) and (2.52), respectively. Perception thresholds are neglected as they would lead to a nonlinear transfer function.
Perceived rotational velocities $\hat{\boldsymbol{\omega}} = [\hat{\omega}_x \ \hat{\omega}_y \ \hat{\omega}_z]^T$ and specific forces $\hat{\boldsymbol{f}} = [\hat{f}_x \ \hat{f}_y \ \hat{f}_z]^T$ are estimated by

$$\hat{\omega}_{j,k} = H_{semi}(s)\ \omega_{j,k} = \frac{\tau_{S,semi}\ \tau_{L,semi}\ s^2}{(1+\tau_{L,semi}\, s)(1+\tau_{S,semi}\, s)}\ \omega_{j,k} \tag{5.10a}$$

$$\text{with } \tau_{S,semi} = 5.73,\ \tau_{L,semi} = 80$$

$$\hat{f}_{j,k} = H_{oto}(s)\ f_{j,k} = \frac{\kappa_{oto}\ (1+\tau_{A,oto}\, s)}{(1+\tau_{L,oto} s)\ (1+\tau_{S,oto} s)}\ f_{j,k} \tag{5.10b}$$

$$\text{with } \tau_{A,oto} = 10,\ \tau_{S,oto} = 0.016,\ \tau_{L,oto} = 5,\ \kappa_{oto} = 0.4$$

where $j = S, V$, $k = x, y, z$ and $H_{semi}(s)$ and $H_{oto}(s)$ are the Laplace-transformed semicircular and otolith perception models, respectively. Equations (5.10a) and (5.10b) are transformed to state space in the form

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{semi} \\ \dot{\boldsymbol{x}}_{oto} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{semi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_{oto} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{semi} \\ \boldsymbol{x}_{oto} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_{semi} \\ \boldsymbol{B}_{oto} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_j \\ \boldsymbol{f}_j \end{bmatrix} \tag{5.11a}$$

$$\hat{\boldsymbol{y}}_j = \begin{bmatrix} \boldsymbol{C}_{semi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_{oto} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{semi} \\ \boldsymbol{x}_{oto} \end{bmatrix} + \begin{bmatrix} \boldsymbol{D}_{semi} \\ \boldsymbol{D}_{oto} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_j \\ \boldsymbol{f}_j \end{bmatrix} \tag{5.11b}$$

with the output $\hat{\boldsymbol{y}}_j = [\hat{\boldsymbol{\omega}}_j \ \hat{\boldsymbol{f}}_j]^T$ for $j = S, V$. The following Section shows the structure of the semicircular and otolith state-space model.

Semicircular System, for $k \in \{x, y, z\}$:

1. State space matrix $\boldsymbol{A}_{semi}$

$$\boldsymbol{A}_{semi} = \text{blkdiag}(\boldsymbol{A}_{semi,k}), \quad \text{with} \quad \boldsymbol{A}_{semi,k} = \begin{bmatrix} 0 & 1 \\ \frac{-1}{\tau_{L,semi}\tau_{S,semi}} & -\frac{\tau_{L,semi}+\tau_{S,semi}}{\tau_{L,semi}\tau_{S,semi}} \end{bmatrix} \tag{5.12}$$

2. Input matrix $\boldsymbol{B}_{semi}$

$$\boldsymbol{B}_{semi} = \text{blkdiag}(\boldsymbol{B}_{semi,k}), \quad \text{with} \quad \boldsymbol{B}_{semi,k} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.13}$$

3. Output matrix $\boldsymbol{C}_{semi}$

$$\boldsymbol{C}_{semi} = \text{blkdiag}(\boldsymbol{C}_{semi,k}), \quad \text{with} \quad \boldsymbol{C}_{semi,k} = \begin{bmatrix} \frac{-1}{\tau_{L,semi}\tau_{S,semi}} & -\frac{\tau_{L,semi}+\tau_{S,semi}}{\tau_{L,semi}\tau_{S,semi}} \end{bmatrix} \tag{5.14}$$

4. Feedthrough matrix $\boldsymbol{D}_{semi}$

$$\boldsymbol{D}_{semi} = \boldsymbol{I}_{3x3} \qquad \text{with the identity matrix } \boldsymbol{I}. \tag{5.15}$$

Otolith System, for $k \in \{x, y, z\}$:

1. State space matrix $\boldsymbol{A}_{oto}$

$$\boldsymbol{A}_{oto} = \text{blkdiag}(\boldsymbol{A}_{oto,k}), \quad \text{with} \quad \boldsymbol{A}_{oto,k} = \begin{bmatrix} 0 & 1 \\ \frac{-1}{\tau_{L,oto}\tau_{S,oto}} & -\frac{\tau_{L,oto}+\tau_{S,oto}}{\tau_{L,oto}\tau_{S,oto}} \end{bmatrix} \tag{5.16}$$

2. Input matrix $\boldsymbol{B}_{oto}$

$$\boldsymbol{B}_{oto} = \text{blkdiag}(\boldsymbol{B}_{oto,k}), \quad \text{with} \quad \boldsymbol{B}_{oto,k} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.17}$$

3. Output matrix $\boldsymbol{C}_{oto}$

$$\boldsymbol{C}_{oto} = \text{blkdiag}(\boldsymbol{C}_{oto,k}), \quad \text{with} \quad \boldsymbol{C}_{oto,k} = \begin{bmatrix} \frac{1}{\tau_{L,oto}\tau_{S,oto}} & \frac{\kappa_{oto}\,\tau_{A,oto}}{\tau_{L,oto}\tau_{S,oto}} \end{bmatrix} \tag{5.18}$$

4. Feedthrough matrix $\boldsymbol{D}_{oto}$

$$\boldsymbol{D}_{oto} = \boldsymbol{0}_{3x3} \qquad \text{with the zero matrix } \boldsymbol{0}. \tag{5.19}$$

5.3.3 Resulting State Space Model

The gravity vector ${}_{H}\boldsymbol{g}$ in equation (5.4) is linearized by a small angle approximation in the form

$${}_{H}\boldsymbol{g} = \begin{bmatrix} -\sin(\theta_{TH}) \\ \cos(\theta_{TH})\sin(\varphi_{TH}) \\ \cos(\varphi_{TH})\cos(\theta_{TH}) \end{bmatrix} g \overset{\text{linearization}}{\approx} \begin{bmatrix} -\theta_{TH} \\ \varphi_{TH} \\ 1 \end{bmatrix} g \tag{5.20}$$

In driving simulation, roll and pitch rotations are typically small, keeping the error of the linearization in a reasonable range. Since the transformations ${}_{H}\ddot{\boldsymbol{r}}_{IT} = {}_{H}\boldsymbol{R}_{I}\ {}_{I}\ddot{\boldsymbol{r}}_{IT}$, ${}_{H}\ddot{\boldsymbol{r}}_{TH} = {}_{H}\boldsymbol{R}_{T}\ {}_{T}\ddot{\boldsymbol{r}}_{TH}$ in equation (5.4) and ${}_{H}\boldsymbol{\omega} = {}_{H}\boldsymbol{J}\ \dot{\boldsymbol{\beta}}_{S}$ in equation (5.6) cannot be linearized by a small angle approximation, the estimated angles $\varphi_{TH}[k]$, $\theta_{TH}[k]$, $\psi_{TH}[k]$ and $\psi_{IT}[k]$ are kept constant over a time horizon H_p.

Based on the linearization, the specific forces $\boldsymbol{f}_S$ and angular velocities $\boldsymbol{\omega}_S$ are expressed by

$$\begin{bmatrix} \boldsymbol{\omega}_S \\ \boldsymbol{f}_S \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{3x1} \\ \boldsymbol{K}_1\,\boldsymbol{d}_S \end{bmatrix} + \begin{bmatrix} {}_{H}\boldsymbol{J}\,\boldsymbol{K}_3 & \boldsymbol{0}_{3x5} \\ \boldsymbol{0}_{3x4} & \boldsymbol{K}_2 \end{bmatrix} \boldsymbol{u}_S \tag{5.21}$$

with

$$\boldsymbol{K}_1 = \begin{bmatrix} \boldsymbol{K}_1^* & \boldsymbol{0}_{3x10} \end{bmatrix}, \qquad \text{with} \qquad \boldsymbol{K}_1^* = \begin{bmatrix} 0 & -g & 0 & 0 \\ g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{5.22}$$

$$\boldsymbol{K}_2 = \begin{bmatrix} {}_{H}\boldsymbol{R}_{T} & {}_{H}\boldsymbol{R}_{I}^* \end{bmatrix}, \qquad \text{with} \qquad {}_{H}\boldsymbol{R}_{I}^* = {}_{H}\boldsymbol{R}_{T} \begin{bmatrix} \text{c}\psi_{IT} & \text{s}\psi_{IT} \\ -\text{s}\psi_{IT} & \text{c}\psi_{IT} \\ 0 & 0 \end{bmatrix}, \tag{5.23}$$

$$\boldsymbol{K}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{5.24}$$

The specific forces and angular velocity in equation (5.21) are fed to the vestibular model (5.11). In order to obtain a single state space model, the kinematic model (5.7), the vestibular model (5.11) and equation (5.21) are combined in the form

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{x}}_{semi} \\ \dot{\boldsymbol{x}}_{oto} \\ \dot{\boldsymbol{d}}_S \end{bmatrix}}_{\dot{\boldsymbol{x}}_S} = \underbrace{\begin{bmatrix} \boldsymbol{A}_{semi} & \boldsymbol{0}_{6x6} & \boldsymbol{0}_{6x14} \\ \boldsymbol{0}_{6x6} & \boldsymbol{A}_{oto} & \boldsymbol{B}_{oto}\,\boldsymbol{K}_1 \\ \boldsymbol{0}_{14x6} & \boldsymbol{0}_{14x6} & \boldsymbol{A}_S \end{bmatrix}}_{\boldsymbol{A}_{pred}} \underbrace{\begin{bmatrix} \boldsymbol{x}_{semi} \\ \boldsymbol{x}_{oto} \\ \boldsymbol{d}_S \end{bmatrix}}_{\boldsymbol{x}_S} + \underbrace{\left[\begin{array}{cc} \boldsymbol{B}_{semi\,H}\boldsymbol{J}\,\boldsymbol{K}_3 & \boldsymbol{0}_{6x5} \\ \boldsymbol{0}_{6x4} & \boldsymbol{B}_{oto}\,\boldsymbol{K}_2 \\ \multicolumn{2}{c}{\boldsymbol{B}_S} \end{array}\right]}_{\boldsymbol{B}_{pred}} \boldsymbol{u}_S \tag{5.25a}$$

$$\underbrace{\begin{bmatrix} \hat{\boldsymbol{y}}_S \\ \boldsymbol{d}_S \end{bmatrix}}_{\boldsymbol{Y}_S} = \underbrace{\begin{bmatrix} \boldsymbol{C}_{semi} & \boldsymbol{0}_{3x6} & \boldsymbol{0}_{3x14} \\ \boldsymbol{0}_{3x6} & \boldsymbol{C}_{oto} & \boldsymbol{0}_{3x14} \\ \boldsymbol{0}_{14x6} & \boldsymbol{0}_{14x6} & \boldsymbol{I}_{14x14} \end{bmatrix}}_{\boldsymbol{C}_{pred}} \underbrace{\begin{bmatrix} \boldsymbol{x}_{semi} \\ \boldsymbol{x}_{oto} \\ \boldsymbol{d}_S \end{bmatrix}}_{\boldsymbol{x}_S} + \underbrace{\left[\begin{array}{cc} \boldsymbol{D}_{semi\,H}\boldsymbol{J}\,\boldsymbol{K}_3 & \boldsymbol{0}_{3x5} \\ \boldsymbol{0}_{3x4} & \boldsymbol{D}_{oto}\,\boldsymbol{K}_2 \\ \multicolumn{2}{c}{\boldsymbol{0}_{14x9}} \end{array}\right]}_{\boldsymbol{D}_{pred}} \boldsymbol{u}_S \tag{5.25b}$$

The linear MPC-scheme requires a linearized, discrete-time state-space model in the form

$$\boldsymbol{x}_S[k+1] = \boldsymbol{A}_d\,\boldsymbol{x}_S[k] + \boldsymbol{B}_d\,\boldsymbol{u}_S[k] \tag{5.26}$$

$$\boldsymbol{Y}_S[k] = \boldsymbol{C}_d\,\boldsymbol{x}_S[k] + \boldsymbol{D}_d\,\boldsymbol{u}_S[k] \tag{5.27}$$

with

the state vector:	$\boldsymbol{x}_S \in \mathbb{R}^{n_x}$,	the state matrix:	$\boldsymbol{A}_d \in \mathbb{R}^{n_x \times n_x}$,
the input vector:	$\boldsymbol{u}_S \in \mathbb{R}^{n_u}$,	the input matrix:	$\boldsymbol{B}_d \in \mathbb{R}^{n_x \times n_u}$,
the output vector:	$\boldsymbol{Y}_S \in \mathbb{R}^{n_y}$,	the output matrix:	$\boldsymbol{C}_d \in \mathbb{R}^{n_y \times n_x}$,
		the feedthrough matrix:	$\boldsymbol{D}_d \in \mathbb{R}^{n_y \times n_u}$

and where k counts the discrete time steps. For a sample time T_{samp} and for piecewise constant inputs $\boldsymbol{u}_S$, the system from equation (5.25) can be solved analytically with the following transformation [106]

$$\begin{aligned} \boldsymbol{x}_S(t) &= e^{\boldsymbol{A}_{pred}t}\boldsymbol{x}_S(0) + \int_0^t e^{\boldsymbol{A}_{pred}(t-\tau)}\boldsymbol{B}_{pred}\boldsymbol{u}_S \mathrm{d}\tau \\ &= e^{\boldsymbol{A}_{pred}t}\boldsymbol{x}_S(0) + e^{\boldsymbol{A}_{pred}t}(-\boldsymbol{A}_{pred}^{-1})\left(e^{-\boldsymbol{A}_{pred}\tau}\Big|_0^t\right)\boldsymbol{B}_{pred}\boldsymbol{u}_S \qquad \text{for } \ \boldsymbol{u}_S = \text{const.} \\ &= e^{\boldsymbol{A}_{pred}t}\boldsymbol{x}_S(0) + \boldsymbol{A}_{pred}^{-1}\left(e^{\boldsymbol{A}_{pred}t} - \boldsymbol{I}_{n_x}\right)\boldsymbol{B}_{pred}\boldsymbol{u}_S \end{aligned} \tag{5.28}$$

Discretizing and applying equation (5.28) with the sample time T_{samp} yields

$$\boldsymbol{x}_S[k+1] = e^{\boldsymbol{A}_{pred}T_{samp}}\,\boldsymbol{x}_S[k] + \boldsymbol{A}_{pred}^{-1}\left(e^{\boldsymbol{A}_{pred}T_{samp}} - \boldsymbol{I}_{n_x}\right)\boldsymbol{B}_{pred}\,\boldsymbol{u}_S[k]. \tag{5.29}$$

The exponential term $e^{\boldsymbol{A}_{pred}T_{samp}}$ is a convergent power series [117]

$$e^{\boldsymbol{A}_{pred}T_{samp}} = \boldsymbol{I} + \boldsymbol{A}_{pred}\,T_{samp} + \frac{\boldsymbol{A}_{pred}^2\,T_{samp}^2}{2!} + \ldots \tag{5.30}$$

The Taylor-series can be linearized for small sample times T_{samp} with

$$e^{\boldsymbol{A}_{pred}T_{samp}} \approx \boldsymbol{I} + \boldsymbol{A}_{pred}\,T_{samp}. \tag{5.31}$$

Thus, the system 5.25 can be discretized with

$$\boldsymbol{x}_S[k+1] = \underbrace{(\boldsymbol{I} + \boldsymbol{A}_{pred}\, T_{samp})}_{\boldsymbol{A}_d}\, \boldsymbol{x}_S[k] + \underbrace{T_{samp}\, \boldsymbol{B}_{pred}}_{\boldsymbol{B}_d}\, \boldsymbol{u}_S[k] \tag{5.32a}$$

$$\boldsymbol{Y}_S[k] = \boldsymbol{C}_d\, \boldsymbol{x}_S[k] + \boldsymbol{D}_d\, \boldsymbol{u}_S[k], \qquad \text{with} \quad \boldsymbol{C}_d = \boldsymbol{C}_{pred}, \quad \boldsymbol{D}_d = \boldsymbol{D}_{pred} \tag{5.32b}$$

which corresponds to an explicit Euler discretization with conditional stability properties. A linear equation in the form

$$\dot{\phi}(t) = \lambda \phi(t) \tag{5.33}$$

exhibits the stability criteria [30]

$$|1 + \lambda T_{samp}| \leq 1. \tag{5.34}$$

Applying equation (5.34) to the system introduced in equation (5.25), λ complies with the eigenvalues λ_i of the system matrix $\boldsymbol{A}_{pred}$ for $i \in \{1 \ldots n_x\}$. In order to estimate a lower and upper limit for T_{samp}, equation (5.34) is solved to

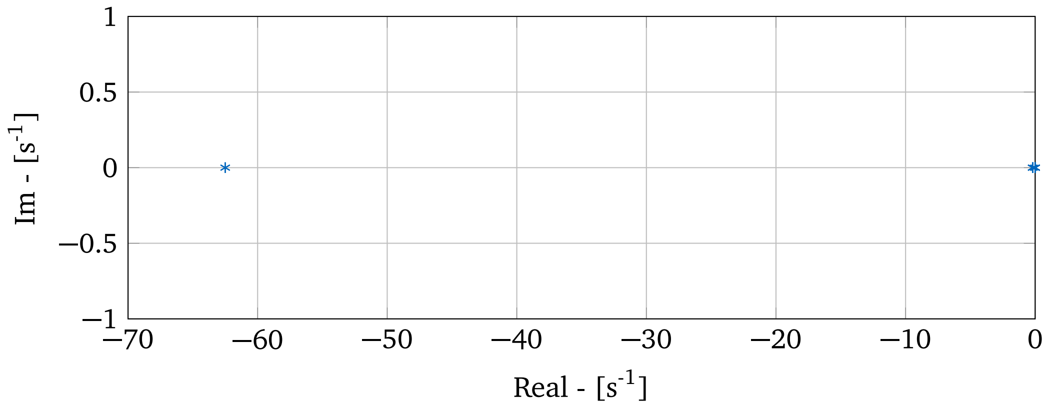


Figure 5.4: Eigenvalues of the system matrix $\boldsymbol{A}_{pred}$. The eigenvalue with the largest real portion in terms of absolute value has the value of $\lambda_{min} = -62.5$.

$$T_{samp} \leq \frac{2}{|\lambda_i|} \qquad \forall i \in \{1 \ldots n_x\}. \tag{5.35}$$

Since all eigenvalues λ_i are located on the left complex half-plane including the origin and only exhibit a real part, the minimum and maximum sample times yield

$$T_{samp,min} = 0\,\text{s} \qquad T_{samp,max} = 0.032\,\text{s}\,. \tag{5.36}$$

The discretized model from equation (5.32) is included to an MPC algorithm. Figure 5.5 schematically shows the principal idea of an MPC method for a single-input, single-output (SISO) system. The index k counts the discrete MPC time steps and $i \in [1 \ldots H_p]$, $H_p < \infty$, counts the discrete steps of the time horizon H_p. The discretized internal model from equation (5.32) is used to predict the future behaviour of the system, starting at the current time k, over the prediction horizon $i = 1 \ldots H_p$. The predicted model's output $\boldsymbol{Y}_S[k,i]$ depends on the input trajectory $\boldsymbol{u}_S[k,i]$. The second argument i of the discrete notation indicates that the variable depends on the conditions at the time k. The solution of a numerical optimization problem yields the predicted input $\boldsymbol{u}_S[k,i]$. The system's output $\boldsymbol{Y}_S[k,i]$ is gained by feeding the prediction model with $\boldsymbol{u}_S[k,i]$. The output $\hat{y}_S[k,i]$ (part of $\boldsymbol{Y}_S$) tracks the reference

trajectory $\hat{\boldsymbol{y}}_V[k,i]$ over the future prediction horizon H_p.
For a calculated input trajectory $\boldsymbol{u}_S[k,i]$, usually only the first input is send a command to the simulator $\boldsymbol{u}_S[k,1]$. The calculation process of the output calculation, the reference generation and the input prediction is repeated at the time step $k+1$ for a shifted time horizon H_p. As the length of the time horizon H_p stays constant and slides along at each time step k, the described MPC scheme is called *receding horizon strategy*. Another possibility is to apply several discrete inputs $[\boldsymbol{u}_S[k,1],\ldots,\boldsymbol{u}_S[k,i^*]]$ of the input trajectory $\boldsymbol{u}_S[k,i]$ and to shift the horizon by the number of applied inputs i^*+1.[32, 106]

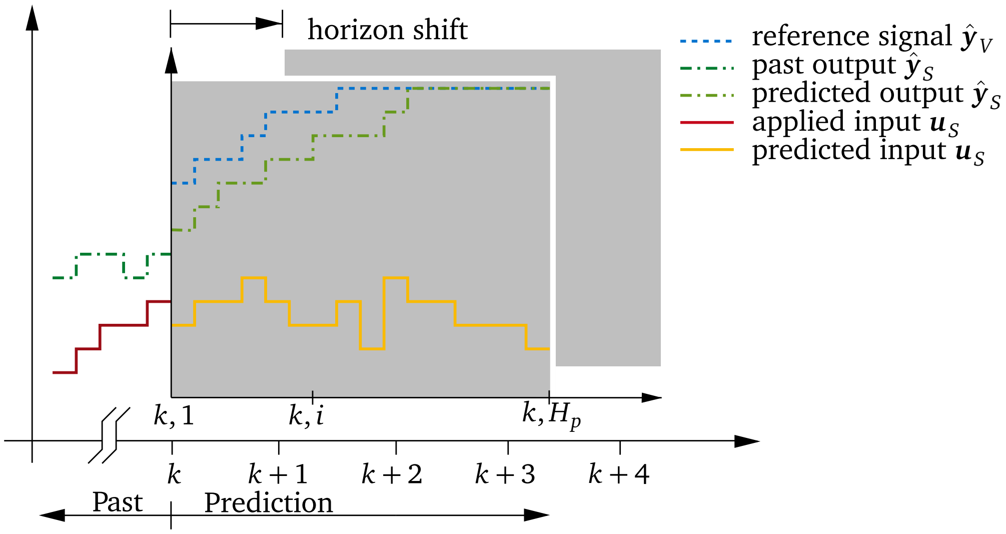

horizon shift
reference signal $\hat{\boldsymbol{y}}_V$
past output $\hat{\boldsymbol{y}}_S$
predicted output $\hat{\boldsymbol{y}}_S$
applied input $\boldsymbol{u}_S$
predicted input $\boldsymbol{u}_S$
$k,1$
k,i
k,H_p
k
$k+1$
$k+2$
$k+3$
$k+4$
Past
Prediction

Figure 5.5: Model predictive control scheme by using a receding horizon strategy.

5.4 Objective Function

The applied objective function J

$$\begin{aligned} J[k] \quad = \quad & \sum_{i=1}^{H_p} \| \hat{\boldsymbol{y}}_S[k,i] - \hat{\boldsymbol{y}}_V[k,i] \|_{\boldsymbol{Q}}^2 + \sum_{i=1}^{H_p} \| \boldsymbol{d}_S[k,i] \|_{\boldsymbol{R}}^2 + \\ & \sum_{i=1}^{H_u} \| \boldsymbol{u}_S[k,i] \|_{\boldsymbol{S}}^2 + \sum_{i=1}^{H_u} \| \Delta\boldsymbol{u}_S[k,i] \|_{\boldsymbol{T}}^2 \end{aligned} \tag{5.37}$$

penalizes deviations between perceived motions in a virtual vehicle $\hat{\boldsymbol{y}}_V$ and in a simulator $\hat{\boldsymbol{y}}_S$ over the time horizon H_p. For an open-loop estimation of the MCA, the driving manoeuvre is known a priori and therefore the inputs $\boldsymbol{u}_V$ can be used over the prediction horizon H_p to calculate the reference trajectory $\hat{\boldsymbol{y}}_V$. In case of a closed-loop simulation with no available prediction, one possibility is to keep $\boldsymbol{u}_V[k,i] = \boldsymbol{u}_V[k,1]$ for $i = 1 \ldots H_u$.
The terms $\boldsymbol{d}_S$, $\boldsymbol{u}_S$ and $\Delta\boldsymbol{u}_S$ penalize deviations from the neutral state to prevent workspace overshoots on position, velocity and acceleration level. Deviations between two successive inputs $\boldsymbol{u}_S$ in the form

$$\begin{aligned} \Delta\boldsymbol{u}_S[k,i] = \boldsymbol{u}_S[k,i] - \boldsymbol{u}_S[k,i-1] \qquad & \text{for } i = 1\ldots H_u, \\ & \text{with } \boldsymbol{u}_S[k,0] = \boldsymbol{u}_S[k-1,1] \ \text{ for } \ i = 1 \end{aligned} \tag{5.38}$$

are included to the objective function in order to penalize changes between the predicted input $\boldsymbol{u}_S[k,i]$ to the previous input $\boldsymbol{u}_S[k,i-1]$. Thus, changes in translational acceleration and angular velocity of two sequential MPC-steps are considered in the objective function in order to gain smoother responses for the input $\boldsymbol{u}_S$. In the following, if the index $i=0$ in $[k,i]$, the values of the previous MPC-step $[k-1,1]$ are taken. Figure 5.6 illustrates the partitioning of the prediction horizon H_p into three parts [106]. H_u is the control horizon for the input $\boldsymbol{u}_S$ with the assumption $H_u \leq H_p$ and $\Delta\boldsymbol{u}_S[k,i]=0$ for $i \geq H_u$, resulting in $\boldsymbol{u}_S[k,i]=\boldsymbol{u}_S[k,H_u]\ \forall i \geq H_u$. This approach leads to a reduction in the number of optimization parameters, resulting in a lower computation time. However, the performance of the MPC is also reduced as $\boldsymbol{u}_S[k,i]$ stays constant for $i \in [H_u, H_p]$. Still, keeping $\boldsymbol{u}_S[k,i]$ constant over $[H_u, H_p]$ increases the stability of the MPC since a higher weighting is implicitly put on the last optimization parameter $\boldsymbol{u}_S[k,H_u]$ in relation to the other optimization parameters. This yields that the workspace motions, which are part of the objective function, are taken into account for a longer time horizon. The constraint horizon H_c assumes the constraints' adherence for $H_c \leq H_p$. As the constraints of the optimization scheme are defined in actuator space and the actuator states dependent on $\boldsymbol{u}_S$ and $\boldsymbol{d}_S$ of the optimization scheme, both signals need to be bounded. Decreasing the number of constraints results in a complexity reduction of the optimization problem to reduce the computation time.

Defining the norms in equation (5.37), $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{T}$ are diagonal weighting matrices. Elements of the $\boldsymbol{Q}$ matrix are estimated by

$$Q_{i,i} = \frac{\kappa_{i,i}}{(\tau_{semi}/\tau_{oto})^2} \tag{5.39}$$

in order to normalize the various optimization parameters to the different units as well as to the minimum perceived errors. It is assumed that the resolution of the vestibular system is as large as the perception threshold for accelerations and angular velocities. Herein, the rotational and translational perception threshold values of [141] are used ($\tau_{semi} = 3°/\text{s}$ and $\tau_{oto} = 0.17\text{m/s}^2$). The factors $\kappa_{i,j}$ additionally weight the optimization parameters relative to each other. Elements of the $\boldsymbol{R}$ and $\boldsymbol{S}$ matrices are calculated analogously by

$$\left.\begin{matrix} R_{i,i} \\ S_{i,i} \end{matrix}\right\} = \frac{\xi_{i,i}}{(\boldsymbol{w}_{S,max}/\boldsymbol{w}_{S,min})^2} \tag{5.40}$$

to normalize the different units and to correlate them to the approximated maximum/ minimum workspace values ($\boldsymbol{w}_{S,max}/\boldsymbol{w}_{S,min}$). Herein, the workspace capabilities listed in Table 2.2 are used. The parameters $\kappa_{i,i}$ and $\xi_{i,i}$ can be set to 1 as an initial assumption, whereas $\kappa_{i,i}$ weight the perceived translational accelerations and perceived rotational velocities to each other and $\xi_{i,i}$ weight the translational and rotational motions of tripod and hexapod. The hand-tuned weights are shown in Appendix D in Table D.1.

5.5 Prediction

As the objective function in equation (5.37) includes $\Delta\boldsymbol{u}_S$, which is defined in equation (5.38), the state-space vector is augmented by

$$\xi[k,i] = \begin{bmatrix} \boldsymbol{x}_S[k,i] \\ \boldsymbol{u}_S[k,i-1] \end{bmatrix}, \qquad (\text{with } \boldsymbol{u}_S[k,0] = \boldsymbol{u}_S[k-1,1] \ \text{ for } \ i=1) \tag{5.41}$$

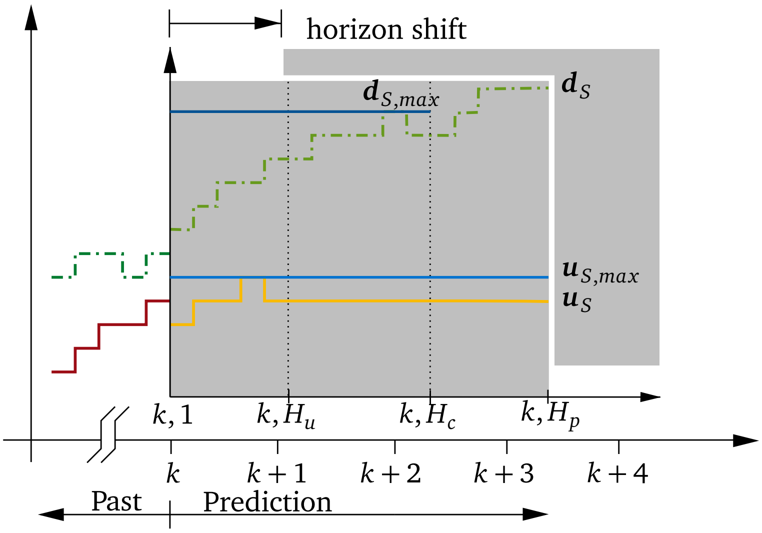


Figure 5.6: Division of the prediction horizon H_p into the control horizon H_u and the constraint horizon H_c.

at a time step $k \in \mathbb{N}$ over $i = 1 \ldots H_p$, leading to the state-space representation

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_S[k,i+1] \\ \boldsymbol{u}_S[k,i] \end{bmatrix}}_{\xi[k,i+1]} = \underbrace{\begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{B}_d \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{A}_{aug}} \underbrace{\begin{bmatrix} \boldsymbol{x}_S[k,i] \\ \boldsymbol{u}_S[k,i-1] \end{bmatrix}}_{\xi[k,i]} + \underbrace{\begin{bmatrix} \boldsymbol{B}_d \\ \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{B}_{aug}} \Delta \boldsymbol{u}_S[k,i] \tag{5.42a}$$

$$\boldsymbol{Y}_S[k,i] = \underbrace{\begin{bmatrix} \boldsymbol{C}_d & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{C}_{aug}} \underbrace{\begin{bmatrix} \boldsymbol{x}_S[k,i] \\ \boldsymbol{u}_S[k,i-1] \end{bmatrix}}_{\xi[k,i]} + \underbrace{\begin{bmatrix} \boldsymbol{D}_d \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{D}_{aug}} \Delta \boldsymbol{u}_S[k,i]. \tag{5.42b}$$

An explicit formulation predicts the future outputs $\boldsymbol{Y}_S[k,i]$ of the system over the horizon $i = 1 \ldots H_p$ under consideration of the current state $\boldsymbol{x}_S[k]$, the predicted input $\boldsymbol{u}_S[k,i]$ for $i = 1 \ldots H_u$, the previous input $\boldsymbol{u}[k,i-1]$ and the input changes $\Delta \boldsymbol{u}_S[k,i]$, $i = 1 \ldots H_u$. The following estimations are based on the conditions at the time step k over the time horizon $i = 1 \ldots H_p$. Applying equation (5.42), the following time steps yield

$$\begin{aligned} \boldsymbol{Y}_S[k,1] &= \boldsymbol{C}_{aug}\xi[k,1] + \boldsymbol{D}_{aug}\Delta \boldsymbol{u}_S[k,1] \\ \boldsymbol{Y}_S[k,2] &= \boldsymbol{C}_{aug}\xi[k,2] + \boldsymbol{D}_{aug}\Delta \boldsymbol{u}_S[k,2] \\ &= \boldsymbol{C}_{aug}\left(\boldsymbol{A}_{aug}\xi[k,1] + \boldsymbol{B}_{aug}\Delta \boldsymbol{u}_S[k,1]\right) + \boldsymbol{D}_{aug}\Delta \boldsymbol{u}_S[k,2] \\ \boldsymbol{Y}_S[k,3] &= \boldsymbol{C}_{aug}\xi[k,3] + \boldsymbol{D}_{aug}\Delta \boldsymbol{u}_S[k,3] \\ &= \boldsymbol{C}_{aug}\left(\boldsymbol{A}_{aug}^2\xi[k,1] + \boldsymbol{A}_{aug}\boldsymbol{B}_{aug}\Delta \boldsymbol{u}_S[k,1] + \boldsymbol{B}_{aug}\Delta \boldsymbol{u}_S[k,2]\right) + \boldsymbol{D}_{aug}\Delta \boldsymbol{u}_S[k,3] \end{aligned} \tag{5.43}$$

The procedure for calculating the remaining outputs over H_p can be performed in a similar manner. The predicted outputs $\boldsymbol{Y}_S$ over the prediction horizon H_p can be expressed in matrix vector form in equation (5.44). The subscript *aug* of the matrices $\boldsymbol{A}_{aug}$, $\boldsymbol{B}_{aug}$, $\boldsymbol{C}_{aug}$, $\boldsymbol{D}_{aug}$ is neglected in the following for the sake of simplicity

$$\underbrace{\begin{bmatrix} \boldsymbol{Y}_S[k,1] \\ \boldsymbol{Y}_S[k,2] \\ \boldsymbol{Y}_S[k,3] \\ \vdots \\ \boldsymbol{Y}_S[k,H_u] \\ \boldsymbol{Y}_S[k,H_u+1] \\ \vdots \\ \boldsymbol{Y}_S[k,H_p] \end{bmatrix}}_{\boldsymbol{Y}_S[k,:]} = \underbrace{\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{H_u-1} \\ \boldsymbol{CA}^{H_u} \\ \vdots \\ \boldsymbol{CA}^{H_p-1} \end{bmatrix}}_{\boldsymbol{\Psi}[k]} \xi[k,1]+ \tag{5.44}$$

$$\underbrace{\begin{bmatrix} \boldsymbol{D} & \boldsymbol{0}_{n_y\times n_u} & \cdots & \cdots & \boldsymbol{0}_{n_y\times n_u} \\ \boldsymbol{CB} & \boldsymbol{D} & \boldsymbol{0}_{n_y\times n_u} & \cdots & \boldsymbol{0}_{n_y\times n_u} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \boldsymbol{D} & \cdots & \boldsymbol{0}_{n_y\times n_u} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{CA}^{H_u-2}\boldsymbol{B} & \boldsymbol{CA}^{H_u-3}\boldsymbol{B} & \cdots & \boldsymbol{CB} & \boldsymbol{D} \\ \boldsymbol{CA}^{H_u-1}\boldsymbol{B} & \boldsymbol{CA}^{H_u-2}\boldsymbol{B} & \cdots & \boldsymbol{CAB} & \boldsymbol{D} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{CA}^{H_p-2}\boldsymbol{B} & \boldsymbol{CA}^{H_p-3}\boldsymbol{B} & \cdots & \boldsymbol{CA}^{H_p-H_u}\boldsymbol{B} & \boldsymbol{D} \end{bmatrix}}_{\boldsymbol{\theta}[k]} \underbrace{\begin{bmatrix} \Delta\boldsymbol{u}_S[k,1] \\ \Delta\boldsymbol{u}_S[k,2] \\ \Delta\boldsymbol{u}_S[k,3] \\ \vdots \\ \Delta\boldsymbol{u}_S[k,H_u] \end{bmatrix}}_{\Delta\boldsymbol{U}_S[k,:]},$$

where $\boldsymbol{Y}_S[k,:]$ and $\Delta\boldsymbol{U}_S[k,:]$ describe the complete column at the time instant k. Similar to equation (5.38), the input vector is

$$\boldsymbol{U}_S[k,:] = \underbrace{\boldsymbol{\kappa}_P\ \boldsymbol{u}_S[k-1,1]}_{\boldsymbol{U}_{S,P}} + \boldsymbol{\kappa}\ \Delta\boldsymbol{U}_S[k,:] \qquad \text{with} \qquad \boldsymbol{\kappa}_P = \boldsymbol{1}_{H_u\times 1}\otimes \boldsymbol{I}_{n_u\times n_u}, \tag{5.45}$$

$$\boldsymbol{\kappa} = \boldsymbol{L}_{H_u\times H_u}\otimes \boldsymbol{I}_{n_u\times n_u}, \tag{5.46}$$

$$\boldsymbol{U}_{S,P} = \boldsymbol{\kappa}_P\ \boldsymbol{u}_S[k-1,1], \tag{5.47}$$

where $\boldsymbol{L}_{H_u\times H_u}$ is a lower triangle-matrix, filled with ones, $\boldsymbol{1}_{H_u\times 1}$ is vector, filled with ones, and $\otimes$ is the Kronecker product.

The objective function in equation (5.37) can be further reduced. The free response of the system 5.44 is defined to

$$\boldsymbol{Y}_{S,f}[k] = \boldsymbol{\Psi}[k]\xi[k,1] \qquad \text{with } \Delta\boldsymbol{U}_S[k,:] = \boldsymbol{0}. \tag{5.48}$$

The deviation $\boldsymbol{E}$ between the free response $\boldsymbol{Y}_{S,f}$ and the reference trajectory $\boldsymbol{Y}_V$ is defined to

$$\boldsymbol{E} = \boldsymbol{Y}_{S,f} - \boldsymbol{Y}_V \tag{5.49}$$

where $\boldsymbol{Y}_V$ includes the reference values $\hat{\boldsymbol{y}}_V[k,i]$ for $i\ldots H_p$. The indication of k and i is partly disregarded in the following transformations for the sake of clarity. The first term in the objective function from equation (5.37) can be rewritten in the form

$$\boldsymbol{Y}_S - \boldsymbol{Y}_V = \boldsymbol{\Psi}\xi + \boldsymbol{\theta}\Delta\boldsymbol{U}_S - \boldsymbol{Y}_V = \boldsymbol{E} + \boldsymbol{\theta}\Delta\boldsymbol{U}_S \tag{5.50}$$

Under consideration of equations (5.45) and (5.50), the objective function from equation (5.37) can be expressed by

$$
\begin{aligned}
J = &(\boldsymbol{E} + \boldsymbol{\theta}\Delta \boldsymbol{U}_S)^T \boldsymbol{Q}(\boldsymbol{E} + \boldsymbol{\theta}\Delta \boldsymbol{U}_S) + (\boldsymbol{U}_{S,P} + \boldsymbol{\kappa}\Delta \boldsymbol{U}_S)^T \boldsymbol{S}(\boldsymbol{U}_{S,P} + \boldsymbol{\kappa}\Delta \boldsymbol{U}_S) \qquad (5.51)\\
&+ \Delta \boldsymbol{U}_S^T \boldsymbol{T} \Delta \boldsymbol{U}_S \\
= &\Delta \boldsymbol{U}_S^T(\boldsymbol{\theta}^T \boldsymbol{Q}\boldsymbol{\theta} + \boldsymbol{\kappa}^T \boldsymbol{S}\boldsymbol{\kappa} + \boldsymbol{T})\Delta \boldsymbol{U}_S + 2\Delta \boldsymbol{U}_S^T(\boldsymbol{\theta}^T \boldsymbol{Q}\boldsymbol{E} + \boldsymbol{\kappa}^T \boldsymbol{S}\boldsymbol{U}_{S,P}) \qquad (5.52)\\
&+ \cancel{\boldsymbol{E}^T \boldsymbol{Q}\boldsymbol{E} + \boldsymbol{U}_{S,P}^T \boldsymbol{S}\boldsymbol{U}_S}
\end{aligned}
$$

The crossed out terms are constant at every discrete time instance over $i = 1 \dots H_p$, and hence cannot be influenced by $\Delta \boldsymbol{U}_S^T$. Combining the terms in equation (5.52), the reduced form of the objective function is

$$
J = \frac{1}{2}\Delta \boldsymbol{U}_S^T \boldsymbol{H}\Delta \boldsymbol{U}_S + \Delta \boldsymbol{U}_S^T \boldsymbol{g} \qquad (5.53)
$$

with the Hesse matrix $\boldsymbol{H}$ and the gradient $\boldsymbol{g}$, respectively defined to

$$
\begin{aligned}
\boldsymbol{H} &= 2(\boldsymbol{\theta}^T \boldsymbol{Q}\boldsymbol{\theta} + \boldsymbol{\kappa}^T \boldsymbol{S}\boldsymbol{\kappa} + \boldsymbol{T}) \qquad (5.54)\\
\boldsymbol{g} &= 2(\boldsymbol{\theta}^T \boldsymbol{Q}\boldsymbol{E} + \boldsymbol{\kappa}^T \boldsymbol{S}\boldsymbol{U}_{S,P}). \qquad (5.55)
\end{aligned}
$$

5.6 Linearized Actuator Constraints

The tripod's and hexapod's actuators exhibit a minimum and a maximum length $q_{n,min}$, $q_{n,max}$ for $n = t, h$ by construction. Actuators' velocities and accelerations are also limited for both systems. These mechanical restrictions are included into the optimization problem. The equations for positions, velocities and accelerations of the actuators are nonlinear because of the kinematic structure of a hexapod/tripod [115]. In this Section, a linearization approach is used to reduce the computation time for each optimization problem. It is the result of collaboration with Christoph Lankes [104].

In Chapter 2.4 , Figure 2.3 depicts the parallel kinematic chain of a parallel robot. For the considered motion system, consisting of tripod and hexapod, the hexapod's kinematic chain includes a CRP, for which reason equation (2.19) is applied. For the tripod, the kinematic chain from equation (2.20) is used. Equation (2.22) calculates the actuator lengths for hexapod $q_{h,i}$, $i = 1 \dots 6$ and tripod $q_{t,i}$, $i = 1 \dots 3$. The workspace configuration $\boldsymbol{w}_S$ is defined for the tripod as

$$
\boldsymbol{w}_{S,t} = [{}_I r_{IT,x} \;\; {}_I r_{IT,y} \;\; \psi_{IT}]^T \qquad (5.56)
$$

and for the hexapod as

$$
\boldsymbol{w}_{S,h} = [{}_T r_{TH,x} \;\; {}_T r_{TH,y} \;\; {}_T r_{TH,z} \;\; \varphi_{TH} \;\; \theta_{TH} \;\; \psi_{TH}]^T. \qquad (5.57)
$$

Starting from the differential equality

$$
\dot{q}_{n,l} = \frac{1}{2q_{n,l}}\frac{\mathrm{d}q_{n,l}^2}{\mathrm{d}t} = \boldsymbol{J}_{n,l}(\boldsymbol{w}_{S,n})\,\dot{\boldsymbol{w}}_{S,n} \qquad \text{for} \begin{cases} n = t, h \\ l = \text{actuator number} \end{cases} \qquad (5.58)
$$

and considering that the actuator lengths are related to the workspace of the system through equation (2.22), the change of actuator length in equation (5.58) can also be written in terms of workspace velocities: applying the chain rule of derivation and considering the matrix/vector representation from equation (2.23). Herein, $\boldsymbol{J}_{n,l}(\boldsymbol{w}_{S,n})$ defines the inverse Jacobian matrix for $n = t, h$ and the actuator number l. The computation of $\boldsymbol{J}_{n,l}$ depends on the current workspace configuration $\boldsymbol{w}_{S,n}$. These nonlinear terms are kept constant over the time horizon H_c. Since the workspace variables at the time instant k are not known, the previous workspace variables $\boldsymbol{w}_{S,n}[k-1,1]$ are used. The method confers a linear relation between workspace and actuator space variables over the time horizon which can be transformed to

$$\begin{aligned} \dot{q}_{n,l}[k,i] &= \boldsymbol{J}_{n,l}(\boldsymbol{w}_{S,n}[k-1,1])\,\dot{\boldsymbol{w}}_{S,n}[k,i] \\ &= \boldsymbol{J}_{n,l,\boldsymbol{d}_S}\,\boldsymbol{d}_S[k,i] + \boldsymbol{J}_{n,l,\boldsymbol{u}_S}\,\boldsymbol{u}_S[k,i] \qquad \text{for} \qquad i = 1 \dots H_c \end{aligned} \tag{5.59}$$

where the matrices $\boldsymbol{J}_{n,l,\boldsymbol{d}_S}$ and $\boldsymbol{J}_{n,l,\boldsymbol{u}_S}$ include the partial derivatives from the Jacobian matrix $\boldsymbol{J}_{n,l}(\boldsymbol{w}_{S,n})$ in equation (5.58). The nonlinear terms of $\boldsymbol{J}_{n,l}(\boldsymbol{w}_{S,n})$ are derived in equation (2.27). The structure of matrices $\boldsymbol{J}_{n,l,\boldsymbol{d}_S}$ and $\boldsymbol{J}_{n,l,\boldsymbol{u}_S}$ is chosen such that the vector $\boldsymbol{d}_S$ (cf. equation (5.8)) and the input vector $\boldsymbol{u}_S$ (cf. equation (5.3)) from the prediction model in equation (5.25) can be used. The split of $\dot{\boldsymbol{w}}_{S,n}$ to $\boldsymbol{d}_S$ / $\boldsymbol{u}_S$ yields, for the tripod,

$$\dot{\boldsymbol{w}}_{S,t} = [\underbrace{{}_I\dot{r}_{IT,x}\;{}_I\dot{r}_{IT,y}}_{\text{part of } \boldsymbol{d}_S}\quad \underbrace{\dot{\psi}_{IT}}_{\text{part of } \boldsymbol{u}_S}\;]^T \tag{5.60}$$

and, for the hexapod,

$$\dot{\boldsymbol{w}}_{S,h} = [\underbrace{{}_T\dot{r}_{TH,x}\;{}_T\dot{r}_{TH,y}\;{}_T\dot{r}_{TH,z}}_{\text{part of } \boldsymbol{d}_S}\;\underbrace{\dot{\varphi}_{TH}\;\dot{\theta}_{TH}\;\dot{\psi}_{TH}}_{\text{part of } \boldsymbol{u}_S}]^T. \tag{5.61}$$

Numerically integrating equation (5.59) yields

$$\begin{aligned} q_{n,l}[k,i+1] &= q_{n,l}[k,i] + T_{samp}\,\dot{q}_{n,l}[k,i] \\ &= q_{n,l}[k,i] + \boldsymbol{A}^*_{act}\xi[k,1] + \boldsymbol{B}^*_{act}\Delta\boldsymbol{u}_S[k,i] \qquad \text{for } i = 1 \dots H_c \end{aligned} \tag{5.62}$$

where the matrices $\boldsymbol{J}_{n,l,\boldsymbol{d}_S}$ and $\boldsymbol{J}_{n,l,\boldsymbol{u}_S}$ are augmented to $\boldsymbol{A}^*_{act}$ and $\boldsymbol{B}^*_{act}$, respectively, using the augmentation of the model's input from $\boldsymbol{u}_S$ to $\Delta\boldsymbol{u}_S$ (cf. equation (5.42)). The vector ξ includes $\boldsymbol{x}_s$ (cf. equation (5.41)) and $\boldsymbol{x}_s$ includes $\boldsymbol{d}_s$ (cf. equation (5.25)). All actuator lengths $q_{n,l}$ from equation (5.62) for $n = t, h$ and the number of actuators l are included in

$$\boldsymbol{q}_S[k,i+1] = \boldsymbol{q}_S[k,i] + \boldsymbol{A}_{act}\xi[k,1] + \boldsymbol{B}_{act}\Delta\boldsymbol{u}_S[k,i] \qquad \text{for } i = 1 \dots H_c \tag{5.63}$$

where the matrices $\boldsymbol{A}^*_{act}$ and $\boldsymbol{B}^*_{act}$ are extended to $\boldsymbol{A}_{act}$ and $\boldsymbol{B}_{act}$ (considering all actuators). The equation (5.63) for the actuator lengths $\boldsymbol{q}_S$ can be expressed by a vector/matrix description over the constraint horizon $H_c > H_u$

$$\boldsymbol{q}_S[k,:] = \begin{bmatrix} \boldsymbol{q}_S[k,1] \\ \boldsymbol{q}_S[k,2] \\ \vdots \\ \boldsymbol{q}_S[k,H_u] \\ \boldsymbol{q}_S[k,H_u+1] \\ \vdots \\ \boldsymbol{q}_S[k,H_c] \end{bmatrix} = \boldsymbol{q}_{S,0} + \boldsymbol{\Psi}_{act}\,\xi[k,1] + \boldsymbol{\theta}_{act}\Delta\boldsymbol{U}_S^T[k,:] \tag{5.64}$$

where the state $\boldsymbol{q}_S[k,1]$ is used over the complete horizon H_c in the form

$$\boldsymbol{q}_{S,0} = (\,\mathbf{1}_{H_c\times 1} \otimes \boldsymbol{I}_{n_g\times n_g}\,)\,\boldsymbol{q}_S[k,1] \tag{5.65}$$

with the number of actuators n_g. The set-up of the matrices $\boldsymbol{\Psi}_{act}$ and $\boldsymbol{\theta}_{act}$ follows a similar procedure as for the matrices $\boldsymbol{\Psi}$ and $\boldsymbol{\theta}$ in equation (5.44). Herein, the prediction model from equation (5.42) is used to calculate the input $\boldsymbol{u}_S[k,i]$ and $\boldsymbol{d}_S[k,i]$ over the time horizon $i=1\dots H_c$. Further results can be found in [104].
To calculate the actuator velocities $\dot{\boldsymbol{q}}_S$ over the time horizon H_c, equation (5.59) can also be transformed to an explicit vector/matrix representation

$$\dot{\boldsymbol{q}}_S[k,:] = \begin{bmatrix} \dot{\boldsymbol{q}}_S[k,1] \\ \dot{\boldsymbol{q}}_S[k,2] \\ \vdots \\ \dot{\boldsymbol{q}}_S[k,H_c] \end{bmatrix} = \boldsymbol{\Psi}_{act,vel}\ \xi[k,1] + \boldsymbol{\theta}_{act,vel}\ \Delta\boldsymbol{U}_S^T[k,:] \tag{5.66}$$

where the matrices $\boldsymbol{\Psi}_{act,vel}$ and $\boldsymbol{\theta}_{act,vel}$ have a similar structure as the matrices $\boldsymbol{\Psi}_{act}$ and $\boldsymbol{\theta}_{act}$. The actuator accelerations $\ddot{\boldsymbol{q}}_S$ are estimated by a discrete differentiation of the actuator velocities $\dot{\boldsymbol{q}}_S$ in the form

$$\ddot{\boldsymbol{q}}_S[k,i+1] = (\dot{\boldsymbol{q}}_S[k,i+1] - \dot{\boldsymbol{q}}_S[k,i])/T_{samp}, \qquad \text{for} \quad i=1\dots H_c-1. \tag{5.67}$$

5.7 Quadratic Programming Problem

The QP-problem is formulated to

$$\min_{\Delta\boldsymbol{U}_S} \quad J[k] \tag{5.68}$$

$$\begin{aligned} s.t. \quad & \Delta\boldsymbol{u}_{S,min} \le \Delta\boldsymbol{u}_S[k,i] \le \Delta\boldsymbol{u}_{S,max}, && \text{for } i=1\dots H_u \\ & \Delta\boldsymbol{u}_S[k,i] = 0, && \text{for } i=(H_u+1)\dots H_p \\ & \boldsymbol{u}_{S,min} \le \boldsymbol{u}_S[k,i] \le \boldsymbol{u}_{S,max}, && \text{for } i=1\dots H_c \\ & \boldsymbol{d}_{S,min} \le \boldsymbol{d}_S[k,i] \le \boldsymbol{d}_{S,max}, && \text{for } i=1\dots H_c \\ & \boldsymbol{q}_{S,min} \le \boldsymbol{q}_S[k,i] \le \boldsymbol{q}_{S,max}, && \text{for } i=1\dots H_c \\ & \dot{\boldsymbol{q}}_{S,min} \le \dot{\boldsymbol{q}}_S[k,i] \le \dot{\boldsymbol{q}}_{S,max}, && \text{for } i=1\dots H_c \\ & \ddot{\boldsymbol{q}}_{S,min} \le \ddot{\boldsymbol{q}}_S[k,i] \le \ddot{\boldsymbol{q}}_{S,max}, && \text{for } i=1\dots H_c. \end{aligned} \tag{5.69}$$

Constraints are included in form of inequality equations for the inputs $\Delta\boldsymbol{u}_S$ and the actuator strokes $\boldsymbol{q}_S$, the actuator velocities $\dot{\boldsymbol{q}}_S$ and the actuator accelerations $\ddot{\boldsymbol{q}}_S$ where the index min and max indicate the lower and upper limit for the respective variable. The present work uses the QP-solver qpOASES [65] to solve the optimization problem with quadratic objectives and linear constraints.

5.8 Simulative Analysis

Results are shown for a manoeuvre which includes an acceleration and deceleration driving scenario in longitudinal direction. For the estimation, a prediction horizon with the discrete time steps $H_u = 50$, $H_c = 50$ and $H_p = 300$ and a sampling time $T_{samp} = 0.01\,\text{s}$ is used. The choice of the applied weighting parameters $\kappa_{i,i}$ and $\xi_{i,i}$, introduced in equations (5.39) and (5.40), are mainly based on a suitable motion distribution between tripod and hexapod system to fully exploit their workspace capabilities. The hand-tuned weights are shown in Appendix D in Table D.1. Perceived motions in a real vehicle using equations (5.10) are used as reference signals to determine deviations between the expected driver's perception compared to the simulator's perception. To classify deviations, an error function is defined

$$e_\xi[k] = \mathrm{sgn}(y_{S,\xi}[k]) \cdot \mathrm{sgn}(y_{V,\xi}[k]) \cdot ||y_{S,\xi}[k] - y_{V,\xi}[k]|| \tag{5.70}$$

for $k = 1 \dots k_{end}$ and $\xi = \{x, y, z, \varphi, \theta, \psi\}$. The sign of the cues coming from the simulator and the vehicle are included in the definition of the error amplitude in such a manner that the error will be negative if the vehicle and the simulator cues are of opposite sign. Participants typically rate sign errors worse compared to scaling errors as the driver perceives an opposite motion to its expectation [37].
The left part of Figure 5.7 shows the specific force ${}_H\hat{f}_{S,x}$ as well as the reference trajectory ${}_V\hat{f}_{V,x}$. The right part of Figure 5.7 clarifies deviations between ${}_H\hat{f}_{S,x} - {}_V\hat{f}_{V,x}$. The dashed line represents the perception threshold based on [141]. Analogous to the estimation of the weighting parameters in equation (5.39), it is assumed that errors below the perception threshold are not perceived. The right plot of Figure 5.7 mainly reveals scaling errors when the reference signal (the left plot) is larger than approximately 1 m/s^2. Sign errors are generally below the perception threshold. Figure 5.8 shows the angular velocities ${}_H\hat{\omega}_{S,y}$ with respect to the y-axes in the H FoR as well as the corresponding reference ${}_V\hat{\omega}_{V,y}$. The right part of Figure 5.8 applies equation (5.70) to the signals ${}_H\hat{\omega}_{S,y}$ and ${}_V\hat{\omega}_{V,y}$ with the corresponding perception threshold of [141]. The values for ${}_H\hat{\omega}_{S,y}$ significantly exceed those for ${}_V\hat{\omega}_{V,y}$ as a result of the tilting function of equation (5.20) providing a translational acceleration through a corresponding rotation of the hexapod platform. Nevertheless, deviations are mainly below the perception threshold.

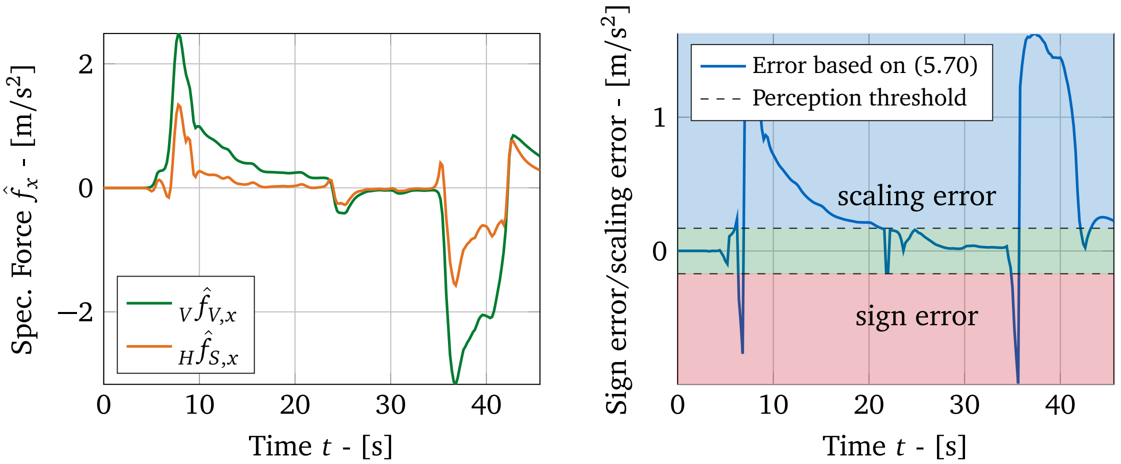


Figure 5.7: Left Plot: Perceived specific forces ${}_H\hat{f}_{S,x}$ and ${}_V\hat{f}_{V,x}$. Right plot: Estimated error between ${}_H\hat{f}_{S,x}$ and ${}_V\hat{f}_{V,x}$ based on equation (5.70).

The following Section focuses on approximations of the actuator states. Figure 5.9 shows the actuator strokes $q_{h,l}$, velocities $\dot{q}_{h,l}$ and accelerations $\ddot{q}_{h,l}$ for the six actuators of the hexapod. For the sake of brevity, the tripod's actuator strokes $q_{t,l}$, actuator velocities and actuator accelerations are depicted in Appendix B. These figures reveal the fulfilment of the corresponding actuator limits. Deviations between the approximated actuator states and the actual actuator states $\boldsymbol{q}_{\text{real}}$, $\dot{\boldsymbol{q}}_{\text{real}}$ and $\ddot{\boldsymbol{q}}_{\text{real}}$ are compared to evaluate the error over the constraint horizon H_c. The bottom, right plot of Figure 5.9 shows the maximum error of the actuator length over the considered manoeuvre. Herein, the maximum error at each sampling point k over the constraint horizon for $i = 1 \dots H_c$ is calculated by

$$\varepsilon_{\text{max,time}}[k] = \max_i(||q_{n,l}[k,i] - q_{n,l,\text{real}}[k,i]||) \tag{5.71}$$

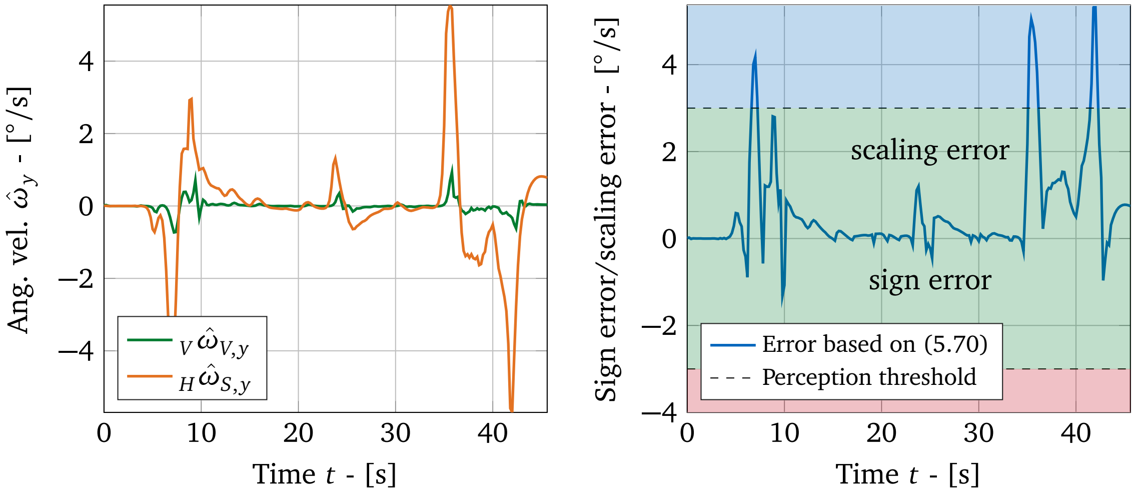


Figure 5.8: Left Plot: Perceived rotational velocity $_H\hat{\omega}_{S,y}$ and $_V\hat{\omega}_{V,y}$. Right plot: Estimated error between $_H\hat{\omega}_{S,y}$ and $_V\hat{\omega}_{V,y}$ based on equation (5.70).

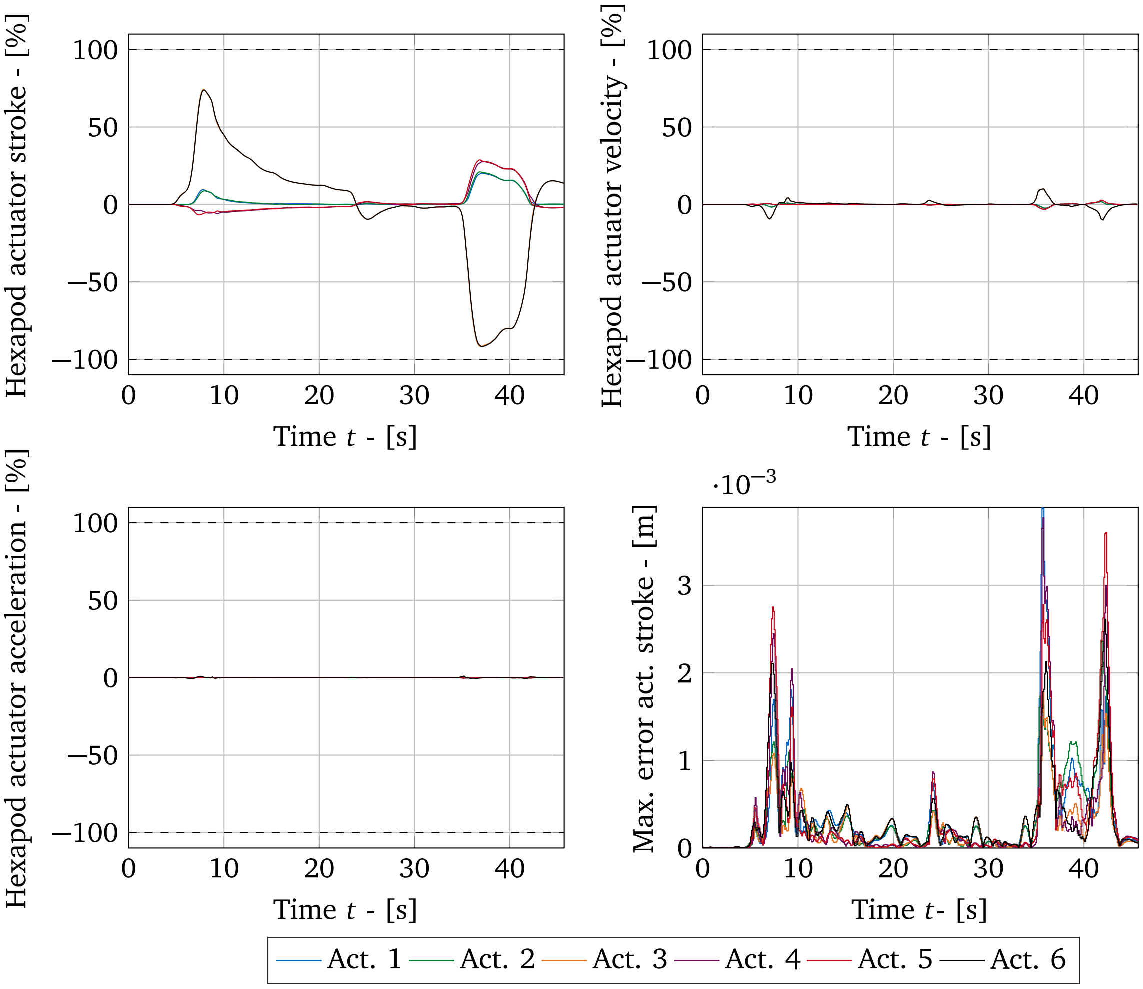


Figure 5.9: Actuator stroke (top, left), velocity (top, right) and acceleration (bottom, left) of the hexapod for the considered manoeuvre. The values are normalized to the maximum and minimum limits. The maximum error in the actuator stroke over the constraint horizon H_c (bottom right).

The maximum magnitudes of the error are in a reasonable range. The error can be compensated by the introduction of safety margins. The estimated error values $\varepsilon_{\text{max,time}}$ depend largely on the driving scenario. A correlation can be seen when comparing the time instances of the error peaks (bottom, right plot of Figure 5.9) and the time instances of the actuator velocities' peaks (top, right plot of Figure 5.9). The peaks of both plots relate to the same time instances which shows that more dynamic motions of the simulator yields higher errors $\varepsilon_{\text{max,time}}$. This is not surprising as at these points the deviation to the operating point increases more rapidly over the time horizon than at time sections with lower actuator motions.

In the following Section, two analyses are performed. First, the influence of the constraint horizon on the actuator error is shown. Second, the influence of the constraint horizon H_c and the input horizon H_u on the deviation between perceived and expected motions is analysed. Regarding the first analysis, Table 5.1 lists two different types of errors for the actuator lengths and actuator velocities of the hexapod and tripod system:

1. The absolute maximum error $\varepsilon_{n,\text{max}}$ is determined by

$$\varepsilon_{n,\text{max}} = \frac{\max\limits_{k,i,l}(||q_{n,l}[k,i] - q_{n,l,\text{real}}[k,i]||)}{q_{n,\text{max}}} \cdot 100\,\%, \tag{5.72}$$

   over the complete manoeuvre $k = 1 \dots k_{end}$, including the constraint horizon $i = 1 \dots H_c$, all actuators l and for the tripod and hexapod $n = t, h$, respectively. The errors are taken relative to the maximum actuator strokes of $q_{n,\text{max}}$ for the hexapod and tripod. In other words, the maximum error $\varepsilon_{n,\text{max}}$ is estimated considering all time horizons H_c for the manoeuvre (k_{end} time instances). Increasing $\varepsilon_{n,\text{max}}$ are assumed for larger H_c as the deviation to the operating point of the linearization increases.

2. The average errors $\varepsilon_{n,\text{avg}}$ determined from the maximum errors for each time horizon H_c over the complete manoeuvre k_{end} and actuators l are calculated by

$$\varepsilon_{n,\text{avg}} = \frac{\sum_{k=1}^{k_{end}} \max\limits_{i,l}(||q_{n,l}[k,i] - q_{n,l,\text{real}}[k,i]||)}{k_{end}\, q_{n,\text{max}}} \cdot 100\,\%, \tag{5.73}$$

   for $i = 1 \dots H_c$, $n = t, h$ and the number of actuators l. In other words, the average error $\varepsilon_{n,\text{avg}}$ calculates the maximum errors for each time horizon H_c (k_{end} errors in total) and averages them for the considered manoeuvre (k_{end} time instances). Thus, a comparison between $\varepsilon_{n,\text{avg}}$ and $\varepsilon_{n,\text{max}}$ gives an indication about the incidences of $\varepsilon_{n,\text{max}}$ to $\varepsilon_{n,\text{avg}}$ for the considered manoeuvre.

The errors $\varepsilon_{n,\text{max}}$ and $\varepsilon_{n,\text{avg}}$ are estimated for the actuator velocities $\dot{q}$ in the same way. The errors are taken relative to the maximum actuator velocities $\dot{q}_{max}$ for the hexapod and tripod.

Table 5.1 includes the errors estimated for a constraint horizon with the discrete time steps $H_c = 25$, $H_c = 50$ and $H_c = 100$ based on the considered driving manoeuvre ($H_u = 50$ in all calculations). Comparing both settings, the errors for the longer horizon exhibit higher values compared to those obtained for the shorter horizon, mainly resulting from the linearization of the actuator states. The longer the horizon, the more pronounced the deviation to the operating point of the linearization becomes. Put differently, this shows that $\varepsilon_{n,\text{max}}$ and $\varepsilon_{n,\text{avg}}$ become smaller towards the beginning of the constraint horizon, which is important in order not to run into actuator limits. The maximum error gives an indication about the safety margin which is needed for the constraints in the optimization.

Table 5.1: Maximal errors $\varepsilon_{n,\max}$ (equation (5.72)) and average errors $\varepsilon_{n,\text{avg}}$ (equation (5.73)) for the hexapod and tripod based on the linearization on the actuator strokes q and velocities $\dot{q}$.

	$H_c = 25$		$H_c = 50$		$H_c = 100$	
	q	$\dot{q}$	q	$\dot{q}$	q	$\dot{q}$
Hexapod						
$\varepsilon_{h,\max}$ [%]	1.03	5.70	1.18	10.08	1.73	10.08
$\varepsilon_{h,\text{avg}}$ [%]	0.11	0.94	0.15	1.94	0.22	1.94
Tripod						
$\varepsilon_{t,\max}$ [%]	0.22	5.65	0.83	17.33	1.81	42.57
$\varepsilon_{t,\text{avg}}$ [%]	0.01	0.62	0.04	1.92	0.11	4.96

Regarding the second analysis, a comparison is performed. Scaling and sign errors are summed up over the considered manoeuvre in equations (5.74) and (5.75) and divided by the number of discrete sampling points k_{end} as well as the otolith's and semicircular's perception threshold ($\tau_{semi} = 3\,°/\text{s}^2$ and $\tau_{oto} = 0.17\,\text{m}/\text{s}^2$ [141]).

$$E_{\xi,scale} = \sum_{k=1}^{k_{end}} \frac{\varepsilon_{\xi}[k]}{k_{end}\ \tau_{semi/oto}} \cdot 100\,\% \qquad \forall \varepsilon_{\xi} > 0 \tag{5.74}$$

$$E_{\xi,sign} = \sum_{k=1}^{k_{end}} \frac{\varepsilon_{\xi}[k]}{k_{end}\ \tau_{semi/oto}} \cdot 100\,\% \qquad \forall \varepsilon_{\xi} < 0. \tag{5.75}$$

$E_{\xi,scale}$ and $E_{\xi,sign}$ are listed for longitudinal accelerations and pitch velocities in Table 5.2.
In the following, effects of the defined error norms $E_{\xi,scale}$ and $E_{\xi,sign}$ and the computation time in relation to horizon's lengths H_u and H_c are discussed. Hereby, the maximum computation time t_{max} to solve an optimization problem at an MPC step k for $k = 2 \dots k_{end}$ is considered. Using a hot-start option in the optimization solver, the initial MPC-step $k = 1$ is not considered because it is performed in the initialization phase of the simulator. The estimations are conducted with an i7-6820HQ CPU @ 2.70GHz in Matlab [109]. It is assumed that a decrease in H_c and H_u leads to a decline in t_{max}.
Results in Table 5.2 show that variations in H_c do not have much influence on $E_{\xi,scale}$ and $E_{\xi,sign}$. However, smaller values of H_c yield lower t_{max} because the number of constraints decreases. As for H_u, modifications have greater effects on $E_{\xi,scale}$, $E_{\xi,sign}$ and t_{max} compared to H_c since the number of optimization parameters are reduced.
In summary, two assumptions can be verified. Firstly, partitioning of the time horizon has no significant effect on the quality of the MCA but leads to significant decreases in the computation time. Secondly, the linearization approach is justifiable because a long-term forecast of the actuator states over the prediction horizon is not implicitly necessary.

5.9 Discussion

The following Section is a digression which discusses the differences between a workspace-based and an actuator-based description of the optimization parameters. The MCA approach

Table 5.2: Summarized sign and scaling errors based on equations (5.74) and (5.75) over the considered manoeuvre.

	$H_c = 25$	$H_c = 50$	$H_c = 100$	$H_u = 25$	$H_u = 50$	$H_u = 100$
		$H_u = 50$			$H_c = 50$	
Longitudinal Accelerations						
$E_{x,scale}$ [%]	222.98	222.98	222.98	235.41	222.98	220.75
$E_{x,sign}$ [%]	-12.93	-12.93	-12.93	-14.41	-12.93	-10.18
Pitch Angular Velocities						
$E_{\theta,scale}$[%]	0.38	0.38	0.38	0.39	0.38	0.35
$E_{\theta,sign}$ [%]	-0.02	-0.02	-0.02	-0.01	-0.02	-0.04
Maximal CPU time						
t_{max} [s]	0.042	0.273	0.334	0.012	0.273	3.63

shown in this Chapter uses workspace coordinates as optimization parameters (cf. equation (5.3)). Thus, the specific forces and the angular velocities can be directly expressed by the relative kinematics. Including the actuator constraints of the motion system to the optimization problem requires to compute the nonlinear, inverse kinematics. Since a usual QP-solver can only handle linear inequality constraints, a direct integration of the inverse kinematics is not possible. Thus, a linearization scheme is described in this Chapter to approximate the inverse kinematics for the parallel structure.
Another possibility is to use an actuator-based approach with the actuator accelerations $\ddot{\boldsymbol{q}}_S$ defined as the optimization parameters in the form

$$\boldsymbol{u}_S = \ddot{\boldsymbol{q}}_S. \tag{5.76}$$

This approach has been published in [49] and the following Section is based on this publication. It is the result of a collaboration with Florian Oberleitner [128]. Advantages lie in the linear integration of the actuator constraints on position, velocity and acceleration level to the optimal control problem. The direct kinematics is used to estimate the specific forces and angular velocity which are expressed in workspace coordinates. As described in Chapter 2.5.1, estimating the direct kinematics for parallel kinematic structures usually implies solving a nonlinear system of equations, which can be handled by a numerical scheme, such as the Newton-Raphson method [115]. This often goes along with a high computational effort. Thus, in [49], two approximations of the direct kinematics are proposed. Results demonstrate significant reductions in the computation time, while showing only small deviations from the exact kinematics. Though, the approach is applied to a hexapod structure. An extension to redundant structures makes a description of the optimization problem considerably more complex. Along with this, the approximations of the direct kinematics are also based on a linearization scheme which yields deviations over the time horizon. The simulator, used in this thesis, is controlled by workspace-based commands and does not provide an interface to directly control the actuators. Thus, the following developments are based on a workspace-based approach of the optimization parameters.

5.10 Chapter Summary

In this Section, an MCA for a nine DoFs motion simulator based on an MPC-approach is proposed. The Section focuses on the distribution of the redundant DoFs based on a global optimization scheme. The algorithm takes into account the nonlinearities of the kinematic chain, comprising a hexapod mounted on a tripod motion platform. The nonlinear terms are handled by a linearization approach in order to obtain a linear prediction model. The method presented here can be applied to other motion systems. The objective function mainly minimizes the deviation between perceived motions of a driver in a real vehicle compared to those of a simulator driver. In order to integrate actuators' limitations on position, velocity and acceleration level for the tripod and hexapod, a linearization approach was used that applied the constraints linearly on the optimization problem, leading to a reduction in computation time.

In the subsequent section, an experiment with test persons is conducted to determine the motion cueing quality of the proposed MCA compared to a benchmark MCA. The algorithm at this stage is utilized as an open-loop application.

Chapter 6

Open-Loop, MPC-Based MCA (MCA$_{\text{OPT}}$)

The content of the following Chapter have been published in [54, 55].

This Chapter describes the evaluation of the proposed MCA approach introduced in Chapter 5. The version of the MCA used in this Chapter is abbreviated by MCA$_{\text{OPT}}$. An experiment was conducted with 35 participants. The experiment aimed at investigating the potentials of the MCA$_{\text{OPT}}$ compared to a benchmark algorithm (MCA$_{\text{BM}}$) which is the default motion cueing that was provided with the motion system (cf. Figure 1.3). The MCA$_{\text{BM}}$ is based on an optimization scheme wherein the reference signals are manipulated by filter elements [77]. The evaluation of the MCA$_{\text{OPT}}$ is performed by using the continuous rating method (CR). In the process, discrepancies between the expected motions obtained from the visualization and the actual perceived motions are rated over time.

6.1 Related Work

Throughout the work, motion discrepancies are divided into [1]

1. Objective Motion Incongruencies (OMIs): Physical deviations between desired vehicle motions and actual simulator motions.

2. Perceived Motion Incongruencies (PMIs): Perceived deviations between expected vehicle motions and perceived simulator motions.

In the experiment, the continuous rating is used to measure the PMIs. In [37, 38], the continuous rating was used in a first experiment to measure PMIs between three different MCAs. Results show that the rating method was able to consistently measure PMIs with participants of different backgrounds. Results were compared to an offline rating method to check the validity of the novel rating method. An advantage of the continuous rating method is that PMIs can be analysed to a higher temporal resolution in order to determine the origin of the motion cueing errors. In [36], the method was applied in a comparison between a filter-based MCA and an optimization-based MCA to measure the motion cueing quality. Similar to the works [37, 38], the rating method showed reliable and repeatable results within and between the participants.

The basic approach of the continuous rating method was applied in the evaluation of the MCA$_{\text{OPT}}$. However, as a new rating scale and experimental design are introduced, ratings are checked for reliability and validity. Additionally, an oral rating is used to validate the continuous rating [36, 37, 137].

6.2 Contribution

One major objective of this Chapter is the evaluation of $\mathrm{MCA_{OPT}}$ which is the reason a driving simulation experiment was conducted. The first research question is aimed at investigating whether the continuous rating procedure is a valid method to determine PMIs in the current setup. Due to the novelty of the rating method, introduced in [36, 37], the current setup is still to be demonstrated. For this reason, ratings are analysed for reliability and validity within and between the participants. Reliability is checked by estimating the Cronbach's alpha [40]. Validity is verified by comparing the results with an additional post-hoc rating [37]. The rating methods used in the study are comparable to previous works [36, 37] but differs in the experimental design. In [36], a comparison between a filter and an optimization-based MCA is conducted whereas, in this study, the potential of the proposed MCA approach against a state-of-the-art, optimization-based benchmark algorithm is performed. Comparing the simulator architectures, the simulator used in [36] probably exhibit higher workspace capabilities. The second research question aims at illustrating the potential of the novel MCA compared to a state-of-the-art optimization-based MCA for the nine DoFs motion system. Before the experiment was conducted, it was unknown whether the two algorithms, which are both optimization-based, would result in measurably different PMIs. In a last step, a rating model $\mathrm{RM_M}$ is developed, which is fitted on the measured ratings to obtain the parameters for a model that can predict PMIs from OMIs. The structure of the model is based on the approach in [37] but differs in weighting parameters and transfer functions.

6.3 MCAs Used for Comparison

This Section describes the configuration of the $\mathrm{MCA_{OPT}}$ and the structure of $\mathrm{MCA_{BM}}$which were used in the experimental study.

6.3.1 $\mathrm{MCA_{OPT}}$

The structure of the $\mathrm{MCA_{OPT}}$ is shown in Figure 5.3. The $\mathrm{MCA_{OPT}}$ did not have the capability to run for closed-loop applications at the time of conducting the experiment. Thus, the algorithm was used as an open-loop MCA. The driving dynamics data was measured for a pre-defined driving manoeuvre. The $\mathrm{MCA_{OPT}}$ calculates the simulator's motion commands in an offline pre-processing step using the measured driving dynamics as an input signal. Since the driving manoeuvre was known a-priori, the inputs $\boldsymbol{u}_V$ could be used over the prediction horizon H_p to calculate the reference trajectory $\boldsymbol{y}_V$ which leads to an ideal prediction (c.f. the objective function introduced in equation (5.37)). The MPC-scheme is conduced with a sample time of $T_{samp} = 0.01\,\mathrm{s}$ and a time horizon $T_{H_p} = H_p \cdot T_{samp} = 300 \cdot 0.01\,\mathrm{s} = 3\,\mathrm{s}$. The hand-tuned weights are shown in Appendix D in Table D.2.

6.3.2 Benchmark $\mathrm{MCA_{BM}}$

A benchmark algorithm from the simulator's manufacturer [77] is taken for evaluation and comparison on the nine DoFs simulator. The algorithm is also based on an optimization scheme and includes a vestibular model. The structure of the objective function is comparable to the one in equation (5.37). The benchmark algorithm applies a prefiltering of the reference $\hat{\boldsymbol{y}}_V$ and exhibits the capability to run in real-time, which constitute main differences between the two algorithms used in the current study. Translational accelerations are filtered into high frequency signals functioning as translational references. Low frequency signals are transformed to respective tilt-angle signals. The MCA$_{\text{OPT}}$, by contrast, uses a single optimization problem to solve the redundant DoFs. An additional difference between the algorithms probably exists on the time horizon. For MCA$_{\text{OPT}}$, the prediction time is 3s with an ideal prediction reference. The prediction time and the prediction strategy are unknown for the benchmark algorithm. The integration method of workspace restrictions is also not stated in [77].

6.4 Experimental Evaluation

The experiment was conducted in the nine DoFs simulator at the BMW Group (cf. Figure 1.3) with 35 participants (8 females) between the age of 23-59 (μ: 33 (cf. equation (6.4)); σ: 10.34 (cf. equation (6.5))). The participants were students or employees at the Technische Universität München or the BMW Group. Of the participants, 17 participants had never experienced a driving simulator. For the remaining participants, the driving simulator experience ranged between: 1-3 times (7), 4-6 times (5), and more than 6 times (6). All the participants own a driving licence. The mean mileage is approx. 15 286 km per year (σ: 9 829 km). One participant had to prematurely terminate the experiment due to technical problems. One participant aborted the experiment because of motion sickness.

6.4.1 Rating Procedure

The measurement procedure and the structure of the experiment follows the procedure proposed in [36, 37]. Participants rate the PMIs continuously using a rotational knob located at the central console of the mockup, depicted in Figure 6.2. Consequently, a time- and position-dependent rating is obtained. The PMIs are rated using an 11-point scale that is displayed on a coloured bar on the driving simulation screen. A rating of 0 represents no PMI („real driving sensation“) whereas a rating of 10 denotes a high PMI („large motion mismatch“). The maximum rating was anchored in a training session before the experiment where several manoeuvres exhibited large OMIs. The participants were instructed such that a rating of 10 was reached at least once in the training session, in order to obtain a reference for the CR (cf. Figure 6.1).

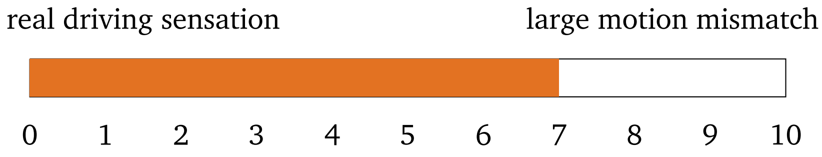


Figure 6.1: Scale [0-10] rating the PMIs.

An additional post-hoc rating (PR) at the end of each CR is used to evaluate the reliability and the validity of the CR [37]. The PR uses the same relative scale as the CR with the exception that ratings higher than 10 are possible to examine the anchoring of the training session. To ensure comparability, the MCA$_{\text{BM}}$ and MCA$_{\text{OPT}}$ are tested for the same prerecorded driving manoeuvre. The driver can completely concentrate on the rating in the offline simulation as driver inputs are not required.

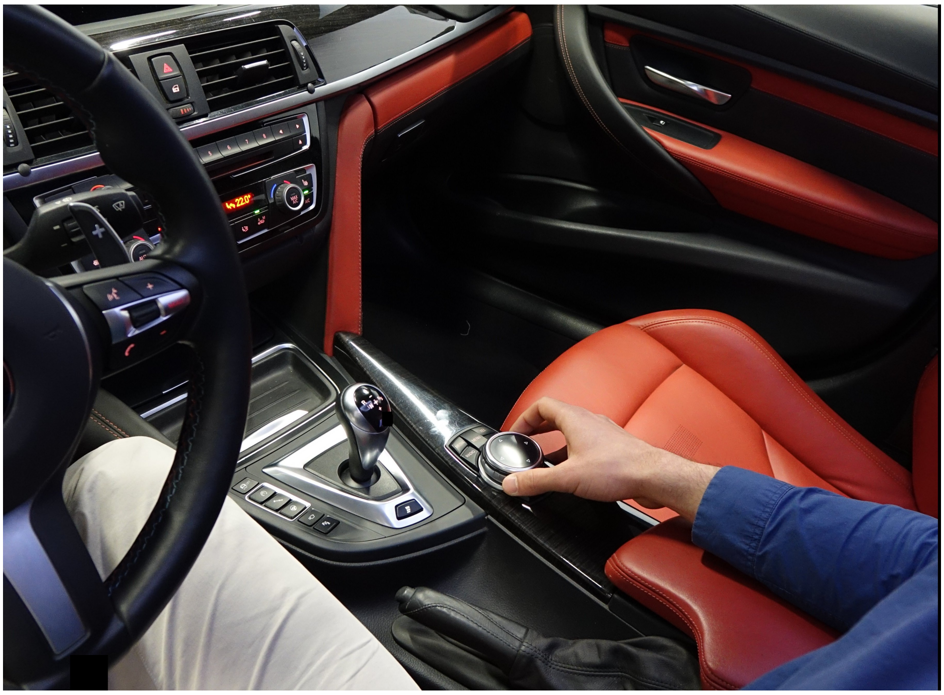


Figure 6.2: Rating knob for the CR located on the central console of the mockup.

6.4.2 Structure of the Experiment

A training exercise at the beginning of each experiment is conducted to familiarize participants with the simulator environment and the rating procedure. The training exercise is performed twice. If the ratings during the training exercise was inconsistent (judged by the experiment leader, based on visual inspection), the training was repeated a third time. The manoeuvre of the training sessions differs from the above-mentioned manoeuvre used in the subsequent experiment. The training manoeuvre contains an overland and an inner-city portion with a total duration of about 2 minutes. The track exhibits several turns and curves with different radii. Besides, various longitudinal acceleration and deceleration manoeuvres are included to cover a broad range of driving dynamics requirements.

Following training, the experiment commenced, in which a pre-recorded manoeuvre was used with a duration of five minutes. Participants were asked to rate the PMIs using CR. The track was repeated four times for each of the two MCAs. Ordering of the eight simulation trials is randomly permuted during the experiment. After each rating, the participant was allowed a short break of two minutes before the next rating started. The total experiment for a participant lasted about 1.5 h.

6.4.3 Independent and Dependent Variables

Independent variable in the experiment is the MCA (two levels: MCA$_{BM}$ and MCA$_{OPT}$). These algorithms are embedded in a within-subject design. By these means, both algorithms are compared, in a four-times randomly paired comparison.

The driving manoeuvre used is a representative simulator drive, consisting of overland and inner-city elements as depicted in Figure 6.3. The track is divided into ten parts which are listed in Table 6.1.

Dependent variables are the two rating methods (CR and PR). The CR rates the PMIs for each of the eight simulation trials. The PR rates the overall PMIs at the end of each simulation trial.

Table 6.1: The driving manoeuvre used in the experiment is divided following ten parts.

1.	*Acceleration procedure*	Medium-strong acceleration to approx. 80 km/h.
2.	*Overland course 1*	Overland course with long curves at approx. 80 km/h.
3.	*Roundabout*	Strong deceleration from approx. 80 km/h to approx. 30 km/h. Taking the first exit in the roundabout (75° turn). Medium-strong acceleration to approx. 80 km/h.
4.	*Overland course 2*	Overland course with largely long curves at 90 km/h.
5.	*Sharp hilltop 1*	Crossing a hill with a sharp top at approx. 60 km/h.
6.	*Sharp turn 1*	Long right turn followed by a sharp left turn at approx. 60 km/h.
7.	*Sharp S-curves*	Sharp S-curves at 50 km/h.
8.	*Sharp turn 2*	Extended 90° left turn at approx. 60 km/h.
9.	*Sharp hilltop 2*	Crossing a hill with a medium sharp top 70 km/h.
10.	*Deceleration procedure*	Medium-strong deceleration from 70 km/h to 0 km/h.

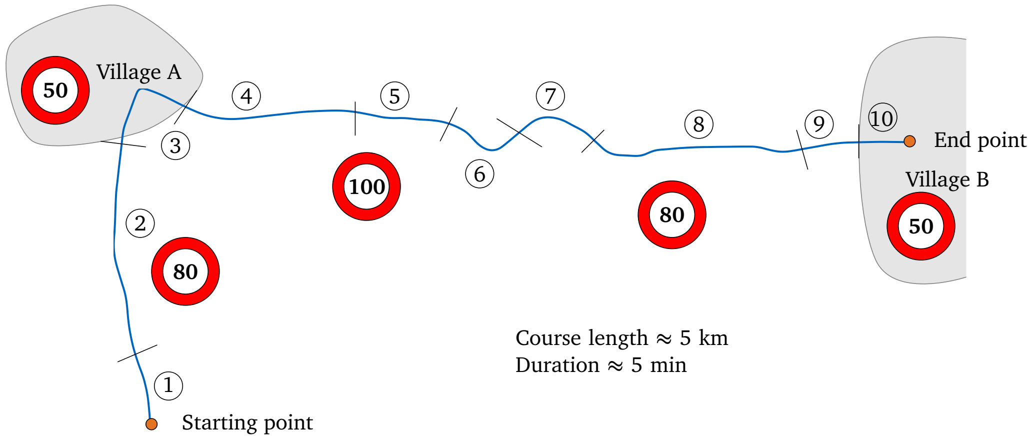


Figure 6.3: The driving manoeuvre used in the experiment, consisting of overland and inner-city elements. The track is divided into ten parts which are listed in Table 6.1.

6.4.4 Structure of the Rating Model

A model proposed in [37] aimed at describing the CR based on the vehicle and simulator motions. The model can be divided into the perception subsystem $\tilde{PS}$ and the rating subsystem $\tilde{RS}$ (see Figure 6.4). An application area consists in the integration of the model into the cost function in equation (5.37) replacing the perception system model (cf. Figure 5.3). Thus, further information is included in the cost function. Besides the perception system model, ratings of perceived motion incongruencies are considered as well. Furthermore, the tuning of the algorithm is simplified by finding a compromise between tilt motions and translational motions.

In this work, the approaches of [37] are adopted, evaluated and applied on the current experiment. In the following, the original model - as proposed in [37], is briefly described. Adaptations that were made in the current study will be discussed later. Input of $\tilde{PS}$ is the differences between the simulator motions $\boldsymbol{y}_S$ and the expected motions $\boldsymbol{y}_V$ of a virtual vehicle for the $x-$, $y-$ and $z-$ translational and rotational directions (cf. Figure 5.3). The six weighting parameters $\tilde{\boldsymbol{W}}$ are determined by a least-square approach. A moving averaging filter in the form

$$H[z] = \frac{1 + z^{-1} + \cdots + z^{-N+1}}{N} \tag{6.1}$$

with the total number of time samples N and with an additional offset C is defined to fit the output $\tilde{P}(t)$ of the system $\tilde{PS}$ to the rating $R(t)$. In [37], the parameters are defined to $N = 3$ and $C = 0.087$. Output of $\tilde{RS}$ is the simulated rating $\tilde{R}(t)$.

The rating model by [37] (referred to here as RM$_{\mathrm{C}}$) is modified to a rating model (referred to here as RM$_{\mathrm{M}}$) that exhibits basically the same structure as in Figure 6.4. Since the parameters used in RM$_{\mathrm{C}}$ are used for lateral motion ratings, a new parameter set is proposed in this work. The weighting parameters $\tilde{\boldsymbol{W}} = [w_1 \dots w_6]$ for the RM$_{\mathrm{M}}$ are obtained by an optimization problem which is defined in the form

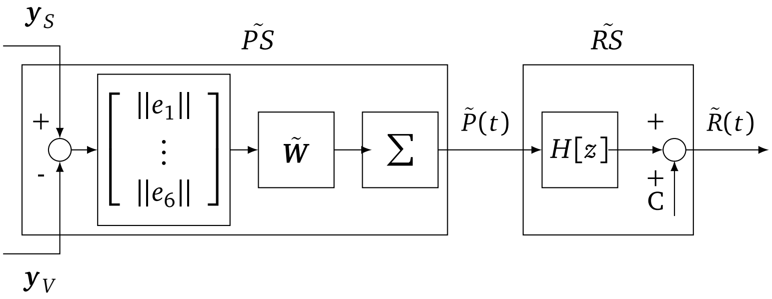


Figure 6.4: Model of the rating system [37].

$$\min_{\tilde{\boldsymbol{W}}} \sum_{l=1}^{N} \left(R[l] - H[z] \sum_{i=1}^{6} \left(w_i \, ||y_{S,i}[l] - y_{V,i}[l]|| \, \right) \right)^2 \tag{6.2}$$
$$\text{s.t.} \quad \tilde{\boldsymbol{W}} \geq \boldsymbol{0}$$

where $\boldsymbol{R}$ is the actual rating of the manoeuvre with sampling times $[1 \dots N]$. The weighting parameters $\tilde{\boldsymbol{W}}$ transform the six-dimensional deviations of the simulation into one dimension to make a comparison with the one dimensional rating $R(t)$ possible. The parameter optimization problem is solved by a global search approach under the use of a multi-start algorithm [107].

The transfer function's dimension N in equation (6.1) is estimated by including the optimization problem from equation (6.2) into an additional grid search scheme. Estimating the root-mean-square (rms) $||R(t) - \tilde{R}(t)||_{rms}$ for each $N \in [1 \dots N_{max}, \mathbb{N}]$, the N-value with the minimum rms-value is taken for the final model.

6.5 Objective Motion Incongruencies (OMIs)

Figure 6.5 compares the resulting physical motions of the MCA$_{\mathrm{BM}}$ and MCA$_{\mathrm{OPT}}$ and the reference motions resulting from the vehicle model, for longitudinal accelerations, lateral accelerations and yaw velocities (which were the main DoFs in this experiment). From these data the OMIs of the two MCAs for the manoeuvre used in this experiment can be determined. The root-mean-squares of $||\mathrm{OMI} = \tilde{y}_V - \tilde{y}_S||$ are listed in Table 6.2 (cf. Figure 5.3). The results show that the rms-values are smaller for $\mathrm{MCA_{OPT}}$ than for $\mathrm{MCA_{BM}}$ for each DoF, indicating that the difference between desired (vehicle) and provided (simulator) motions was smaller for $\mathrm{MCA_{OPT}}$. In addition, the relative deviations

$$\tau_{\mathrm{OMI}} = \frac{\mathrm{rms}(\mathrm{OMI}_{\mathrm{MCA_{OPT}}}) - \mathrm{rms}(\mathrm{OMI}_{\mathrm{MCA_{BM}}})}{\mathrm{rms}(\mathrm{OMI}_{\mathrm{MCA_{BM}}})} \; [\%] \tag{6.3}$$

of the root-mean-squared OMIs of $\mathrm{MCA_{BM}}$ and $\mathrm{MCA_{OPT}}$ for each DoF are added to Table 6.2.

Table 6.2: Root-mean-square of the OMIs.

	Spec. forces. - [m/s^2]			Ang. vel. [rad/s]		
	x	y	z	x	y	z
$\mathrm{MCA_{BM}}$	0.625	0.833	0.134	0.017	0.015	0.065
$\mathrm{MCA_{OPT}}$	0.491	0.571	0.118	0.015	0.007	0.055
τ_{OMI} [%]	-21.4	-31.4	-12.4	-12.1	-54.0	-15.8

6.6 Experimental Results

To answer the research questions, formulated in Section 6.4, first the reliability and the validity of the rating is examined. In a second step, the ratings of the PMIs between both algorithms are analysed.

To check the central tendency (significance of mean-values) of two dependent samples, the paired t-test (test statistic = t) for normally distributed samples and the Wilkoxon signed rank test (test statistic = z) for non-normally distributed samples is used. The Lilliefors-test is used for testing normal distribution of a sample. Distinctions between three or more mean values of dependent samples are tested by a repeated measures ANOVA (test statistic = F) if Mauchly's sphericity test is positive ($p > 0.05$). Otherwise the Friedman-test (test statistic = χ^2) is used.

Throughout the thesis, the following statistical abbreviations are used for a sample $\boldsymbol{x}$ with N measurements:

- mean value

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \tag{6.4}$$

- standard deviation

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}} \tag{6.5}$$

- standard error

$$\sigma^* = \frac{\sigma}{\sqrt{N}}. \tag{6.6}$$

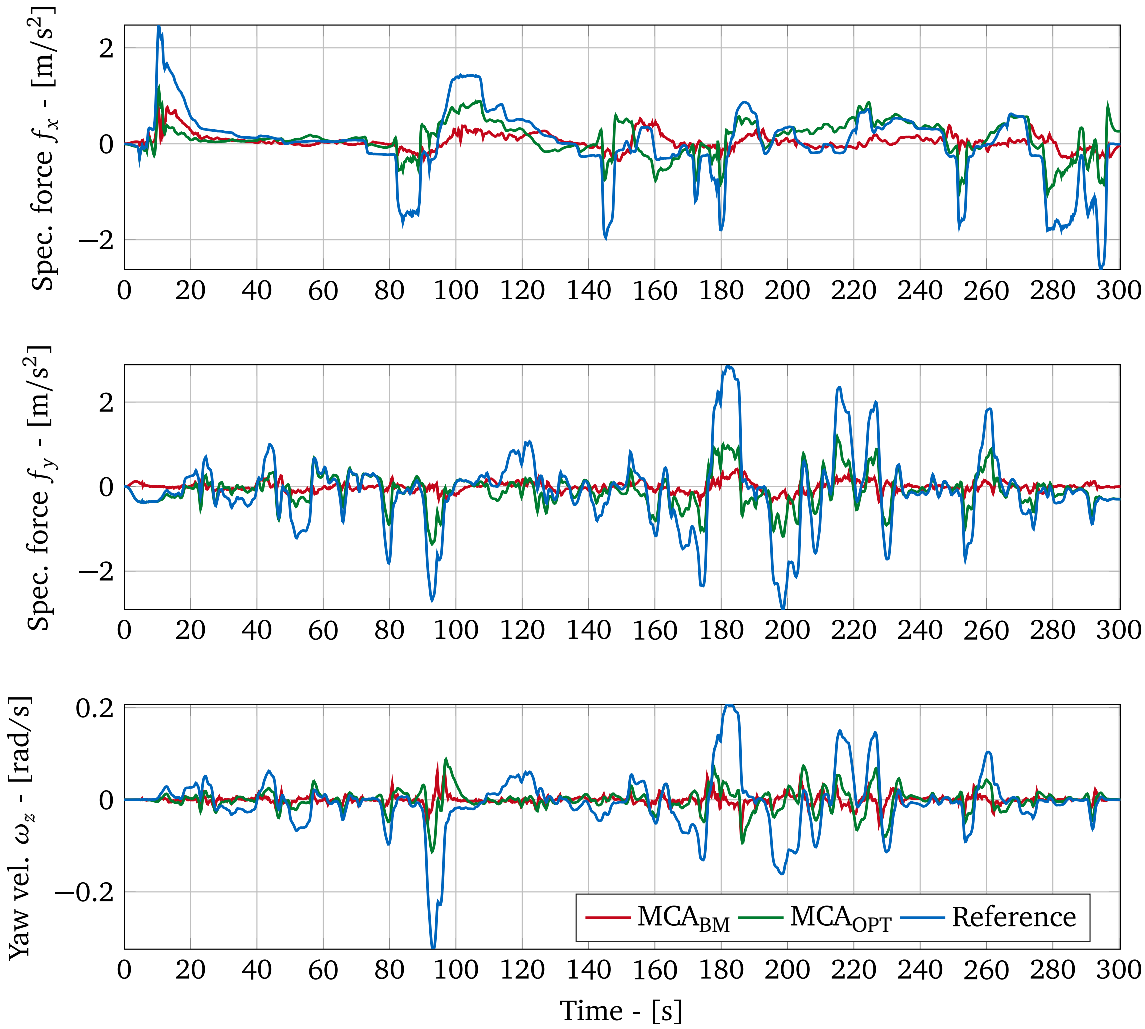


Figure 6.5: Longitudinal specific force f_x and lateral specific force f_y and angular velocity ω_z for the MCA$_{\text{BM}}$ and MCA$_{\text{OPT}}$ compared to the reference motions of a real car.

6.6.1 Reliability and Validity

Reliability is a measure of the internal consistency and the interrelatedness of ratings. It is most commonly estimated by the Cronbach's α [40]. The author in [127] suggests that a Cronbach's α bigger than 0.7 is acceptable. In our experiment, 26 participants out of 33 (2 participants aborted the experiment) obtain a higher Cronbach's α than 0.7 (cf. green markers Figure 6.6). Three participants have a Cronbach's α between $0.65 - 0.7$ (cf. orange otimes Figure 6.6). For those, based on the suggestion in [35, 166], the average interitem correlation is estimated for internal consistency which must be in the range of $0.15 - 0.50$. The average interitem correlations for the three rating trails per participant ID are [ID$_2$ = 0.33; ID$_4$ = 0.19; ID$_{11}$ = 0.23]. As all values are in the stated range, the three rating trails are also considered for further evaluations, which increases the content validity and leads to a visual clustering as depicted in Figure 6.6 [187]. Four Cronbach's α are in the range between $0.1 - 0.5$, which is why they are excluded from the study (cf. red circles Figure 6.6). Hence, a mean Cronbach's α of 0.77 (σ: 0.06) for the 29 participants is reached.

Validity specifies whether the specification of the experiment is measured [18]. In this study, the continuous rating is compared to a generally accepted post-hoc rating method [37]. Figure 6.7 shows the course of the mean ratings for MCA$_{\text{BM}}$ and MCA$_{\text{OPT}}$ over the manoeuvre

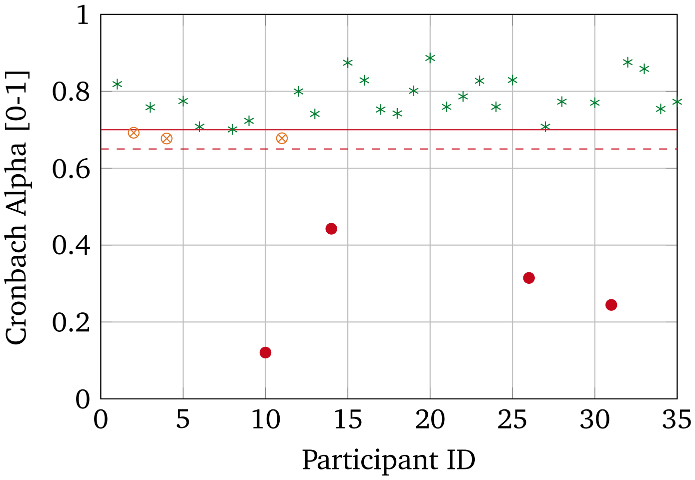


Figure 6.6: Cronbach's alpha for the 33 participants.

for CR and the mean for PR. [37] stated a significant linear relationship between the mean PR and the maximum value of the CR. In our case, no correlation between those values can be seen as an offset between both rating methods exists. Nevertheless, deviations between both MCAs for each rating system exhibit a significant linear relationship which can be estimated by the Pearson's correlation coefficient ($r = 0.4358$, $p < 0.05$). Here, the difference between both MCAs is estimated for the maximum CR as well as for the PR. Both differences are normalized by a z-transformation because of the different scales. The non-zero correlation confirms the assumption that the CR used in this experiment is a valid method.

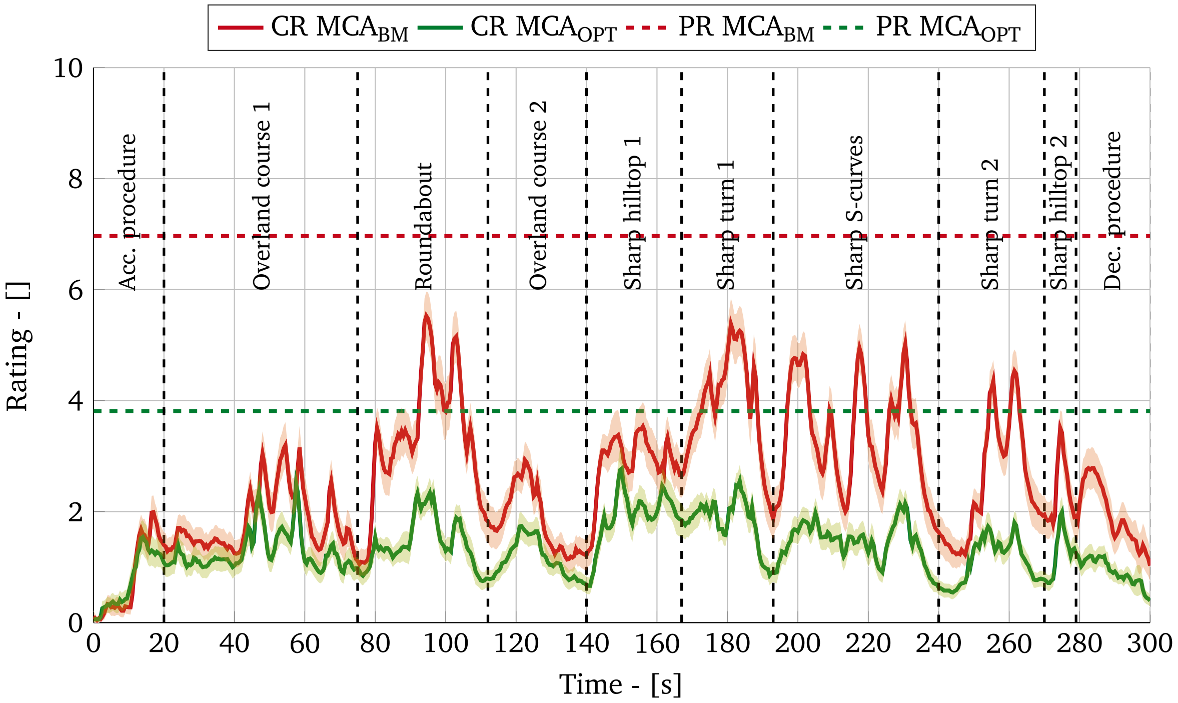


Figure 6.7: Mean CR with standard error (transparent area) and mean PR for the MCA$_{BM}$ and MCA$_{OPT}$ over the complete manoeuvre. Dashed lines separate the manoeuvre into the 10 sections as defined in 6.4.3.

6.6.2 Overall Rating

Post-hoc Rating: Figure 6.8 shows a significant difference ($t(28) = 8.6948, p < 0.001$) between the mean PR for MCA_{BM} ($\mu_{MCA_{BM}} = 6.97$; $\sigma^*_{MCA_{BM}} = 0.48$) and MCA_{OPT} ($\mu_{MCA_{OPT}} = 3.81$; $\sigma^*_{MCA_{OPT}} = 0.35$). As a reminder, the participants rated PMIs, where a higher value means a larger incoherence, hence worse motion cueing quality. The MCA_{OPT} reveals less motion discrepancies than the MCA_{BM}. The mean of the four rating pairs is depicted in Figure 6.8. The mean value of the PR does not change significantly for the four trials of the MCA_{BM} ($MCA_{BM} : F(3,84) = 0.9648, p > 0.05$). This statement cannot be confirmed for the MCA_{OPT} ($MCA_{OPT} : F(3,84) = 4.333, p < 0.01$), which is why the assumption of a consistent rating for the four trials can only be confirmed for the MCA_{BM}. By observing Figure 6.8, the last three ratings of MCA_{OPT} do not change significantly ($MCA_{OPT} : F(2,56) = 0.361, p > 0.05$) for which reason the ratings are subjectively assessed as sufficiently consistent.
Continuous Rating: The mean continuous rating of all participants for MCA_{BM} and MCA_{OPT} with the standard-error shows Figure 6.7. The overall mean values between MCA_{BM} ($\mu_{MCA_{BM}} = 2.5467$, $\sigma_{MCA_{BM}} = 1.9267$) and MCA_{OPT} ($\mu_{MCA_{OPT}} = 1.3347$, $\sigma_{MCA_{OPT}} = 1.1929$) also differs significantly ($t(28) = 7.2877$, $p < 0.001$).

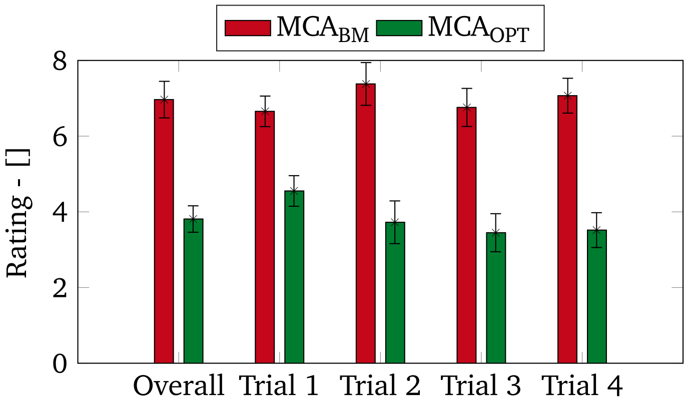


Figure 6.8: Mean PR for trails 1-4 with standard error.

6.6.3 Manoeuvre-Based Rating

The continuous rating is divided into different manoeuvres, listed in Section 6.4.3. Table 6.3 lists the mean values for the MCA_{BM} and MCA_{OPT} for each section which are checked for significance ($t - /z-$statistic). The differences between the mean values are significant for all values ($p < 0.001$) except for the „acc. procedure“ and „overland course 1 / 2“ sections. In the latter sections, there is a significant difference in motion incoherence between both algorithms. Table 6.4 checks variations between the four ratings per section and MCA ($F - /\chi^2-$statistic). If the ratings do not indicate any significant variations, a consistent rating between the four trials can be assumed.

6.6.4 Rating Model

In the first step, the weighting parameters $\tilde{\boldsymbol{W}}$ in equation (6.2) of RM_M are estimated for the OMIs $\boldsymbol{y}_{V,MCA} - \boldsymbol{y}_{S,MCA}$ and the mean ratings $R_{MCA_{BM}}/R_{MCA_{OPT}}$ for all participants which

Table 6.3: Mean values of the CR of MCA_{BM} and MCA_{OPT} within the sections. Largest values are plotted in bold. For each section the two mean values are checked for significant difference.

	mean value μ		
Section	MCA_{BM}	MCA_{OPT}	$t(28)-/z(28)$–stat.
Acc. procedure	0.8029	0.7475	$z = 0.7$ $p > 0.05$
Overland course 1	1.8262	1.2759	$z = 1.4463$ $p > 0.05$
Roundabout	3.3325	1.4353	$t = 8.1441$ $p < 0.001$
Overland course 2	1.8374	1.1382	$z = 1.6951$ $p > 0.05$
Sharp hilltop 1	2.8810	**1.9086**	$t = 4.5848$ $p < 0.001$
Sharp turn 1	**3.9073**	1.8207	$t = 6.5817$ $p < 0.001$
Sharp S-curves	3.3335	1.4503	$t = 7.0959$ $p < 0.001$
Sharp turn 2	2.5012	1.0977	$z = 3.7479$ $p < 0.001$
Sharp hilltop 2	2.3722	1.2553	$t = 4.8288$ $p < 0.001$
Dec. procedure	1.9297	0.9147	$t = 4.818$ $p < 0.001$

Table 6.4: Variations between the four ratings per section and MCA. It is assumed that there is no significant difference for $p > 0.05$ between the four ratings ($F(3,84)-/\chi^2(3)-$ stat.).

Section	MCA_{BM}	MCA_{OPT}
Acc. procedure	$\chi^2 = 1.8679$	$\chi^2 = 0.8588$
Overland course 1	$F = 1.0319$	$\chi^2 = 2.4201$
Roundabout	$F = 1.5809$	$F = 0.5917$
Overland course 2	$F = 1.4773$	$\chi^2 = 1.1912$
Sharp hilltop 1	$F = 2.7891^*$	$F = 2.3507$
Sharp turn 1	$F = 1.4352$	$\chi^2 = 8.7931^*$
Sharp S-curves	$F = 2.1041$	$F = 2.956^*$
Sharp turn 2	$F = 2.3438$	$F = 0.6744$
Sharp hilltop 2	$F = 1.1159$	$F = 3.46^*$
Dec. procedure	$F = 2.8125^*$	$F = 0.4241$

* significant differences between the four ratings ($p < 0.05$)

Table 6.5: $\tilde{W}$ for RM_{C} , $\text{RM}_{\text{M,BM}}$ and $\text{RM}_{\text{M,OPT}}$.

	Roll	Pitch	Yaw	X	Y	Z
RM_{C}	0.047	0.003	0.012	0.000	0.227	0.389
$\text{RM}_{\text{M,BM}}$	0.3822	57.0788	0.1703	0.8030	1.6844	0.3693
$\text{RM}_{\text{M,OPT}}$	31.1977	10^{-5}	10^{-5}	1.3744	0.6127	0.8272

are shown in Figure 6.7. Table 6.5 contrasts the estimated weighting parameters $\tilde{W}$ for the $\text{RM}_{\text{M,BM}}$, $\text{RM}_{\text{M,OPT}}$ and RM_{C}.

The dimension N of equation (6.1), introduced in Section 6.4.4, is estimated to $N_{\text{MCA}_{\text{BM}}} = 10$ and $N_{\text{MCA}_{\text{OPT}}} = 9$ for the MCA_{BM} and MCA_{OPT}, respectively. The maximum dimension is chosen to $N_{max} = 10$ with a sample time $t_{samp} = 0.2$ s to receive a workable model dimension. Higher values for N_{max} do not change the quality essentially. The offset parameter C of Figure 6.4 is set to 0. The values of $\tilde{W}$ between the models in Table 6.5 greatly vary which is a first indicator that the rating models are far from universal. Still, all three rating models are based on different OMIs resulting in varying weighting parameters. Applicability of the models to other OMIs is part of this Section.

In the following, the OMIs and the ratings of both MCAs are used to evaluate the proposed RM$_\text{M}$. Using the procedure described above, the parameters for both models, RM$_{\text{M,BM}}$ and

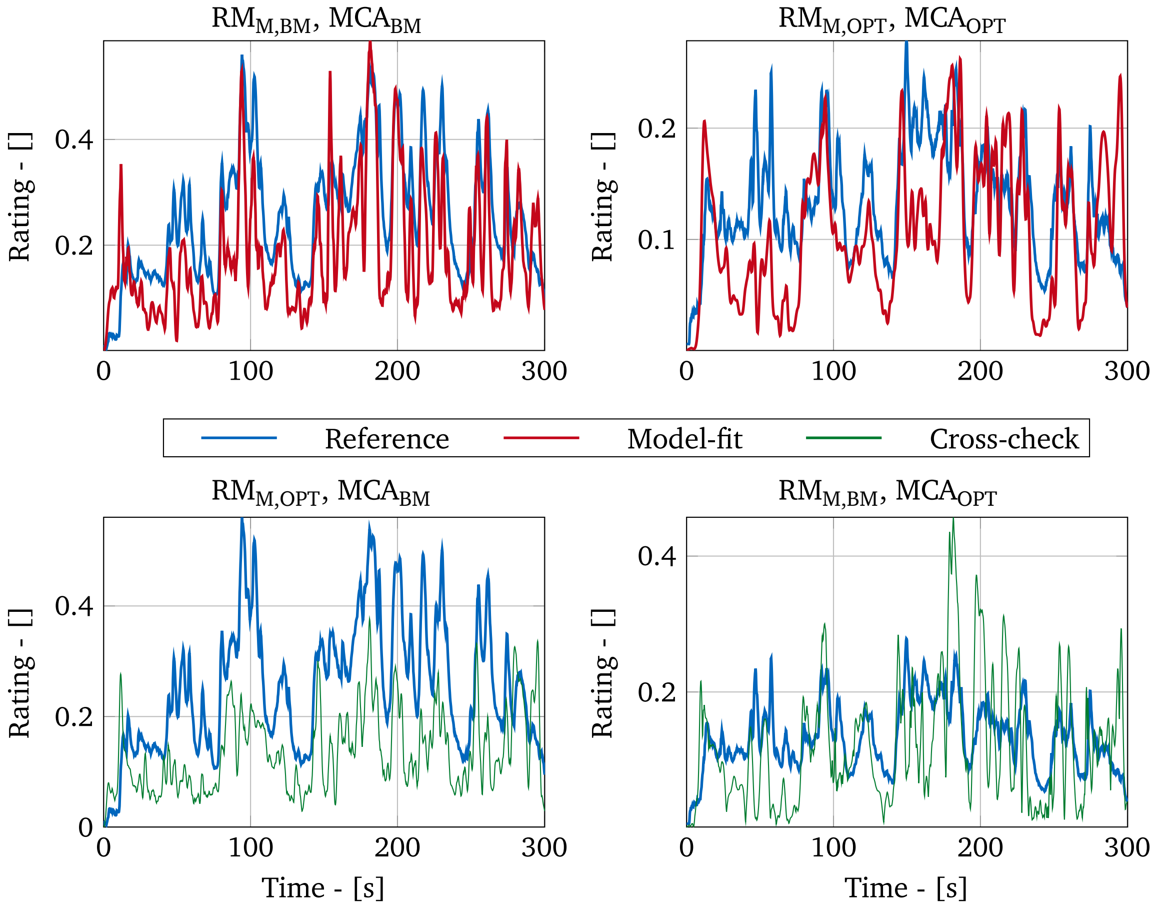


Figure 6.9: Reference Rating of the MCA$_{\text{BM}}$ and MCA$_{\text{OPT}}$. Upper plots: Modelled rating from the RM$_{\text{M,BM}}$ (model-fit) and RM$_{\text{M,OPT}}$ (model-fit). Lower plots: Modelled rating from the RM$_{\text{M,OPT}}$ (cross-check) and RM$_{\text{M,BM}}$ (cross-check).

RM$_{\text{M,OPT}}$, were estimated. The upper left plot of Figure 6.9 shows the resulting model-fit compared to the normalized reference rating for MCA$_{\text{BM}}$, which was obtained by dividing the measured rating by the maximum rating value (10). The upper right plot shows the results of the model-fit for RM$_{\text{M,OPT}}$ compared to the normalized reference rating for MCA$_{\text{OPT}}$. The results show that the introduced model approximates the measured rating to an appropriate degree.

In order to evaluate the adaptivity of the model to other OMIs, the parametrized models were also used for a cross-check, i.e. RM$_{\text{M,OPT}}$ was applied to the MCA$_{\text{BM}}$ data and vice-versa. The lower left plot of Figure 6.9 depicts the simulated rating using the RM$_{\text{M,OPT}}$ with the OMIs of the MCA$_{\text{BM}}$; the lower right plot of Figure 6.9 vice-versa. Thus, the generality of the proposed models for the considered manoeuvre can be investigated which is essential for the integration in a general cost function. The rms of the deviations between reference rating and the respective simulated rating for the OMIs of MCA$_{\text{BM}}$ and MCA$_{\text{OPT}}$ are listed in Table 6.6. Additionally, Table 6.6 includes the relative deviation r_{RM} between model-fit and cross-check results. Comparing the values in both columns of Table 6.6, deviations of the cross-check are larger compared to the deviations of the model-fit which is corroborated by

Table 6.6: Rms for model-fit (labelled with *) / cross-check to reference rating.

	OMIs $\mathrm{MCA_{BM}}$	OMIs $\mathrm{MCA_{OPT}}$
$\mathrm{RM_{M,BM}}$	0.1071^*	0.0798
$\mathrm{RM_{M,OPT}}$	0.1568	0.0598^*
r_{RM} [%]	46.41	33.44

the larger visual deviations for the cross-check results visible in Figure 6.9 and large relative deviations r_{RM}. This indicates that model quality deteriorates when the model is applied to simulator motions that are different from the ones that the model parameters were fitted on. Or more accurately phrased: quality was shown to drop when the model was used to predict the PMIs rating for a different MCA. It should be noted that the vehicle motion was the same in both cases and only the simulator motion (MCA output) varied. Although this result is not surprising, it does point out that care should be taken in the use of such models. If the model is to serve as (part of) an MCA cost function, the model's capacity to correctly predict the effect of changes in the MCA is obviously essential. How and by how much the cross-MCA-validity can be improved is still to be investigated.

6.7 Chapter Summary

This Chapter evaluates the optimization-based $\mathrm{MCA_{OPT}}$ in a comparison with a benchmark, optimization-based algorithm. To outline the potential of the $\mathrm{MCA_{OPT}}$, a time horizon of 3s with an ideal prediction of the vehicle motion is used. The algorithm is evaluated in an experiment with 35 participants. The continuous rating, introduced in [36, 37], is applied to obtain a time and position dependent rating. Results show a consistent rating for the participants with an average Cronbach's α of 0.77. The continuous rating is also compared to a generally accepted-, verbal overall rating in order to test the validity of the rating. The paper shows correlations between the overall rating and the continuous rating. Motion mismatches were rated higher for the benchmark algorithm which indicates the $\mathrm{MCA_{OPT}}$ has the potential to further improve the motion cueing quality. The ratings are used to assess a rating model introduced in [37]. As the stated model is not intended to be applied to other OMIs, the rating model is modified with new weighting parameters and an extended transfer function. A reasonable prediction of the reference rating can be achieved with the modified model $\mathrm{RM_M}$. However, when the model is used to predict the rating obtained for different MCA the quality decreases substantially. This indicates that the usage of rating models in an MCA objective function requires careful evaluation.

Chapter 7

Closed-Loop, MPC-Based MCA (MCA$_{\text{MPC}}$)

The content of the following Chapter have been published in [48].

In this chapter, a closed-loop, optimization-based MCA is proposed, showing the ability to find an optimal motion during an active drive for the nine DoFs driving simulator (cf. Figure 1.3). The structure of the algorithm is based on the MCA$_{\text{OPT}}$ which is evaluated in Chapter 6. As further developments and modifications are made to the MCA$_{\text{OPT}}$, the version described in this Chapter is called MCA$_{\text{MPC}}$. The basic idea of the algortihm is that an MPC-scheme solves the optimization problem at sampled time instances, minimizing discrepancies between desired and provided motion percepts. The optimization is solved online, making the MCA ready for real-time applications. It is the result of collaboration with Dimitar Hristakiev [88].

For evaluation of the MCA$_{\text{MPC}}$, an experimental study determines PMIs. The experiment includes two other MCAs which are classified as open-loop MCA$_{\text{OFF}}$ (cf. Chapter 4) and closed-loop MCA$_{\text{BM}}$ (cf. Chapter 6.3.2).

7.1 Contribution

One objective of this Chapter is to enable the online capability of the optimization based MCA$_{\text{MPC}}$. Methods are shown to reduce the computational efforts for solving the optimization problem of the proposed MCA$_{\text{MPC}}$ in a real time environment.

The second objective includes the evaluation of the MCA$_{\text{MPC}}$. The first research question intends to evaluate the motion cueing quality of the MCA$_{\text{MPC}}$ compared to a closed-loop, state-of-the-art MCA. The benchmark algorithm (MCA$_{\text{BM}}$) was provided by the simulator's manufacturer and uses optimization and prefiltering techniques to resolve the redundant DoFs. The MCA$_{\text{MPC}}$, by contrast, solves the DoFs in a single, constrained optimization problem which is embedded in an MPC-scheme. The comparison reveals whether the closed-loop structure of the approach, introduced in Chapter 5, has the potential to increase the motion cueing quality towards a closed-loop, state-of-the-art algorithm. The second research question aims to determine the loss of motion cueing quality of the MCA$_{\text{MPC}}$ compared to the open-loop, optimization-based MCA$_{\text{OFF}}$. For the MCA$_{\text{OFF}}$, a global optimal control problem is solved with an ideal prediction over the complete manoeuvre resulting in an assumed optimal trajectory for the considered manoeuvre. The MCA$_{\text{MPC}}$, by contrast, uses a simpler prediction: The current driving dynamics reference values at the current time step are kept constant over a finite time horizon. Assuming that the MCA$_{\text{OFF}}$ sets an upper benchmark, results reveal how much the motion cueing quality is influenced by the prediction strategy and the horizon length.

7.2 Extensions for Real-Time Applications

The basic structure of the MCA$_{\mathrm{MPC}}$ is introduced in Chapter 5, including:

1. the basic principals of MPC methods,
2. the kinematic structure of the redundant motion system,
3. the linearized prediction model and the linearized actuator constraints,
4. the formulation of the optimization problem and
5. the distinction to other optimization-based approaches.

In Chapter 6, an open-loop version (MCA$_{\mathrm{OPT}}$) was used in an experimental study. At that time of research, the MPC-based MCA was not yet ready for closed-loop executions and the MCA$_{\mathrm{OPT}}$ was used as a proof-of-concept. The algorithm runs sequentially and did not fulfil real-time requirements. The experimental study aimed at illustrating the potential of the open-loop version, compared to an open-loop, state-of-the-art MCA. To increase the pre-positioning capabilities, an ideal prediction strategy and a time horizon of 3 s were applied. The motion cueing quality was obtained with the continuous rating method, which was already applied in other experiments, such as [36, 37, 137]. In the conducted experiment, motion mismatches were rated higher for the benchmark algorithm. This indicates that the MCA$_{\mathrm{OPT}}$ has the potential to further improve the motion cueing quality in an open-loop version. However, two open questions remained:

1. Which modifications are necessary to achieve a closed-loop execution?
2. What is the influence on the motion cueing quality arising from the modifications (required to achieve closed-loop execution)?

The following Section describes the extensions of the MCA$_{\mathrm{OPT}}$ to the MCA$_{\mathrm{MPC}}$.

7.2.1 Optimization Problem

The optimization problem, described in equation (5.69), is reduced to

$$\begin{aligned}
&\min_{\Delta \boldsymbol{U}_S} \quad J[k] && \\
&s.t. \quad \Delta \boldsymbol{u}_{S,min} \leq \Delta \boldsymbol{u}_S[k,i] \leq \Delta \boldsymbol{u}_{S,max} && \text{for } i = 1 \ldots H_u \\
&\qquad\ \ \Delta \boldsymbol{u}_S[k,i] = 0 && \text{for } i = (H_u+1) \ldots H_p \\
&\qquad\ \ \boldsymbol{q}_{S,min} \leq \boldsymbol{q}_S[k,i] \leq \boldsymbol{q}_{S,max} && \text{for } i = 1 \ldots H_c
\end{aligned} \tag{7.1}$$

where the objective function J, defined in equation (5.37), remains unchanged. The constraints on velocity and acceleration level are neglected in order to reduce the computation time to allow real-time applications. Figure 5.9 and Figure B.1 (cf. Appendix B) show that the most critical constraints are on position level. Whereas, a relatively large safety margin is left for the actuator velocities and actuator accelerations. Still, the analysis is limited to one example and does not generally guarantee the workspace adherence on velocity and acceleration level.

For the prediction, the current driving dynamics motion percepts $\hat{\boldsymbol{y}}_V$ are kept constant. The discrete horizon steps are chosen to $H_p = 150$, $H_u = 10$ and $H_c = 30$. The hand-tuned weights are shown in Appendix D in Table D.3.

7.2.2 Parallelization

The MCA$_{\text{MPC}}$ is implemented in a C++ environment in Ubuntu, Linux and uses *qpOASES* [65] as an active-set QP-solver. Depending on the system's states, an active set-algorithm activates or deactivates constraints in the optimization leading to runtime variances in the optimization solver (see Appendix C.3). An increasing number of active-constraints leads to a higher dimension of the optimal control problem which can increase the computation time [13]. Usually, the implemented MPC-scheme sends the first control output $\Delta\boldsymbol{u}[k,1]$ to the motion system. Since the execution time for obtaining a solution for the optimal control problem varies, hard real-time deadlines cannot be guaranteed. Thus, in case a deadline is missed, the algorithm sends signals from the previous sequence $\Delta\boldsymbol{u}[k-1,i]$ until an updated control output is available.

The preparation and the calculation of the QP-problem is segmented into various threads which run in parallel. The optimization problem, formulated in equation (5.69), is transformed to a condensed form

$$\begin{aligned}
\min_{\Delta\boldsymbol{U}} \quad & \frac{1}{2}\Delta\boldsymbol{U}^T\boldsymbol{H}\,\Delta\boldsymbol{U} + \Delta\boldsymbol{U}^T\boldsymbol{g} \\
\text{s.t.} \quad & \boldsymbol{l_{bA_c}} \leq \boldsymbol{A}_c\,\Delta\boldsymbol{U} \leq \boldsymbol{u_{bA_c}} \\
& \boldsymbol{l_{bu}} \leq \Delta\boldsymbol{U} \leq \boldsymbol{u_{bu}}
\end{aligned} \tag{7.2}$$

which is required by the QP-solver. In equation (7.2), $\boldsymbol{H}$ and $\boldsymbol{g}$ are the Hessian matrix and gradient vector, respectively [13]. The constraint matrix $\boldsymbol{A}_c$, the lower bound vector $\boldsymbol{l_{bA_c}}$ and the upper bound vector $\boldsymbol{u_{bA_c}}$ contain the inequality constraints. The lower bound vector $\boldsymbol{l_{bu}}$ and the upper bound vector $\boldsymbol{u_{bu}}$ contain the inequality constraints of $\Delta\boldsymbol{U}$. Six parallel computation threads separate the calculation of:

1. the reference generation,
2. the matrix $\boldsymbol{H}$,
3. the vector $\boldsymbol{g}$,
4. the constraint matrices/vectors,
5. the prediction model update and
6. the QP-solver.

An additional parent thread manages the six child threads, refreshes the system with updated inputs and sends the outputs to the simulator.

Figure 7.1 schematically shows the synchronization strategy of the threads which are divided into two systems: A real-time system with hard feedback deadlines (driving simulation framework) and a real-time system with soft real-time deadlines (MCA$_{\text{MPC}}$) [158].

The driving simulation framework (upper system in Figure 7.1) constitutes an interface with the MCA$_{\text{MPC}}$ by updating the vehicle motions (inputs for the MCA$_{\text{MPC}}$) and sending outputs to the simulator (outputs of the MCA$_{\text{MPC}}$). The framework runs with a sampling time $T_{samp} = 10\,\text{ms}$ under hard feedback deadlines with the discrete time steps k. Missing to send an output to the simulator within the hard feedback deadline can lead to a failure of the system [158].

The parallel calculation of the threads described above is synchronised with the simulation

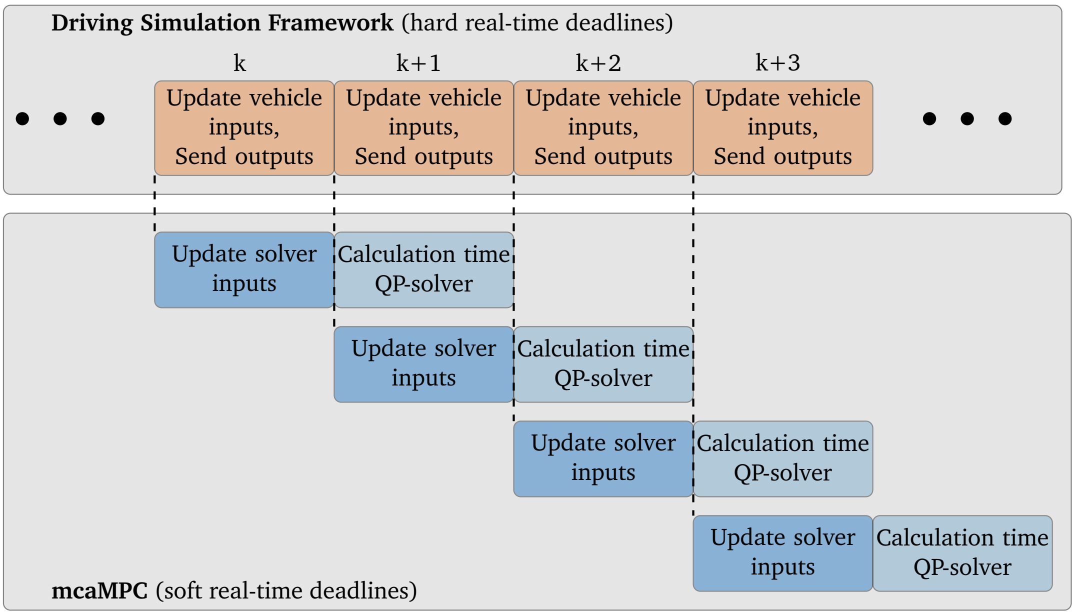

Driving Simulation Framework (hard real-time deadlines)
k
k+1
k+2
k+3
Update vehicle inputs, Send outputs
Update vehicle inputs, Send outputs
Update vehicle inputs, Send outputs
Update vehicle inputs, Send outputs
Update solver inputs
Calculation time QP-solver
Update solver inputs
Calculation time QP-solver
Update solver inputs
Calculation time QP-solver
Update solver inputs
Calculation time QP-solver
mcaMPC (soft real-time deadlines)

Figure 7.1: Schematically synchronization strategy between the driving simulation framework and the MCA$_{\mathrm{MPC}}$.

framework. A lot of effort is put into merging a hard feedback deadline system with a soft feedback deadline system. The MCA$_{\mathrm{MPC}}$ exhibits a soft feedback deadline as the computation times of the QP-solver vary, and thus cannot guarantee hard feedback time. Two tasks are distinguished in the MCA$_{\mathrm{MPC}}$: First, updating the inputs for the QP-solver ("Update solver inputs" in Figure 7.1), including: the reference generation, the Hessian matrix $\boldsymbol{H}$, the gradient vector $\boldsymbol{g}$, the constraint matrices/vectors, the prediction model. Executing these steps is computationally expensive and can take a few milliseconds which is especially caused by building the high dimensional terms $\boldsymbol{H}$ and $\boldsymbol{g}$. The dimension of these terms depends on the chosen number of discrete time steps H_p for the prediction horizon as shown in equation (5.44). The discrete time steps H_p for the prediction horizon is chosen to obtain computation times which are considerably less than 10 ms for the task "Update solver inputs". In the second task, the QP-solver is running to compute the optimal control problem ("Calculation time QP-solver" in Figure 7.1).

Thus, not only the varying computation times of the QP-solver but also the computational efforts for updating the QP-solver inputs are considered in the synchronization. For the chosen prediction lengths, a sequential execution of these tasks is hardly possible in real-time. Thus, a parallelization technique is used by running the QP-solver in the upcoming discrete time step $k+1$ as depicted in Figure 7.1. This method generally leads to a system feedback latency of at least two discrete time steps. Though, in case the computation time of the QP-solver takes less than 10 ms, a simulator output at every discrete time step k is available. The latency depends on the computation time of the QP-solver and is considered in the system by predicting the QP-solver's computation time. Otherwise, the induced system delay could lead to an instability of the MPC-scheme as the delay is not implied in the prediction model [39]. The calculated system states ξ of the prediction model in equation (5.44) are updated with the actual duration of the task "Calculation time QP-solver" to prevent a divergence between the states ξ of the prediction model and the states of the real plant, which can cause instability. The discrete time steps H_p, H_c and H_u for the prediction horizons are chosen to obtain calculation times for the solver which are less than 10 ms for most of the runs.

Figure 7.2 shows the number of sent optimal control solutions per optimization step for an online drive. In case the number of optimal solutions is one, only the first optimal control output is sent to the simulator hardware and the hard cycle time requirements of 10 ms are ensured. For higher numbers, also a quasi-optimal solution is sent, as the current optimal control step is still running and the 10 ms update cycle time cannot be fulfilled. For the considered manoeuvre (same as in the experiment), the runtime requirements are ensured in 99.73% (Intel i7-6820HQ CPU 2.70GHz, 8 cores).

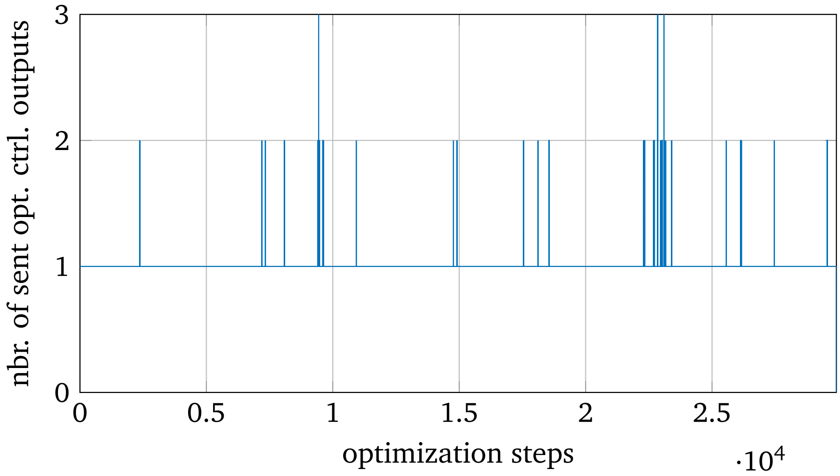


Figure 7.2: Number of sent optimal control outputs for each optimization step during an online drive.

7.2.3 Handling Infeasibility

Due to large disturbances in the objective function or different behaviours between the real system and the prediction model, the optimal control problem can become infeasible. This means that the system cannot be kept within the defined constraints. Excessively large driver demands (large values of $\hat{\boldsymbol{y}}_V$ in equation (5.37)) are a typical reason when the optimizer may be faced with an infeasible problem, for example by executing an extremely strong deceleration manoeuvre. The growing deviation between desired motions and provided simulator motions increases the objective function J extensively. The optimizer calculates outputs with high accelerations and tilting angles. As a consequence, some of the system's actuators are brought to their limits. There are many more possibilities for the occurrence of these issues which are very difficult to anticipate [106]. In this Section, three features of the MCA$_{\text{MPC}}$ are presented to circumvent these problems:

1. *Slack variables* $\boldsymbol{\eta}$ are additional virtual control inputs, which are added to the optimal control problem in order to relax the constraints. The optimal control problem is modified to

$$\begin{aligned} \min_{\Delta \boldsymbol{U}, \boldsymbol{\eta}} \; & J^* = J + ||\boldsymbol{\eta}||_{\boldsymbol{P}}^2 \\ \text{s.t.} \; & \boldsymbol{l}_{\boldsymbol{b}A_c} + \boldsymbol{\eta} \leq \boldsymbol{A}_c \Delta \boldsymbol{U} \leq \boldsymbol{u}_{\boldsymbol{b}A_c} + \boldsymbol{\eta} \\ & \boldsymbol{l}_{\boldsymbol{bu}} \leq \Delta \boldsymbol{U} \leq \boldsymbol{u}_{\boldsymbol{bu}} \\ & \boldsymbol{0} \leq \boldsymbol{\eta} \leq \boldsymbol{l}_{b\eta} \end{aligned} \tag{7.3}$$

   where the diagonally structured weighting matrix $\boldsymbol{P}$ contains very large values. In this case, $\boldsymbol{\eta}$ is mainly penalized when the constraints are violated, otherwise the optimizer has the incentive to keep them low [106]. In MCA$_{\text{MPC}}$, slack variables are used for the actuator strokes and the yaw angle of the tripod. In order to stay away from the physical actuator bounds, the actuator limitations are modified for the control problem by introducing some safety margins.

2. *Reference limitation:* The reference signal $\boldsymbol{y}_V$ can contain large values especially for longitudinal/lateral accelerations and yaw velocities. The cueing for long-lasting accelerations and angular velocities is physically limited by the workspace envelope of the motion system. Thus, $\boldsymbol{y}_V$ is bounded to an upper and lower limit. For the translational accelerations and the yaw velocity, the bound is set to $\pm 2\text{m/s}^2$ and to $\pm 20°/\text{s}$, respectively.

3. *Parameter tuning:* Generally, longer time horizons H_p, H_u and H_c lead to a more preventive behaviour of the workspace management. However, a constant, long-lasting reference can lead to predictions which are not feasible as the prediction strategy is only a valid approximation for short time intervals. The relation between the weighting parameters in the matrices $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{T}$ (cf. equation (5.37)) influences the workspace management. Increasing weighting values in the matrices $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{T}$ enhances the neutral push which leads to more conservative motions, and thus reduces the risk for obtaining infeasible solutions.

7.3 Experimental Evaluation

The experiment compares three MCAs:

1. Closed-loop MCA_{MPC} (cf. Chapter 7.2)
   The computations were done offline and applied in open-loop tests in order to be able to compare the algorithm in an objective manner and use the CR.

2. Open-loop MCA_{OFF} (cf. Chapter 4)
   The optimization is performed over the complete, pre-defined manoeuvre, and hence uses an ideal reference signal. Depending on the chosen weighting parameters and the used simulator model, an optimal platform trajectory is obtained. The hand-tuned weights are shown in Appendix D in Table D.5.

3. Closed-loop MCA_{BM} (cf. Chapter 6.3.2)
   The computations were done offline and applied in open-loop tests.

The following research questions are formulated for the experiment:

1. Does the CR method show a reliable and valid rating between the three MCAs?

2. Does the MCA_{OFF} exhibit the lowest PMIs?

3. Does the MCA_{MPC} have a higher motion cueing quality compared to the MCA_{BM}?

4. Which motion channels are mainly responsible for the PMIs?

The experiment was conducted in the nine DoFs simulator at the BMW Group with 30 participants (4 females) between the age of 18-63 (μ: 35.9 years; σ: 13.2). The participants were employees from the BMW Group. Nine participants had never experienced a driving simulator. For the remaining participants, the driving simulator experience ranged between:

1-3 times (9), 4-6 times (4), and more than 6 times (8). All the participants have a driving licence. The average mileage is approx. 19 667 km per year (σ: 11 342 km). Three participants aborted the experiment because of motion sickness.

7.3.1 Rating Procedure

The measurement procedure and the structure of the experiment follows the procedure described in Chapter 6. Participants rate the PMI continuously using a rotational knob located at the central console of the mockup. Consequently, a time- and position-dependent rating is obtained. The PMI is rated using an 11-point scale that is displayed by a bar which length and colour depends on the current rate value. The bar is shown in front of the driver inside the virtual world. A rating of 0 represents no PMI ("real driving sensation") whereas a rating of 10 denotes a high PMI ("large motion mismatch"). The maximum rating was calibrated in a training session before the experiment where several manoeuvres exhibited large OMIs. The participants were instructed such that a rating of 10 was reached at least once in the training session, in order to obtain a reference for the CR. The manoeuvre of the training sessions differs from the manoeuvre used in the subsequent experiment. The three dropouts already felt motion sick after the training session.

Following the training phase, the experiment commenced, in which the same pre-recorded driving manoeuvre as in the experimental study, described in Chapter 6.4.3, was used. The manoeuvre is a representative simulator drive, consisting of overland and inner-city elements with a duration of 5 min. The track is divided into ten sections listed in Table 6.1.

Participants were asked to rate PMIs using the CR. The track was repeated three times for each of the three MCAs. Ordering of the nine simulation trials is randomly permuted during the experiment. An additional PR at the end of each CR is used to evaluate the reliability and the validity of the CR [37]. The PR uses the same relative scale as the CR. To ensure comparability, the MCA_{BM}, MCA_{MPC} and MCA_{OFF} are tested for the same pre-recorded driving manoeuvre. The driver can completely concentrate on the rating in the offline simulation as driver inputs are not required. After each rating, participants were allowed a short break of two minutes before the next rating started. The total experiment for a participant lasted about 1.5 h.

7.3.2 Independent and Dependent Variables

Independent variable in the experiment is the MCA (three levels: MCA_{BM}, MCA_{MPC}, MCA_{OFF}). Since the CR is not possible during a closed-loop drive and an open-loop MCA is part of the comparison, the simulator motion with the MCA_{MPC} and MCA_{BM} was precomputed. The MCA_{MPC} was computed in a real-time environment, simulating an online driving session. The algorithm uses the vehicle inputs of the pre-recorded manoeuvre and saves the simulator outputs for usage in the experimental open-loop session. The MCAs are tuned to exploit the motion system's workspace as much as possible. The three algorithms are embedded in a within-subject design. By these means, the algorithms are compared, in a three-times randomly ordered comparison.

Dependent variables are the CR and the PR.

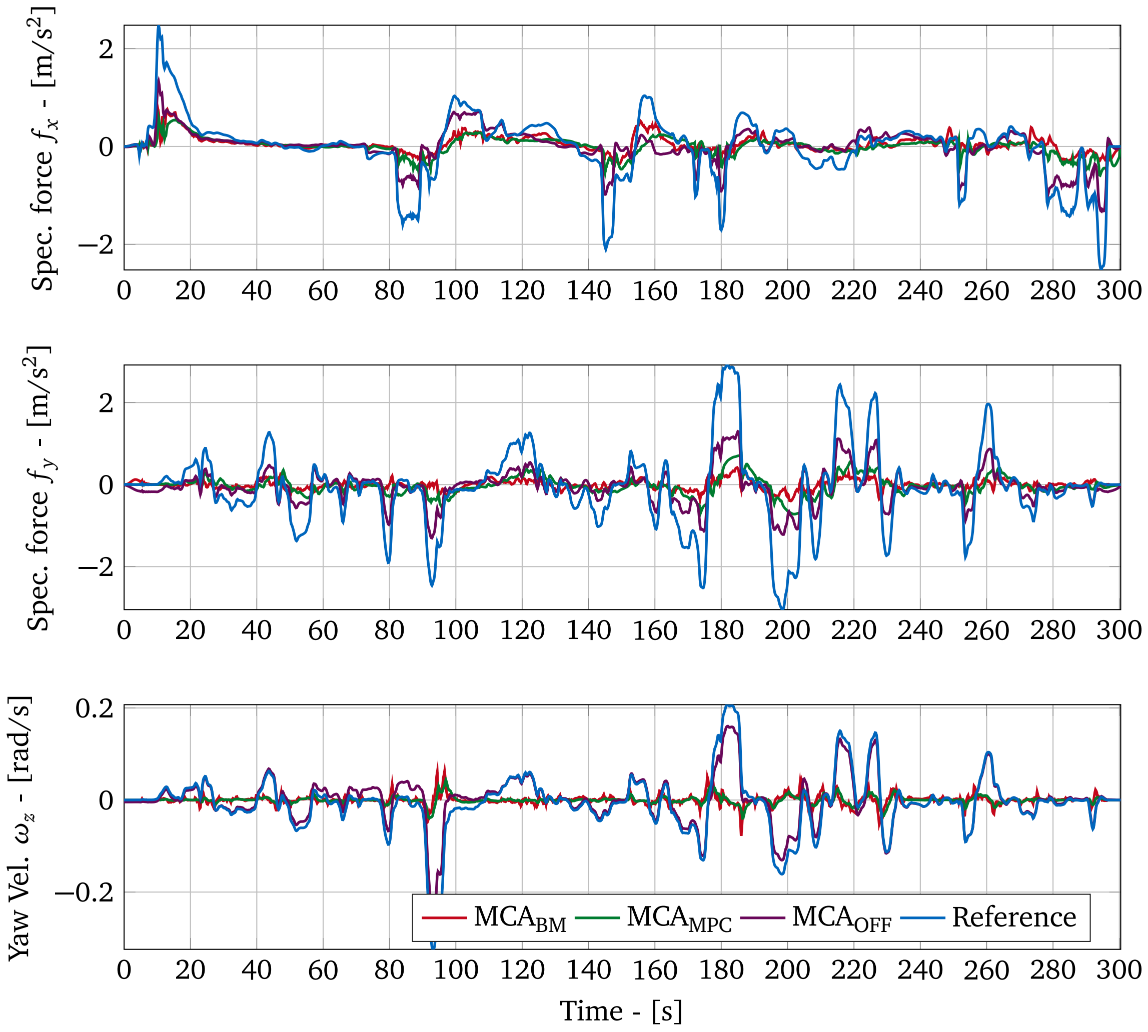


Figure 7.3: Resulting specific forces f_x, f_y and angular velocity ω_z for the MCA$_{\mathrm{BM}}$, MCA$_{\mathrm{MPC}}$ and MCA$_{\mathrm{OFF}}$ compared to the reference motions of the virtual vehicle.

7.4 Objective Motion Incongruencies

Figure 7.3 compares the resulting physical motions of the three MCAs and the reference motions resulting from the vehicle model, for longitudinal accelerations, lateral accelerations and yaw velocities as these were the main motion channels in this experiment. The OMIs of the three MCAs are determined by the rms over the manoeuvre and are listed in Table 7.1. The results show that the rms-values are smallest for MCA$_{\mathrm{OFF}}$ for each motion channel. This observation is not surprising as the prediction strategy and the prediction length is ideal for the MCA$_{\mathrm{OFF}}$. That is why, in this work, the MCA$_{\mathrm{OFF}}$ is taken as a reference for the other two MCAs. Comparing the online-capable MCAs, the rms-values of the MCA$_{\mathrm{MPC}}$ are smaller to the MCA$_{\mathrm{BM}}$ except for the translational z-direction. In order to evaluate the deviations of the two online-capable MCAs to the reference MCA$_{\mathrm{OFF}}$, the relative deviations

$$\tau_{\mathrm{OMI}_k} = \frac{\mathrm{rms}(\mathrm{OMI}_k) - \mathrm{rms}(\mathrm{OMI}_{\mathrm{MCA}_{\mathrm{OFF}}})}{\mathrm{rms}(\mathrm{OMI}_{\mathrm{MCA}_{\mathrm{OFF}}})} \quad [\%] \tag{7.4}$$

of the rms-OMIs of MCA$_{\mathrm{OFF}}$ and $k = \{\mathrm{MCA}_{\mathrm{MPC}}, \mathrm{MCA}_{\mathrm{BM}}\}$ for each motion channel are added to Table 7.1. The τ_{OMI_k} values show that the deviations are highest for the longitudinal acceleration, the lateral acceleration and the yaw velocity for both MCA$_{\mathrm{BM}}$ and MCA$_{\mathrm{MPC}}$.

Table 7.1: Rms of the OMIs for $\mathrm{MCA_{BM}}$, $\mathrm{MCA_{MPC}}$ and $\mathrm{MCA_{OFF}}$ and the relative deviations to the reference $\mathrm{MCA_{OFF}}$. The largest values are plotted in bold.

	Trans. acc. - [m/s^2]			Ang. Vel. [rad/s]		
	x	y	z	x	y	z
$\mathrm{MCA_{BM}}$	0.5191	**0.88**	0.134	0.017	0.015	**0.065**
$\mathrm{MCA_{MPC}}$	0.500	**0.791**	0.144	0.013	0.010	**0.062**
$\mathrm{MCA_{OFF}}$	0.346	**0.560**	0.110	0.014	0.009	**0.018**
$\tau_{\mathrm{OMI_{MCA_{BM}}}}$	50.1 %	**57.8** %	22.1 %	21.2 %	58.4 %	**261.7** %
$\tau_{\mathrm{OMI_{MCA_{MPC}}}}$	**44.6** %	41.2 %	30.3 %	5.9 %	3.9 %	**244.9** %

7.5 Experimental Results

The following Section firstly examines the reliability and the validity of the ratings, before analysing the ratings of the PMI. To check the central tendency (significance of mean-values) between three or more dependent samples, a repeated measures ANOVA (test statistic = F) is tested if Mauchly's sphericity test is positive ($p > 0.05$). Otherwise the Friedman-test (test statistic = χ^2) is used. Definitions for the mean value μ, the standard deviation σ and the standard error σ^* can be found in Section 6.6 in equations (6.4)-(6.6).

7.5.1 Reliability and Validity

Reliability is a measure of the internal consistency and the interrelatedness of ratings [18]. In this experiment, reliability is estimated by the interitem correlation. Therefore, for each participant, the Pearson correlation matrix between the three trails of each independent variable ($\mathrm{MCA_{BM}}$, $\mathrm{MCA_{MPC}}$ and $\mathrm{MCA_{OFF}}$) is calculated. The mean value of the side diagonals is estimated as they indicate the linear relationship between the trails. In order to obtain a single value for the reliability criterion, the average of the mean correlation values of the three independent variables is used. Figure 7.4 shows the correlation values for 27 participants of the experiment (3 participants aborted the experiment). Based on the suggestion in [35], a critical value for the interitem correlation of 0.15 is assumed for the experiment. Participants with a interitem correlation between $0.15 - 0.2$ are individually analysed for the rating's consistency by visual examination of the data. Generally, excluding ratings from the analysis reduces the statistical power of the experiment. Based on the examination, the three critical participants ($\mathrm{ID_5}$, $\mathrm{ID_{15}}$, $\mathrm{ID_{17}}$) are included in the evaluation (cf. orange markers in Figure 7.4) and one participant ($\mathrm{ID_{14}}$) is excluded from the study (cf. red marker Figure 7.4) because of a too low correlation value.

In order to analyse whether the specification of the experiment is measured (validity), the continuous rating is compared to the generally accepted PR method [37].

Figure 7.5 shows the course of the mean ratings for $\mathrm{MCA_{BM}}$, $\mathrm{MCA_{MPC}}$ and $\mathrm{MCA_{OFF}}$ over the manoeuvre for CR and for PR (dashed lines). [37] used the Pearson correlation to show a significant linear relationship between the mean PR and the maximum value of the CR. In the experiment, we also identify a correlation between those values for the $\mathrm{MCA_{MPC}}$ and $\mathrm{MCA_{OFF}}$ as depicted in Table 7.2. There is an additional, significant linear relationship for deviations between the MCAs for each rating system. These correlations are also included in Table 7.2. The three MCAs yield three differences using the maximum CR and the PR. The non-zero correlations confirm the assumption that in this experiment the CR measures the motion cueing

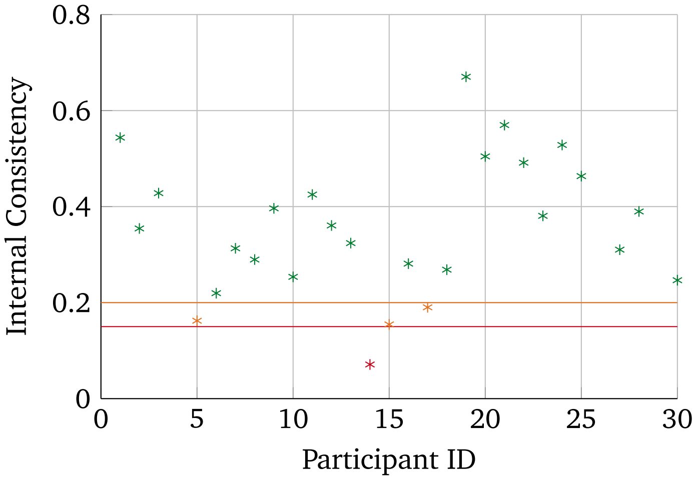


Figure 7.4: Internal consistency of the ratings per participant.

quality.

Table 7.2: Pearson correlations between the CR and the PR to analyse the validity of the CR.

MCA_{BM} $r = 0.29$, $p < 0.20$	MCA_{MPC} $r = 0.43$, $p < 0.05$	MCA_{OFF} $r = 0.47$, $p < 0.05$
MCA_{BM} / MCA_{MPC} $r = 0.44$, $p < 0.05$	MCA_{MPC} / MCA_{OFF} $r = 0.33$, $p < 0.15$	MCA_{BM} / MCA_{OFF} $r = 0.32$, $p < 0.15$

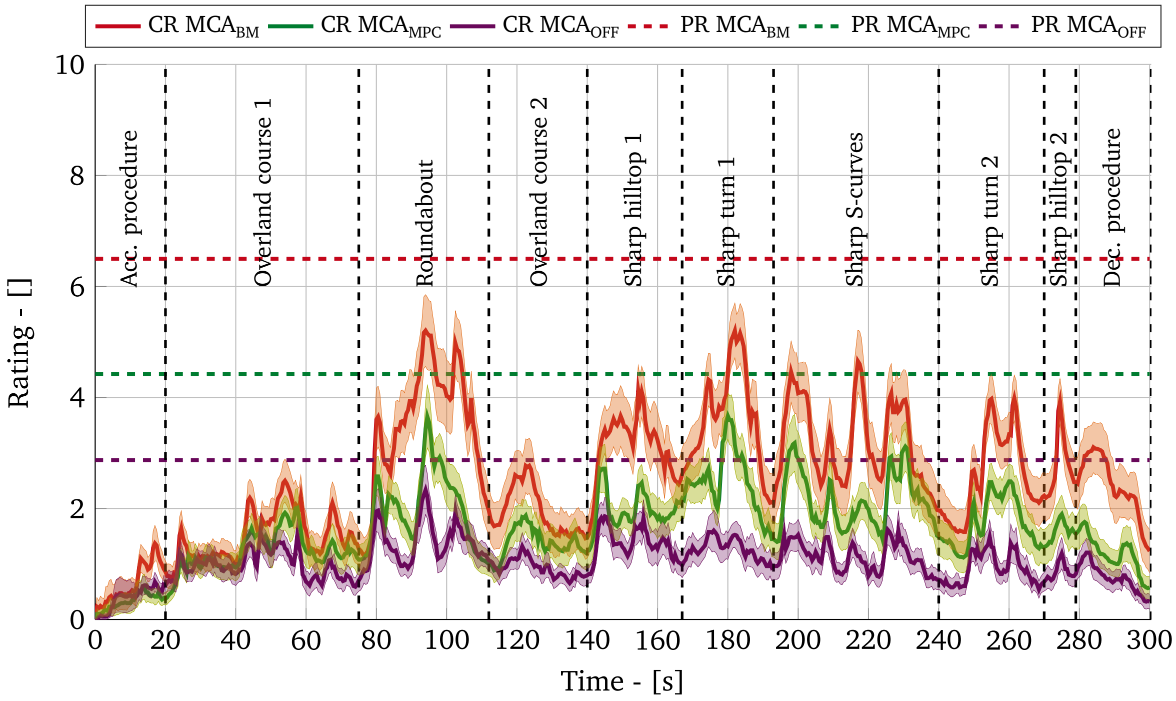


Figure 7.5: Mean CR with standard error (transparent area) and mean PR for the MCA_{BM}, MCA_{MPC} and MCA_{OFF} over the complete manoeuvre. Dashed lines separate the manoeuvre into the 10 sections as defined in Section 6.4.3.

7.5.2 Overall Rating

In the following, the statistics of the PR and the CR are shown, whereas a higher value means large PMI, hence worse motion cueing quality.

Post-hoc Rating: Figure 7.6 shows a significant difference ($\chi^2(2) = 44.06, p < 0.001$) between the mean PR for MCA_{BM} ($\mu_{\text{MCA}_{\text{BM}}} = 6.5$; $\sigma^*_{\text{MCA}_{\text{BM}}} = 0.47$), MCA_{MPC} ($\mu_{\text{MCA}_{\text{MPC}}} = 4.42$; $\sigma^*_{\text{MCA}_{\text{MPC}}} = 0.36$) and MCA_{OFF} ($\mu_{\text{MCA}_{\text{OFF}}} = 2.87$; $\sigma^*_{\text{MCA}_{\text{OFF}}} = 0.23$). The MCA_{MPC} reveals less PMI than the MCA_{BM}. The MCA_{OFF} exhibits the highest motion cueing quality of the three MCAs. This suggests that improving the prediction quality and the prediction length of MCA_{MPC} could further increase its motion cueing quality. The mean value of the PR does not change significantly for the three trials of the MCAs as apparent in Figure 7.6 ($\text{MCA}_{\text{BM}} : F(2,50) = 1.32, p > 0.05$, $\text{MCA}_{\text{MPC}} : F(2,50) = 1.05, p > 0.05$, $\text{MCA}_{\text{OFF}} : F(2,50) = 0.76, p > 0.05$).

Continuous Rating: The mean continuous rating of all participants for the three MCAs with the standard-error shows Figure 7.5. The overall mean values between MCA_{BM} ($\mu_{\text{MCA}_{\text{BM}}} = 2.53$, $\sigma_{\text{MCA}_{\text{BM}}} = 2.23$), MCA_{MPC} ($\mu_{\text{MCA}_{\text{MPC}}} = 1.65$, $\sigma_{\text{MCA}_{\text{MPC}}} = 1.55$) and MCA_{OFF} ($\mu_{\text{MCA}_{\text{OFF}}} = 1.04$, $\sigma_{\text{MCA}_{\text{OFF}}} = 1.16$) also differs significantly ($\chi^2(2) = 37.23$, $p < 0.001$). Considering the results of the CR and the PR, the MCA_{MPC} has the potential to further improve the motion cueing quality compared to a state-of-the-art MCA.

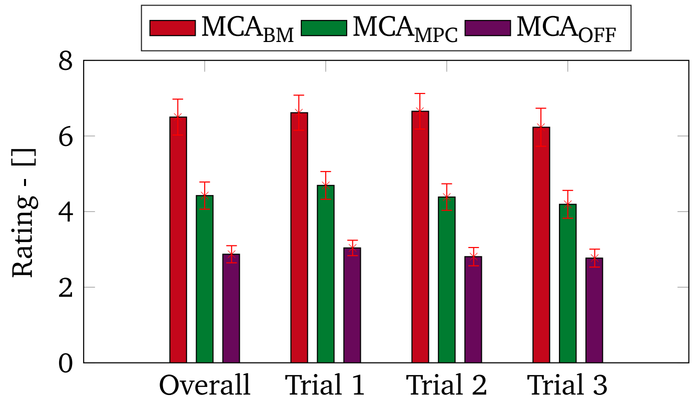


Figure 7.6: Overall mean PR and mean PR for trails 1-3 with standard error.

7.5.3 Manoeuvre-Based Rating

The continuous rating enables a manoeuvre based evaluation. Therefore, the track is divided into manoeuvres as listed in Section 6.4.3. Table 7.3 lists the mean values for the MCA_{BM}, MCA_{MPC} and MCA_{OFF} for each section which differ significantly ($F - / \chi^2$-statistic) except for the overland course 2. This section mainly consists of a straight road with long lasting curves, resulting in low/medium driver demands. Almost each section (except acc. procedure), MCA_{OFF} exhibits the lowest mean values, followed by the MCA_{MPC}. The MCA_{BM} reveals the highest ratings in each section.

Table 7.3 checks variations between the three ratings per section and MCA ($F-/\chi^2-$statistic). If the ratings do not indicate any significant variations, a consistent rating between the three trials can be assumed. The consistency is shown for each section of the three MCAs, except for two sections of the MCA_{OFF}.

Table 7.3: Mean values of the CR of MCA$_{\text{BM}}$, MCA$_{\text{MPC}}$, and MCA$_{\text{OFF}}$ within the sections. Largest values are plotted in bold. For each section the three mean values are checked for significant difference.

Section	mean value μ MCA$_{\text{BM}}$	MCA$_{\text{MPC}}$	MCA$_{\text{OFF}}$	$F(2,50)-/$ $\chi^2(2)$–stat.
Acc. procedure	0.65	0.33	0.39	$\chi^2 = 13.03^*$
Overland c. 1	1.50	1.21	0.98	$F = 7.02^*$
Roundabout	3.51	2.00	**1.39**	$\chi^2 = 34.85^*$
Overland c. 2	1.94	1.38	0.93	$\chi^2 = 5.41^{***}$
Sharp hilltop 1	3.11	1.97	1.38	$\chi^2 = 31.00^*$
Sharp turn 1	**3.62**	**2.47**	1.27	$\chi^2 = 30.69^*$
Sharp S-curves	3.19	2.14	1.13	$\chi^2 = 23.38^*$
Sharp turn 2	2.58	1.73	0.91	$\chi^2 = 40.69^*$
Sharp hilltop 2	2.79	1.59	0.87	$\chi^2 = 33.77^*$
Dec. procedure	2.40	1.18	0.74	$\chi^2 = 21.55^*$

$^*(p < 0.01)$, $^{**}(p < 0.05)$, $^{***}(p > 0.05)$

Table 7.4: Variations between the three ratings per section and MCA. It is assumed that there is no significant difference for $p > 0.05$ between the three ratings ($F(2,50)-/\chi^2(2)-$ stat.).

Section	MCA$_{\text{BM}}$	MCA$_{\text{MPC}}$	MCA$_{\text{OFF}}$
Acc. procedure	$\chi^2 = 1.14$	$\chi^2 = 2.34$	$F = 1.39$
Overland c. 1	$F = 0.04$	$F = 1.46$	$\chi^2 = 1.46$
Roundabout	$F = 0.35$	$\chi^2 = 0.69$	$F = 1.03$
Overland c. 2	$F = 1.02$	$F = 1.67$	$\chi^2 = 3.2$
Sharp hilltop 1	$F = 0.20$	$F = 2.37$	$\chi^2 = 0.79$
Sharp turn 1	$F = 0.04$	$F = 1.62$	$F = 1.14$
Sharp S-curves	$F = 0.10$	$\chi^2 = 0.92$	$F = 0.68$
Sharp turn 2	$F = 0.48$	$F = 0.57$	$\chi^2 = 6.32^*$
Sharp hilltop 2	$F = 0.31$	$\chi^2 = 1.09$	$F = 4.11^*$
Dec. procedure	$\chi^2 = 2.39$	$\chi^2 = 4.54$	$\chi^2 = 4.72$

* significant differences between the three ratings ($p < 0.05$)

7.5.4 Correlations between OMIs / PMIs (Perceived Motion Incongruencies)

In order to determine which OMI have the greatest impact on the resulting rating in Figure 7.5 for each MCA, correlations between the different motion channels of the OMI and the mean CR are estimated. The OMI are split into six motion channels: Three specific forces f and three rotational angular velocities $\boldsymbol{\omega}$ which are acting at the cockpit reference point of the driver. The correlation value r is estimated for each specific manoeuvre outlined in Section 6.4.3 (cf. Table 6.1), using the OMIs of the respective MCA and the respective mean CR. In order to make general statements, the specific manoeuvres are grouped according to the main motion channels in the manoeuvre. As described in Section 7.4, the longitudinal and lateral accelerations as well as the yaw angular velocity are the main motion channels in the study. Thus, the specific manoeuvres are classified according to these channels and are listed in Table 7.6. *Group 1* includes accelerations and decelerations in the longitudinal direction. Lateral accelerations and yaw angular velocities are combined in *Group 2* as both motion channels occur in combination (e.g. turns and curves).

Table 7.5: Correlations r between OMI and mean ratings and maximum OMI ε_{OMI} depending on group 1 and group 2 (cf. Table 7.6) for the six motion channels. Largest and second largest r for each motion channel are plotted in bold.

	Group 1: Longitudinal acceleration manoeuvres					
	MCA$_{\text{BM}}$		MCA$_{\text{MPC}}$		MCA$_{\text{OFF}}$	
	r	ε_{OMI}	r	ε_{OMI}	r	ε_{OMI}
	[-]	[m/s^2]	[-]	[m/s^2]	[-]	[m/s^2]
f_x	**0.34**	**2.314**	**0.50**	**2.045**	0.44	1.199
f_y	0.12	1.893	0.39	1.64	0.21	0.941
f_z	**0.44**	**0.401**	0.40	0.399	**0.57**	**0.29**
	[-]	[rad/s]	[-]	[rad/s]	[-]	[rad/s]
ω_x	−0.08	0.036	−0.10	0.029	−0.06	0.023
ω_y	0.22	0.061	0.22	0.029	0.29	0.023
ω_z	0.24	0.095	**0.47**	**0.087**	**0.45**	**0.037**
	Group 2: Lateral acceleration and yaw angular velocity manoeuvres					
	MCA$_{\text{BM}}$		MCA$_{\text{MPC}}$		MCA$_{\text{OFF}}$	
	r	ε_{OMI}	r	ε_{OMI}	r	ε_{OMI}
	[-]	[m/s^2]	[-]	[m/s^2]	[-]	[m/s^2]
f_x	0.02	1.519	0.11	1.590	0.133	0.795
f_y	**0.74**	**2.930**	**0.63**	**2.626**	**0.70**	**1.838**
f_z	−0.1	0.698	−0.01	0.751	0.07	0.431
	[-]	[rad/s]	[-]	[rad/s]	[-]	[rad/s]
ω_x	0.09	0.067	0.18	0.052	0.31	0.062
ω_y	−0.04	0.108	0.27	0.041	0.23	0.039
ω_z	**0.82**	**0.33**	**0.71**	**0.304**	**0.43**	**0.067**

The maximum OMI ε_{OMI} for each group are shown in order to validate the plausibility of the correlation factor r. Table 7.5 lists the correlation factors r and the maximum OMI ε_{OMI} for each motion channel in relation to the specified groups in Table 7.6. The significant values are estimated for the three MCAs. The largest and second largest r of the six motion channels are plotted in bold.

Considering Group 1, the first thing to notice is the high correlations in f_x for all three MCAs. These relations are plausible as this channel is mainly responsible for the perception of longitudinal accelerations. The large correlations in f_z for all three MCAs in Group 1 do not match with the relatively smaller ones of Group 2, since we do not assume considerable deviations between both groups in f_z. With regard to the maximum OMI ε_{OMI}, which are even larger for Group 2, we assume that the correlations between the OMI in f_z and the CR are not reasonable. The correlations of Group 1 in ω_z for the MCA$_{\text{MPC}}$ and MCA$_{\text{OFF}}$ exhibit relatively large values although the yaw velocity is not a main motion channel for longitudinal accelerations. Comparing the maximum OMI ε_{OMI} of Group 1 and Group 2, the values are considerably greater for Group 2. This deviation is plausible as ω_z constitutes a main motion channel of Group 2. Thus, the high correlations of Group 1 in ω_z are questionable.

Evaluating the results of Group 2 indicates that correlations are greatest in ω_z and in f_y for

the three MCAs. This analysis is not surprising, as these motion channels are primarily active for the considered manoeuvres. These observations correlate with the ε_{OMI} values in ω_z and in f_y.
To sum up, analysing the correlations between OMI and PMI is an indicator to determine the dominant OMI for the CR method. Still, a bare comparison between the correlation values can lead to misleading conclusions.

Table 7.6: Classification of the specific manoeuvres defined in Section 6.4.3.

Group 1: (Long. acc.)	Acc. procedure, Sharp hilltop 1, Dec. procedure
Group 2: (Lat. acc./yaw vel.)	Curves roundabout, Sharp turn1, Sharp S-curves, Sharp turn 2

7.6 Chapter Summary

This Chapter shows methods to execute an MPC-based MCA in an online application for a nine DoFs driving simulator. Difficulties consist in the sequential calculation of an optimal control problem under the restrictions of hard real-time execution and deadlines. The parallel execution of calculation processes reduces the computation time considerably and makes a stable online drive possible. Other problems arise due to infeasible solutions of the optimal control problem. Introducing slack variables and limiting the reference signal improves stability. For the evaluation, the MCA$_{\text{MPC}}$ is executed in a real-time environment. Motion cueing quality is measured in an experimental study, with the following results:

1. The continuous rating can be applied for a comparison with three different MCAs. Statistical analysis indicates valid and reliable results compared to a general accepted PR.

2. Participants' rating show strong correlations to the dominant motion channels (longitudinal / lateral acceleration and the yaw velocity).

3. The structure of the closed-loop MCA$_{\text{MPC}}$ has the potential to further improve the motion cueing quality compared to a commercially available state-of-the-art, closed-loop MCA. Ratings of the MCA$_{\text{MPC}}$ exhibits less perceived motion incongruencies for the continuous rating and the PR compared to a closed-loop, benchmark MCA.

4. The open-loop, optimization-based MCA$_{\text{OFF}}$ shows the lowest PMIs, and thus the highest motion cueing quality of the compared MCAs. The algorithm exploits the full reference over the complete manoeuvre to optimize the simulator motion. The difference between the ratings of the MCA$_{\text{MPC}}$ and the MCA$_{\text{OFF}}$ indicates the further potential of the proposed closed-loop algorithm. Moreover, it shows the influence of the prediction strategy and the prediction length on the PMIs.

Regarding the results of the study, improving the prediction strategy and extending the time horizon can significantly increase the motion cueing quality.

Chapter 8

Closed-Loop, Hybrid MCA (MCA_{HYB})

The content of the following Chapter have been published in [52].

Current closed-loop, optimization-based motion cueing algorithms (MCAs) use a driver model to predict a future driving dynamics reference. These models are often inaccurate and/or computationally expensive because the future driving behaviour is unknown. In some cases the vehicle's trajectory is known in advance. In such open-loop simulations a driver sits passively in a vehicle and is being driven through a pre-recorded manoeuvre. In this case, optimization-based MCAs can compute an optimal trajectory for a pre-defined manoeuvre in a pre-processing step as described in Chapter 4 and 6. This Chapter presents the development of an MCA which uses the optimal trajectory of an open-loop, optimization-based MCA as a reference in a closed-loop simulation resulting in a quasi-optimal pre-positioning of the motion platform. Deviations between closed-loop driver and the reference are compensated by a closed-loop, state-of-the-art MCA. By combining a closed-loop MCA with the predictions obtained by an open-loop MCA, a hybrid motion cueing algorithm is obtained. One of the challenges for the implementation of a hybrid MCA is how to merge the data of the driver with the reference. Therefore, in a preparatory experiment, the driving behaviour of various drivers was measured and analysed. Then, a follow-up experiment was conducted to evaluate the novel hybrid MCA using the continuous rating method in an open-loop simulation. In order to analyse deviations between open-loop and closed-loop rating, a novel rating method for closed-loop simulations was developed and applied. Here, participants gave a section-wise oral rating during a closed-loop drive. Results show correlations between the open-loop and the closed-loop rating method. Both ratings indicate an improvement in motion cueing quality for the hybrid MCA.

8.1 Contribution

For the redundant motion system shown in Figure 1.3, a novel MCA approach is proposed in the present work. The so-called *hybrid MCA* (MCA_{HYB}) combines a closed-loop MCA and the optimal motion cueing results of an open-loop approach. In this study, a filter-based MCA is employed as a closed-loop algorithm which mainly consists of linear transfer functions [151]. Advantages of this approach lie in the relatively low computational cost for a closed-loop application. Nevertheless, the exploitation of the workspace by such an algorithm is typically quite low. For the open-loop algorithm, an optimization-based MCA is employed where an optimal motion cueing, and thus an optimal pre-positioning is obtained during the offline pre-processing. Based on a driver pool for a pre-defined manoeuvre, an average driver is calculated to obtain a driving dynamics reference. An optimal control problem minimizes deviations between the resulting accelerations and angular velocities at the simulator's driving position and the driving dynamics reference. The minimization is performed over the complete time horizon of the pre-defined manoeuvre. Thus, depending on the chosen weighting parameters and the used simulator model, an optimal platform positioning is obtained. Aside from this, the optimization problem considers the nonlinear kinematic chain of the motion system and includes the actuator limitations as constraints which allows exploiting the full extent of the workspace. For an open-loop method, real-time requirements do not have to be fulfilled. Thus, the difference between desired and provided percept can be minimized further compared to a closed-loop application. During the online simulation in the simulator, the results from the previously obtained output from the open-loop algorithm is used as a reference for the simulator's motions. Differences between the inputs from the online driver and the reference are corrected by the filter-based MCA. This way, a hybrid MCA merges the benefits of a perfectly predictive, optimization-based MCA with a real-time capable algorithm. It is the result of collaboration with Maximilian Spannagl and Samir Agabekov [2, 163].

8.2 Use Case

In order to employ the hybrid MCA, the route that is to be simulated should be known beforehand. This is however usually the case in most driving simulation studies. A reference for this route needs to be obtained, i.e. the data of a "reference driver". During the closed-loop simulation, the data of the closed-loop driver and the reference are combined. For the reference, where the driving dynamics data is known, an optimal simulator trajectory is calculated with the optimization-based MCA. During the closed-loop simulation, the differences between the reference driver and the current simulator driver are corrected by a closed-loop MCA. For good results, it is therefore required that the reference driver is representative, or in other words that the differences between the reference and closed-loop driver are not too large.

This Section deals with the question to which extent the driving profiles of various drivers differ for the same pre-defined course. The focus is put on the longitudinal acceleration, lateral acceleration and the yaw velocities, as realizing these motions on a motion system are typically particularly challenging. For this purpose, three test persons performed a five-minute drive five times each on the simulator presented in Table 2.2. An optimization-based MCA provided by the manufacturer was used to provide motion cueing for these drives [77]. To compare between the different manoeuvres, the individual signals of all manoeuvres have to be interpolated as a function of the position along the drive. The manoeuvre exhibits several acceleration and deceleration profiles, turns with different radii, hill-tops, urban sections and a roundabout, resulting in a broad range of driving manoeuvres (see red marked sections in Figure 8.1), listed in Table 8.1. The overall manoeuvre is the same as in the Chapters 6 and 7 but the choice of the sections differs. This is because a closed-loop rating method is used which requires sufficient time between the specific manoeuvres defined in Table 8.1.

8.2.1 Lateral Dynamics

For the lateral acceleration a_y, a relation between the longitudinal velocity v_x and the curve radius R exists [153]

$$a_y = \frac{v_x^2}{R} \quad [\mathrm{m/s^2}]. \tag{8.1}$$

Table 8.1: The driving manoeuvre used in the experiment is divided following seven parts.

1.	*Acc. procedure*:	Medium-strong acceleration to approx. 80 km/h.
2.	*Roundabout*:	Strong deceleration from approx. 80 km/h to approx. 30 km/h. Taking the first exit in the roundabout (75° turn). Medium-strong acceleration to approx. 100 km/h.
3.	*Sharp hilltop*:	Crossing a hill with a sharp top at approx. 60 km/h.
4.	*Sharp turn 1*:	Sharp left turn at approx. 60 km/h.
5.	*Sharp S-curves*:	Sharp S-curves at 60 km/h.
6.	*Sharp turn 2*:	Extended 90° left turn at approx. 60 km/h.
7.	*Dec. procedure*:	Medium-strong deceleration from 50 km/h to 0 km/h.

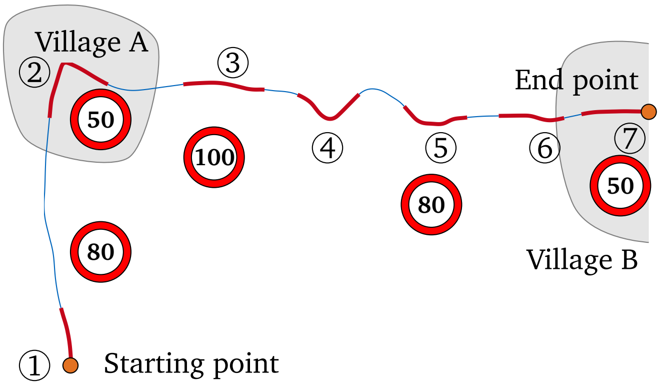


Figure 8.1: Course of the considered manoeuvre with speed limits.

Based on equation (8.1), there are two factors which influence the reference of a_y. First, the longitudinal velocity v_x has a quadratic influence on a_y. Assuming that the driver sticks to speed limits and considers the appropriate velocity for the course, differences between various manoeuvres are limited. Second, as a single-lane course was used in this study, the variability between R across drives was relatively small.

Figure 8.2 shows the lateral acceleration for the considered course. Although the profiles indicate various amplitudes, the data is quite homogeneous. The mean variance over the manoeuvres is significantly smaller compared to the longitudinal acceleration (cf. Table 8.2). Thus, in contrast to the longitudinal acceleration, the quality of the reference that can be obtained for the lateral direction is reasonable.

Table 8.2: Mean variance between the various manoeuvres for longitudinal acceleration, lateral acceleration and yaw velocity.

	Longitudinal acceleration	Lateral acceleration	Yaw velocity
Mean(Var(X))	0.2681 [m/s^2]	0.0576 [m/s^2]	0.00014 [rad/s]

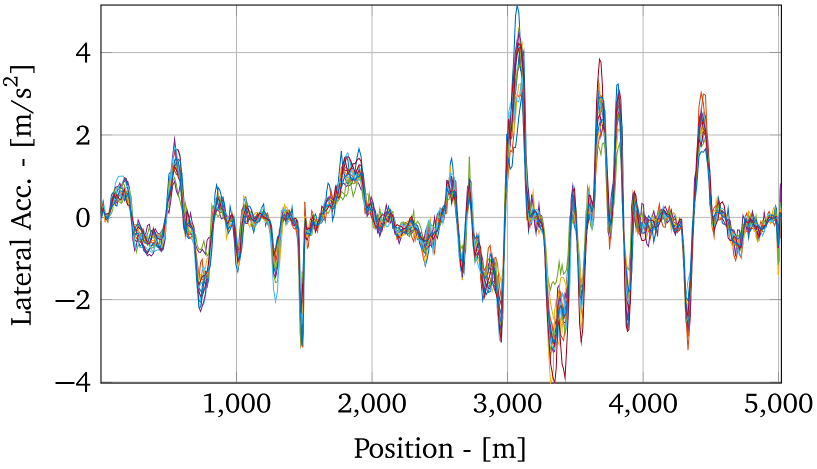


Figure 8.2: Comparison of the lateral acceleration of 15 recorded manoeuvres for the defined reference course.

8.2.2 Yaw Dynamics

The yaw velocity $\dot{\psi}$ shows characteristics similar to the lateral dynamics concerning the comparability of various drivers. The equation

$$\dot{\psi} = \frac{v_x}{R} \quad [\text{rad/s}] \tag{8.2}$$

indicates a linear relation between the longitudinal velocity v_x and the curve radius R [153]. Assuming only minor trajectory deviations between the drivers in the single-lane course, the variability in R between drives is relatively small and the yaw velocity mainly depends on v_x. As most drivers adjust their longitudinal velocity depending on the curvature, the yaw velocities deviate slightly between the drivers (cf. Figure 8.3). The low mean variance between the manoeuvres in Table 8.2 supports this statement.

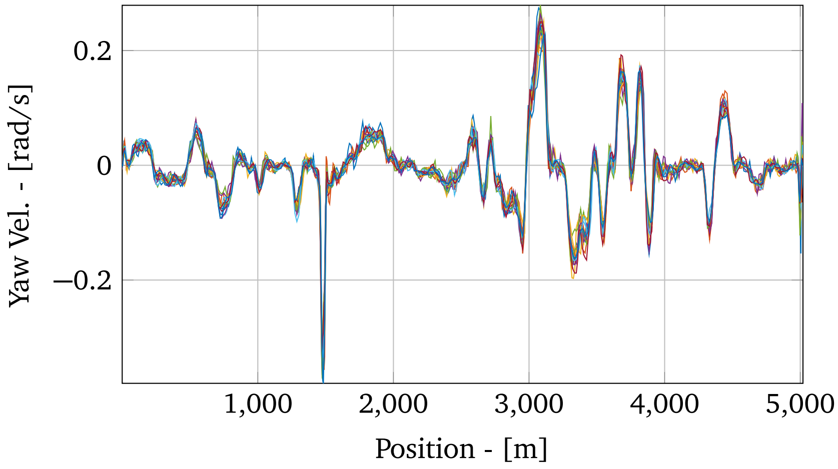


Figure 8.3: Comparison of the yaw velocity of 15 recorded manoeuvres for the defined reference course.

8.2.3 Longitudinal Dynamics

Several events affect the longitudinal dynamics of a driver. Traffic rules can cause positive or negative longitudinal accelerations at certain points. These include speed limit signs, traffic

lights or road signs. Moreover, a change in the longitudinal velocity can result from the road ahead, for example when approaching sharp turns, junctions or roundabouts. To a certain extent, all these events allow conclusions about the expected longitudinal acceleration.
However, there are a number of uncertainties that make an accurate prediction difficult. For example, the point of acceleration or deceleration is uncertain, in case of a speed limit change. Most drivers do not precisely adhere to speed limits. Individual driving style can be seen as another factor since, for instance, the acceleration profile of sporty drivers differs from comfortable drivers.
Figure 8.4 shows the longitudinal accelerations for the considered course which indicates that the acceleration profiles are only roughly comparable for the 15 manoeuvres. To further quantify the variations, the mean variance (Mean(Var(X))) over the manoeuvres is listed in Table 8.2. Compared to the results of the lateral and the yaw dynamics, the estimated variance value is fairly high. Summarizing, longitudinal dynamics vary strongly, both between drivers and across drives the point of acceleration differs significantly which makes obtaining a reliable reference in longitudinal direction difficult.

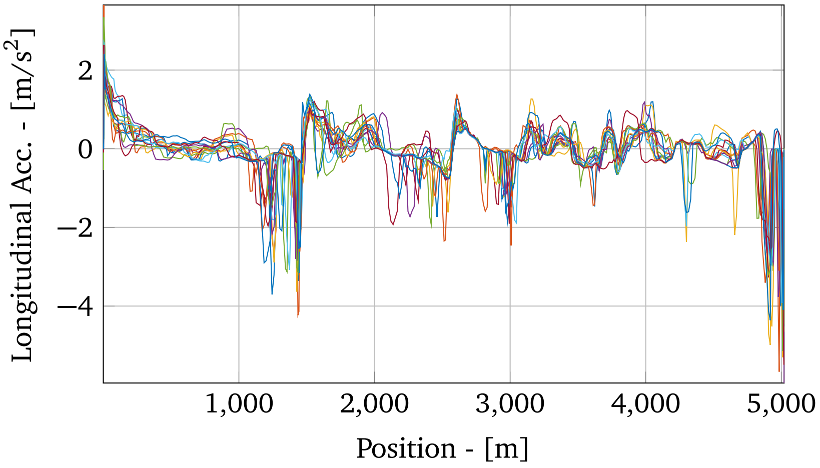


Figure 8.4: Comparison of the longitudinal acceleration of 15 recorded manoeuvres for the defined reference course.

8.2.4 Consequences

In order to use the potential of the optimization-based reference trajectory in a hybrid MCA, it is necessary that differences between the driving dynamics of the closed-loop driver and the reference are only small. Based on the analysis in Section 8.2, the following summarizing statements can be made:

- The course of the driving simulation is to be pre-defined and has to be observed by the closed-loop driver.
- The hybrid MCA is especially suitable for single lane courses, whereas courses with multiple lanes, like highways, are situations where a hybrid approach might not be favourable.
- The presence of a speed limit is crucial, as the velocity plays a major role in the reference of the acceleration.

- Other road users lead to unpredictable driving manoeuvres which deteriorates the motion cueing quality. For instance, the driving behaviour under presence of other road user is difficult to predict.

8.3 Simulative Comparison

Chapter 4.4 includes a simulative analysis between the filter-based MCA$_{\text{FILT}}$ and the optimization -based MCA$_{\text{OFF}}$. The evaluation reveals that there is a large potential in the motion cueing quality, demonstrated by the MCA$_{\text{OFF}}$ compared to the MCA$_{\text{FILT}}$. Other works, like [36, 54], compared an open-loop, optimization -based MCA with an online capable MCA, in an experimental simulation. Participants rated the deviation between perceived and expected stimuli in order to specify the motion cueing quality. The rating results confirmed the motion cueing potential of the optimization -based algorithms. A major advantage of these algorithms is that - in contrast to closed-loop MCAs - they have access to a perfect prediction of the future driving behaviour and can use this information. Thus, in this work, a method is proposed to use the promising motion cueing potentials of open-loop, optimization -based MCAs in a closed-loop simulation.

8.4 Structure of MCA$_{\text{HYB}}$

The concept of a *hybrid MCA* (MCA$_{\text{HYB}}$) is to combine an open-loop, optimization-based MCA (MCA$_{\text{OFF}}$) with a closed-loop, real-time capable MCA. In this work, a filter-based MCA (MCA$_{\text{FILT}}$) is taken as a closed-loop algorithm. The motion cueing quality of the MCA$_{\text{FILT}}$ is improved by using additional information of a precalculated, open-loop MCA$_{\text{OFF}}$. In the following the statements, listed at the end of Section 8.2.4, are employed in the use-case studied here.

Figure 8.5 illustrates the structure of the hybrid MCA approach. The driving dynamics data of a pre-defined course serves as a reference. Based on this data, the MCA$_{\text{OFF}}$ calculates an optimal trajectory depending on the chosen weighting parameters and the kinematic motion system model. Results of the optimal control problem are taken as the reference platform motion for the open-loop simulation. As the driving behaviour of a closed-loop driver differs from the reference, deviations are compensated by the closed-loop MCA$_{\text{FILT}}$. Due to speed differences, the timing of the closed-loop and open-loop driving dynamics data will diverge over time, as discussed in Section 8.2. In order to account for this, in the block "position matching", the reference data is mapped onto the current online data based on the current position on the road. For each time step t_S during the simulation, the position matching finds the time step t_{ref} at which the reference driver was located closest to the current closed-loop driver. Based on the observations from Section 8.2, this approach is applied for the lateral acceleration as well as for the yaw velocity. For the longitudinal direction, where variations are large, a different approach is taken as will be described later.

8.4.1 Reference Data Base

The following Section describes the calculation of the reference data, consisting of the driving dynamics data $[{}_V\boldsymbol{a}_V^{ref}, \dot{\boldsymbol{\beta}}_V^{ref}]$ and the vehicle positions ${}_I\boldsymbol{p}_V^{ref}$, as shown in Figure 8.7. It is assumed that the motion cueing quality improves, when differences between online and

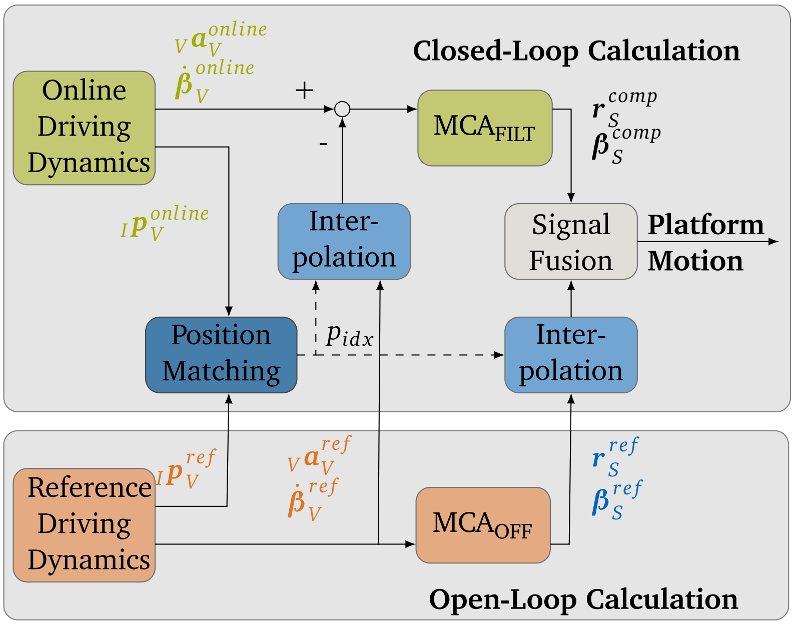

Closed-Loop Calculation
Online Driving Dynamics
$_V\boldsymbol{a}_V^{online}$
$\dot{\boldsymbol{\beta}}_V^{online}$
+
-
MCA$_{\text{FILT}}$
$\boldsymbol{r}_S^{comp}$
$\boldsymbol{\beta}_S^{comp}$
$_I\boldsymbol{p}_V^{online}$
Inter-polation
Signal Fusion
Platform Motion
Position Matching
p_{idx}
Inter-polation
Reference Driving Dynamics
$_I\boldsymbol{p}_V^{ref}$
$_V\boldsymbol{a}_V^{ref}$
$\dot{\boldsymbol{\beta}}_V^{ref}$
MCA$_{\text{OFF}}$
$\boldsymbol{r}_S^{ref}$
$\boldsymbol{\beta}_S^{ref}$
Open-Loop Calculation

Figure 8.5: Schematic structure of the hybrid MCA with the data branches: Online driving dynamics signal (green), reference driving dynamics data (orange) and reference optimal motion cueing data (blue).

reference drivers are small as less corrective action has to be executed by the MCA$_{\text{FILT}}$. The question arises, which driving style is appropriate for the reference driver? In a typical experimental study with test persons, a broad range of driving styles are present. Based on this fact, a method is required to obtain a representative reference manoeuvre, based on a large data base of pre-recorded drives. The approach taken in this study was as follows: Several drivers drove the pre-defined manoeuvre on the considered simulator. During the drives, motion cueing was provided by a standard MCA that was already available. The average of the recorded driving data is subsequently taken. This approach yields two advantages: First, a representative mean velocity is obtained which has a great influence on the lateral- and yaw dynamics, as seen in equations (8.1) and (8.2). Second, a representative trajectory concerning the lane position is obtained. Although the use-case concerns a single-lane road, there are still deviations in the driving positions between different drivers, e.g. due to various steering corrections.

In support of this argument, the upper plot of Figure 8.6 shows the x/y-positions in a curve for 15 different simulator drives, as well as their average trajectory obtained from them. Herein, the different steering points are visible. The lower plot of Figure 8.6 exemplifies the smooth averaged lateral acceleration compared to individual drives which shows considerable deviations in the amplitude and the course.

As shown in Figure 8.5, the averaged driving dynamics data $[_V\boldsymbol{a}_V^{ref}, \dot{\boldsymbol{\beta}}_V^{ref}]$ is used for the MCA$_{\text{FILT}}$ and the MCA$_{\text{OFF}}$. For the MCA$_{\text{OFF}}$, the reference signal $\boldsymbol{y}_V$ is as indicated in equation (4.1) using $[_V\boldsymbol{a}_V^{ref}, \dot{\boldsymbol{\beta}}_V^{ref}]$ (cf. Figure 4.1). As a reminder, $\boldsymbol{y}_V$ contains the specific forces and resulting angular velocities at the CRP of the virtual vehicle (cf. equation (4.1)). Based on the results from Section 8.2, deviations between different drivers are smallest for lateral and yaw motions. Thus, the reference data $\boldsymbol{y}_V$ for the optimization problem, defined in equation (4.6), are estimated with the following inputs

$$\dot{\boldsymbol{\beta}}_V^{ref} = \begin{bmatrix} \dot{\varphi}_V^{ref} \\ 0 \\ \dot{\psi}_V^{ref} \end{bmatrix}, \qquad _V\boldsymbol{a}_V^{ref} = \begin{bmatrix} 0 \\ _Va_{V,y}^{ref} \\ 0 \end{bmatrix}. \tag{8.3}$$

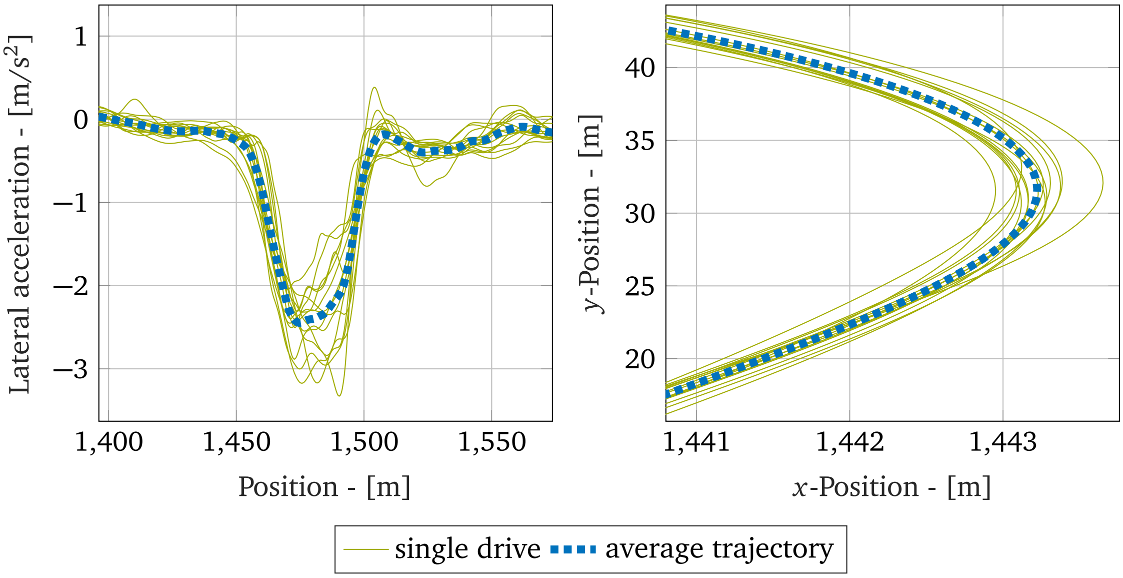


Figure 8.6: Upper plot: x/y-position in a curve for 15 different drives as well as their mean trajectory. Lower plot: smooth averaged lateral acceleration compared to individual drives.

The solution of the full time optimal control problem yields the reference motion cueing $[\boldsymbol{r}_S^{ref}, \boldsymbol{\beta}_S^{ref}]$. Although the reference data in equation (8.3) does not include motions in longitudinal, pitch and vertical direction, the reference motion cueing $[\boldsymbol{r}_S^{ref}, \boldsymbol{\beta}_S^{ref}]$ can comprise motions in these directions because of the redundant motion system structure (cf. equation (2.17)).

8.4.2 Data Branches and Position Matching

The MCA$_{\mathrm{HYB}}$ consists of two data branches as depicted in Figure 8.5 and Figure 8.7. On the one hand, the reference data base consists of the driving dynamics data $[{}_V\boldsymbol{a}_V^{ref}, \dot{\boldsymbol{\beta}}_V^{ref}]$ and the vehicle positions ${}_I\boldsymbol{p}_V^{ref}$ defined in an initial FoR I. Based on this data, the MCA$_{\mathrm{OFF}}$ calculates the trajectories $[\boldsymbol{r}_S^{ref}, \boldsymbol{\beta}_S^{ref}]$ for the considered reference manoeuvre. Thus, at every time step $t_S[k]$, a position ${}_I\boldsymbol{p}_V^{ref}[k]$ is associated with a motion cueing signal $[\boldsymbol{r}_S^{ref}[k], \boldsymbol{\beta}_S^{ref}[k]]$.
On the other hand, the online driver's data comprises the current driving dynamics state $[{}_V\boldsymbol{a}_V^{online}, \dot{\boldsymbol{\beta}}_V^{online}]$ and the current vehicle position ${}_I\boldsymbol{p}_V^{online}$. At every time step $t_{online}[l]$, the position matching algorithm searches for the minimum distance between the current position ${}_I\boldsymbol{p}_V^{online}[l]$ and a reference position ${}_I\boldsymbol{p}_V^{ref}[k]$. Subsequently, the following signals are read out from the reference data branch at the index k:

- The driving dynamics data $[{}_V\boldsymbol{a}_V^{ref}, \dot{\boldsymbol{\beta}}_V^{ref}]$, which is partly used for the MCA$_{\mathrm{FILT}}$.
- The reference motion cueing signals $[\boldsymbol{r}_S^{ref}, \boldsymbol{\beta}_S^{ref}]$.

The MCA$_{\mathrm{HYB}}$ was implemented in MATLAB/Simulink [108] and exported to C++ for integration into the target system of the driving simulation environment. The reference data is imported during the initialization phase of the driving simulation experiment.

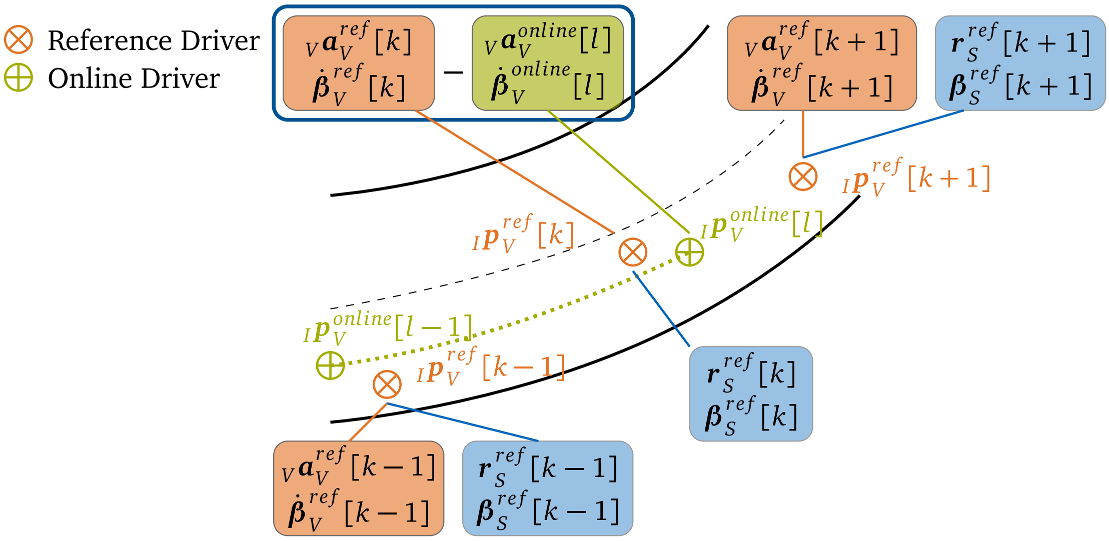

⊗ Reference Driver
⊕ Online Driver
${}_V\boldsymbol{a}_V^{ref}[k]$ $\dot{\boldsymbol{\beta}}_V^{ref}[k]$
−
${}_V\boldsymbol{a}_V^{online}[l]$ $\dot{\boldsymbol{\beta}}_V^{online}[l]$
${}_V\boldsymbol{a}_V^{ref}[k+1]$ $\dot{\boldsymbol{\beta}}_V^{ref}[k+1]$
$\boldsymbol{r}_S^{ref}[k+1]$ $\boldsymbol{\beta}_S^{ref}[k+1]$
${}_I\boldsymbol{p}_V^{ref}[k+1]$
${}_I\boldsymbol{p}_V^{ref}[k]$
${}_I\boldsymbol{p}_V^{online}[l]$
${}_I\boldsymbol{p}_V^{online}[l-1]$
${}_I\boldsymbol{p}_V^{ref}[k-1]$
$\boldsymbol{r}_S^{ref}[k]$ $\boldsymbol{\beta}_S^{ref}[k]$
${}_V\boldsymbol{a}_V^{ref}[k-1]$ $\dot{\boldsymbol{\beta}}_V^{ref}[k-1]$
$\boldsymbol{r}_S^{ref}[k-1]$ $\boldsymbol{\beta}_S^{ref}[k-1]$

Figure 8.7: Schematic scheme of the position matching between an online driver and a reference driver using the data branches shown in Figure 8.5 (analogue colour selection).

8.4.3 Interpolation

The position matching algorithm needs to account for the varying driving speed of the online driver, which may differ from the reference manoeuvre. Matching problems can arise when the index jumps for several increments or stays the same for a number of time samples. In this case, differentiating leads to discontinuities and numerical problems which arise in the induced reference signals. For this reason, it is not sufficient to search for a discrete index. The approach presented in this work is to orthogonally project the online driver's position onto the reference driver's trajectory as depicted in Figure 8.8. The trajectory is a linear interpolation between the discrete reference positions ${}_I\boldsymbol{p}_V^{ref}$. In an interval subsequent to the last calculated interpolation value I_{IV}, the system searches for the closest appropriated points of the reference driver. In Figure 8.8, these points are marked with B (nearest index) and C (second nearest index) and form a triangle with the current online driver's position A. The projection onto the reference trajectory, and thus the interpolation value I_{IV} can be calculated using geometric relationships which are explained in the following Section.
Using the triangle's side lengths a, b, c estimated by the position values A, B, C, the triangle's height h_a is calculated by

$$h_a = \frac{2 \cdot A_{abc}}{a}. \tag{8.4}$$

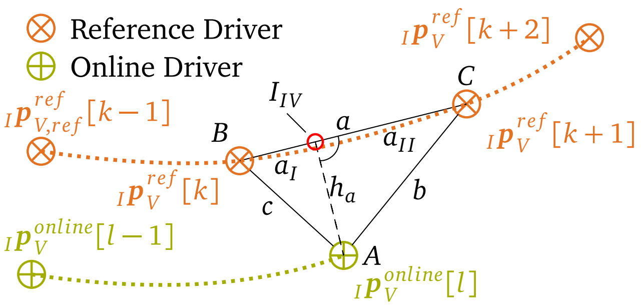
The triangle's area is defined by

$$A_{abc} = \sqrt{s(s-a)(s-b)(s-c)} \tag{8.5}$$

where $s = 0.5 \cdot (a+b+c)$ using the Heron's formula [98]. The interpolation point I_{IV} is the intersection of the segment BC and the segment AI_{IV} which are perpendicular to each other. Based on the index k, I_{IV} is calculated by

$$I_{IV} = \begin{cases} k + \frac{a_I}{a}, & \text{for index of } C > \text{index of } B \\ k - \frac{a_I}{a}, & \text{for index of } \ C < \text{index of } B \end{cases} \tag{8.6}$$

where $a_I = \sqrt{c^2 - h_a^2}$. In the last step, the reference motion cueing signals $[\boldsymbol{r}_S^{ref}, \boldsymbol{\beta}_S^{ref}]$ are interpolated using I_{IV}.



Reference Driver
Online Driver
$_I\boldsymbol{p}_V^{ref}[k+2]$
C
I_{IV}
$_I\boldsymbol{p}_{V,ref}^{ref}[k-1]$
B
a
a_{II}
$_I\boldsymbol{p}_V^{ref}[k+1]$
a_I
h_a
b
$_I\boldsymbol{p}_V^{ref}[k]$
c
$_I\boldsymbol{p}_V^{online}[l-1]$
A
$_I\boldsymbol{p}_V^{online}[l]$

Figure 8.8: Geometric representation of the orthogonal projection of the online driver's position onto the reference driver's trajectory (analogue colour selection as in Figure 8.5 and Figure 8.7).

8.4.4 Signal Fusion

As shown in Figure 8.5, the platform motions are composed of a superposition between the correction signals $[\boldsymbol{r}_S^{comp}, \boldsymbol{\beta}_S^{comp}]$ (estimated by MCA$_{\mathrm{FILT}}$) and the reference signals $[\boldsymbol{r}_S^{ref}, \boldsymbol{\beta}_S^{ref}]$ (estimated by MCA$_{\mathrm{OFF}}$) in the form

$$\boldsymbol{p}_{S,hyb} = \boldsymbol{r}_S^{comp} + \boldsymbol{p}_S^{ref} \tag{8.7}$$

$$\boldsymbol{\beta}_{S,hyb} = \boldsymbol{\beta}_S^{comp} + \boldsymbol{\beta}_S^{ref}. \tag{8.8}$$

The generation of the reference signals $[\boldsymbol{r}_S^{ref}, \boldsymbol{\beta}_S^{ref}]$ is described in Section 8.4.1.

8.4.5 Longitudinal Acceleration

As noted in Section 8.4.2, the longitudinal cueing cannot be performed in an analogous fashion to the lateral and yaw cueing, due to the large variations between drivers and between drives. Still, certain information about the future longitudinal dynamics is utilized in the MCA$_{\mathrm{HYB}}$ to improve the motion cueing quality with respect to MCA$_{\mathrm{FILT}}$. Here, a pre-positioning technique is proposed to enhance the workspace management.

Figure 8.4 shows the longitudinal acceleration profiles of 15 drives. The plot indicates areas with high positive or negative acceleration magnitudes at roughly the same road section. These acceleration events are associated with specific events or occurrences in the pre-defined course, like village entrances or speed limits. These occurrences cause similar behaviour across drivers. An algorithm was developed that automatically extracts the locations where positive or negative acceleration events can be expected based on the acceleration profiles of various drives. The algorithm searches for grouped events, which indicate the occurrence of multiple accelerations or decelerations with a large magnitude. The upper plot of Figure 8.9 shows both the single large events and automatically identified events, resulting in one acceleration manoeuvre and four deceleration manoeuvres.

These events were used to calculate a pre-positioning trajectory in the preparation phase of the driving simulation experiment. The longitudinal acceleration can be reproduced by a constant translational motion and/or a tilting motion below the rotational threshold. Both ways are used for the event based pre-positioning in the present work.

Translational pre-positioning techniques are reported in several works, such as [85, 150]. A translational pre-positioning function moves the motion system in the direction opposite to the expected manoeuvre. For example, in preparation for a positive acceleration, the motion platform moves to the rear position of the workspace. Thus, more motion space is available for the acceleration demand. The pre-positioning motion preferably is below the

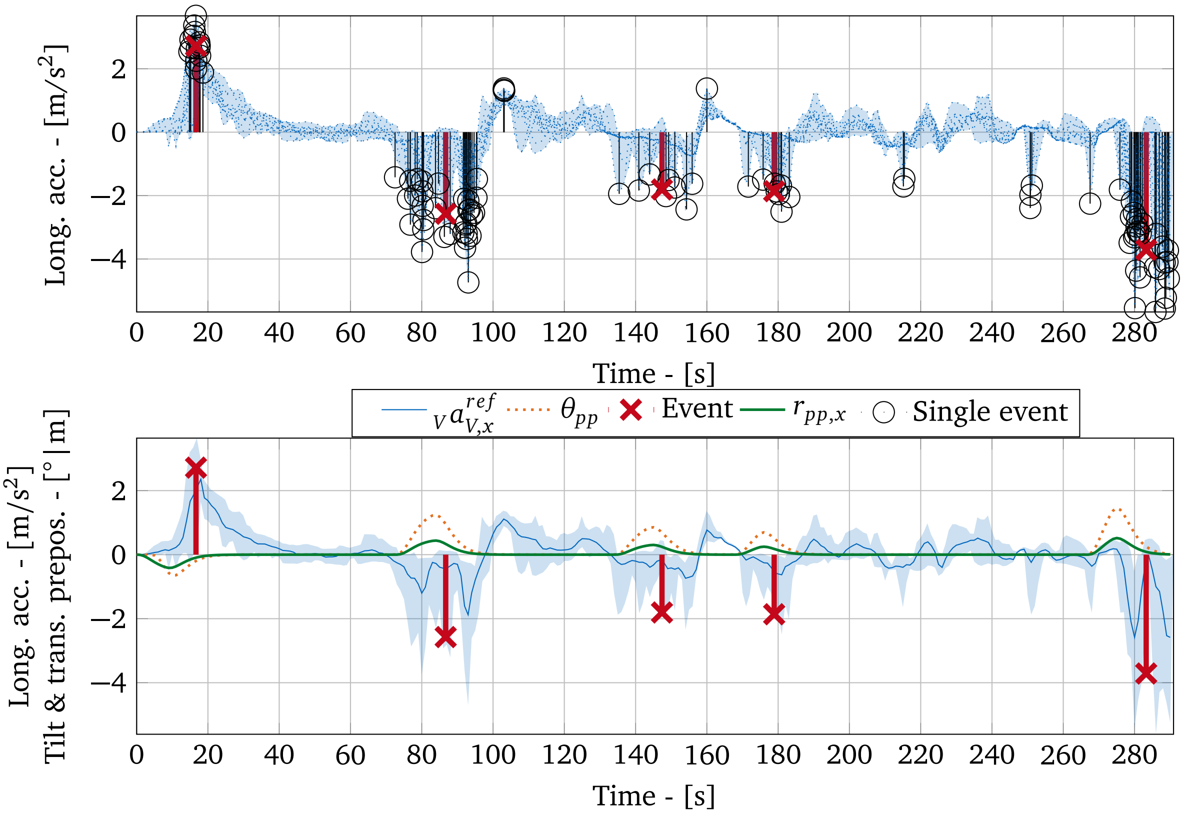


Figure 8.9: Upper plot: Identified (positive and negative) acceleration events of the considered single drives and the resulting grouped events used by the pre-positioning. Lower plot: Calculated signals for the longitudinal and angular pre-positioning of tripod and hexapod, respectively. The blue halftone indicates the range of the longitudinal acceleration.

human perception threshold. A maximum pre-positioning stroke $r_{pp,max}$ for the tripod and a pre-positioning acceleration $\ddot{r}_{pp}$, which is below the translational acceleration threshold was defined. The time t_{pp} to reach the pre-positioning position is estimated by

$$t_{pp} = k_{pp}\sqrt{\frac{2 \cdot r_{pp,max}}{\ddot{r}_{pp}}} \tag{8.9}$$

where k_{pp} is a factor to compensate position deviations of the pre-positioning events. The pre-positioning distance $r_{pp,x}$ varies depending on the intensity of the manoeuvre. The maximum acceleration peak determines the maximum pre-positioning $r_{pp,max}$. The position r_{pp} of the remaining events is proportional to $r_{pp,max}$. The pre-positioning signal was added to the output of MCA$_{\text{HYB}}$. The position matching synchronizes the pre-positioning signal with actual road position of the simulator driver. In order to receive a smooth acceleration profile, the pre-positioning signal is filtered using a third order low-pass filter. The lower plot of Figure 8.9 shows the filtered pre-positioning signal $r_{pp,filt,x}$ relative to the reference manoeuvre. In this study, the translational pre-positioning was only implemented for the tripod because of the small workspace capacities of the hexapod. The signal $r_{pp,filt}$ is differentiated twice and added to the acceleration $\ddot{r}_{S,tri,x}$ from the MCA$_{\text{FILT}}$.

In order to evaluate the functionality of the longitudinal pre-positioning, the deceleration manoeuvre before the roundabout from the course described in Section 8.2 is taken here as an example. Figure 8.10 shows the tripod's position in $_I x$-direction with and without pre-positioning function. The workspace restrictions are approximated by a constant upper and

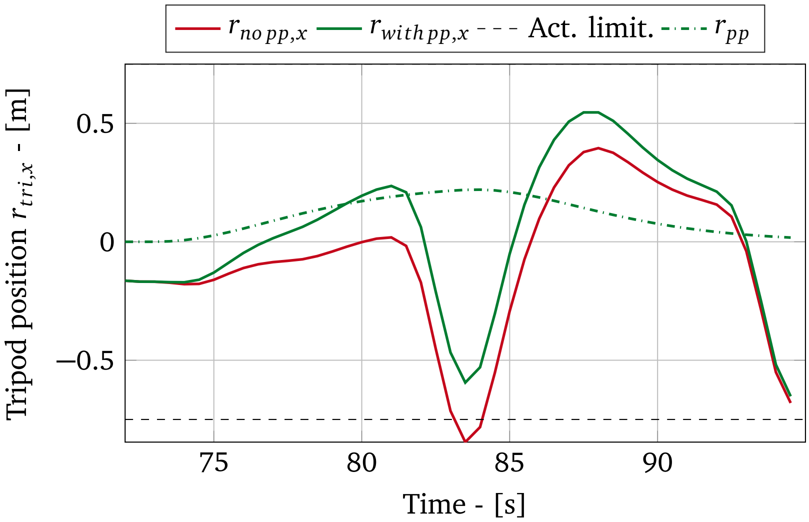


Figure 8.10: Tripod's position in $_I x$-direction with and without pre-positioning function. The platform sticks to its workspace limits applying the pre-positioning function, whereas no pre-positioning leads to a workspace overshoot.

lower bound. Using no pre-positioning leads to a workspace overshoot for the considered manoeuvre. Whereas, applying the pre-position, the tripod platform prematurely moves to the opposite direction leading to an effective increase in the available workspace.

The potential of improvement in the motion quality depends on the translational workspace capabilities of the motion system. Prepositioning for systems with small motion capabilities is not very effective and only leads to small improvements for short acceleration peaks. In this work, a tilting pre-positioning is additionally proposed which improves the cueing of long-lasting accelerations. Reproducing long-lasting accelerations is challenging, as they can lead to very large position excursions which quickly leads to reaching the simulator position bounds. Tilting at a low tilt rate is perceived as a slow-onset long-lasting longitudinal acceleration. Increasing the tilting rate increases the likelihood of perceiving the rotation. Analogously to the procedure of the longitudinal pre-positioning, acceleration events are indicated in the reference data which are used to calculate the tilting pre-positioning signal. Herein, just the positions of the acceleration peaks are relevant. A time period $t_{pp,\theta}$ before the occurrence of an acceleration event, the hexapod is tilted with a constant angular velocity $\omega_{pp,\theta}$. Due to the premature tilting procedure, a perceived translational acceleration is present before the actual acceleration manoeuvre leading to a higher perceived acceleration peak. The tilting velocity $\omega_{pp,\theta}$ and the pre-positioning angle θ_{pp} are bound to a maximum value in order to limit the perception error. Here, the deceleration manoeuvre before the roundabout was taken as an example again. Figure 8.11 shows the tilting angles with and without pre-tilting. A maximum pre-positioning angle of 2° is applied, resulting in a perceived longitudinal acceleration of approximately $0.34\mathrm{m/s^2}$. The premature tilting angles lead to an earlier perceived longitudinal acceleration and a false cue. Figure 8.12 shows the result of combining the translational and rotational pre-positioning techniques. Again, the previously considered deceleration manoeuvre is used. The pre-positioning techniques reveal an increase of the maximum deceleration value and a better tracking behaviour of the reference signal in comparison with using no pre-positioning methods.

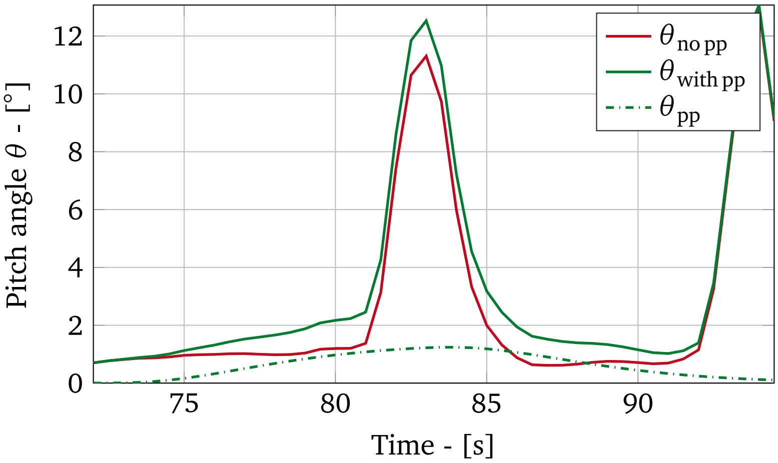


Figure 8.11: Tilting angles with and without rotational pre-positioning for a deceleration manoeuvre.

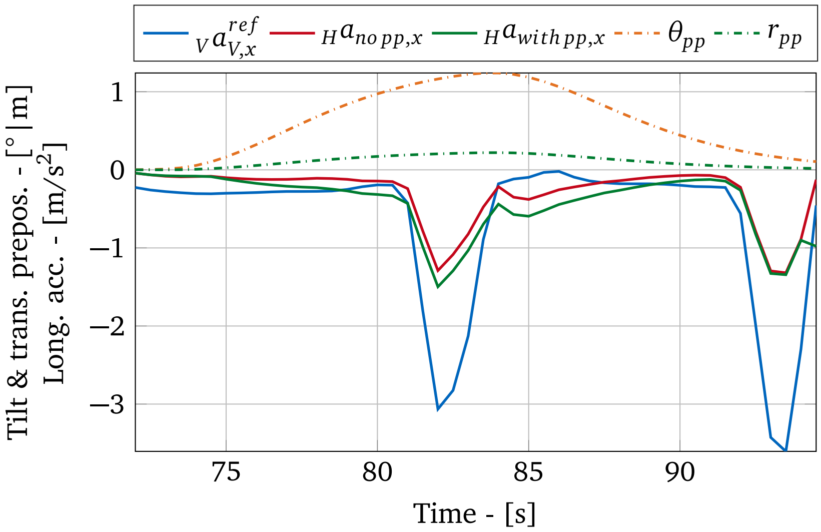


Figure 8.12: Resulting longitudinal accelerations using the translational and rotational pre-positioning techniques for a deceleration manoeuvre.

8.5 Experimental Evaluation

An experimental study was conducted to evaluate the motion cueing quality of the MCA_{HYB}. Herein, the MCA_{HYB} is compared to a classical, filter-based MCA_{FILT} (introduced in Section 3) which functions as a reference algorithm. OMIs and PMIs are used for evaluation.
The experiment was conducted in the simulator shown in Figure 1.3 with twelve participants between the age of 20-30. The participants were employees of the BMW Group or students of the Technische Universität München. Of the participants, 7 participants had never experienced a driving simulator. For the remaining participants, the driving simulator experience ranged between: 1-3 times (3), and more than 6 times (2). All the participants have a driver's licence.

8.5.1 Experimental Structure

The basic layout of the experiment follows previous works, such as [36, 37, 54] and includes three different rating methods. First, the CR method measured PMIs with a rotational knob during a simulator drive. This rating method was employed when the MCAs are used in an open-loop structure, so that the participants were passively driven through a pre-defined manoeuvre. As this work introduces an online-capable MCA, also a new rating method (section-wise post-hoc rating - SPR) was developed and employed which enables an evaluation of a closed-loop simulation. Here, the participants were asked to report PMIs orally after pre-defined sections during an active drive. Finally, after each drive, the participants rated the PMIs of the whole manoeuvre in a PR.

8.5.2 Research Questions

The study includes the following research questions for evaluation:

- Does the novel MCA$_{\text{HYB}}$ have a higher motion cueing quality than the MCA$_{\text{FILT}}$ using the CR in an open-loop simulation and using the SPR in a closed-loop simulation?
- Are significant correlations measurable in the rating results between the CR and the SPR?

8.5.3 MCAs Used for Comparison

For the comparison, a state-of-the-art, filter-based MCA is taken as a reference. Similar algorithms are used in simulators, such as [135, 144, 150]. In [36], a filter-based MCA was used as a benchmark in a comparison to an optimization-based MCA. The algorithm proposed in the current work can be used in an open-loop and in a closed-loop simulation. For the open-loop test, a recorded drive, which differs from the average drive, was chosen as an input. For the comparison to MCA$_{\text{HYB}}$, the MCA$_{\text{FILT}}$ was selected. This algorithm is already included inside the MCA$_{\text{HYB}}$. Comparing the two provides the clearest insight in the improvements that were made by the extensions that MCA$_{\text{HYB}}$ provide. Furthermore, the same filter parameters are chosen for the MCA$_{\text{HYB}}$ and the MCA$_{\text{FILT}}$, in order to avoid the possibility that deviations between both MCAs are based on the parametrization. Both MCAs are scaled to maximally exploit the workspace of the simulator. A more conservative scaling was used for the closed-loop simulation than for the open-loop simulation in order to preserve workspace capacities for unforeseen manoeuvres in the online drive. The applied scaling factors k_i for $i \in \{x, y, z, \varphi, \theta, \psi\}$ (six input signals) are listed in Table 8.3. Comparing the scaling factors, higher values can be used for the MCA$_{\text{HYB}}$ because of the improved workspace exploitation of the algorithm. The hand-tuned weights, used in the MCA$_{\text{OFF}}$ to calculate the reference trajectory for the MCA$_{\text{HYB}}$, are shown in Appendix D in Table D.6.

Table 8.3: Applied scaling factors k_i for $i \in \{x, y, z, \varphi, \theta, \psi\}$ for the open-loop and closed-loop simulation.

	open-loop		closed-loop	
	k_x	$k_{y,z,\varphi,\theta,\psi}$	k_x	$k_{y,z,\varphi,\theta,\psi}$
MCA$_{\text{FILT}}$	0.2	0.3	0.2	0.2
MCA$_{\text{HYB}}$	0.2	0.6	0.2	0.4

8.5.4 Rating Procedure

For the CR, participants rate PMIs using a rotational knob which is located at the middle console of the mockup. The scale ranges from $0-10$ and a visual representation of the rating scale was included in the driving scene shown to the driver. The scale was located in the centre of the driver's field of view, at approximately the location where a heads-up display would be located. A rating of 0 means low PMIs, thus a high motion cueing quality and a rating of 10 represents high PMIs (low motion cueing quality). The CR is logged with a sample rate of 10 ms, resulting in a quasi-time-continuous rating profile for the specific manoeuvres. As the participants have to fully concentrate on the rating, the CR can only be obtained in open-loop simulations.

As an additional measure to the CR method, an additional PR at the end of each trial is conducted. Thus, correlations between the results of the CR and the PR can be analysed. For the PR, the same rating scale as for the CR is applied.

As the MCA_{HYB} is online-capable, it can also be evaluated in a closed-loop drive. This, however, excludes the possibility of obtaining a CR. Therefore, a different approach was developed for the rating during closed-loop drives, which was the SPR. The seven sections that were rated using the SPR are indicated in Figure 8.1. The experiment leader kept in constant contact with the actively driving participants. The sections that were to be rated, we announced in advanced. Participants were asked to give an oral rating for each section using the same scale as for the previous rating methods. Also here, the participants gave a PR at the end of each closed-loop drive.

8.5.5 Experimental Design

The experiment was structured in three parts: First, in order to familiarize the participants with the CR method and the simulation environment, the participants conducted a training exercise. Some manoeuvres of the training exercise exhibited large PMIs, for which the participants were instructed to use the whole rating scale.

In the second part, the experiment started and the CR was used in an open-loop simulation. The reference manoeuvre for MCA_{HYB} was obtained from a measured data pool, as described in Section 8.4.1. Data from a single driver were used to as data of the online driver. Prior to the experiment, the offline motion cueing was calculated for a MCA_{FILT} and a MCA_{HYB}, for a pre-recorded manoeuvre. During the experiment these results were replayed. Both algorithms were rated twice in a randomly pairwise order. After each trial, the participants were asked for a PR^{I}. Each trial lasted approximately 5 min. After four ratings, a short break was taken.

In the last part of the experiment, both MCAs were driven in a closed-loop simulation, so the participants experienced the MCA_{HYB} and the MCA_{FILT} during an active drive. After defined sections in the manoeuvre, indicated in Figure 8.1, the participants gave an SPR. The MCA_{FILT} and MCA_{HYB} were rated twice in a random pairwise comparison. The participants were instructed to adhere the speed limits and the traffic regulations. The SPR were recorded by the experiment leader. At the end of each drive, a PR^{II} was conducted to rate the overall PMIs of the drive.

After each drive, a break was offered. The experiment lasted approximately 2 h per participant.

8.5.6 Independent and Dependent Variables

Independent variable in the experiment is the MCA (two levels: MCA$_{\mathrm{FILT}}$ and MCA$_{\mathrm{HYB}}$) and the simulation type (two levels: open-loop and closed-loop). These algorithms are embedded in a within-subject design. By these means, the algorithms are compared in a two-times randomly ordered comparison, once in a closed-loop simulation and once in an open-loop simulation.
Dependent variables are the ratings obtained through the three rating methods (CR, PR and SPR).

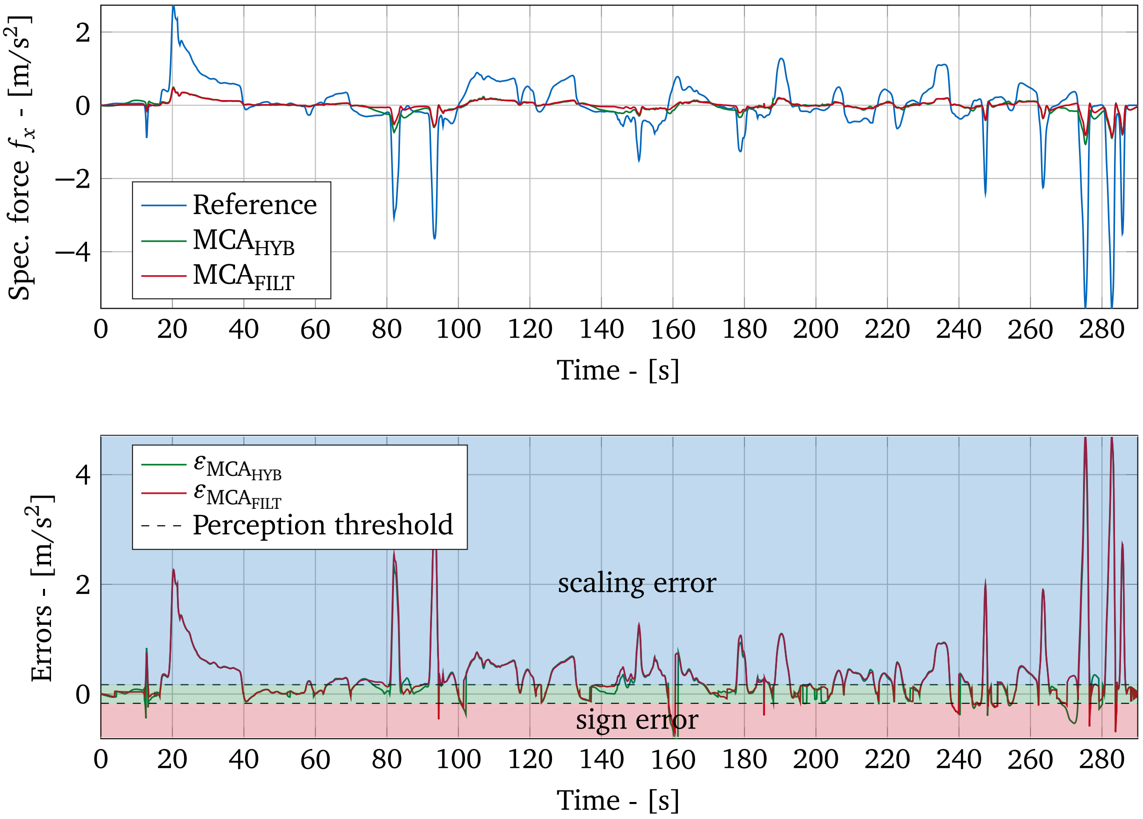


Figure 8.13: Upper plot: Longitudinal specific forces of the unscaled online driver ${}_{V}f_{V,ref,x}$ and the motion cueing signals ${}_{H}f_{hyb,x}$ and ${}_{H}f_{filt,x}$ for a closed-loop test. Lower plot: Respective motion cueing error according to equation (5.70).

8.6 Objective Motion Incongruencies

In the following Section, the resulting longitudinal and lateral accelerations are analysed as these DoFs are considered as the most relevant motion channels of the experiment. Figure 8.13 shows the resulting physical longitudinal accelerations at the CRP of MCA$_{\mathrm{HYB}}$, MCA$_{\mathrm{FILT}}$ and the online driver for the manoeuvre described in Section 8.2. To classify deviations to the online driver, the error equation (5.70) from Chapter 5.8 is used which distinguishes between sign errors and scaling errors using the unscaled $\boldsymbol{y}_V$ as reference. Sign errors are rated worse compared to scaling errors as the driver perceives an opposite motion to its expectation [67].
The errors $\boldsymbol{\varepsilon}_x$ for the MCA$_{\mathrm{FILT}}$ and MCA$_{\mathrm{HYB}}$ are plotted on the lower part of Figure 8.13. The deviations from the online driver, i.e. the errors, for the two algorithms are of a similar

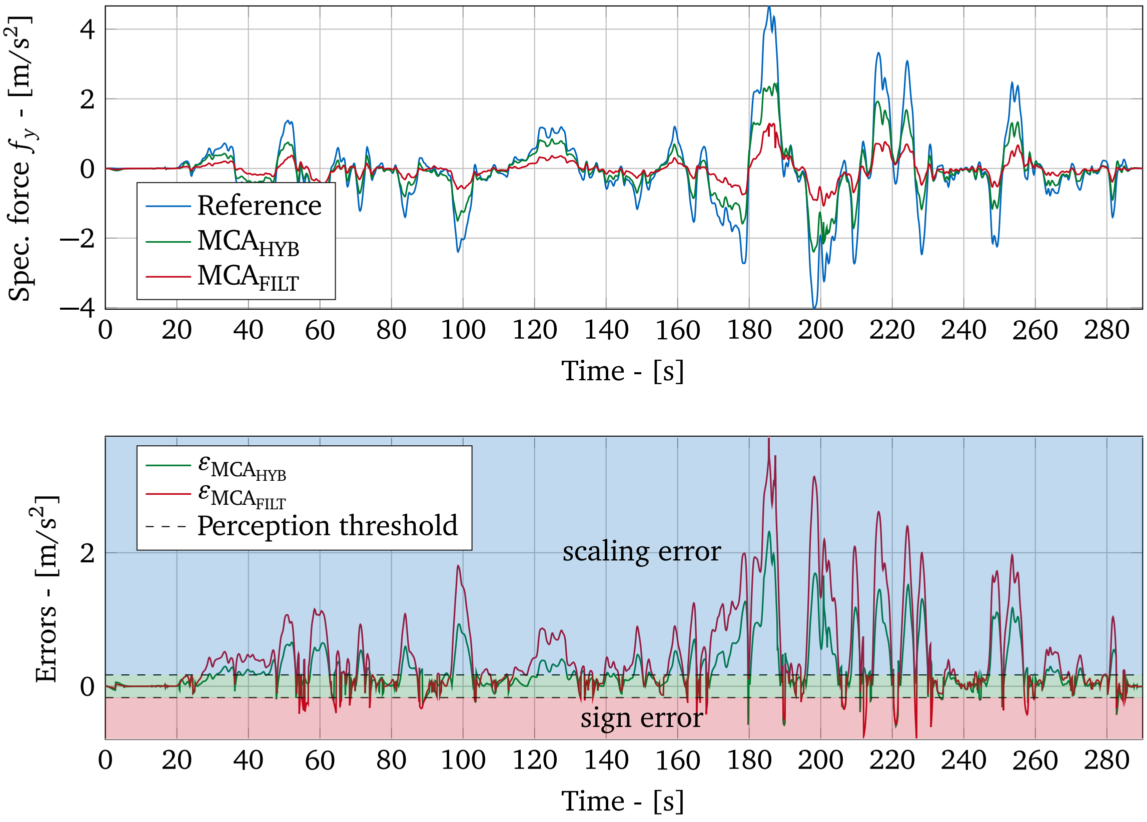


Figure 8.14: Upper plot: Lateral specific forces of the unscaled online driver ${}_V f_{V,ref,y}$ and the motion cueing signals ${}_H f_{hyb,y}$ and ${}_H f_{filt,y}$ for a closed-loop test. Lower plot: Respective motion cueing error according to equation (5.70).

magnitude throughout the manoeuvre. At the locations where pre-positioning was applied (approx. time instances: 80 s, 180 s and 270 s) the MCA$_{\mathrm{HYB}}$ shows a smaller error than MCA$_{\mathrm{FILT}}$, demonstrating the added benefit of the pre-positioning approach. It also reveals that the pre-positioning motions are mainly below the perception threshold.

The resulting lateral accelerations for the MCA$_{\mathrm{HYB}}$, MCA$_{\mathrm{FILT}}$ and the online driver are depicted in the upper plot of Figure 8.14. The lower part of Figure 8.14 reveals the resulting errors $\boldsymbol{\varepsilon}_y$ applying equation (5.70). Visual observation of the results shows that sign errors of MCA$_{\mathrm{HYB}}$ are mainly below the perception threshold. Scaling errors of the MCA$_{\mathrm{HYB}}$ generally exhibit lower values compared to the MCA$_{\mathrm{FILT}}$. To further quantify the results, the rms-ε_{rms} of the deviations in lateral direction for both algorithms are estimated to: $\varepsilon_{\mathrm{rms,\ MCA_{HYB}}} = 0.4696$; $\varepsilon_{\mathrm{rms,\ MCA_{FILT}}} = 0.8394$.

8.7 Experimental Results

The evaluation of the experiment is sectioned into three parts: First, the CR in an open-loop simulation - second, the SPR in a closed-loop simulation and the third part examines correlations between the CR and the SPR.

To check the central tendency (significance of mean-values) of two dependent samples, the paired t-test (test statistic = t) for normally distributed samples and the Wilkoxon signed rank test (test statistic = z) for non-normally distributed samples is used. The Lilliefors-test is used for testing normal distribution of a sample. Definitions for the mean value μ, the standard deviation σ and the standard error σ^* can be found in Section 6.6 in equations (6.4)-(6.6).

8.7.1 Open-Loop Continuous Rating

Reliability and Validity

Reliability is a measure of the internal consistency and the interrelatedness of ratings [18]. In this experiment, reliability is estimated by the Pearson correlation. Therefore, for each participant, the correlation matrix between the two trials of each independent variable (MCA$_{\text{FILT}}$ and MCA$_{\text{HYB}}$) is calculated. The side diagonal value of the correlation matrix indicates the linear relationship between the two trials. In order to obtain a single value for the reliability criterion, the average of the mean correlation values for the two independent variables is used [93]. Figure 8.15 shows the correlation values for twelve participants which were included in the analysis. By visual examination of the data, a critical value for the Pearson correlation of 0.2 is assumed for the experiment. The reliability of the rating method has been shown in other works, such as [36, 37, 54]. The inclusion of participants is more important than the exclusion as exclusion reduces the statistical power. Nevertheless, two participants (ID$_6$, ID$_7$) are excluded from the study (cf. red markers Figure 8.15) because of a too low correlation value. For the further evaluation, ten participants are considered.

In order to analyse whether the intended property was indeed measured during the experiment (validity), the continuous rating is compared to the generally accepted PR method [37]. Figure 8.16 shows the mean CR for MCA$_{\text{FILT}}$ and MCA$_{\text{HYB}}$ obtained during the manoeuvre and the mean PR^I (dashed lines). A significant linear relationship (Pearson correlation) between the two ratings for both MCAs exists ($r = 0.65$, $p < 0.05$). The Pearson correlation is estimated using the differences between both MCAs for the maximum mean CR and the PR^I. Herein, the respective CR value for each participant is taken at the time step of the maximal, overall mean CR. The non-zero correlations confirm the assumption that in this experiment the CR measures the motion cueing quality.

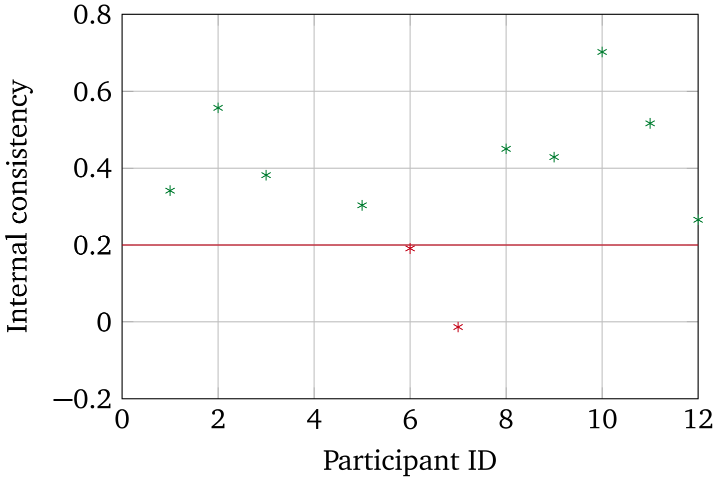


Figure 8.15: Internal consistency of the ratings per participant.

Overall Rating

Participants rated PMIs, where a higher value means a larger incoherence, hence worse motion cueing quality.

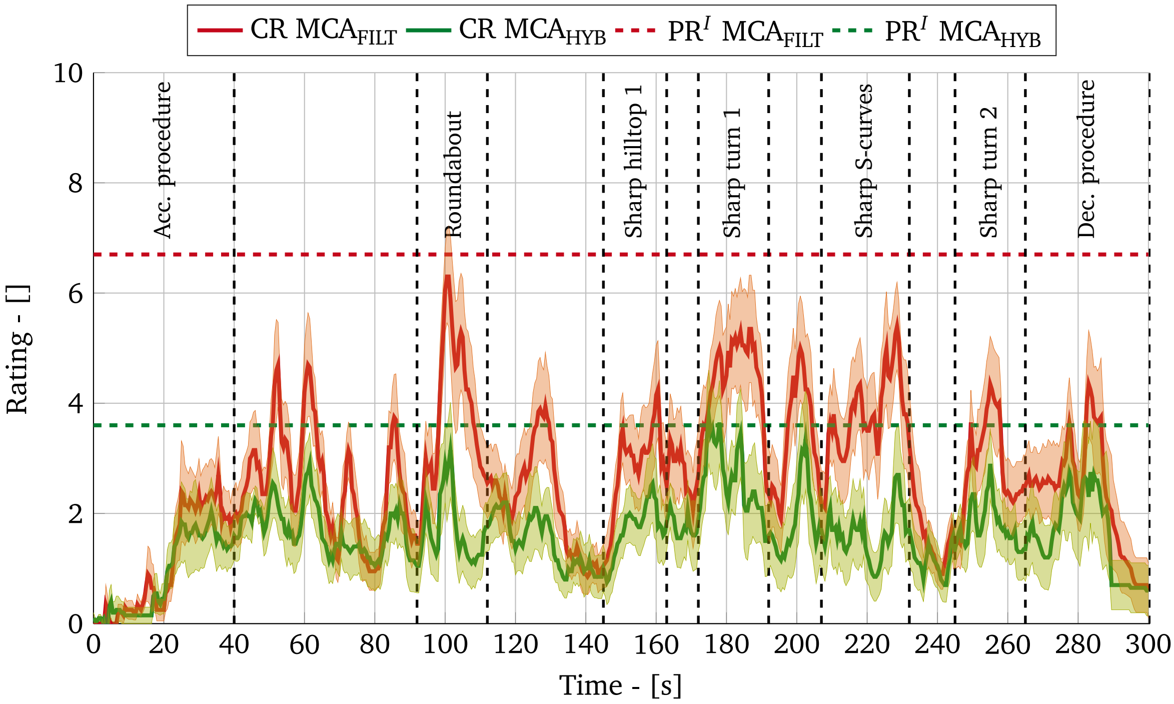


Figure 8.16: Mean CR with standard error (transparent area) and mean PR^I for the MCA$_{FILT}$ and MCA$_{HYB}$ over the complete manoeuvre. Dashed lines separate the manoeuvre into the seven sections as defined in 8.2.

Post-hoc Rating (PR^I): Figure 8.17 shows a significant difference ($t(9) = 6.08, p < 0.001$) between the mean PR^I for MCA$_{FILT}$ ($\mu_{MCA_{FILT}} = 6.7$; $\sigma^*_{MCA_{FILT}} = 0.62$) and MCA$_{HYB}$ ($\mu_{MCA_{HYB}} = 3.60$; $\sigma^*_{MCA_{HYB}} = 0.36$). The MCA$_{HYB}$ indicates less motion discrepancies than the MCA$_{FILT}$. The mean of the two rating pairs for the PR^I is depicted in Figure 8.17. The mean value of the PR^I does not change significantly for the two trials of the MCA$_{FILT}$ (MCA$_{FILT}$: $z(9) = -0.576, p > 0.05$) and the MCA$_{HYB}$ (MCA$_{HYB}$: $t(9) = 0.515, p > 0.05$) for which reason the ratings are assessed as sufficiently consistent.

Continuous Rating: Figure 8.16 shows the mean continuous rating of all participants for MCA$_{FILT}$ and MCA$_{HYB}$ with the standard error. The overall mean values between MCA$_{FILT}$ ($\mu_{MCA_{FILT}} = 2.64$, $\sigma^*_{MCA_{FILT}} = 0.65$) and MCA$_{HYB}$ ($\mu_{MCA_{HYB}} = 1.60$, $\sigma^*_{MCA_{HYB}} = 0.55$) do not differ significantly ($z(9) = 1.47$, $p < 0.15$). By visual examination, the CR of the MCA$_{HYB}$ exhibits a lower rating than the MCA$_{FILT}$ almost over the complete simulation.

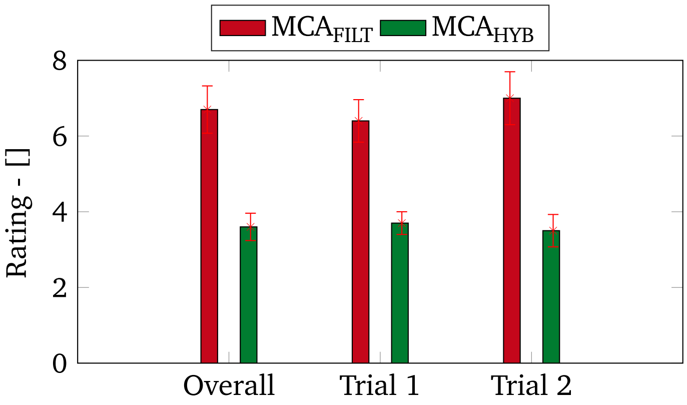


Figure 8.17: Overall mean PR^I and mean PR^I for trials 1-2 with standard error in an open-loop drive.

Manoeuvre based Rating

The mean values of the CR for the manoeuvres defined in Section 8.2 are listed in Table 8.4, including the results of a significance test. For all considered manoeuvres, the $\mathrm{MCA_{HYB}}$ exhibits a lower rating compared to the $\mathrm{MCA_{FILT}}$. Especially in manoeuvres with strong lateral accelerations and high yaw rates, the $\mathrm{MCA_{HYB}}$ shows less PMIs. This provides further evidence that using the offline data in the $\mathrm{MCA_{HYB}}$ significantly improves the motion cueing quality.

Table 8.4: Significant differences between the mean values of $\mathrm{MCA_{FILT}}$ and $\mathrm{MCA_{HYB}}$ within the sections. Largest values are plotted in bold.

	mean value		
Section	$\mathrm{MCA_{FILT}}$	$\mathrm{MCA_{HYB}}$	$t(9)-/z(9)$–stat.
Acc. procedure	1.08	0.87	$t = 2.96^*$
Roundabout	3.29	1.71	$t = 5.04^*$
Sharp hilltop 1	2.81	1.72	$t = 1.66^*$
Sharp turn 1	4.44	**2.62**	$z = 3.55^*$
Sharp S-curves	**4.77**	1.63	$z = 3.97^*$
Sharp turn 2	2.85	1.78	$t = 3.94^*$
Dec. procedure	2.32	1.57	$t = 3.76^*$

$^*(p < 0.01)$

8.7.2 Closed-loop Section-wise Post-hoc Rating

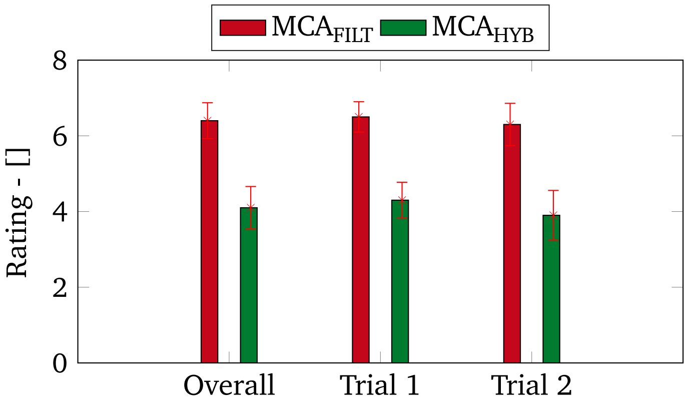


Figure 8.18: Overall mean PR^{II} and mean PR^{II} for trials 1-2 with standard error in a closed-loop drive.

Overall PR^{II}: In the second part of the experiment, participants rated PMIs during an online drive. They gave an oral rating after seven pre-defined sections within a single simulation and an overall PR^{II} after each completed simulation trial. Figure 8.18 depicts the significant difference ($t(9) = 5.28, p < 0.01$) between the mean of the $\mathrm{MCA_{FILT}}$ ($\mu_{\mathrm{MCA_{FILT}}} = 6.4$, $\sigma^*_{\mathrm{MCA_{FILT}}} = 0.47$) and the $\mathrm{MCA_{HYB}}$ ($\mu_{\mathrm{MCA_{HYB}}} = 4.1$, $\sigma^*_{\mathrm{MCA_{HYB}}} = 0.56$) for the overall PR^{II}. Similar to the PR^{I} in the open-loop experimental part, the $\mathrm{MCA_{HYB}}$ shows a higher motion cueing quality compared to the $\mathrm{MCA_{FILT}}$. Figure 8.18 also shows the mean PR^{II} for the first and second trial over all participants. The PR^{II} between the open-loop simulation (cf. Figure 8.17) and closed-loop simulation (cf. Figure 8.18) indicate a significant correlation. The Pearson correlation coefficient for the PR^{II} are:

1. MCA_{FILT}: $r = 0.73,\ p < 0.05$,
2. MCA_{HYB}: $r = 0.78,\ p < 0.01$.

Section-wise Post-hoc Rating: The boxplot in Figure 8.19 shows the median SPR for the seven considered manoeuvres. The boxes indicate the 25th and 75th percentile. The thin vertical line presents the spread for the minimum and maximum rating. For all seven manoeuvres, the MCA_{HYB} exhibits a lower median than the MCA_{FILT}. Linear correlations reveal the maximum SPR of the seven ratings and the overall PR^{II} at the end of the online drive for each MCA:

1. MCA_{FILT}: $r = 0.94,\ p < 0.01$,
2. MCA_{HYB}: $r = 0.91,\ p < 0.01$.

The significant values reveal a correlation between the novel SPR and the generally accepted PR^{II}.
Table 8.5 lists the Pearson correlation coefficients between the CR of the open-loop simulation and SPR of the closed-loop simulation for the seven considered manoeuvres. A significant value is found for 9 out of 14 correlations (all correlations are positive), which provides some indication of the validity of the applied SPR. On the other hand, there are minor deviations in the results of both rating systems, which could point to the influence between an open-loop and a closed-loop rating method.

Table 8.5: Pearson correlation coefficients between the CR and SPR for the seven considered manoeuvres.

Section	MCA_{FILT}	MCA_{HYB}
Acc. procedure	$r = 0.26^{***}$	$r = 0.12^{***}$
Roundabout	$r = 0.65^{**}$	$r = 0.76^{**}$
Sharp hilltop 1	$r = 0.78^{*}$	$r = 0.83^{*}$
Sharp turn 1	$r = 0.72^{**}$	$r = 0.81^{*}$
Sharp S-curves	$r = 0.68^{**}$	$r = 0.78^{*}$
Sharp turn 2	$r = 0.74^{**}$	$r = 0.59^{***}$
Dec. procedure	$r = 0.40^{***}$	$r = 0.59^{***}$

$^{*}(p < 0.01)$, $^{**}(p < 0.05)$, $^{***}(p > 0.05)$

8.8 Chapter Summary

This Chapter proposes an MCA consisting of a combination between an open-loop optimization-based MCA_{OFF} and a closed-loop MCA_{FILT}. A preliminary analysis indicated advantages of an open-loop, optimization-based MCA in contrast to a state-of-the-art filter-based algorithm. Using reference driving dynamics data for a pre-defined manoeuvre in the open-loop, optimization-based MCA leads to an quasi-optimal pre-positioning motion, entailing an improved workspace management. Thus, deviations between reference motions and motions provided by the simulator are minimized to a further extent resulting in a higher motion cueing quality.
The Chapter shows a method on how to use such a precalculated optimal trajectory for a specific manoeuvre in a closed-loop simulation environment. The driving behaviour of various drivers was recorded for a pre-defined manoeuvre. Results indicate minor deviations between the drives for lateral accelerations and yaw angular velocities. Using this knowledge,

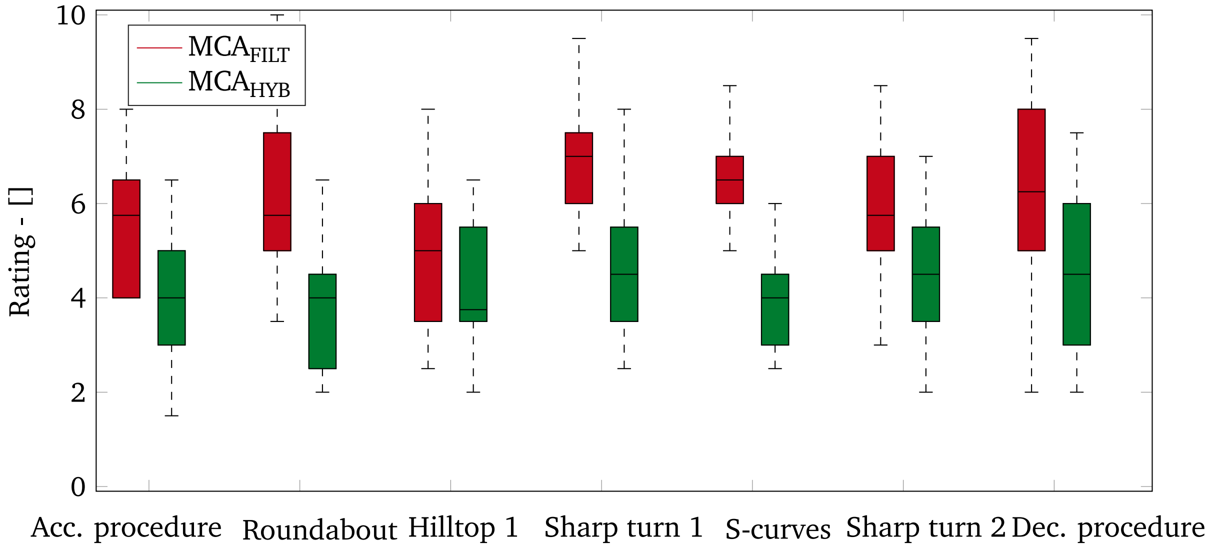


Figure 8.19: Median SPR for the seven considered manoeuvres in a closed-loop drive. The boxes indicate the 25th and 75th percentile and the thin vertical line presents the spread for the minimum and maximum rating.

an open-loop, optimal control problem estimates the optimal trajectory for an averaged driver in a pre-processing step. In the closed-loop simulation, a position matching function matches the optimal trajectory with the current state of the online driver. Driving dynamics deviations between open-loop and closed-loop driver are corrected by a filter-based MCA_{FILT}.

The MCA_{HYB} was evaluated in an experiment with twelve participants in a pairwise comparison with the MCA_{FILT}. In an open-loop simulation, the results of the continuous rating show a significant improvement in motion cueing quality for the MCA_{HYB}. By grouping specific manoeuvres of the overall simulation into sections, the mean continuous rating of the MCA_{HYB} reveals less perceived motion incongruencies compared to the MCA_{FILT}. A correlation between the results of the continuous rating and a generally accepted post-hoc rating method shows the validity of the rating results. For the second part of the experiment, a novel rating method was introduced consisting of an active drive in a closed-loop simulation. For pre-defined sections within the overall simulation, participants gave an oral post-hoc rating (SPR). The results indicate less PMIs for the MCA_{HYB} than for the MCA_{FILT}. Considering the three applied rating methods, for each, PMIs are rated lower for the MCA_{HYB}, which provides some good evidence that the algorithm significantly provides a higher motion cueing quality. Correlations between the CR and the SPR provide some confidence in the validity of the novel closed-loop rating method.

Chapter 9

Conclusion

The thesis describes several methods of MCAs which calculate the motions of a redundant driving simulator under consideration of its workspace capabilities. This Chapter summarizes the methods, discusses the results and gives motivations for future work.

9.1 Summary and Discussion

Developing new approaches of MCAs requires to deal with the simulator kinematics and the human perception system. Chapter 2 introduces the motion platform (DiM), which is considered throughout the thesis. The simulator structure, consisting of hexapod and tripod, is classified among other driving simulator topologies from literature. Using the relative kinematics, the three absolute accelerations and the three angular velocities working on the driver can be expressed. These six motion channels enable a comparison between vehicle motions and simulator motions to obtain the objective motion incongruencies. The Chapter describes advantages to integrate information of the vestibular system to an MCA, which can be summarized to: First, the perception thresholds can be exploited in the motion cueing algorithm. Herein, tilt coordination is a common technique to reproduce long-lasting accelerations by slowly tilting the motion platform preferably below the rotational perception threshold. Second, a vestibular model can be integrated to an MCA to reproduce only perceivable motions by the motion system. A vestibular model estimates perceived driver motions out of physical driver motions. In the chapter, different approaches of vestibular models are discussed and analysed. Results show that the transmission behaviour among the considered vestibular models is similar for low- to mid-frequency ranges but differs for higher frequencies.

Chapter 3 presents the basic functionalities of filter-based MCAs which mainly consist of linear transfer functions. The filter-based MCA_{FILT} is applied to the DiM simulator by using a frequency splitter which divides the signal into three signal domains correlating with the workspace capabilities of hexapod, tripod and tilt motions. To enhance the workspace exploitation for a predicted manoeuvre, heuristic pre-positioning techniques are described which provide an opposite motion in advance to the expected one. As filter-based MCAs do not consider workspace restrictions, a workspace supervision module is proposed which checks the workspace adherence of each actuator of a parallel robot. The module stops the simulator in case of an exceedance and provides a fade-in function to return to the desired simulator motion.

Chapter 4 proposes the open-loop, optimization-based MCA_{OFF} which uses a single optimal control problem to calculate the simulator motions. The objective function minimizes deviations between vehicle motions and simulator motions which are acting on the driver. The algorithm is used for open-loop simulations which enables an ideal prediction strategy for a pre-recorded manoeuvre over the full time horizon. Results yield an optimal trajectory of the motion simulator. A simulative comparison shows a significant reduction of objective motion incongruencies of the MCA_{OFF} compared to the MCA_{FILT} which gives good evidence in improving the motion cueing quality.

Chapter 5 shows an approach for an online-capable optimization-based MCA. A model-predictive control framework solves an optimization problem at discrete time steps with a sampling time of 10 ms. A linearized prediction model which predicts the future driving dynamics is included to a quadratic objective function with linear inequality constraints. A simulative analysis shows two effects. First, a partitioning of the time horizon into a prediction horizon, input horizon and constraint horizon reduces the computational effort without a significant loss in motion cueing quality. The linearization of the nonlinear inverse kinematics between workspace and actuator space is justifiable because a long-term forecast of the actuator states over the prediction horizon is not absolutely necessary.

In a first development stage, the algorithm (named MCA_{OPT}) was used as a proof-of-concept in a closed-loop application. An experimental study measured the potential of the MCA_{OPT} to a state-of-the-art benchmark MCA_{BM}. The design and the results of the study are presented in Chapter 6. The continuous rating (CR) method was used where participants rated perceived motion incongruencies during an open-loop simulation. Results show a consistent rating among the participants. Correlations to a general accepted post-hoc rating provides good evidence that the CR is a valid rating method to measure the motion cueing quality. The rating of the MCA_{BM} shows higher perceived motion incongruencies which demonstrates the potential of the MCA_{OPT}. Ratings were used to develop a rating model which predicts perceived motion incongruencies from objective motion incongruencies. A simple baseline model of [37] was modified with new weighting parameters and an extended transfer function. Results show a reasonable prediction for inputs which are used for the training of the model. Applying inputs which are not included in the training decreases the prediction quality substantially.

As the MCA_{OPT} was only applicable to open-loop simulations, a large effort was put in the real-time capability of the algorithm. The new developments were integrated to the MCA_{MPC} which is shown in Chapter 7. The MCA_{MPC} uses the model predictive control scheme which has to give feedback under hard real-time deadlines. The MCA_{MPC} was evaluated in comparison with the MCA_{BM} and the MCA_{OFF} using the CR method. Results show less perceived motion incongruencies for the MCA_{MPC} compared to the state-of-the-art MCA_{BM}. This is an important result as it indicates the potential of MCA_{MPC} and reveals an improvement in motion cueing quality for the closed-loop algorithm. The MCA_{OFF} demonstrates the least perceived motion incongruencies of the three considered MCAs. Showing the highest motion cueing quality is not surprising, since the MCA_{OFF} uses an ideal prediction strategy and the full time horizon as a reference. Taking the results of the MCA_{OFF} as an upper benchmark, a higher prediction accuracy along with a longer time horizon can substantially improve the motion cueing quality of the MCA_{MPC}. In the study, the MCA_{MPC} takes the current driving dynamic states as constant over the time horizon which is only a valid prediction for a short time instant. Thus, extending the time horizon length with this approach yields an increasingly inaccurate prediction as deviations to the current driving dynamics state are rising. The capability to use quadratic optimization approaches in motion cueing algorithms is a powerful approach. However, the optimization-problem can still become infeasible as the motion cueing highly depends on the choice of the hand-tuned weighting parameters which may not be the best choice for every driving manoeuvre.

Chapter 8 describes the structure of the MCA_{HYB} which combines the benefits of the MCA_{FILT} and the MCA_{OFF}. Saying it more precisely, the real-time capability of the MCA_{FILT} is used to make use of an optimal trajectory, calculated by the MCA_{OFF}, in a closed-loop simulation. An experimental analysis showed slight variations in driving dynamics data between different drives for a pre-defined manoeuvre concerning the lateral acceleration and the yaw angular velocity. Based on this data, a driving dynamics reference is obtained. The MCA$_{\mathrm{OFF}}$ uses the data as an input to calculate an optimal trajectory for the pre-defined and pre-recorded manoeuvre. This is only done for lateral acceleration, roll velocity and yaw velocity since the driving dynamic deviations between drivers are too large for the longitudinal acceleration. Thus, the longitudinal acceleration, the vertical acceleration and the pitch velocity are generated by a filter based MCA with pre-positioning derived from the test drives. In a closed-loop simulation, a function matches the position of the reference with that of the online driver. At the matching positions, the simulator provides the optimal trajectory. Driving dynamics deviations between reference and online driver are compensated with the MCA$_{\mathrm{FILT}}$. The MCA$_{\mathrm{HYB}}$ is evaluated in an experimental study consisting of two parts. In the first part, the CR was used which compares the MCA$_{\mathrm{HYB}}$ with the MCA$_{\mathrm{FILT}}$ in order to evaluate the developments and extensions of the MCA$_{\mathrm{HYB}}$. Ratings exhibit lower perceived motion incongruencies of the MCA$_{\mathrm{HYB}}$ compared to the MCA$_{\mathrm{FILT}}$ which indicates an improvement in motion cueing quality. The second part uses the section-wise post-hoc rating method in a closed-loop simulation. Participants gave an oral rating after pre-defined sections during an active drive. Again, minor perceived motion incongruencies are rated for the MCA$_{\mathrm{HYB}}$. Correlations between the section-wise post-hoc rating and the CR demonstrate the validity of the closed-loop rating method.

To give an overview of the motion cueing quality of all MCAs which are used within this work, Figure 9.1 shows the CR of the MCA$_{\mathrm{BM}}{}^1$ (used in Chapter 6.3.2), MCA$_{\mathrm{OPT}}$, MCA$_{\mathrm{BM}}{}^2$ (used in Chapter 7.3, same setting between MCA$_{\mathrm{BM}}{}^1$ and MCA$_{\mathrm{BM}}{}^2$), MCA$_{\mathrm{MPC}}$, MCA$_{\mathrm{OFF}}$, MCA$_{\mathrm{FILT}}$ and MCA$_{\mathrm{HYB}}$. The CR was measured in different experiments with different experimental designs. Thus, the following analysis gives only a bare indication about the perceived motion incongruencies for the manoeuvre described in Chapter 6.4.3. To further quantify the plots, Table 9.1 lists the rms-values of the CR over the complete manoeuvre. Based on these data, the following can be observed:

- The plots and the rms-values of the MCA$_{\mathrm{BM}}{}^1$ and the MCA$_{\mathrm{BM}}{}^2$ show a high correlation (Pearson correlation $r = 0.943$, $p < 0.001$). In both studies, described in Chapter 6.3.2 and 7.3, the same benchmark MCA is taken. This observation gives an indication that the CR-method is a valid procedure to rate motion incongruencies as the results of the CR are measured from two different groups. Along with this, it increases the significance of the following observations as it supports the comparability of the CRs between the experimental studies.

- The MCA$_{\mathrm{OFF}}$ shows the lowest rating of the considered CRs. This observation gives a good evidence that the motion cueing gained by MCA$_{\mathrm{OFF}}$ obtains the highest motion cueing quality of the considered MCAs in this work.

- The MCA$_{\mathrm{OPT}}$ shows less perceived motion incongruencies than the MCA$_{\mathrm{MPC}}$. This supports the result that a higher prediction accuracy and a longer time horizon can further improve the motion cueing quality of the MCA$_{\mathrm{MPC}}$.

- Comparing the CR of the MCA$_{\mathrm{HYB}}$ and the MCA$_{\mathrm{MPC}}$ confirms that both closed-loop MCAs are valid approaches to improve the motion cueing quality compared to state-of-the-art approaches.

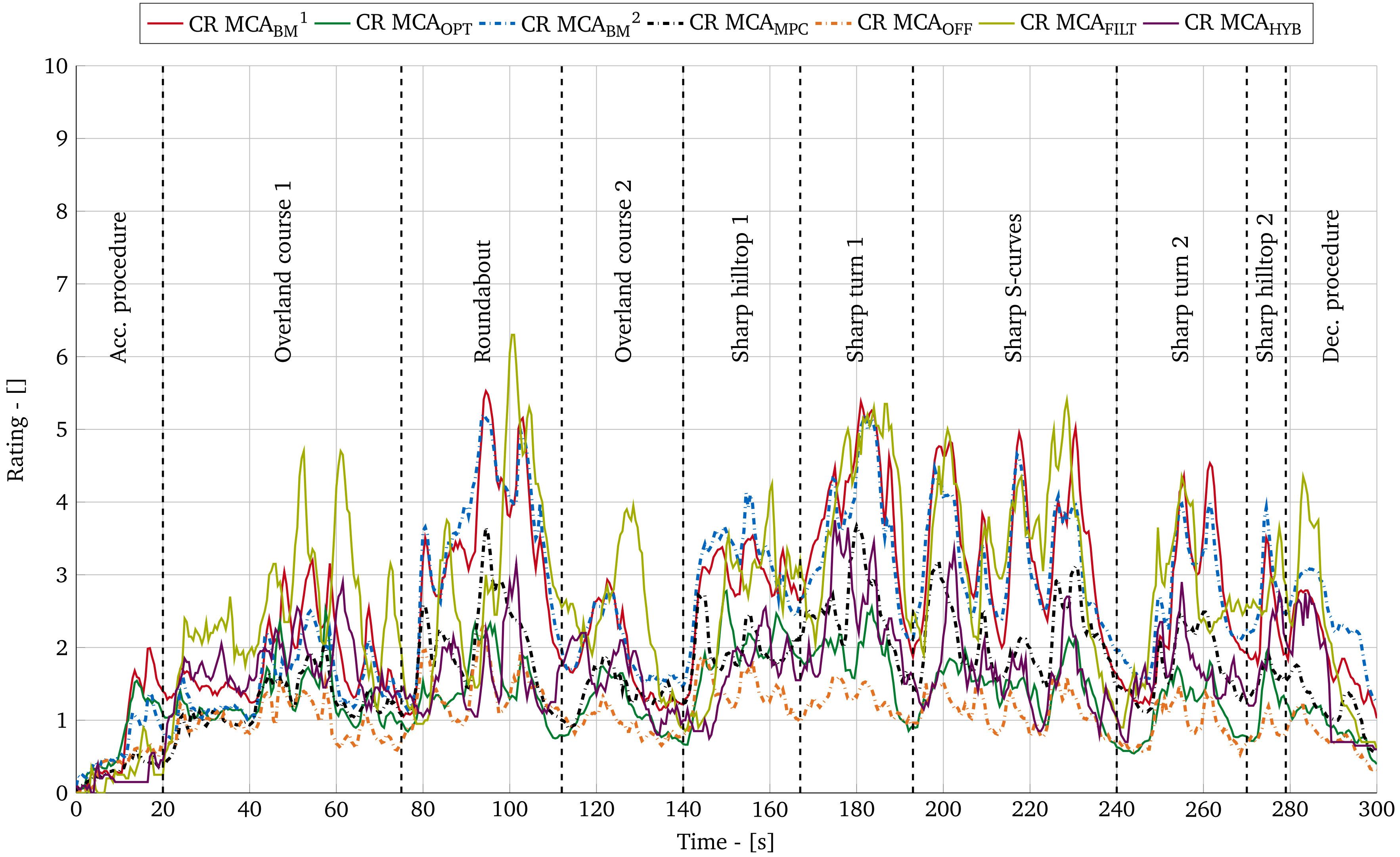


Figure 9.1: Mean CR for the MCA$_{BM}$^{1} (used in Chapter 6.3.2), MCA$_{OPT}$, MCA$_{BM}$^{2} (used in Chapter 7.3), MCA$_{MPC}$, MCA$_{OFF}$, MCA$_{FILT}$ and MCA$_{OFF}$^{2} over the complete manoeuvre. Dashed lines separate the manoeuvre into the seven sections as defined in Chapter 6.4.3.

Table 9.1: Rms-values of the $\text{MCA}_{\text{BM}}{}^{1}$ (used in Chapter 6.3.2), MCA_{OPT}, $\text{MCA}_{\text{BM}}{}^{2}$ (used in Chapter 7.3), MCA_{MPC}, MCA_{OFF}, MCA_{FILT} and MCA_{HYB}.

	$\text{MCA}_{\text{BM}}{}^{1}$	MCA_{OPT}	$\text{MCA}_{\text{BM}}{}^{2}$	MCA_{MPC}	MCA_{OFF}	MCA_{FILT}	MCA_{HYB}
rms-value	2.804	1.425	2.771	1.790	1.106	2.941	1.734

9.2 Recommendations for Future Work

From the experience gained through the work and under consideration of the results and discussion above, several suggestions for future research can be made:

- **Higher prediction accuracy of the MCA_{MPC}:**
  A limiting factor in the performance of the MCA_{MPC} is the prediction strategy which takes the current vehicle dynamics states constant over the prediction horizon. A prediction model could include the following components:

  - *Driving course model:* A use-case could be a pre-defined driving course for a closed-loop simulation which means that all turns, roundabouts, junctions etc. are known in advance. Based on this information, a driving course model could predict an average path of a driver (without driving dynamics). The model could use data from offline drives, such as described in Chapter 8.4.1, or could calculate the path online based on the current driving position. Typical path planning approaches of autonomous driving, such as [44, 103], could be applied.
  - *Driver model:* A model of the driver predicts the driver commands. The driver prediction is the most critical part of the overall prediction model as the time point and the extent of driver actions are unknown. The analysis in Chapter 8.2 provides good evidence for this observation, which shows large deviations in the longitudinal acceleration profiles for different drives. Though, a motion profile, which gives a tendency or probability over the considered time horizon, could be sufficient to give an estimate of the driver commands.
  - *Vehicle model:* Based on the outputs of the driver model, the vehicle model estimates the vehicle dynamics, such as the translational accelerations and angular velocities. A linear or nonlinear single track model could be a first approach [153].

  The following has to be considered: The motion cueing quality depends on the prediction accuracy, saying that a bad prediction yields a minor motion cueing quality. A limiting factor is the computational effort of the prediction model, meaning: The model needs to be simple enough to guarantee real-time capability for a closed-loop simulation. But, at the same time, the model has to be sufficiently reliable to predict the vehicle motions over a future time horizon. Another limiting factor are false predictions. For example in case the model predicts an acceleration manoeuvre but the driver is actually decelerating. Considering these large deviations between desired vehicle motions and actual motions in the objective function of the optimal control problem can lead to infeasibility of the optimization solver. The example demonstrates that the current driving dynamics states have to be considered in the prediction model as an initial condition.

- **Increase the prediction length of the MCA_{MPC}:**
  Along with a higher prediction accuracy, increasing the prediction length can further improve the motion cueing quality. The hard feedback deadlines of the real-time framework combined with uncertain calculation times of the optimizer are restricting factors.

In the thesis, the prediction model is integrated with an explicit Euler method which is a common technique in MPC-applications [32, 106]. Still, higher order integration methods enable larger time step sizes, yielding less optimization parameters over a consistent time horizon.

- **Optimization solver of the MCA$_{\text{MPC}}$:**
  An active-set QP-solver [65] with a hot-start function, explicitly developed for MPC real-time applications, is used in the MCA$_{\text{MPC}}$. Still, other real-time capable solvers are possible. Especially interior-point-based solvers may show advantages towards active-set methods because of a more efficient handling of the constraints. An analysis of different solvers concerning their real-time applicability for the considered problem can be part of future work.

- **Variation of the MCA$_{\text{FILT}}$ in the MCA$_{\text{HYB}}$:**
  The MCA$_{\text{HYB}}$ uses the filter-based MCA$_{\text{FILT}}$ for compensating deviations between the driving dynamics of the reference and the online driver. The MCA$_{\text{FILT}}$ is a simple-structured motion cueing approach which exhibits good tuning characteristics. In future works, other MCAs can be tested instead of the MCA$_{\text{FILT}}$. A possibility could be to use an optimization-based MCA which includes a prediction model and considers the constraints of the motion system.

- **Reference generation in the MCA$_{\text{HYB}}$:**
  As described in Chapter 8.4.1, the reference of the MCA$_{\text{HYB}}$ constitutes an average of several simulator drives. The process for preparing these drives with different participants for a pre-defined manoeuvre is quite cumbersome. An idea would be to use a computer-based driver, which includes a prediction model as described in the recommendations above. A limiting factor would be the accuracy of the prediction model which has a huge influence on the calculated optimal trajectory of the manoeuvre.

- **Reference-based MCA:**
  Extending the single reference-based approach, used by the MCA$_{\text{HYB}}$, with several references would yield an exclusively reference-based MCA. The idea is to calculate the optimal simulator trajectory for various drives of a pre-defined manoeuvre in order to obtain a motion cueing data base for the considered course. A neural network or a reinforcement learning algorithm uses the data to match the appropriate motion cueing data with the current state of the online driver. Thus, compared to the MCA$_{\text{HYB}}$ not only a single averaged reference is considered but an extensive data base. The data base could consider different driving styles, vehicle trajectories, traffic situations, etc..

- **Rating models:**
  The accuracy and the adaptivity of the rating models described in Chapter 6.4.4 are limited. Other approaches, such as neural networks, often show more accurate model-fitting characteristics. Rating methods which are based on neural networks are analysed within the work using the CR data gathered in the experimental studies. Results show a similar accuracy to the models described in Chapter 6.4.4, generally speaking: The accuracy for a training-set is significantly higher but validation tests, using non-training data, reduces the accuracy considerably. Insufficient training data could be the reason for the decrease in quality. Broadly extending the CR data base by using data from several experimental studies could improve the accuracy and adaptivity of these models.

Appendix A

Cardan Angles

In automotive industry, based on ISO8855 [92], Cardan rotations (or Tait-Bryan rotations) define a rotation sequence between the earth-fixed FoR I and the body-fixed FoR N. The Cardan angles ψ, θ, φ express a rotation sequence of three elementary rotations about the ${}_Iz - {}_1y - {}_2x-$ axis as shown in Figure A.1. Hereby, the 1 FoR and the 2 FoR are auxiliary coordinates systems. In this thesis, the Cardan sequence from an I FoR to an N FoR is defined as:

1. Rotation about the ${}_Iz$-axis with the angle ψ,
2. Rotation about the ${}_1y$-axis with the angle θ,
3. Rotation about the ${}_2x$-axis with the angle φ.

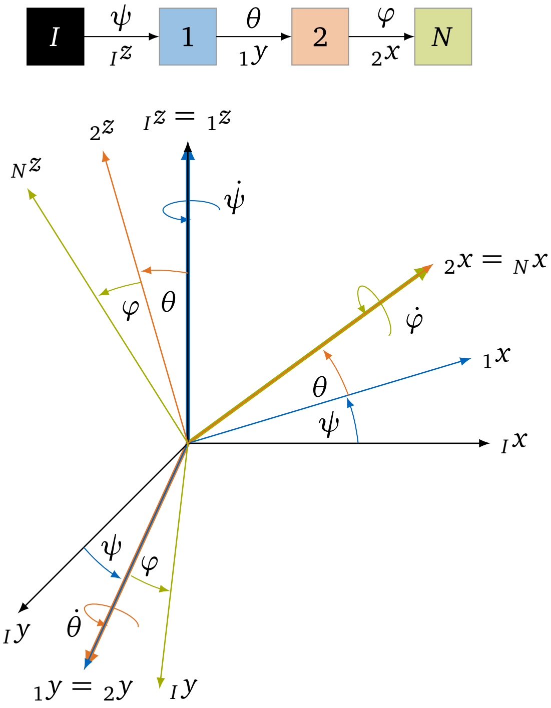


Figure A.1: Cardan sequence with the Cardan angles ψ, θ, φ.

According to Figure A.1, a Cardan sequence for a vector $\boldsymbol{r}$ can be expressed by

$$\begin{aligned}
{}_N\boldsymbol{r} &= {}_N\boldsymbol{R}_2\,{}_2\boldsymbol{R}_1\,{}_1\boldsymbol{R}_I\,{}_I\boldsymbol{r} \qquad &\text{(A.1)}\\
&= \boldsymbol{R}_x(\varphi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi)\,{}_I\boldsymbol{r} &\text{(A.2)}\\
&= {}_N\boldsymbol{R}_I\,{}_I\boldsymbol{r} &\text{(A.3)}\\
&= \begin{bmatrix} \mathrm{c}(\psi)\mathrm{c}(\theta) & \mathrm{c}(\theta)\mathrm{s}(\psi) & -\mathrm{s}(\theta) \\ \mathrm{c}(\psi)\mathrm{s}(\varphi)\mathrm{s}(\theta)-\mathrm{c}(\varphi)\mathrm{s}(\psi) & \mathrm{c}(\varphi)\mathrm{c}(\psi)+\mathrm{s}(\varphi)\mathrm{s}(\psi)\mathrm{s}(\theta) & \mathrm{c}(\theta)\mathrm{s}(\varphi) \\ \mathrm{s}(\varphi)\mathrm{s}(\psi)+\mathrm{c}(\varphi)\mathrm{c}(\psi)\mathrm{s}(\theta) & \mathrm{c}(\varphi)\mathrm{s}(\psi)\mathrm{s}(\theta)-\mathrm{c}(\psi)\mathrm{s}(\varphi) & \mathrm{c}(\varphi)\mathrm{c}(\theta) \end{bmatrix} {}_I\boldsymbol{r}. &\text{(A.4)}
\end{aligned}$$

with the abbreviations s and c for sin and cos, respectively. Rotations about the z-, y- and x- axis are defined as

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \text{ and}$$

$$\boldsymbol{R}_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\varphi) \end{bmatrix}. \tag{A.5}$$

An inversion of the rotation sequence is defined as

$$\begin{aligned}
{}_I\boldsymbol{R}_N &= ({}_N\boldsymbol{R}_I)^T \qquad &\text{(A.6)}\\
&= \boldsymbol{R}_z^T(\psi)\boldsymbol{R}_y^T(\theta)\boldsymbol{R}_x^T(\varphi) &\text{(A.7)}\\
&= \boldsymbol{R}_z(-\psi)\boldsymbol{R}_y(-\theta)\boldsymbol{R}_x(-\varphi). &\text{(A.8)}
\end{aligned}$$

Following the procedure above, the angular velocity $\boldsymbol{\omega}$ is obtained by the sum of relative angular velocities with the amplitudes $\dot\varphi$, $\dot\theta$, $\dot\psi$ about the relative axis. One finds

$$\begin{aligned}
\boldsymbol{\omega}_{IN} &= \boldsymbol{\omega}_{I1} + \boldsymbol{\omega}_{12} + \boldsymbol{\omega}_{2N} \qquad &\text{(A.9)}\\
&= \dot\psi\,\boldsymbol{e}_{Iz} + \dot\theta\,\boldsymbol{e}_{1y} + \dot\varphi\,\boldsymbol{e}_{2x}, &\text{(A.10)}
\end{aligned}$$

with a unit vector $\boldsymbol{e}$ defined in the respective FoR. The angular velocity in the N FoR is defined as

$$\begin{aligned}
{}_N\boldsymbol{\omega}_{IN} &= \dot\varphi\,{}_N\boldsymbol{e}_{2x} + \dot\theta\,{}_N\boldsymbol{e}_{1y} + \dot\psi\,{}_N\boldsymbol{e}_{Iz} \qquad &\text{(A.11)}\\
&= \dot\varphi\,{}_N\boldsymbol{R}_2\,{}_2\boldsymbol{e}_{2x} + \dot\theta\,{}_N\boldsymbol{R}_1\,{}_1\boldsymbol{e}_{1y} + \dot\psi\,{}_N\boldsymbol{R}_I\,{}_1\boldsymbol{e}_{Iz} &\text{(A.12)}\\
&= \left[\boldsymbol{R}_x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\ \boldsymbol{R}_x\boldsymbol{R}_y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ \boldsymbol{R}_x\boldsymbol{R}_y\boldsymbol{R}_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right] \begin{bmatrix} \dot\varphi \\ \dot\theta \\ \dot\psi \end{bmatrix} &\text{(A.13)}\\
&= \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\varphi) & \cos(\theta)\sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\theta)\cos(\varphi) \end{bmatrix} \dot{\boldsymbol{\beta}} &\text{(A.14)}\\
&= {}_N\boldsymbol{J}_\omega\,\dot{\boldsymbol{\beta}} &\text{(A.15)}
\end{aligned}$$

with the Jacobian matrix ${}_N\boldsymbol{J}_\omega$ and the Cardan angles' derivatives summed up in $\dot{\boldsymbol{\beta}} = [\dot\varphi,\, \dot\theta,\, \dot\psi]^T$.

Following equation (A.12), the angular velocity ${}_I\boldsymbol{\omega}_{IN}$ in the I-FoR can be described by

$$
\begin{aligned}
{}_I\boldsymbol{\omega}_{IN} &= \dot{\varphi}\,{}_I\boldsymbol{e}_{{}_2x} \;+\; \dot{\theta}\,{}_I\boldsymbol{e}_{{}_1y} \;+\; \dot{\psi}\,{}_I\boldsymbol{e}_{{}_Iz} && \text{(A.16)}\\
&= \left[\boldsymbol{R}_z^T\,\boldsymbol{R}_y^T \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\; \boldsymbol{R}_z^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\; \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right] \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} && \text{(A.17)}\\
&= \begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi) & 0 \\ \sin(\psi)\cos(\theta) & \cos(\psi) & 0 \\ -\sin(\theta) & 0 & 1 \end{bmatrix} \dot{\boldsymbol{\beta}} && \text{(A.18)}\\
&= {}_I\boldsymbol{J}_{\boldsymbol{\omega}}\,\dot{\boldsymbol{\beta}} && \text{(A.19)}
\end{aligned}
$$

In the present work, the Cardan angles between the T FoR and the H FoR, used for the simulator structure in Figure 2.4, are labelled with the subscripts TH to $\boldsymbol{\beta}_{TH} = [\varphi_{TH}, \theta_{TH}, \psi_{TH}]$.

Appendix B

Tripod's Actuator States

The Figure B.1 corresponds to Section 5.8.

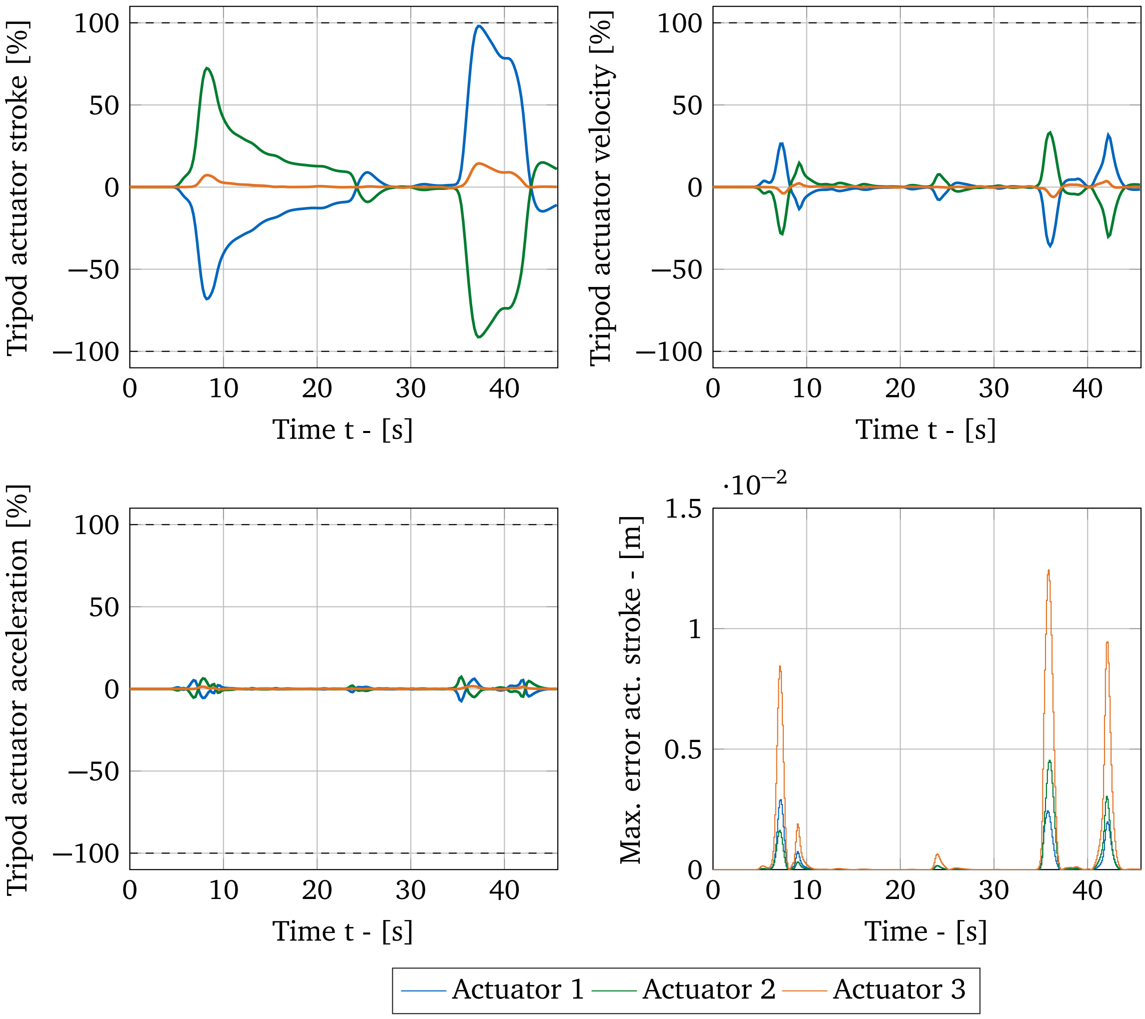


Figure B.1: Actuator stroke (top, left), velocity (top, right) and acceleration (bottom, left) of the tripod for the considered manoeuvre. The values are normalized to the maximum and minimum limits. The maximum error in the actuator stroke over the constraint horizon H_c (bottom right).

Appendix C

Optimization

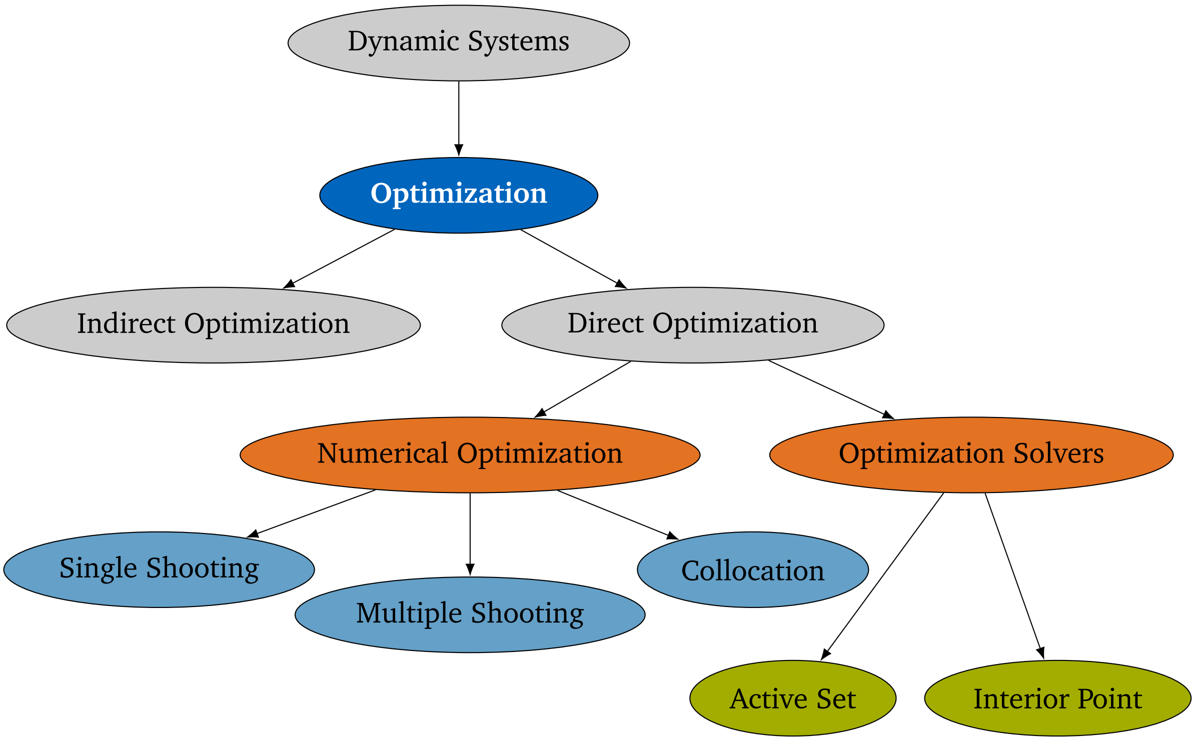


Figure C.1: Overview of numerical optimization approaches.

Figure C.1 shows an overview of numerical optimization methods which are introduced in the following Section. A model approximates the real system by a dynamic system of ordinary differential equations (ODE) in the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t) \qquad \forall t \in [t_0, t_e] \tag{C.1}$$

with the time t, the states $\boldsymbol{x}(t)$, the inputs $\boldsymbol{u}(t)$ and the time independent parameters $\boldsymbol{p}$. The time t is defined in an interval with the starting time t_0 and with the end time t_e. An optimal control problem (OCP) is defined as

$$\min_{\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{p},t_e} \quad J(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, \boldsymbol{p},\, t_e) \tag{C.2}$$

$$\begin{aligned} &\text{subject to:} \\ \forall t \in [t_0, t_e]: \quad &\dot{\boldsymbol{x}}(t) \;=\; \boldsymbol{f}(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, \boldsymbol{p},\, t_e) \\ &\boldsymbol{0} \;=\; \boldsymbol{R}(\boldsymbol{x}(t_0),\, \boldsymbol{u}(t_0),\, \boldsymbol{x}(t_e),\, \boldsymbol{u}(t_e),\, \boldsymbol{p},\, t_e) \\ &\boldsymbol{0} \;\leq\; \boldsymbol{S}(t,\, \boldsymbol{x}(t),\, \boldsymbol{u}(t),\, \boldsymbol{p},\, t_e) \end{aligned}$$

which minimizes the objective function J over the time interval $[t_0, t_e]$. The differential equations (C.1) are considered in the optimization as equality constraints. The vector $\boldsymbol{R}$ includes the initial and end conditions at the time $t_0 = 0$ and t_e for the system variables $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, $\boldsymbol{p}$, t_e. The vector $\boldsymbol{S}$ contains the system-, input-, parameter- and time-constraints.
Another problem class which is related to OCPs are boundary value problems (BVPs) which determine the solution of an ODE subject to the boundary conditions, for example at the initial state $t = t_0$ and at the final state $t = t_e$.
A distinction can be made between indirect and direct methods, when solving the problem described in equation (C.2). [177]
Indirect procedures pursue a "first optimize, then discretize" strategy by transforming the BVP into an initial value problem (IVP). The IVP is solved iteratively until the boundary conditions are fulfilled. Direct methods discretize the OCP to a finite-dimensional problem and then solve a discretized problem with numerical methods. Thus, a "first discretize, then optimize" strategy is pursued. Typical discretization methods are:

- Single shooting,
- Multiple Shooting and
- Collocation. [132]

C.1 Numerical Optimization

Due to the increasing performance of direct optimization solvers and the rising CPU power in recent years, the present work uses direct optimization methods. For further information concerning indirect methods, it is referred to literature, see e.g. [13, 132, 177].

Example: For the sake of simplicity and clarity, the following Section deals with an easier problem than described in equation (C.2). The example is taken from [16] and it determines the initial value $x(t_0)$ of a BVP, so that the IVP over $[t_0, t_e]$ fulfils the boundary condition at t_e in the form

$$\begin{aligned} \dot{x}(t) &= f(t, x(t)), \qquad \forall t \in [t_0, t_e] \\ c &= x(t_e). \end{aligned} \tag{C.3}$$

C.1.1 Single Shooting

The discretization scheme transforms the BVP, described in equation (C.3), to an IVP. As illustrated in Figure C.2, an iterative scheme with the loop index k is applied. The IVP is

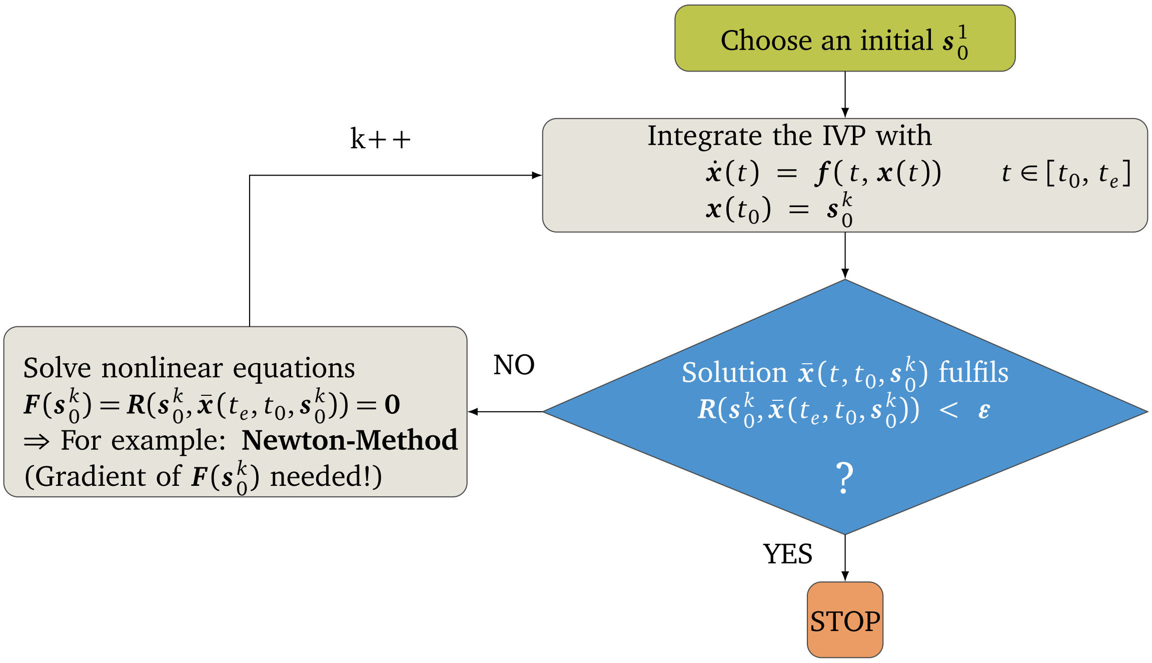

Choose an initial s_0^1
Integrate the IVP with
$\dot{x}(t) = f(t, x(t)) \quad t \in [t_0, t_e]$
$x(t_0) = s_0^k$
k++
Solution $\bar{x}(t, t_0, s_0^k)$ fulfils
$R(s_0^k, \bar{x}(t_e, t_0, s_0^k)) < \varepsilon$
?
NO
Solve nonlinear equations
$F(s_0^k) = R(s_0^k, \bar{x}(t_e, t_0, s_0^k)) = 0$
⇒ For example: **Newton-Method**
(Gradient of $F(s_0^k)$ needed!)
YES
STOP

Figure C.2: Single Shooting scheme to solve an IVP.

integrated over the time horizon $[t_0, t_e]$ with the initial guess $x(t_0) = s_0^k$. The trajectory $\bar{x}(t, t_0, s_0^k)$ yields the solution of the IVP (but $\bar{x}(t, t_0, s_0^k)$ not necessarily fulfils the final condition $c = x(t_e)$ of the BVP). In case the solution $\bar{x}(t, t_0, s_0^k)$ of the ODE in equation (C.3) fulfils the boundary conditions $R(s_0^k, \bar{x}(t_e, t_0, s_0^k)) < \varepsilon$ within a tolerance ε, the iteration scheme stops. Otherwise, a system of nonlinear equations $F(s_0^k) = R(s_0^k, \bar{x}(t_e, t_0, s_0^k)) = \mathbf{0}$ is solved, for instance, by applying the Newton-Method described in equation (2.35). This step is computationally expensive as the gradient of $F(s_0^k)$ is needed to determine the variation of the upcoming initial guess s_0^{k+1}. [16]

C.1.2 Multiple Shooting

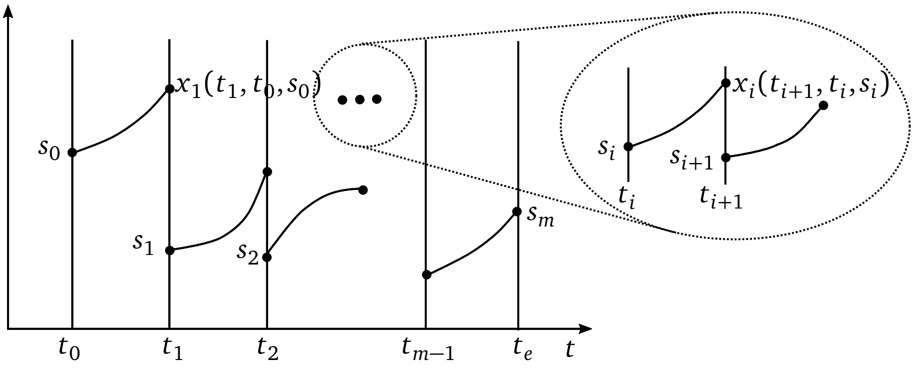


Figure C.3: Multiple Shooting scheme to solve an OCP (based on [16]).

In contrast to the single shooting method, the multiple shooting discretization divides the interval $[t_0, t_e]$ into m subintervals $[t_i, t_{i+1}]$, $i = 0, \dots, m-1$ with

$$t_0 < t_1 < \cdots < t_m = t_e, \tag{C.4}$$

as shown in Figure C.3. The time distance between the intervals can vary. For each subinterval, a parameter $s_i \in \mathbb{R}^n$ is introduced which defines a parametrized initial value on the interval $[t_i, t_{i+1}]$, $i = 0, \ldots, m-1$. Thus, compared to single shooting methods, $n \cdot m$ additional optimization parameters s_i are used. The ODE $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t))$ is integrated on each interval $[t_i, t_{i+1}]$ for the estimates s_i. The transition conditions

$$\boldsymbol{x}(t_{i+1}, t_i, \boldsymbol{s}_i) - \boldsymbol{s}_{i+1} = 0, \quad i = 0, \ldots, m-1 \tag{C.5}$$

are integrated as $n{\cdot}m$ additional boundary conditions compared to the single shooting method (cf. Figure C.3). The boundary conditions are transformed to the respective parameters $\boldsymbol{s}_0$, $\boldsymbol{s}_m$ in the form

$$\boldsymbol{R}(\boldsymbol{x}(t_0), \boldsymbol{x}(t_e)) = \boldsymbol{0} \iff \boldsymbol{R}(\boldsymbol{s}_0, \boldsymbol{s}_m) = \boldsymbol{0}, \quad \boldsymbol{R} \in \mathbb{R}^n. \qquad [16]$$

Similar to the iterative scheme of the single shooting (described in Figure C.2), the multiple shooting can also be solved by a Newton Method algorithm. The optimization parameters s_i can be integrated simultaneously on each subinterval $[t_i, t_{i+1}]$, $i = 0, \ldots, m-1$. The solution $\bar{\boldsymbol{x}}(t, t_i, \boldsymbol{s}_i), i = 0, \ldots, m-1$ fulfils the ODE at the subintervals t_i, $i = 0, \ldots, m$.

C.1.3 Collocation

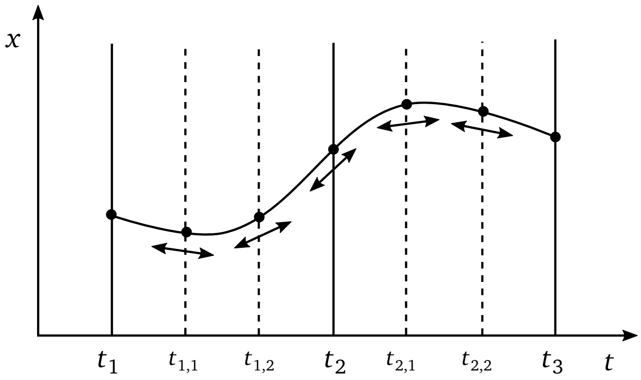


Figure C.4: Numerical discretization of a time interval using the collocation method (based on [16]).

The collocation method transforms the OCP to a subspace, so that the differential equations are fulfilled at the collocation points, as shown in Figure C.4. A time grid Π^N is introduced which divides the time horizon $[t_0, t_e]$ into N intervals

$$\Pi^N : \ t_0 < t_1 < \cdots < t_N = t_e \tag{C.6}$$

The intervals do not necessarily need to be equidistant. A further refinement of the interval $[t_i, t_{i+1}]$

$$0 \le \rho_0 < \rho_1 < \cdots < \rho_{k-1} \le 1 \tag{C.7}$$

leads to the collocation points, which are defined in the form

$$t_{ij} = t_i + \rho_j(t_{i+1} - t_i), \quad j = 0, \ldots, k-1 \tag{C.8}$$

for the interval $[t_i, t_{i+1}]$, $i = 0, \ldots, N-1$, resulting in $N \cdot k + 1$ collocation points. Discretizing the states $\boldsymbol{x}(t)$ and the inputs $\boldsymbol{u}(t)$ at the collocation points leads to the parameter vector $\boldsymbol{Y}$

$$\boldsymbol{Y} = [\boldsymbol{x}_{0,0}, \ldots, \boldsymbol{x}_{0,k}, \ldots, \boldsymbol{x}_{(N-1),0}, \ldots, \boldsymbol{x}_{(N-1),k}, \boldsymbol{x}_{N,0}], \qquad \in \mathbb{R}^{N(n_x \cdot k+1)}.$$

Different trial functions can be found in literature, such as [42, 58, 59, 100, 133, 178, 179]. Generally, polynomial functions or splines are introduced within the N-intervals:

- Polynomials of order d over the complete time horizon $[t_0, t_e]$,
- Piecewise polynomials of order d

$$\boldsymbol{x}_{\Pi_N}(\tau) = \sum_{l=0}^{d} c_{l,i} \left[\frac{\tau - t_i}{t_{i+1} - t_i} \right]^l \quad \text{for} \quad t_i \leq \tau \leq t_{i+1}. \tag{C.9}$$

Since the differential equations are approximated by trial functions, no integration method is used compared to shooting methods. The fulfilment of the differential equations at the collocation points is given by the dynamic constraints

$$\dot{\boldsymbol{x}}_{\Pi}(t_{ij}) = \boldsymbol{f}(t_{ij}, \boldsymbol{x}(t_{ij})), \quad 0 < i < N-1, \ \ 0 < j < k-1. \tag{C.10}$$

To avoid discontinuities at the interval intersections, equality constraints

$$\boldsymbol{x}_{\Pi_i}(t_{i+1}) = \boldsymbol{x}_{\Pi_{i+1}}(t_{i+1}), \quad 0 < i < N-1 \tag{C.11}$$

ensure continuity at the polynomial interfaces. Start conditions and ending conditions of the OCP are also included as constraints of the trial functions. [16]

C.1.4 Discussion

The pros and cons of the three discretization strategies are listed in Table C.1. The choice of the discretization method depends on the defined problem. The single shooting is a simple concept, where the stability highly depends on the initial value $\boldsymbol{s}_0$. Multiple shooting methods exhibit a higher complexity but generally show a better stability behaviour. Parallel computation methods and the exploitation of matrix structures can further decrease the computation time. Collocation methods are not based on a numerical integration scheme but on trial functions which approximate the differential equations. Trial functions ensure the adherence of the differential equations at the collocation points. Thus, the complexity and the accuracy of the method depends on the chosen discretization method. [16]

C.2 Optimization with Equality Constraints

The optimization problem is expressed as

$$\min_{x} \quad J(\boldsymbol{x}) \qquad \boldsymbol{x} \in \mathbb{R}^n, n < \infty \tag{C.12}$$

s. t.:

$$\boldsymbol{c}_e(\boldsymbol{x}) = 0 \tag{C.13}$$

with the objective function $J(\boldsymbol{x})$ and the equality constraints $\boldsymbol{c}_e(\boldsymbol{x})$. The scalar *Lagrange*-function with the *Lagrange*-multipliers $\boldsymbol{\lambda}$ can be described by

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = J(\boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{c}_e(\boldsymbol{x}) \tag{C.14}$$

As a condition for a stationary point, the gradient of the *Lagrange*-function exhibits a zero crossing at the desired point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$, which is defined as

$$\nabla_x \ L(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{G}_e^T(\boldsymbol{x})\boldsymbol{\lambda} = \boldsymbol{0} \tag{C.15}$$

$$\nabla_\lambda \ L(\boldsymbol{x}, \boldsymbol{\lambda}) = -\boldsymbol{c}_e(\boldsymbol{x}) = \boldsymbol{0} \tag{C.16}$$

Table C.1: Comparison of the pros (+) and cos (-) of the three discretization methods: single shooting, multiple shooting and collocation [16].

Single shooting	Multiple shooting	Collocation
+ Simple Concept + Simple Implementation	+ Integrations are separated, thus can be performed simultaneously + Exploitation of the Gradient's matrix structure to reduce the numerical effort + Higher stability for larger time horizons compared to single shooting	+ Polynomials fulfil $\mathbb{C}^l$ continuity (in case: actuator requirement) + Many options in the discretization/ approximation of the OCP
- Solution of IVP not guaranteed for bad $\boldsymbol{s}_0$ - Convergence of Newton-Method not guaranteed for bad $\boldsymbol{s}_0$ - Problems for large time horizons	- Higher complexity in implementation compared to single shooting - Larger system of equations compared to single shooting	- Polynomials can lead to undesirable oscillations - Solve large system of equations

with $\boldsymbol{g} = \nabla_{\boldsymbol{x}}\ J$ and $\boldsymbol{G}_e = \nabla_{\boldsymbol{x}}\ \boldsymbol{c}_e$. In order to determine whether the stationary point is a minimum, a sufficient condition is to verify the Hesse matrix $\boldsymbol{H}_L$ for positive definiteness. The Hesse matrix is defined to

$$\boldsymbol{H}_L = \nabla_{\boldsymbol{xx}}\ L(\boldsymbol{x},\boldsymbol{\lambda}) = \nabla_{\boldsymbol{xx}} J(\boldsymbol{x}) - \nabla_{\boldsymbol{xx}} \boldsymbol{\lambda}^T\ \boldsymbol{c}_e(\boldsymbol{x}). \tag{C.17}$$

The solution of the equations (C.15) and (C.16) yields the values $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$, for example by applying a Newton Method. A 1^{st}-order Taylor-series of the equations (C.15) and (C.16) obtains the stationary points in the form

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{G}_e^T(\boldsymbol{x})\boldsymbol{\lambda} + \boldsymbol{H}_L(\boldsymbol{x})\ [\bar{\boldsymbol{x}} - \boldsymbol{x}] - \boldsymbol{G}_e^T(\boldsymbol{x})[\bar{\boldsymbol{\lambda}} - \boldsymbol{\lambda}] \tag{C.18}$$

$$\boldsymbol{0} = -\boldsymbol{c}_e(\boldsymbol{x}) - \boldsymbol{G}_e(\boldsymbol{x})\ [\bar{\boldsymbol{x}} - \boldsymbol{x}] \tag{C.19}$$

Further transformations of the equations (C.18) and (C.19) lead to a linear system-of-equations, which is called Karush-Kuhn-Tucker (KKT) equations

$$\begin{bmatrix} \boldsymbol{H}_L(\boldsymbol{x}) & \boldsymbol{G}_e^T(\boldsymbol{x}) \\ \boldsymbol{G}_e(\boldsymbol{x}) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\rho} \\ -\bar{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g}(\boldsymbol{x}) \\ -\boldsymbol{c}_e(\boldsymbol{x}) \end{bmatrix} \tag{C.20}$$

with $\boldsymbol{\rho} = [\bar{\boldsymbol{x}} - \boldsymbol{x}]$. The system can be solved by a numerical solver for linear systems of equations. [13]

C.3 Optimization with Inequality Constraints

The optimization problem is defined as

$$\min_{\boldsymbol{x}} \quad J(\boldsymbol{x}) \qquad \boldsymbol{x} \in \mathbb{R}^n, n < \infty \tag{C.21}$$

s. t.:

$$\boldsymbol{c}_e(\boldsymbol{x}) = 0$$
$$\boldsymbol{c}_i(\boldsymbol{x}) \geq 0$$

Several algorithms are available for these problems. As depicted in Figure C.5, the following Section briefly introduces two commonly used solvers which are applied in the course of the thesis.

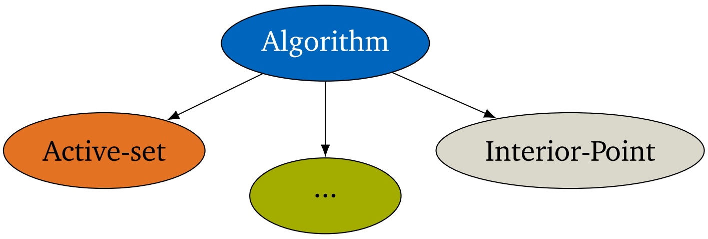


Figure C.5: Solvers for optimization problems with inequality constraints.

C.3.1 Active-Set Solver

A feasible region specifies all feasible points of a solution $\boldsymbol{x}^*$. By contrast, at least one point of a solution $\boldsymbol{x}^*$ violates the constraints in an infeasible region. At a solution $\boldsymbol{x}^*$, constraints can be distinguished into active and inactive:

1. *Active-set* $\mathbb{A}$ defines the constraints which are treated as equality constraints:

$$\boldsymbol{c}_i(\boldsymbol{x}^*) = 0 \quad \text{for} \quad i \in \mathbb{A}. \tag{C.22}$$

2. *Inactive-set* $\mathbb{A}'$ defines the constraints which are strictly satisfied for $\boldsymbol{x}^*$:

$$\boldsymbol{c}_i(\boldsymbol{x}^*) > 0 \quad \text{for} \quad i \in \mathbb{A}'. \tag{C.23}$$

Active-set constraints are integrated as equality constraints to the optimization problem, for example, by using the methods described in Section C.2. Inactive constraints can be neglected in the optimization step. An *active-set strategy* identifies the active constraints at the solution $\boldsymbol{x}^*$ by the Lagrange multipliers λ^* with

$$\lambda_i^* \geq 0 \quad \text{for} \quad i \in \mathbb{A}. \tag{C.24}$$

Consequently, in case $\lambda_i^* \geq 0$, the respective solution $\boldsymbol{x}_i^*$ is part of the active-set and in case $\lambda_i^* < 0$, the solution $\boldsymbol{x}_i^*$ can be deleted from the active set. [13]
Advantages of active-set methods are that only active-set constraints are considered in the optimization, which can fasten the optimization process. Since the active-set constraints can only change slightly between each iteration step, warm start approaches are possible. These

approaches use the solution of the previous step for the upcoming step, assuming that the solution only varies slightly. Thus, warm start options are often used in real-time applications. However, for some problems, active-set methods can be disadvantageous, for example when the active-set changes from step to step. In these cases, the constraint matrix varies which makes the solution computationally expensive. [125]

C.3.2 Interior-Point Solver

Interior-point methods introduce slack variables $\boldsymbol{s} \in \mathbb{R}^m$ to the optimal control problem by

$$\min_{\boldsymbol{x},\boldsymbol{s}} \qquad J(\boldsymbol{x}) - \mu \sum_{i=1}^{m} \ln(s_i) \tag{C.25}$$

s. t.:

$$\begin{aligned} \boldsymbol{c}_e(\boldsymbol{x}) &= 0 \\ \boldsymbol{c}_i(\boldsymbol{x}) - \boldsymbol{s} &= 0 \\ (\boldsymbol{s} &\geq \boldsymbol{0}) \end{aligned}$$

with the weighting μ. One possibility to integrate the slack variables to the objective function is by a natural logarithm function. As long as $\boldsymbol{s} \geq \boldsymbol{0}$, the inequality constraints $\boldsymbol{c}_i(\boldsymbol{x}) \geq 0$ are satisfied. The logarithmic term in the objective function prevents the slack variables $\boldsymbol{s}$ from becoming negative as the cost of J increases considerably when $\boldsymbol{s}$ converges to 0. In Figure C.6, a natural logarithm function is shown which demonstrates

$$\lim_{s \to 0} -\ln(s) \;\rightarrow\; \infty. \tag{C.26}$$

The convergence can be specified by modifications of the logarithmic term in the objective function or by the weighting μ. [125]

Interior-point methods usually show fast convergence properties as the major computational effort is spent on solving the KKT-equations. By exploiting the structure of the Jacobian matrix, an efficient linear solver can reduce the computation time. Still, it has to be taken into account that all constraints are considered in the calculation even if some are inactive. Another restriction is that for some solvers, the initial values have to be feasible which can be quite difficult for some problems. [125]

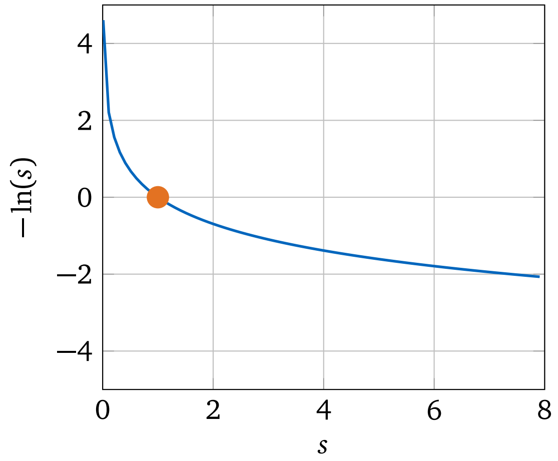


Figure C.6: Cost function $-\ln(s)$ of the slack variable s.

Appendix D

Weighting Parameters

The equations (5.39) and (5.40) show an approach to determine the weighting parameters. All parameters are additionally hand-tuned, depending on the motion system's workspace capabilities and the driving dynamics data.

Table D.1: Hand-tuned weightings parameters for the objective function in equation (5.37). Used for the simulative evaluation in Chapter 5.8.

Weighting matrix *Q*					
Specific forces and angular velocities					
f_x	103.8063	f_y	34.6021	f_z	12.7551
ω_x	364.7563	ω_y	506.6060	ω_z	485.6222
Weighting matrix *R*					
Hexapod					
x_H	19133	y_H	24000	z_H	30992
$\dot{x}_H$	375	$\dot{y}_H$	519	$\dot{z}_H$	586
φ_H	7386.3	θ_H	7386.3	ψ_H	7386.3
Tripod					
x_T	109.375	y_T	124.446	ψ_T	262.6250
$\dot{x}_T$	24.220	$\dot{y}_T$	31.108		
Weighting matrix *S*					
Hexapod					
$\ddot{x}_H$	2.4	$\ddot{y}_H$	2.4	$\ddot{z}_H$	1.23
$\dot{\varphi}_H$	162.09	$\dot{\theta}_H$	174.78	$\dot{\psi}_H$	162.09
Tripod					
$\ddot{x}_T$	0.276	$\ddot{y}_T$	0.4	$\dot{\psi}_T$	6.03
Weighting matrix *T*					
$\boldsymbol{T} = 100 \cdot \boldsymbol{S}$					

Table D.2: Hand-tuned weightings parameters for the objective function in equation (5.37). Used for the MCA$_{\text{OPT}}$ in the comparison between the MCA$_{\text{OPT}}$ and the MCA$_{\text{BM}}$ in Chapter 6.3.

Weighting matrix Q					
Specific forces and angular velocities					
f_x	69.2042	f_y	103.8063	f_z	1530.6
ω_x	364.7563	ω_y	506.6060	ω_z	971.2444
Weighting matrix R					
Hexapod					
x_H	19 133	y_H	24 000	z_H	10331
$\dot{x}_H$	375	$\dot{y}_H$	519	$\dot{z}_H$	195
φ_H	7386.3	θ_H	7386.3	ψ_H	7386.3
Tripod					
x_T	46.875	y_T	53.334	ψ_T	68.2825
$\dot{x}_T$	10.38	$\dot{y}_T$	13.332		
Weighting matrix S					
Hexapod					
$\ddot{x}_H$	2.4	$\ddot{y}_H$	2.4	$\ddot{z}_H$	0.246
$\dot{\varphi}_H$	117.065	$\dot{\theta}_H$	126.23	$\dot{\psi}_H$	117.065
Tripod					
$\ddot{x}_T$	0.1035	$\ddot{y}_T$	0.15	$\dot{\psi}_T$	1.206
Weighting matrix T					
$\boldsymbol{T = S}$					

Table D.4: Hand-tuned weightings parameters for the objective function in equation (4.6). Used for the MCA$_{\text{OFF}}$ in the comparison between the MCA$_{\text{FILT}}$ and the MCA$_{\text{OFF}}$ in Chapter 4.4.

Weighting matrix Q					
Specific forces and angular velocities					
f_x	0.1	f_y	0.1	f_z	0.1
ω_x	0.6	ω_y	0.6	ω_z	10
Weighting matrix R					
Hexapod					
x_H	0.25	$\dot{x}_H$	0.25	$\ddot{x}_H$	0.25
y_H	0.25	$\dot{y}_H$	0.25	$\ddot{y}_H$	0.25
z_H	0.25	$\dot{z}_H$	0.25	$\ddot{z}_H$	0.25
φ_H	3	$\dot{\varphi}_H$	25		
θ_H	3	$\dot{\theta}_H$	25		
ψ_H	2	$\dot{\psi}_H$	25		
Tripod					
x_T	0.03	$\dot{x}_T$	0.03	$\ddot{x}_T$	0.03
y_T	0.03	$\dot{y}_T$	0.03	$\ddot{y}_T$	0.03
ψ_T	0.3	$\dot{\psi}_T$	0.1		

Table D.5: Hand-tuned weightings parameters for the objective function in equation (4.6). Used for the MCA$_{\text{OFF}}$ in the comparison between the MCA$_{\text{MPC}}$, MCA$_{\text{BM}}$ and the MCA$_{\text{OFF}}$ in Chapter 7.2.

Weighting matrix Q					
Specific forces and angular velocities					
f_x	0.1	f_y	0.1	f_z	0.7
ω_x	1	ω_y	1	ω_z	70
Weighting matrix R					
Hexapod					
x_H	0.25	$\dot{x}_H$	0.25	$\ddot{x}_H$	0.25
y_H	0.25	$\dot{y}_H$	0.25	$\ddot{y}_H$	0.25
z_H	0.25	$\dot{z}_H$	0.25	$\ddot{z}_H$	0.25
φ_H	10	$\dot{\varphi}_H$	25		
θ_H	10	$\dot{\theta}_H$	25		
ψ_H	2	$\dot{\psi}_H$	25		
Tripod					
x_T	0.03	$\dot{x}_T$	0.03	$\ddot{x}_T$	0.03
y_T	0.03	$\dot{y}_T$	0.03	$\ddot{y}_T$	0.03
ψ_T	0.3	$\dot{\psi}_T$	0.1		

Table D.3: Hand-tuned weightings parameters for the objective function in equation (5.37). Used for the MCA$_{\text{MPC}}$ in the comparison between the MCA$_{\text{MPC}}$, MCA$_{\text{BM}}$ and the MCA$_{\text{OFF}}$ in Chapter 7.2.

Weighting matrix $\boldsymbol{Q}$					
Specific forces and angular velocities					
f_x	0.1	f_y	0.1	f_z	0.1
ω_x	1	ω_y	1	ω_z	1
Weighting matrix $\boldsymbol{R}$					
		Hexapod			
x_H	0.6	y_H	0.6	z_H	0.6
$\dot{x}_H$	0.4	$\dot{y}_H$	0.4	$\dot{z}_H$	0.3
φ_H	6	θ_H	6	ψ_H	40
		Tripod			
x_T	0.11	y_T	0.11	ψ_T	2
$\dot{x}_T$	0.01	$\dot{y}_T$	0.01		
Weighting matrix $\boldsymbol{S}$					
		Hexapod			
$\ddot{x}_H$	0.5	$\ddot{y}_H$	0.5	$\ddot{z}_H$	0.5
$\dot{\varphi}_H$	40	$\dot{\theta}_H$	40	$\dot{\psi}_H$	25
		Tripod			
$\ddot{x}_T$	0.01	$\ddot{y}_T$	0.01	$\dot{\psi}_T$	1
Weighting matrix $\boldsymbol{T}$					
$\boldsymbol{T} = \boldsymbol{S}$					

Table D.6: Hand-tuned weightings parameters for the objective function in equation (4.6). Used for the MCA$_{\text{OFF}}$ to calculate the reference trajectory in Chapter 8.5.3.

Weighting matrix $\boldsymbol{Q}$					
Specific forces and angular velocities					
f_x	0.1	f_y	0.1	f_z	0.1
ω_x	0.5	ω_y	0.5	ω_z	1
Weighting matrix $\boldsymbol{R}$					
		Hexapod			
x_H	0.25	$\dot{x}_H$	0.25	$\ddot{x}_H$	0.25
y_H	0.25	$\dot{y}_H$	0.25	$\ddot{y}_H$	0.25
z_H	0.25	$\dot{z}_H$	0.25	$\ddot{z}_H$	0.25
φ_H	0.5	$\dot{\varphi}_H$	25		
θ_H	0.5	$\dot{\theta}_H$	25		
ψ_H	2	$\dot{\psi}_H$	25		
		Tripod			
x_T	0.02	$\dot{x}_T$	0.02	$\ddot{x}_T$	0.02
y_T	0.02	$\dot{y}_T$	0.02	$\ddot{y}_T$	0.02
ψ_T	0.1	$\dot{\psi}_T$	0.2		

Bibliography

[1] Advisory Group for Aerospace Research and Development. *Fidelity of simulation for pilot training*. Tech. rep. North Atlantic Treaty Organisation, 1980.

[3] Allerton, D. *Principles of Flight Simulation*. John Wiley & Sons Ltd, 2009.

[4] Andersson, J. "A General-Purpose Software Framework for Dynamic Optimization". PhD thesis. Arenberg Doctoral School, KU Leuven, 2013.

[5] Augusto, B. and Loureiro, R. *Motion cueing in the Chalmers driving simulator-A model predictive control approach*. Master thesis. Chalmers University of Technology. 2009.

[6] Baarspul, M. "Flight simulation techniques with emphasis on the generation of high fidelity 6 DOF motion cues". In: *Proceedings of the 15th Congress of the International Council of the Areonautical Sciences* (Sept. 1986). London, UK.

[7] Baumann, G., Riemer, T., Liedecke, C., Rumbolz, P., and Schmidt, A. "How to build Europe's largest eight-axes motion simulator". In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2012). Paris, France.

[9] Beghi, A., Bruschetta, M., and Maran, F. "A real time implementation of MPC based Motion Cueing strategy for driving simulators". In: *Proceedings of the 51st IEEE Conference on Decision and Control* (2012). DOI: 10.1109/CDC.2012.6426119.

[10] Bemporad, A. "Reducing conservativeness in predictive control of constrained systems with disturbances". In: *Proceedings of the 37th IEEE Conference on Decision and Control* (1998). DOI: 10.1109/CDC.1998.758479.

[11] Berthoz, A., Bles, W., Bülthoff, H. H., Gràcio, B. J. C., Feenstra, P., Filliard, N., Hühne, R., Kemeny, A., Mayrhofer, M., Mulder, M., Nusseck, H. G., Pretto, P., Reymond, G., Schlüsselberger, R., Schwandtner, J., Teufel, H., Vailleau, B., Paassen, M. M. van, Vidal, M., and Wentink, M. "Motion Scaling for High-Performance Driving Simulators". In: *IEEE Transactions on Human-Machine Systems* 43 (May 2013), pp. 265–276. DOI: 10.1109/TSMC.2013.2242885.

[12] Bertin, R., Collet, C., Espié, S., and Graf, W. "Objective measurement of simulator sickness and the role of visual-vestibular conflict situations". In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (2005). Orlando, USA.

[13] Betts, J. T. *Practical methods for optimal control and estimation using nonlinear programming*. Siam, 2010.

[14] Bielaczek, C. "Die Auswirkung der aktiven Fahrerbeeinflussung auf die Fahrsicherheit (German)". In: *Automobiltechnische Zeitschrift 101* 9 (1999), pp. 714–724. DOI: 10.1007/BF03224277.

[15] Bittanti, S. *The Riccati equation*. Springer, 1991.

[16] Bock, H., Diehl, M., Kirches, C., Mombaur, K., and Sager, S. *Optimierung bei gewöhnlichen Differentialgleichungen (German)*. Tech. rep. University of Heidelberg, 2014.

[17] Boer, E., Yamamura, T., Kuge, N., and Girshick, A. “Experiencing the Same Road Twice: A Driver Centered Comparison between Simulation and Reality.” In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (2000). Paris, France.

[18] Bortz, J. and Doering, N. *Forschungsmethoden und Evaluation (German)*. Springer, 2002.

[19] Brazalez, A., Matey, L., Nunez, B., and Paul, A. “Simulation Tools”. In: *Intelligent Vehicles: Enabling Technologies and Future Developments*. Ed. by Jiménez, F. Butterworth-Heinemann, 2018.

[20] Bremer, H. and Pfeiffer, F. *Elastische Mehrkörpersysteme (German)*. Teubner Studienbücher Mechanik, 1992.

[22] Briot, S. and Khalil, W. *Dynamics of Parallel Robots*. Springer, 2015. DOI: 0.1007/978-3-319-19788-3s.

[23] Brooks, J. O., Goodenough, R. R., Crisler, M. C., Klein, N. D., Alley, R. L., Koon, B. L., Logan, W. C., Ogle, J. H., Tyrrell, R. A., and Wills, R. F. “Simulator sickness during driving simulation studies”. In: *Accident Analysis & Prevention* (May 2010), pp. 788–796. DOI: 10.1016/j.aap.2009.04.013.

[24] Bruschetta, M., Cenedese, C., and Beghi, A. “A real-time, MPC-based Motion Cueing Algorithm with Look-Ahead and driver characterization”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 61 (2017), pp. 38–52. DOI: 10.1016/j.trf.2017.04.023.

[25] Bruschetta, M. and Maran, F. “A nonlinear MPC based Motion Cueing implementation for a 9 DOFs dynamic simulator platform”. In: *53rd IEEE Conference on Decision and Control* (2014). DOI: 10.1109/CDC.2014.7039773.

[26] Bruschetta, M., Maran, F., and Beghi, A. “A fast implementation of MPC-based motion cueing algorithms for mid-size road vehicle motion simulators”. In: *Vehicle System Dynamics* (2017), pp. 802–826. DOI: 10.1080/00423114.2017.1280173.

[27] Bruschetta, M., Maran, F., and Beghi, A. “A Nonlinear, MPC-Based Motion Cueing Algorithm for a High-Performance, Nine-DOF Dynamic Simulator Platform”. In: *IEEE Transactions on Control Systems Technology* 25.2 (Mar. 2017). DOI: 10.1109/TCST.2016.2560120.

[28] Bruschetta, M., Maran, F., Beghi, A., and Minen, D. “An MPC approach to the design of motion cueing algorithms for a high performance 9 DOFs driving simulator”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2014). Paris, France.

[29] Buschmann, T. “Simulation and Control of Biped Walking Robots”. PhD thesis. Technische Universität München, Chair of Applied Mechanics, 2010.

[30] Butcher, J. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons Ltd, 2008.

[31] Butler, J., Smith, S., Beykirch, K., Bülthoff, H. H., et al. “Visual vestibular interactions for self motion estimation”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Oct. 2006). Paris, France.

[32] Camacho, E. F. and Bordons, C. *Model predictive control*. Springer, 2007.

[33] Chapron, T. and Colinot, J.-P. “The new psa peugeot-citroen advanced driving simulator overall design and motion cue algorithm”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2007). Iowa City, USA.

[34] Chen, S.-H. and Fu, L.-C. "An optimal washout filter design with fuzzy compensation for a motion platform". In: *IFAC Proceedings Volumes* 44 (2011), pp. 8433–8438. DOI: 10.3182/20110828-6-IT-1002.03176.

[35] Clark, L. A. and Watson, D. "Constructing validity: Basic issues in objective scale development". In: *Psychological Assessment* 7 (1995), pp. 309–319. DOI: 10.1037/1040-3590.7.3.309.

[36] Cleij, D., Venrooij, J., Pretto, P., Katliar, M., Bülthoff, H., Steffen, D., Hoffmeyer, F., and Schöner, H.-P. "Comparison between filter- and optimization-based motion cueing algorithms for driving simulation". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 61 (2017), pp. 53–68. DOI: 10.1016/j.trf.2017.04.005.

[37] Cleij, D., Venrooij, J., Pretto, P., Pool, D. M., Mulder, M., and Bülthoff, H. H. "Continuous Subjective Rating of Perceived Motion Incongruence During Driving Simulation". In: *IEEE Transactions on Human-Machine Systems* 48 (2017), pp. 17–29. DOI: 10.1109/THMS.2017.2717884.

[38] Cleij, D., Venrooij, J., Pretto, P., Pool, D., Mulder, M., and Bülthoff, H. "Continuous rating of perceived visual-inertial motion incoherence during driving simulation". In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2015). Tübingen, Germany.

[39] Cortes, P., Rodriguez, J., Silva, C., and Flores, A. "Delay Compensation in Model Predictive Current Control of a Three-Phase Inverter". In: *IEEE Transactions on Industrial Electronics* 59 (2012), pp. 1323–1325. DOI: 10.1109/TIE.2011.2157284.

[40] Cronbach, L. J. "Coefficient alpha and the internal structure of tests". In: *Psychometrika* 16 (1951), pp. 297–334. DOI: 10.1007/BF02310555.

[41] Dagdelen, M., Reymond, G., Kemeny, A., Bordier, M., and Maïzi, N. "Model-based predictive motion cueing strategy for vehicle driving simulators". In: *Control Engineering Practice* 17 (2009), pp. 995–1003. DOI: 10.1016/j.conengprac.2009.03.002.

[42] Dickmanns, E. D. and Well, K. H. "Approximate solution of optimal control problems using third order Hermite polynomial functions". In: *Optimization Techniques IFIP Technical Conference* (July 1975). Novosibirsk, pp. 158–166.

[43] Dobbeck, R. and Lincke, W. "Simulator for Volkswagen Cars". In: *ATZ Automobiltechnische Zeitschrift 76* 2 (1974), pp. 37–41.

[44] Dolgov, D., Thrun, S., Montemerlo, M., and Diebel, J. "Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)". In: *Ann Arbor* (June 2008). Chicago, USA.

[45] Drop, F., Olivari, M., Katliar, M., and Bülthoff. "Model Predictive Control Cueing: Online Prediction and Washout Tuning". In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2018). Antibes, France.

[46] Drosdol, J., Kading, W., and Panik, F. "The Daimler-Benz Driving Simulator". In: *Vehicle System Dynamics* 14 (1985), pp. 86–90. DOI: 10.1080/00423118508968804.

[57] Engen, T., Lervag, L.-E., and Moen, T. "Evaluation of IVIS/ADAS using Driving Simulators. Comparing Performance Measures in Different Environments". In: *16th ITS World Congress and Exhibition on Intelligent Transport Systems and Services* (Sept. 2009). Stockholm , Sweden.

[58] Enright, P. J. and Conway, B. A. "Discrete approximations to optimal trajectories using direct transcription and nonlinear programming". In: *Journal of Guidance, Control, and Dynamics* 15.4 (Oct. 1992). DOI: 10.2514/3.20934.

[59] Enright, P. J. and Conway, B. A. “Optimal finite-thrust spacecraft trajectories using collocation and nonlinear programming”. In: *Journal of Guidance, Control, and Dynamics* 14.5 (Oct. 1991). DOI: 10.2514/3.20739.

[60] Fang, Z., Tsushima, M., Kitahara, E., Machida, N., Wautier, D., and Kemeny, A. “Fast MPC based motion cueing algorithm for a 9 DOF driving simulator with a yaw table”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2017). Stuttgart, Germany.

[61] Fang, Z., Colombet, F., Collinet, J.-C., Kemeny, A., and Center, I. S. “Roll tilt thresholds for 8 DOF driving simulators”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2014). Paris, France.

[62] Fang, Z. and Kemeny, A. “An efficient MPC based motion cueing delay compensation algorithm for driving simulator”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2015). Tübingen, Germany.

[63] Feddersen, W. E., Sandel, T., Teas, D., and Jeffress, L. “Localization of High-Frequency Tones”. In: *The Journal of Acoustical Society of America* 29.9 (1957). DOI: 10.1121/1.1909356.

[65] Ferreau, H., Kirches, C., Potschka, A., Bock, H., and Diehl, M. “qpOASES: A parametric active-set algorithm for quadratic programming”. In: *Mathematical Programming Computation* 6.4 (Dec. 2014), pp. 327–363. DOI: 10.1007/s12532-014-0071-1.

[66] Fischer, M., Eriksson, L., and Oeltze, K. “Evaluation of methods for measuring speed perception in a driving simulator”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2012). Paris, France.

[67] Fischer, M. “Motion-cueing-Algorithmen für eine realitätsnahe Bewegungssimulation (German)”. In: *Berichte aus dem DLR-Institut für Verkehrssystemtechnik* 5 (2009). URL: http://www.digibib.tu-bs.de/?docid=00030516.

[68] Fischer, M., Sehammer, H., and Palmkvist, G. “Motion cueing for 3-, 6-and 8-degrees-of-freedom motion systems”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2010). Paris, France.

[69] Fortmüller, T. and Meywerk, M. “The influence of yaw movements on the rating of the subjective impression of driving”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Dec. 2005). Orlando, USA, pp. 362–313.

[70] Garrett, N. J. and Best, M. C. “Model predictive driving simulator motion cueing algorithm with actuator-based constraints”. In: *Vehicle System Dynamics* 51 (2013), pp. 1151–1172. DOI: 10.1080/00423114.2013.783219.

[71] Gianaros, P., Muth, E., J.T., M., Levine, M., and Stern, R. M. “A questionnaire for the assessment of the multiple dimensions of motion sickness”. In: *Aviat Space Environ Med.* 72 (2001), pp. 115–119.

[72] Glasauer, S. and Israël, I. “Otolithic Thresholds Influence the Perception of Passive Linear Displacement”. In: *Acta Otolaryngol* 115.sup520 (1995), pp. 41–44.

[73] Goldberg, J. M. and Fernandez, C. “Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. 3. Variations among units in their discharge properties.” In: *Journal of Neurophysiology* 34 (1971), pp. 676–684. DOI: 10.1152/jn.1971.34.4.676.

[74] Golding, J. F. “Motion sickness susceptibility”. In: *Autonomic Neuroscience* 129 (2006), pp. 67–76. DOI: 10.1016/j.autneu.2006.07.019.

[75] Gosselin, C. and Angeles, J. “Singularity analysis of closed-loop kinematic chains”. In: *IEEE Transactions on Robotics and Automation* 6.3 (June 1990), pp. 281–290. DOI: 10.1109/70.56660.

[76] Gough and Whitehall, S. G. “Universal Tyre Test Machine”. In: *Proceedings of 9th International Congress FISITA* (May 1962), pp. 117–137.

[77] VI-Grade. *VI-MotionCueing Documentation (not online - included in delivery of the DiM)*. 2016. URL: https://www.vi-grade.com/.

[78] Grant, P. and Reid, L. “Motion Washout Filter Tuning: Rules and Requirements”. In: *Journal of Aircraft* 34 (1997). DOI: 10.2514/2.2158.

[79] Greenberg, J. A. and Park, T. J. “The Ford driving simulator”. In: *SAE Technical Paper* (1994).

[80] Greig, G. L. “Masking of motion cues by random motion: comparison of human performance with a signal detection model”. PhD thesis. University of Toronto, 1988.

[81] Groen, E. and Bles, W. “How to use body tilt for the simulation of linear self motion”. In: *Journal of Vestibular Research* 14.5 (2004), pp. 375–385.

[82] Groen, J. and Jongkees, L. “The threshold of angular acceleration perception”. In: *The Journal of physiology* 107.1 (1948), pp. 1–7.

[83] Groot, S. de, Winter, J. C. de, Mulder, M., and Wieringa, P. A. “Nonvestibular motion cueing in a fixed-base driving simulator: Effects on driver braking and cornering performance”. In: *Presence: Teleoperators and Virtual Environments* 20.2 (2011), pp. 117–142. DOI: 10.1162/pres_a_00039.

[84] Gundry, A. “Thresholds to roll motion in a flight simulator”. In: *Journal of Aircraft* 14.7 (1977). DOI: 10.2514/3.58832.

[85] Hansson, P., Stenbeck, A., Kusachov, A., Bruzelius, F., and Augusto, B. “Prepositioning of driving simulator motion systems”. In: *International Journal of Vehicle Systems Modelling and Testing* 10.3 (2015). DOI: 10.1504/IJVSMT.2015.070165.

[86] Highstein, S. M., Fey, R. R., and Popper, A. N. *The Vestibular System*. Springer, 2004.

[87] Hosman, R. and Van der Vaart, J. *Vestibular models and thresholds of motion perception. Results of tests in a flight simulator*. Tech. rep. Delft University of Technology, 1978.

[89] Huang, C.-I. and Fu, L.-C. “Human vestibular based (HVB) senseless maneuver optimal washout filter design for VR-based motion simulator”. In: *IEEE International Conference on Systems, Man and Cybernetics* (July 2006). Taipei, Taiwan. DOI: 10.1109/ICSMC.2006.384835.

[90] Isermann, R. “Cascade Control Systems”. In: *Digital Control Systems*. Springer Berlin Heidelberg, 1991. DOI: 10.1007/978-3-642-86420-9_5.

[91] *ISO 1151-2:1985: Flight dynamics — Concepts, quantities and symbols — Part 2: Motions of the aircraft and the atmosphere relative to the Earth*. 1985.

[92] *ISO 8855: Road vehicles - Vehicle dynamics and road-holding ability - Vocabulary*. 2011.

[93] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*. Springer, 2014.

[94] Jamson, H., Horrobin, A. J., and Auckland, R. A. “Whatever Happened to the LADS? Design and Development of the New University of Leeds Driving Simulator”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (2007).

[95] Katliar, M., De Winkel, K., Venrooij, J., Pretto, P., and Bülthoff, H. “Impact of MPC Prediction Horizon on Motion Cueing Fidelity”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2015). Tübingen, Germany.

[96] Katliar, M., Drop, F., Teufel, H., Diehl, M., and Bülthoff, H. “Real-Time Nonlinear Model Predictive Control of a Motion Simulator Based on a 8-DOF Serial Robot”. In: *2018 European Control Conference (ECC)* (2018). Piscataway, USA, pp. 1529–1535. DOI: 10.23919/ECC.2018.8550041.

[97] Katliar, M., Olivari, M., Drop, F., Nooij, S., Diehl, M., and Bülthoff, H. “Offline Motion Simulation Framework: Optimizing Motion Simulator Trajectories and Parameters”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2018). Antibes, Paris.

[98] Kendig, K. “Is a 2000-Year-Old Formula Still Keeping Some Secrets?” In: *The American Mathematical Monthly* 107 (2000), pp. 402–415.

[99] Kingma, H. “Thresholds for perception of direction of linear acceleration as a possible evaluation of the otolith function”. In: *BMC Ear, Nose and Throat Disorders* 5 (June 2005). DOI: 10.1186/1472-6815-5-5.

[100] Kraft, D. “On converting optimal control problems into nonlinear programming problems”. In: *Computational mathematical programming*. Ed. by Schittkowski, K. Springer, Berlin, Heidelberg, 1985. DOI: 10.1007/978-3-642-82450-0_9.

[101] Kuge, N., Kubota, M., and Itoh, K. “A Study on a Motion Algorithm Based on a Driver-Centered Approach”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2002). Paris, France.

[102] Kurutas, S. *Modellierung der Dynamik eines Hexapods (German)*. Master thesis. Deutsches Zentrum für Luft- und Raumfahrt. 2006.

[103] Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., and How, J. P. “Real-time motion planning with applications to autonomous urban driving”. In: *IEEE Transactions on Control Systems Technology* 17.5 (2009), pp. 1105–1118. DOI: 10.1109/TCST.2008.2012116.

[105] LaViola Jr, J. J. “A discussion of cybersickness in virtual environments”. In: *ACM SIGCHI Bulletin* 32.1 (2000), pp. 47–56. DOI: 10.1145/333329.333344.

[106] Maciejowski, J. M. *Predictive control: with constraints*. Pearson education, 2002.

[107] MathWorks. *MATLAB Global Optimization Toolbox, R2016a*. 2016.

[108] *MATLAB Release 2015b, The MathWorks, Inc.,Natick, Massachusetts, United States*.

[109] *MATLAB Release 2016a, The MathWorks, Inc.,Natick, Massachusetts, United States*.

[111] Meehan, M., Razzaque, S., Whitton, M. C., and Brooks, F. P. “Effect of latency on presence in stressful virtual environments”. In: *virtual reality, 2003. Proceedings. IEEE* (2003). Los Angeles, USA. DOI: 10.1109/VR.2003.1191132.

[112] Meiry, J. L. and Young, L. R. “A revised dynamic otolith model”. In: *Aerospace Medicine* 39.6 (1968), pp. 606–608.

[113] Meiry, J. L. “The vestibular system and human dynamic space orientation”. PhD thesis. Massachusetts Institute of Technology, 1965.

[114] Merlet, J. P. “Determination of 6D Workspaces of Gough-Type Parallel Manipulator and Comparison between Different Geometries”. In: *The International Journal of Robotics Research* 19.9 (1999). DOI: 10.1177/02783649922066646.

[115] Merlet, J. P. *Parallel robots*. Springer Science & Business Media, 2012.

[116] Mittelstaedt, H. “A new solution to the problem of the subjective vertical”. In: *Naturwissenschaften* 70.272 (1983), pp. 272–281. DOI: 10.1007/BF00404833.

[117] Moler, C. and Van Loan, C. “Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later”. In: *Siam Review* 45.1 (2003), pp. 3–49. DOI: 10.1137/S00361445024180.

[118] Mollenhauer, M. A. *Simulator adaptation syndrome literature review*. Tech. rep. Realtime Technologies Inc., 2004.

[119] Mourant, R. R. and Thattacherry, T. R. “Simulator sickness in a virtual environments driving simulator”. In: *Proceedings of the human factors and ergonomics society annual meeting* 44.5 (2000), pp. 534–537. DOI: 10.1177/154193120004400513.

[121] Murgovski, N. *Vehicle modelling and washout filter tuning for the chalmers vehicle simulator*. Master thesis. Lund University, IEA. 2007.

[122] Nahon, M. and Reid, L. “Simulator motion-drive algorithms - A designer’s perspective”. In: *Journal of Guidance, Control, and Dynamics* 13.2 (1990). DOI: 10.2514/3.20557.

[123] Nehaoua, L., Mohellebi, H., Amouri, A., Arioui, H., Espie, S., and Kheddar, A. “Design and Control of a Small-Clearance Driving Simulator”. In: *IEEE Transactions on Vehicular Technology* 57.2 (2008), pp. 736–746. DOI: 10.1109/TVT.2007.905336.

[124] Nesti, A., Masone, C., Barnett-Cowan, M., Giorano, P. R., Bülthoff, H. H., and Pretto, P. “Roll rate thresholds and perceived realism in driving simulation”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2012). Paris, France.

[125] Nocedal, J. and Wright, S. J. *Numerical optimization*. Springer New York, 2006.

[126] Nordmark, S., H., J., Listrom, M., and Palmkvist, G. “A moving base driving simulator with wide angle visual system”. In: *Proceedings of the TRB conference session on "Simulation and instrumentation for the 80s", 64th Annual Meeting, Transportation Research Boarded* (1985). Washington, USA.

[127] Nunnally, J. C. *Psychometric theory*. McGraw-Hill, 1978.

[129] Oppenheim, A. V., Willsky, A. S., and Hamid, S. *Signals and Systems*. Pearson education, 2014.

[130] Ormsby, C. C. and Young, L. R. “Integration of semicircular canal and otolith information for multisensory orientation stimuli”. In: *Mathematical Biosciences* 34.1-2 (1977), pp. 1–21. DOI: 10.1016/0025-5564(77)90032-3.

[131] Ormsby, C. C. “Model of human dynamic orientation”. PhD thesis. Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 1974. URL: http://hdl.handle.net/1721.1/9857.

[132] Papageorgiou, M., Leibold, M., and Buss, M. *Optimierung - Statische, dynamische, stochastische Verfahren für die Anwendung (German)*. Springer, 2015.

[133] Paris, S. W. and Hargraves, C. R. “Direct trajectory optimization using nonlinear programming and collocation”. In: *Journal of Guidance, Control, and Dynamics* 10.4 (1987). DOI: 10.2514/3.20223.

[134] Pfeiffer, F. and Schindler, T. *Introduction to Dynamics*. Springer, 2015.

[135] Pitz, J. *Vorausschauender Motion-Cueing-Algorithmus für den Stuttgarter Fahrsimulator (German)*. Springer Vieweg, 2017.

[136] Pretto, P., Nusseck, H.-G., Teufel, H., and Bülthoff, H. “Effect of lateral motion on drivers’ performance in the MPI motion simulator”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Feb. 2009). Monaco.

[137] Pretto, P., Venrooij, J., Nesti, A., and Bülthoff, H. H. “Perception-Based Motion Cueing: A Cybernetics Approach to Motion Simulation”. In: *Recent Progress in Brain and Cognitive Engineering*. Ed. by Lee, S.-W., Bülthoff, H. H., and Müller, K.-R. Dordrecht: Springer Netherlands, 2015, pp. 131–152. DOI: 10.1007/978-94-017-7239-6_9.

[138] Proske, U. and Gandevia, S. C. “The kinaesthetic senses”. In: *The Journal of physiology* 587 (2009), pp. 4139–4146. DOI: 10.1113/jphysiol.2009.175372.

[139] Qaisi, I. and Treachtler, A. “Constrained linear quadratic optimal controller for motion control of ATMOS driving simulator”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2012). Paris, France. DOI: 10.1109/ICSMC.2012.6378073.

[140] Reed, M. and Green, P. “Comparison of driving performance on-road and in a low-cost simulator using a concurrent telephone dialling task”. In: *Ergonomics* 42.8 (1999), pp. 1015–1037. DOI: 10.1080/001401399185117.

[141] Reid, L. and Nahon, M. A.-. *Flight simulation motion-base drive algorithms: part 1. Developing and testing equations*. Tech. rep. University of Toronto, 1985.

[142] Reid, L. and Nahon, M. A. “Flight Simulation Motion-Base Drive Algorithms.: Part 2, Selecting The System Parameters”. In: *Utias Report* (1986). University of Toronto Institute for Aerospace Studies.

[143] Reymond, G., Kemeny, A., Droulez, J., and Berthoz, A. “Contribution of a motion platform to kinesthetic restitution in a driving simulator”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 1999). Paris, France.

[144] Reymond, G. and Kemeny, A. “Motion cueing in the Renault driving simulator”. In: *Vehicle System Dynamics* 34.4 (2000), pp. 249–259. DOI: 10.1076/vesd.34.4.249.2059.

[145] Riccio, G. E. and Stoffregen, T. A. “An ecological theory of motion sickness and postural instability”. In: *Ecological psychology* 3.3 (1991), pp. 195–240. DOI: 10.1207/s15326969eco0303_2.

[146] Riebe, S. “Aktive Schwingungsisolierung und Bahnregelung von Hexapodsystemen (German)”. PhD thesis. Technische Universität München, Chair of Applied Mechanics, 2005.

[147] Romano, R., Sadraei, E., and Markkula, G. “Rapid Tuning of the Classical Motion Cueing Algorithm”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2017). Stuttgart, Germany.

[148] Romano, R., Markkula, G., Boer, E. R., Jamson, A. H., Bean, A., Tomlinson, A., Horrobin, A., and Sadraei, E. “An Objective Assessment of the Utility of a Driving Simulator for Low Mu Testing”. In: *CoRR* abs/1812.02945 (2018). URL: http://arxiv.org/abs/1812.02945.

[149] Sadraei, E., Romano, R., Advani, S., Jamson, A., Chappell, P., Markkula, G., Bean, A., and Boer, E. “Understanding Cue Utility in Controlled Evasive Driving Manoeuvres: Optimizing Vestibular Cues for Simulator & Human Abilities”. In: *IFAC-PapersOnLine* 49.19 (2016). 13th IFAC Symposium on Analysis, Design, and Evaluation ofHuman-Machine Systems HMS 2016, pp. 414–419. DOI: https://doi.org/10.1016/j.ifacol.2016.10.601.

[150] Sammet, T. *Motion-Cueing-Algorithmen für die Fahrsimulation (German)*. Fortschritt-Berichte VDI: Reihe 12, Verkehrstechnik/Fahrzeugtechnik. VDI-Verlag, 2007.

[151] Schmidt, S. F. and Conrad, B. *Motion drive signals for piloted flight simulators*. Tech. rep. Washington, USA. NASA, 1970.

[152] Schöner, H. "Erprobung und Absicherung im dynamischen Fahrsimulator (German)". In: *SIMVEC - Simulation und Erprobung in der Fahrzeugentwicklung: Berechnung, Prüfstands- und Straßenversuch* 14 (2014). VDI.

[153] Schramm, D., Hiller, M., and Bardini, R. *Vehicle Dynamics - Modeling and Simulation*. Springer, 2017, pp. 223–236.

[154] Schwienbacher, M. "Efficient Algorithms for Biped Robots - Simulation, Collision Avoidance and Angular Momentum Tracking". PhD thesis. Technische Universität München, Chair of Applied Mechanics, 2013.

[156] Sharma, K., Haddadin, S., Heindl, J., Bellmann, T., Parusel, S., Rokahr, T., Minning, S., and Hirzinger, G. "Serial Kinematics based Motion Simulator - Evaluation of safety of the Passenger". In: *The 7th International Conference on the Safety of Industrial Automated Systems (SIAS-2012)* (2012). Montreal, Canada.

[157] Al-Shihabi, T. and Mourant, R. R. "A framework for modeling human-like driving behaviors for autonomous vehicles in driving simulators". In: *Proceedings 5th International Conference on Autonomous agents* (June 2001), pp. 286–291.

[158] Shin, K. G. and Ramanathan, P. "Real-time computing: A new discipline of computer science and engineering". In: *Proceedings of the IEEE* 82.1 (1994), pp. 6–24. DOI: 10.1109/5.259423.

[159] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. *Robotics - Modelling, Planning and Control*. Springer-Verlag London, 2009.

[160] Siciliano, B. and Khatib, O. *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[161] Sivan, R., Ish-Shalom, J., and Huang, J.-K. "An optimal control approach to the design of moving flight simulators". In: *IEEE Transactions on Systems, Man, and Cybernetics* 12.6 (1982), pp. 818–827. DOI: 10.1109/TSMC.1982.4308915.

[162] Slob, J. "State-of-the-Art driving simulators, a literature survey". In: *DCT report, Eindhoven University of Technology* (2008).

[164] Steinhausen, W. "Über den Nachweis der Bewegung der Cupula in der intakten Bogengangsampulle des Labyrinthes bei der natürlichen rotatorischen und calorischen Reizung (German)". In: *Pflüger's Archiv für die gesamte Physiologie des Menschen und der Tiere* 228.1 (Dec. 1931), pp. 322–328. DOI: 10.1007/BF01755501.

[165] Stewart, D. "A platform with six degrees of freedom". In: *Proceedings of the institution of mechanical engineers* 180.1 (1965), pp. 371–386. DOI: 10.1243/PIME_PROC_1965_180_029_02.

[166] Streiner, D. L. "Starting at the beginning: An introduction to coefficient alpha and internal consistency". In: *Journal of Personality Assessment* (2003). DOI: 10.1207/S15327752JPA8001_18.

[167] Suikat, R. *The new dynamic driving simulator at DLR*. Tech. rep. Institute of Transportation Systems, Deutsches Zentrum für Luft- und Raumfahrt, 2005.

[168] Telban Robert J abd Houck, J. A. and Cardullo, F. M. "Motion cueing algorithm development: Human-centered linear and nonlinear approaches". In: *NASA Scientific and Technical Information* (2005).

[169] Telban, R. J., Wu, W., Cardullo, F. M., and Houck, J. A. “Motion cueing algorithm development: Initial investigation and redesign of the algorithms”. In: *National Aeronautics and Space Administration (NASA)* (2000).

[170] The University of Iowa, USA. 13.07.2016. The National Advanced Driving Simulator. URL: https://www.nads-sc.uiowa.edu/sim_nads1.php.

[171] Tortora, G. J. *Principles of anatomy and physiology*. John Wiley & Sons, 2009.

[173] Treisman, M. “Motion sickness: an evolutionary hypothesis”. In: *Science* 197.4302 (1977), pp. 493–495. DOI: 10.1126/science.301659.

[174] Valjamae, A., Larsson, P., Vastfjall, D., and Kleiner, M. “Travelling without moving: Auditory scene cues for translational self-motion”. In: *Meeting of the International Conference on Auditory Display* (2005). URL: http://hdl.handle.net/1853/50193.

[175] Van Egmond, A., Groen, J., and Jongkees, L. “The mechanics of the semicircular canal”. In: *The Journal of physiology* 110.1-2 (1949), pp. 1–17. DOI: 10.1113/jphysiol.1949.sp004416.

[176] Venrooij, J., Cleij, D., Katliar, M., Pretto, P., Bülthoff, H., Steffen, D., Hoffmeyer, F., and Schöner, H. “Comparison between filter-and optimization-based motion cueing in the Daimler Driving Simulator”. In: *Proceedings of the Dr* (Sept. 2016). Paris, France.

[177] von Stryk, O. and Bulirsch, R. “Direct and indirect methods for trajectory optimization”. In: *Annals of Operations Research* 37.1 (Dec. 1992), pp. 357–373. DOI: 10.1007/BF02071065.

[178] von Stryk, O. *Numerical solution of optimal control problems by direct collocation*. Springer, 1993.

[179] von Stryk, O. “Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen (German)”. In: *VDI Reihe* 441 (1994).

[180] Wächter, A. and Biegler, L. T. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. DOI: 10.1007/s10107-004-0559-y.

[181] Wagner, P., Zöller, C., and Winner, H. “Safety Analysis for Wheeled Mobile Driving Simulators”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2016). Paris, France.

[182] Walch, M., Frommel, J., Rogers, K., Schüssel, F., Hock, P., Dobbelstein, D., and Weber, M. “Evaluating VR Driving Simulation from a Player Experience Perspective”. In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (May 2017). Denver, USA, pp. 2982–2989.

[183] Warren, W. H. “Self-motion: Visual perception and visual control”. In: *Perception of space and motion* (1995), pp. 263–325. DOI: 10.1016/B978-012240530-3/50010-9.

[184] Wei, C., Romano, R., Merat, N., Wang, Y., Hu, C., Taghavifar, H., Hajiseyedjavadi, F., and Boer, E. R. “Risk-based autonomous vehicle motion control with considering human driver’s behaviour”. In: *Transportation Research Part C: Emerging Technologies* 107 (2019), pp. 1–14. DOI: https://doi.org/10.1016/j.trc.2019.08.003.

[185] Weiß, C. *Control of a Dynamic Driving Simulator: Time-Variant Motion Cueing Algorithms and Prepositioning*. Master thesis. Deutsches Zentrum für Luft- und Raumfahrt. 2006.

[186] Wertheim, A. H., Mesland, B. S., and Bles, W. “Cognitive suppression of tilt sensations during linear horizontal self-motion in the dark”. In: *Perception* 30.6 (2001), pp. 733–741. DOI: 10.1068/p3092.

[187] Wieland, A., Durach, C. F., Kembro, J., and Treiblmaier, H. “Statistical and judgmental criteria for scale purification”. In: *Supply Chain Management: An International Journal* 22.4 (2017), pp. 321–328. DOI: 10.1108/SCM-07-2016-0230.

[188] Young, L. R. and Oman, C. M. “Part II. A model for vestibular adaptation to horizontal rotation”. In: *NASA* 40.10 (1970), pp. 1076–1080.

[189] Zacharias, G. L. *Motion cue models for pilot-vehicle analysis*. Tech. rep. Aerospace medical research Laboratory, 1978.

[190] Zeeb, E. “Daimler’s new full-scale, high-dynamic driving simulator–a technical overview”. In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2010). Paris, France.

[191] Zenner, H. P. “Der Gleichgewichtssinn und die Bewegungs- und Lageempfindung des Menschen (German)”. In: *Physiologie des Menschen*. Ed. by Schmidt, R. F., Lang, F., and M., H. Springer-Lehrbuch. Springer, Berlin, Heidelberg, 1999. Chap. 17, pp. 251–258. DOI: 10.1007/978-3-642-01651-6_17.

Publications by the Author

[47] Ellensohn, F., Breyer, M., Schwienbacher, M., Venrooij, J., and Rixen, D. "A Filter-Based Motion Cueing Algorithm for a Redundant Driving Simulator". In: *Proceedings in Mathematics and Applied Mechanics* 18.1 (2018). DOI: 10.1002/pamm.201800445.

[48] Ellensohn, F., Hristakiev, D., Schwienbacher, M., Venrooij, J., and Rixen, D. "Evaluation of an Optimization Based Motion Cueing Algorithm Suitable for Online Application (accepted)". In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2019). Strasbourg, France.

[49] Ellensohn, F., Oberleitner, F., Schwienbacher, M., Venrooij, J., and Rixen, D. "Actuator-Based Optimization Motion Cueing Algorithm". In: *Proceedings in 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (July 2018). Auckland, New Zealand, pp. 1021–1026. DOI: 10.1109/AIM.2018.8452464.

[50] Ellensohn, F., Schwienbacher, M., Venrooij, J., and Rixen, D. "Comparison of a Filter- and a Model Predictive Control Based Motion Cueing Algorithm". In: *Proceedings in Mathematics and Applied Mechanics* 17.1 (2017), pp. 787–788. DOI: 10.1002/pamm.201710361.

[51] Ellensohn, F., Schwienbacher, M., Venrooij, J., and Rixen, D. "Motion Cueing Algorithm for a 9 DoF Driving Simulator: MPC with Linearized Actuator Constraints". In: *SAE Technical Papers* (Apr. 2018). Detroit, USA. DOI: 10.4271/2018-01-0570.

[52] Ellensohn, F., Spannagl, M., Agabekov, S., Venrooij, J., Schwienbacher, M., and Rixen, D. "A Hybrid Motion Cueing Algorithm (in review process)". In: *Control Engineering Practice* (2019).

[53] Ellensohn, F., Venrooij, J., Schwienbacher, M., and Rixen, D. "Comparison of MCAs Based on Optimal Filters and on MPC: Influence of the Time Horizon". In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2017). Suttgart, Germany.

[54] Ellensohn, F., Venrooij, J., Schwienbacher, M., and Rixen, D. "Experimental evaluation of an optimization-based motion cueing algorithm". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 62 (2018), pp. 115–125. DOI: 10.1016/j.trf.2018.12.004.

[55] Ellensohn, F., Venrooij, J., Schwienbacher, M., and Rixen, D. "Experimental evaluation of an optimization-based motion cueing algorithm (short version)". In: *Proceedings of the Driving Simulation Conference, Driving Simulation Association* (Sept. 2018). Antibes, France.

[56] Ellensohn, F., Schwienbacher, M., Venrooij, J., and Rixen, D. "Motion Cueing Algorithm for a 9-DoF Driving Simulator: MPC with Linearized Actuator Constraints". In: *SAE International Journal Connected and Automated Vehicles* 2.3 (July 2019). DOI: 10.4271/12-02-03-0010.

Supervised Student Theses

The following student theses were supervised by the author at the chair of applied mechanics and may have been incorporated into this thesis.

[2] Agabekov, S. *Implementierung und Evaluierung eines hybriden Motion Cueing Algorithmus (German)*. Master thesis. Technische Universität München. 2019.

[8] Baur, M. *Motion Cueing Algorithmus zur skalierten Querdynamik-Wiedergabe an einem Fahrsimulator (German)*. Master thesis. Technische Universität München. 2018.

[21] Breyer, M. *Offline Motion Cueing Algorithmus für einen 9-DoF Fahrsimulator (German)*. Master thesis. Technische Universität München. 2017.

[64] Feng, Q. *Implementation of an Optimal Adaptive Motion-Cueing-Algorithm*. Semester thesis. Technische Universität München. 2017.

[88] Hristakiev, D. *MPC-based Motion Cueing Algorithm for Online Application*. Master thesis. Technische Universität München. 2018.

[104] Lankes, C. *Implementierung eines offline Motion Cueing Algorithmus basierend auf einer Modellprädiktiven Regelung (German)*. Master thesis. Technische Universität München. 2017.

[110] Mayr, L. *Filter- and Optimization-Based Motion Cueing Algorithm for a Driving Simulator*. Master thesis. Technische Universität München. 2018.

[120] Müller, T.-C. *Entwicklung einer entkoppelten Einzelgelenkregelung für einen Hexapoden (German)*. Semester thesis. Technische Universität München. 2017.

[128] Oberleitner, F. *Optimierungs-gestützter Motion-Cueing-Algorithmus: Implementierung eines Gelenkraum-basierten Prädiktionsmodells (German)*. Semester thesis. Technische Universität München. 2017.

[155] Seguin, L. *Developemnt of a Software-in-the-Loop Environment to Evaluate Motion-Cueing-Algorithms*. Semester thesis. Technische Universität München. 2017.

[163] Spannagl, M. *Hybrider Motion-Cueing-Algorithmus für einen 9-DoF Fahrsimulator (German)*. Master thesis. Technische Universität München. 2018.

[172] Treis, M. *Dynamikmodellierung eines Hexapods (German)*. Semester thesis. Technische Universität München. 2018.