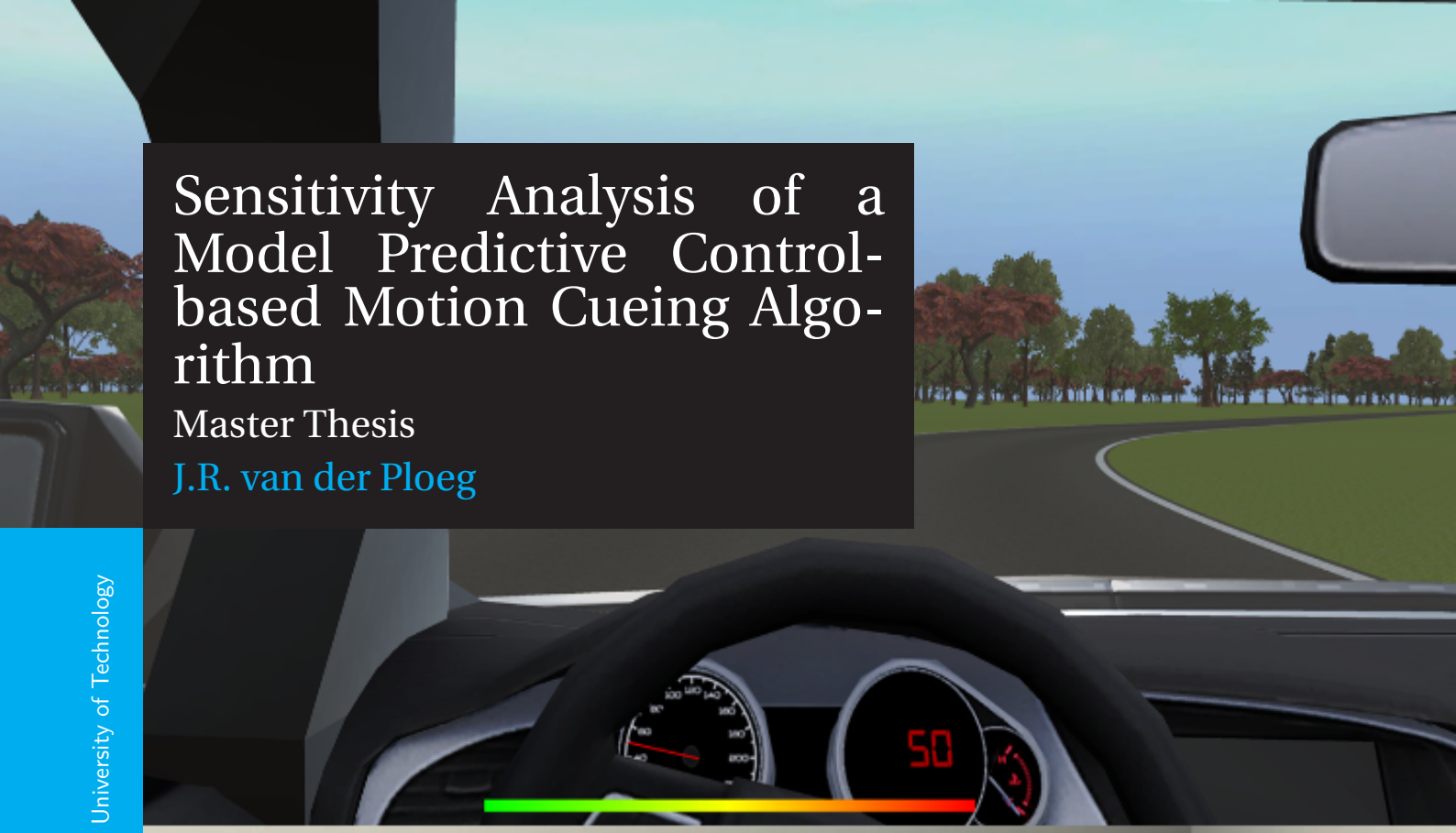
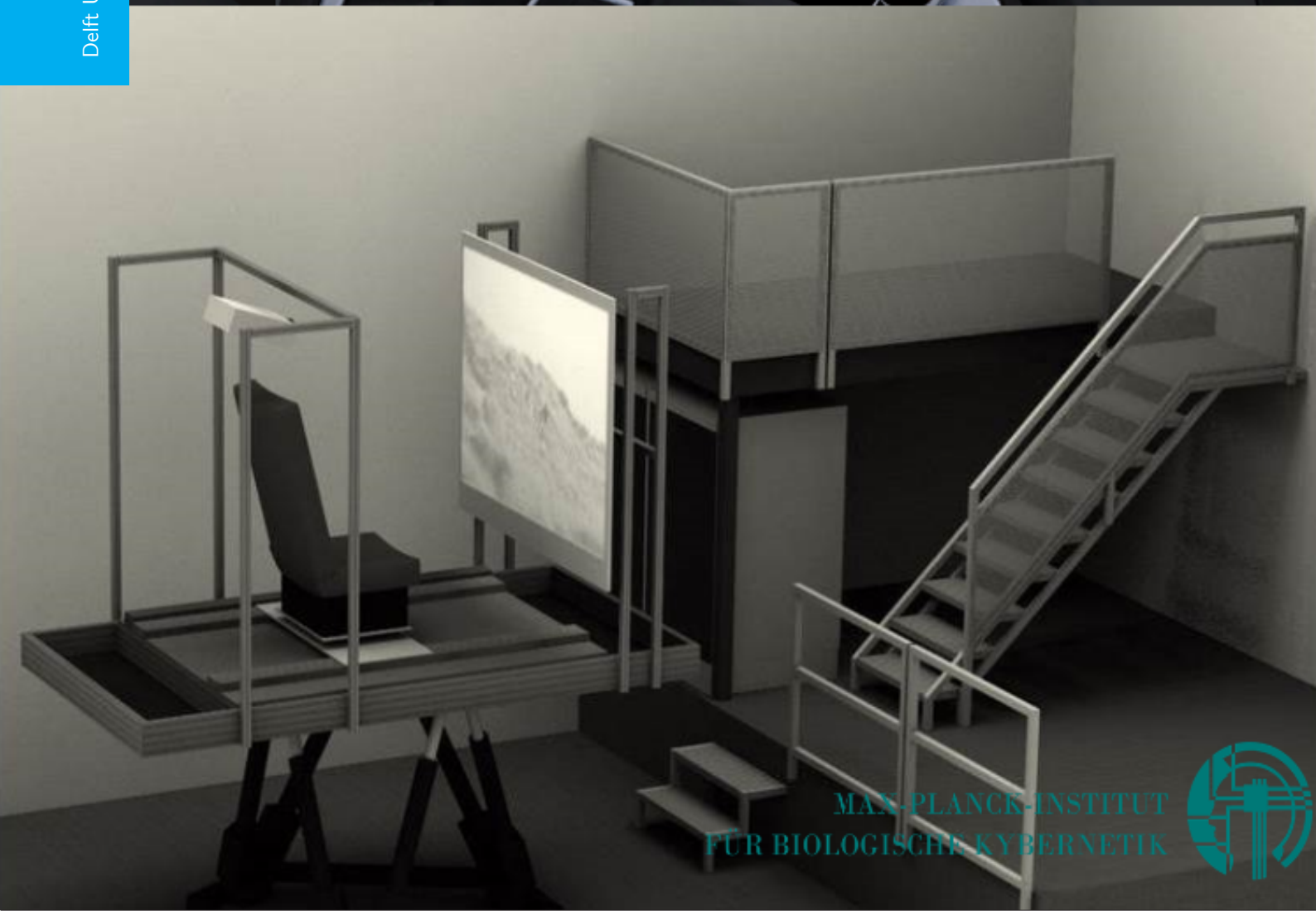
Sensitivity Analysis of a Model Predictive Control-based Motion Cueing Algorithm

Master Thesis

J.R. van der Ploeg

Delft University of Technology





TU Delft Delft University of Technology

Challenge the future

Sensitivity Analysis of a Model Predictive Control-based Motion Cueing Algorithm

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering at Delft University of Technology

J.R. van der Ploeg

December 20, 2018

TU Delft Delft University of Technology

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
CONTROL AND SIMULATION

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled "**Sensitivity Analysis of a Model Predictive Control-based Motion Cueing Algorithm**" by **Joost van der Ploeg** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 20th December 2018

Readers:

prof.dr.ir. M. Mulder

ir. D. Cleij

dr.ir. D. M. Pool

prof. dr. ir. D. A. Abbink

ACKNOWLEDGEMENTS

Although this master's Thesis was an individual research project, I could not have done it all by myself. Many people helped me in getting the most out of this project, and I would like to thank all of them. First, my two daily supervisors Daan and Diane stood by me throughout the entire project. They were always there to discuss my work with a critical and helpful attitude, which definitely helped in making my Thesis of a higher quality. I would also like to thank Max for his interest in my work and for making time to give guidance every once in a while.

I am also very thankful for the opportunity to do this research at the Max Planck Institute for Biological Cybernetics. It was a great experience and I have learned a lot from all of my colleagues there. Thank you to all colleagues from the Motion Perception and Simulation group. Special thanks go to Frank, my internship supervisor, who learned me a lot about the MPC-algorithm and helped me out more than once, even when he was not my supervisor anymore. Thank you to prof. Bülthoff as well, for giving me this opportunity.

Also my friends deserve a thank you. Marc, Luc, Tom, Tijmen and Aalok, thank you for all the fun we've had over the years. Our fierce, but at the same time friendly discussions have given me great joy. On top of that, we made the world a better place (in theory).

Finally, thank you to my family for always supporting me. You have always said that I should be doing what I like and have stood by me at all times, for which I am very grateful. And thank you Eva, for always believing in me and being there for me. Thank you for never leaving my side.

Joost van der Ploeg, December 2018

Contents

List of Figures ix

List of Tables xiii

I IEEE Paper
To be graded for final Thesis 1

II Preliminary Thesis Report
This part has already been graded for AE4020 Literature Study 21

1 Introduction 23
1.1 Problem Statement . . . 23
1.2 Project Goal . . . 24
1.3 Research Scope . . . 24
1.4 Report Structure . . . 25

2 Simulator Motion Cueing 27
2.1 Introduction to Simulator Motion Cueing . . . 27
2.2 Motion Simulators . . . 28
2.3 Types of Motion Cueing Algorithms . . . 29
2.3.1 Classical washout filter . . . 29
2.3.2 Adaptive washout filter . . . 30
2.3.3 Optimal control-based . . . 30
2.3.4 Model predictive control . . . 30

3 Model Predictive Control-Based Motion Cueing Algorithm 33
3.1 Principle of Model Predictive Control in Motion Cueing . . . 33
3.2 Model Predictive Control-Based Motion Cueing Algorithm of the MPI . . . 34
3.2.1 Parameters of the Cost Function of the Algorithm . . . 35
3.2.2 Prediction Strategies . . . 37

4 Metrics 39
4.1 Objective Metrics . . . 39
4.1.1 Root Mean Square Error . . . 39
4.1.2 Pearson Correlation Coefficient . . . 40
4.2 Perceived Motion Quality . . . 40

5 Sensitivity Analysis of the Output Error Weight Parameters 43
5.1 Control, Dependent and Independent variables . . . 43
5.2 Output Error Weight Parameter Sensitivity . . . 43
5.2.1 Comparative Classical Washout Filter . . . 44
5.2.2 MPC Sensitivity Plots . . . 45
5.2.3 Two-Way Sensitivity Plots . . . 52
5.3 Sensible Parameter Value Ranges . . . 58

6 Experiment Proposal 59
6.1 Method . . . 60
6.2 Apparatus . . . 60
6.3 Goal and Hypotheses . . . 60
6.4 Final Words . . . 61

Bibliography 63

III Paper Appendices
To be graded for final Thesis 67

A Sensitivity Plots 69
A.1 Sensitivity Plots . . . 69
A.2 Two-way Sensitivity Plots . . . 74

B Motion Plots for All Conditions Including IMU Data 91

C Experiment Forms 97

D Participant Rating Plots 103

List of Figures

2.1 The hexapod simulator of the Max Planck Institute for Biological Cybernetics. . . . 28
2.2 Simona Research Simulator of the TU Delft . . . 28
2.3 A schematic representation of the classical washout filter . . . 29
2.4 Block diagram of an Optimal control-based MCA. . . . 30

3.1 A graphical overview of the principle of MPC in motion cueing . . . 34

4.1 Simulator motion of the curve manoeuvre for a classical washout filter. . . . 40
4.2 The view as is seen through the eyes of participants, with the turning knob located at the bottom right. . . . 41
4.3 The turning knob with which participants can indicate the perceived motion mismatch. The rating bar on the background shows the current rating. . . . 41

5.1 A comparison of multiple classical washout filters. . . . 44
5.2 The RMSE between desired and actual ax, plotted against the parameter values of the output error weight parameters. . . . 45
5.3 The PCC between desired and actual ax, plotted against the parameter values of the output error weight parameters. . . . 46
5.4 The motion data when setting W_r to 85. . . . 46
5.5 The RMSE between desired and actual ay, plotted against the parameter values of the output error weight parameters. . . . 47
5.6 The PCC between desired and actual ay, plotted against the parameter values of the output error weight parameters. . . . 47
5.7 The RMSE between desired and actual az, plotted against the parameter values of the output error weight parameters. . . . 48
5.8 The PCC between desired and actual az, plotted against the parameter values of the output error weight parameters. . . . 48
5.9 The RMSE between desired and actual p, plotted against the parameter values of the output error weight parameters. . . . 49
5.10 The PCC between desired and actual p, plotted against the parameter values of the output error weight parameters. . . . 49
5.11 The RMSE between desired and actual q, plotted against the parameter values of the output error weight parameters. . . . 50
5.12 The PCC between desired and actual q, plotted against the parameter values of the output error weight parameters. . . . 50
5.13 The RMSE between desired and actual r, plotted against the parameter values of the output error weight parameters. . . . 51
5.14 The PCC between desired and actual r, plotted against the parameter values of the output error weight parameters. . . . 51
5.15 RMSE when changing Wax and Wq simultaneously. . . . 53
5.16 RMSE when changing Wax and Way simultaneously. . . . 53
5.17 RMSE when changing Wax and Wr simultaneously. . . . 54
5.18 RMSE when changing W_{ay} and W_p simultaneously. . . . 55
5.19 The output motion of the simulator with high W_{ay} and W_p values. . . . 56
5.20 The output motion of the simulator with medium W_{ay} and W_p values. . . . 57
5.21 RMSE when changing W_{ay} and W_az simultaneously. . . . 57

6.1 The visuals that subjects will see during the experiment. . . . 60

6.2 The turning knob with which participants can indicate the perceived motion mismatch. The rating bar on the background shows the current rating. . . . 60
6.3 The output of the simulator for all nine experiment conditions. . . . 62

A.1 The RMSE between desired and actual ax, plotted against the parameter values of the output error weight parameters. . . . 69
A.2 The PCC between desired and actual ax, plotted against the parameter values of the output error weight parameters. . . . 70
A.3 The RMSE between desired and actual ay, plotted against the parameter values of the output error weight parameters. . . . 70
A.4 The PCC between desired and actual ay, plotted against the parameter values of the output error weight parameters. . . . 70
A.5 The RMSE between desired and actual az, plotted against the parameter values of the output error weight parameters. . . . 71
A.6 The PCC between desired and actual az, plotted against the parameter values of the output error weight parameters. . . . 71
A.7 The RMSE between desired and actual p, plotted against the parameter values of the output error weight parameters. . . . 71
A.8 The PCC between desired and actual p, plotted against the parameter values of the output error weight parameters. . . . 72
A.9 The RMSE between desired and actual q, plotted against the parameter values of the output error weight parameters. . . . 72
A.10 The PCC between desired and actual q, plotted against the parameter values of the output error weight parameters. . . . 72
A.11 The RMSE between desired and actual r, plotted against the parameter values of the output error weight parameters. . . . 73
A.12 The PCC between desired and actual r, plotted against the parameter values of the output error weight parameters. . . . 73
A.13 RMSE when changing Wax and Way simultaneously. . . . 74
A.14 RMSE when changing Wax and Waz simultaneously. . . . 75
A.15 RMSE when changing Wax and Wp simultaneously. . . . 75
A.16 RMSE when changing Wax and Wq simultaneously. . . . 76
A.17 RMSE when changing Wax and Wr simultaneously. . . . 76
A.18 RMSE when changing Way and Waz simultaneously. . . . 77
A.19 RMSE when changing Way and Wp simultaneously. . . . 77
A.20 RMSE when changing Way and Wq simultaneously. . . . 78
A.21 RMSE when changing Way and Wr simultaneously. . . . 78
A.22 RMSE when changing Waz and Wp simultaneously. . . . 79
A.23 RMSE when changing Waz and Wq simultaneously. . . . 79
A.24 RMSE when changing Waz and Wr simultaneously. . . . 80
A.25 RMSE when changing Wp and Wq simultaneously. . . . 80
A.26 RMSE when changing Wp and Wr simultaneously. . . . 81
A.27 RMSE when changing Wq and Wr simultaneously. . . . 81
A.28 PCC when changing Wax and Way simultaneously. . . . 82
A.29 PCC when changing Wax and Waz simultaneously. . . . 82
A.30 PCC when changing Wax and Wp simultaneously. . . . 83
A.31 PCC when changing Wax and Wq simultaneously. . . . 83
A.32 PCC when changing Wax and Wr simultaneously. . . . 84
A.33 PCC when changing Way and Waz simultaneously. . . . 84
A.34 PCC when changing Way and Wp simultaneously. . . . 85
A.35 PCC when changing Way and Wq simultaneously. . . . 85
A.36 PCC when changing Way and Wr simultaneously. . . . 86
A.37 PCC when changing Waz and Wp simultaneously. . . . 86
A.38 PCC when changing Waz and Wq simultaneously. . . . 87
A.39 PCC when changing Waz and Wr simultaneously. . . . 87
A.40 PCC when changing Wp and Wq simultaneously. . . . 88

A.41 PCC when changing Wp and Wr simultaneously. 88
A.42 PCC when changing Wq and Wr simultaneously. 89

B.1 Simulator output for condition C1, including the desired output and the measured IMU data. . 92
B.2 Simulator output for condition C2, including the desired output and the measured IMU data. . 92
B.3 Simulator output for condition C3, including the desired output and the measured IMU data. . 93
B.4 Simulator output for condition C4, including the desired output and the measured IMU data. . 93
B.5 Simulator output for condition C5, including the desired output and the measured IMU data. . 94
B.6 Simulator output for condition C6, including the desired output and the measured IMU data. . 94
B.7 Simulator output for condition C7, including the desired output and the measured IMU data. . 95
B.8 Simulator output for condition C8, including the desired output and the measured IMU data. . 95
B.9 Simulator output for condition C9, including the desired output and the measured IMU data. . 96

D.1 Rating over time per trial for participant 1 104
D.2 Mean rating over time for participant 1 104
D.3 Rating over time per trial for participant 2 105
D.4 Mean rating over time for participant 2 105
D.5 Rating over time per trial for participant 3 106
D.6 Mean rating over time for participant 3 106
D.7 Rating over time per trial for participant 4 107
D.8 Mean rating over time for participant 4 107
D.9 Rating over time per trial for participant 5 108
D.10 Mean rating over time for participant 5 108
D.11 Rating over time per trial for participant 6 109
D.12 Mean rating over time for participant 6 109
D.13 Rating over time per trial for participant 7 110
D.14 Mean rating over time for participant 7 110
D.15 Rating over time per trial for participant 8 111
D.16 Mean rating over time for participant 8 111
D.17 Rating over time per trial for participant 9 112
D.18 Mean rating over time for participant 9 112
D.19 Rating over time per trial for participant 10 113
D.20 Mean rating over time for participant 10 113
D.21 Rating over time per trial for participant 11 114
D.22 Mean rating over time for participant 11 114
D.23 Rating over time per trial for participant 12 115
D.24 Mean rating over time for participant 12 115
D.25 Rating over time per trial for participant 13 116
D.26 Mean rating over time for participant 13 116
D.27 Rating over time per trial for participant 14 117
D.28 Mean rating over time for participant 14 117
D.29 Rating over time per trial for participant 15 118
D.30 Mean rating over time for participant 15 118
D.31 Rating over time per trial for participant 16 119
D.32 Mean rating over time for participant 16 119
D.33 Rating over time per trial for participant 17 120
D.34 Mean rating over time for participant 17 120
D.35 Rating over time per trial for participant 18 121
D.36 Mean rating over time for participant 18 121
D.37 Rating over time per trial for participant 19 122
D.38 Mean rating over time for participant 19 122
D.39 Rating over time per trial for participant 20 123
D.40 Mean rating over time for participant 20 123

List of Tables

3.1 Summary of the parameters of the cost function of the MPC-controller and the variable constraints. . . . 36
3.2 Summary of the parameter values of the cost function of the MPC-controller as will be used initially in this research. . . . 37

4.1 RMSE vallues for each motion channel when using a classical washout filter. . . . 40
4.2 PCC vallues for each motion channel when using a classical washout filter. . . . 40

5.1 Summary of the parameter values of the cost function of the MPC-controller as will be used as standard values. . . . 44
5.2 The relative changes in RMSE and PCC for motion channel ax when changing the output error weight parameters. . . . 46
5.3 The relative changes in RMSE and PCC for motion channel ay when changing the output error weight parameters. . . . 47
5.4 The relative changes in RMSE and PCC for motion channel az when changing the output error weight parameters. . . . 48
5.5 The relative changes in RMSE and PCC for motion channel p when changing the output error weight parameters. . . . 49
5.6 The relative changes in RMSE and PCC for motion channel q when changing the output error weight parameters. . . . 50
5.7 The relative changes in RMSE and PCC for motion channel r when changing the output error weight parameters. . . . 51
5.8 Relative changes in RMSE per motion channel for all parameter combinations involving W_{ax}. . . . 52
5.9 Relative changes in RMSE per motion channel for all parameter combinations involving W_{ay}. . . . 54
5.10 Relative changes in RMSE per motion channel for all parameter combinations involving W_{az}. . . . 55
5.11 Relative changes in RMSE per motion channel for all parameter combinations involving W_{p}. . . . 55
5.12 Relative changes in RMSE per motion channel for all parameter combinations involving W_{q}. . . . 56
5.13 Relative changes in RMSE per motion channel for all parameter combinations involving W_{r}. . . . 56

6.1 The parameter values of all nine experiment conditions. . . . 59
6.2 Simulator actuator velocity and acceleration limits. . . . 61
6.3 The RMSE and PCC values for all experiment conditions per motion channel. . . . 61

I

IEEE Paper

To be graded for final Thesis

Sensitivity Analysis and Experimental Testing of a Model Predictive Control-based Motion Cueing Algorithm During Curve Driving Simulation

J. R. van der Ploeg, Supervisors: D. Cleij, D. M. Pool, M. Mulder, H. H. Bülthoff

***Abstract*—Motion simulators are important for research and training purposes. In an attempt to increase the motion cueing quality of motion simulators, Motion Cueing Algorithms (MCAs) based on Model Predictive Control (MPC) are being developed. Research has shown that MPC-based MCAs have the potential to improve the motion cueing quality with respect to classical washout filter-based approaches by utilising the available physical space more effectively. This paper investigates the sensitivity of an MPC-based MCA when changing the internal parameters of the cost function of the algorithm. A preliminary analysis explored all possible combinations of parameters using the Root Mean Square Error (RMSE) and Pearson Correlation Coefficient (PCC). Based on those results, the effect of a select set of parameter combinations on the perceived motion quality was tested in a passive driving simulation experiment. It was shown that the error weight parameter on the lateral acceleration (W_{ay}) has more influence on the perceived motion quality than the error weight parameter on the roll rate (W_p). It was also found that the established baseline condition was rated as the best perceived motion quality. Additionally, combination of the RMSE of the translational accelerations plus a weighted RMSE of the rotational rates above the perception threshold showed a high correlation with the mean continuous rating, therefore having the potential to become a method for predicting how humans perceive motion quality.**

I. Introduction

Increasing use is made of full-motion vehicle simulators. They are being used to investigate human perception, cognition and action, for example by research organisations and by R&D departments of companies. Next to research, they are also used to test new innovations in a safe and controllable environment. Several studies have found that the addition of motion in simulations increases the perceived realism, while also increasing the performance of drivers [1] [2]. However, bad or false motion cues may decrease these benefits, meaning that no motion may be favoured over bad or false motion [3]. The challenge in providing motion cues is to simulate the vehicle motions as accurately as possible, while keeping the motion platform within its boundaries. This is generally handled by a so-called Motion Cueing Algorithm (MCA).

Currently, most of the MCAs are based on washout filters [4], [5], but this type of algorithm is not able to handle physical limits of the motion platform explicitly. In order to fit the motion envelope within the physical limits of the motion platform, the MCA is usually tuned such that the worst-case scenario does not exceed the limits [6]. The consequence is that for all other scenarios, the MCA performs sub-optimal. Searching for a solution to this problem resulted in the development of an adaptive washout filter, where the washout filter parameters are adjusted in real-time using an overlaying cost function [7] [8] [9]. Still, the physical limits are not explicitly accounted for. Therefore, an MCA based on Model Predictive Control (MPC) is gaining popularity [10] [11] [12], because MPC is able to explicitly account for physical limits and therefore uses the physical motion space more effectively [10].

Multiple comparisons between the classical washout filter-based MCAs and newly developed MPC-based MCAs have shown that MPC has the potential to provide an improvement in motion cueing quality with respect to filter-based approaches [13] [12], but this comparison only included one combination of parameter settings. MPC contains many parameters that have an influence on the motion cueing quality. [14] has investigated the influence of the prediction horizon length on motion cueing quality by determining the increase in computational costs and motion cueing quality for larger prediction horizons. [15] proposed a strategy to predict future motion during online driving simulation, where it is assumed that driving behaviour is somewhat constant, as is the case when looking at racing drivers. [16] presented an implementation of MPC that incorporates all actuators and still runs in real-time, despite the computational costs. Both [16] and [17] implemented an MPC-based MCA on non-conventional motion simulators. All these efforts help in creating a better MCA by improving one of the aspects of the MPC-algorithm.

Although offline simulations with an unlimited prediction horizon and perfect prediction have shown promising results for MPC-based MCAs [13], the main challenge lies with online simulations. They have a worse prediction quality, and on top of that they require the prediction horizon to be finite due to the computational costs of the algorithm and the unreliability of longer predictions. This increases the need for tuning the parameters of the MPC-based MCA (note that these are different from parameters in the Classical Washout Filter). Currently, little is known about how all these parameters influence the behaviour of an MPC-based MCA. Initially, it was thought the parameters of an MPC-based MCA were more intuitive to tune, because they all have a clear function that could be directly related to the simulator output (in contrast

J. R. van der Ploeg, D. M. Pool and M. Mulder are with the Control & Simulation division of the Faculty of Aerospace Engineering, Delft University of Technology, 2629HS Delft, the Netherlands

D. Cleij and H. H. Bülthoff are with the Max Planck Institute for Biological Cybernetics, 72076 Tbingen, Germany

to parameters such as damping or cut-off frequency). For example, increasing the error weight parameter on longitudinal acceleration should increase the tracking performance of the longitudinal acceleration. However, the large number of parameters and their interaction make this a complex problem. This research will help in understanding the behaviour of an MPC-based MCA and in tuning the parameters of the algorithm.

The goal of this research is therefore to investigate the influence of the parameters of the cost function of the MPI MPC-based MCA on the perceived motion quality. First, the change in the Root Mean Square Error (RMSE) and Pearson Correlation Coefficient (PCC) due to changing the parameters has been measured. Initially, only one parameter was changed each time, after which two parameters were varied to analyse the interaction effect between all parameters. Based on these results, a passive driving simulation experiment has been performed on a hexapod-based motion simulator, where the continuous perceived motion mismatch rating [18] of 20 participants has been measured.

This paper is structured as follows: first, the MPC-based MCA that is considered in this research is explained in Section II. Section III then briefly summarises the preliminary analysis of the influence of the error weight parameters on the behaviour of the algorithm by measuring the RMSE and PCC. From this analysis followed an experiment with human participants to measure the perceived motion quality and verify the used approach. The setup of this experiment is explained in Section IV. The results are presented in Section V and discussed in Section VI. Finally, Section VII draws the main conclusions from this research.

II. Model Predictive Control Motion Cueing

Model Predictive Control (MPC) is a technique initially developed to meet specially tailored control needs of nonlinear systems such as nuclear power plants and oil platforms [19]. Those processes need to be optimised at each control step, as the system characteristics change over time. MPC fits those needs, because MPC is a technique that explicitly optimises the future behaviour of a system over a specific time interval, the prediction horizon, based on the system input, the current state of the system and the future state of the system (which is based on the current state and inputs). It does this by minimising the squared error between a reference signal of the future and the calculated future signals. In other words, it solves an optimal control problem of a future trajectory, and this optimal control problem is solved at each time step to determine the next control step.

Recently, it has been proposed to use MPC to control motion simulators [10], as the next step towards better motion cueing. MPC algorithms are able to use the available physical space of motion simulators more effectively, since an optimal trajectory is calculated based on the current state of the system. Filter-based MCAs are usually scaled down to be able to accommodate the worst-case scenario, causing the process to be sub-optimal for all other scenarios. On top of that, MPC-based MCAs handle limits explicitly, therefore producing an optimised result while respecting the constraints. This also means that the algorithm does not have to be tuned in order to stay within the physical limits of the system. Instead, the tuning process is more focused on improving the performance and stability of the system.

Several institutions have implemented MPC-based MCAs on motion simulators [20] [11] [12], but this research focuses on the algorithm developed at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany, where the MPC-algorithm is also implemented on novel motion platforms, such as the Cable-Robot Simulator [16] and the CyberMotion simulator [17], an eight degrees of freedom serial robot simulator. However, in order to make the results of this study applicable to a wider audience, this research will be performed on a commonly available hexapod motion simulator.

The MPI MPC-based MCA includes an explicit model of the actuators of the motion platform, therefore directly optimising the control input. Although this increases the computational cost (due to a larger optimal control problem), the algorithm runs in real-time by utilising several computational tricks and simplifications, as explained in [16]. One of them is that instead of converging to the optimal solution of the optimal control problem each time-step, a maximum number of iterations per time-step can be set, meaning that a sub-optimal solution is taken in order to be able to continue to the next time step at a sufficiently high control frequency. However, doing this consecutively causes the algorithm to converge to the optimal solution anyway over multiple time-steps [21].

One of the challenges of MPC algorithms is determining the reference signals of the future motion. One could think of that as the desired trajectory of the system, because the algorithm will try to make the system do exactly as that reference signal. For some systems, such as controlling temperature, this is rather straightforward, since the future is predictable. However, in case of motion cueing for human-in-the-loop simulations, making a prediction of the future is difficult and risky.

Based on the predicted future motion, MPC determines the optimal future trajectory of the simulator each time step, meaning that the entire prediction horizon is optimised. Figure 1 shows the principle of MPC. The algorithm tries to track a reference signal by calculating the required control inputs that cause the system to approach that reference signal. Note that in general, only the first time step $(k+1)$ is used to control the motion platform. For the next time step, the entire prediction horizon is optimised again, creating a new optimised command for the motion platform at every time step.

A. Parameters of the Cost Function of the Algorithm

The optimisation of the future trajectory of the simulator is done via a cost function that minimises the squared error between reference values and actual values of the output signals (y_k), state signals (x_k), input signals (u_k) and the terminal state (x_n), over a certain prediction horizon (N) [16]. The mathematical form is shown in (1) - (3).

$$u_k = \arg\min_{u_k} \frac{1}{N} \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_n(x_n) \tag{1}$$

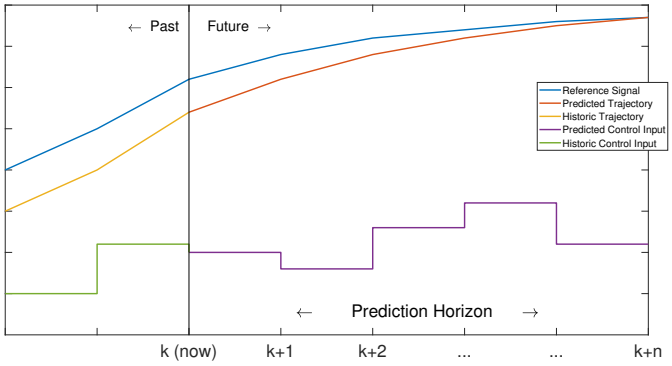


Fig. 1. Graphical representation of the principle of MPC, adapted from [11].

where:

$$l_k(x_k, u_k) = \left(y(x_k, u_k) - \hat{y}_k\right)^2 W_y + \left(x_k - \hat{x}_k\right)^2 W_x + \left(u_k - \hat{u}_k\right)^2 W_u \quad (2)$$

and:

$$l_n(x_n) = ||x_n - \hat{x}_n||^2 W_{x_n} \quad (3)$$

Many parameters have an influence on the behaviour of the MPC algorithm, as can be seen in (1) - (3), where W_y, W_u, W_x and W_{x_n} are all diagonal matrices with error weight parameters on the diagonal. This means that if the value of these parameters is increased, there will be a higher penalty on the squared error between the desired output and the output of the MPC-controller for that specific signal. Terminal state parameters (W_{x_n}) are basically the same as the normal state parameters, but terminal state parameters are the last state values in the prediction horizon. This can be used to ensure that the future prediction always ends up at a certain point, which can increase the stability of the algorithm.

There are nine output error weights (W_y, three translational accelerations, three angular velocities and three angular accelerations), twelve state error weights (W_x, three positions, three angles, three translational velocities and three angular velocities), twelve terminal state error weights (W_{x_n}, again three positions, three angles, three translational velocities and three angular velocities) and six input error weights (W_u, three translational accelerations and three angular accelerations).

As already explained before, MPC needs a reference trajectory against which the output error should be minimised, so for all these signals ($\hat{y}_k, \hat{x}_k, \hat{u}_k$ and $\hat{x}_n$), there needs to be a reference signal as well. Most of the reference signals are straightforward and logically chosen. First of all, the state reference values represent the state towards which the motion platform will perform washout, generally the motion platform's neutral position.

There is no need for input filtering, due to the fact that hexapod-based motion simulators only have limited physical possibilities. Should one perform a similar analysis on a different simulator with more physical possibilities, such as the CyberMotion Simulator of the MPI [22], this set of parameters could be of interest to reduce extreme behaviour such as constant spinning.

There is also no need for increasing stability by setting a value for the terminal state parameters (only in case of instability). Therefore, the input and terminal state reference signals are of little importance and thus set to zero. Finally, the output reference signals are the most interesting and difficult ones, due to their dependence on a prediction of what the output will be in the future, meaning that these will not be constant.

There are a lot of parameters that influence the behaviour of the MPC-controller. Some parameters have a direct influence on the simulator output and therefore the influence can easily be deduced, but other parameters may not impact the behaviour of the algorithm in a clear way, which makes it more difficult to explain the behaviour. Note that the following description is based on having the reference signals for the input, state and terminal state all zero, as that is the case that will be used in this research. First of all, for the prediction horizon, a longer prediction horizon generally increases the performance, but the risk of an inaccurate prediction increases with longer prediction horizons, as well as the computational cost [14].

Second, the error weights put a penalty on the error of that specific signal. Higher output error weights (W_y) should improve tracking of the predicted reference, higher state error weights (W_x) increase the washout, high input error weights (W_u) penalise high input values, and high terminal state error weights (W_{x_n}) are there to improve the stability of the algorithm.

As a standard practice at the MPI, the values for the output error weights are chosen to reflect the variance of specific force in m/s^2 and angular velocity in rad/s for typical vehicle manoeuvres [16]. The input error weights are set to a very small value, because in general there is no need for input filtering on a hexapod-based motion simulator. The terminal state error weights are set to a very small value as well, because this is only necessary to tune in case of unstable behaviour.

TABLE I
OVERVIEW OF THE PARAMETER VALUES OF THE COST FUNCTION OF THE MPC-CONTROLLER IN THE BASELINE CONFIGURATION.

N	20			
Δt	0.1			
	$ax\ ay\ az$	p q r	$\dot{p}\ \dot{q}\ \dot{r}$	
W_y	1 1 1	10 10 10	0 0 0	
W_u	0.01 0.01 0.01	- - -	0.01 0.01 0.01	
$\hat{y_k}$		Dependent on MPC prediction		
$\hat{u_k}$	0 0 0	- - -	0 0 0	
	$x\ y\ z$	$\phi\ \theta\ \psi$	$\dot{x}\ \dot{y}\ \dot{z}$	p q r
W_x	8.2 5.1 3.6	5.4 3.7 6.8	0 0 0	0 0 0
W_{x_n}	0.01 0.01 0.01	0.01 0.01 0.01	0.01 0.01 0.01	0.01 0.01 0.01
$\hat{x_k}$	0 0 0	0 0 0	0 0 0	0 0 0
$\hat{x_n}$	0 0 0	0 0 0	0 0 0	0 0 0

Then finally, for the state error weights (W_x) an optimisation has been performed. The manoeuvre that will be used to analyse the influence of some of the parameters is a car

driving through a curve, and the state error weights have been optimised such that with the baseline settings for W_y, W_u and W_{x_n}, the simulator platform moves back to its neutral position within a reasonable time span. This has been done by optimising a cost function that calculates the the total error of all output signals during the curve, while putting a very high penalty on all motion at some time after the curve, such that the motion platform is in its neutral position before a next curve starts. A summary of the parameters as they will be used initially is shown in Table I.

B. Prediction Strategies

A very important but tricky aspect of the MPC-controller is the future prediction of y. For passive simulations, this is straightforward, as the future trajectory is know. However, this is not the case when uncertainties in the control loop or online motion cueing are present, which is most often the case. There are multiple options if that is the situation.

The easiest and most straightforward to implement is to assume that there will be no change in the near future and keep the predicted output constant. The downside of a constant prediction method is that longer prediction horizons cause the algorithm to show behaviour opposite to what is desired. If for example the current output is an acceleration of 1 m/s^2 to the left and the prediction horizon is 10 seconds, the algorithm will prepare for such a prediction by moving to the right, such that a larger movement to the left can be made. But because the prediction is constant, the algorithm will keep doing this until the motion changes. The result is that an acceleration to the left is simulated by providing an acceleration to the right, which is undesirable.

Another, similar option is to use the current output values and scale them linearly, however it is difficult to actually implement that, as the value could go up or down, so keeping it constant is a safer method.

A completely different approach would be to have an actual prediction of what is going to happen, based on a virtual driver or a recording of a previous simulation run. Then comparing the current position of the simulated vehicle to the one from the recording enables the algorithm to predict what is going to happen. However, for this method to work, the behaviour of the recorded driving and the actual driver should be very similar, because else the prediction is false. This method is only suitable for experienced drivers [15], such as racing drivers. Therefore, in this research a constant prediction will be used.

The prediction horizon is defined by two parameters, the number of steps and the time in between steps. Small time steps are only useful when the behaviour of the system changes rapidly. Furthermore, a long prediction time is only beneficial if the prediction is accurate, and even then at a certain point the gain in performance is not worth the increased computational costs. For uncertain predictions, it can even make the performance worse. At the time of performing this analysis, the maximum number of prediction steps that could be handled in real-time was a around 100, therefore, to ensure that this version of an MPC-based MCA can run in real-time for most practical implementations, it has been decided to limit the number of steps to 20 for this research. In order to still have a meaningful prediction time, a time step of 0.1 seconds is taken, such that the prediction time is 2.0 seconds. For driving a vehicle, not a lot will change in 0.1 seconds, so this is a safe value to take.

III. Output Weight Sensitivity Analysis

An initial analysis on the sensitivity of the MPC-based MCA of the MPI due to varying the output error weight parameters (W_y) of the cost function has been performed. To investigate all possible parameter combination, only the objective metrics Root Mean Square Error (RMSE) and Pearson Correlation Coefficient (PCC) are evaluated, since these can be determined without the need for human-in-the-loop experiment. The RMSE is an indication of the performance of signal tracking (therefore penalising both magnitude and shape errors) and the PCC (which penalises shape errors) has been shown to correlate with the perceived motion quality [23].

Magnitude or scaling errors are not perceived as bad when staying within a certain range [24], but shape errors (which are a result of false cues) lead to large decreases in perceived motion quality, with [6] arguing that they are the most detrimental to perceived motion quality. See (4) and (5) for mathematical expressions of the RMSE and PCC, respectively. In these equations, N = number of data points, y_i = actual value, $\hat{y}_i)$ = reference value, μ = mean and σ = standard deviation.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{4}$$

$$PCC(y, \hat{y}) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{y_i - \mu_y}{\sigma_y}\right)\left(\frac{\hat{y}_i - \mu_{\hat{y}}}{\sigma_{\hat{y}}}\right) \tag{5}$$

The vehicle motion is obtained by letting $CarSim$ perform a curve driving manoeuvre. A baseline condition has been established such that the values for the output error weights reflect the variance of specific force in m/s^2 and angular velocity in rad/s^{-1} for typical vehicle manoeuvres [16]. The MPC parameter values are listed in Table I. First, all parameters have been changed one by one, therefore only looking at their individual sensitivity. After that, two parameters have been changed simultaneously to investigate their interaction effects.

In the first analysis, it became clear that all parameters mostly influence their corresponding motion channel, namely that increasing a certain error weight parameter decreased the RMSE and increased the PCC for that same motion channel. The RMSE and PCC of a_y due to varying all six parameters individually can be found in Figures 2 and 3. The parameter names have the concerned motion channel as subscript, i.e., W_{ax} is the error weight parameter on the longitudinal acceleration.

It is clear that W_{ay} has the most influence on a_y. Even a small increase in W_{ay} already causes the algorithm to produce a smaller RMSE in a_y than an optimised classical washout

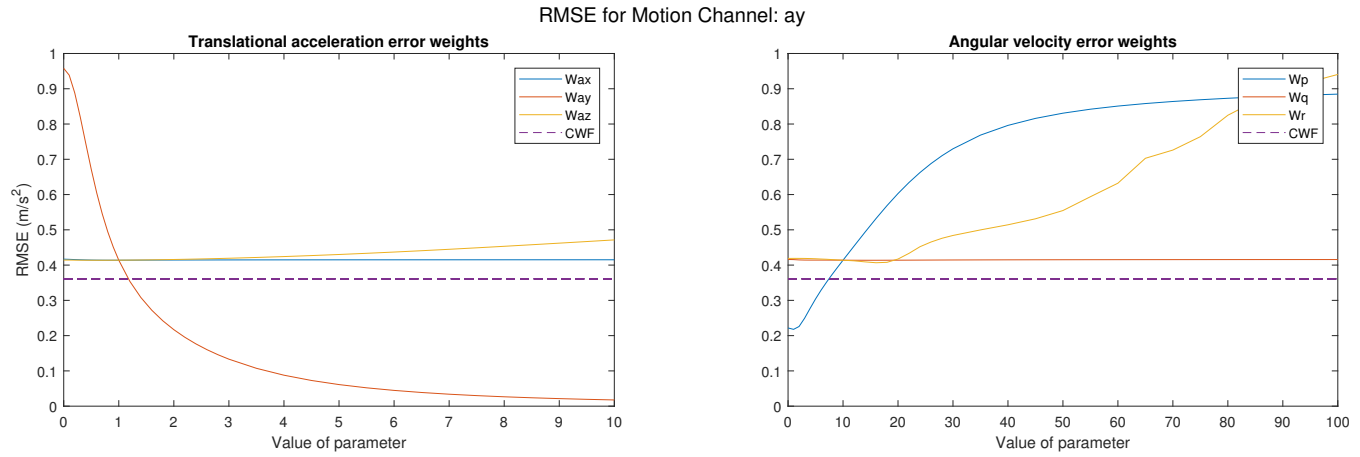


Fig. 2. RMSE of a_y when changing all parameters individually.

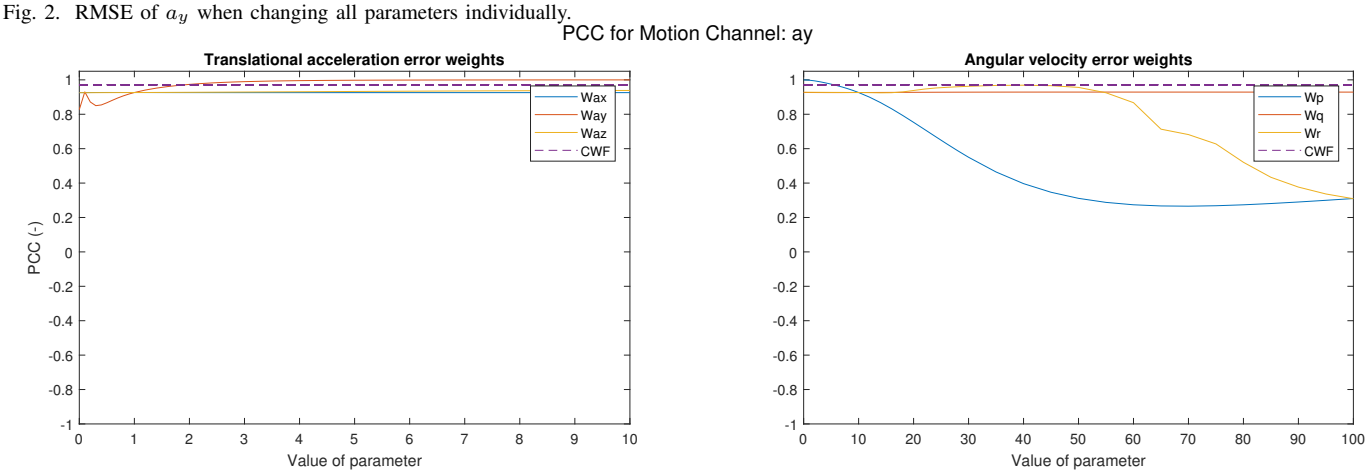


Fig. 3. PCC of a_y when changing all parameters individually.

filter. The RMSE in a_y can be decreased by as much as 95% when increasing W_{ay}. Next to that, it can be observed that W_p also affects a_y, which is due to tilt-coordination. In this example, W_p negatively affects the cueing accuracy for a_y, since the RMSE increases and the PCC decreases due to an increase of W_p. The RMSE in a_y is increased by as much as 114% when increasing W_p. Both of these effects are expected, but on top of that, W_r seemed to affect most other motion channels. This is caused by the fact that yaw rate is difficult to simulate on hexapod based motion simulators as it is only possible to reproduce a yaw rate for a short amount of time before the physical limits are reached.

After the individual sensitivity had been established, all parameters were tested against each other to analyse the effects of changing two parameters simultaneously. To limit the scope of this research, the maximum number of parameters that will simultaneously be varied is two. The effect of changing two parameters simultaneously can also still be graphically shown (in a 3-dimensional plot), which is often important in gaining an understanding of what is happening. The most interesting parameter combination turned out to be W_{ay} and W_p, due to them both having an effect on a_y, the most dominant stimulus during curve driving.

Note that in the analysis, the translational error weight parameters range from 0-10 and the angular rates range from 0-100. This difference in scale has the same reason as the choice of values for the parameters in the baseline case, i.e., the difference in units. This means that to achieve a penalty of equal magnitude on the angular rates in rad/s, the weight needs to be around ten times as high as the weight on the translational accelerations in m/s^2. The maxima of 10 m/s^2 and 100 rad/s where chosen as the change in RMSE or PCC becomes rather small for these high weight values.

In some cases, undesired behaviour starts happening way before these values, making it less interesting to investigate higher values. Undesired behaviour does not mean introducing large errors, but introducing oscillating behaviour on certain motion channels as a result of increasing certain parameters too much, for example increasing W_{ay} and W_p simultaneously. If we look at Figure 4, we see that if both W_{ay} and W_p are increased, a high RMSE for a_x, q and r occurs, which is the result of oscillating behaviour on those motion channels. Figure 6 shows the simulator output when both W_{ay} and W_p are increased too far, where it can be seen that all other motion channels show oscillating behaviour. This happens because at this point, a small performance increase in a_y or p outweighs the large error on other motion channels.

Figures 4 and 5 show the RMSE and PCC, respectively, of

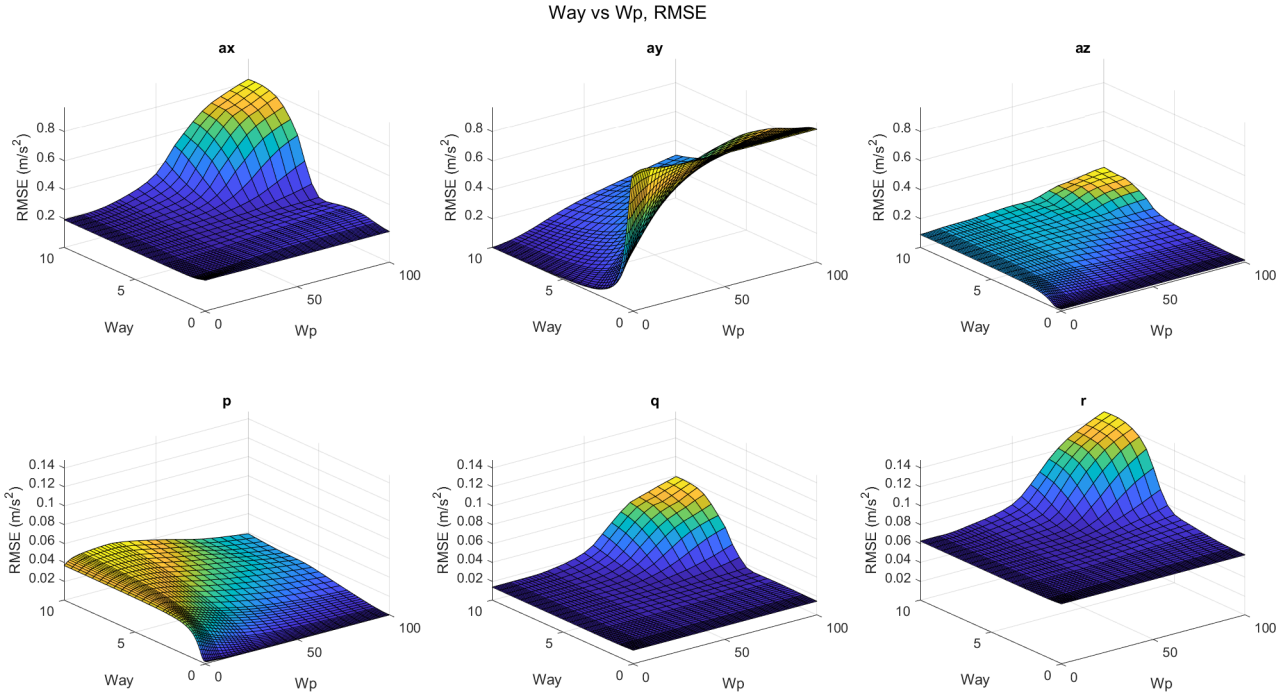


Fig. 4. Sensitivity of RMSE due to varying W_{ay} and W_p simultaneously.

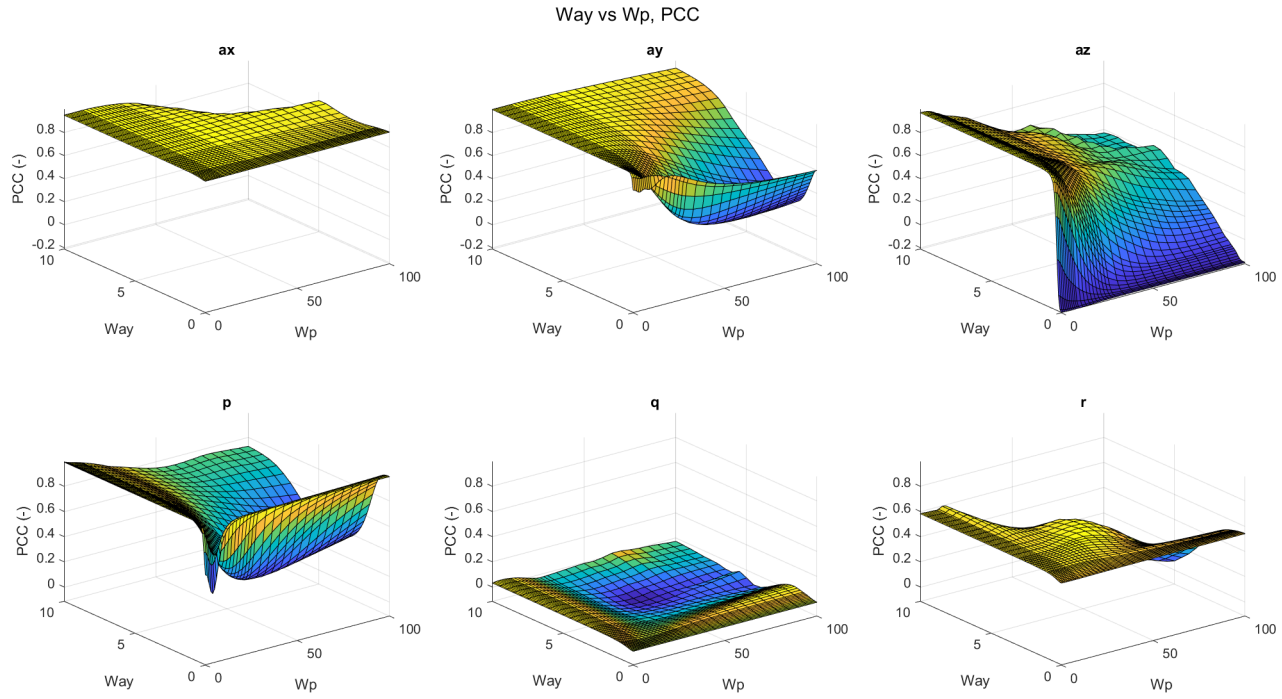


Fig. 5. Sensitivity of PCC due to varying W_{ay} and W_p simultaneously.

all motion channels due to varying W_{ay} and W_p. Generally speaking, we see the following behaviour when W_{ay} and W_p are varied. First, increasing W_{ay} results in a lower RMSE in a_y, but also in a higher RMSE in p, meaning that a better reproduction of the lateral acceleration is realised at the cost of a false cue in roll rate. Decreasing W_{ay} has a reversed effect,

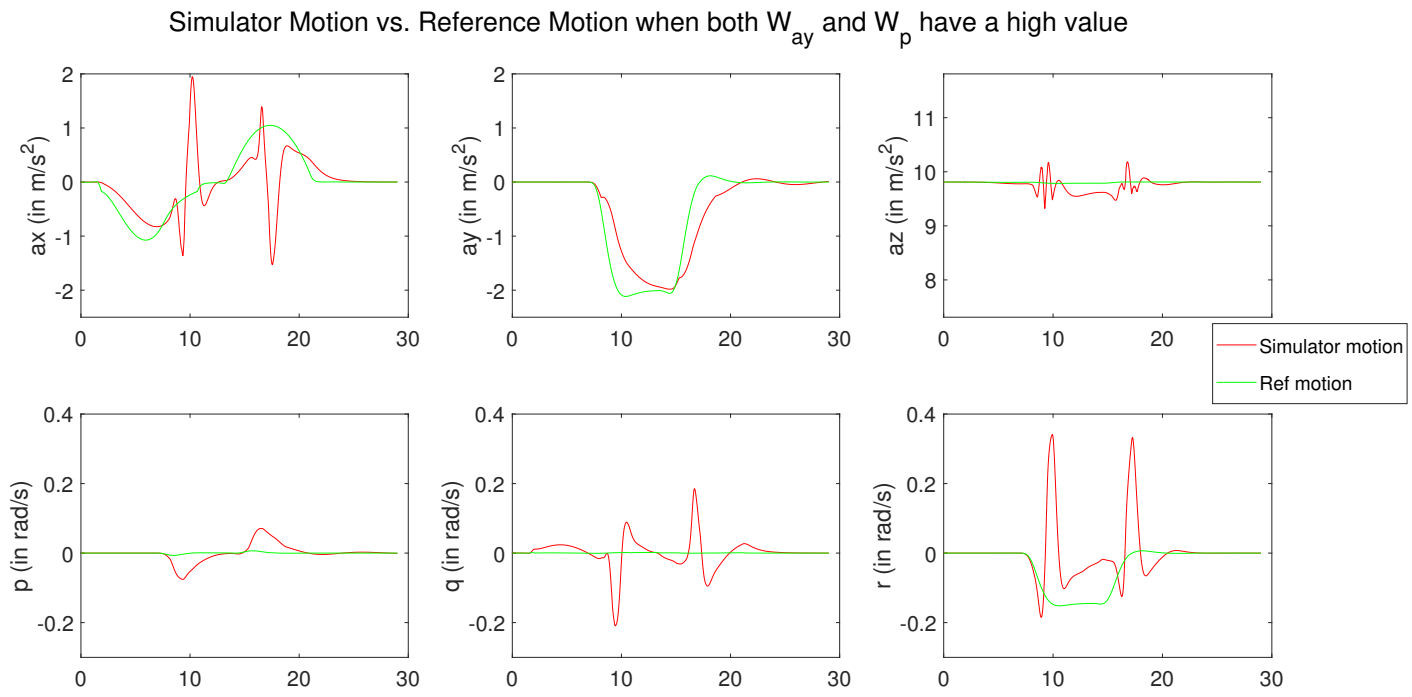


Fig. 6. Simulator output when setting both W_{ay} and W_p to a high value.

which causes a reduced tracking performance of a_y. Increasing W_p reduces the RMSE in p, meaning that the false cue that is generated due to tilt-coordination, also reduces. However, this comes at the cost of a reduced tracking performance of a_y. And decreasing W_p allows for more use of tilt-coordination, causing the RMSE in p to increase.

Finally, it can be seen that increasing either W_{ay} or W_p does not result in a drastic change in RMSE or PCC, but when W_{ay} and W_p are increased simultaneously, the RMSE on the remaining motion channels drastically increases. This is because if both W_{ay} and W_p are increased too much, a very small performance increase in a_y or p is realised at the cost of large errors on other motion channels. Therefore we can say that W_{ay} and W_p show interaction effects. The RMSE in a_y can be decreased by as much as 95% when increasing W_{ay} with respect to the baseline condition, but this reduces to 66% when W_p is increased as well. This is an expected result, since a_y and p are related via tilt-coordination.

We can conclude that W_{ay} mainly affects the sustained part of the curve, because that is where a_y is most dominantly present. W_p mainly affects the curve onset and exit, as there is only a roll rate during those segments. Increasing both parameters too much results in undesired behaviour, as was seen in Figure 6.

The impact of the six output error weight parameters on the performance of the MPC-based MCA has been studied by looking at the RMSE and PCC between the desired and actual simulator output signals with a range of different values of the parameters. But these metrics are merely an indication of the quality of a simulator, thus these results are not conclusive. To verify the predicted sensitivity of the MPC-based MCA output and thus motion cueing quality, a passive driving simulation experiment was performed. In this experiment, the perceived motion cueing quality is measured by using the continuous rating method, first used in [18].

IV. Method

A. Control, Dependent and Independent Variables

The state error weight, terminal state error weight and input error weight parameters are all control variables, they will remain constant throughout the entire analysis. This research will only focus on the effects of changing W_{ay} and W_p, since they showed the most interesting behaviour and are most important in curve driving simulation. The MPC parameter values, vehicle manoeuvre and prediction strategy are the same as used before. And finally, the dependent variable is the continuous mismatch rating (CR), because it has been shown that this is a good indication of the perceived motion quality [25] and with the CR it is possible to analyse the results per section of the curve, which is not possible with only a single metric. The MPI hexapod simulator (Bosch Rexroth eMotion-1500-6DOF-650-MK1) (see Figures 7 and 8) was used for the experiment, a commonly available simulator.

B. Experiment Procedure

All of the nine conditions were played back to back, i.e., a single simulation trial consisted of nine curves. An initial acceleration and final deceleration were included to make the trial more realistic. As part of the continuous rating method [25], two training trials were performed to let the participants get used to the method and to get a feeling for

Fig. 7. The front view of the MPI hexapod simulator.

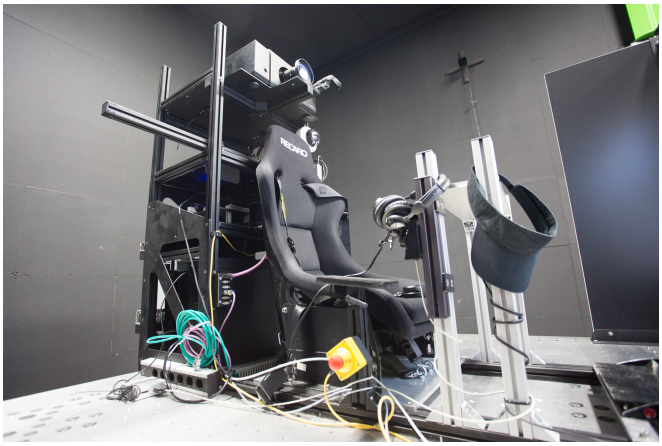


Fig. 8. The seat of the MPI hexapod simulator.

which motion feels good and which motion feels bad. After that, participants had to rate three repeated measurement trials, in order to get rid of outlier behaviour and to be able to quantify whether participants rated consistently. One trial took approximately 6 minutes.

Twenty participants participated in the experiment. During the experiment, participants saw computer-generated visuals (from Unity) projected on a screen in front of them that matched the vehicle motion that was generated in $CarSim$, see Figure 9. Based on those visuals, participants had a certain expectation of what they would feel if they would be situated in a real car. Because they are not in a real car but in a motion simulator, discrepancies can occur between their expectation and what they really perceive. Participants could report this by turning a knob that indicates to what extent they feel a mismatch between the visuals and the motion, also known as the continuous rating method [18] [25]. The rating knob is shown in Figure 10. Participant's ratings were shown on the screen and measured throughout the experiment. The continuous rating captures the perceived motion incongruence over time of the simulation. Finally, to help in the interpretation of the CR results, a questionnaire was filled in after the experiment in which participants were asked to indicate how they decided on a certain rating. On top of that, after each experiment trial a sickness score was asked to monitor motion sickness during the experiment.

Fig. 9. The experiment visuals projected on the simulator's screen.

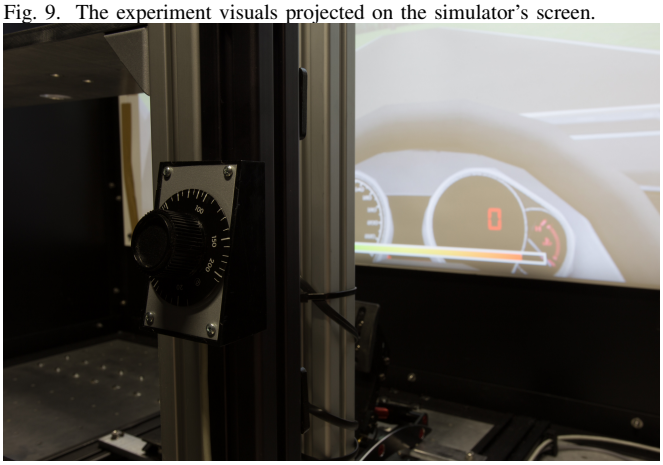


Fig. 10. The turning knob with which participants provided their continuous rating. The rating bar on the screen shows the current rating.

C. Experiment Conditions

The baseline condition (see Table I) served as the reference MPC condition. From there W_{ay} and W_p were increased and decreased the same relatively. Both of these parameters were varied independently, meaning that in total, the full factorial of nine experimental conditions was tested. The final result has been a trade-off between getting a maximum difference between the experiment conditions and keeping the simulation comfortable, since human subjects will take place inside a simulator. This means that there must be no negative PCC on any of the motion channels, since this indicates that there are false cues present that are in opposite direction of what is expected and this might cause motion sickness. Taking both

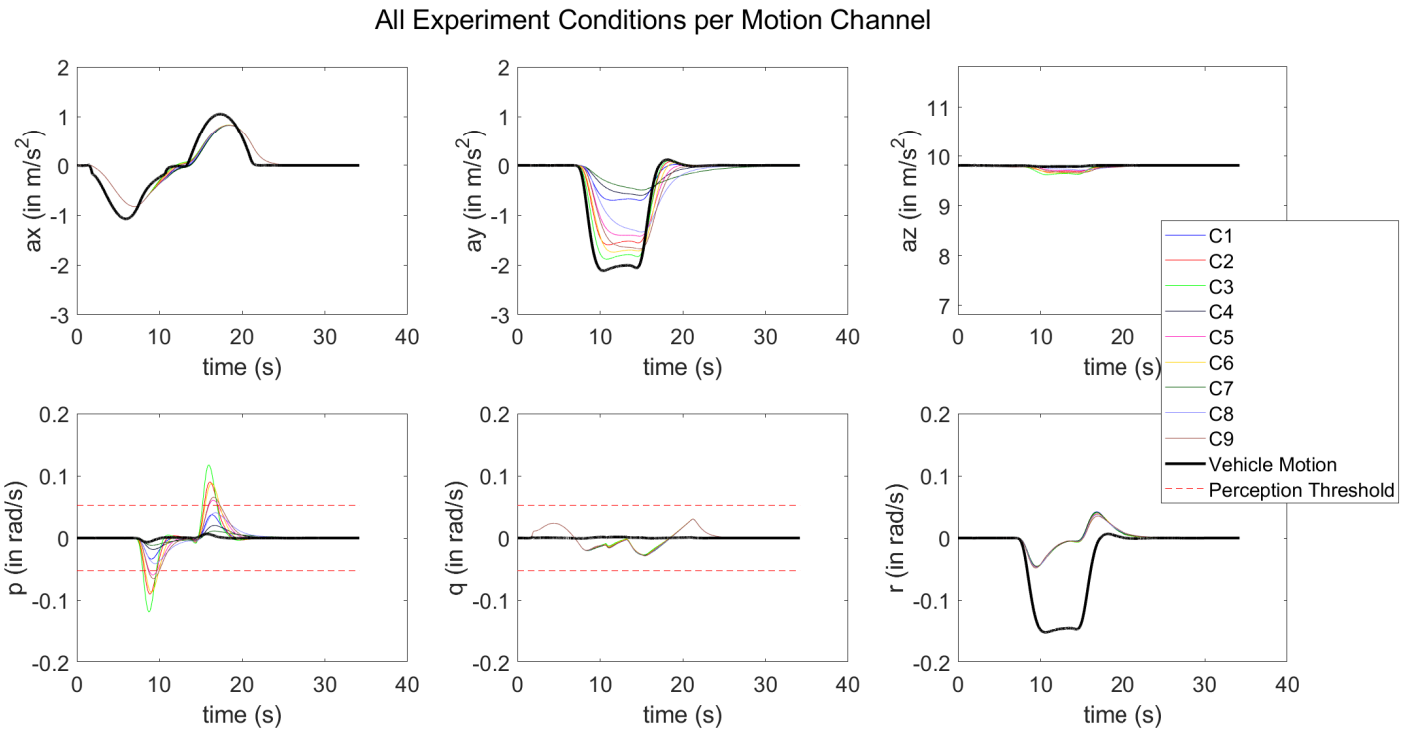


Fig. 11. The output of the simulator for all nine experiment conditions, including perception thresholds for tilt rates.

of these goals into account resulted in both parameter values being increased and decreased by 60%, meaning that the tested parameter values for the experiment were indicated in Table II. Figure 11 shows the traces of the simulator outputs for all nine experiment conditions, including the perception threshold for tilt rates of 3 deg/s [26], or 0.0524 rad/s. It can be seen that C2, C3 and C6 surpass the perception threshold clearly, while C5 and C9 just surpass the threshold. Figures 12 and 13 show the RMSE and PCC values for all experiment conditions. It can be observed that increasing W_{ay} lowers the RMSE of ay, but increases the RMSE of p, while W_p does the opposite.

TABLE II
THE PARAMETER VALUES, CORRESPONDING TOTAL RMSE AND AVERAGE PCC OF ALL NINE EXPERIMENT CONDITIONS.

W_p	W_{ay}	Condition	Total RMSE	Average PCC
	0.4	C1	1.767	0.622
4	1	C2	1.547	0.727
	1.6	C3	1.500	0.743
	0.4	C4	1.814	0.525
10	1	C5	1.621	0.666
	1.6	C6	1.563	0.705
	0.4	C7	1.856	0.459
16	1	C8	1.694	0.593
	1.6	C9	1.626	0.656

D. Hypotheses

The goal of this experiment was to measure the effect of the MPC-based MCA's output error weight parameters on the perceived motion cueing quality. Based on the results of the sensitivity analysis in Section III, the following hypotheses were formulated:

- **H1** Varying W_{ay} *will have more influence on the perceived motion quality than* W_p. Table II shows the summed RMSE (the angular rates received a weight of 10 to compensate for the difference in units) and the average PCC per condition. Using these values, we can calculate the change in RMSE and PCC due to a change in either W_{ay} or W_p with respect to the middle values. Adding these values gives us a total change in RMSE or PCC due to changing one of the parameters. Doing this for both parameters and both metrics yields a total change in RMSE of 46.3% due to W_{ay} and 22.1% due to W_p and a total change in PCC of 76.9% due to W_{ay} and 63.4% due to W_p. Since W_{ay} has more influence on both the RMSE and PCC, it is expected that W_{ay} also has more influence on the perceived motion quality.
- **H2** *C8 will be rated as the best condition.* [26] found that the perception threshold for tilt rates is 3 $^\circ/s$. This means that participants will notice tilt-coordination when the value of 3 $^\circ/s$ is exceeded, which should be given a high mismatch rating since this is a false cue. Only C1, C4, C7 and C8 stay below the perception threshold, and from these conditions, C8 has the lowest RMSE and the highest PCC on the relevant motion channels. Therefore it is expected that C8 will receive the lowest CR.

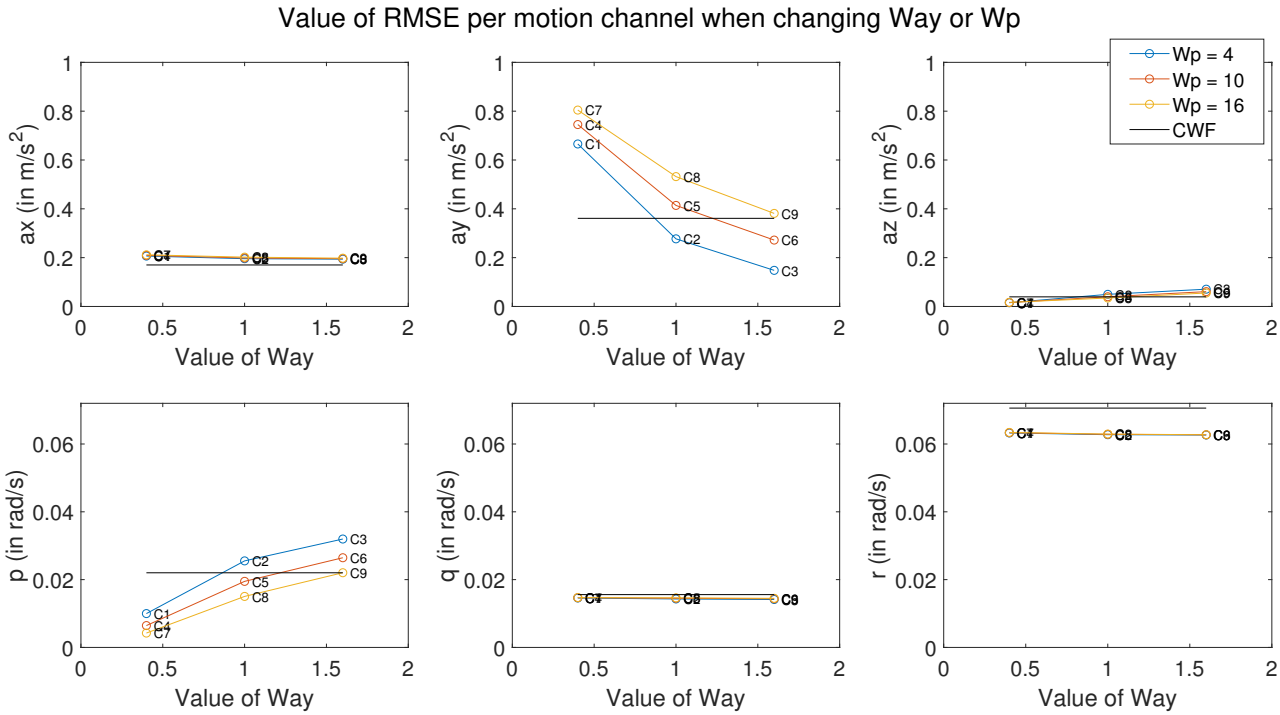


Fig. 12. RMSE values for all experiment conditions. The reference Classical Washout Filter (CWF) has been included for comparison.

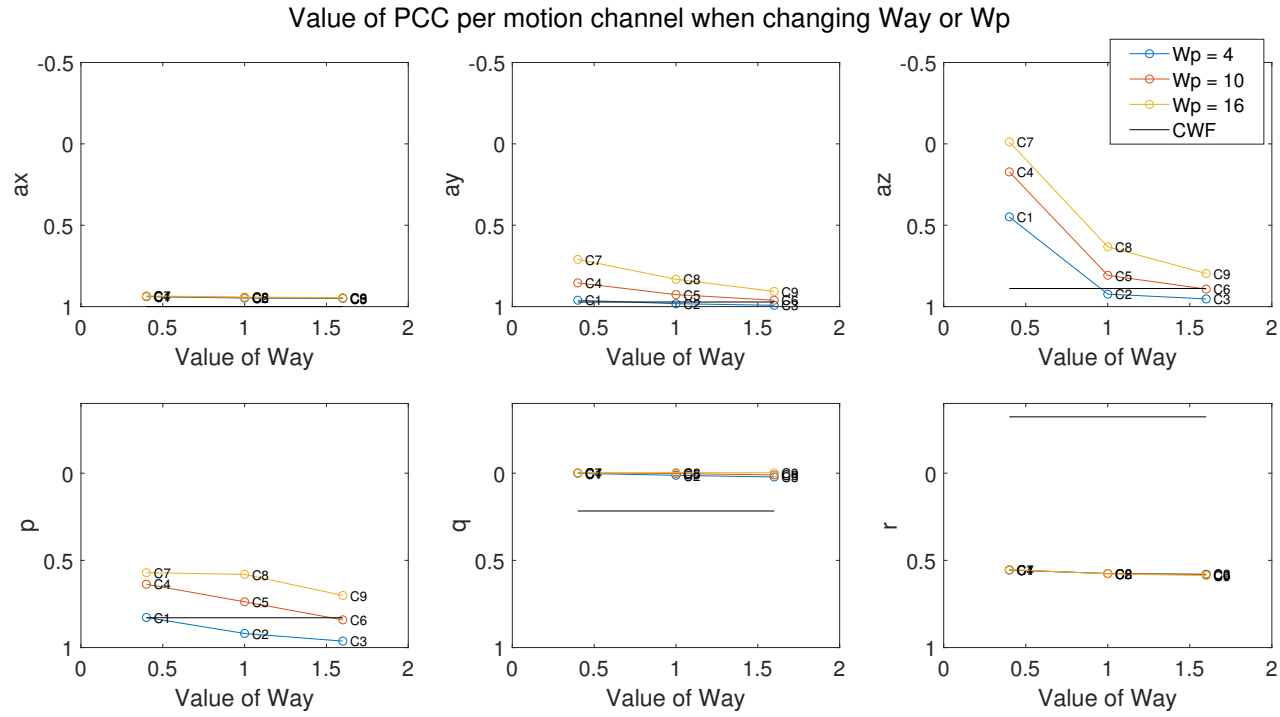


Fig. 13. PCC values for all experiment conditions. The reference CWF has been included for comparison.

E. Data Analysis

The only dependent variable is the CR, which is measured over time. Before analysing the data, it will be checked whether participants were consistent using Cronbach's Alpha, a measure of internal consistency. Additionally, the mean ratings for left and right curves will be compared for each condition individually to verify that there are no differences in rating a left or right curve. This is done because the visuals were not symmetric due to the fact that visuals where created from the perspective of the driver (left seat). After the usability of the data has been determined, the mean CR of all participants per condition will be used to compare the different conditions.

On top of that, because W_{ay} and W_p affect different parts of the curve, the mean CR of all participants per condition will also be determined for each curve segment separately, in order to analyse the results in different curve segments. First, the effect of varying the parameters W_{ay} and W_p will be investigated by looking at what happens during each curve segment due to varying the parameters. After that, the performance of the conditions will be analysed and it will be determined which condition was perceived as most realistic.

Finally, in order to determine whether the RMSE and PCC are indicative for the perceived motion quality, a correlation coefficient will be calculated between the RMSE or PCC of a combination of motion channels and the mean continuous rating. Note that adding the RMSE values includes a weight of 10 on the rotational rates due to the difference in units. Also note that the mean continuous rating indicates a mismatch, therefore a lower value indicates a better motion quality. This implies that for a positive correlation, the PCC should be checked for inverse correlation.

V. Results

Two out of the 20 participants have been excluded from the results due to a poor consistency in giving their rating (Cronbach's Alpha < 0.7). Additionally, it has been checked whether there is a difference in rating a left or a right turn. A Wilcoxon test for all conditions individually showed no significant difference between the given ratings for left or right curves ($p > 0.05$). These findings are explained in more detail in Appendix A.

A. Effect of Changing Parameters

Figure 14 shows the average mean CR over time for all experiment conditions. It seems that C5 received the lowest CR over time, but the highest CR over time is not entirely clear from this figure. C7 has the highest peak, while C3 is rated higher during the curve onset and exit. Table III shows the mean ratings per curve segment for all conditions, where the lowest means are shown in *italics* and the highest values are in **bold**. C3 received the highest mean rating over the entire curve, curve onset and curve exit, but during the sustained part C7 is rated worse. This is caused by the fact that C3 had the highest W_{ay} and the lowest W_p, resulting in the lowest RMSE in a_y but the highest false cue in p during the curve onset and exit. Conversely, C7 had the lowest W_{ay} and the highest W_p, resulting in the highest RMSE in a_y, which is felt most during steady curve driving. Figure 15 shows the spread of the mean ratings per curve segment in a boxplot, where it can again be seen that C3 received a high mismatch rating during the curve onset (red), and C7 received a high mismatch rating during the sustained part (green).

TABLE III
MEAN CONTINUOUS RATINGS OF THE ENTIRE CURVE PER CURVE SEGMENT PER CONDITION.

	Mean rating over			
Condition	Entire curve	Curve onset	Sustained part	Curve exit
C1	0.2528	0.1182	0.4724	0.1131
C2	0.2629	0.1653	0.4184	0.1663
C3	**0.3246**	**0.2403**	0.4911	**0.2010**
C4	0.3019	0.1665	0.5186	0.1669
C5	*0.1604*	*0.0879*	*0.2557*	0.1139
C6	0.2319	0.1464	0.3677	0.1478
C7	0.3177	0.1399	**0.5652**	0.1864
C8	0.1967	0.1038	0.3406	*0.1101*
C9	0.2009	0.1141	0.3203	0.1387

In Section IV, we determined the total change in RMSE and PCC due to changing the parameters. This procedure can be done for the mean CR as well, resulting in a total change in mean CR of 223.8% due to changing W_{ay} and 161.4% due to changing W_p. A two-way repeated measures ANOVA test confirms that there is a significant effect due to W_{ay} ($p < 0.05$), while the effect of W_p is not statistically significant ($p > 0.05$). There is also an interaction effect present ($p < 0.05$). Table IV shows the ANOVA results of the two-way repeated measures ANOVA tests on the entire curve, but also for each section of the curve. Interestingly, W_p shows a significant effect during the curve onset, while W_{ay} only shows an effect on the entire curve. This can be explained by the fact that W_{ay} influences the general feeling during the curve, while W_p mainly influences the motion during the curve onset and exit, because there high tilt rates generally occur.

For all curve segments, an interaction effect between the two parameters is present. This means that the effect of one parameter is influenced by the value of the other parameter. Figure 16 shows the mean rating per condition, where it can be seen that for low values of W_{ay} (0.4), increasing W_p results in a worse mean rating, for medium values of W_{ay} (1.0), the middle value of W_p results in the best mean rating and for high values of W_{ay} (1.6), increasing W_p results in a better mean rating. This stresses the importance of incorporating interaction effects in this type of research.

B. Performance of Conditions

The mean rating over time of all participants was shown in Figure 14, where it can be seen that the average rating differs per condition. Table III lists the numeric values of the mean ratings per condition, where it can be seen that C5 (baseline condition) received the lowest mean CR overall, therefore indicating that this is the most preferred condition. Figure 17 shows a boxplot of the mean ratings of all participants per

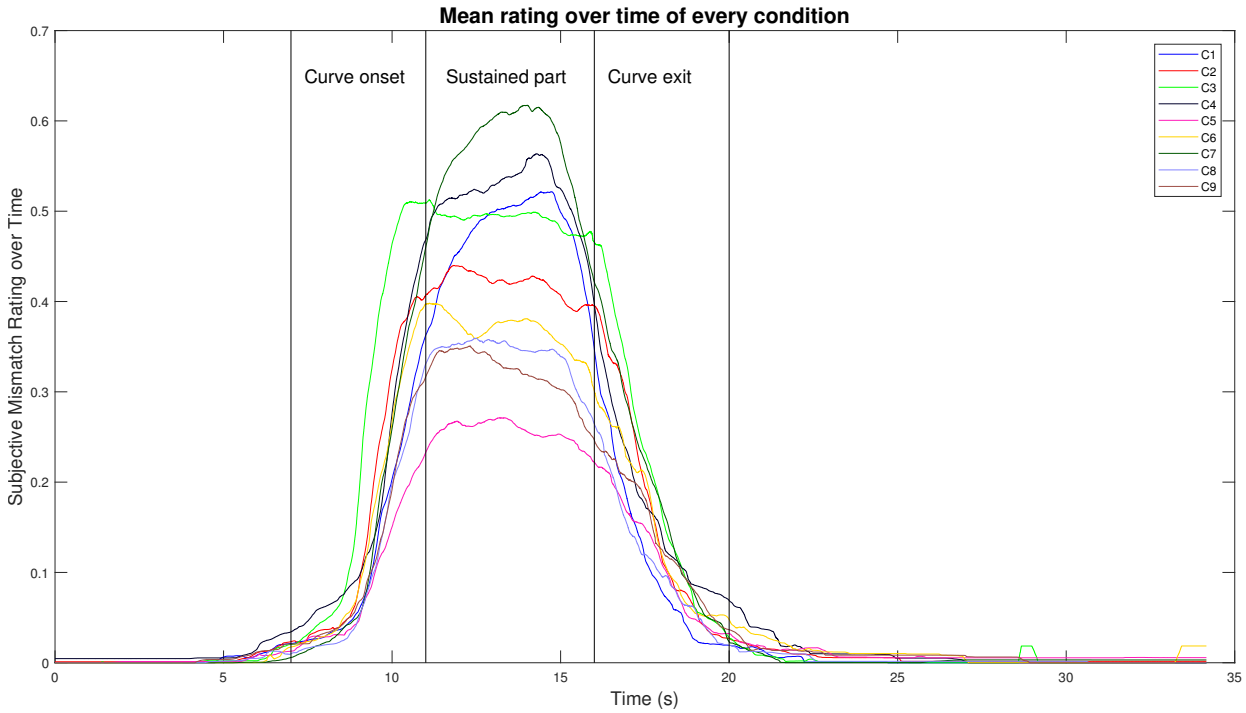


Fig. 14. The mean rating over time for all experiment conditions.

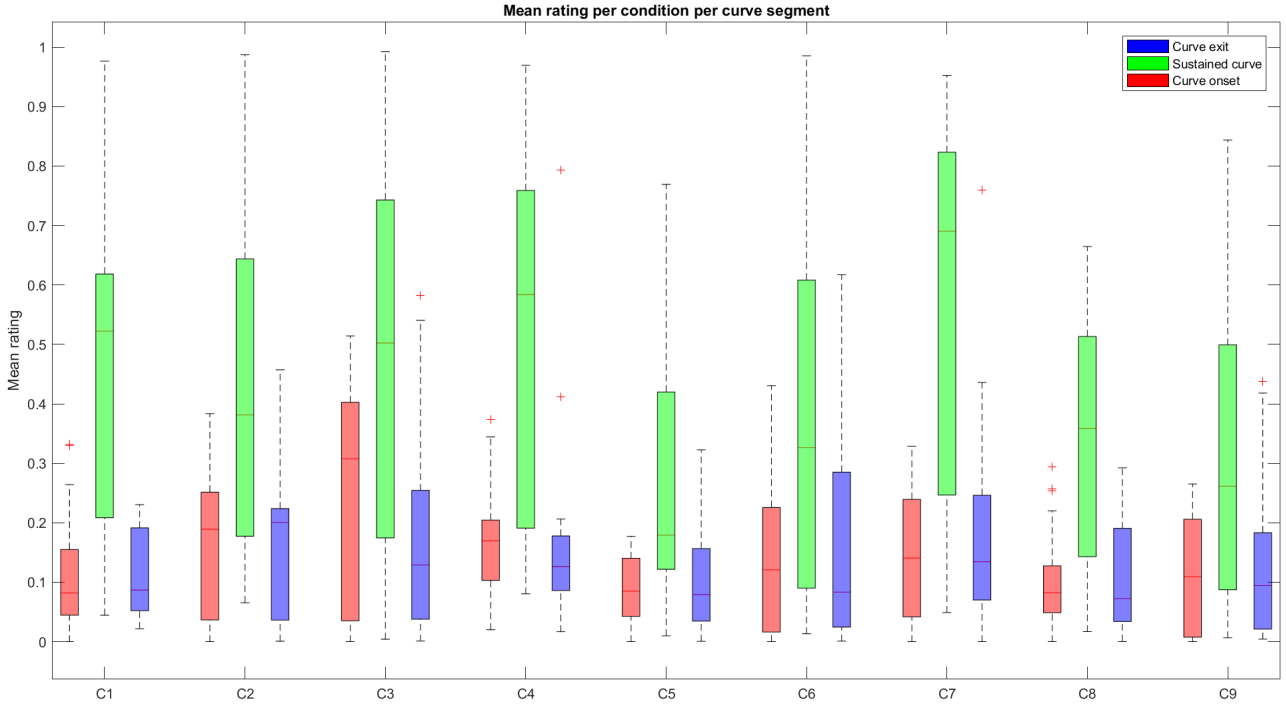


Fig. 15. Boxplot of the mean ratings per curve segment for all experiment conditions.

condition, where it can be seen that some median ratings lie within the spread of other conditions. Nevertheless, a Friedman Test on all nine conditions (18 participants, 3 repetitions) shows that there is a strong significant overall effect ($p < 0.01$). This is because smaller effects during each curve segment that are not significant on their own, add up to an overall significant effect.

To determine whether C5 differs significantly from the other conditions, a Wilcoxon signed-rank test was performed between C5 and all other conditions (see Appendix C). Based

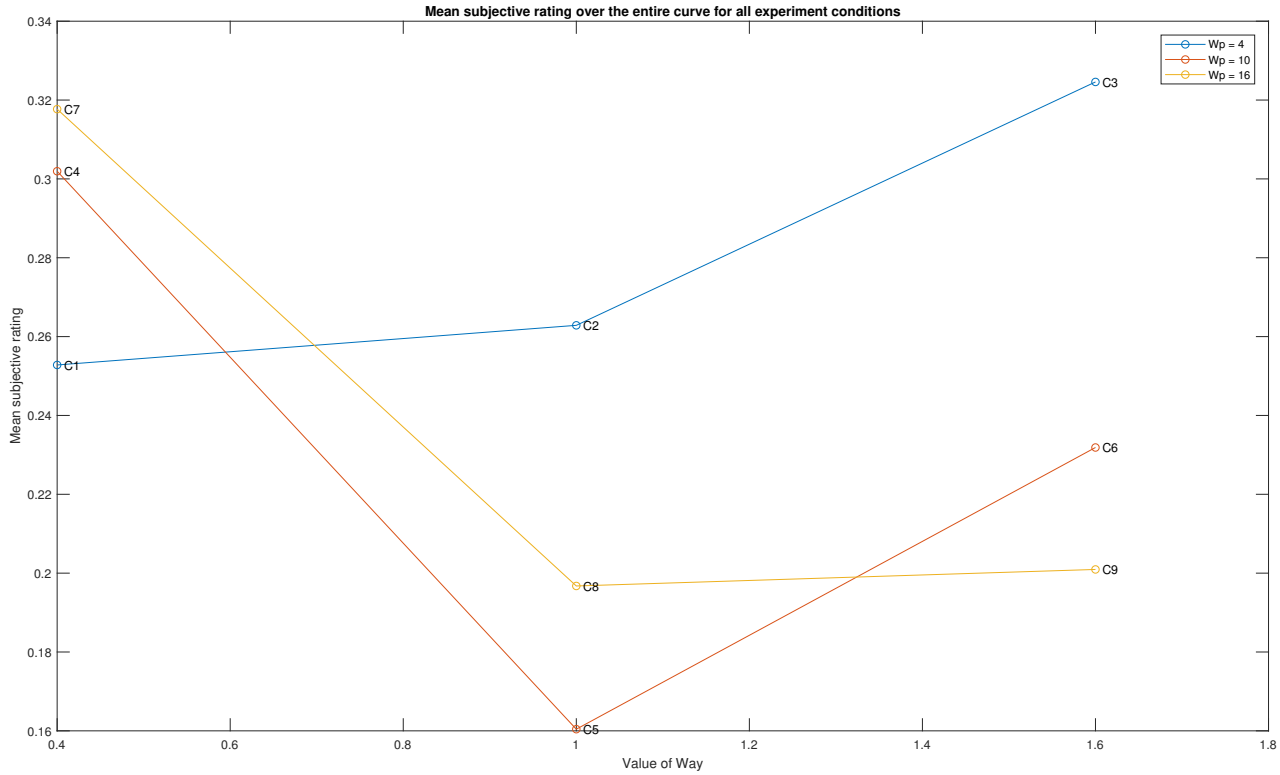


Fig. 16. The mean overall rating of all experiment conditions.

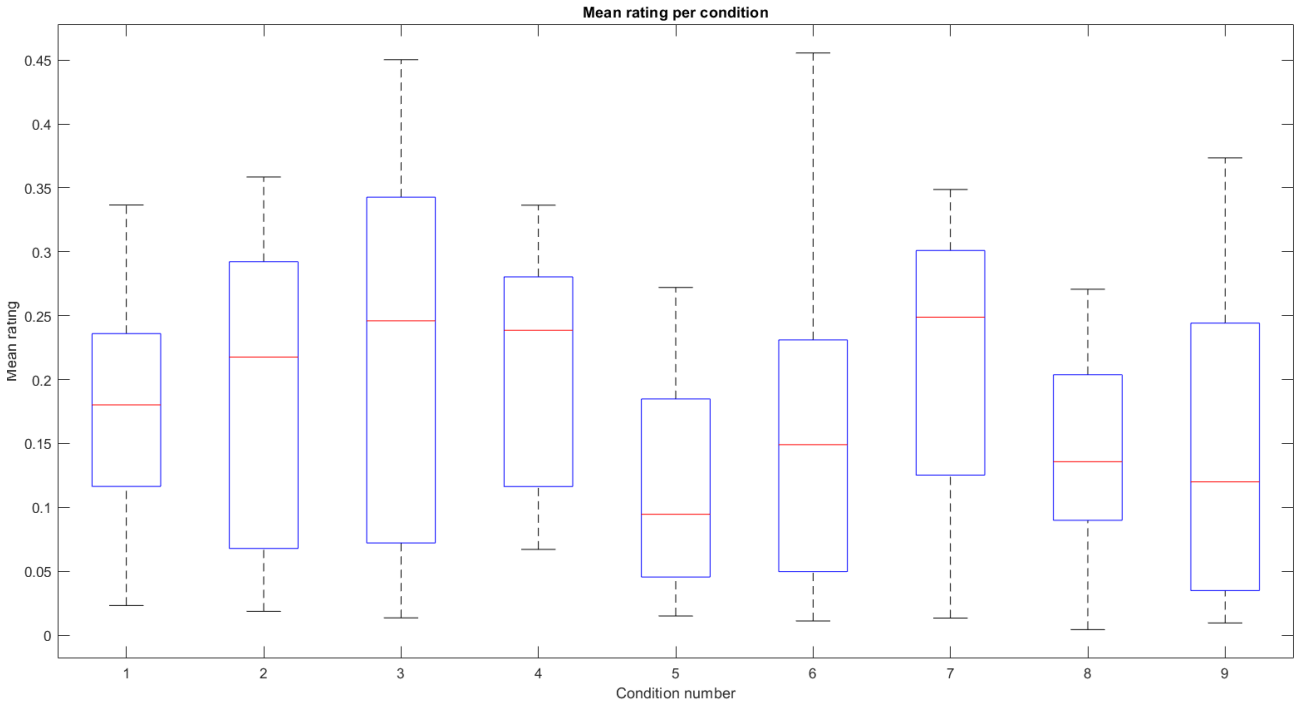


Fig. 17. Boxplot of the mean ratings for all experiment conditions.

TABLE IV
ANOVA RESULTS OF THE TWO-WAY REPEATED MEASURES ANOVA TEST ON THE EFFECTS OF BOTH PARAMETERS AND THEIR INTERACTION AND OF THE FRIEDMAN TEST.

	W_{ay}		W_p		Interaction		Friedman Test	
	F	Sig.[1]	F	Sig.[1]	F	Sig.[1]	F	Sig.[1]
Entire curve	3.65	*	1.69	-	8.08	**	20.4	**
Curve onset	1.56	-	6.01	**	8.70	**	11.7	-
Sustained part	2.54	-	3.16	-	5.43	**	18.7	*
Curve exit	0.40	-	0.62	-	4.77	**	3.4	-

[1] '**' indicates a strong significant effect ($p < 0.01$), '*' denotes significance ($p < 0.05$) and '-' means no significance ($p > 0.05$).

on this, C1, C2, C3, C4, C6 and C7 have been rated significantly worse than C5 ($p < 0.05$). However, when imposing a Sidak-correction ($p < 0.00639$, 8 tests), only C2 and C3 are significantly different from C5. This means that the results suggest that there is a difference, but due to the number of tests, a false positive becomes more likely and therefore the significance level tightens.

C. Correlation between RMSE/PCC and CR

Lastly, a correlation coefficient between the CR and the RMSE or PCC of a combination of motion channels has been calculated to determine whether any of the combinations are a good indication of what humans perceive as good or bad. Table V shows the values of the correlation coefficient for the different combinations of motion channels with RMSE and PCC. It can be seen that none of the combinations show a strong correlation (correlation > 0.6).

TABLE V
CORRELATION VALUES BETWEEN THE RMSE OR PCC OF A COMBINATION OF MOTION CHANNELS WITH THE MEAN CONTINUOUS RATING.

Signals	Correlation with RMSE	Inverse correlation with PCC
all	0.2293	0.2576
a_y, p	0.2145	-0.0362
a_x,a_y,p,q,r	0.2172	-0.0138
a_x,a_y,a_z	0.2038	0.3858
p,q,r	-0.1680	-0.1320

Additionally, although the RMSE and PCC do not show a strong correlation with the mean continuous ratings, another interesting observation can be made. If we compare the mean rating per experiment condition in Figure 16 with the RMSE in a_y in Figure 12 (upper middle plot), we see a similar trend for most conditions, except for C2, C3 and C6. These three conditions all surpassed the perception threshold for roll rate by a large margin and received a worse rating than one would predict based on the RMSE in a_y. This means that the RMSE in a_y is a good indicator when looking at the perceived motion quality during curve driving, but an error in p above the perception threshold decreases the perceived motion quality. Therefore, it will be checked whether a sum of the RMSE in a_y and the RMSE in $p_{abovethreshold}$ times a certain weight will result in a high correlation with the mean continuous mismatch rating. The RMSE in $p_{abovethreshold}$ is similar to other RMSE calculations, but now only the error above the perception threshold is considered. Figure 18 shows the correlation of CR with the RMSE in a_y and a weighted RMSE in $p_{abovethreshold}$ and indeed, there are weights that result in a high correlation. A weight of 49.4 results in the highest correlation of 0.9799, therefore having the potential to predict the distribution of perceived motion quality.

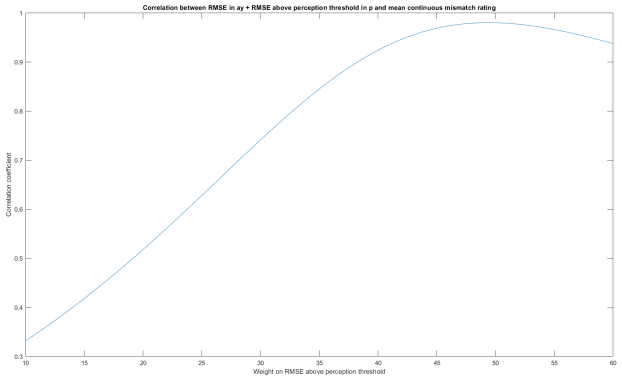


Fig. 18. Correlation between RMSE in a_y + a weighted RMSE above the perception threshold in p and the mean continuous mismatch rating.

VI. DISCUSSION

This paper presents a first investigation into the sensitivity of an MPC-based MCA to varying the parameters of its cost function. An experiment was set up based on the results of a preliminary analysis, where participants experienced nine different motion conditions as a result of changing the lateral acceleration and roll rate error weight parameters (W_{ay} and W_p, respectively) of the cost function of the MPC-algorithm. Participants continuously rated their perceived mismatch between visual and inertial information, which is indicative of the perceived motion quality [25].

It was found that W_{ay} has more influence on the mean continuous rating than W_p, just as was hypothesised based on the RMSE and PCC predictions. Conditions where W_{ay} was set to a higher relative value than W_p (C2, C3 and C6) resulted in a perceptible false cue in p, which is reflected by higher mean CRs during the curve onset and exit. On the other hand, most conditions where W_{ay} was set to a lower relative value than W_p (C4 and C7) resulted in poor tracking of a_y, which is reflected by higher mean CRs during the sustained part of the curve. With respect to the baseline condition, the following behaviour was observed:

- Increasing W_{ay} ensures that the algorithm tries to decrease the error in a_y more, but this might cause a perceptible false cue in p due to tilt-coordination.
- Increasing W_p causes the algorithm to decrease the error in p, but at the cost of the performance of a_y,

because a hexapod-based motion simulator is not able to reproduce sustained accelerations without the use of tilt-coordination.
- Decreasing W_{ay} also results in poor tracking of a_y.
- Decreasing W_p allows for more tilt-coordination, which introduces perceptible tilt rates, which is reflected in the CR.

Generally speaking, the following conclusions can be drawn: W_{ay} mainly influences the tracking performance of ay, i.e. increasing W_{ay} reduces the RMSE in a_y. W_p directly affects the curve onsets and exits, since only there a roll rate is present. Therefore W_p influences the tilt-coordination that is used. A two-way repeated measures ANOVA test indicated that the effect of W_{ay} is statistically significant, while the effect of W_p is not. Therefore, we can conclude that W_{ay} indeed has more influence on the perceived motion quality for the considered curve driving scenario and we can accept hypothesis **H1**.

The second hypothesis stated that C8 would be rated as the best condition, as C8 had the lowest total RMSE, while not exceeding the perception threshold in p due to tilt-coordination. However, it was found that not C8 but C5 (baseline condition) received the lowest overall rating, while C5 also contains perceptible tilt rates due to tilt-coordination, according to [26]. On the contrary, [27] argues that priority should be given to the minimisation of translational acceleration errors, even at the cost of perceptible tilt rates, which explains why C5 received a slightly lower mean CR than C8 (due to having a lower RMSE in a_y).

This means that the baseline condition, where the weight values approximately compensate for the difference in units (as stated in [16]), is a good compromise between tracking all stimuli while not introducing large perceptible tilt rates due to tilt-coordination. Although it was shown that C5 received the lowest mean continuous rating, no significant differences were found between C5 and C8. At the same time, it can also not be shown that C8 is the best performing condition. Therefore **H2** is rejected.

Finally, it was also checked whether there is any correlation between the predicted RMSE or PCC and the measured CR. From the correlation coefficients for different combinations of the RMSE or PCC with the mean continuous mismatch rating (Table V), it can be concluded that there is no strong correlation between the mean continuous rating and one of the combinations listed. However, a weighted sum of the RMSE in a_y and RMSE in $p_{abovethreshold}$ showed a high correlation of 0.9799. Humans prefer strong translational accelerations, to the point where priority should be given to the minimisation of translational acceleration errors, even if that means that perceptible tilt rates are introduced [27]. This means that the relevant translational accelerations (in this case lateral acceleration) should show correlation with the CR. But we can see that surpassing the perception threshold eventually leads to a worse perceived motion quality, which explains why the sum of the RMSE in a_y and a weighted (49.4) RMSE in $p_{abovethreshold}$ shows a high correlation with the CR. This result implies that a small RMSE in p due to tilt-coordination does not affect the perceived motion quality, but the influence of the RMSE in $p_{abovethreshold}$ is five times as large as the influence of the RMSE in a_y (after accounting for a difference in units). Therefore we can conclude that false roll rate cues above the perception threshold are more detrimental to the perceived motion quality than lateral acceleration scaling errors, for this curve driving scenario.

However, this result should be handled with care, because even though it follows from reason that a curve driving motion simulation has a high perceived motion quality when the RMSE in a_y (the dominant stimulus) is small and a large false cue in p decreases the perceived motion quality, there is no logical explanation for the weight penalty on the RMSE in $p_{abovethreshold}$. Note that in this experiment, the RMSE of all other motion channels remained fairly constant, but those will also affect the CR. Therefore, predicting the distribution of the perceived motion quality by looking at the RMSE of specific forces plus a weighted RMSE of tilt rates above the perception threshold could be tested to verify that this is indeed the case in general and not only for this experiment. Also note that although [23] found that the PCC is a good indicator for the perceived motion quality, no such conclusion can be drawn from this research.

This research is a first investigation into the sensitivity of the MPC-algorithm due to changing the parameters, and due to the number of parameters of the algorithm, most of them needed to be kept constant. But as is apparent from this research, even two parameters can change the behaviour of the algorithm significantly. Therefore, the results of this research are only applicable when all other parameters are as used in this research. Nevertheless, it will help future users in the tuning process by providing an understanding of the influence of the parameters of the cost-function. Future studies could investigate the impact of different parameters, such that an overview is created of the influence of all parameters on the behaviour of the algorithm.

VII. Conclusion

In this paper, the effect of percentage-wise variation of two parameters (W_{ay} and W_p) of the cost-function of the MPI MPC-based MCA on the behaviour of the algorithm has been studied. Both investigated parameters had a significant impact on the perceived motion quality, although it was found in this experiment that W_{ay} causes an overall larger change in the continuous rating than W_p. Increasing W_{ay} introduces perceptible errors in p, decreasing W_{ay} decreases the tracking performance of ay, increasing W_p also decreases the tracking performance of ay and decreasing W_p also introduces perceptible errors in p. These observations should help future users of an MPC-based MCA in tuning the parameters of such an algorithm and in understanding the behaviour. Surprisingly, it was concluded from the experiment that condition C5 (the baseline condition) received the lowest continuous perceived mismatch rating, therefore having the highest perceived motion quality. Finally, a combination of the RMSE of a_y plus a weighted RMSE of p above the perception threshold showed a high correlation with the mean continuous ratings, therefore having the potential to become a method for predicting how humans perceive motion quality.

REFERENCES

[1] J. H. Hogema, M. Wentink, and G. P. Bertollini, "Effects of Yaw Motion on Driving Behaviour, Comfort and Realism," in *Proceeding of the Driving Simulation Conference 2012, Paris, France*, Sep. 2012.

[2] P. R. Grant, B. Yam, R. J. A. W. Hosman, and J. A. Schroeder, "Effect of Simulator Motion on Pilot Behavior and Perception," *Journal of Aircraft*, vol. 43, no. 6, pp. 1914–1924, November/December 2006.

[3] A. R. Valente Pais, M. Wentink, M. M. van Paassen, and M. Mulder, "Comparison of Three Motion Cueing Algorithms for Curve Driving in an Urban Environment," *Presence: Teleoperators & Virtual Environments*, vol. 18, no. 3, pp. 200–221, Jun. 2009.

[4] B. Conrad and S. F. Schmidt, "Motion Drive Signals for Piloted Flight Simulators," National Aeronautics and Space Administration, Ames Research Center, Tech. Rep. NASA CR-1601, 1970.

[5] M. A. Nahon and L. D. Reid, "Simulator Motion-Drive Algorithms: A Designer's Perspective," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 2, pp. 356–362, 1990.

[6] P. R. Grant and L. D. Reid, "Motion Washout Filter Tuning: Rules and Requirements," *Journal of Aircraft*, vol. 34, no. 2, pp. 145–151, 1997.

[7] R. V. Parrish, J. E. Dieudonne, R. L. Bowles, and D. J. Martin, Jr., "Coordinated Adaptive Washout for Motion Simulators," *Journal of Aircraft*, vol. 12, no. 1, pp. 44–50, 1975.

[8] M. A. Nahon, L. D. Reid, and J. Kirdeikis, "Adaptive Simulator Motion Software with Supervisory Control," *Journal of Guidance, Control and Dynamics*, vol. 15, no. 2, pp. 376–383, 1992.

[9] A. R. Naseri and P. R. Grant, "An Improved Adaptive Motion Drive Algorithm," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, San Francisco (CA)*, no. AIAA-2005-6500, 2005.

[10] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Mazi, "Model-based predictive motion cueing strategy for vehicle driving simulators," *Control Engineering Practice*, vol. 17, pp. 995–1003, 09 2009.

[11] M. Baseggio, A. Beghi, M. Bruschetta, F. Maran, and D. Minen, "An MPC approach to the design of motion cueing algorithms for driving simulators," in *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA*, Oct. 2011.

[12] N. J. Garrett and M. C. Best, "Model predictive driving simulator motion cueing algorithm with actuator-based constraints," *Vehicle System Dynamics*, vol. 51, no. 8, pp. 1151–1172, 2013.

[13] D. Cleij, J. Venrooij, P. Pretto, M. Katliar, H. H. Bülthoff, D. Steffen, F. W. Hoffmeyer, and H.-P. Schner, "Comparison between filter-and optimization-based motion cueing algorithms for driving simulation." in *Transportation Research Part F: Traffic Psychology and Behaviour*, 2017.

[14] M. Katliar, K. N. D. Winkel, J. Venrooij, P. Pretto, and H. H. Bülthoff, "Impact of mpc prediction horizon on motion cueing fidelity." in *Driving Simulator Conference*, 2015.

[15] A. Beghi, M. Bruschetta, and F. Maran, "A real time implementation of MPC based Motion Cueing strategy for driving simulators," in *Proceedings of the 51st IEEE Conference on Decision and Control, Maui, Hawaii*, Dec. 2012.

[16] M. Katliar, J. Fisher, G. Frison, M. Diehl, H. Teufel, and H. H. Bülthoff, "Nonlinear model predictive control of a cable-robot-based motion simulator." in *International Federation of Automatic Control*, 2017.

[17] M. Katliar, F. Drop, H. Teufel, M. Diehl, and H. H. Bülthoff, "Real-time nonlinear model predictive control of a motion simulator based on a 8-dof serial robot." in *17th European Control Conference (ECC 2018)*, 2018.

[18] D. Cleij, J. Venrooij, P. Pretto, D. M. Pool, M. Mulder, and H. H. Bülthoff, "Continuous rating of perceived visual-inertial motion incoherence during driving simulation," in *Proceedings of the Driving Simulation Conference 2015 Europe, Tübingen, Germany*, Sep. 2015.

[19] J. Qin and T. Badgwell, "An overview of industrial model predictive control technology," in *AIChE Symposium Series*, vol. 93, 01 1997.

[20] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. MaÃzi, "MPC Based Motion Cueing Algorithm: Development and Application to the ULTIMATE Driving Simulator," in *Proceedings of the Driving Simulation Conference 2004 Europe, Paris, France*, 2004, pp. 221–233.

[21] M. Diehl, H. Bock, J. P. Schlder, R. Findeisen, Z. Nagy, and F. Allgwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, pp. 577–585, 06 2002.

[22] F. M. Nieuwenhuizen and H. H. Bülthoff, "The MPI CyberMotion Simulator: A Novel Research Platform to Investigate Human Control Behavior," *Journal of Computing Science and Engineering*, vol. 7, no. 2, pp. 122–131, Jun. 2013.

[23] S. Casas-Yrurzum, I. Coma, J. V. Riera, and M. Fernández, "Motion-cuing algorithms - characterization of users' perception," *Human Factors The Journal of the Human Factors and Ergonomics Society*, vol. 57, 02 2015.

[24] A. Berthoz, W. Bles, H. H. Bülthoff, B. J. Correia Gräcio, P. Feenstra, N. Filliard, R. Hähne, A. Kemeny, M. Mayrhofer, M. Mulder, H.-G. Nusseck, P. Pretto, G. Reymond, R. Schlässelberger, J. Schwandter, H. J. Teufel, B. Vailleau, M. M. van Paassen, M. Vidal, and M. Wentink, "Motion Scaling for High-Performance Driving Simulators," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 3, pp. 265–276, May 2013.

[25] D. Cleij, J. Venrooij, P. Pretto, D. M. Pool, M. Mulder, and H. H. Bülthoff, "Continuous Subjective Rating of Perceived Motion Incongruence During Driving Simulation," *IEEE Transactions on Human-Machine Systems*, 2017.

[26] E. Groen and W. Bles, "How to use body tilt for the simulation of linear self motion," *Journal of Vestibular Research: Equilibrium and Orientation, 14(5), 375385*, 2004.

[27] H. Jamson, "Motion cueing in driving simulators for research applications," Ph.D. dissertation, The University of Leeds, Institute for Transport Studies, Nov. 2010.

APPENDIX A
PARTICIPANT FILTERING

All participants have performed three experiment runs, and the use of the continuous subjective rating method is only valid when participants are consistent over multiple runs. Therefore, Cronbach's Alpha will be used as a measure of consistency and in case of a poor score, participants will be excluded from the results. Cronbach's Alpha is calculated by using Equation 6, where K = number of items (in this case, 3), Total Variance is the variance of the items' sum and Item Variance is the variance of the individual items. An example of a poor consistency is shown in Figure 19, where it can be seen that the three experiment runs are not rated consistently, which also follows from a low Cronbach's Alpha (0.2326). A high consistency is shown in Figure 20, where all experiment runs are rated very similar and Cronbach's Alpha is high (0.9609). Table VI shows the values of Cronbach's Alpha for all participants and based on this, participants 10 and 18 were removed from the results due to a poor consistency.

$$c = \frac{K}{K-1}\left(\frac{TotalVariance - \Sigma_{i=1}^{K} ItemVariance}{TotalVariance}\right) \quad (6)$$

APPENDIX B
DIFFERENCES BETWEEN RATING A LEFT OR A RIGHT CURVE

One participant indicated to find it difficult to give an equal rating for left/right curves, since the visuals are not symmetrical due to the driver being positioned in the left seat of the vehicle. Therefore, it has been checked whether there is a difference in rating between left and right curves by performing a Wilcoxon signed rank test on the data for each curve. The results of this can be found in Table VII. Because there is no significant difference between a left or a right curve for all conditions, all data can be

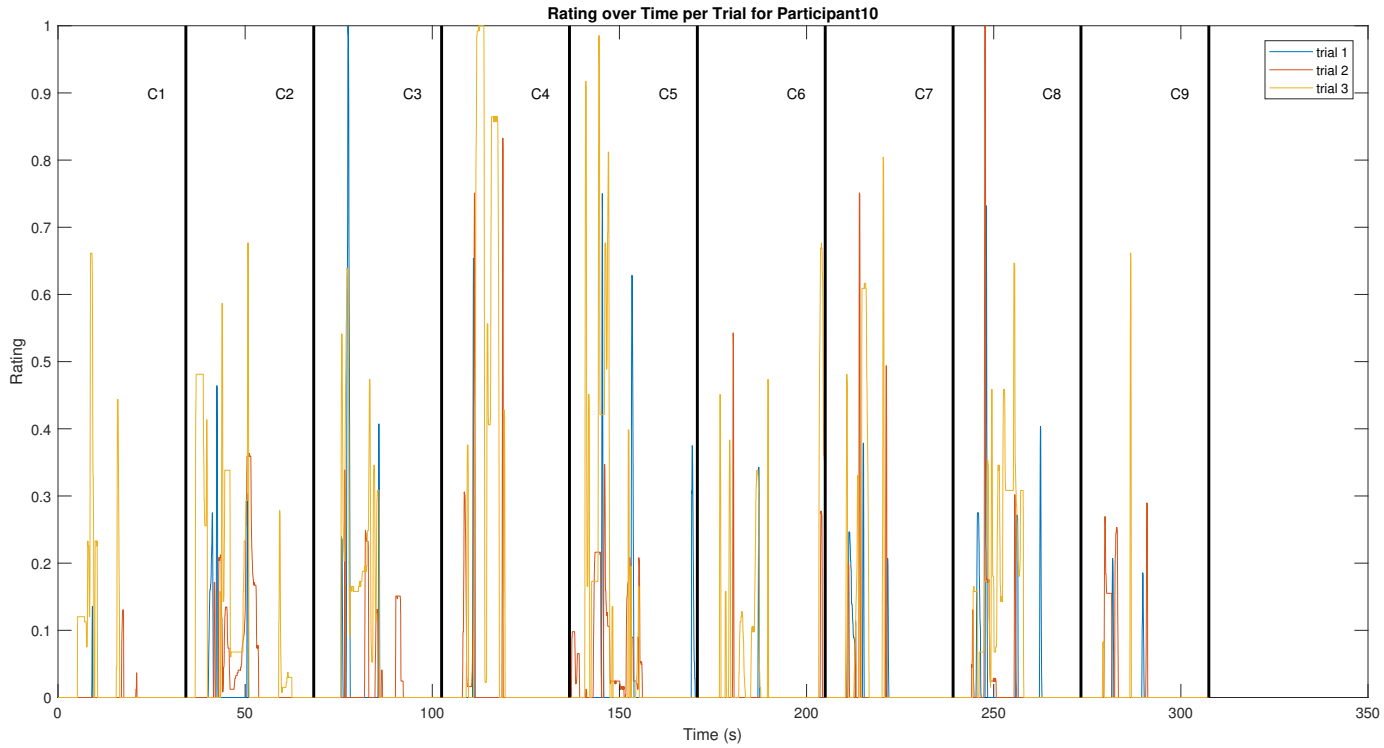


Fig. 19. Rating over time per experiment run for participant 10.

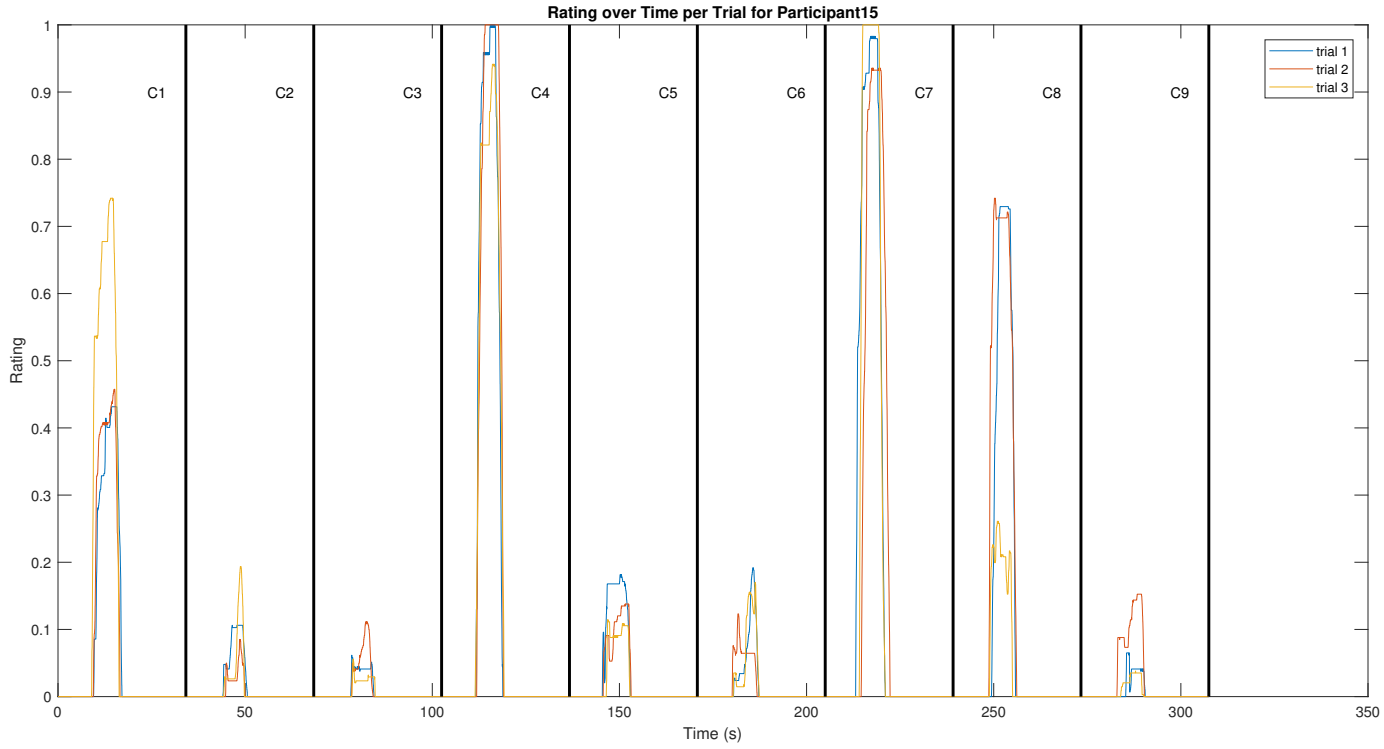


Fig. 20. Rating over time per experiment run for participant 15.

TABLE VI
THE VALUES OF CRONBACH'S ALPHA FOR ALL PARTICIPANTS. BOLD VALUES INDICATE INCONSISTENT PARTICIPANTS.

Participant	Cronbach's Alpha		Participant	Cronbach's Alpha
1	0.8848		11	0.9174
2	0.9536		12	0.9059
3	0.8604		13	0.9273
4	0.9325		14	0.9224
5	0.9106		15	0.9609
6	0.9151		16	0.8225
7	0.9219		17	0.8487
8	0.8945		**18**	**0.6180**
9	0.9181		19	0.8650
10	**0.2326**		20	0.9715

TABLE VII
P-VALUES OF THE WILCOXON SIGNED RANK TEST ON LEFT/RIGHT DATA FOR EACH EXPERIMENT CONDITION.

Condition	p-value
1	0.0854
2	0.5277
3	0.6475
4	0.0840
5	0.4631
6	0.3271
7	0.3028
8	0.6051
9	0.2311

APPENDIX C
WILCOXON SIGNED RANK TEST RESULTS

To determine if there are significant differences between experiment conditions, a Wilcoxon signed-rank test was performed between all experiment conditions individually. The results are shown in Table VIII.

TABLE VIII
P-VALUES OF THE WILCOXON SIGNED RANK TEST BETWEEN ALL CONDITIONS.

	C1	C2	C3	C4	C5	C6	C7	C8
C2	0.983							
C3	0.372	0.012						
C4	0.122	0.586	0.679					
C5	0.043	0.002	0.001	0.018				
C6	0.528	0.231	0.001	0.306	0.022			
C7	0.094	0.420	0.913	0.500	0.008	0.215		
C8	0.011	0.199	0.053	0.014	0.157	0.679	0.014	
C9	0.327	0.005	0.001	0.085	0.102	0.094	0.094	0.879

II

Preliminary Thesis Report

This part has already been graded for AE4020 Literature Study

1

INTRODUCTION

In these days, increasing use is made of motion simulators. They are being used to investigate human perception, cognition and action, for example by research organisations and by R&D departments of companies. Next to research, they are also used to test new innovations in a safe and controllable environment. Several studies found that the addition of motion in simulations increases the perceived realism, while also increasing the performance of drivers [1] [2]. However, bad or false motion cues decrease these aspects, meaning that no motion is favoured over bad or false motion [3]. The challenge in providing motion cues is to simulate the vehicle motions as accurately as possible, while keeping the motion platform within its boundaries. This is handled by a so called motion cueing algorithm (MCA).

1.1. PROBLEM STATEMENT

Currently, most of the MCAs are based on washout filters [4], [5]. This strategy is based on motion washout, where the simulator will slowly move back to its neutral position whenever possible. On top of that, the strategy makes use of tilt coordination, where low frequency motion is filtered out, since low frequency motion required a lot of physical space and would cause the motion platform to exceed its physical limits. Instead, this low frequency motion is replicated by tilting the motion platform without exceeding human perceptual thresholds. The gravity, now under an angle, gives the feeling of accelerating in a certain direction. The high frequency motion is not filtered but usually scaled down. Both of these techniques help in keeping the simulator within its physical limits, which is ultimately done by tuning this algorithm. There are two downsides to this approach. Firstly, tuning the washout filter has to be done by an expert through a trial-and-error process, since the parameters that have to be tuned (cut-off frequency, damping and gain) are not directly related to the output of the simulator. Secondly, the goal of the tuning process is to keep the simulator within its physical limits, while providing the most realistic motion. It is not possible to explicitly account for simulator limits, so this is done by selecting the worst case scenario of the entire simulation and making sure this scenario does not exceed the simulator limits. The consequence of this tuning process is that all motion will be scaled down the same amount, which is undesirable.

To overcome these longstanding challenges in motion cueing, a new type of algorithm, based on model predictive control, is currently being developed by the Max Planck Institute for Biological Cybernetics in Tübingen [6]. The principle behind MPC-based MCAs is that at every time step, the future motion of the simulator is predicted for some time interval (prediction horizon), and optimal simulator control inputs are calculated for this interval using a dynamic model of the simulator. Each time step, only the first of these optimal control inputs is send to the simulator. In case of offline simulations (where the control input is known in advance), the future cues are already known, meaning that the prediction can be called perfect. In case of online simulations (with the user giving control inputs), predicting future motions becomes a lot more difficult, because it is not possible to perfectly predict human behaviour.

Offline simulations with an unlimited prediction horizon and perfect prediction have shown promising results for MPC-based MCAs [7]. However, online simulations have a worse prediction quality, and on top of

that they require the prediction horizon to be limited due to the computational costs of this algorithm and the unreliability of longer predictions. This increases the need for parameter tuning, but currently there is no clear overview on how all parameters influence the behaviour of the MPC-controller, while it would be of great help in understanding the behaviour of an MPC-based MCA and in tuning the parameters of the algorithm.

1.2. PROJECT GOAL

The goal of this research is to investigate the sensitivity of the parameters of the cost function of the MPC-based MCA of the Max Planck Institute for Biological Cybernetics. The algorithm has a large number of adjustable parameters, and this project will only incorporate the most important parameters in the sensitivity analysis. Firstly, an offline sensitivity analysis will be carried out, where the effects of changing parameters will be measured using objective measures related to perceived fidelity (using computer simulations). This report will be concluded after this analysis, but the research continues by executing a human-in-the-loop experiment to determine the effects on human perception and to validate the sensitivity analysis. The results of this experiment will be reported in a paper that will be written at a later stage.

From the literature review, the following research question has been constructed: *How do the error weight parameters of the cost function of a model predictive control-based motion cueing algorithm influence the human perceived motion on a hexapod motion simulator?*
This main question can be subdivided into the following sub-questions that will be answered in this report:

- What are the challenges in motion cueing in general?
- What are the challenges of a model predictive control-based motion cueing algorithm?
- What is the purpose of all parameters of the MPC-controller?
- How can the influence of changing the parameters on the perceived fidelity be measured/analysed objectively?
  - Which objective measures are related to the perceived cueing quality?
  - How are several of these measures combined into a single measure of sensitivity?
  - How should this single measure of sensitivity be analysed, knowing that the sensitivity can depend on other parameters?

Based on the outcome of the first part of this research, an experiment will be conducted in which the perceived fidelity will be measured using human participants. The main question for that last part will be:

- How do the parameters influence the perceived fidelity of the motion presented by the simulator?

1.3. RESEARCH SCOPE

The aim of this research is to deliver a thorough analysis, so instead of incorporating all possible parameters in the analysis, the scope of this research will be narrowed down by focusing on the most important aspects of this problem. In order to do this, a number of (big) decisions should be made beforehand. First of all, there are numerous parameters that influence the behaviour of the MPC controller, the cost function alone already contains over 100 parameters. Their functions will be explained later in Chapter 3, but for now it is important to understand that only the output error weights will be investigated thoroughly. This is because these parameters are most directly related to the output of the MCA.

Secondly, there are two different types of simulators available for this research at the Max Planck Institute for Biological Cybernetics, the CyberMotion Simulator (CMS) and the CyberPod Simulator (CHS). The CMS is an 8-degree-of-freedom motion simulator, consisting of a robot arm with a cabin attached, mounted on a linear rail system [8]. Although this simulator has more physical space and more dexterity [9], it is also a much less common simulator. Since this research is meant to give future users of an MPC-based MCA a first insight in its possibilities, it makes more sense to perform the research on a commonly available simulator. Although a hexapod simulator has more physical limits than the CMS, which might even influence the results, this research will be done using the CHS.

Next to that, the type of manoeuvre also influences the behaviour of the algorithm. Sustained accelerations are the most difficult manoeuvres to simulate, since either a large physical space is needed to generate such an acceleration, or tilt coordination has to be applied. Most simulators are mainly limited by their physical space, making tilt coordination a necessary technique. Sustained curve driving is the most challenging manoeuvre for motion simulators and research on driving simulation often looks at curve driving [10] [11], so the manoeuvre that will be used throughout this project is a car driving through a sustained curve as well.

Finally, a decision has to be made about the type of prediction that will be used. All possible options will be discussed later in Chapter 3, but the the scope of this research will be limited to the parameters of the cost function. Therefore, the prediction strategy will be kept simple, meaning that the prediction of future motions will be the same as the current motion, also called a constant prediction.

1.4. REPORT STRUCTURE

The structure of this preliminary thesis is built up as follows. First, in Chapter 2, a general introduction to motion cueing is given. The classical way of translating desired motion to simulator motion is discussed, along with some information about the most common simulator, the hexapod. This chapter will be concluded with an overview on all available motion cueing algorithms.

Chapter 3 goes deeper into the subject of a model predictive control based MCA. The basic principle of MPC will be explained, after which the version of the Max Planck Institute for Biological Cybernetics will be shown in more detail. The functions of all parameters will be explained, and also different types of producing a prediction of the motion in the future will be described.

Then a short look into objective and subjective metrics for motion simulation follows in Chapter 4, which is basically a preparation for the sensitivity analysis of the output error weight parameters. In Chapter 5, this sensitivity analysis is conducted, where first the use case is described. The sensitivity of the parameters is measured using multiple metrics that are described in Chapter 4, along with an explanation of the behaviour of the MPC-controller.

Lastly, in Chapter 6, the results of the sensitivity analysis are summarised, along with a proposal for an experiment in which the perceived sensitivity to changes in the MPC-parameters will be measured using a continuous subjective rating method, first used by [12].

2

SIMULATOR MOTION CUEING

Simulations have been around for a very long time and a lot of different types of simulations exist. In a simulation, the goal is to mimic a certain situation, setting or environment, such that a subject placed in the simulator experiences a situation as if everything happening is real. The main advantages of simulators is that they provide a controllable environment that is easily manipulated, allowing for a safer and cheaper way of testing certain innovative or dangerous situations, compared to real-life testing [13]. Although acquisition costs can be high for simulators, so can the costs for building prototypes, and creating situations in a simulation environment is usually cheaper than creating the same situation in the real world. Another advantage is that simulators enable repetitions of the exact same situation over and over again and on top of that, many variables are easily changed in a simulation environment, making simulators being favoured over real-life testing in many research experiments.

While simulators can have endless possibilities, they have one big disadvantage, and that is the fact that it is difficult to determine to what extent the simulation corresponds to the real world situation. This can only be done by comparing the simulation to the real world situation, but in some cases this defeats the purpose of a simulator. And next to the problem of validating the resemblance of the simulation and the real world situation, motion simulators potentially introduce motion sickness. There is still no consensus on what causes motion sickness, and thus it is very hard to determine whether motion sickness in a simulator is caused by the simulator or the simulated situation. Nevertheless, motion simulators are an important instrument in both research and development, as well as in training. Converting the desired motion into simulator motion is handled by a motion cueing algorithm (MCA).

2.1. INTRODUCTION TO SIMULATOR MOTION CUEING

Training pilots/professional drivers and studying human motion perception and control behaviour is often done through the use of motion simulators [14] [15] [16]. A motion simulator is a device capable of giving the operator the feeling that the vehicle is moving, which is done through actual movements of the motion platform. It has been shown that motion cues positively affect driving behaviour and improve performance [17], and that the reproduction of inertial cues contributes to a realistic driving behaviour [18] [19]. Therefore it is important that the driver of the simulator perceives motion as realistic, but one should keep in mind that a simulation without any motion cues is favoured over one with bad or false motion cues [3]. The desired inertial cues are usually obtained from vehicle models or from real vehicle data. An MCA translates these desired inertial cues into simulator platform motion. Reproducing the desired inertial signals is not as easy as it sounds, because they consist of first and second order derivatives of angles and positions, respectively. Integrating these quantities over time will result in large values for positions and angles, but a motion simulator has physical limits. This means that the desired quantities cannot be reproduced one-to-one and thus discrepancies will occur between the desired cues and the cues that the simulator produces. The purpose of an MCA is to reproduce the desired inertial cues as accurately as possible, while staying within the simulator limits (displacements, velocities and accelerations).

Motion simulators are being used to simulate many different types of motion, for example car driving, fly-

ing, sailing, being in an elevator, riding an attraction, basically everything that moves. In all these cases, the goal is to have the human perceive the motion as if he or she is actually in the real situation. So in a good motion simulator, a human does not perceive being in a motion simulator. And in that wording lies the key mechanism of most motion simulators: perception, because the human vestibular system is not perfect. The main shortcoming of the human vestibular system is the fact that we are not able to sense low angular velocities and accelerations, or small deviations in them. Also, it is hard for a human to distinguish between being tilted or being subjected to an acceleration, which allows us to make use of so called tilt-coordination. Tilt-coordination is a technique where the feeling of a long, sustained, translational acceleration is given by tilting the platform of a motion simulator, rather than actually moving the platform a long distance. For this to work, the tilting of the simulator platform must happen without the human noticing, which is only possible due to the inability of humans of detecting low tilt rates. The actual value at which humans start noticing tilt rates is dependent on many factors, but a commonly used value is $3°/s$ [20], meaning that using only tilt-coordination for the simulation of linear motion is perceived as linear motion if the tilt rate is below $3°/s$. However, this value is increased significantly in case of an active driving task [21], up to double the value of what has been found earlier [22].

Humans are able to perceive inertial forces and angular velocities, which is why motion simulators aim to reproduce those specific signals. But as already mentioned before, humans are not perfect in perceiving accelerations and velocities, meaning that slight variations in these vestibular stimuli are not perceived as different and low forces and velocities are not noticed at all. For example, if a certain acceleration force is scaled down by as much as 10 %, the human cannot distinguish it from the unscaled acceleration force [23]. This entails that a perfect reproduction of inertial forces is not necessary for a perfect motion simulation. Therefore it can be argued that in a motion simulation with human subjects, perceptual realism is more important than a perfect reproduction of desired vehicle motion, although both are related. However, perceptual fidelity is a lot more difficult to measure, due to its dependence on the human subject.

2.2. MOTION SIMULATORS

There are many different types of motion simulators around, for example the the NASA Ames Vertical Motion Simulator [24] or the CyberMotion Simulator of the Max Planck Institute for Biological Cybernetics [8]. But the most common motion simulator has the Stewart platform or hexapod configuration as its motion system [25], of which many variations exist. The Stewart platform is a mechanism that is able to cover all six degrees of freedom (three linear and three angular movements) using six legs. Older and larger types of the hexapod simulator often make use of hydraulically driven legs, such as the Simona Research Simulator of the TU Delft (see Figure 2.2), and newer and smaller ones are driven electrically, such as the Bosch eMotion 1500 of the Max Planck Institute for Biological Cybernetics (see Figure 2.1).

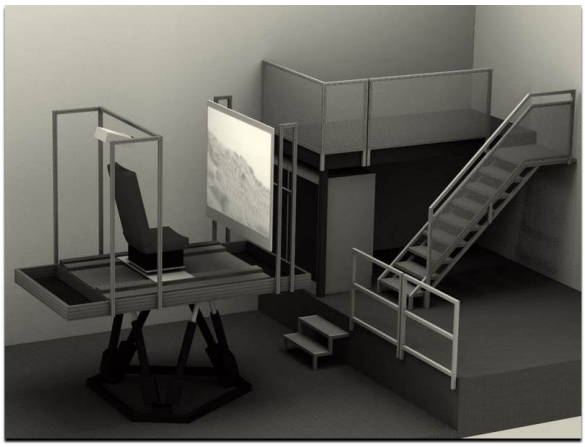


Figure 2.1: The hexapod simulator of the Max Planck Institute for Biological Cybernetics.

Figure 2.2: Simona Research Simulator of the TU Delft

2.3. TYPES OF MOTION CUEING ALGORITHMS

Currently, motion cueing algorithms can be subdivided in four different types: classical washout filters, adaptive classical washout filters, optimal control-based MCA's and finally model predictive control-based MCA's [26]. The most common type of MCA is still the oldest of these four, the classical washout filter, which is basically a set of motion filters that translate desired motion into motion that fits within the physical limits of the simulator.

2.3.1. CLASSICAL WASHOUT FILTER

The original way of designing an MCA [4] [5] consists of two parts. It filters out low-frequency motions, the ones that are slowly changing, because this type of motion would require a lot of simulator space. These long, sustained motions are instead simulated by tilting the motion platform. This causes the gravity vector to be no longer aligned with the vertical axes of the human body, creating a feeling of acceleration due to the inability of humans of distinguishing between the two motions. This technique is called tilt-coordination [3]. Of course, tilting the platform to simulate low-frequency motions should preferably be done below the detection threshold of $3°/s$, because otherwise the human will notice being rotated. So when a real vehicle performs an acceleration manoeuvre, a motion simulator will move forward to simulate the high-frequency acceleration and simultaneously the platform will rotate so simulate the low-frequency motion. The forward motion of the motion platform cannot last long due to the physical limits, but the keeping the platform at an angle can last forever.

While the low-frequency motion is filtered and simulated using tilt-coordination, the high-frequency motion and is only scaled. A schematic representation of how this algorithm can be implemented is shown in Figure 2.3. It can be seen that the linear acceleration is divided into high-frequency and low-frequency motion, and both the high-frequent acceleration and the rotational velocities are only scaled. At all times, the algorithm tries to move the simulator back to its neutral position, which is called washout, hence the name washout filter. This also has to be done below the detection threshold of humans, because this motion is often opposite of what should be happening. Washout is done because the possible movements of the motion platform are highest when the motion platform is in the middle of physical limits.

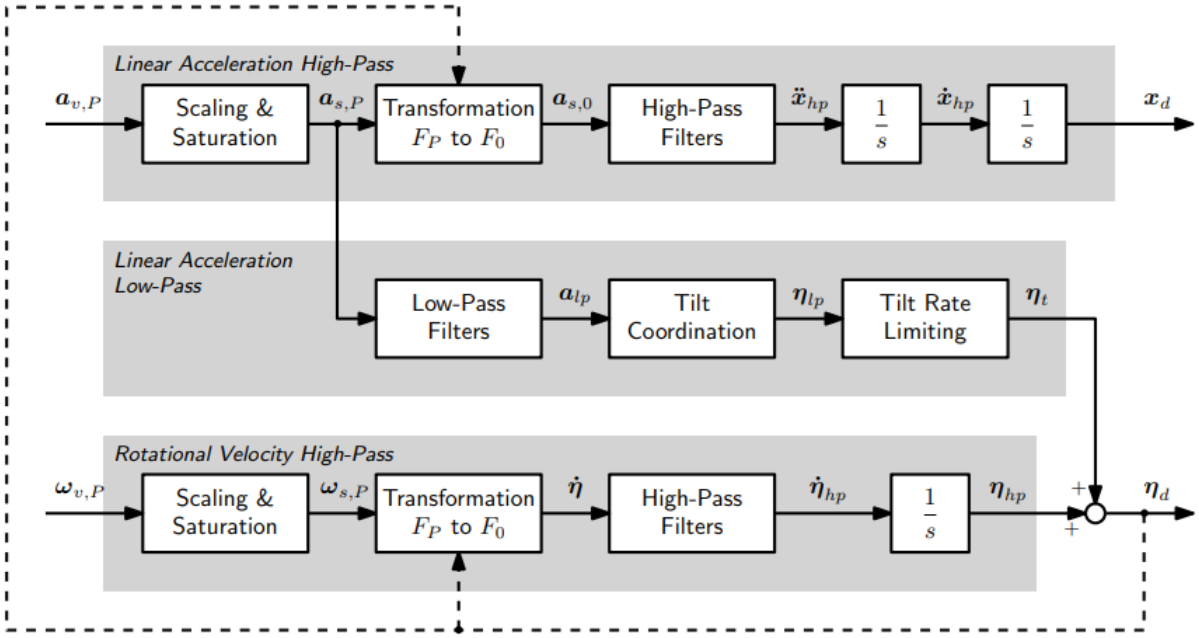


Figure 2.3: A schematic representation of the classical washout filter [27].

A classical washout filter, as the name suggests, filters the desired motion, but this algorithm does not know the boundaries of the physical space of the simulator. So while this is the simplest implementation of an MCA,

the tuning process is not as straightforward as one might think. The low-pass and high-pass filters have to be tuned, which include the order of the filters, the break-off frequency and the gains for each motion channel. Generally, tuning is done through a trial-and-error process in which the motion is tuned down until the worst case scenario stays within the physical limits of the simulator. Basically, tuning is a trade-off between needed physical space and simulator fidelity. Usually an expert is needed to understand the influence of the parameters on the motion of the simulator.

2.3.2. ADAPTIVE WASHOUT FILTER

The second MCA [28] [29] [30] is similar to the classical washout filter, in the sense that the algorithm still has the same two parts, which are high-pass filters and a washout component. But now there is an overlaying cost function that determines the filter gains based on the motion error, motion magnitude and the changing adaptive parameters. The advantage of this MCA is that it is able to optimise the filter gains based on the current state of the motion platform, causing the algorithm to handle the physical space more efficiently. Tuning this type of algorithm is also similar to the classical washout filter, but the parameters of the cost function are more intuitive, because there the trade-off is between limiting the platform movement and reproducing the desired vehicle motion.

2.3.3. OPTIMAL CONTROL-BASED

Adaptive washout filters optimise the parameters of the classical washout filters, but there is another type of MCA that focuses more on the optimisation of the classical washout filter itself, rather than its parameters. Optimal control-based MCAs [31] handle the motion cueing problem as an optimal control problem. The difference between simulator motion and desired motion is seen as an error that should be minimised, or in case a vestibular model is included, the perception error will be minimised. This is done with a so-called cost function, in which weight parameters determine which error is penalised the most, or which parameter of the algorithm is to be minimised the most. A block diagram of this algorithm can be found in Figure 2.4. Despite the complexity of this MCA, the tuning is simpler than the other MCAs discussed so far, because the parameters that can be tuned are much more intuitive than the ones for the washout filters.

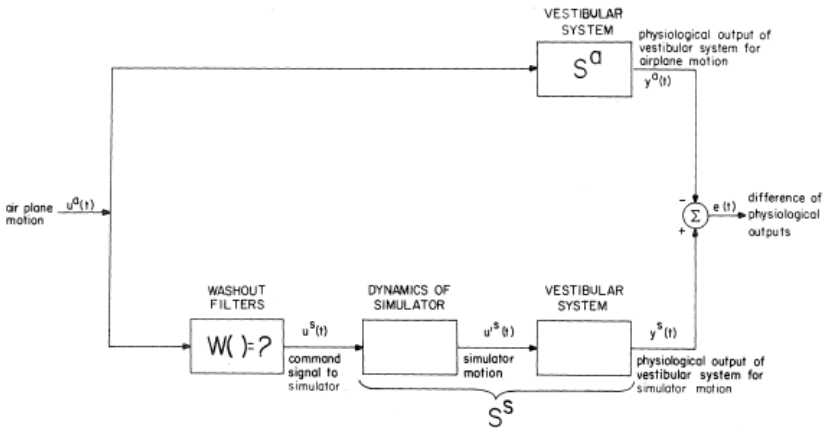


Figure 2.4: Block diagram of an Optimal control-based MCA. [31]

2.3.4. MODEL PREDICTIVE CONTROL

The last and most recently developed MCA, as proposed in [32], [33] and [34], is based on the optimal control-based MCA, but now the optimisation also takes into account the future trajectory of the vehicle motion. Because of such a prediction, the algorithm is able to anticipate future events and prepare for them. The algorithm finds a sequence of future actions to minimise the error between the desired and actual motion over a certain prediction horizon while explicitly accounting for the physical limits of the simulator, meaning that the simulator motion is optimised for a certain time segment, and not only for the current time.

The main advantage of such an MPC-based MCA is that the controller is able to determine the input, based on the current state of the simulator. Whereas a classical washout filter has to be tuned down for the worst case scenario, and thus also all other scenarios are tuned down, MPC-based MCAs can behave differently in each situation, such that it uses the simulator space more effectively. Another benefit of using MPC for motion cueing is that tuning the algorithm will be more intuitive, although at the same time it also becomes more complex due to the interaction between the many different parameters. Instead of cut-off frequencies, damping and gain factors, the user can adjust error weight parameters that are closely related to the motion produced by the simulator. The tuning is similar to the one for optimal control-based MCAs, meaning that there are weight parameters that emphasise which motion errors are to be minimised. On top of that, the prediction horizon can be adjusted, where a larger prediction horizon means that there is more knowledge of the future taken into account, which enables the algorithm to anticipate better when simulator limits will be reached, improving the stability of the system.

Although a larger prediction horizon seems like a good option, one must keep in mind that there are downsides as well to an increased prediction horizon. First of all, while it might result in less motion errors due to having more knowledge of the future, a longer prediction horizon is more susceptible to an inaccurate prediction, and a false prediction is unwanted as this potentially increases false cues or decreases the stability of the system. Next to that, a larger prediction horizon comes at the cost of higher computational costs, because the optimal control problem at hand becomes larger. It is important to realise that the computational costs are limited by the hardware, causing it to be the major limitation when tuning the algorithm.

3

Model Predictive Control-Based Motion Cueing Algorithm

The previous chapter dealt with simulator motion cueing in general, and ended with a short description of a new type of MCA that has some advantages over the filter-based algorithms. There are again many different types of implementations, so in this chapter, first the basic principle of MPC in motion cueing is explained, after which more detailed information is given on the version that is used by the Max Planck Institute for Biological Cybernetics. Finally, some recent results are shown to indicate the potential of algorithm.

3.1. Principle of Model Predictive Control in Motion Cueing

Model predictive control (MPC) is a technique that is used as an identification technique and to control complex systems [35], the latter being of importance here. It is used in many different industries, developed initially to meet specially tailored control needs of systems such as nuclear power plants and oil platforms [35]. These industries needed something to control their systems that optimised the process at each control step, while the system is allowed to change. MPC is a technique that is able to optimise the future behaviour of a system over a specific time interval, the prediction horizon. It does this based on the system input, the current state of the system and the future state of the system (which is based on the current state and inputs), by minimising the squared error between a reference signal of the future and the calculated future signals. In other words, it solves an optimal control problem of a future trajectory, and this optimal control problem is solved at each time step. One of the challenges of this algorithm is that a reference signal of the future has to be established. One could think of that as the desired trajectory of the system, because the algorithm will try to make the system do exactly as that reference signal. For some systems, such as controlling temperature, this is rather straightforward. But in other cases where the future is uncertain, such as MCAs for driving simulators, determining such a reference signal of future prediction is risky, due to the fact that a human is involved in the control loop. More on this will be discussed in Subsection 3.2.2.

While MPC originates from the energy industry, it has been implemented in many other industries, of which the simulation industry is of importance here. In motion cueing algorithms for motion simulators, MPC is used to determine the movements of the motion platform, based on the current state of the platform and the control inputs. The objective function of MPC in MCAs is to minimise the error between the future reference motion cues and the actual simulator motion cues, meaning that the entire prediction horizon is optimised. But only the first time step is used to control the motion platform, because the next time step, the entire prediction horizon is optimised again, creating a new command for the motion platform at every time step. A graphical overview of the principle of MPC, as it is implemented in motion cueing algorithms, is shown in Figure 3.1.

One can also opt to include a model of the vestibular system of the human in the control loop. In that case, the objective function will not minimise the error between reference motion and actual motion, but rather try to mimic the vestibular stimuli of the human according to the model. This allows for more possible ways of providing a sense of motion, since the human vestibular system is not perfect, as has been explained in

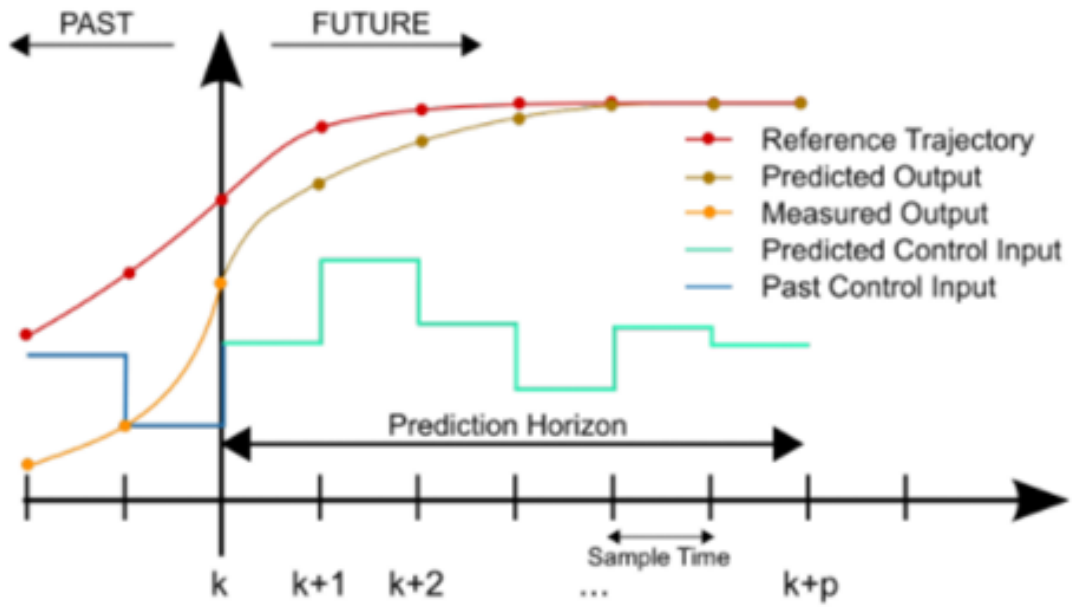


Figure 3.1: A graphical overview of the principle of MPC in motion cueing, from [33].

Section 2.1.

One of the main advantages of an MPC-based MCA over the classical filter-based approaches is that the current state and constraints of the motion platform are explicitly accounted for in the algorithm. Therefore, this algorithm does not have to be tuned in order to stay within the physical limits of the system. Instead, the tuning process is fully focused on improving the performance and stability of the system.

Another advantage is that MPC algorithms are able to use the available physical space of motion simulators more effectively, since an optimal trajectory is calculated based on the current state of the system. Filter-based MCAs are scaled down entirely for the worst-case scenario, causing the process to be sub-optimal for all other scenarios. MPC is more dynamic and therefore, in theory, has the potential to have a higher performance than filter-based approaches.

There are many different implementations of an MPC-based MCA. The ULTIMATE simulator of Renault has a hexapod motion system that is positioned on a combined XY table, meaning that there are eight degrees of freedom, equipped with an MPC-based MCA [36]. At the university of Padova, Italy, MPC is applied to the motion simulators to control the driving simulators. They are working on incorporating a model of the human vestibular system into the MPC algorithm [33] and on deriving a suitable prediction strategy to improve the algorithm [37]. And also at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany, research is being done on implementing and improving an MPC-based MCA for motion simulators. On top of the commonly used hexapod motion simulator, the MPC algorithm is also implemented on different motion platforms, such as the novel Cable-Robot Simulator [6] and on the CyberMotion simulator [38], an eight degrees of freedom serial robot simulator. The MPC-based MCA that will be investigated in this research is the one developed at the Max Planck Institute for Biological Cybernetics for the hexapod simulator. In Section 3.2, this version of the algorithm will be explained in more detail.

3.2. MODEL PREDICTIVE CONTROL-BASED MOTION CUEING ALGORITHM OF THE MPI

The MPI MPC-based MCA is set up different than other reported MPC-based MCAs, such as the one from Padova. Due to high computational costs that are associated with MPC, [37] uses a strategy called 'move-blocking', which reduces the size of the optimal control problem by assuming that several of the input values,

or its derivatives, are constant over several time-steps [39]. But as already stated in [37], the calculated optimal sequence is an approximation because some values are taken from the previously calculated solution. On top of 'move-blocking', they also make use of a strategy based on 'explicit MPC' [40], which basically means that the algorithm tries to determine the physical limits by looking up the limits in a pre-calculated table, which is based on the current state of the motion system. These strategies enable the MPC-based MCA to run real-time.

The MPI has been able to make the MPC-based MCA run real-time as well, while also incorporating all actuators inside the model. This increases the size of the optimal control problem, but several computational tricks and simplifications have been applied in order for the algorithm to run real-time. One of them is that instead of converging to the optimal solution of the optimal control problem each time-step, a maximum number of iterations per time-step can be set, meaning that a sub-optimal solution is taken in order to be able to continue to the next time step at a sufficiently high control frequency. However, doing this consecutively causes the algorithm to converge to the optimal solution anyway over multiple time-steps [41].

There are two main parts of the MPC-based MCA of the MPI that are worth a detailed explanation. First, it has been stated that the objective of an MPC-based MCA is to minimise the error between a reference signal and the predicted future signals of the system. It does this by minimising a cost function. There are different ways of making such a function, but in this report, only the one from the MPI will be explained in Subsection 3.2.1. Second, the reference signals in the algorithm are interesting, since we are dealing with driving simulation in which a human might be involved in the control loop. The different possibilities are described in Subsection 3.2.2.

3.2.1. PARAMETERS OF THE COST FUNCTION OF THE ALGORITHM

The cost function of the MPC-controller minimises the squared error between reference values and actual values of the output signals (y_k), state signals (x_k), input signals (u_k) and the terminal state (x_n), over a certain prediction horizon (N) [6]. The mathematical form is shown below.

$$\text{Minimise: } \frac{1}{N}\int_{k=0}^{N-1} l_k(x_k,u_k) + l_n(x_n) \tag{3.1}$$

Where

$$l_k(x_k,u_k) = ||y(x_k,u_k) - \hat{y_k}||^2 W_y + ||x_k - \hat{x_k}||^2 W_x + ||u_k - \hat{u_k}||^2 W_u \tag{3.2}$$

And

$$l_n(x_n) = ||x_n - \hat{x_n}||^2 W_{x_n} \tag{3.3}$$

The MPI version of an MPC-based MCA has many parameters that have an influence on the behaviour of the algorithm, as can be seen in the functions above, where W_y, W_u, W_x and W_{x_n} are all vectors of error weight parameters. This means that if the value of these parameters is increased, there will be a higher penalty on the squared error between the desired output and the output of the MPC-controller for that specific signal. There are nine output error weights (W_y, three translational accelerations, three angular velocities and three angular accelerations), twelve state error weights (W_x, three positions, three angles, three translational velocities and three angular velocities), twelve terminal state error weights (W_{x_n}, again three positions, three angles, three translational velocities and three angular velocities) and six input error weights (W_u, three translational accelerations and three angular accelerations). Terminal state parameters are basically the same as the normal state parameters, but terminal state parameters are the last state values in the prediction horizon. This can be used to ensure that the future prediction always ends up at a certain point, which can increase the stability of the algorithm. As already explained before, MPC needs a reference trajectory against which the error should be minimised, so for all these signals, there needs to be a reference signal as well. Most of the reference signals are straightforward and logically chosen. First of all, the state reference is the values towards which the motion platform will washout, so this is chosen to be its neutral position. Since there is no need yet for input filtering or increasing stability by setting a value for the terminal state parameters, these reference signals are of little importance and thus set to zero. Finally, the output reference signal is where things get interesting, due to their dependence on a prediction of what the output will be in the future, meaning that these will not be constant. This will be discussed in Subsection 3.2.2.

This MPC algorithm has been set up in such a way that all physical limits can easily be changed. These constraints include limits on the states, terminal states and inputs. The variable constraints allow this algorithm to be generic and thus easily integrated on a different simulator. Adding these parameters results in the total list of parameters, as is summarised in Table 3.1.

Symbol	Name	Number of parameters
N	Prediction horizon	1
Δt	Prediction time step	1
W_y	Output error weights	9
W_x	State error weights	12
W_u	Input error weights	6
W_{x_n}	Terminal state error weights	12
$\hat{y_k}$	Output reference	9
$\hat{x_k}$	State reference	12
$\hat{u_k}$	Input reference	6
$\hat{x_n}$	Terminal state reference	12
-	State bounds	12
-	Terminal state bounds	12
-	Input bounds	6

Table 3.1: Summary of the parameters of the cost function of the MPC-controller and the variable constraints.

As can be seen, there are a lot of parameters that influence the behaviour of the MPC-controller, some are straightforward, others may not directly indicate how they affect the performance. Note that the following description is based on having the reference signals for the input, state and terminal state all zero, as that is the case that will be used in this research. First of all, the prediction horizon is not as straightforward as one might think. It can be expected that a longer prediction horizon increases the performance, but the risk of an inaccurate prediction increases with longer prediction horizons and a longer prediction horizon increases computational costs. More on this will be discussed in Subsection 3.2.2. Second, the error weights put a penalty on the error of that specific signal. Higher output error weights should increase tracking performance, higher state error weights increase the washout, high input error weights penalise high input values and high terminal state error weights are there to improve the stability of the algorithm. As a standard practice, the values for the output error weights are chosen to reflect the variance of specific force in m/s^2 and angular velocity in rad/s^{-1} for typical vehicle manoeuvres [6]. The input error weights are set to a very small value, because in general, there is no need for input filtering. Should one perform a similar analysis on a different simulator with more physical possibilities, this set of parameters could be of interest to reduce extreme behaviour. The terminal state error weights are set to a very small value as well, because this is only necessary to tune in the case of unstable behaviour. Then finally, for the state error weights an optimisation has been performed. The manoeuvre that will be used to analyse the influence of some of the parameters is a car driving through a corner, and the state error weights have been optimised such that in the standard configuration that has just been explained, the simulator platform moves back to its neutral position within a reasonable time span. This has been done by designing a cost function that tries to minimise the total error of all vestibular signals during the curve, while putting a very high penalty on the motion at some time after the curve. A summary of the parameters as they will be used initially is shown in Table 3.2.

Investigating all these parameters simultaneously will become too broad and time-consuming, therefore only a selection will be analysed in detail. Terminal state error weight parameters (W_{x_n}) have an effect on the motion, but are there to generate simulator stability, so there is no need in tuning them after there is stability. Input error weight parameters (W_u) also make sure that the simulator has stability, for example in the case of the CyberMotion simulator, no input filtering causes the simulator to perform extreme motions (without necessarily making it better). But since this is not an issue on hexapod simulators, these are of no interest in this research. State error weight parameters (W_x) cause the simulator to washout. Although they are important, all they do is determine to which extent the motion platform will perform washout behaviour, which is not needed in case of a hexapod motion system. Therefore the output error weight parameters (W_y) are the ones that will be investigated in this research, since those most directly influence the tracking performance of the MCA. And from these, only the three translational acceleration and three angular velocity error weight

Parameters	Values
W_y	[1 1 1 10 10 10 0 0 0]
W_x	[8.1805 5.1302 3.5618 5.3556 3.6541 6.8171 0 0 0 0 0 0
W_u	[0.01 0.01 0.01 0.01 0.01 0.01]
W_{x_n}	[0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01]
$\hat{y}_k$	Dependent on prediction strategy
$\hat{x}_k$	[0 0 0 0 0 0 0 0 0 0 0 0 0]
$\hat{u}_k$	[0 0 0 0 0 0]
$\hat{x}_n$	[0 0 0 0 0 0 0 0 0 0 0 0 0]

Table 3.2: Summary of the parameter values of the cost function of the MPC-controller as will be used initially in this research.

parameters are selected, since these reflect the human vestibular system.

3.2.2. PREDICTION STRATEGIES

A very important but tricky aspect of the MPC-controller is the prediction, which creates the future output reference signal. If there is no human involved in the control loop, or any other uncertain factor, it is straightforward as the future is known. But it is not in case of uncertainties in the control loop or online motion cueing, which is often the case as many simulations require a human subject to steer a vehicle, and in that case the future is uncertain. There are multiple options if that is the situation. The easiest and most straightforward to implement is to assume that there will be no change in the near future and keep the predicted output equal to the current output. The downside of this method is that longer prediction horizons cause the algorithm to do unexpected things. If for example the current output is an acceleration of 1 m/s^2 to the left and the prediction horizon is 10 seconds, the algorithm will prepare for such a prediction by moving to the right, such that a larger movement to the left can be made. But because the prediction is constant, the algorithm will keep doing this until the motion changes. The result is that an acceleration to the left is simulated by providing an acceleration to the right, which is undesirable. Another, similar option is to use the current output values and scale them accordingly, however it is difficult to actually implement that, as the value could go up or down, so keeping it constant is a safer method. A completely different approach would be to have an actual prediction of what is going to happen, based on a virtual driver or a recording of a previously made simulation. Then comparing the current position of the simulated vehicle to the one from the recording enables the algorithm to know what is going to happen. But for this method to work, the behaviour of the recorded driving and the actual driver should be very similar, because else the prediction is false. This method is only suitable for experienced drivers [37], such as racing drivers. Therefore, for this research a constant prediction will be used.

The length of the prediction is the result of two parameters, the amount of steps and the time in between steps. Small time steps are only useful when the behaviour of the system would be able to change significantly within that time. And the total prediction time is only beneficial if prediction is accurate, and even then at a certain point the gain in performance is not worth the increased computational costs. And for uncertain predictions, it can even make the performance worse. At the time of performing the analysis, the maximum amount of steps can could be handled in real-time was a little below 100, therefore, to ensure that this version of an MPC-based MCA can run real-time on other installations, it has been decided to limit the amount of steps to 20 for this research. In this way, the results will be useful to the majority of users of this algorithm in their first stage of implementation. In order to still have some meaningful prediction time, a time step of 0.1 seconds it taken, such that the prediction time is 2.0 seconds. For driving a vehicle, not a lot will change in 0.1 seconds, so this is a safe value to take.

When this research was already nearly finished, a new option had been implemented to introduce a nonlinear spacing of the prediction horizon. This means that it is possible to have a small time step in the near future, medium time steps after that and finally large time steps in the far future, such that more emphasis can be put on the near future. However, this will not be considered in this research.

4

METRICS

Assessing simulator motion cueing quality is a difficult process. The ultimate goal of motion simulators is to give the subject the feeling of actually being inside the vehicle, and to have the subject behave the same as if he or she is inside the real vehicle. unfortunately, it is not always possible to compare the behaviour of subjects inside a simulator and inside a real vehicle due to various reasons. Therefore, often use is made of multiple measures that are somehow related to behavioural and perceived fidelity. In this chapter, some frequently used objective metrics are described and a method to measure perceived motion quality is explained. The objective metrics will be used in the initial analysis in Chapter 5, because it is a cheap and fast way to analyse a lot of different settings of the MPC-based MCA. After this initial analysis, promising and interesting settings will be analysed further in an experiment, where the perceived motion quality will be measured.

4.1. OBJECTIVE METRICS

Many studies have been performed that focused on determining objective metrics related to human perception. We know that the human vestibular system is not perfect, which is exploited by motion simulators. We also know how the human vestibular system determines motion, namely it senses specific forces using the otoliths and it senses angular velocity using the semi circular canals. But there is still a lot of uncertainty about how these signals are processed in the brain, therefore it is difficult to determine the effect of certain signals on the perceived fidelity of a simulation. One way to find out if certain objective metrics are an indication of perceptual fidelity is to actually measure the perceptual fidelity by using human subjects and comparing them to objectively measured metrics. Such a study found that the Pearson Correlation Coefficient between the desired signals and the actual simulator signals is correlated with the perception of humans [42]. Next to that, the Root Mean Square Error is a frequently used metric when comparing signals. Both of these metrics will be explained below.

4.1.1. ROOT MEAN SQUARE ERROR

In motion simulators, the goal is to achieve realistic driving behaviour of subjects. It has been shown that humans are not able to differentiate between two motion cues if the differences are small [23]. In addition, there is an absolute threshold under which humans are not able to determine if they are moving of not [43]. Thus it can be argued that small errors do not matter much in motion simulation, but large errors are to be avoided. Therefore, the Root Mean Square Error (RMSE) between the desired and actual motion is a good objective measure to define the motion quality of a motion simulator. The following equation shows the calculation for the RMSE for each motion channel individually, where y is the actual simulator output, $\bar{y}$ is the desired simulator output and N is the number of data points.

$$RMSE = \sqrt{\frac{(y - \hat{y})^2}{N}} \tag{4.1}$$

Figure 4.1 shows the simulator motion when using a classical washout filter. Using the above equation, we can calculate the RMSE for each motion channel individually, resulting in the values in Table 4.1.

ax	ay	az	p	q	r
0.1700	0.3610	0.0394	0.0220	0.0156	0.0706

Table 4.1: RMSE vallues for each motion channel when using a classical washout filter.

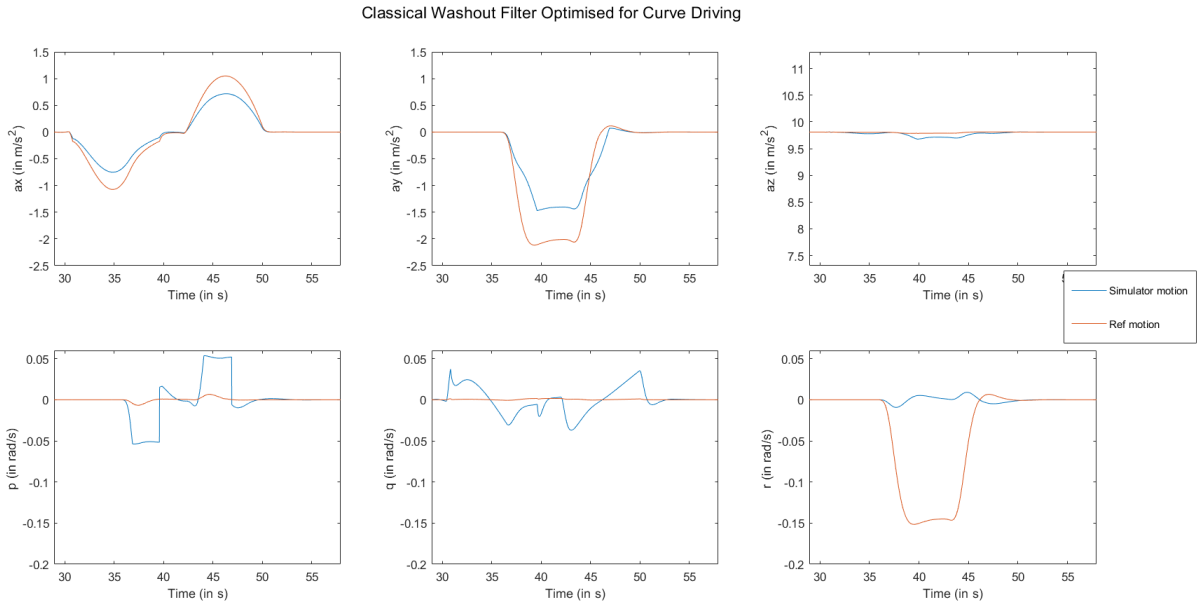


Figure 4.1: Simulator motion of the curve manoeuvre for a classical washout filter.

4.1.2. PEARSON CORRELATION COEFFICIENT

The RMSE is a useful metric that penalises large errors, but errors that only consists of a magnitude error are not always bad. Moreover, multiple experiments suggest that motion scale factors below 1 are preferred [11]. Therefore, a second objective metric that does not penalise motion scale factors, but rather looks at the resemblance between the two signals, is proposed, called the Pearson Correlation Coefficient (PCC). It is a coefficient that gets closer to 1 when two signals are similar in shape, even if the magnitude differs. The following equation is used to calculate the PCC for each motion channel individually, where μ is the mean of the data set, σ is the standard deviation of data and N is the number of data points.

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{A_i-\mu_A}{\sigma_A}\right)\left(\frac{B_i-\mu_B}{\sigma_B}\right) \tag{4.2}$$

Using the above equation, we can calculate the PCC for each motion channel individually, resulting in the values in Table 4.2.

ax	ay	az	p	q	r
0.9999	0.9718	0.8888	0.8291	0.2167	-0.3232

Table 4.2: PCC vallues for each motion channel when using a classical washout filter.

4.2. PERCEIVED MOTION QUALITY

Objective metrics are useful when many different simulations have to be analysed, because it is a faster and cheaper option than using human subjects. This means that they are ideal for initial analyses, but ultimately, when determining the perceived motion quality, a subjective experiment will have to be performed. Many methods exist to assess the perceived motion quality of a simulation, such as asking human subjects to rate certain manoeuvres based on a rating scale, for example on a scale of 1 to 10. Others have more detailed questionnaires that ask participants to rate certain parts of a manoeuvre by drawing a line, where the length

of the line indicates the realism of those parts [3]. But what these methods are missing is the ability to detect time-varying ratings and the effect of these time-varying ratings on the overall perception of the simulation. Therefore, this research will make use of a continuous subjective rating method that asks subjects to rate the motion mismatch between what they would expect, based on what they see on their visuals, and what they feel [12]. This is done by turning a knob, based on the perceived mismatch experienced by participants. If they perceive a large mismatch, they can turn the knob to the right, causing the rating bar to become increasingly more filled and more red. If the perceived mismatch becomes smaller, they can turn back to the left and the bar becomes more empty and green. Figure 4.2 show what participants will see during the experiment, with the turning knob located at the right hand side and the rating bar located at the bottom of the screen in front of them. Figure 4.3 shows the turning knob in more detail. This method has been used and validated in other studies as well, where subjects were able to give a rating of the motion mismatch using a turning knob [7] [44] [10], making it a sound decision to use this method here as well.

Figure 4.2: The view as is seen through the eyes of participants, with the turning knob located at the bottom right.

Figure 4.3: The turning knob with which participants can indicate the perceived motion mismatch. The rating bar on the background shows the current rating.

5

Sensitivity Analysis of the Output Error Weight Parameters

This chapter will present an initial analysis on the sensitivity of the error weight parameters of the cost function of the MPC-based MCA of the MPI. Here, only the objective metrics will be measured to be able to analyse all possible combinations of parameters within a reasonable time-span. First, the control, dependent and independent variables will be described to establish the analysis setting. Then the sensitivity of the MPC-based MCA will be determined in the following two phases, because it is unknown whether the parameters influence each others sensitivity.

- First, all parameters will be changed one by one, therefore only looking at their individual sensitivity. This will be done by changing the parameters slightly and running the simulation again, such that we are able to determine the RMSE and PCC for every parameter value. This will result in a RMSE/PCC versus parameter value plot for each parameter.
- After the individual sensitivity has been established, all parameter combinations will be tested against each other to analyse the effects of changing two parameters simultaneously.

From these two analyses followed some interesting, unwanted behaviour, which led to the determination of sensible ranges for all parameters. From all these results, a selection will be made that will be further investigated in an experiment, of which the proposal is described in Chapter 6. Note that when there are no units displayed, the translational accelerations are in m/s^2 and the rotational rates are in rad/s.

5.1. Control, Dependent and Independent variables

As has already been explained, analysing all possible parameter combinations is not feasible within the allocated time for this project, which means that the number of parameters that will be investigated has been narrowed down to just the six output error weights: three translational accelerations (W_{ax}, W_{ay} and W_{az}) and three angular velocities (W_p, W_q and W_r). The values for the standard setting can be found in Table 5.1. The state error weight, terminal state error weight and input error weight parameters are control variables, they will remain constant throughout the entire analysis. The manoeuvre consists of a car first driving straight at 70 km/h, slowing down to 50 km/h, doing a curve of 60° with a radius of curvature of approximately 90m and then accelerating back to 70 km/h. As has been discussed in Subsection 3.2.2, a constant prediction of 20 steps of 0.1 seconds will be used, because this is a straightforward implementation and thus easily accessible for new users of this algorithm. A model of the MPI hexapod simulator will be used, since this is a commonly available simulator. And finally, the dependent variables consist of the metrics that are described in Chapter 4.

5.2. Output Error Weight Parameter Sensitivity

Determining the sensitivity of the MPC-based MCA due to changing the output error weight parameters will be done by looking at how changing these parameters affect the metrics described in Chapter 4, the RMSE and

Table 5.1: Summary of the parameter values of the cost function of the MPC-controller as will be used as standard values.

Parameters	Values
W_y	[1 1 1 10 10 10 0 0 0]
W_x	[8.1805 5.1302 3.5618 5.3556 3.6541 6.8171 0 0 0 0 0 0
W_u	[0.01 0.01 0.01 0.01 0.01 0.01]
W_{x_n}	[0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01]

PCC. First of all, a classical washout filter will be tuned such that the results of this analysis can be compared to something we already know. Then the effect of changing one parameter at a time will first be analysed, after which the effect of changing two parameter simultaneously will be examined.

5.2.1. COMPARATIVE CLASSICAL WASHOUT FILTER

In order to understand the performance of the different settings of the MPC-controller, a comparison will be made with a currently available motion cueing strategy, which is the classical washout algorithm as described in [27] and [45]. As had already been explained in Chapter 2, this algorithm also needs to be tuned. A number of parameter sets have been tested, from which the best performing set will be chosen, such that it can be argued that the benchmark is of relatively high quality. With this, it will be possible to indicate if the MPC-based MCA performs better/worse than a good CWF MCA, given certain parameters.

One of the parameter sets has been optimised for curve driving in car simulation, which has also been used in [44], while the other three parameters sets are documented in published papers [3, 17, 46]. The comparison of the filter responses for all specific forces and rotational rates can be found in Figure 5.1. We are dealing with curve driving simulation, so the parameter set that replicates the lateral specific force best while not introducing large perceptible false cues will be used as baseline for the analysis. In this case, that is the parameter set that has been optimised for curve driving manoeuvres [44], which will simply be called the classical washout filter for the remainder of this report.

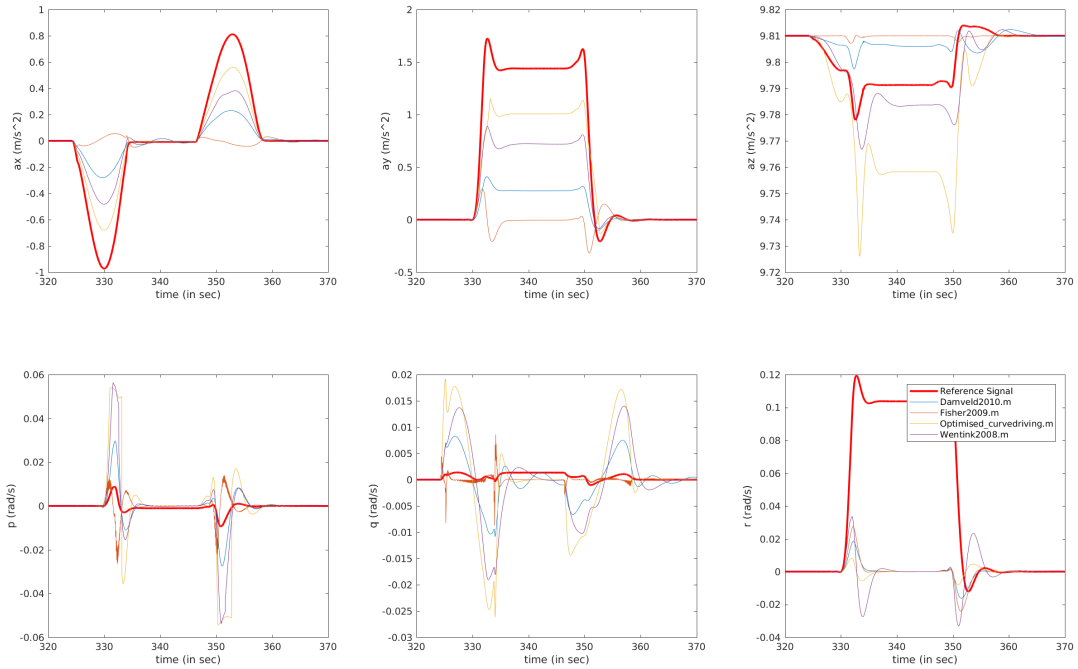


Figure 5.1: A comparison of multiple classical washout filters.

5.2.2. MPC SENSITIVITY PLOTS

The sensitivity of the MPC-algorithm due to changing the output error weight parameters will be determined by calculating the change in RMSE/PCC with respect to the baseline condition (Equation 5.1), while constantly changing one parameter slightly at a time. For each simulation, the RMSE and PCC will be calculated for each motion channel (three translational accelerations and three angular velocities). The outcome will be plotted against the parameter value, resulting in a plot where one can see what happens to the RMSE or PCC when the parameters change. The translational accelerations and angular velocities are plotted separately due to the difference in parameter values. All the plots can be found below. Under each figure, an explanation will be given of what can be seen in the figures.

$$\text{Relative change} = \frac{\text{RMSE/PCC at max parameter value} - \text{Baseline RMSE/PCC}}{\text{Baseline RMSE/PCC}} \tag{5.1}$$

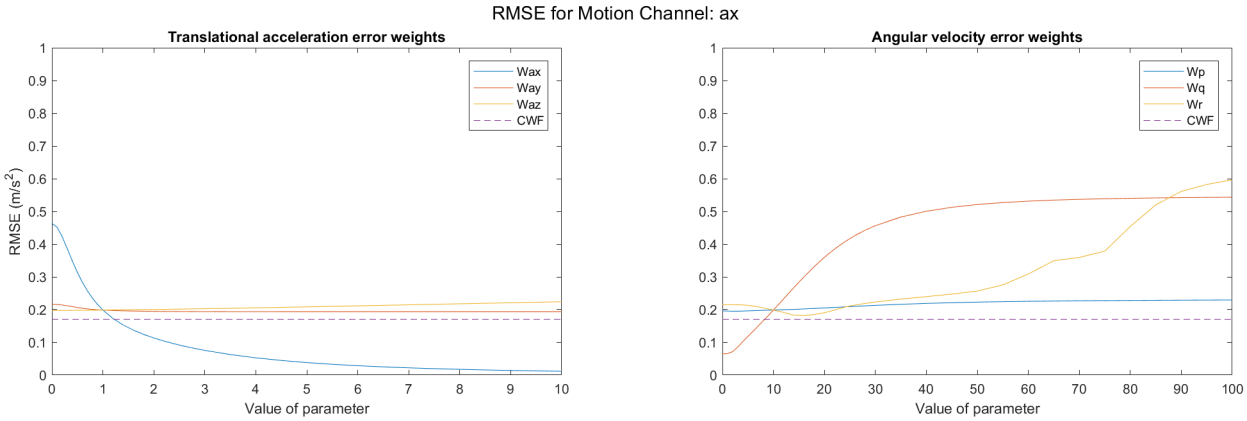


Figure 5.2: The RMSE between desired and actual ax, plotted against the parameter values of the output error weight parameters.

What we can see in Figure 5.2 is that when W_{ax} is increased, the RMSE for motion channel ax is decreased, as one would expect. Even a slight increase already makes have a lower RMSE than the CWF. Changing the other two translational error weight parameters does not result in a significantly different RMSE. For the angular velocity error weight parameters, we see that increasing W_q also increases the error of the x-acceleration, because a higher W_q decreases the allowance of having a pitch rate error, thus decreasing the possibility of using tilt-coordination. This is an effect that we will be seeing more often, because due to the small physical space, simulation accelerations will be a trade-off between performing tilt-coordination and decreasing angular velocity errors. Lastly, W_r also increases the error on motion channel ax. This effect requires a look at what the simulator is doing to understand what is happening. The motion data of the simulator when setting W_r to 85 can be found in Figure 5.4. When trying to track the yaw rate signal too much, other signals start oscillating. This is because when the simulator has reached its maximum yaw angle, it can still generate a little yaw rate by pitching and rolling. Figure 5.3 shows the PCC of the x-acceleration due to changing all six parameters. W_{ax}, W_{ay} and W_{az} do not change the correlation of ax, but W_q and W_r have an effect, although these effects are the same as mentioned before.

Table 5.2 shows the relative changes in RMSE and PCC for motion channel ax, when changing the parameters. It can be observed that Increasing W_{ax} has the most influence on reducing the RMSE for motion channel ax, which is as expected, since W_{ax} should reduce the error on motion channel ax. Just as W_q increases the error, because it restricts the use of tilt-coordination. The change in PCC due to changing W_q is very high, but this is due to the fact that the PCC in the baseline condition is near zero, so a small change will result in a large relative change.

Table 5.2: The relative changes in RMSE and PCC for motion channel ax when changing the output error weight parameters.

Parameter	W_{ax}	W_{ay}	W_{az}	W_p	W_q	W_r
Relative change in RMSE	-0.9412	-0.02601	0.1279	0.1552	1.735	2.003
Relative change in PCC	0.0585	0.006546	-0.005379	-0.0271	-0.3402	-0.4869

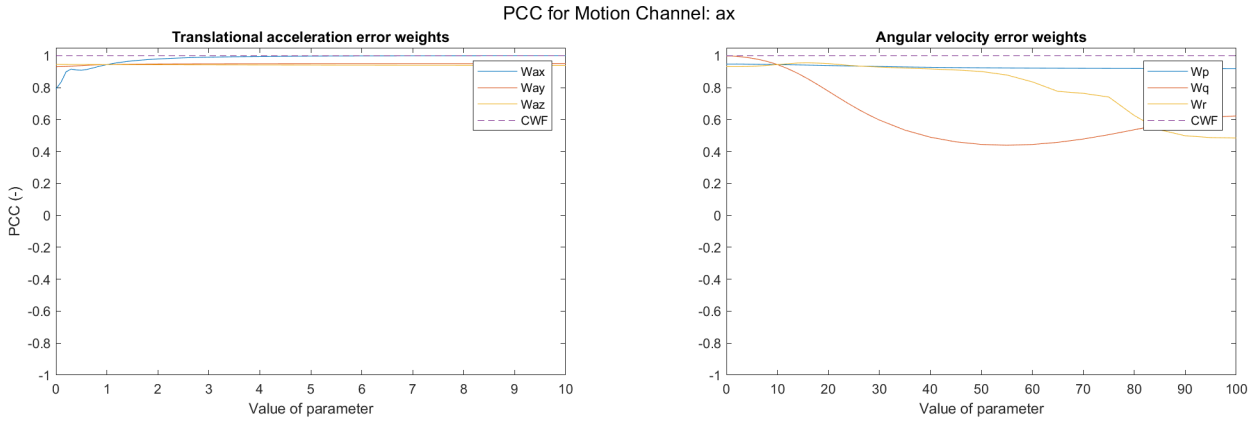


Figure 5.3: The PCC between desired and actual ax, plotted against the parameter values of the output error weight parameters.

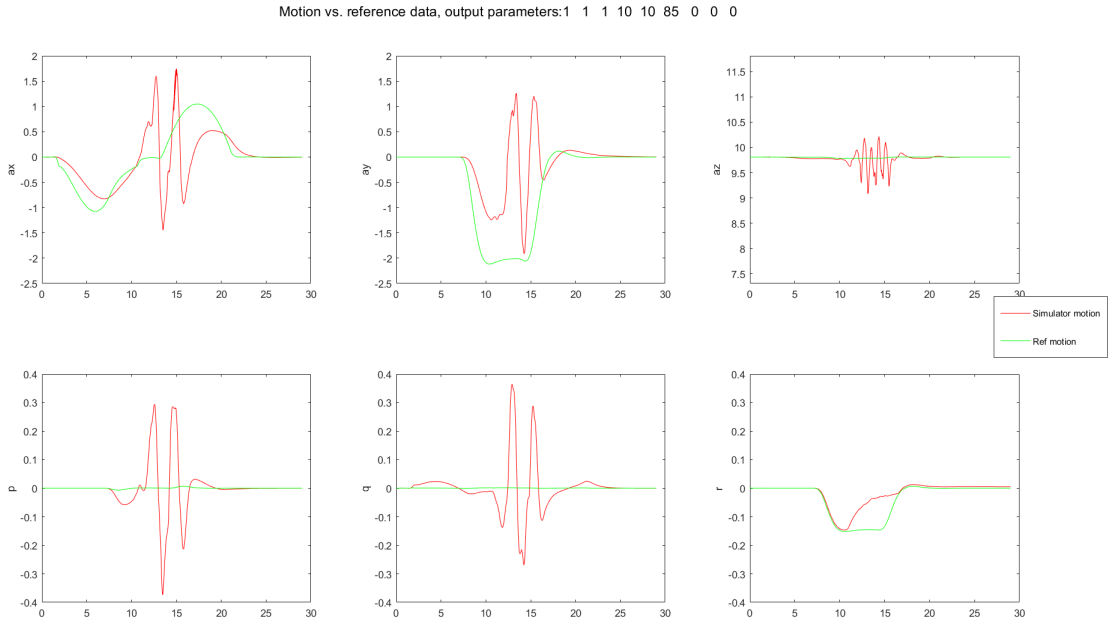


Figure 5.4: The motion data when setting W_r to 85.

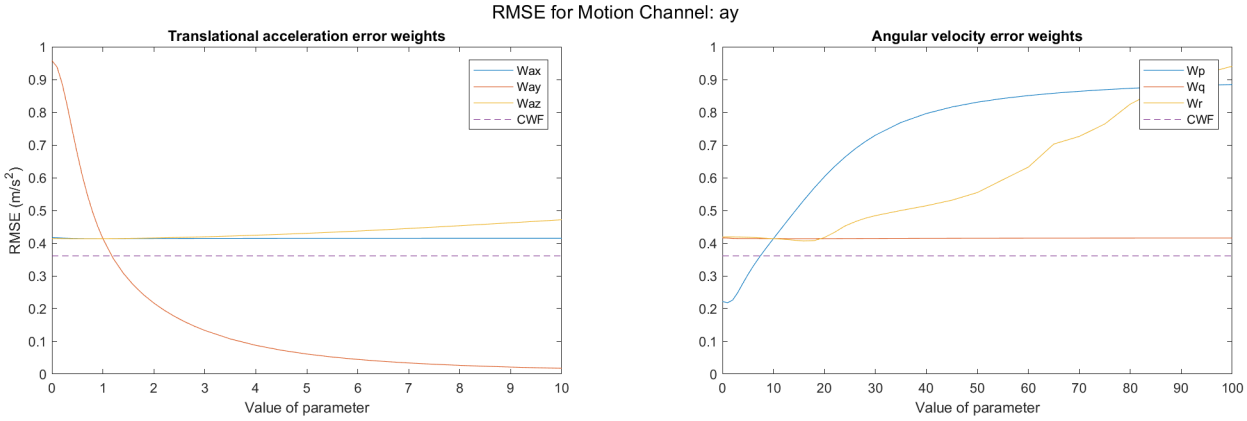


Figure 5.5: The RMSE between desired and actual ay, plotted against the parameter values of the output error weight parameters.

Figures 5.5 and 5.6 show the metrics for the y-acceleration. Similar to ax, here also a trade-off can be seen between tracking the y-acceleration or decreasing the roll rate error. Note that the effect is larger than for ay, but that is due to the manoeuvre used, which has more lateral than longitudinal accelerations. Again, even a slight increase in W_{ay} makes the RMSE lower than for a CWF. And also here, increasing Wr introduces larger errors that are explained by the oscillating behaviour as is seen in Figure 5.4. Another interesting observation is that the RMSE decreases exponentially when increasing Way, meaning that after a certain point, the error will not decrease much anymore.

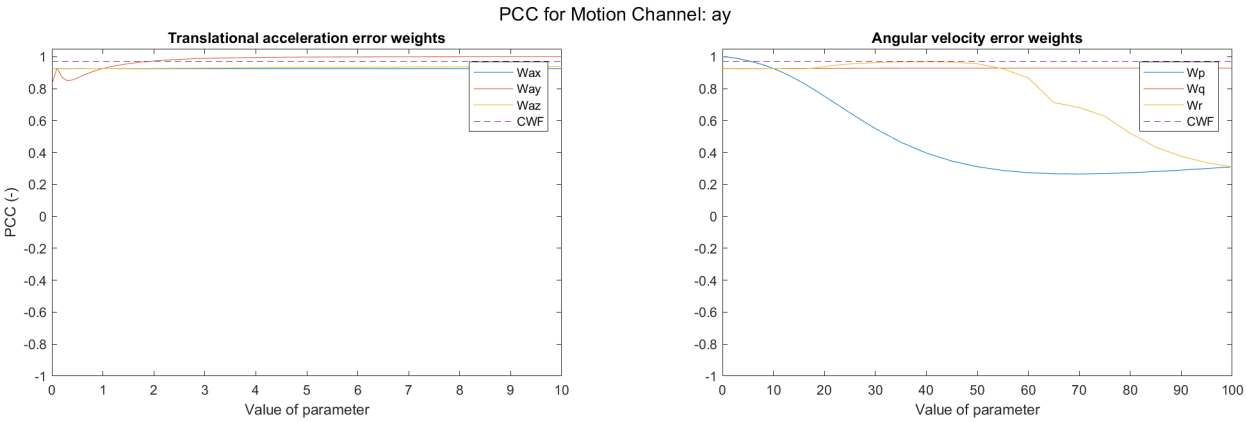


Figure 5.6: The PCC between desired and actual ay, plotted against the parameter values of the output error weight parameters.

Table 5.3 shows the change in metrics for ay. The effect of changing W_{ay} and W_p on motion channel ay is comparable to the effect of changing W_{ax} and W_q on motion channel ax, because in both cases the rotational rates are used to produce a feeling of sustained acceleration via tilt-coordination. Only increasing W_{ay} can result in almost perfect tracking of ay, since setting the parameter to 10 already reduces the RMSE by 95 %.

Table 5.3: The relative changes in RMSE and PCC for motion channel ay when changing the output error weight parameters.

Parameter	W_{ax}	W_{ay}	W_{az}	W_p	W_q	W_r
Relative change in RMSE	0.002845	-0.9574	0.139	1.138	0.004357	1.272
Relative change in PCC	-0.0006491	0.0793	0.01304	-0.6652	0.002604	-0.6663

The z-acceleration does not show a lot of sensitive behaviour, other than losing correlation when parameters like Wp and Wr are increased or when Waz itself is very small, as seen in Figure 5.8. Also here the oscillating behaviour is seen in Figure 5.7, although smaller than for the x-acceleration and y-acceleration.

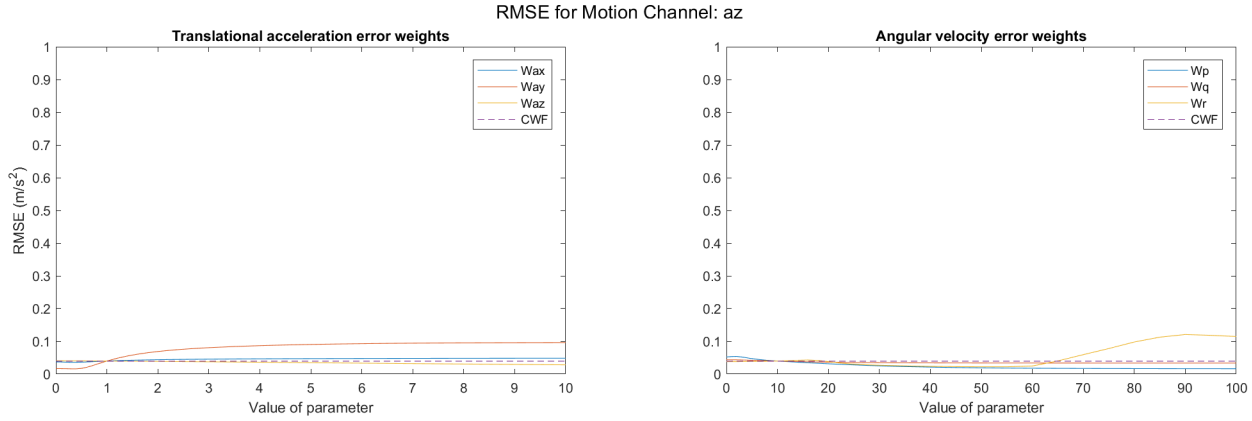


Figure 5.7: The RMSE between desired and actual az, plotted against the parameter values of the output error weight parameters.

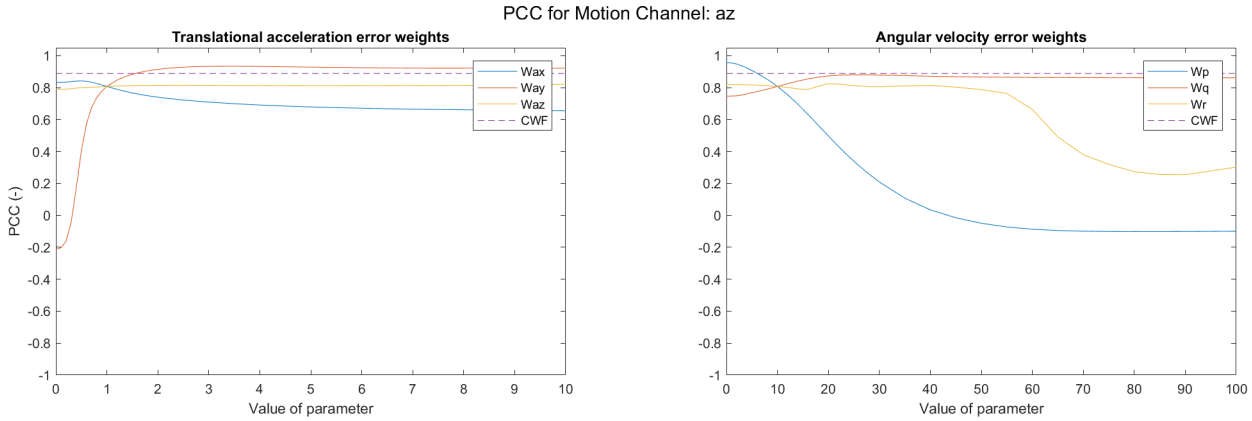


Figure 5.8: The PCC between desired and actual az, plotted against the parameter values of the output error weight parameters.

Table 5.4 shows the sensitivity of az, and it can be seen that also here, W_{az} has the most influence on the corresponding motion channel, az, although the change in RMSE is lower compared to changes for ax and ay. This is likely caused by the fact that az is not dominantly present in driving simulation, but we can still conclude that W_{az} has a lower impact on the performance of the MPC-algorithm than W_{ax} and W_{ay}.

Table 5.4: The relative changes in RMSE and PCC for motion channel az when changing the output error weight parameters.

Parameter	W_{ax}	W_{ay}	W_{az}	W_p	W_q	W_r
Relative change in RMSE	0.2197	1.419	-0.2912	-0.591	-0.1566	1.882
Relative change in PCC	-0.1877	0.1412	0.016	-1.123	0.06694	-0.6256

The RMSE for the roll rate error (Figure 5.9) barely shows any sensitivity, apart from the oscillating behaviour of increasing Wr. This is due to the fact that there is no large desired roll rate, so the error is only large when roll rate is used to simulate other stimuli, as is done in tilt-coordination. This effect can be identified by the small increase in RMSE when increasing Way. The PCC for the roll rate (Figure 5.10) does show interesting behaviour, because increasing Wp first reduces the PCC, before it increases again. This is because by increasing Wp, the algorithm tries to reduce the roll rate error, but in doing so it introduces a very small false cue. Therefore the error decreases slightly but also the PCC decreases. This is solved when Wp is increased further.

In Table 5.5, the changes in RMSE and PCC can be seen for motion channel p. An interesting observation is that the changes in RMSE for motion channel p due to changes in W_{ay} and W_p are approximately of equal magnitude as the changes in RMSE for motion channel ay, but now W_p reduces the RMSE and W_{ay} increases the RMSE. Again, increasing W_p reduces the RMSE of the corresponding motion channel, p, by almost 95 %, indicating a high influence.

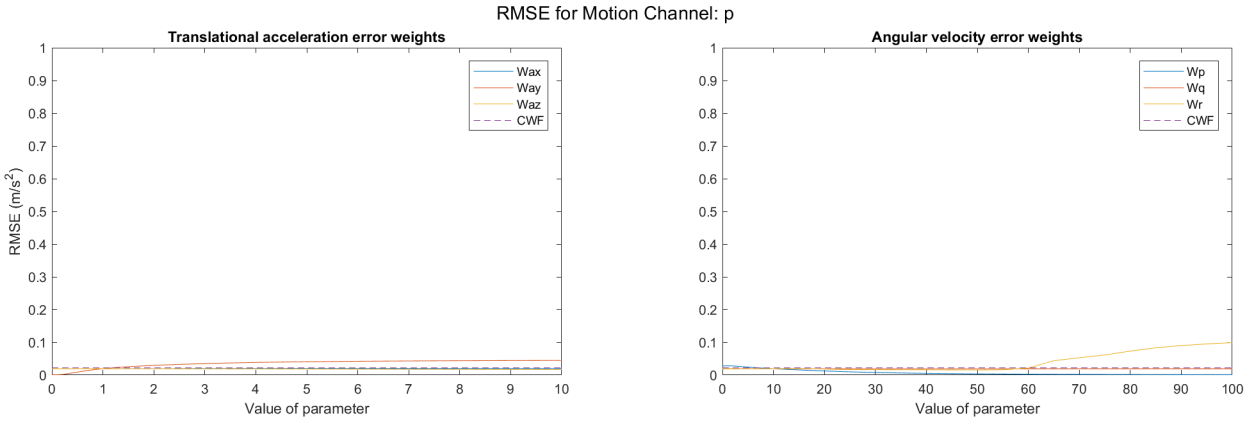


Figure 5.9: The RMSE between desired and actual p, plotted against the parameter values of the output error weight parameters.

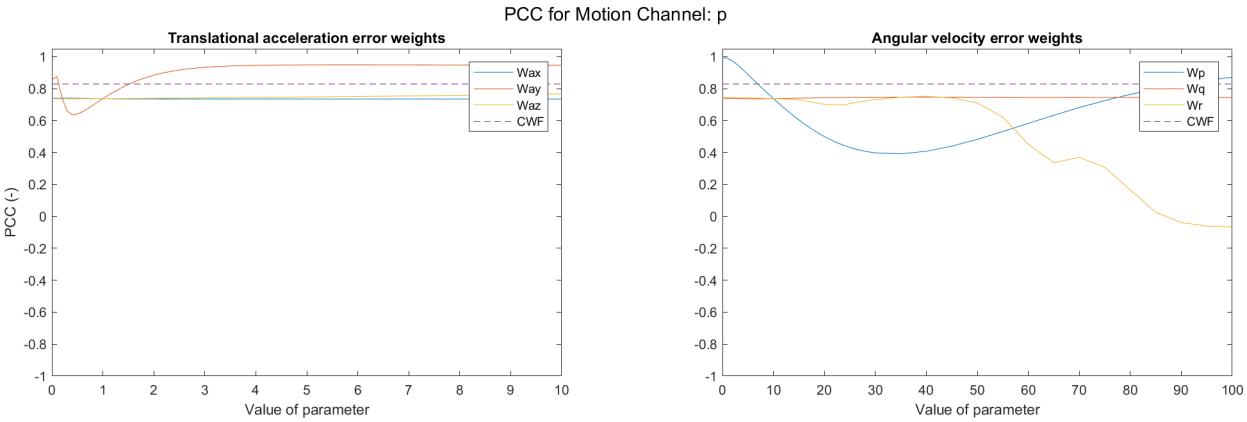


Figure 5.10: The PCC between desired and actual p, plotted against the parameter values of the output error weight parameters.

Table 5.5: The relative changes in RMSE and PCC for motion channel p when changing the output error weight parameters.

Parameter	W_{ax}	W_{ay}	W_{az}	W_p	W_q	W_r
Relative change in RMSE	0.01833	1.293	-0.1462	-0.9438	-0.01044	4.091
Relative change in PCC	-0.002945	0.2845	0.04056	0.1823	0.01011	-1.09

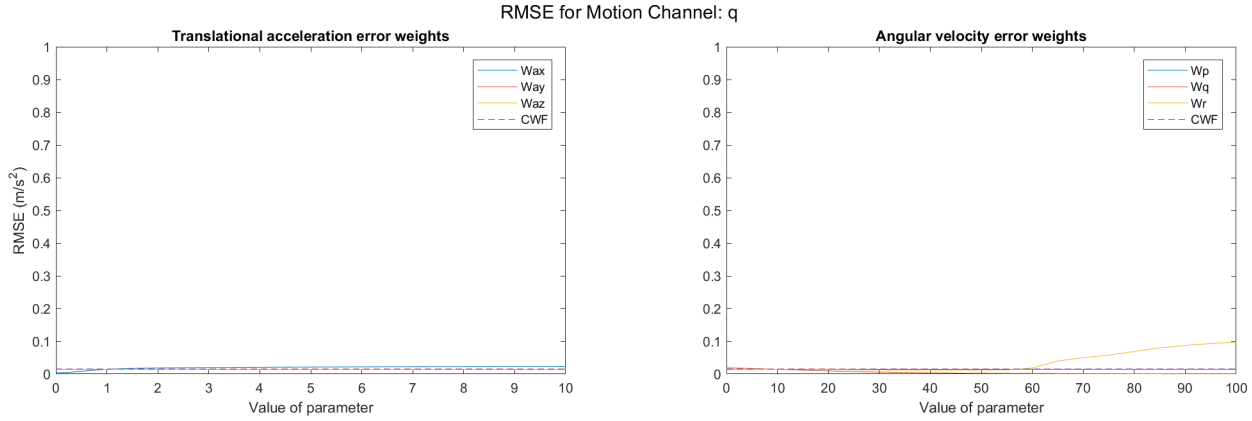


Figure 5.11: The RMSE between desired and actual q, plotted against the parameter values of the output error weight parameters.

The pitch rate shows similar behaviour as the roll rate, which is expected since they both have a counter part, x-acceleration and y-acceleration, that require angular rates to perform tilt-coordination.

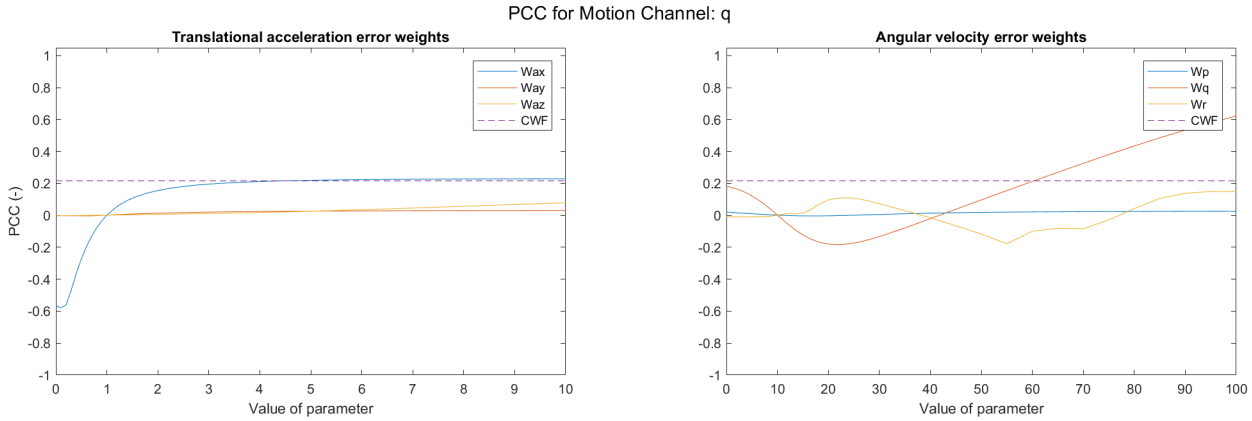


Figure 5.12: The PCC between desired and actual q, plotted against the parameter values of the output error weight parameters.

Table 5.6 shows the change in RMSE and PCC for motion channel q, and again it can be observed that the effect of changing W_{ay} and W_p on motion channel p is comparable to the effect of changing W_{ax} and W_q on motion channel q. Note that the immensely high increase in PCC due to changing W_q is due to the fact that the PCC in the baseline condition is very close to zero, making any change results in a large relative change.

Table 5.6: The relative changes in RMSE and PCC for motion channel q when changing the output error weight parameters.

Parameter	W_{ax}	W_{ay}	W_{az}	W_p	W_q	W_r
Relative change in RMSE	0.5617	-0.03812	-0.06576	0.01715	-0.9544	5.75
Relative change in PCC	122.0	15.31	41.19	12.9	332.4	78.94

The yaw rate is always difficult to simulate, because most of the times driving a curve requires more yaw rate than a hexapod simulator can provide. Therefore, no matter what you do, you will often fall short on providing yaw rate in a motion simulator. That is also seen in Figures 5.13 and 5.14. The RMSE only slight decreases when increasing Wr, but as seen earlier in Figure 5.4, this has bad consequences, as this increased performance of the yaw rate decreases the performance of all other motion channels.

Finally, Table 5.7 shows the change in RMSE and PCC for motion channel r. Interestingly, all parameters, except for W_r, have a negative effect on the tracking of r. This is because the MPC-algorithm is already not able to track r very well, and putting increasing other parameters makes it even worse. Increasing W_r does result in a reduction in RMSE for motion channel r, although not as large as for other parameters and their corresponding motion channel.

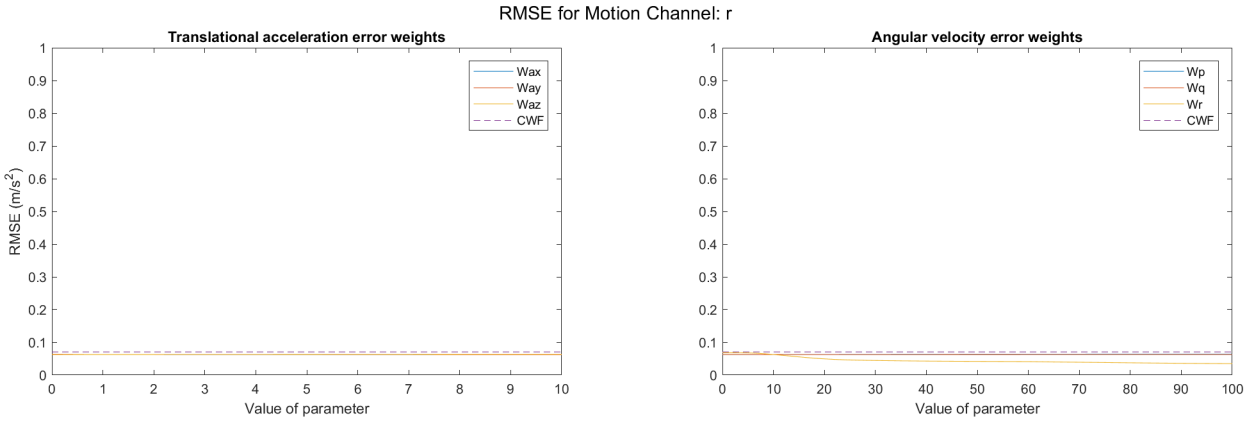


Figure 5.13: The RMSE between desired and actual r, plotted against the parameter values of the output error weight parameters.

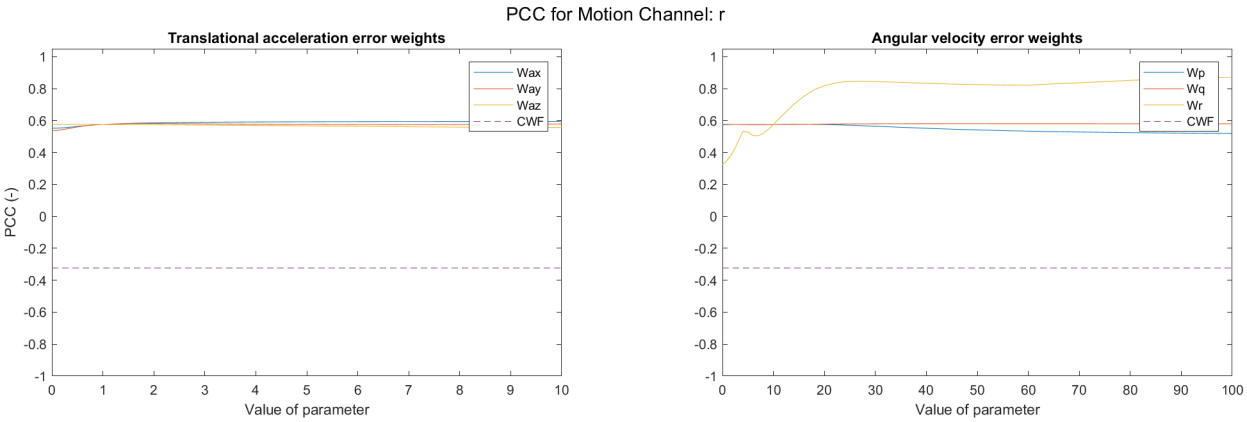


Figure 5.14: The PCC between desired and actual r, plotted against the parameter values of the output error weight parameters.

Table 5.7: The relative changes in RMSE and PCC for motion channel r when changing the output error weight parameters.

Parameter	W_{ax}	W_{ay}	W_{az}	W_p	W_q	W_r
Relative change in RMSE	-0.005011	-0.003741	0.01118	0.01712	0.002315	-0.444
Relative change in PCC	0.03174	0.005922	-0.03257	-0.09728	0.008043	0.5162

We have seen that all output error weight parameters mostly influence their corresponding motion channels, although it can also be observed that due to tilt-coordination, other parameters have an influence as well, often reducing the performance of that specific motion channel. We have also seen that changing W_{ax}, W_{ay}, W_p and W_q have more influence on their corresponding motion channels than W_{az} and W_r. An important thing to notice is that in some cases, multiple parameters have an influence on one motion channel, therefore it is important to see what happens when both are changed simultaneously, which will be done in the next section.

5.2.3. TWO-WAY SENSITIVITY PLOTS

The previous subsection looked into the behaviour of different individual motion channels while individually changing the output error weight parameters. During that part of the analysis, one output error weight parameter changed, while all others remained constant. This subsection will look at what happens when two output error weight parameters are changed simultaneously, by doing the exact same thing as before, but now three-dimensional plots are created. This also shows whether two parameters are influenced by each other, which is the case for some of them. First, two parameters will be changed simultaneously to see if the sensitivity of the parameters changes, in other words, it will be checked whether the parameters influence each other. This will be done by determining the relative change, using Equation 5.1 with $BaselineRMSE/PCC$ being the baseline condition and also when one of the other parameters is increased. This means that we will be able to see if the sensitivity changes when one other parameter is changed as well. When relevant, the parameter combinations will be shown visually in 3D-plots to explain some of the behaviour. All 3D-plots of the possible parameter combinations can be found in Appendix A.2.

The parameter sensitivity correlation will be determined per parameter, resulting in a table with all sensitivities calculated as a result of increasing one or two parameters. The bold lines indicate the individual sensitivity due to changing only one parameter, calculated per motion channel. The other rows show the sensitivities when the parameter from the first column is increased as well. A different value between the rows indicate that the parameter from the first column has an effect on the sensitivity of the bold parameter.

Table 5.8 shows the sensitivity correlation of W_{ax} with the other parameters. It can be seen that W_q reduces the effect of changing W_{ax} when looking at ax, meaning that these parameters are correlated. This is as expected, since providing ax is mostly done via tilt-coordination, which introduces an error in q. Therefore changing W_{ax} influences ax and q, while W_q also influences both motion channels. Note that introducing both these parameters has a very large effect on q, but this is because the RMSE is nearly zero when only increasing W_q, causing the relative change to be deceptive. The effect is visualised in Figure 5.15,

Table 5.8: Relative changes in RMSE per motion channel for all parameter combinations involving W_{ax}.

Added parameter	ax	ay	az	p	q	r
W_{ax}	**-0.9412**	**0.0028**	**0.2197**	**0.0183**	**0.5617**	**-0.0050**
W_{ay}	-0.9391	0.0139	0.0698	0.0037	0.6001	-0.0089
W_{az}	-0.9449	-0.0085	0.3503	0.0182	0.6664	-0.0043
W_p	-0.9479	0.0119	0.7570	0.0198	0.5448	0.0045
W_q	-0.5146	-0.1217	1.5540	0.4282	21.6756	0.1827
W_r	-0.9321	-0.1260	0.0657	-0.2033	-0.3346	0.1627

To illustrate the behaviour of the MPC-algorithm when changing W_{ax} and W_{ay}, Figure 5.16 shows what happens when W_{ax} and W_{ay} are changed simultaneously. It can be seen that these two parameters are barely influenced by each other, because for each motion channel, only one parameter changes the RMSE. If we look at the upper middle plot, ay, we see that W_{ax} does not influence the RMSE, not in the standard case but also not for higher values of W_{ay}. Therefore, we can say that these two parameters are uncorrelated. This also means that the related plots from Subsection 5.2.2 can be used solely.

Figure 5.17 belongs to W_{ax} and W_r. This figure is shown because here it can be seen that increasing W_r too much introduces large errors, even if other parameters increase as well. Although the effect of W_r is slightly suppressed, it is still there.

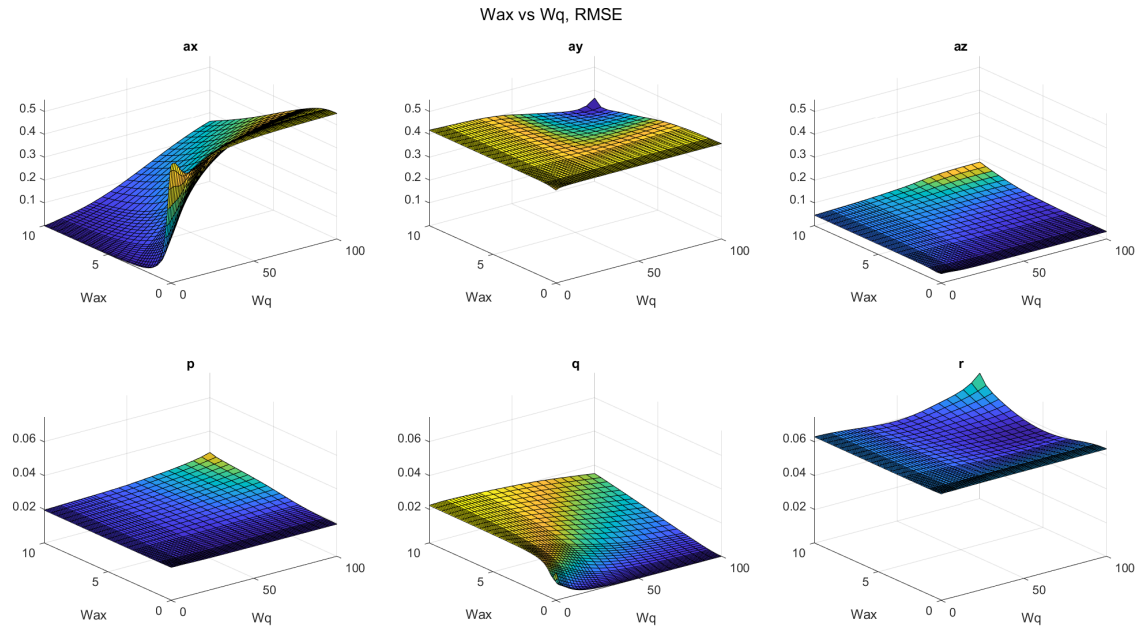


Figure 5.15: RMSE when changing Wax and Wq simultaneously.

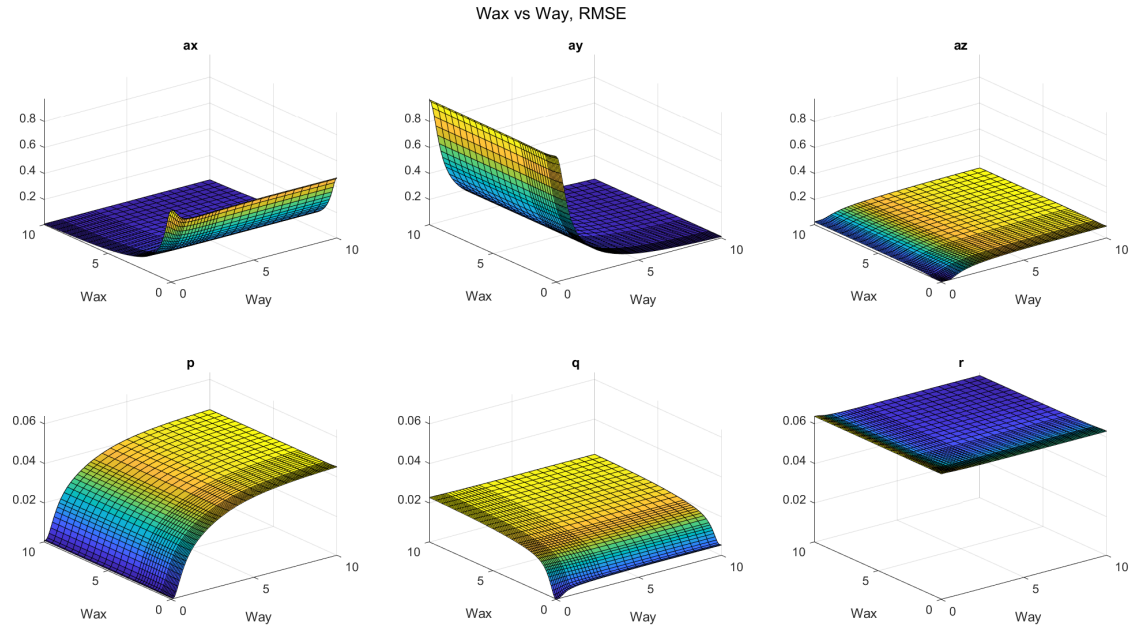


Figure 5.16: RMSE when changing Wax and Way simultaneously.

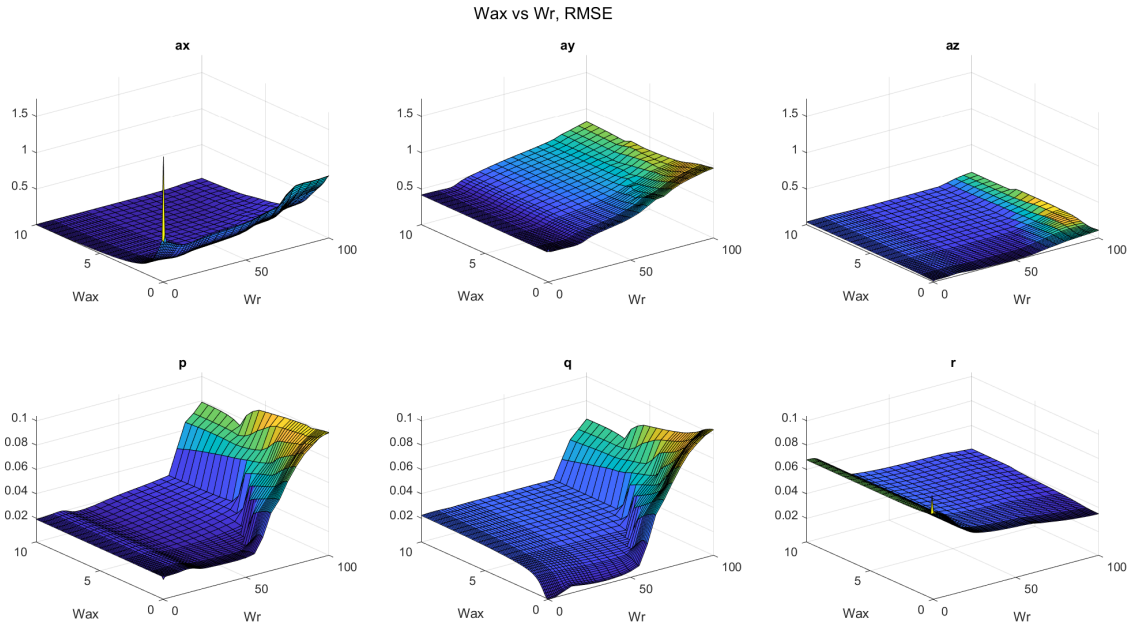


Figure 5.17: RMSE when changing Wax and Wr simultaneously.

The sensitivity correlation of W_{ay} with the other parameters can be found in Table 5.9. We see that W_{ay} and W_p, two parameters related via tilt-coordination, are correlated with each other. Figure 5.18 illustrates this behaviour, where it can be seen that all motion channels are influenced by the simultaneous increase of W_{ay} and W_p. If we look at what happens with the simulator motion (Figure 5.19), we see that ay is followed fairly well and the error in p is rather small, but all other stimuli start oscillating. This is because at a certain point, the values for W_{ay} and W_p are so high that a very small error in y-acceleration or roll rate is weighted more than a large error of the other stimuli, causing the algorithm to do weird things in order to decrease the error in y-acceleration or roll rate by a final little bit. Setting both parameter to a lower value shows that the difference in tracking y-acceleration and roll rate only slightly decreases, but other errors are reduced by much more, as is seen in Figure 5.20.
Also W_r seems to be correlated with W_{ay}. When looking at the 3D-plot, it becomes clear that this is caused by the fact that increasing W_r too much introduces oscillating behaviour, similar to the behaviour when both W_{ay} and W_p are increased too much. This behaviour was already shown in Figure 5.4.

Table 5.9: Relative changes in RMSE per motion channel for all parameter combinations involving W_{ay}.

Added parameter	ax	ay	az	p	q	r
W_{ax}	0.0085	-0.9569	1.1220	1.2600	-0.0145	-0.0076
W_{ay}	**-0.0260**	**-0.9574**	**1.4193**	**1.2929**	**-0.0381**	**-0.0037**
W_{az}	0.1214	-0.9380	2.1333	1.6892	0.0030	0.0152
W_p	2.5817	-0.6610	12.2521	15.8741	4.3445	1.3060
W_q	-0.0770	-0.9579	1.7536	1.3203	-0.1034	0.0007
W_r	-0.4739	-0.9484	-0.0577	-0.5861	-0.8284	0.2754

When looking at the sensitivity correlation of W_{Az} in Table 5.10, it seems that there is a lot of correlation. However, the sensitivities are all very small and if we look at the interaction between W_{az} and W_ay in Figure 5.21, we see that there is little to no correlation, even though the relative change in RMSE would suggest there is an effect.

The largest sensitivity correlation for parameter W_p is with W_{ay}, as can be seen in Table 5.12. This effect has already been explained when discussing W_{ay}, therefore it will not be discussed here anymore. Also correlation seems to be present between W_p and W_r, and also between W_q and W_r as can be seen in Table 5.12. If we

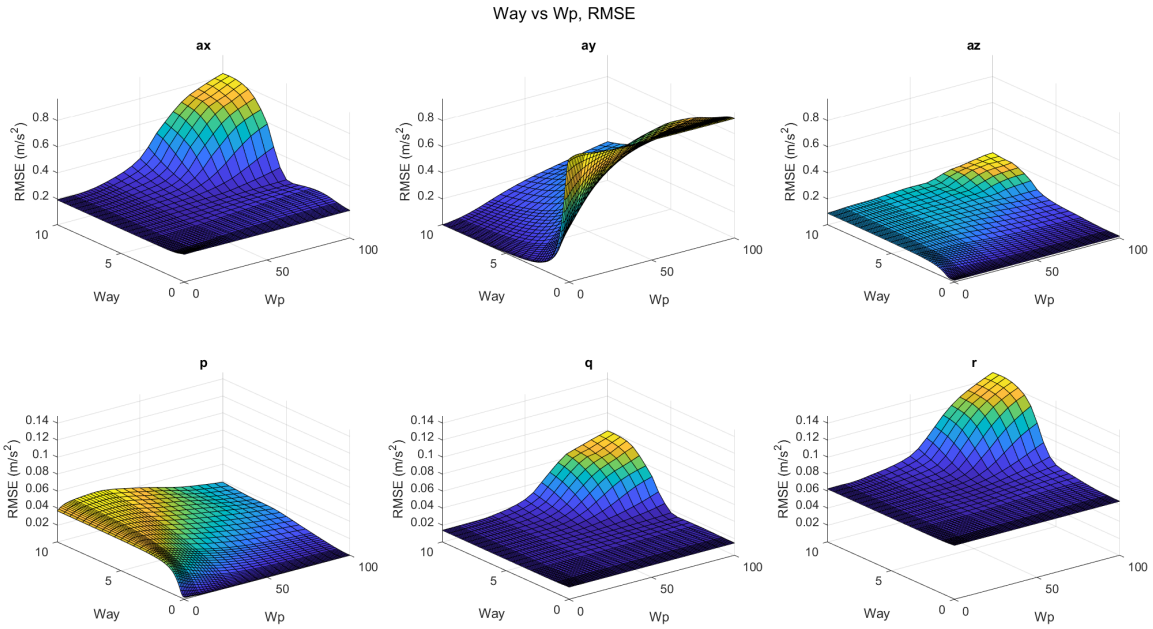


Figure 5.18: RMSE when changing W_{ay} and W_p simultaneously.

Table 5.10: Relative changes in RMSE per motion channel for all parameter combinations involving W_{az}.

Added parameter	ax	ay	az	p	q	r
W_{ax}	0.0566	0.1261	-0.2153	-0.1463	-0.0031	0.0119
W_{ay}	0.2986	0.6585	-0.0820	0.0013	-0.0259	0.0304
W_{az}	**0.1279**	**0.1390**	**-0.2912**	**-0.1462**	**-0.0658**	**0.0112**
W_p	-0.0604	0.0019	0.2443	0.0098	-0.0280	0.0021
W_q	0.0147	0.1067	-0.2658	-0.1009	-0.0036	0.0121
W_r	-0.1360	-0.0481	-0.7528	-0.0646	-0.0800	0.0369

look at the sensitivity correlation of W_r (Table 5.13), we can see that W_r seems to affect all other parameters. This is due to the fact that when W_r is increased too much, other motion channels start to show oscillating behaviour due to r being difficult to simulate, as was already explained in Subsection 5.2.2 and is clear from Figure 5.4. It can be concluded that W_r is correlated with all other parameters due to r being very difficult to simulate for a hexapod-based motion simulator. This leads to W_r having a strong impact and therefore should be tunes with care.

Table 5.11: Relative changes in RMSE per motion channel for all parameter combinations involving W_p.

Added parameter	ax	ay	az	p	q	r
W_{ax}	0.0244	1.1569	-0.4108	-0.9437	0.0062	0.0268
W_{ay}	3.2479	16.0185	1.2406	-0.5862	4.6516	1.3543
W_{az}	-0.0377	0.8802	-0.2819	-0.9335	0.0583	0.0080
W_p	**0.1552**	**1.1375**	**-0.5910**	**-0.9438**	**0.0171**	**0.0171**
W_q	0.0640	1.1519	-0.7092	-0.9414	0.0492	0.0082
W_r	-0.4607	-0.0275	-0.8660	-0.9768	-0.8200	0.1794

Note that a complete overview of all 3D-plots for each parameter combination can be found in Appendix A.2.

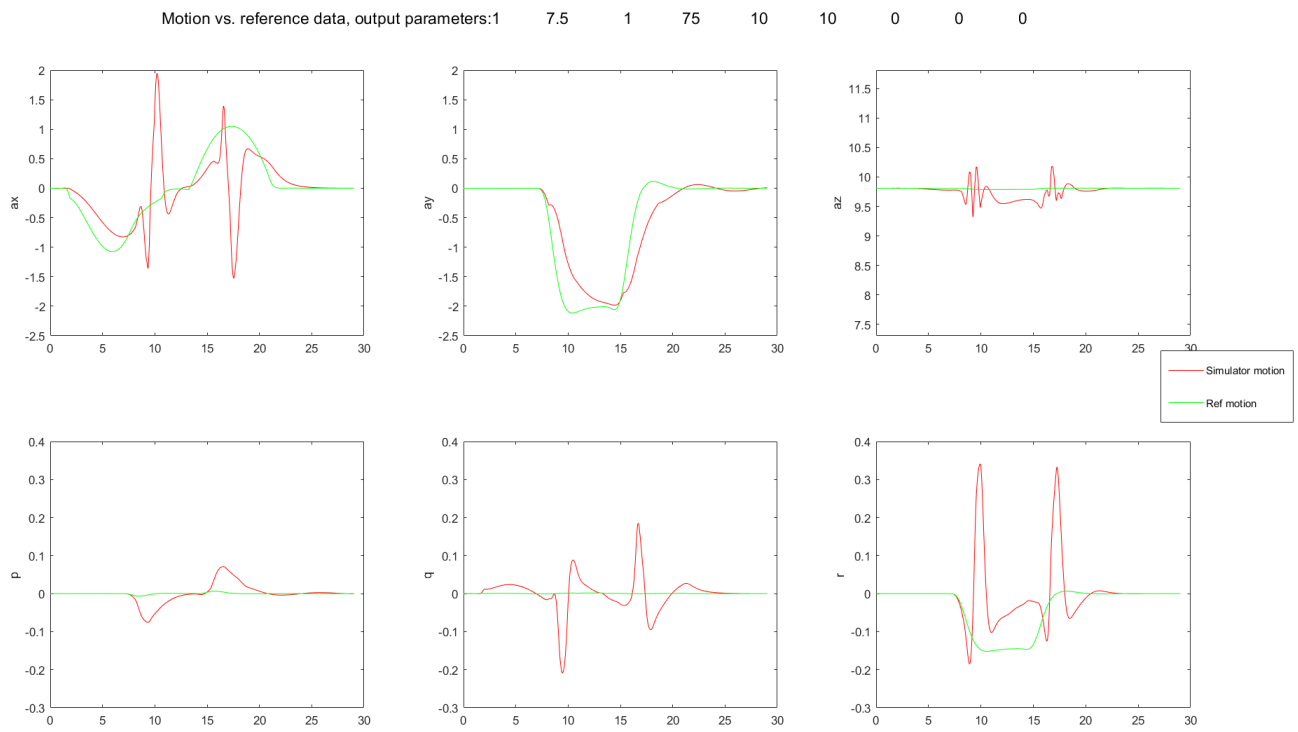


Figure 5.19: The output motion of the simulator with high W_{ay} and W_p values.

Table 5.12: Relative changes in RMSE per motion channel for all parameter combinations involving W_q.

Added parameter	ax	ay	az	p	q	r
W_{ax}	21.5770	-0.1204	0.7661	0.3879	-0.3376	0.1914
W_{ay}	1.5919	-0.0075	-0.0400	0.0014	-0.9575	0.0067
W_{az}	1.4605	-0.0241	-0.1263	0.0420	-0.9513	0.0033
W_p	1.5194	0.0111	-0.4004	0.0315	-0.9529	-0.0065
W_q	**1.7351**	**0.0044**	**-0.1566**	**-0.0104**	**-0.9544**	**0.0023**
W_r	-0.1378	-0.3147	-0.9103	-0.8228	-0.9801	0.1829

Table 5.13: Relative changes in RMSE per motion channel for all parameter combinations involving W_r.

Added parameter	ax	ay	az	p	q	r
W_{ax}	2.4670	0.9804	1.5178	2.9828	1.8761	-0.3502
W_{ay}	0.6224	1.7562	0.1223	-0.0810	0.2040	-0.2881
W_{az}	1.3008	0.8992	0.0051	4.5773	5.6476	-0.4298
W_p	0.4022	0.0339	-0.0557	1.1040	0.1947	-0.3553
W_q	-0.0532	0.5505	-0.6935	-0.0882	1.9483	-0.3438
W_r	**2.0035**	**1.2724**	**1.8816**	**4.0910**	**5.7505**	**-0.4440**

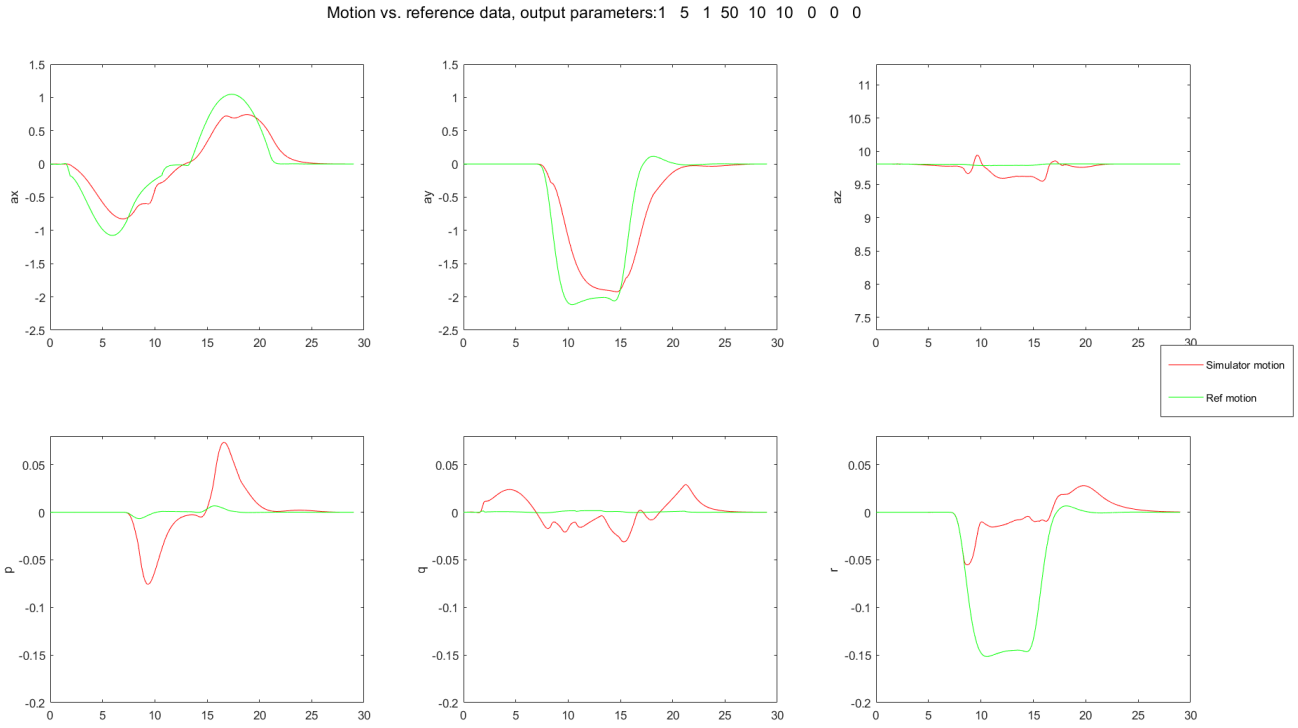


Figure 5.20: The output motion of the simulator with medium W_{ay} and W_p values.

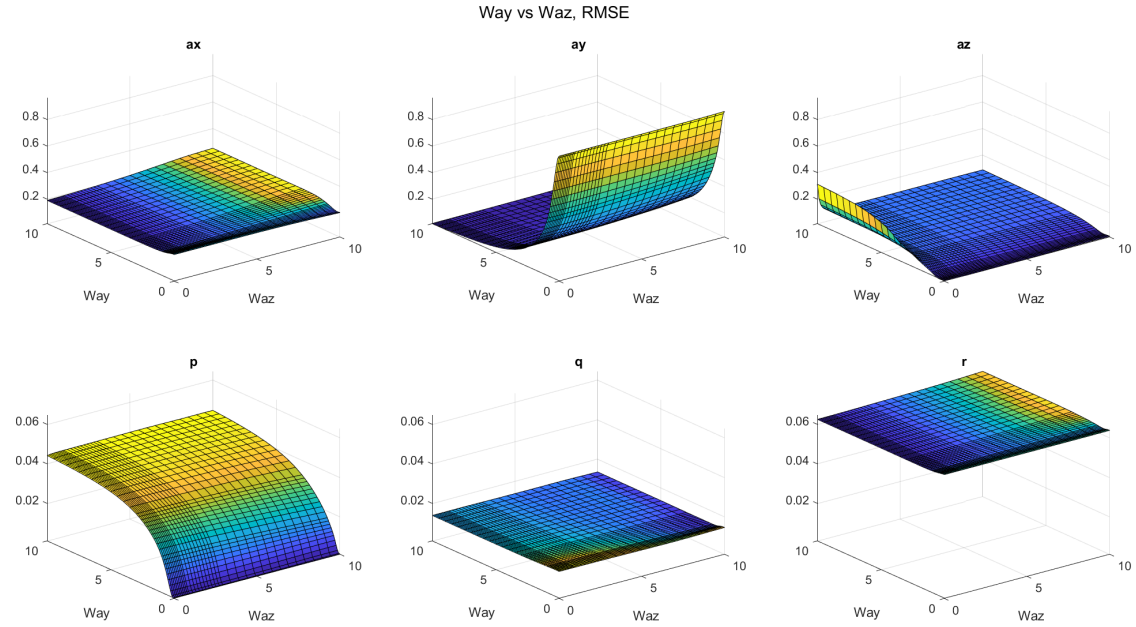


Figure 5.21: RMSE when changing W_{ay} and W_az simultaneously.

5.3. SENSIBLE PARAMETER VALUE RANGES

The previous section has shown what happens when the output error weight parameters are changed, and it became apparent that not all combination deliver satisfying results. In some cases, the algorithm becomes unstable and starts to produce oscillations on some motion channels, something that is undesirable. This section will explain why the used parameter ranges were chosen.

First, the translational error weight parameters range from 0-10 and the angular rates range from 0-100. This difference in scale has the same reason as the values of the parameters for the standard case, and that is due to the difference in units. Looking at the values for common vehicle manoeuvres, the values for translational accelerations in m/s^2 are approximately ten times as high as the values for angular rates in rad/s. This means that to achieve a penalty of equal magnitude on the angular rates in rad/s, the weight needs to be ten times as high as the weight on the translational accelerations in m/s^2.

Second, the translational accelerations are only shown until the value 10 and the angular rates until the value 100. This is because after that, the change in RMSE or PCC becomes rather small in most cases. Other combinations show that undesired behaviour starts happening way before these values, making it obsolete to investigate higher values. Undesired behaviour does not mean that large false cues are introduced, since this is often an expected result. For example, ensuring a perfect tracking of ay causes large errors in p, and this is not undesired behaviour but it is simply a logical consequence. What is meant with undesired behaviour is the oscillating behaviour of certain motion channels as a result of increasing certain parameters too much. Note that this applies because all other parameter values, thus including the state, terminal state and input error weight parameters, have values in that same range. Changing all parameters the same relatively will produce the same result.

There are two cases where undesired behaviour occurs. First, W_r has to be tuned carefully, as increasing it too much introduces oscillating behaviour. Figure 5.17 already shows that increasing W_r above 50 increases the RMSE rapidly. This is also the case when changing one of the other parameters, therefore W_r should not be set to values above 50. The second case is when W_{ay} and W_p are increased simultaneously. Figure 5.18 already showed that the RMSE starts to increase rapidly when both parameters are increased. The point where this starts to happen is around $W_{ay} = 5$ and $W_p = 50$, but increasing these parameters individually does not introduce the unwanted behaviour. This also supports the need to analyse the behaviour of the MPC-algorithm when multiple parameters are changed.

6

EXPERIMENT PROPOSAL

Now that the first part of the analysis has been performed, it is time to summarise the results and investigate a selection of parameter values more thorough by performing an experiment with human subjects. The impact of the six output error weight parameters on the performance of the MPC-based MCA has been studied by looking at the RMSE and PCC of the desired and actual simulator output signals with many different values of the parameters. But as has already been discussed, these metrics are merely an indication of the quality of a simulator, thus these results are not conclusive. In order to verify this approach of looking at motion cueing quality and to actually determine the effect of changing the output error weight parameters on the motion cueing quality, an experiment is proposed. In this experiment, the motion cueing quality will be determined by using the continuous rating method, explained in Section 4.2.

Since we are dealing with a curve driving manoeuvre, performed by a car, ay is the most dominant of the stimuli. The second most important stimulus is the yaw rate, but as has been observed, simulating yaw rate is difficult on a hexapod simulator. Increasing W_r does not really improve the performance of the simulation and quickly results in undesirable behaviour. Therefore, it has been decided that W_{ay} and W_p will be investigated further in the experiment, since W_p has the highest correlation with W_{ay} due to their interaction in tilt-coordination for producing ay. The idea is to have a baseline condition, which will be the standard MPC condition as is explained in Subsection 3.2.1, and increase and decrease the parameters the same relatively. Then both of these parameters will be increased on their own, but also simultaneously, meaning that in total, there will be nine experimental conditions. After careful consideration, it has been determined that both parameter values are increased and decreased by 60 %, meaning that the parameter values for the experiment are as indicated in Table 6.1. The final result has been a trade-off between getting a large enough difference between the experiment conditions and keeping the simulation comfortable, since human subjects will take place inside a simulator. This means that there must be no negative PCC on any of the motion channels, since this indicates that there are false cues present that are in opposite direction of what is expected and this might cause motion sickness.

Table 6.1: The parameter values of all nine experiment conditions.

Experiment condition	Wax	Way	Waz	Wp	Wq	Wr
1	1	0.4	1	4	10	10
2	1	1	1	4	10	10
3	1	1.6	1	4	10	10
4	1	0.4	1	10	10	10
5 (MPC baseline)	1	1	1	10	10	10
6	1	1.6	1	10	10	10
7	1	0.4	1	16	10	10
8	1	1	1	16	10	10
9	1	1.6	1	16	10	10

6.1. METHOD

20 human subjects will be asked to take place inside the hexapod simulator of the Max Planck Institute for Biological Cybernetics. During the simulation, the participant will see computer-generated visuals projected on a screen in front of them that match the vehicle motion, see Figure 6.1. Based on those visuals, participants should have a certain expectation of what they would feel if they would be situated in a real car. But because they are not in a real car but in a motion simulator, discrepancies can occur between their expectation and what they really perceive. If this occurs, participants should indicate this by turning a knob that indicates to what extent they feel a mismatch, also known as the continuous rating method described in Section 4.2. The knob is shown in Figure 6.2.

Figure 6.1: The visuals that subjects will see during the experiment.

Figure 6.2: The turning knob with which participants can indicate the perceived motion mismatch. The rating bar on the background shows the current rating.

The manoeuvre is exactly the same as used in the analysis, consisting of a car first driving straight at 70 km/h, slowing down to 50 km/h, doing a curve of 60° with a radius of curvature of approximately 90m and then accelerating back to 70 km/h. Note that all control, dependent and independent variables are the same as described in Section 5.1, but now only two of the output error weight parameters will be changed. In order to speed up the experiment, all of the nine conditions will be played back to back, meaning that one trial consists of nine curves. An initial acceleration and final deceleration will be included to make the trial more realistic. Since doing a continuous mismatch rating is not a straightforward method for participants, two training trials will be performed to let the participant get used to the method and to get a feeling for what motion feels good and what motion feels bad. After that, participants will have to rate three more trials, in order to get rid of outlier behaviour and to make sure participant rate consistently.

During the experiment, the continuous rating will be measured. The continuous rating shows the time-varying perception of the simulation, and in order to help in the interpretation of these results, a questionnaire will be filled in afterwards that asks participant how they decided on a certain rating. On top of that, after each experiment trial a sickness score is asked to monitor motion sickness during the experiment. The experiment instructions and questionnaire that will be handed to participant can be found in Appendix C.

6.2. APPARATUS

The experiment will be performed on the hexapod simulator of the Max Planck Institute for Biological Cybernetics, a commonly available motion simulator. Specifications of this simulator can be found in Table 6.2. The visuals will be created in Unity and projected on a screen using a real-time projector running at 120 Hz, with a horizontal field of view of 66 ° and a vertical field of view of 39 °. Figures 6.1 and 6.2 show the visuals that participants will see and the knob that will be used to give a rating, respectively.

6.3. GOAL AND HYPOTHESES

The goal of the experiment is to investigate the influence of the error weight parameters on the motion cueing quality, more specifically W_{ay} and W_p. It is expected to see lower mismatch rating when the RMSE is small

	Minimum Position	Maximum Position	Velocity	Acceleration
Surge	-0.499 m	0.628 m	± 0.79 m/s	±7.0 m/s^2
Sway	-0.506 m	0.506 m	± 0.81 m/s	±7.0 m/s^2
Heave	-0.383 m	0.372 m	± 0.55 m/s	±10.0 m/s^2
Roll	-24.01 \degree	24.01 \degree	± 34.3 \degree/s	±250 \degree/s^2
Pitch	-25.05 \degree	28.02 \degree	± 37.4 \degree/s	±250 \degree/s^2
Yaw	-27.25 \degree	27.25 \degree	± 41.3 \degree/s	±500 \degree/s^2

Table 6.2: Simulator actuator velocity and acceleration limits.

and the PCC is close to 1. The values for the RMSE and PCC per motion condition for all experiment conditions are shown in Table 6.3 and the simulator output is shown in Figure 6.3. Looking at the values, we can see that C3 (condition number 3) has the lowest RMSE in ay, but this also means that the RMSE in p in the highest, due to the use of tilt-coordination. And if we look at Figure 6.3, we can see that p rises to twice the perception threshold of 3 $°/s$ [20], or 0.05 rad/s. But it might also be possible that participants do not notice p due to being distracted by the visuals or other present motion. On top of that, [23] argues that priority should be given to the minimisation of specific force errors, even if this introduces perceptible tilt rates, since these contribute more to the realism of driving simulation. This can be explained due to the fact that the human vestibular system is not perfect and in case of strong expected motion, the vestibular system might associate all strong motion with good and correct motion, but this can also vary between humans. Therefore it is possible that C3 will rated as the best condition. In this case, C7 will probably be rated as the worst conditions, because there the RMSE of ay is the highest of all conditions, because there no tilt-coordination is used due to the parameter settings.

Conditions C2, C3 and C6 all have a tilt-coordination part that surpasses the 3 $°/s$ perception threshold, and if the false cues are noticed, these conditions will probably receive a high mismatch rating. In this case, C5 and C9 will likely receive the lowest mismatch rating, and thus be rated most realistic, since these conditions provide the lowest RMSE while still staying under the perception threshold for p.

Table 6.3: The RMSE and PCC values for all experiment conditions per motion channel.

	RMSE						PCC					
Condition	ax	ay	az	p	q	r	ax	ay	az	p	q	r
1	0.2065	0.6657	0.0153	0.0100	0.0146	0.0633	0.9387	0.9608	0.4488	0.8276	0.0007	0.5556
2	0.1957	0.2770	0.0497	0.0255	0.0143	0.0627	0.9475	0.9822	0.9233	0.9192	0.0121	0.5755
3	0.1938	0.1479	0.0708	0.0320	0.0141	0.0626	0.9494	0.9919	0.9528	0.9629	0.0217	0.5791
4	0.2091	0.7452	0.0160	0.0064	0.0147	0.0633	0.9364	0.8542	0.1721	0.6367	-0.0021	0.5546
5	0.1986	0.4139	0.0398	0.0195	0.0145	0.0629	0.9445	0.9264	0.8077	0.7375	0.0019	0.5756
6	0.1956	0.2720	0.0607	0.0265	0.0143	0.0627	0.9472	0.9608	0.8925	0.8409	0.0099	0.5812
7	0.2116	0.8045	0.0168	0.0042	0.0147	0.0634	0.9344	0.7098	-0.0122	0.5704	-0.0008	0.5525
8	0.2021	0.5320	0.0347	0.0150	0.0146	0.0629	0.9414	0.8319	0.6323	0.5803	-0.0033	0.5773
9	0.1978	0.3812	0.0552	0.0220	0.0145	0.0627	0.9447	0.9068	0.7969	0.7012	-0.0024	0.5870

6.4. FINAL WORDS

This reports ends with the experiment proposal that has been described in this chapter. The data that will be gathered during this experiment will be used to test the hypotheses and the results will be reported in the form of a paper.

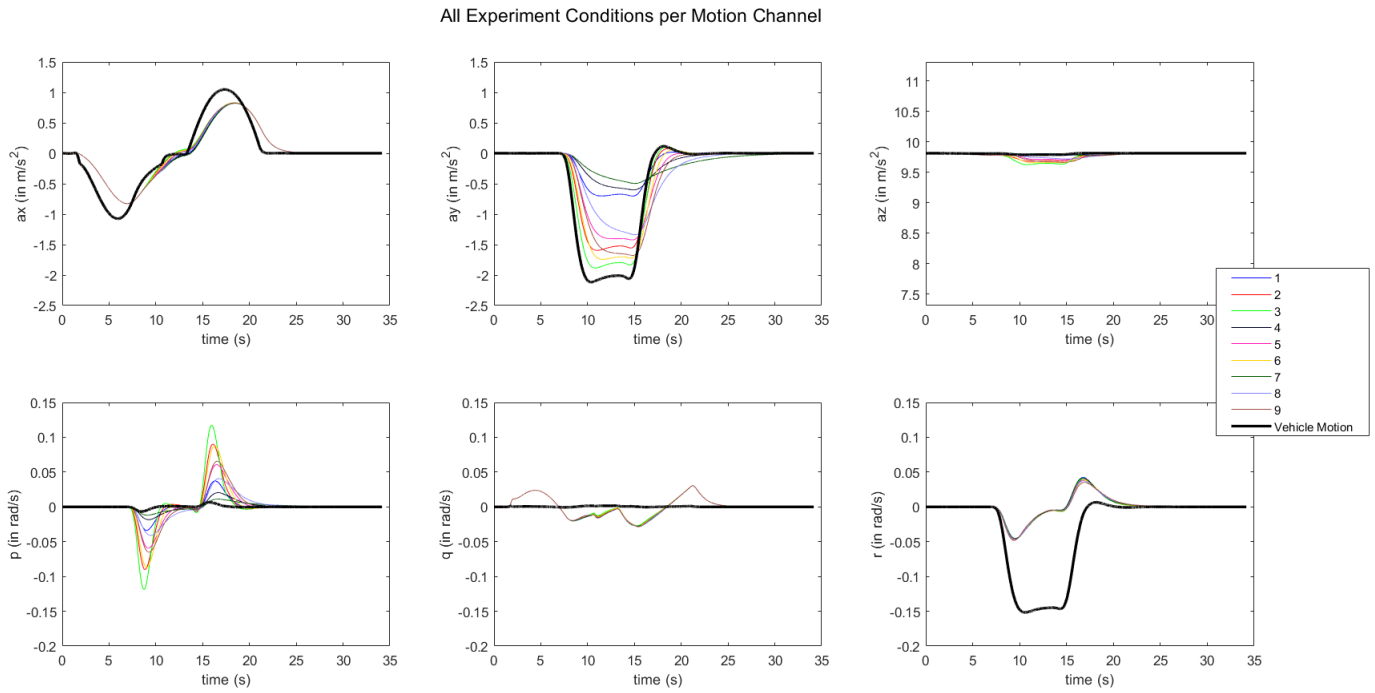


Figure 6.3: The output of the simulator for all nine experiment conditions.

BIBLIOGRAPHY

[1] J. H. Hogema, M. Wentink, and G. P. Bertollini, *Effects of Yaw Motion on Driving Behaviour, Comfort and Realism,* in *Proceeding of the Driving Simulation Conference 2012, Paris, France* (2012).

[2] P. R. Grant, B. Yam, R. J. A. W. Hosman, and J. A. Schroeder, *Effect of Simulator Motion on Pilot Behavior and Perception,* Journal of Aircraft **43**, 1914 (2006).

[3] A. R. Valente Pais, M. Wentink, M. M. van Paassen, and M. Mulder, *Comparison of Three Motion Cueing Algorithms for Curve Driving in an Urban Environment,* Presence: Teleoperators & Virtual Environments **18**, 200 (2009).

[4] B. Conrad and S. F. Schmidt, *Motion Drive Signals for Piloted Flight Simulators,* Tech. Rep. NASA CR-1601 (National Aeronautics and Space Administration, Ames Research Center, 1970).

[5] M. A. Nahon and L. D. Reid, *Simulator Motion-Drive Algorithms: A Designer's Perspective,* Journal of Guidance, Control, and Dynamics **13**, 356 (1990).

[6] M. Katliar, J. Fisher, G. Frison, M. Diehl, H. Teufel, and H. H. Bülthoff, *Nonlinear model predictive control of a cable-robot-based motion simulator.* in *International Federation of Automatic Control* (2017).

[7] D. Cleij, J. Venrooij, P. Pretto, M. Katliar, H. H. Bülthoff, D. Steffen, F. W. Hoffmeyer, and H.-P. Schöner, *Comparison between filter-and optimization-based motion cueing algorithms for driving simulation.* in *Transportation Research Part F: Traffic Psychology and Behaviour* (2017).

[8] F. M. Nieuwenhuizen and H. H. Bülthoff, *The MPI CyberMotion Simulator: A Novel Research Platform to Investigate Human Control Behavior,* Journal of Computing Science and Engineering **7**, 122 (2013).

[9] P. R. Giordano, C. Masone, J. Tesch, M. Breidt, M. Pollini, and H. H. Bülthoff, *A novel framework for closed-loop robotic motion simulation - part i: Inverse kinematicsdesign.* in *IEEE International Conference on Robotics and Automation* (2010).

[10] T. D. van Leeuwen, D. Cleij, D. M. Pool, M. Mulder, and H. H. Bülthoff, *Time-varying perceived motion mismatch due to motion scaling in curve driving simulation,* Transportation Research Part F: Traffic Psychology and Behaviour (2018), 10.1016/j.trf.2018.05.022.

[11] A. Berthoz, W. Bles, H. H. Bülthoff, B. J. Correia Gräcio, P. Feenstra, N. Filliard, R. Hähne, A. Kemeny, M. Mayrhofer, M. Mulder, H.-G. Nusseck, P. Pretto, G. Reymond, R. Schlässelberger, J. Schwandter, H. J. Teufel, B. Vailleau, M. M. van Paassen, M. Vidal, and M. Wentink, *Motion Scaling for High-Performance Driving Simulators,* IEEE Transactions on Human-Machine Systems **43**, 265 (2013).

[12] D. Cleij, J. Venrooij, P. Pretto, D. M. Pool, M. Mulder, and H. H. Bülthoff, *Continuous rating of perceived visual-inertial motion incoherence during driving simulation,* in *Proceedings of the Driving Simulation Conference 2015 Europe, Tübingen, Germany* (2015).

[13] E. Blana, *A survey of driving research simulators around the world,* in *Institute of Transport Studies, University of Leeds* (1996).

[14] D. M. Pool and P. M. T. Zaal, *A Cybernetic Approach to Assess the Training of Manual Control Skills,* in *Proceedings of the 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Kyoto, Japan* (2016).

[15] F. M. Nieuwenhuizen, M. Mulder, M. M. van Paassen, and H. H. Bülthoff, *The Influence of Motion System Characteristics on Pilot Control Behaviour,* in *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Portland, Oregon, Aug. 8-11,* AIAA-2011-6321 (2011).

[16] K. W. Kallus, K. Tropper, and W. Boucsein, *The Importance of Motion Cues in Spatial Disorientation Training for VFR-Pilots,* The International Journal of Aviation Psychology **21**, 135 (2011).

[17] H. J. Damveld, J. L. G. Bonten, R. Happee, M. M. van Paassen, and M. Mulder, *Motion filter design for driver observation in hexapod car simulators,* in *IEEE International Conference on Systems Man and Cybernetics (SMC), 2010* (2010) pp. 1344 – 1351.

[18] R. J. A. W. Hosman and J. C. van der Vaart, *Effects of Vestibular and Visual Motion Perception on Task Performance,* Acta Psychologica **48**, 271 (1981).

[19] P. R. Lakerveld, H. J. Damveld, D. M. Pool, K. van der El, M. M. van Paassen, and M. Mulder, *The Effects of Yaw and Sway Motion Cues in Curve Driving Simulation,* in *Proceedings of the 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Kyoto, Japan* (2016).

[20] E. Groen and W. Bles, *How to use body tilt for the simulation of linear self motion,* Journal of Vestibular Research: Equilibrium and Orientation, 14(5), 375–385 (2004).

[21] A. R. Valente Pais, D. M. Pool, A. M. de Vroome, M. M. van Paassen, and M. Mulder, *Pitch Motion Perception Thresholds During Passive and Active Tasks,* Journal of Guidance, Control, and Dynamics **35**, 904 (2012).

[22] A. Nesti, C. Masone, M. Barnett-Cowan, P. Robuffu Giordano, H. H. Bülthoff, and P. Pretto, *Roll Rate Thresholds and Perceived Realism in Driving Simulation,* in *Proceedings of the Driving Simulation Conference 2012 Europe, Paris 6-7 September 2012* (2012) pp. 1–6.

[23] H. Jamson, *Motion Cueing in Driving Simulators for Research Applications*, Ph.D. thesis, The University of Leeds, Institute for Transport Studies (2010).

[24] B. L. Aponso, S. D. Beard, and J. A. Schroeder, *The NASA Ames Vertical Motion Simulator – A Facility Engineered for Realism,* in *Proceedings of the Royal Aeronautical Society Spring 2009 Flight Simulation Conference, 3-4 June 2009, London, UK* (2009).

[25] D. Stewart, *A Platform With Six Degrees of Freedom,* Institution of Mechanical Engineers, Proceedings 1965-1966 **180 Part I**, 371 (1966).

[26] N. GARRETT and M. BEST, *Driving simulator motion cueing algorithms - a survey of the state of the art.* in *Proceedings of the 10th International Symposium on Advanced Vehicle Control (AVEC), Loughborough, UK, 22nd-26th August* (2010) pp. 183–188.

[27] L. D. Reid and M. A. Nahon, *Flight Simulation Motion-Base Drive Algorithms. Part 1: Developing and Testing the Equations*, Tech. Rep. UTIAS 296 (University of Toronto, Institute for Aerospace Studies, 1985).

[28] R. V. Parrish, J. E. Dieudonne, R. L. Bowles, and D. J. Martin, Jr., *Coordinated Adaptive Washout for Motion Simulators,* Journal of Aircraft **12**, 44 (1975).

[29] M. A. Nahon, L. D. Reid, and J. Kirdeikis, *Adaptive Simulator Motion Software with Supervisory Control,* Journal of Guidance, Control and Dynamics **15**, 376 (1992).

[30] A. R. Naseri and P. R. Grant, *An Improved Adaptive Motion Drive Algorithm,* in *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, San Francisco (CA)*, AIAA-2005-6500 (2005).

[31] R. Sivan, J. Ish-Shalom, and J.-K. Huang, *An Optimal Control Approach to the Design of Moving Flight Simulators,* IEEE Transactions on Systems, Man, and Cybernetics **SMC-12**, 818 (1982).

[32] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, *Model-based predictive motion cueing strategy for vehicle driving simulators,* Control Engineering Practice **17**, 995 (2009).

[33] M. Baseggio, A. Beghi, M. Bruschetta, F. Maran, and D. Minen, *An MPC approach to the design of motion cueing algorithms for driving simulators,* in *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA* (2011).

[34] N. J. Garrett and M. C. Best, *Model predictive driving simulator motion cueing algorithm with actuator-based constraints,* Vehicle System Dynamics **51**, 1151 (2013).

[35] J. Qin and T. Badgwell, *An overview of industrial model predictive control technology,* in *AIChE Symposium Series*, Vol. 93 (1997).

[36] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. MaÃ¯zi, *MPC Based Motion Cueing Algorithm: Development and Application to the ULTIMATE Driving Simulator,* in *Proceedings of the Driving Simulation Conference 2004 Europe, Paris, France* (2004) pp. 221–233.

[37] A. Beghi, M. Bruschetta, and F. Maran, *A real time implementation of MPC based Motion Cueing strategy for driving simulators,* in *Proceedings of the 51st IEEE Conference on Decision and Control, Maui, Hawaii* (2012).

[38] M. Katliar, F. Drop, H. Teufel, M. Diehl, and H. H. Bülthoff, *Real-time nonlinear model predictive control of a motion simulator based on a 8-dof serial robot.* in *17th European Control Conference (ECC 2018)* (2018).

[39] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, *Move blocking strategies in receding horizon control,* in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, Vol. 2 (2004) pp. 2023–2028 Vol.2.

[40] A. Beghi, M. Bruschetta, and F. Maran, *A real time implementation of MPC based Motion Cueing strategy for driving simulators,* in *Proceedings of the 51st IEEE Conference on Decision and Control, Maui, Hawaii* (2012).

[41] M. Diehl, H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, *Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations,* Journal of Process Control **12**, 577 (2002).

[42] S. Casas-Yrurzum, I. Coma, J. V. Riera, and M. Fernández, *Motion-cuing algorithms - characterization of users' perception,* Human Factors The Journal of the Human Factors and Ergonomics Society **57** (2015).

[43] H. M. Heerspink, W. R. Berkouwer, O. Stroosma, M. M. van Paassen, M. Mulder, and J. A. Mulder, *Evaluation of Vestibular Thresholds for Motion Detection in the SIMONA Research Simulator,* in *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, August 15–18, San Francisco (CA)*, AIAA-2005-6502 (2005).

[44] J. Venrooij, P. Pretto, M. Katliar, S. Nooij, A. Nesti, M. Lächele, K. N de Winkel, and D. Cleij, *Perception-based motion cueing: validation in driving simulation,* in *DSC 2016 Europe: Driving Simulation Conference & Exhibition, 31-38.* (2015).

[45] P. R. Grant, *The Development of a Tuning Paradigm for Flight Simulator Motion Drive Algorithms*, Ph.D. thesis, University of Toronto, Institute for Aerospace Studies (1996).

[46] M. Fischer, *Motion-Cueing-Algorithmen für eine realitätsnahe Bewegungssimulation*, Ph.D. thesis, Technische Universität Braunschweig, Fakultät Maschinenbau (2009).

[47] D. Cleij, J. Venrooij, P. Pretto, D. M. Pool, M. Mulder, and H. H. Bülthoff, *Continuous Subjective Rating of Perceived Motion Incongruence During Driving Simulation,* IEEE Transactions on Human-Machine Systems (2017), 10.1109/THMS.2017.2717884.

III

Paper Appendices

To be graded for final Thesis

A

Sensitivity Plots

In this appendix, the sensitivity plots are shown. Appendix A.1 shows the RMSE and PCC for each motion channel, where it can be seen how these metrics change when one of the parameters is varied. Appendix A.2 shows three-dimensional plots, where the RMSE and PCC are shown per motion channel as a result of varying two parameters simultaneously.

A.1. Sensitivity Plots

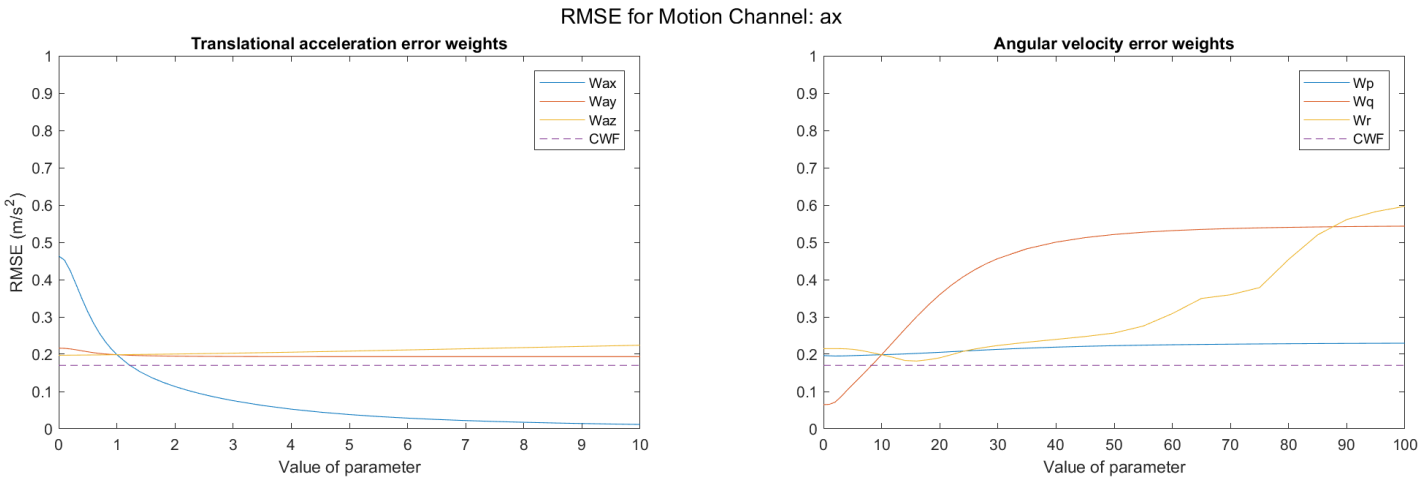


Figure A.1: The RMSE between desired and actual ax, plotted against the parameter values of the output error weight parameters.

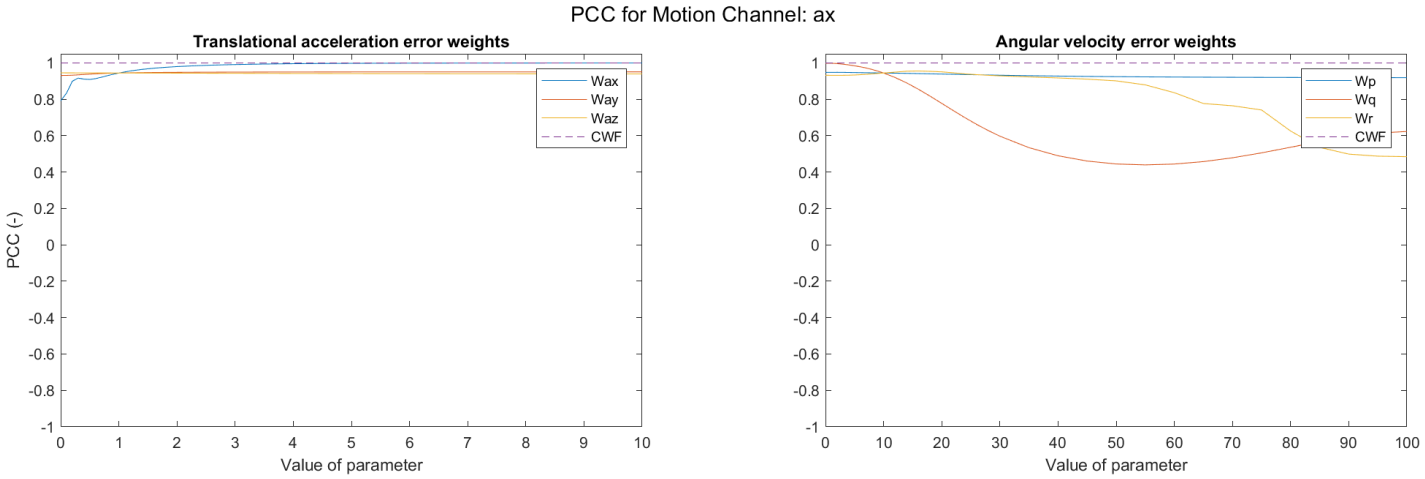


Figure A.2: The PCC between desired and actual ax, plotted against the parameter values of the output error weight parameters.

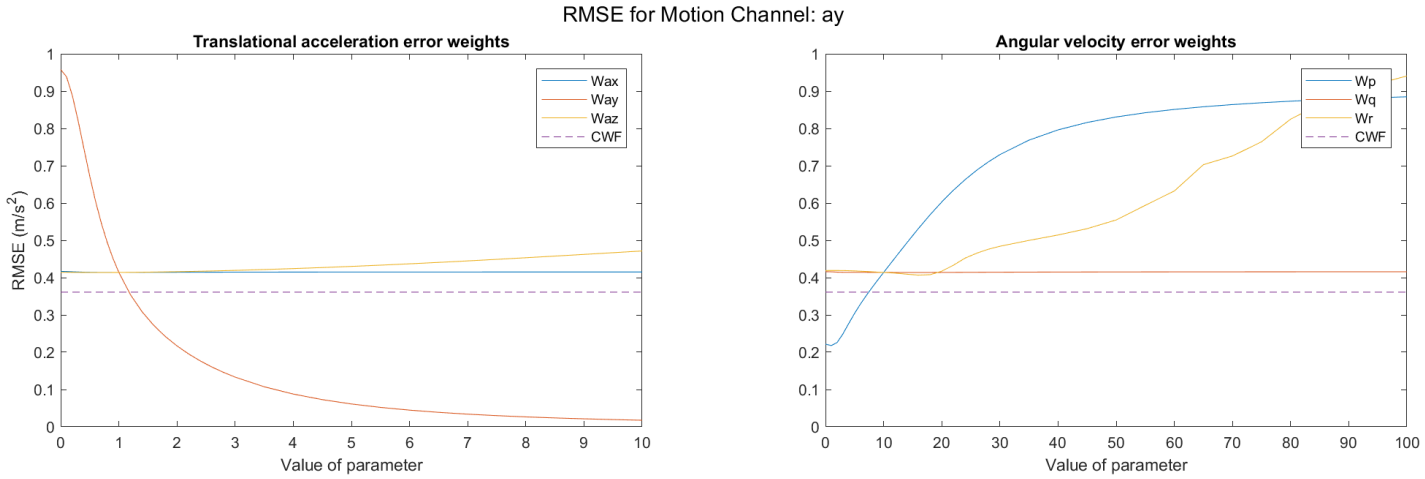


Figure A.3: The RMSE between desired and actual ay, plotted against the parameter values of the output error weight parameters.

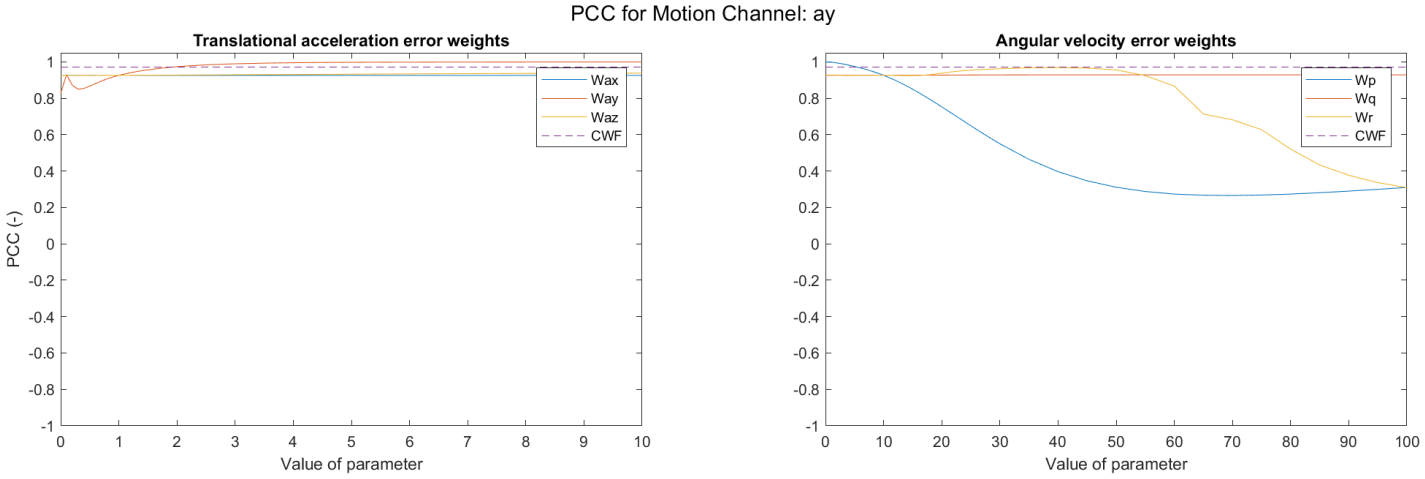


Figure A.4: The PCC between desired and actual ay, plotted against the parameter values of the output error weight parameters.

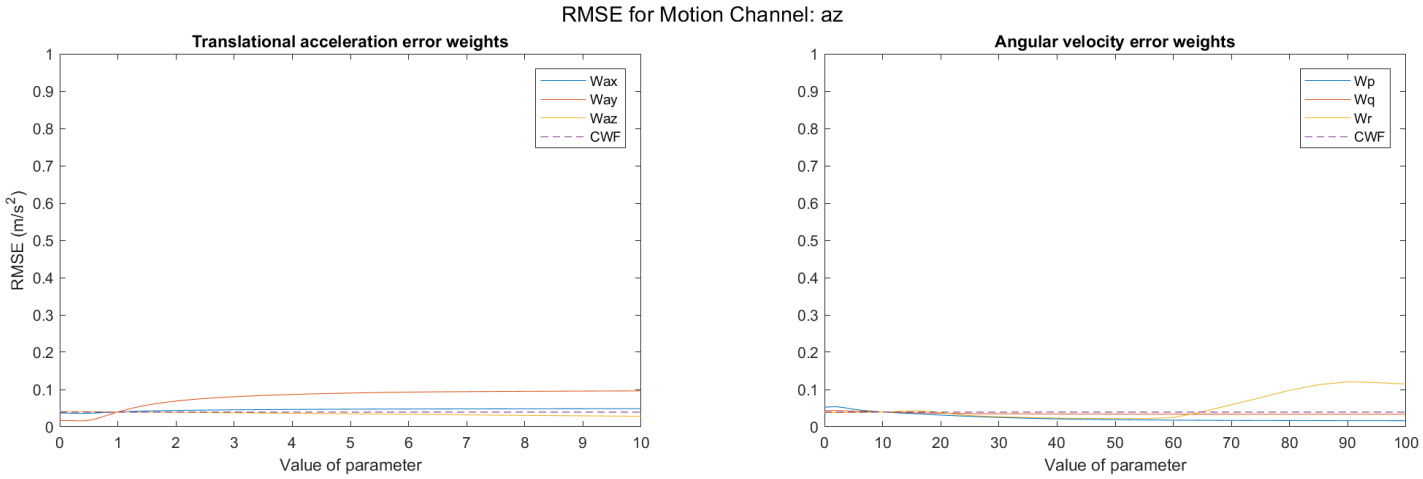


Figure A.5: The RMSE between desired and actual az, plotted against the parameter values of the output error weight parameters.

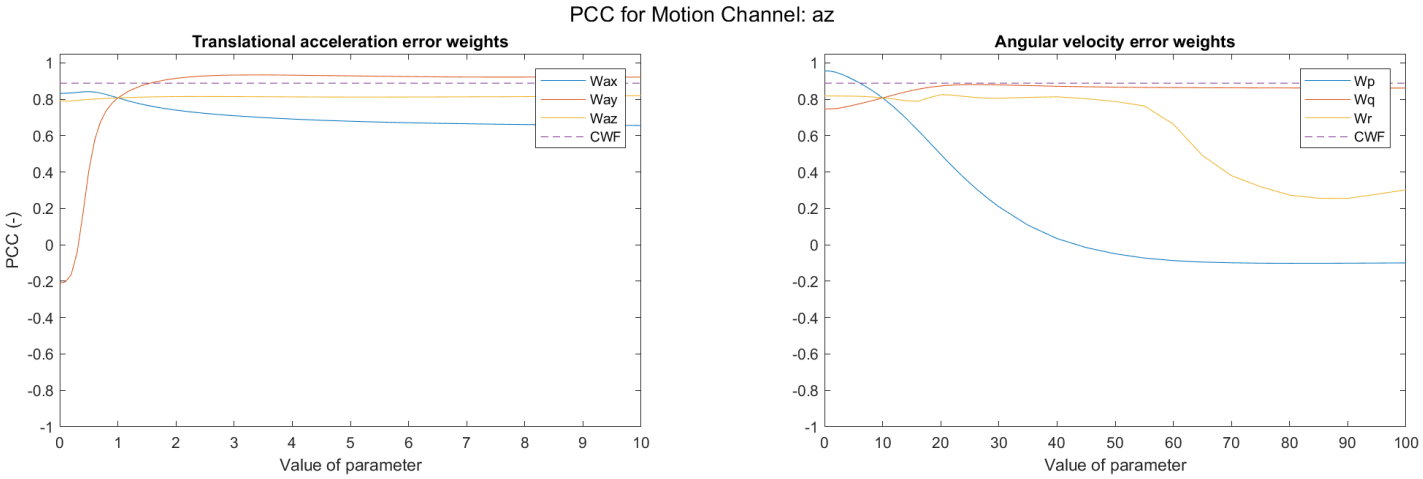


Figure A.6: The PCC between desired and actual az, plotted against the parameter values of the output error weight parameters.

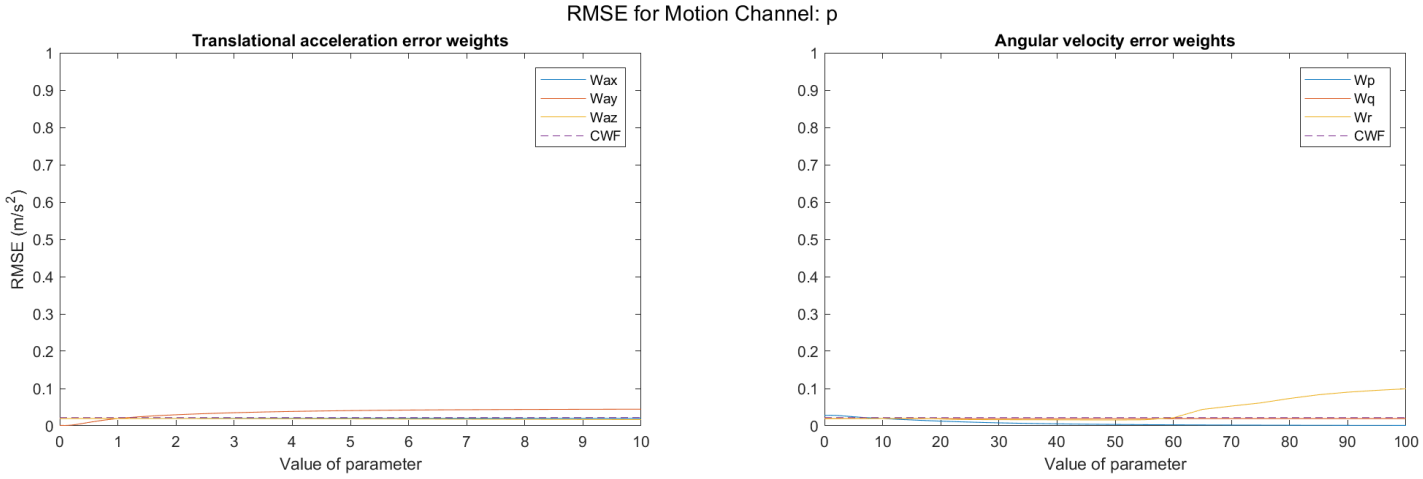


Figure A.7: The RMSE between desired and actual p, plotted against the parameter values of the output error weight parameters.

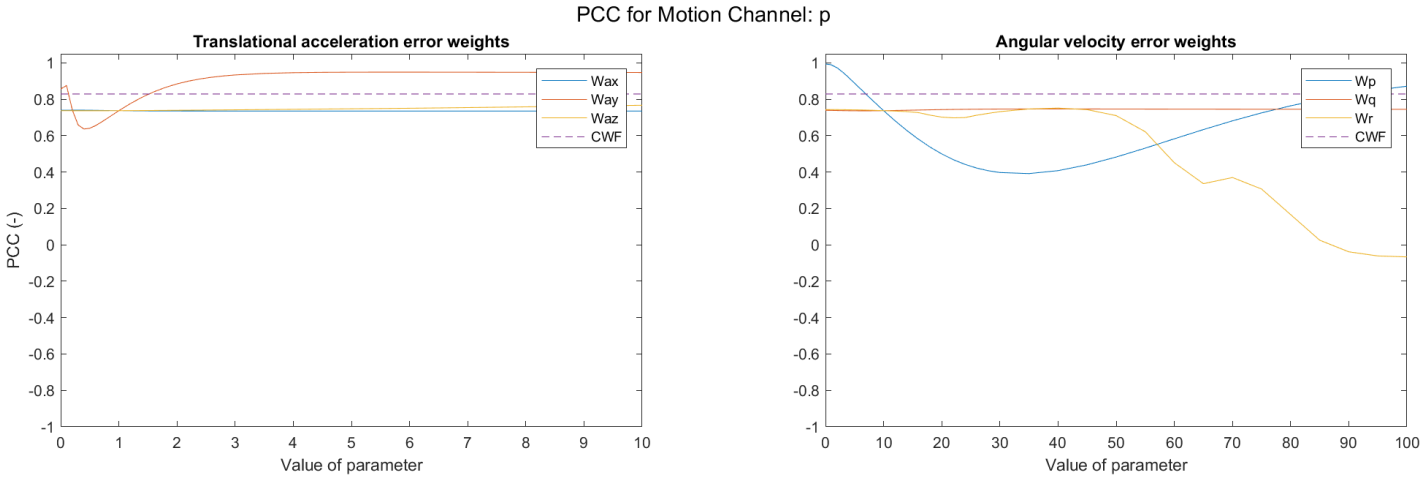


Figure A.8: The PCC between desired and actual p, plotted against the parameter values of the output error weight parameters.

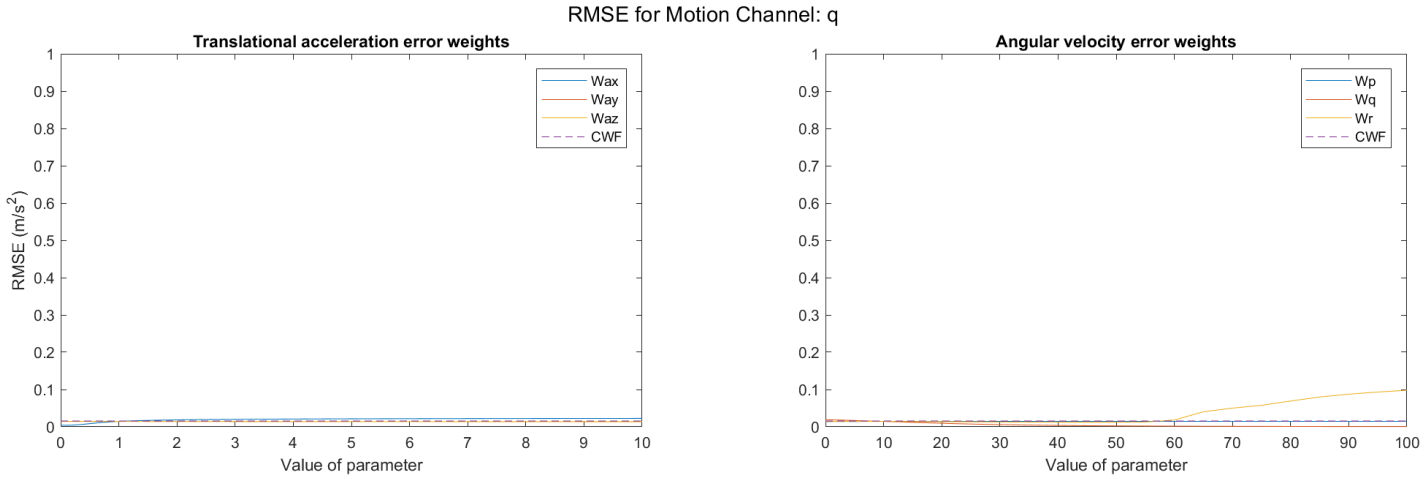


Figure A.9: The RMSE between desired and actual q, plotted against the parameter values of the output error weight parameters.

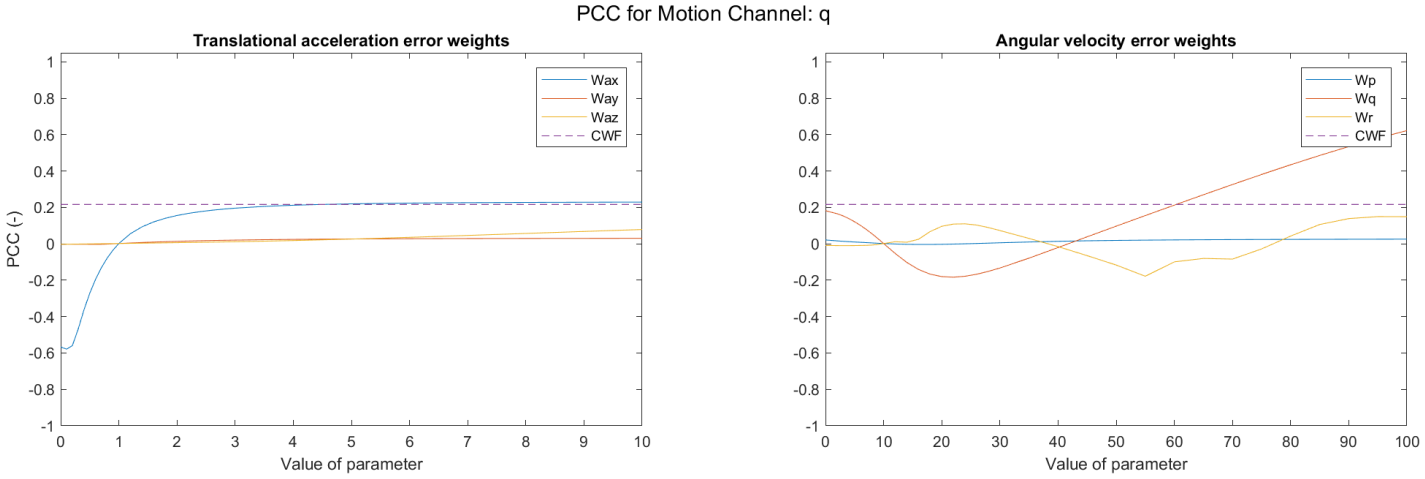


Figure A.10: The PCC between desired and actual q, plotted against the parameter values of the output error weight parameters.

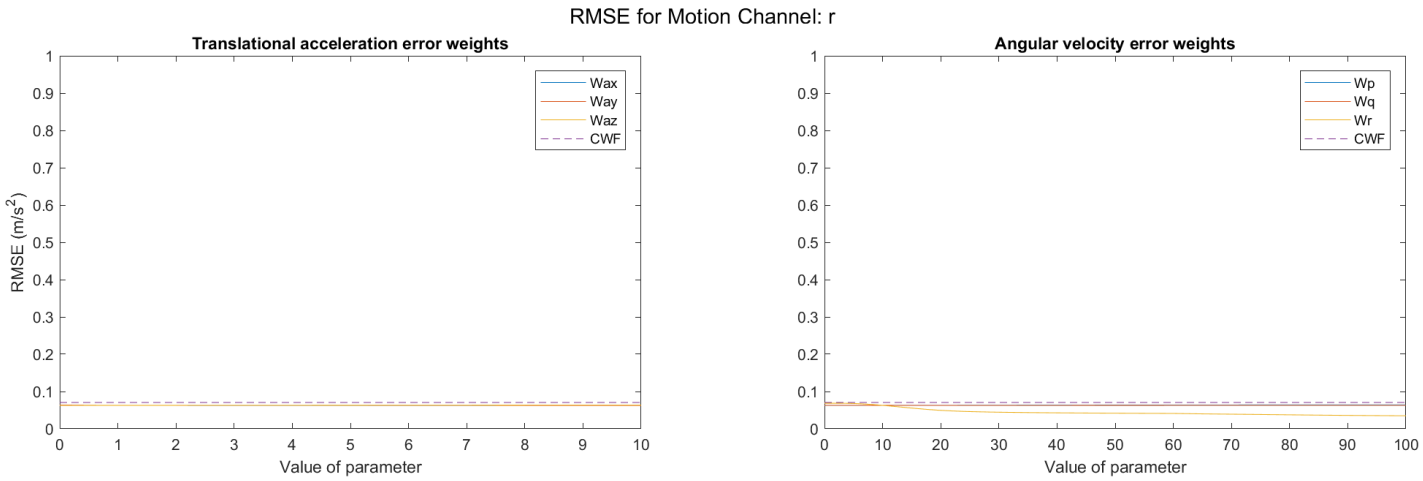


Figure A.11: The RMSE between desired and actual r, plotted against the parameter values of the output error weight parameters.

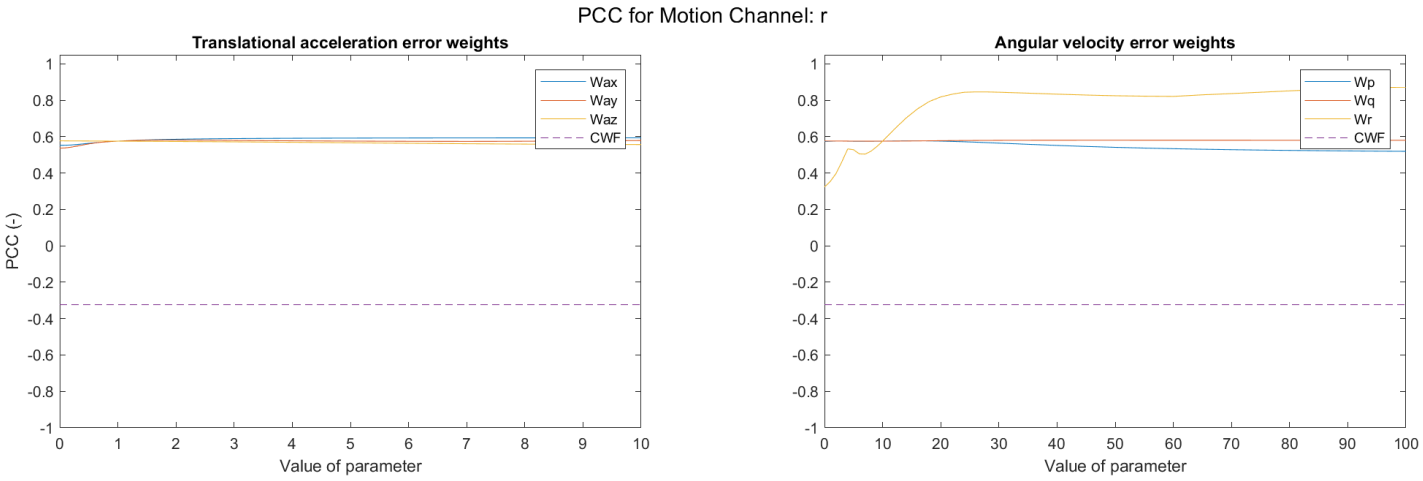


Figure A.12: The PCC between desired and actual r, plotted against the parameter values of the output error weight parameters.

A.2. TWO-WAY SENSITIVITY PLOTS

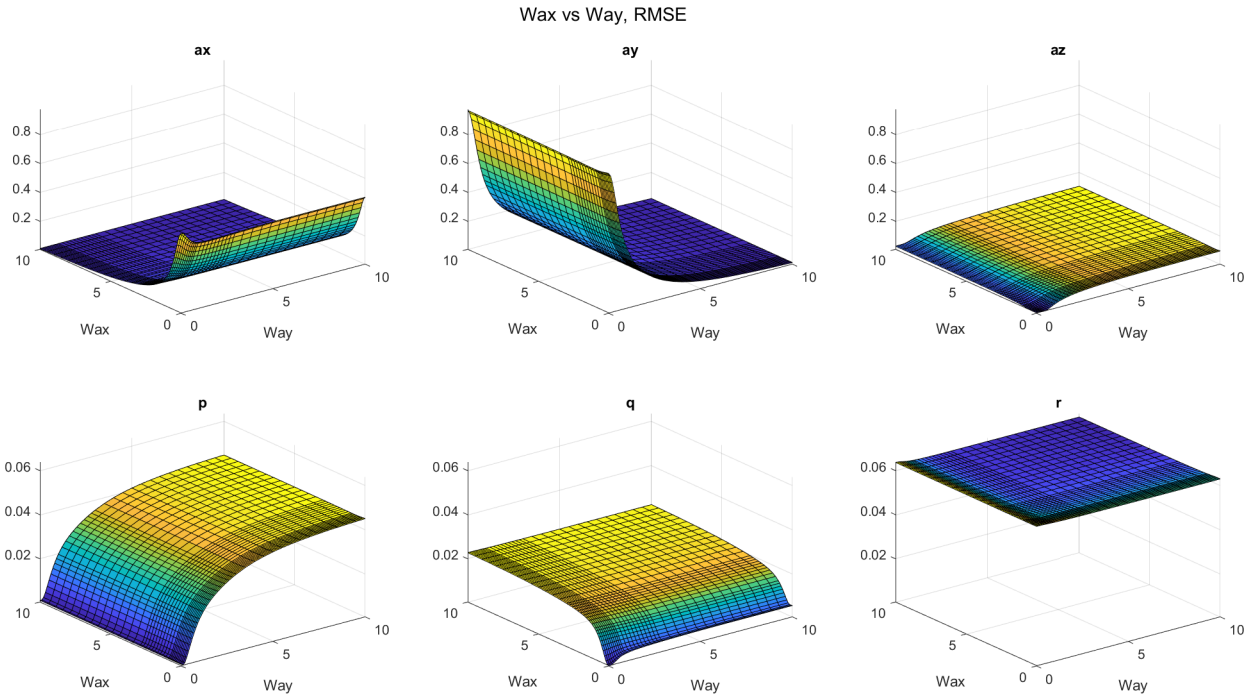


Figure A.13: RMSE when changing Wax and Way simultaneously.

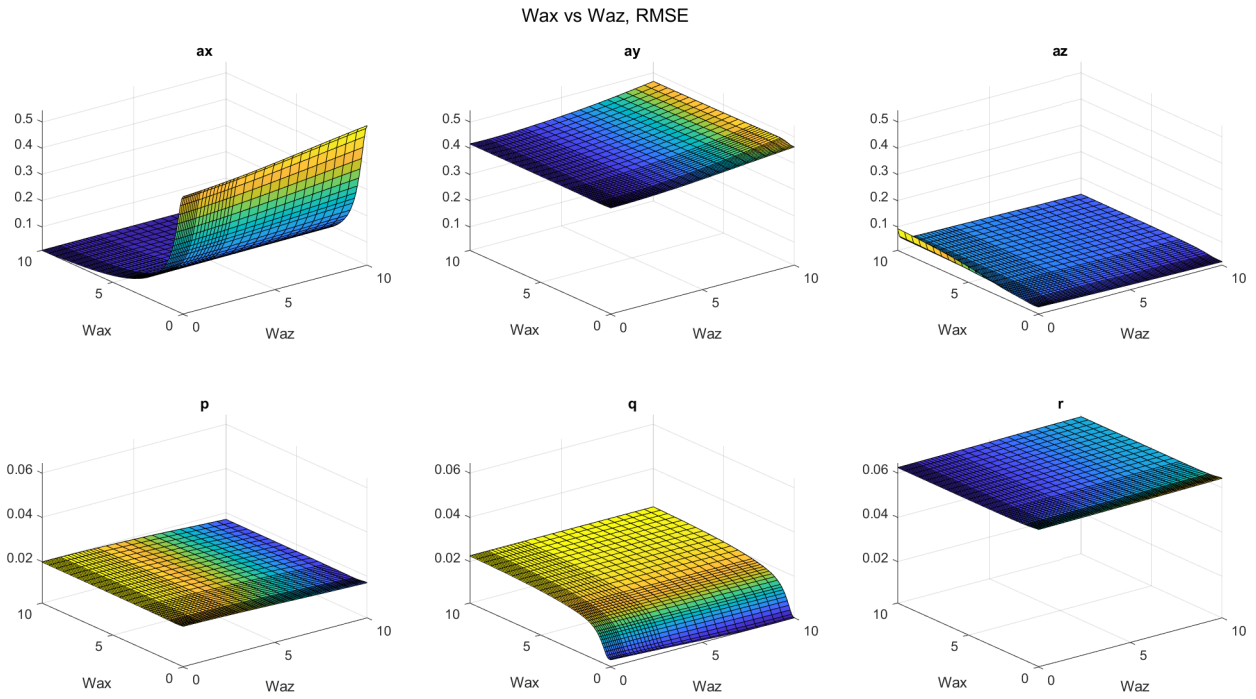


Figure A.14: RMSE when changing Wax and Waz simultaneously.

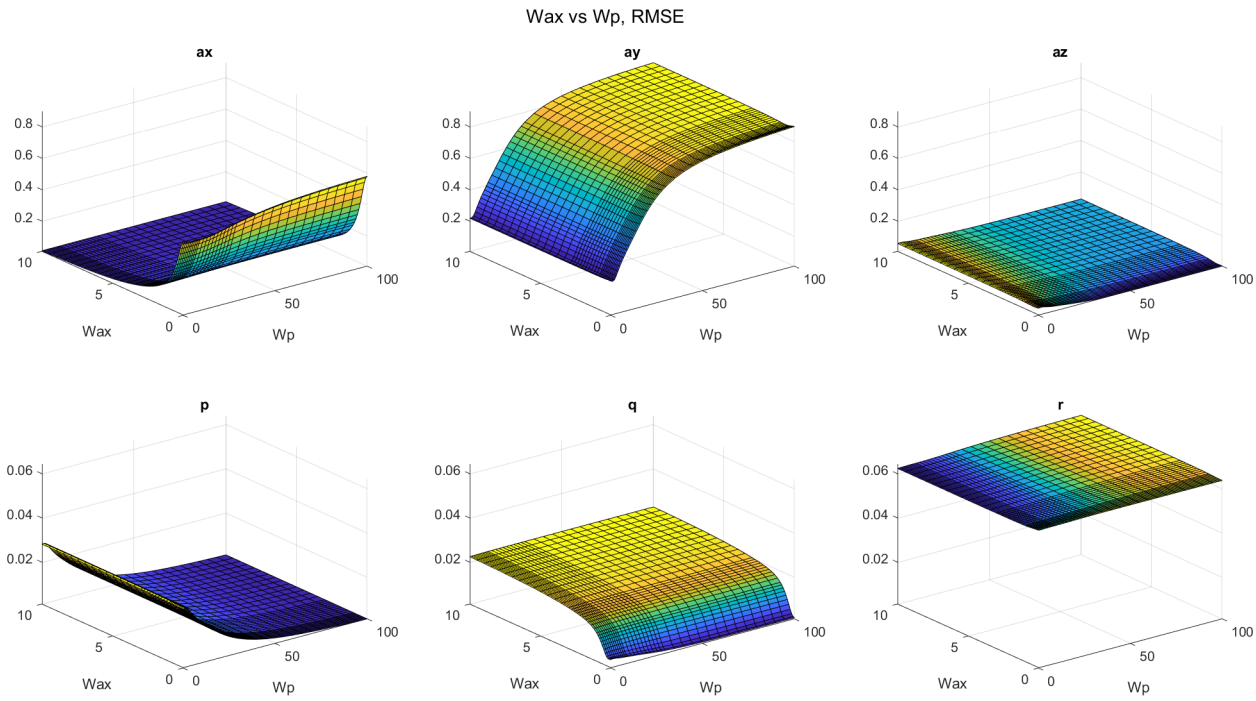


Figure A.15: RMSE when changing Wax and Wp simultaneously.

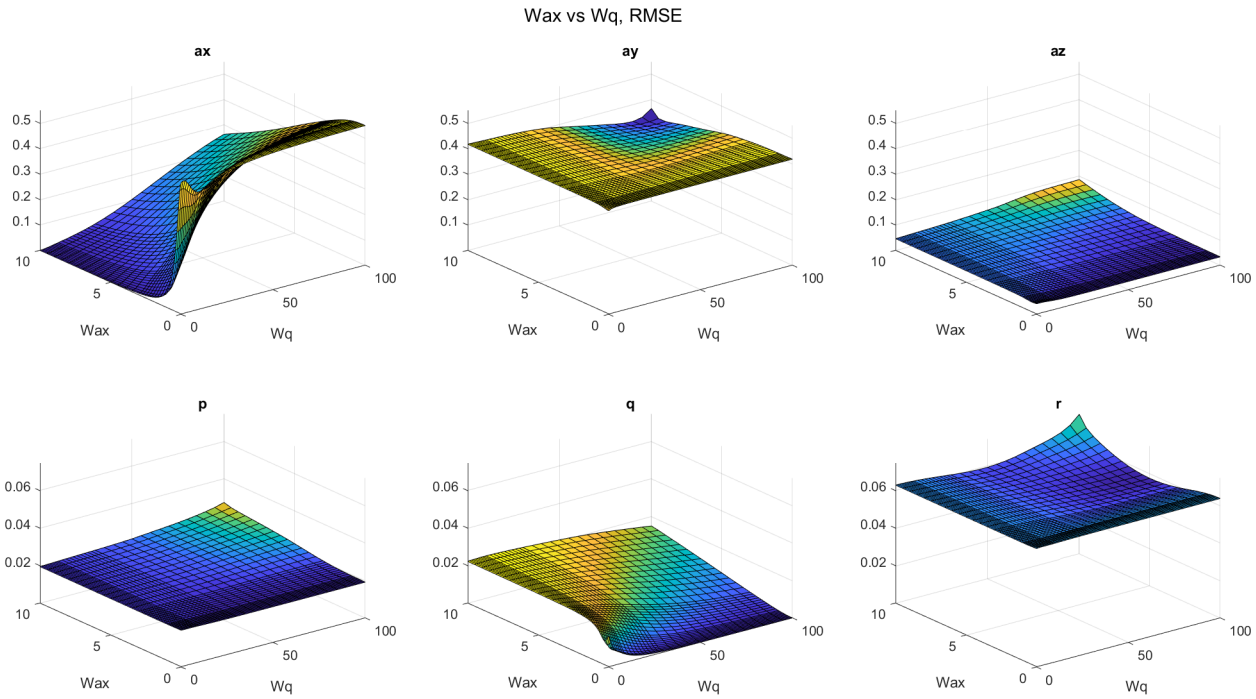


Figure A.16: RMSE when changing Wax and Wq simultaneously.

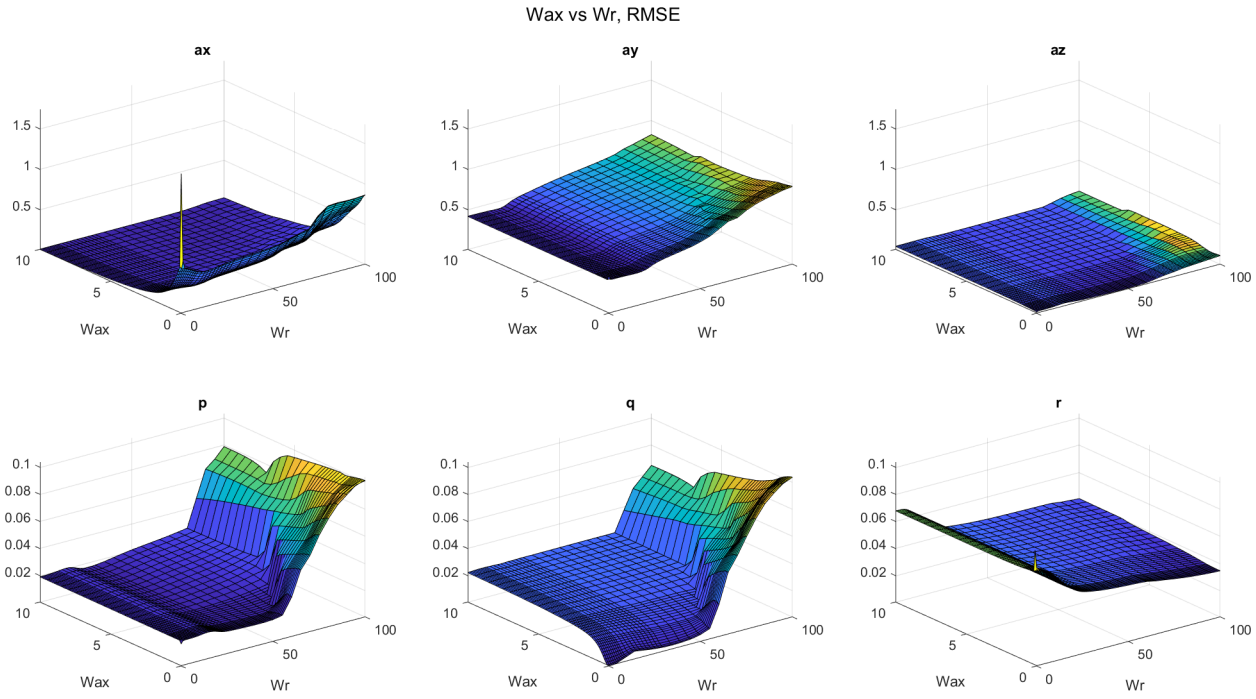


Figure A.17: RMSE when changing Wax and Wr simultaneously.

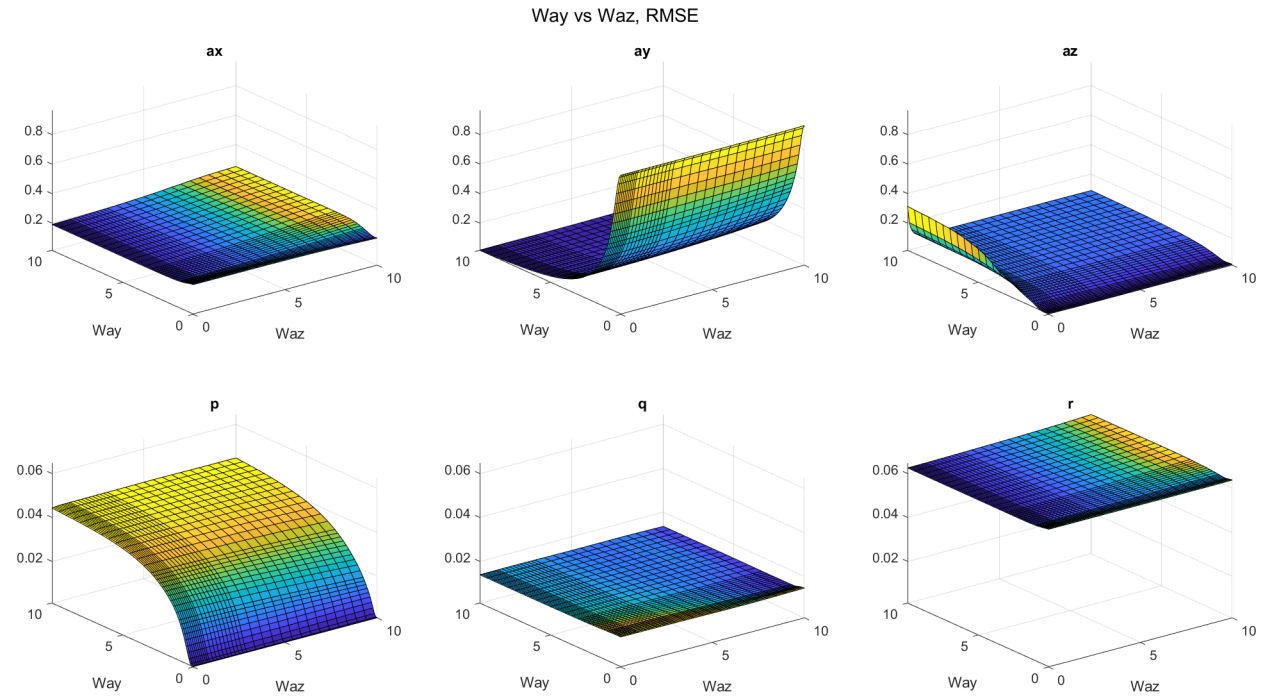


Figure A.18: RMSE when changing Way and Waz simultaneously.

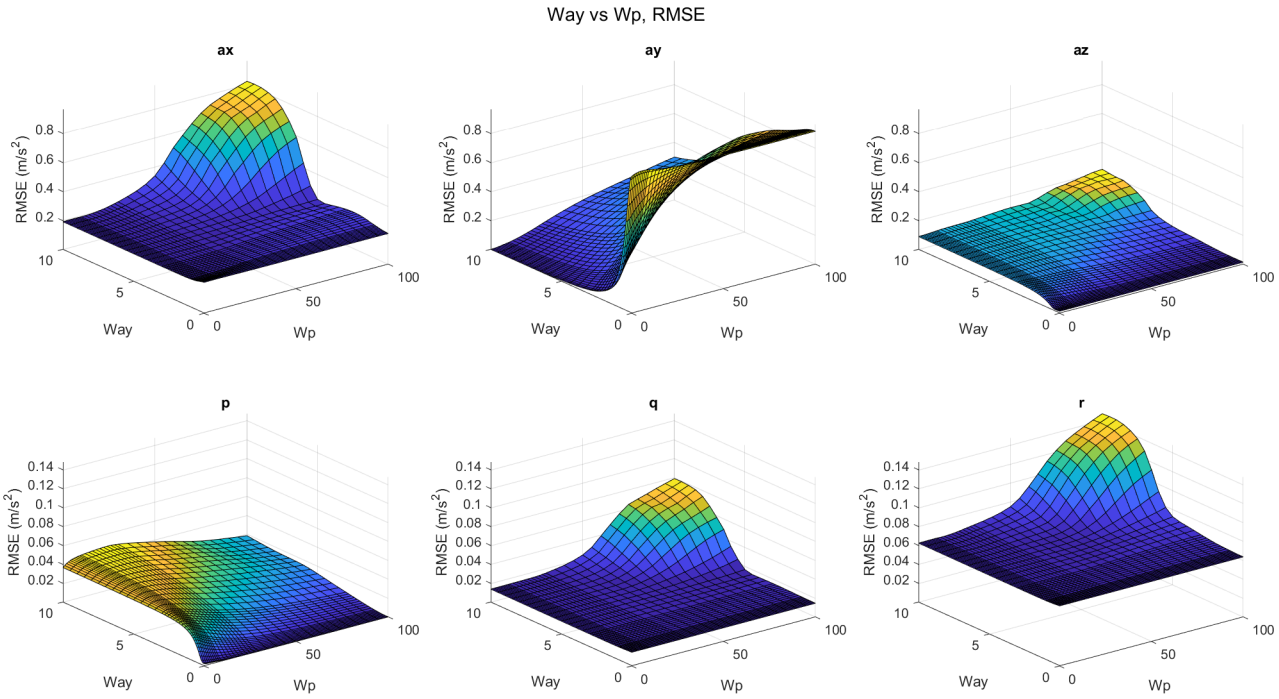


Figure A.19: RMSE when changing Way and Wp simultaneously.

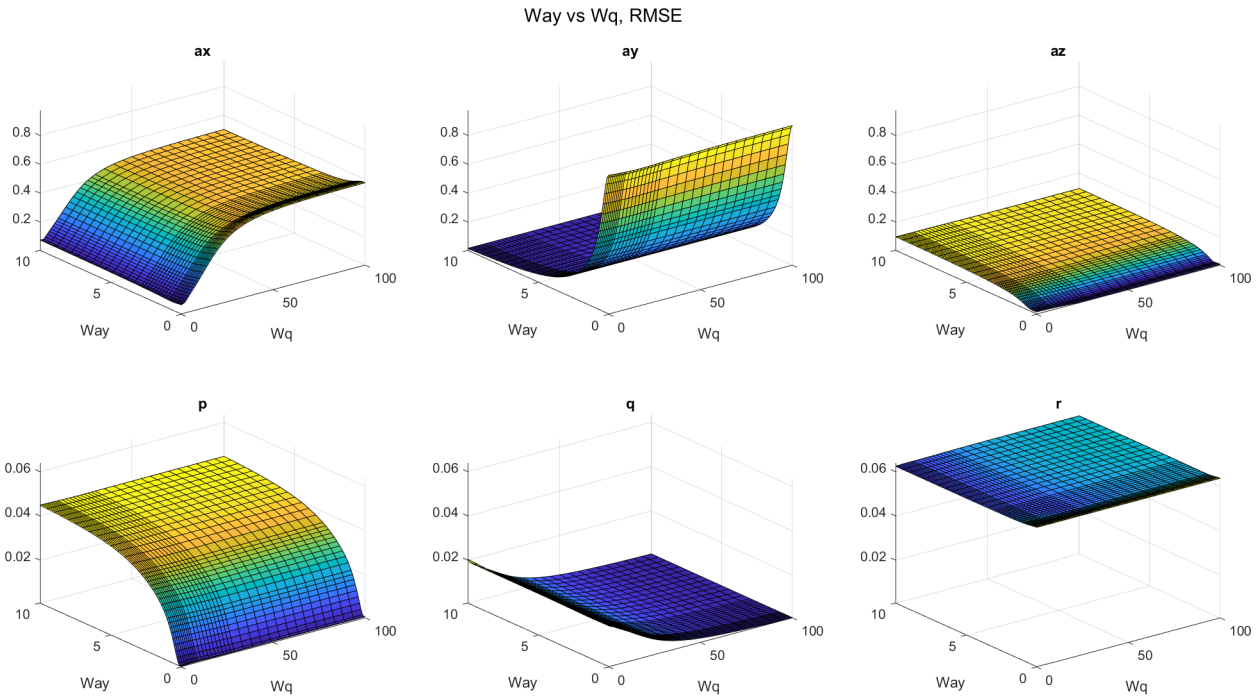


Figure A.20: RMSE when changing Way and Wq simultaneously.

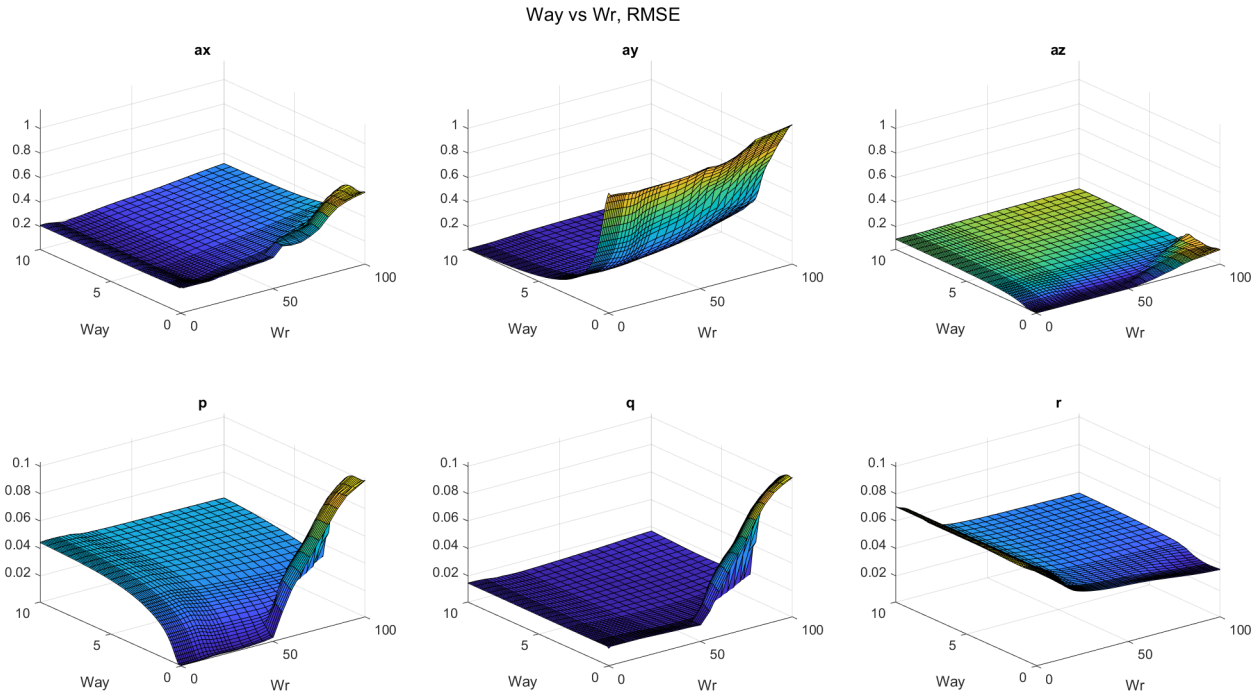


Figure A.21: RMSE when changing Way and Wr simultaneously.

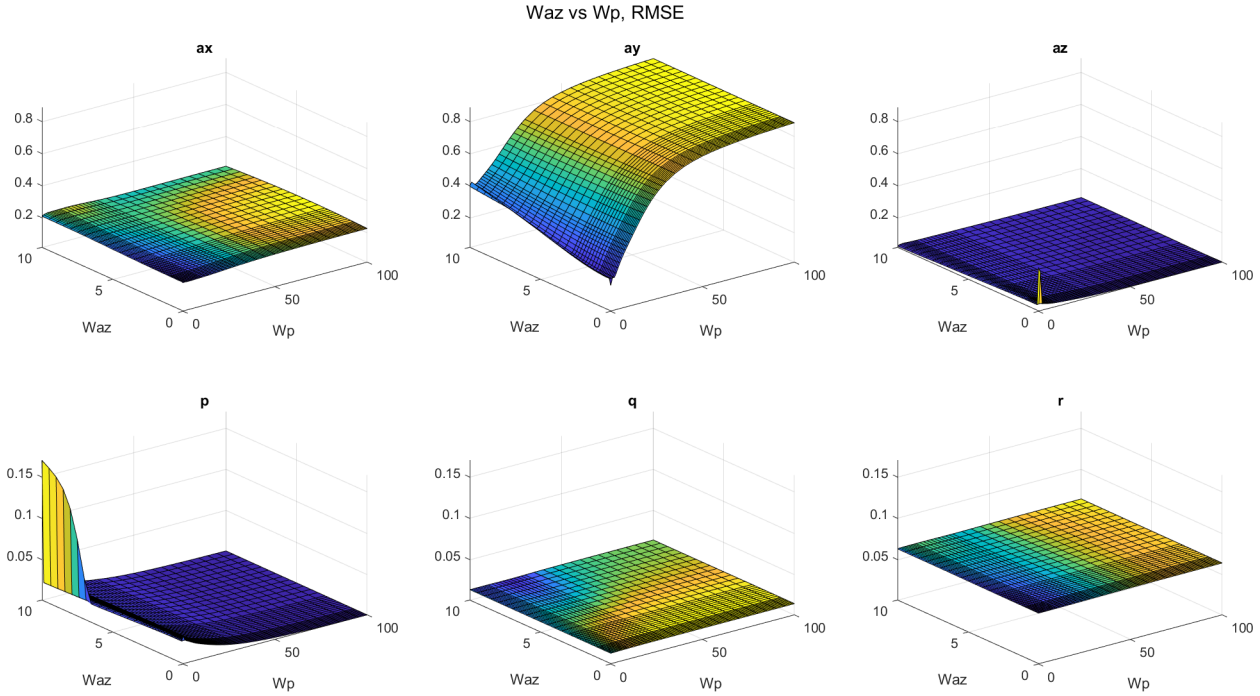


Figure A.22: RMSE when changing Waz and Wp simultaneously.

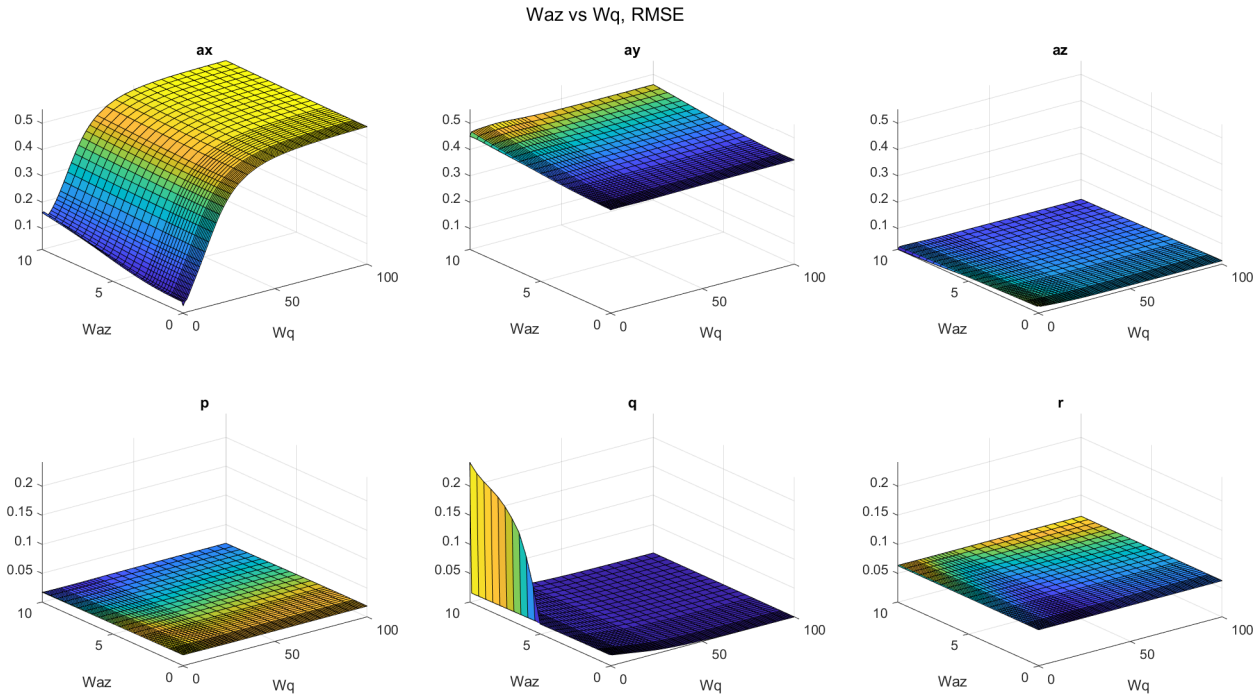


Figure A.23: RMSE when changing Waz and Wq simultaneously.

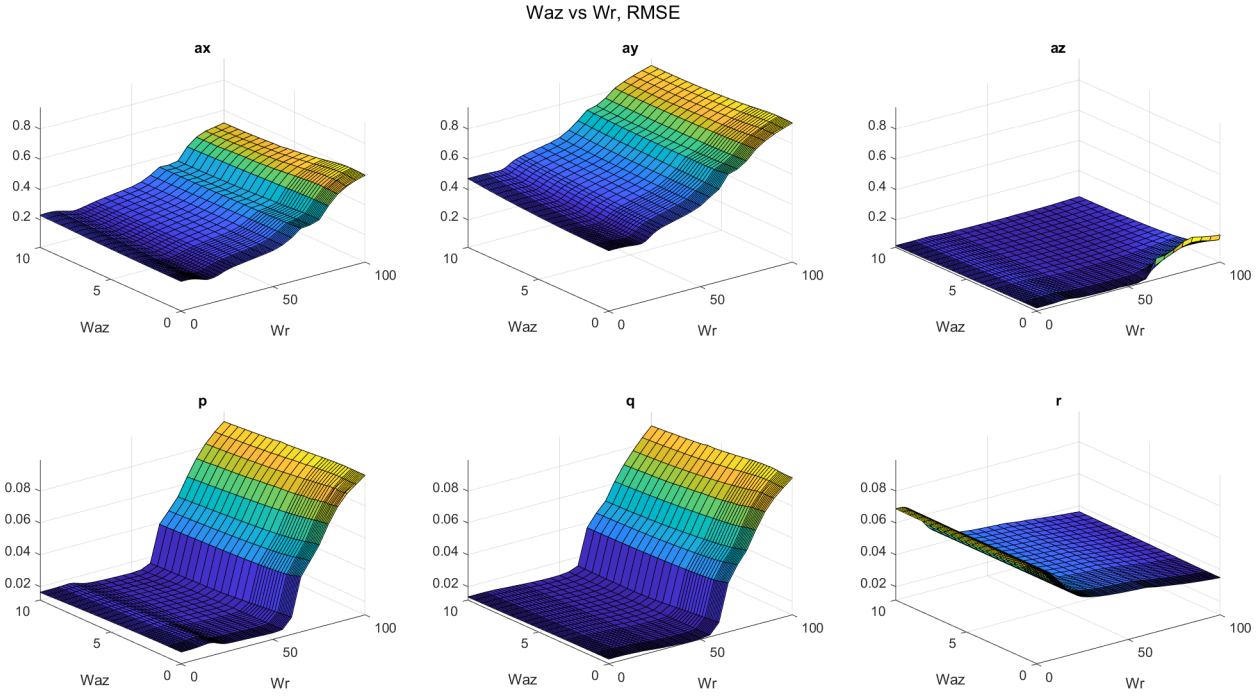


Figure A.24: RMSE when changing Waz and Wr simultaneously.

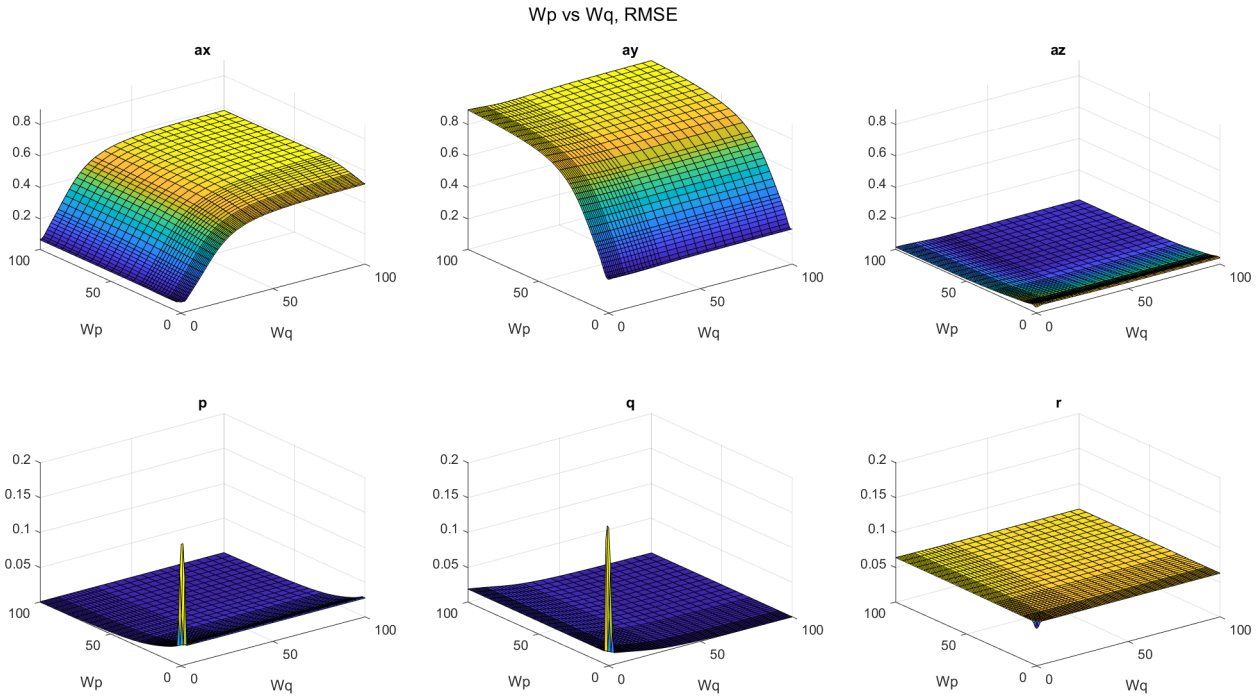


Figure A.25: RMSE when changing Wp and Wq simultaneously.

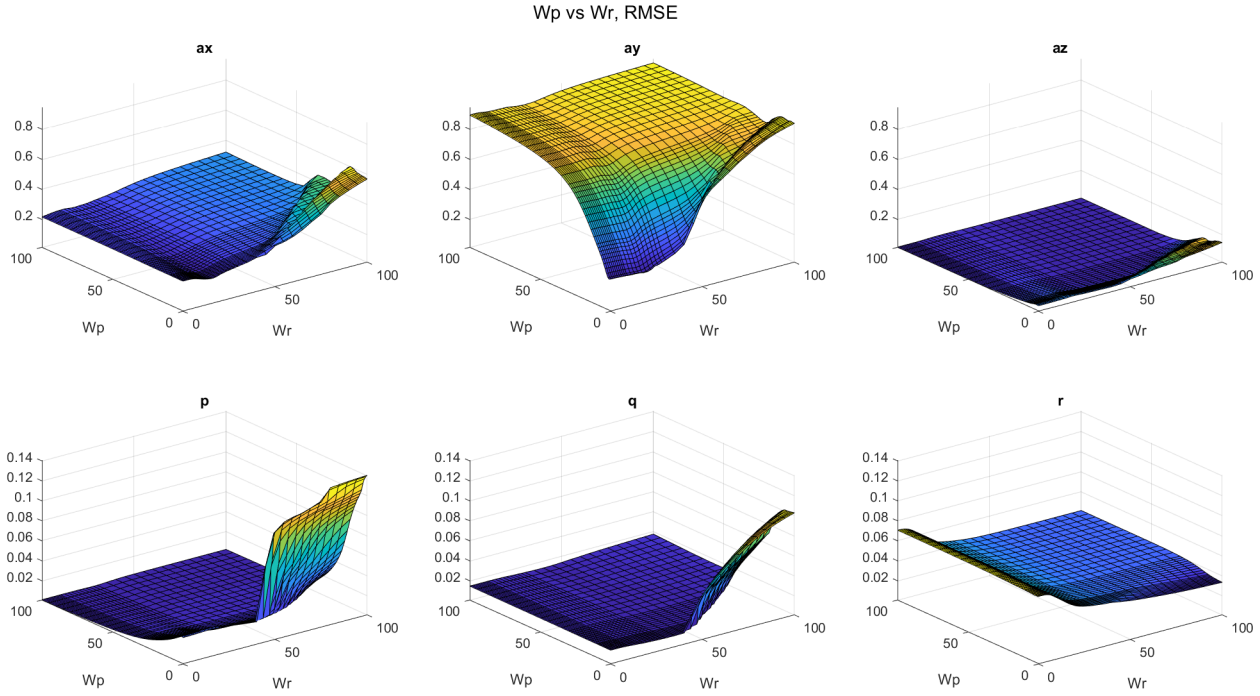


Figure A.26: RMSE when changing Wp and Wr simultaneously.

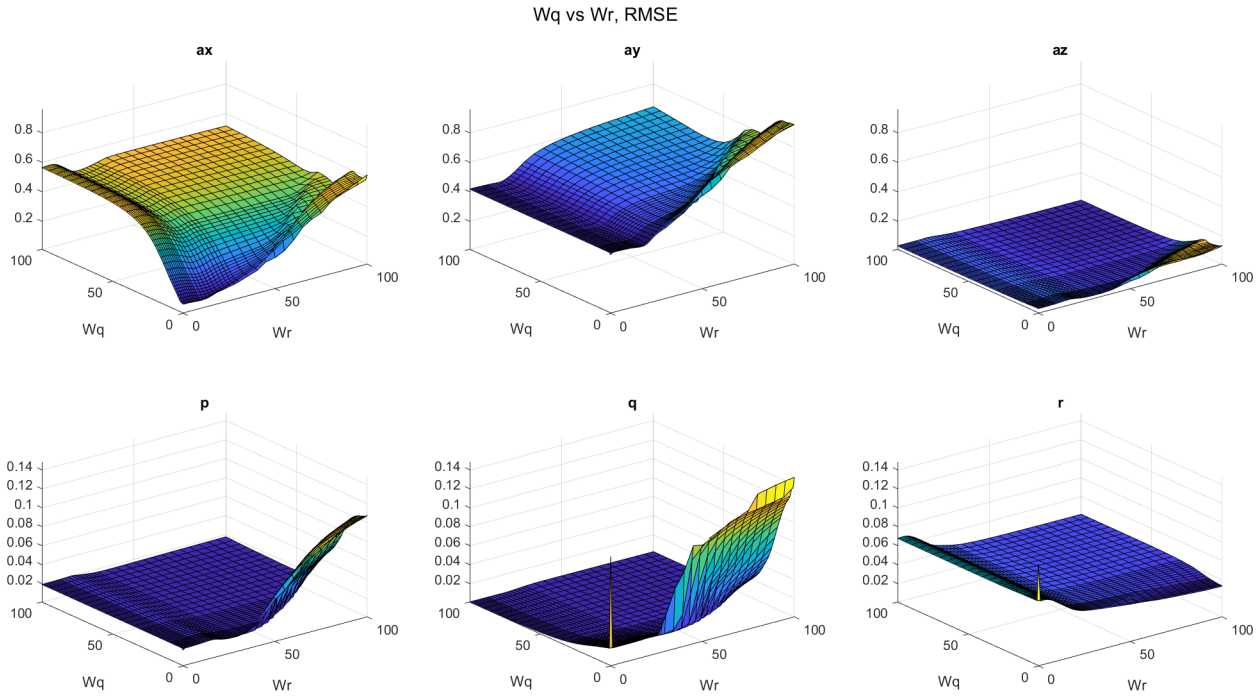


Figure A.27: RMSE when changing Wq and Wr simultaneously.

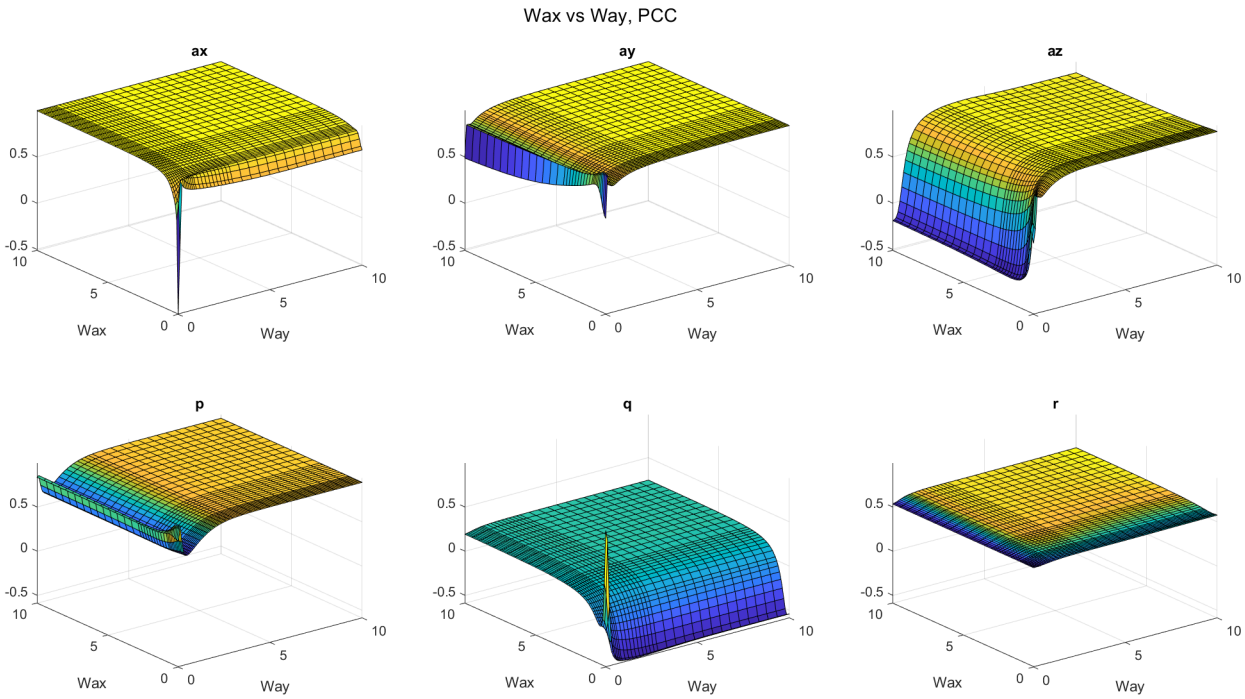


Figure A.28: PCC when changing Wax and Way simultaneously.

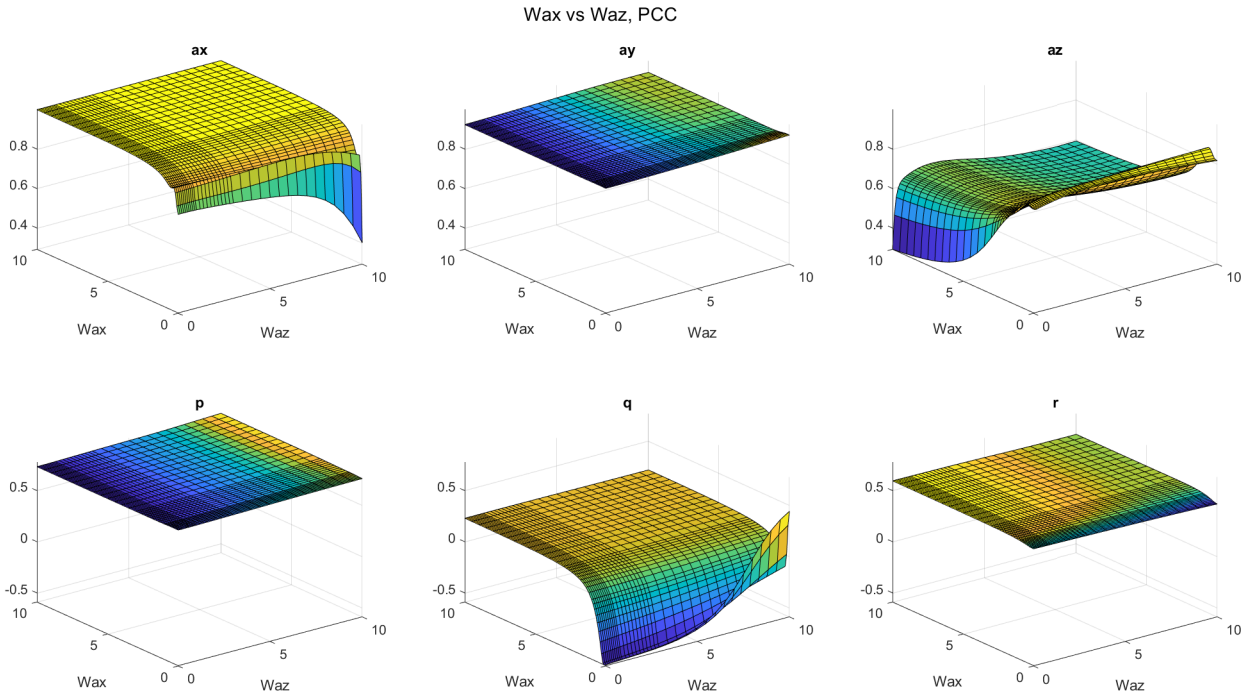


Figure A.29: PCC when changing Wax and Waz simultaneously.

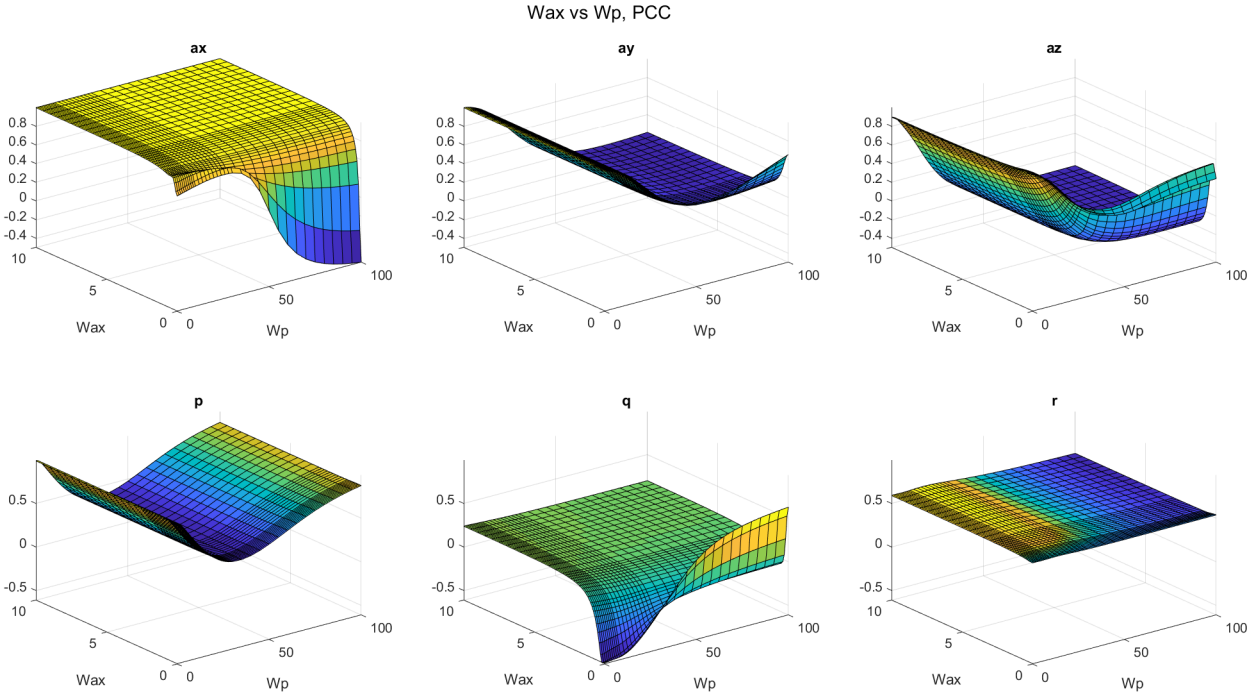


Figure A.30: PCC when changing Wax and Wp simultaneously.

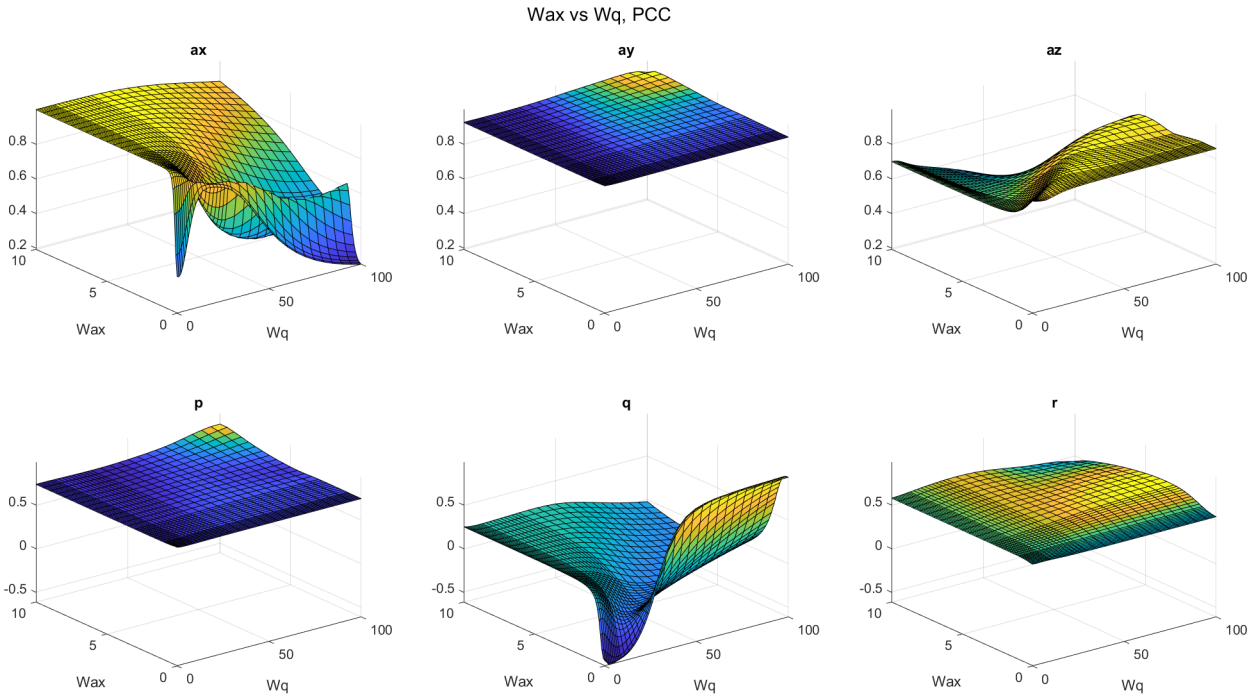


Figure A.31: PCC when changing Wax and Wq simultaneously.

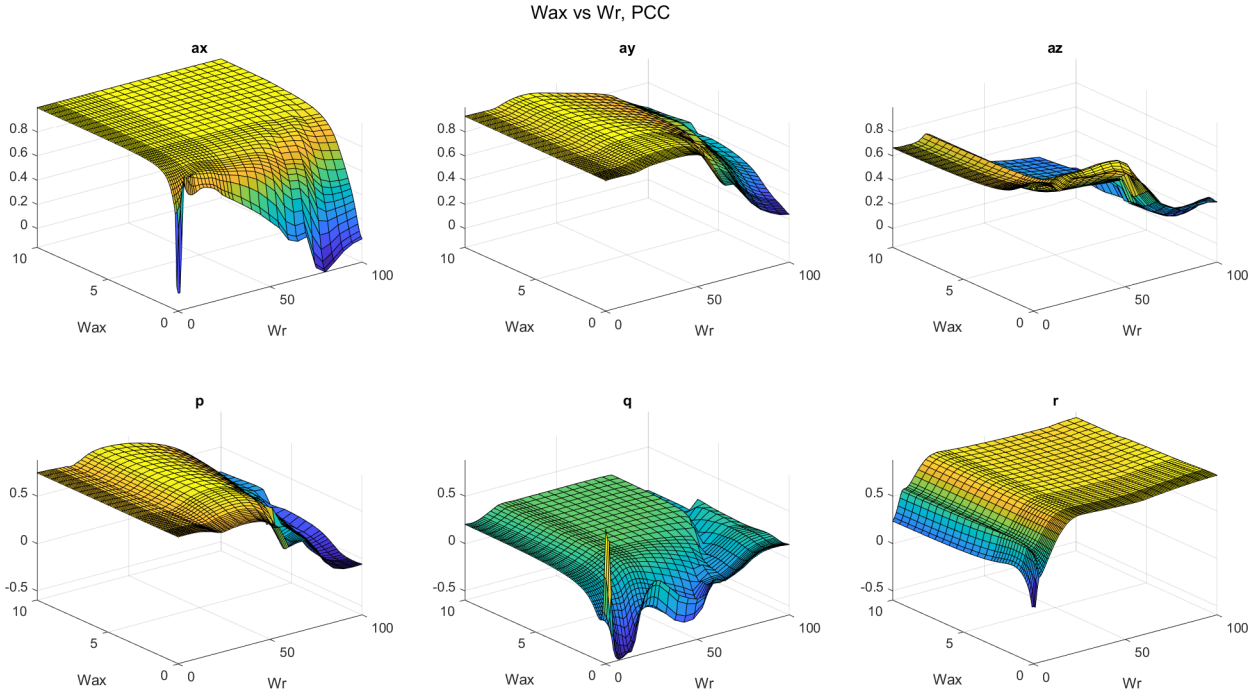


Figure A.32: PCC when changing Wax and Wr simultaneously.

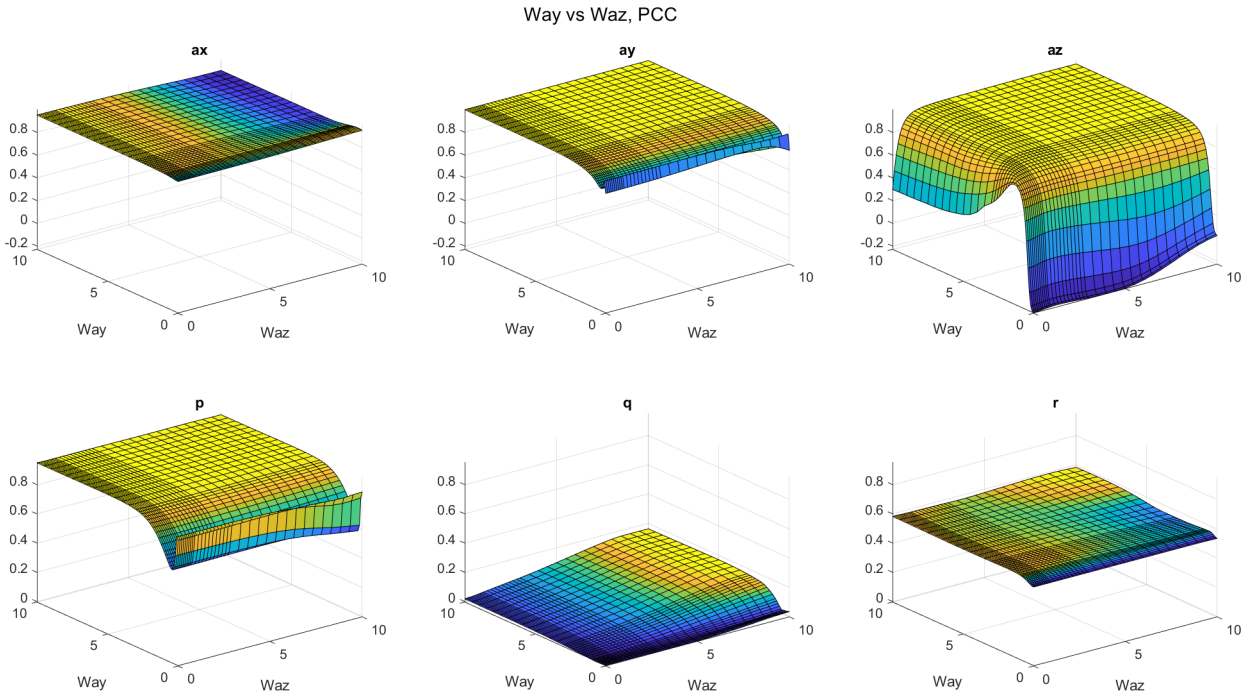


Figure A.33: PCC when changing Way and Waz simultaneously.

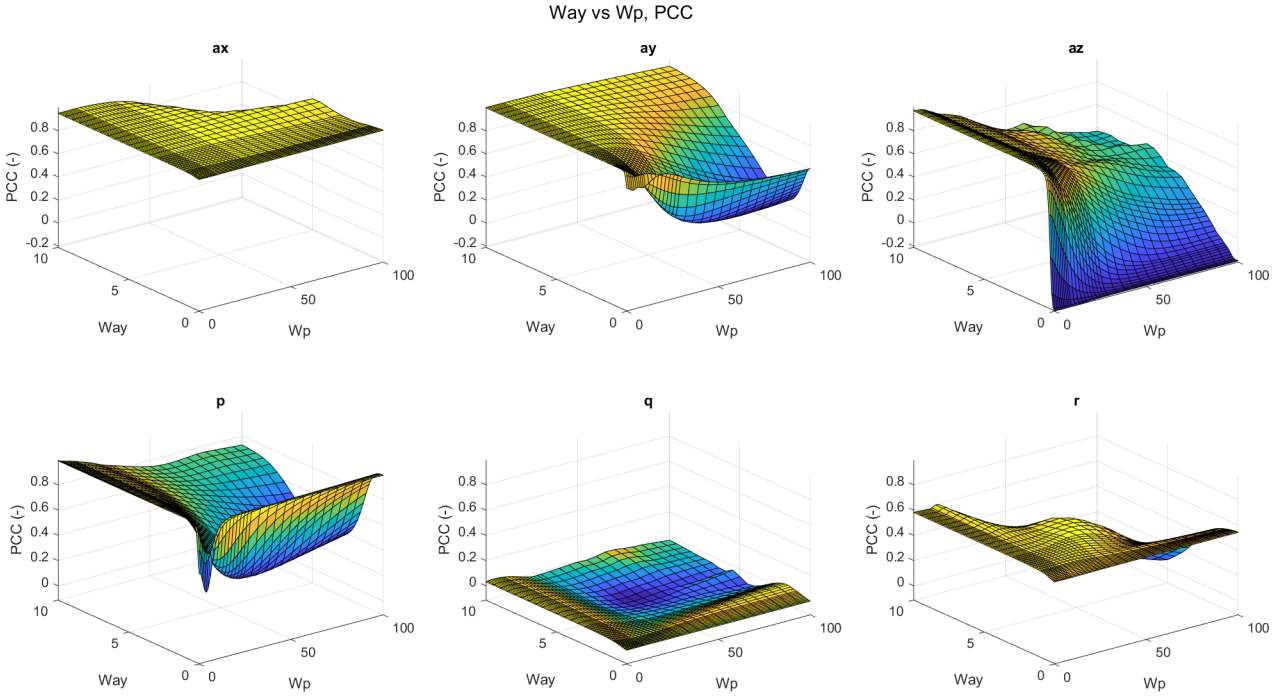


Figure A.34: PCC when changing Way and Wp simultaneously.

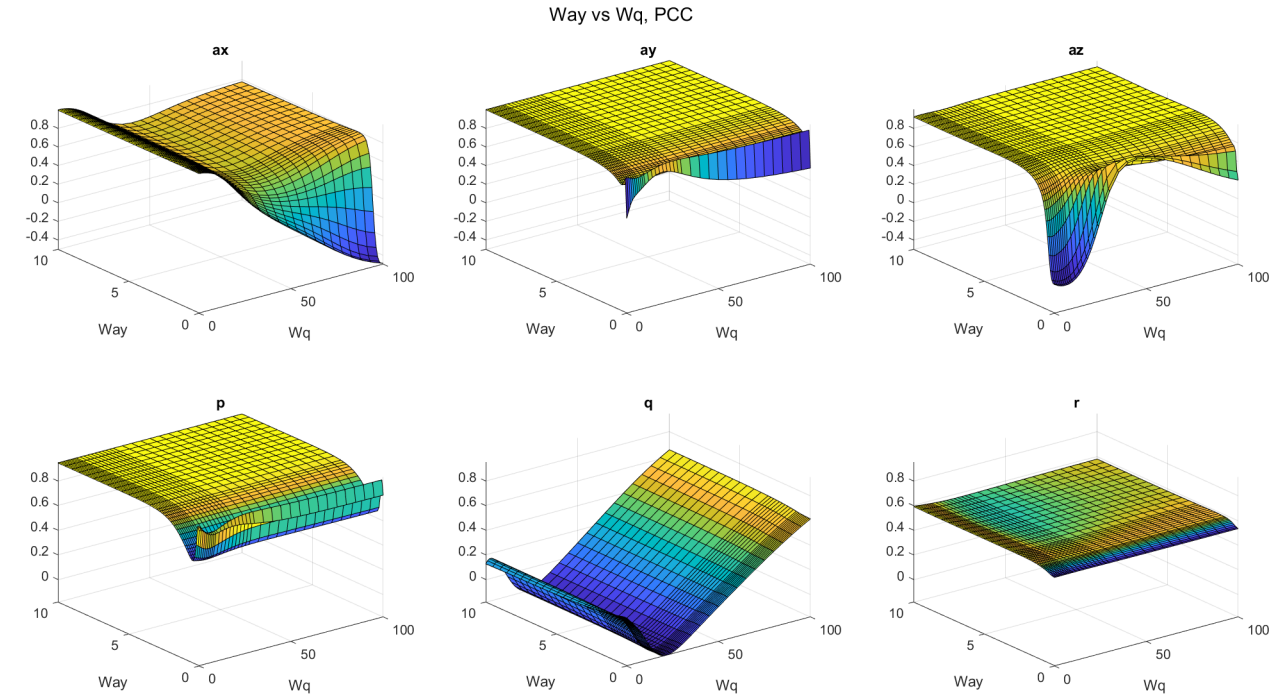


Figure A.35: PCC when changing Way and Wq simultaneously.

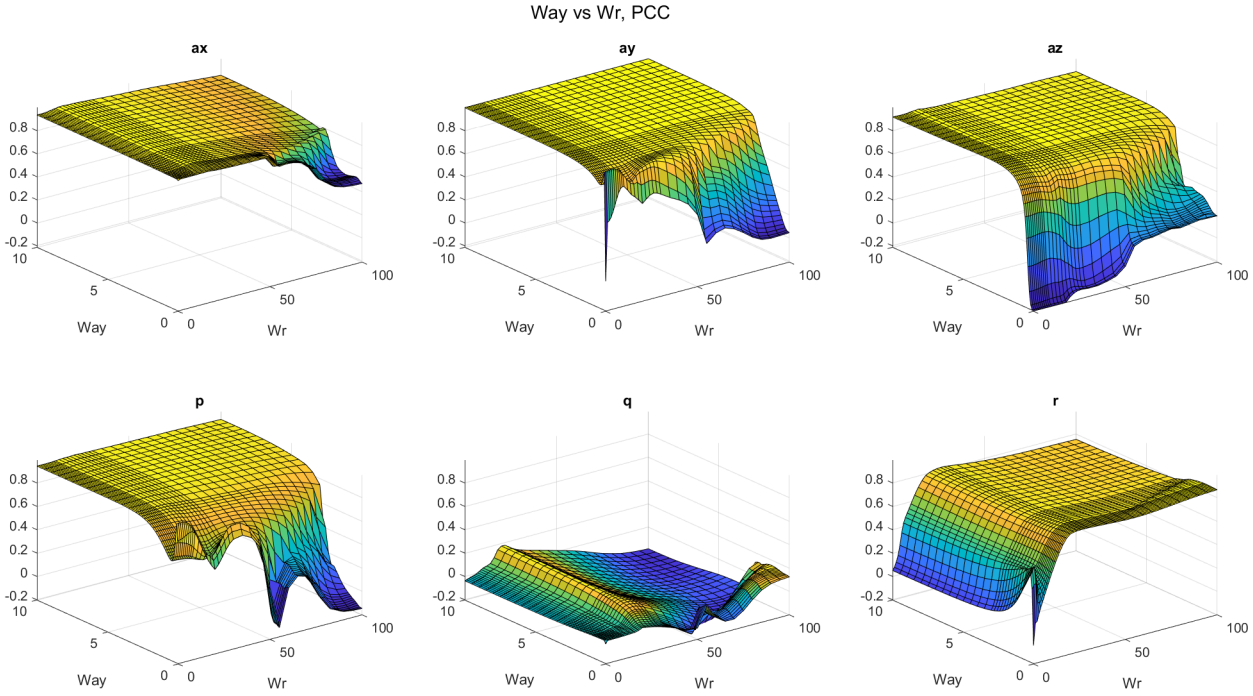


Figure A.36: PCC when changing Way and Wr simultaneously.

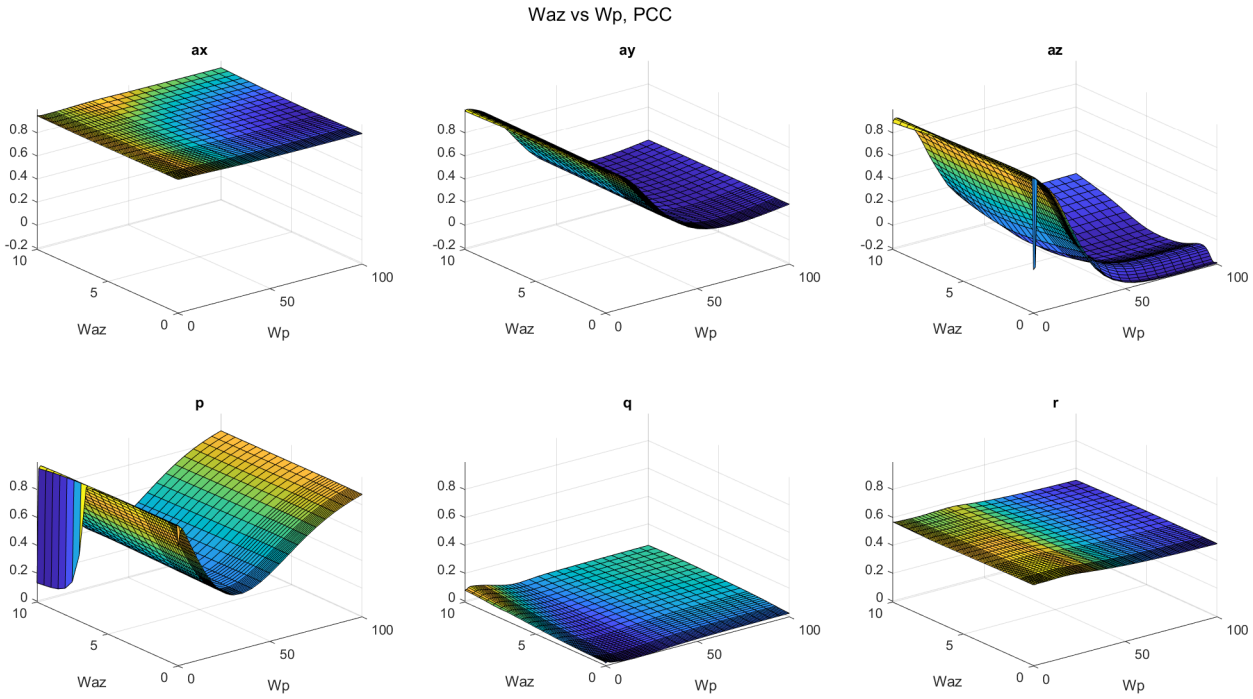


Figure A.37: PCC when changing Waz and Wp simultaneously.

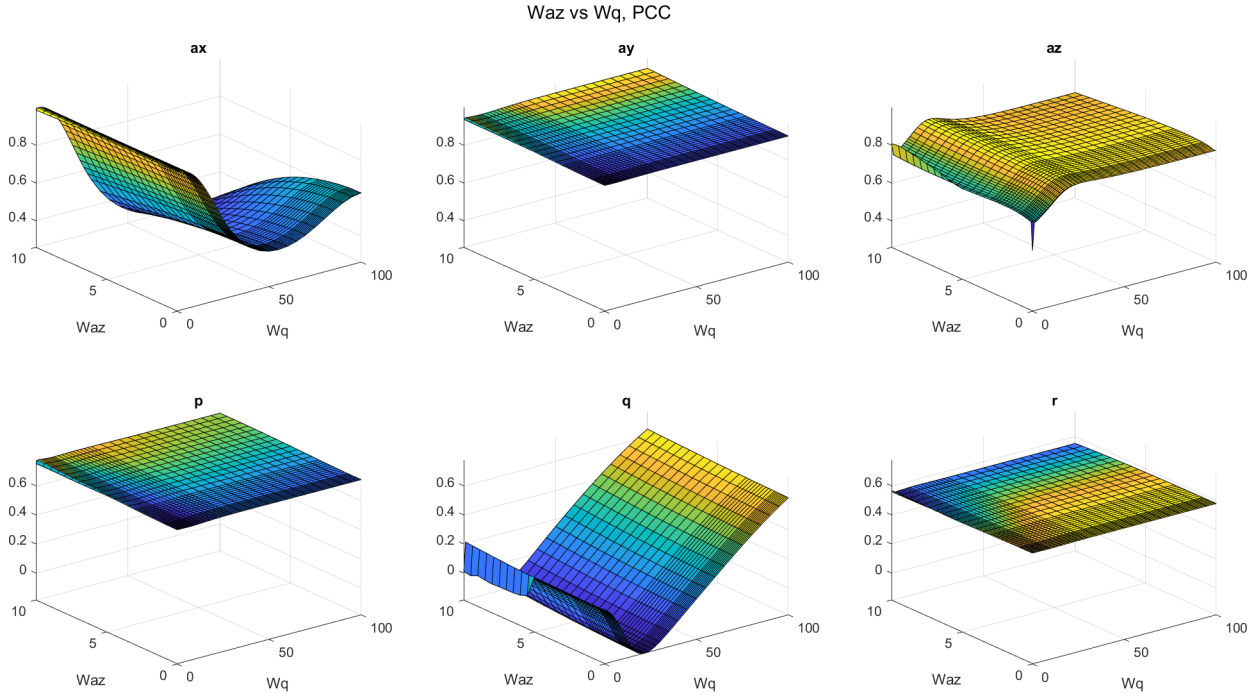


Figure A.38: PCC when changing Waz and Wq simultaneously.

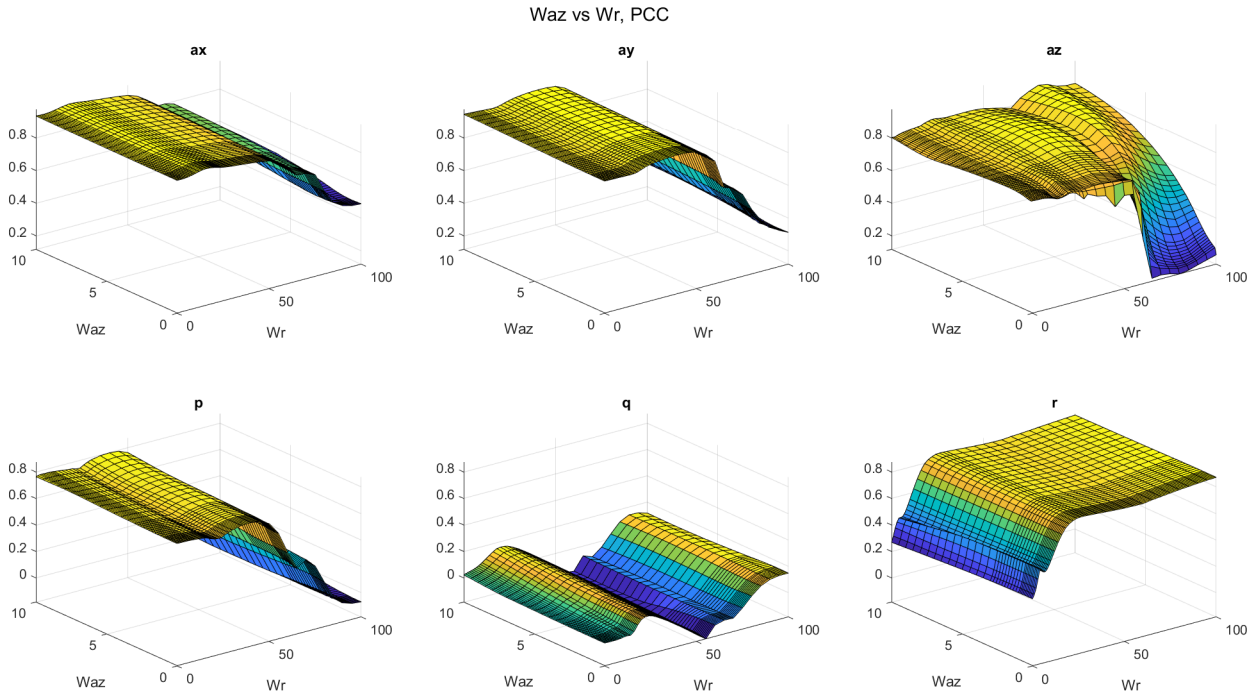


Figure A.39: PCC when changing Waz and Wr simultaneously.

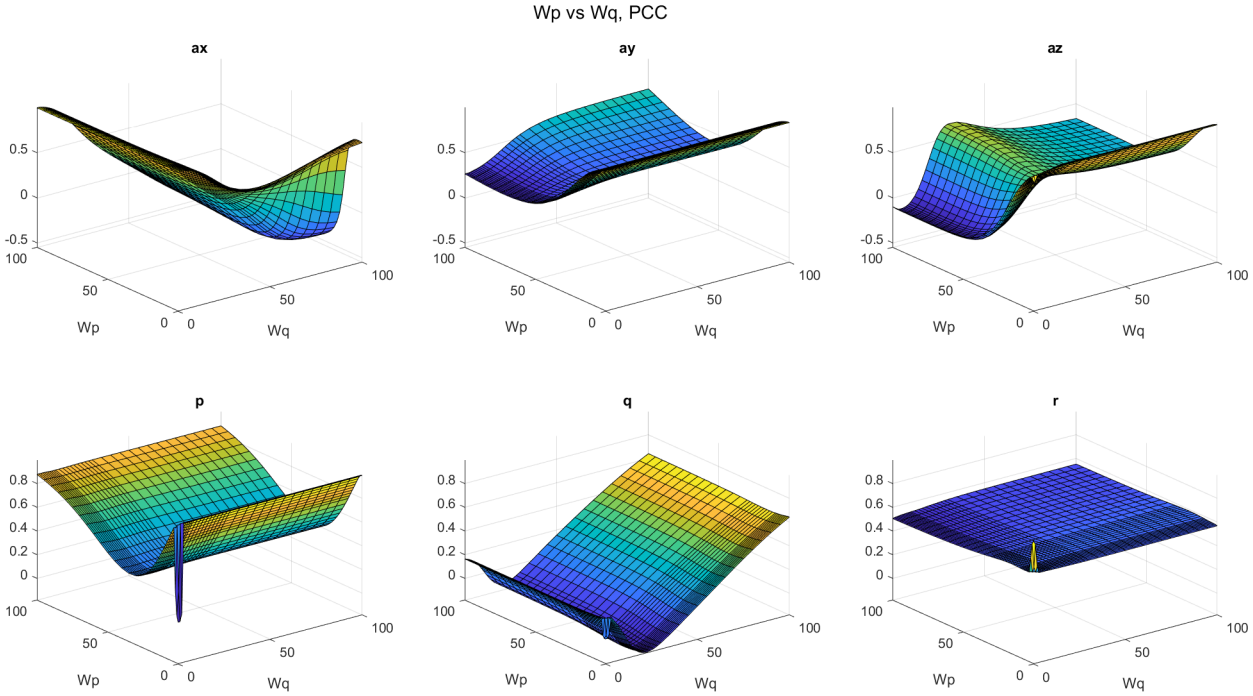


Figure A.40: PCC when changing Wp and Wq simultaneously.

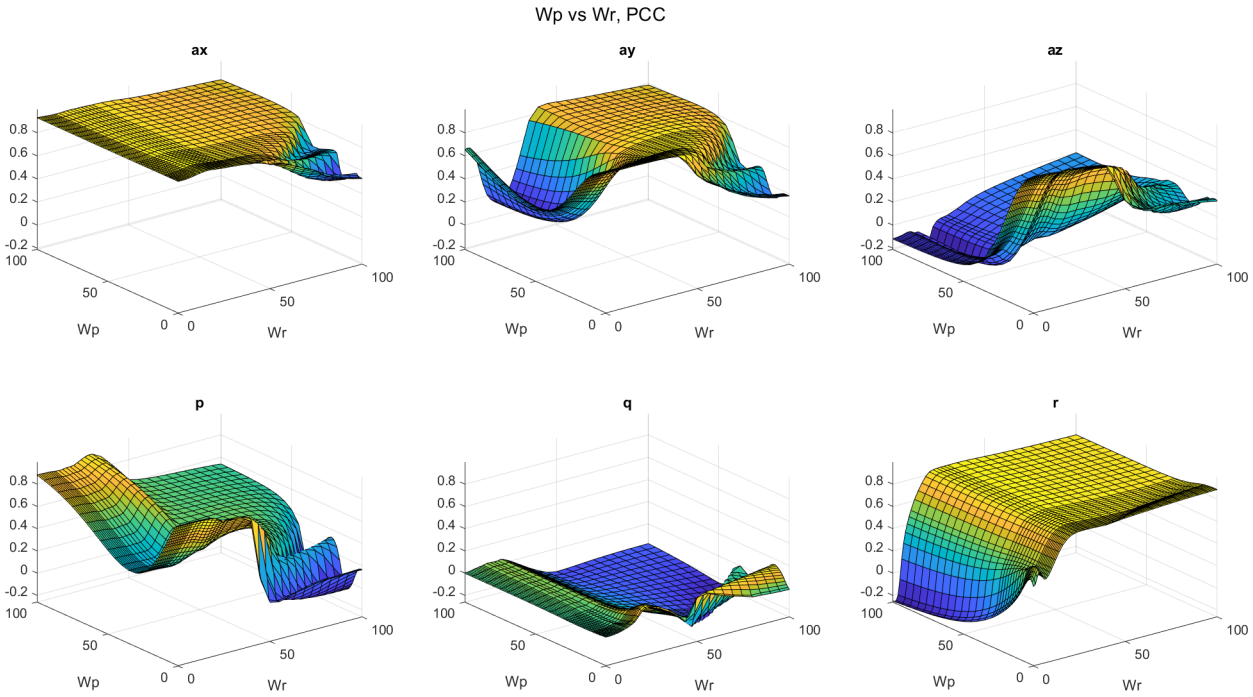


Figure A.41: PCC when changing Wp and Wr simultaneously.

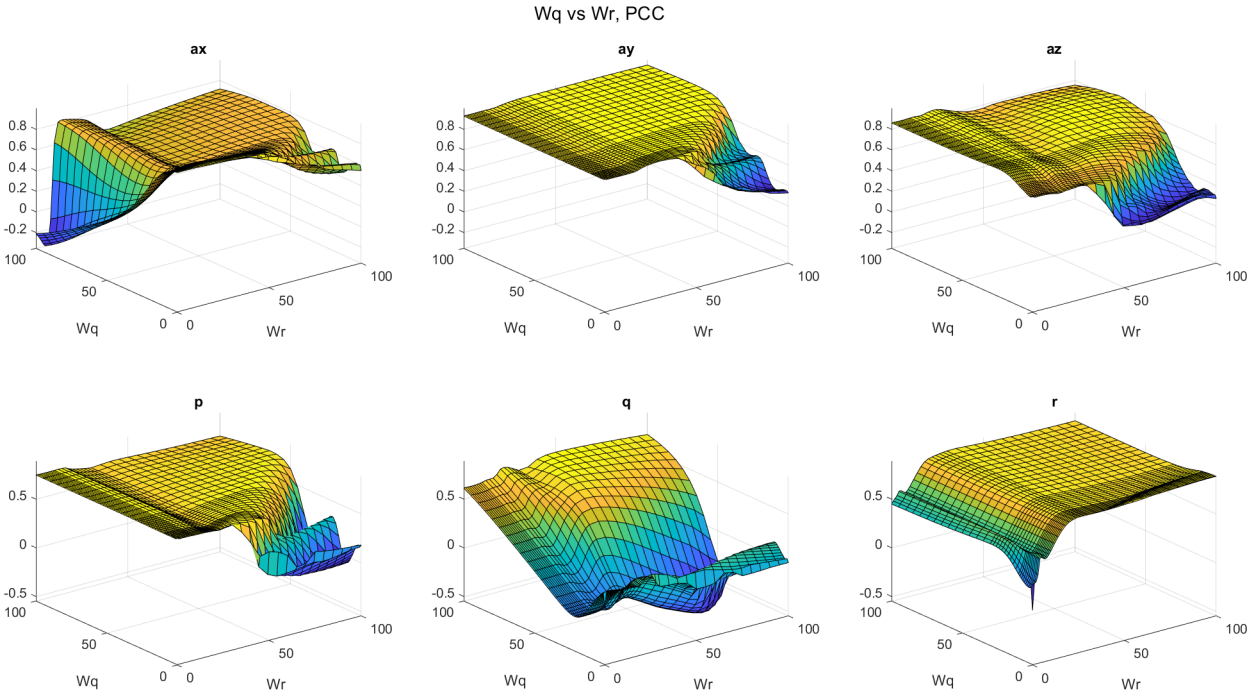


Figure A.42: PCC when changing Wq and Wr simultaneously.

B

Motion Plots for All Conditions Including IMU Data

All experiment conditions had different parameters values, causing the simulator output to be different for each condition. The traces of the simulator outputs for each condition are shown in this Appendix. Also the desired motion is added, which is what the simulator is trying to achieve. And on top of that, the IMU data is added to show that the simulator actually did what was calculated.

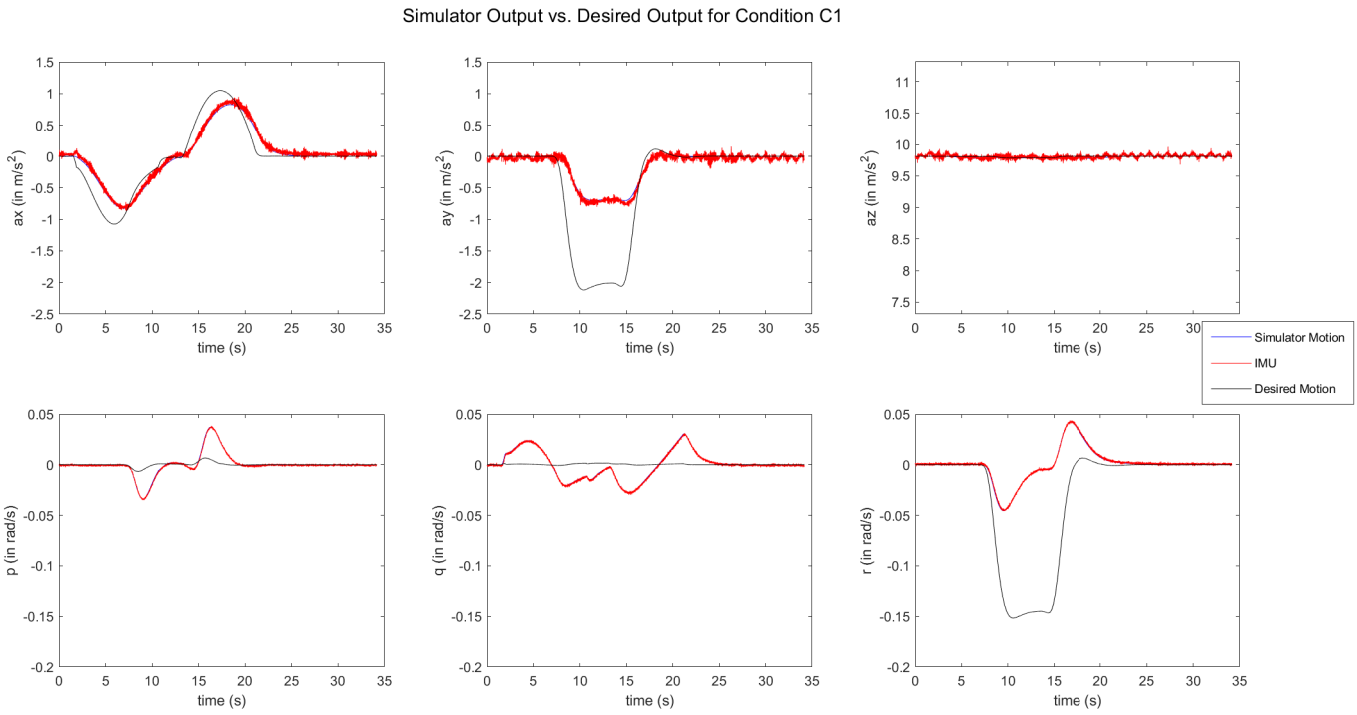


Figure B.1: Simulator output for condition C1, including the desired output and the measured IMU data.

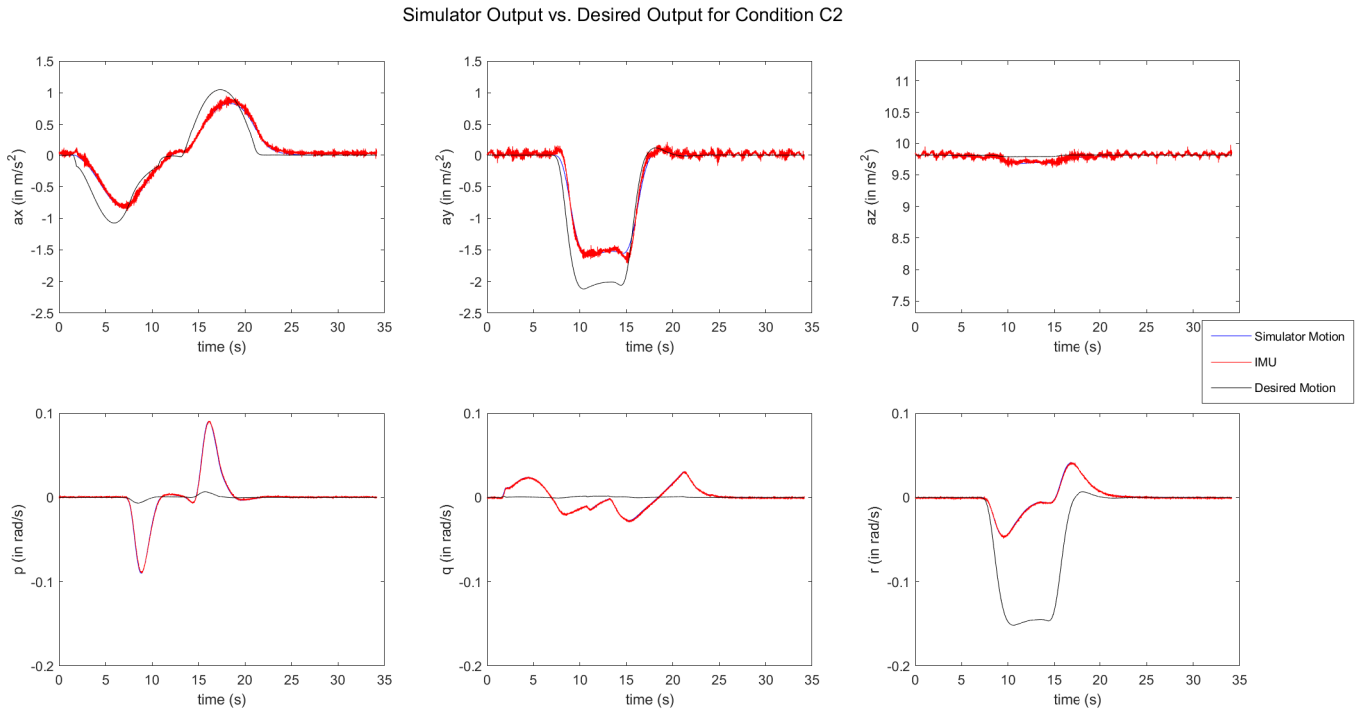


Figure B.2: Simulator output for condition C2, including the desired output and the measured IMU data.

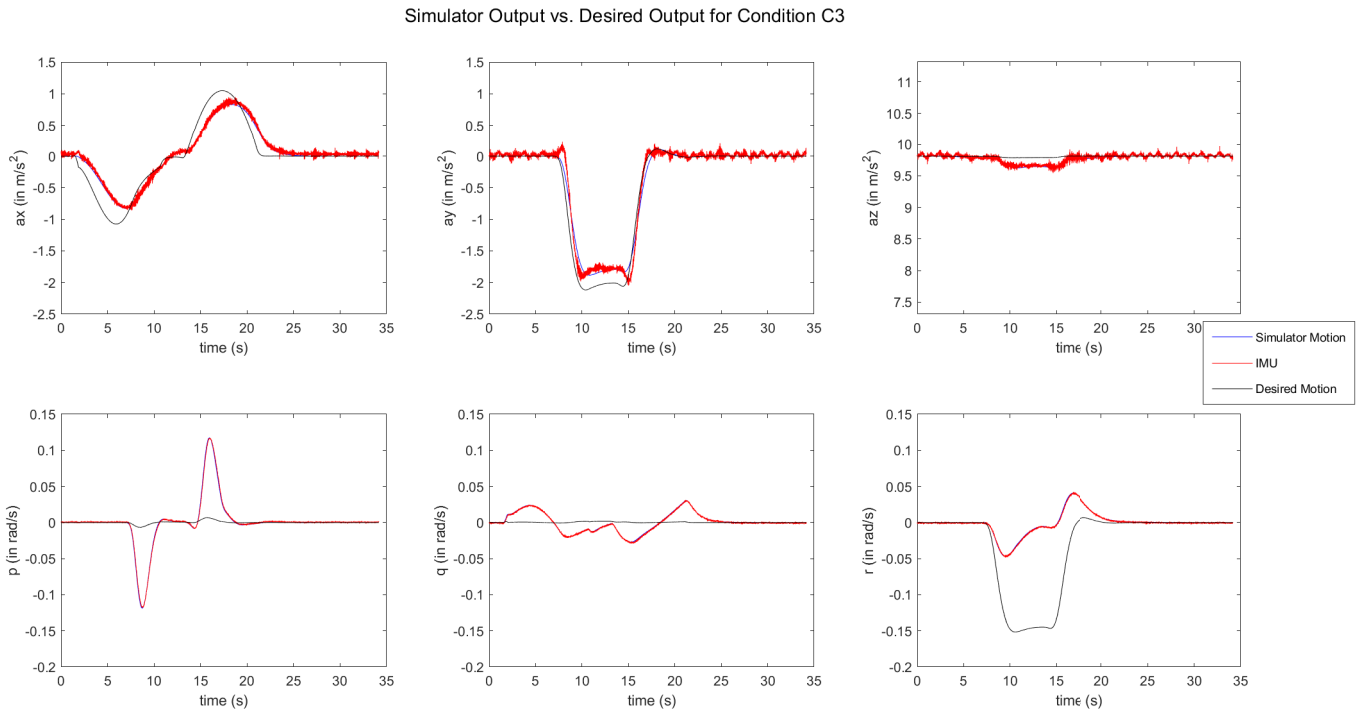


Figure B.3: Simulator output for condition C3, including the desired output and the measured IMU data.

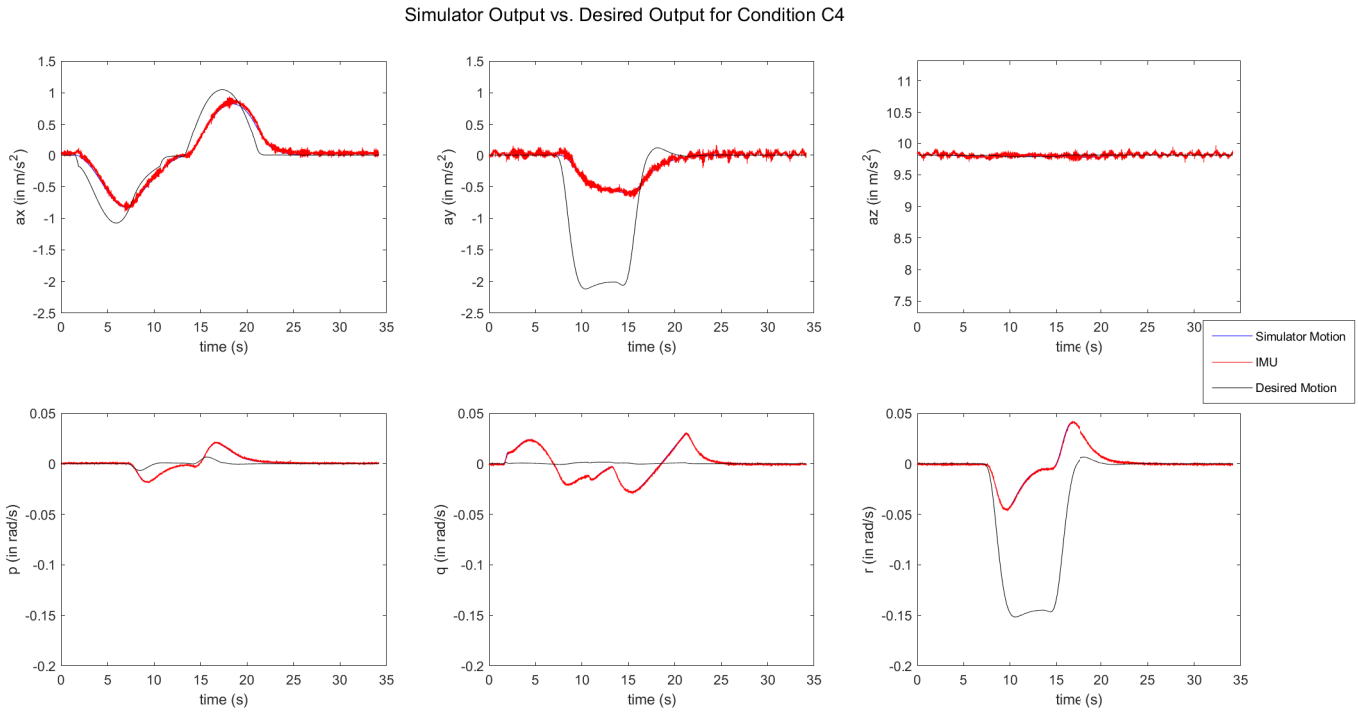


Figure B.4: Simulator output for condition C4, including the desired output and the measured IMU data.

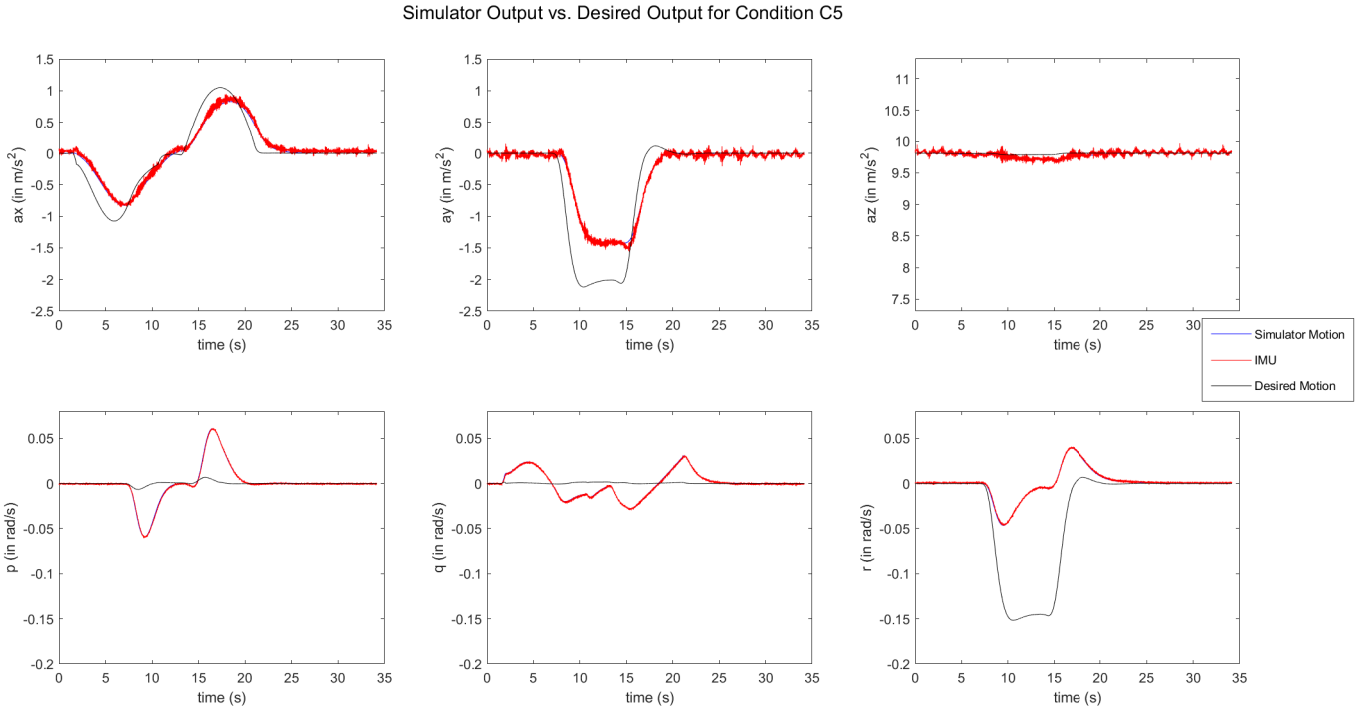


Figure B.5: Simulator output for condition C5, including the desired output and the measured IMU data.

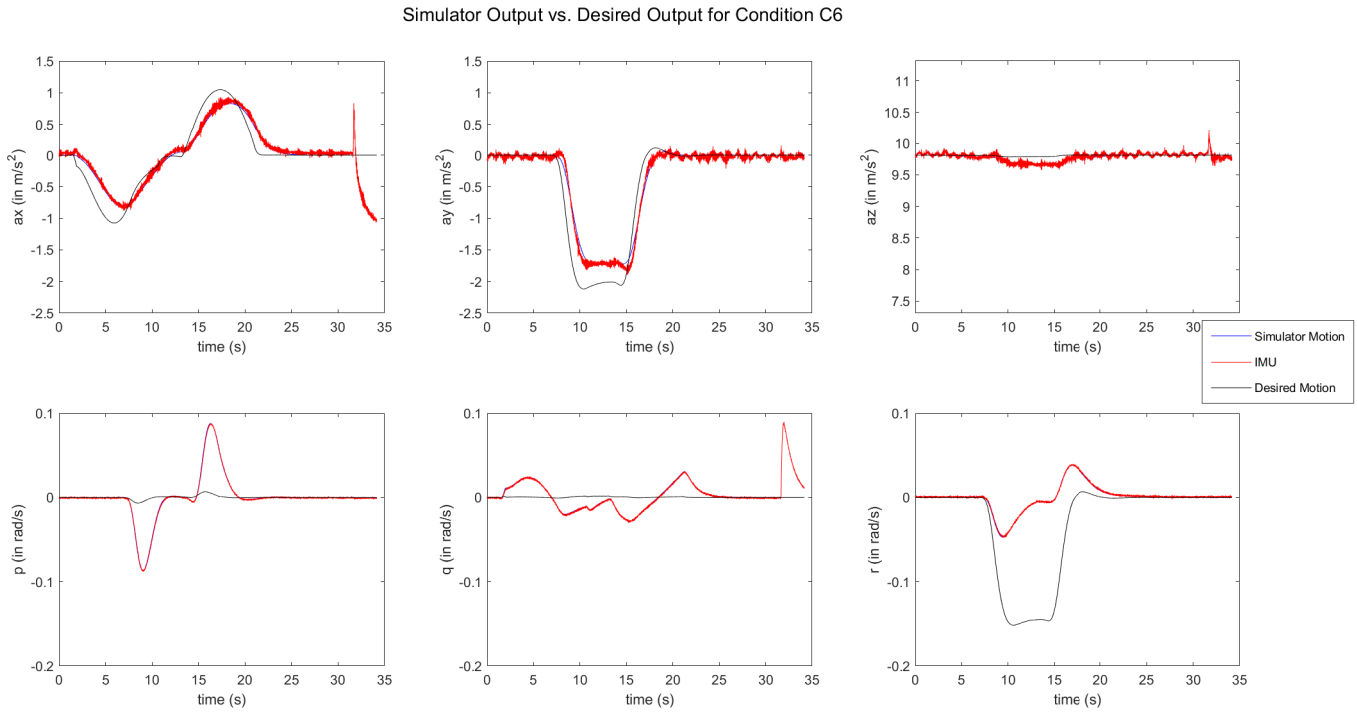


Figure B.6: Simulator output for condition C6, including the desired output and the measured IMU data.

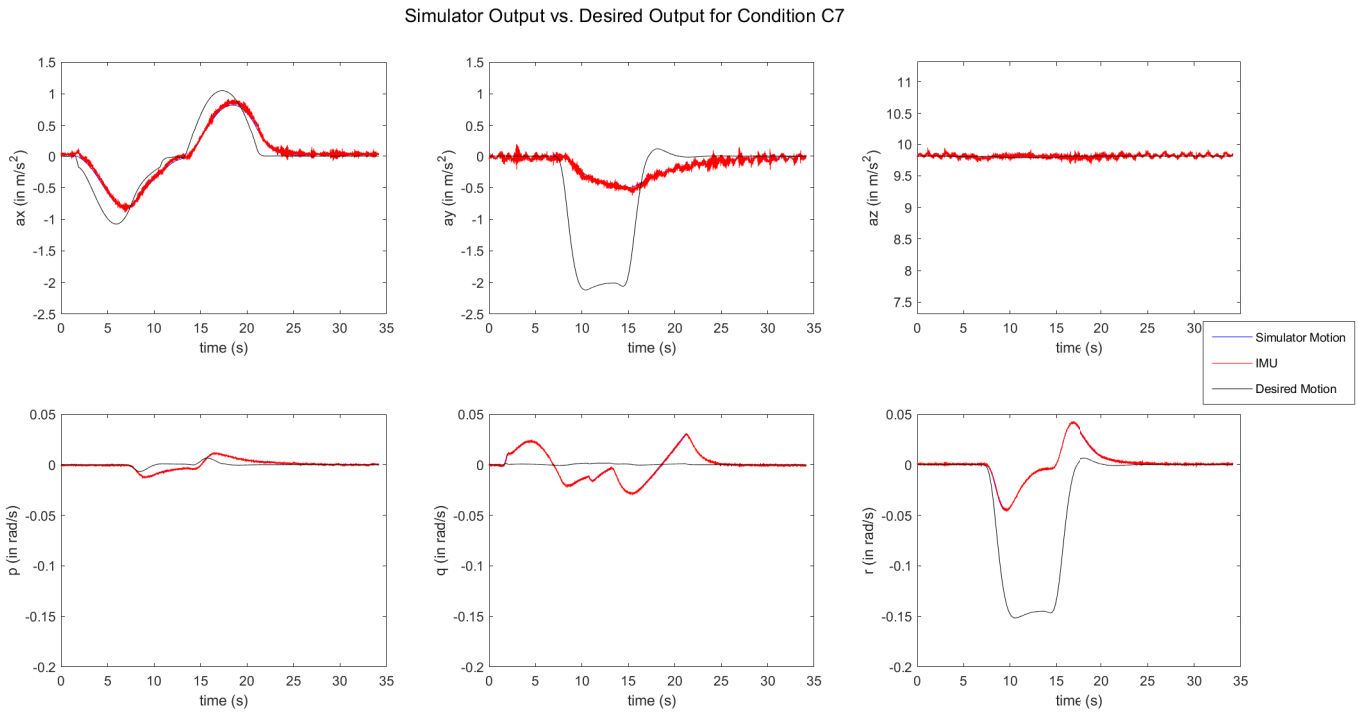


Figure B.7: Simulator output for condition C7, including the desired output and the measured IMU data.

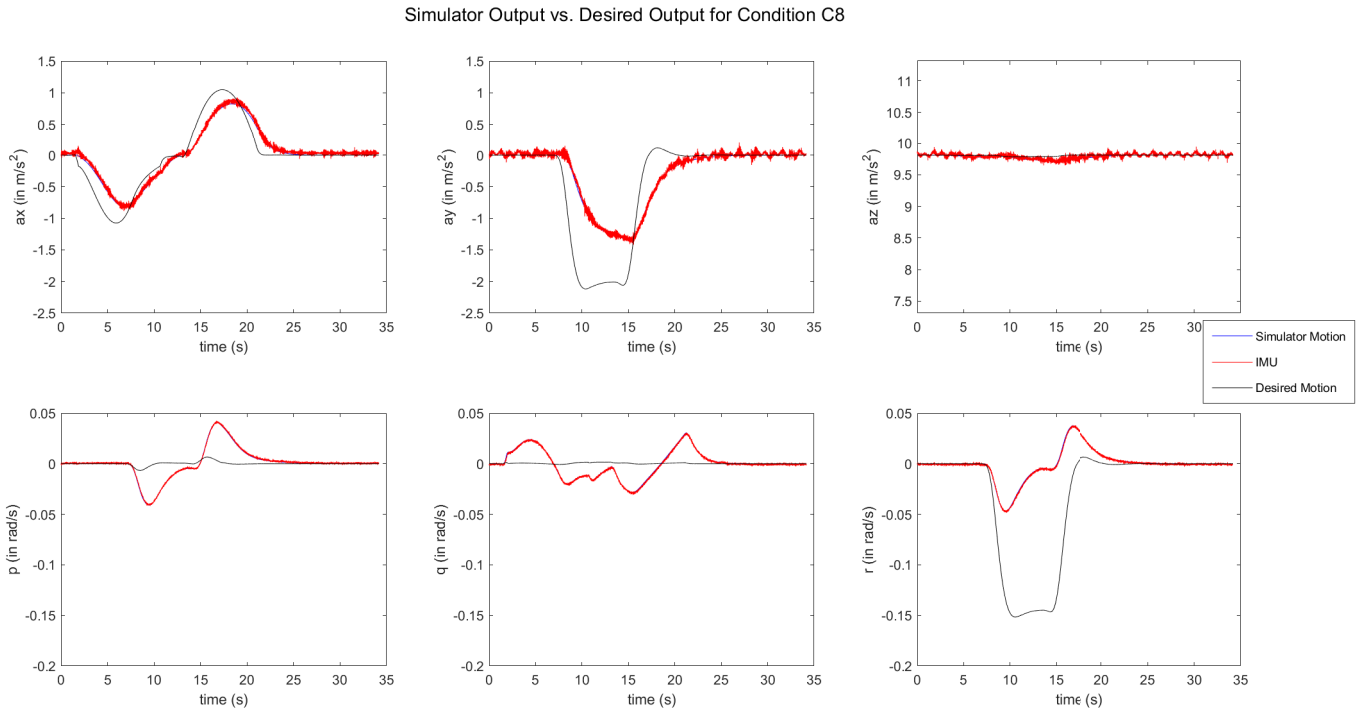


Figure B.8: Simulator output for condition C8, including the desired output and the measured IMU data.

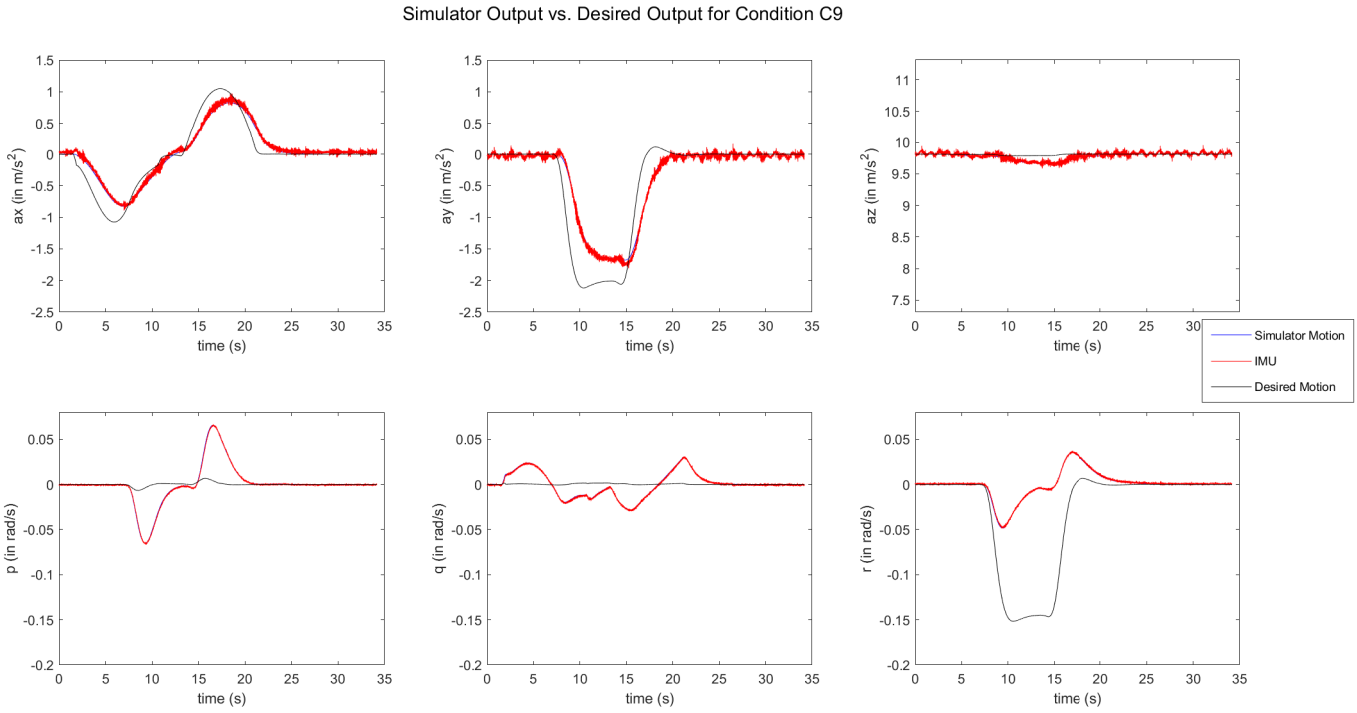


Figure B.9: Simulator output for condition C9, including the desired output and the measured IMU data.

C

Experiment Forms

Participants were instructed to first read an instruction form, where the experiment procedure was explained in detail. After that, the experimenter asked questions to verify that the participants understood their task. During the training phase, it was again checked if participants understood their task by observing their rating. Finally, the instructions were briefly repeated after the training trials, such that everything was clear for the experiment trials. After the experiment was finished, participants were asked to fill in a questionnaire that evaluated how they gave their rating. Both forms are shown below.

Instructions for participants

Aim of the experiment

In an ideal world, motion simulators would replicate real motion perfectly. Unfortunately, motion simulators are limited in their physical space, meaning that the simulator motion might not always be what you would expect. The goal of this experiment is to investigate the perceived realism of a couple of different settings of a motion simulator. We will do so by asking participants (you) to rate the mismatch between visual and inertial stimuli while driving in the CyberPod Simulator.

Your Task

You will be positioned in the CyberPod Simulator, where you will experience five six-minute driving simulations in total, of which two are training sessions and the other three experiment trials. Each experiment trial consists of an initial acceleration from 0-70 km/h, followed by nine 60^{o} curves (left or right), and finally a deceleration from 70-0 km/h. In each curve, the car decelerates from 70-50 km/h, performs the curve manoeuvre, and accelerates from 50-70 km/h again. This information is also shown on the dashboard that you can see in front of you.

The motion that you feel might not always be as you would expect, based on what you see. It is therefore important that you focus on the motion that you feel, because it is your task to rate the mismatch between the motion that you feel, and the motion that you would expect based on what you see. In order for you to be able to focus on the motion that you feel, you do not have to drive the car, meaning that you are a passenger. The visual steering wheel and speedometer can help you imagine what a real car would feel like (see Figure 1 below).

Figure 1 The view during driving

The rating

It is possible that the motion that you feel is not the same as compared to what you would expect when driving a real car, based on what you see. Your task is to continuously rate the mismatch between the motion that you feel, and the motion that you would expect based on the visuals, which are always 100% correct. Possible mismatches include, but are not limited to:

- The motion may feel weaker or stronger than the visuals indicate
- The direction of the motion may feel wrong when compared to the visuals
- The timing of the motion may be incorrect
- There is motion where there shouldn’t be any according to the visuals, or vice versa

During each trial, all nine curves are set up identical (so the visuals are the same), but the motion that you feel might feel different each time. The differences may be small or even unnoticeable, but your task is to rate the mismatch as you perceive it at that time. The initial acceleration and final deceleration are exactly the same for each condition, therefore we ask you to only rate the mismatch during the nine curves.

In order to give a rating of the mismatch, you can turn a knob at your right hand side. Turning this knob to the left means that you think the motion you feel is coherent with what you would expect, turning the knob to the right means that you think there is a discrepancy between what you feel and what you would expect. During the training session, you will experience all possible motions of the experiment, and you will determine which one is the worst motion (highest mismatch). This motion will get a maximum rating. A minimum rating can be given when you think there is no mismatch between what you feel and what you expect to feel. Use these two extreme points and base all your rating on this scale. Remember to not include the initial acceleration and final deceleration in determining the worst motion, we are only interested in the curve driving part.

Training

The rating scale that you use, should range from no mismatch (knob turned to the left) to worst that you have felt in the entire trial (knob turned to the right). In order for you to determine what the worst motion is, you will first get to feel all the motions before you have to start rating. During this run, it is important that you determine which of the motions you would rate as the worst. After this first training trial, you will get to experience all the motions another time, but this time you'll have to give a rating for all the motion mismatches. Try to base your rating on the previously experienced trial, in which you determined what the worst motion was. This motion should get a maximum mismatch rating, which will look like the bar as indicated below in Figure 2.

Figure 2 Knob turned all the way to the right (largest mismatch)

At moments when you think there is no mismatch, you should turn the knob all the way to the left, until the rating bar disappears. Now that you have determined the minimum and maximum rating, you have calibrated your rating scale. Please try to use this rating scale consistently throughout the rest of the experiment.

The Experiment

After the training session, there is some time for a (small) break. If you still have any questions regarding the rating, now is the time to ask. When everything is clear, it is time to begin the experiment trials. The same nine curves will be simulated three times, each time in a different order. All that you have to do is continuously rate the mismatch between what you think you should be feeling and what you are actually feeling. Use the rating scale that you calibrated during the training session. Try to be as consistent as possible, meaning that you give the worst motion of the entire trial the highest possible rating.

Afterwards

When you are done with the trials, we will ask you to fill out a questionnaire, after which you will receive your payment (if you are not an employee of the MPI) and then you are finished.

Safety/comfort

The goal of this experiment is to gather data regarding the quality of certain settings of the motion simulator, not to make you sick, so it is important that we stop the experiment before you are really sick. Therefore we ask you to give an indication of how you are feeling in between every trial, according to the table below.

Symptom		Score
No problems		0
Slight discomfort but no specific symptoms		1
Dizziness, warm, headache, stomach awareness, sweating, etc.	Vague	2
	Some	3
	Medium	4
	Severe	5
Nausea	Some	6
	Medium	7
	Severe	8
	Retching	9
Vomiting		10

To summarise:

When rating the mismatch:

- Always use the motion that you expect based on what you see as the reference motion.
- Don't overthink your rating, we are interested in your first impression of the motion.
- A late rating is better than no rating, so try to rate all mismatches, even if you feel you are too late
- Try to be consistent during all trials.

Important

Please keep in mind that should you start to feel bad/sick/anything unpleasant, please let the experimenter know such that the simulator can be paused or stopped. Also if you want a break, notify the experimenter. Note that stopping is possible at all times, participating in this experiment is fully voluntarily.

MISC scores:

Continuous rating of motion quality in the CyberPod Simulator - Questionnaire

Subject ID: Date:

1. How would you rate your current state?

1	2	3	4	5	6	7	8	9
Tired								Energetic
Demotivated								Motivated
Distracted								Concentrated
Weak								Fit
Ill								Healthy

2. Please give an indication on how much the following factors influenced your mismatch rating (cross one of the boxes):

	Low			Medium		High	
The motion that I experienced was stronger than I expected							
The motion that I experienced was weaker than I expected							
The motion that I experienced was in a wrong direction							
The motion that I experienced came later than I expected							
The motion that I experienced came earlier than I expected							
The motion that I experienced felt slippery (like sliding on an icy road)							
I experienced motion when I did not expect to experience motion							
I did not experience motion when I expected to experience motion							
I experienced jerky motions							
I experienced rotations during the curves							
I experienced rotations during accelerating/decelerating							
Other:							

3. How much experience do you have driving a car (km/year)?

0	<1000	<5000	<10000	>10000
O	O	O	O	O

4. How often you do travel by car (either as passenger or driver)?

Never	Yearly	Monthly	Weekly	Multiple times per week	Daily
O	O	O	O	O	O

General comments:

……

……

……

……

End of questionnaire, thank you!

D

Participant Rating Plots

The dependent measure of the experiment, described in the IEEE paper and in Chapter 6 from Part II, was the continuous rating [47]. Each participant performed three experiment trials, and this appendix shows the rating over time per participant per trial, as well as the mean rating over time per participant.

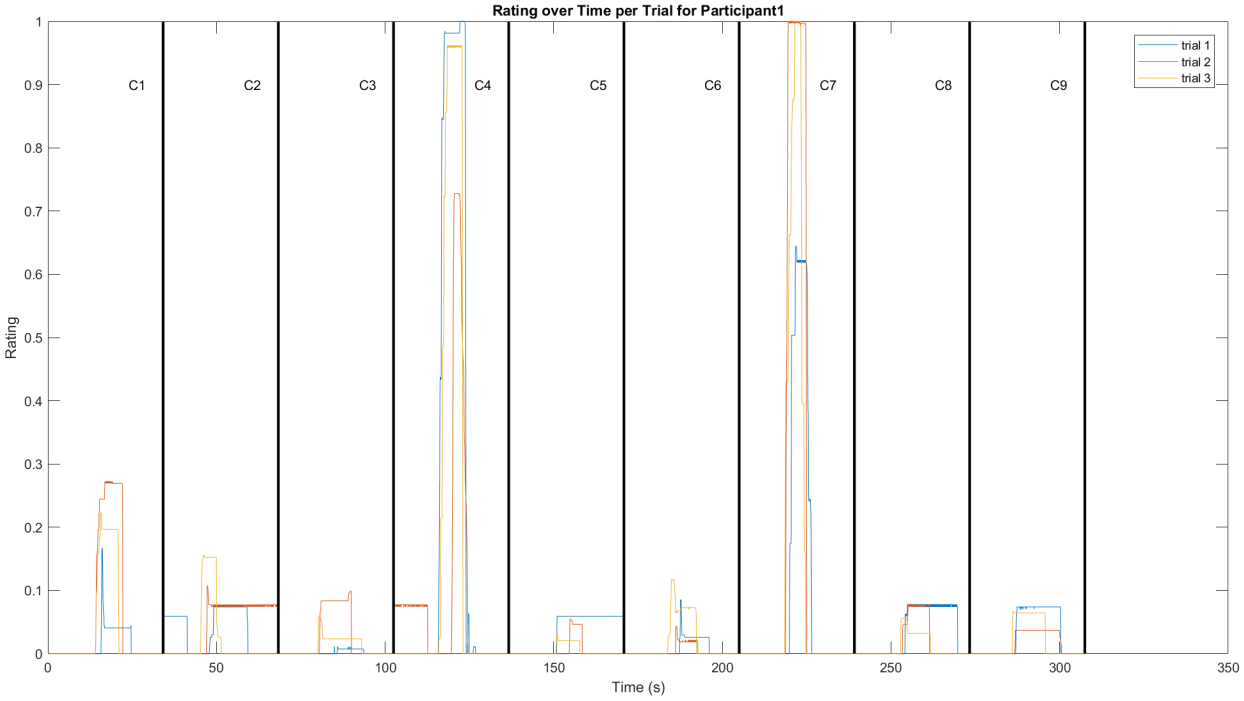


Figure D.1: Rating over time per trial for participant 1

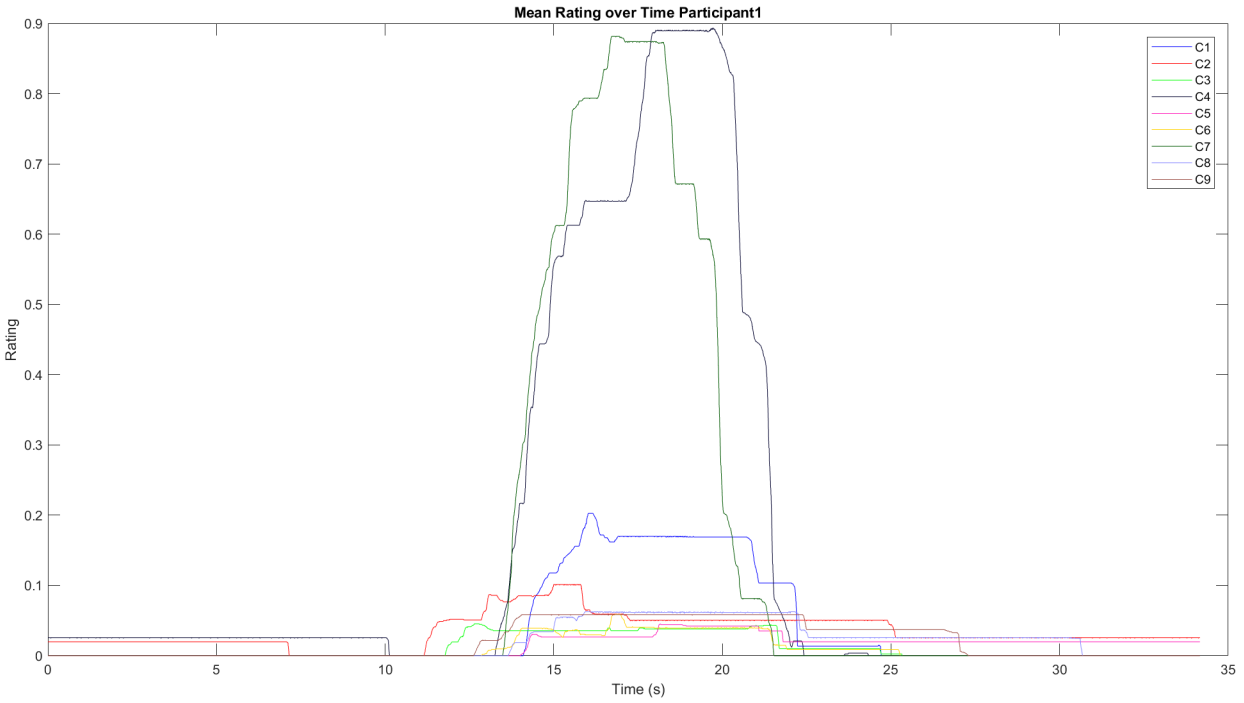


Figure D.2: Mean rating over time for participant 1

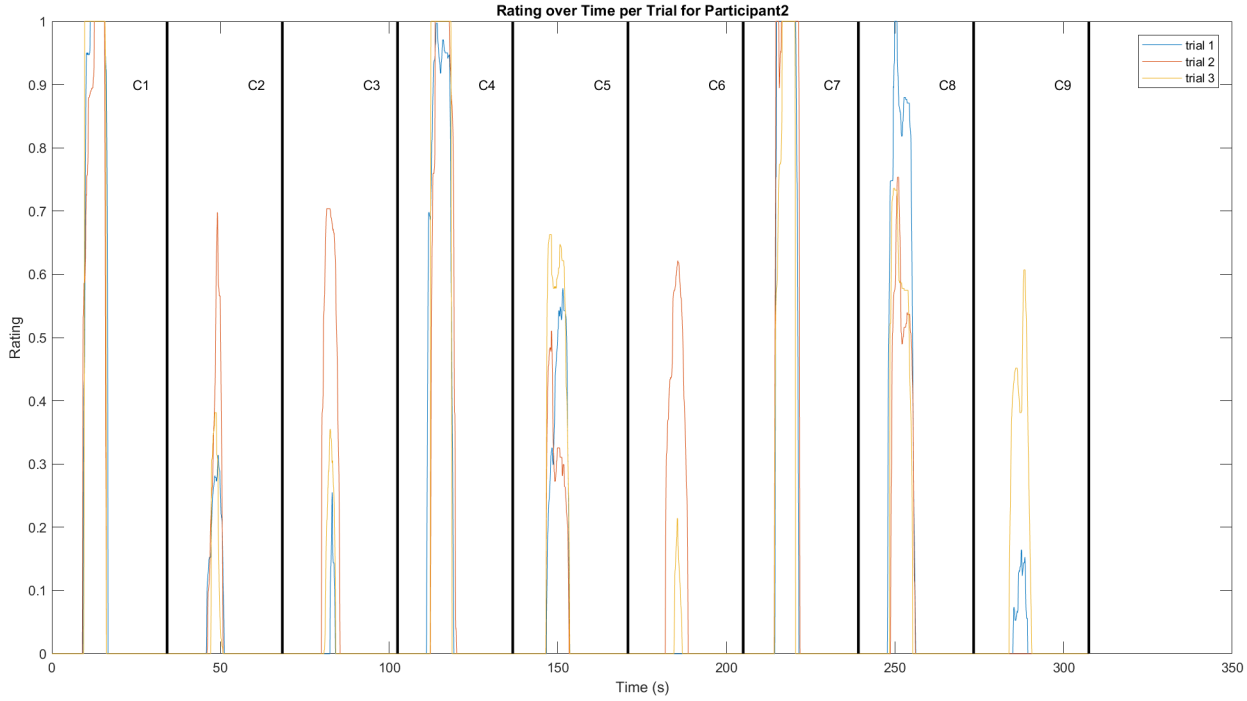


Figure D.3: Rating over time per trial for participant 2

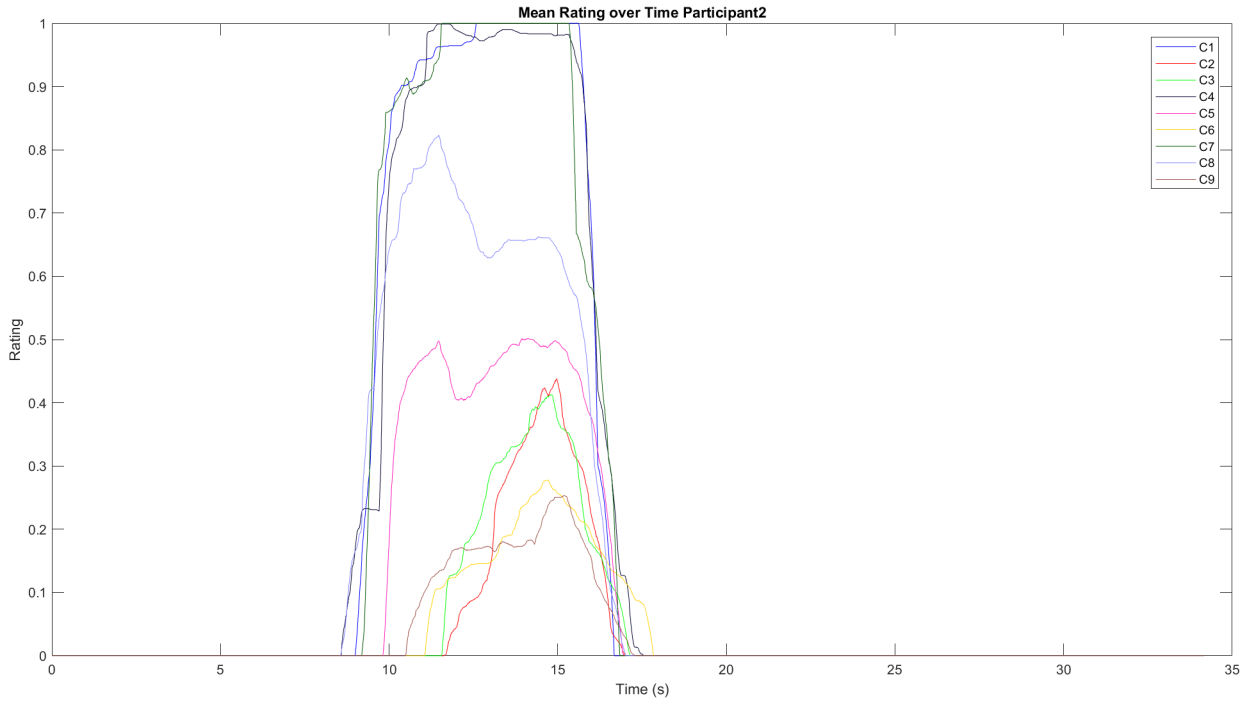


Figure D.4: Mean rating over time for participant 2

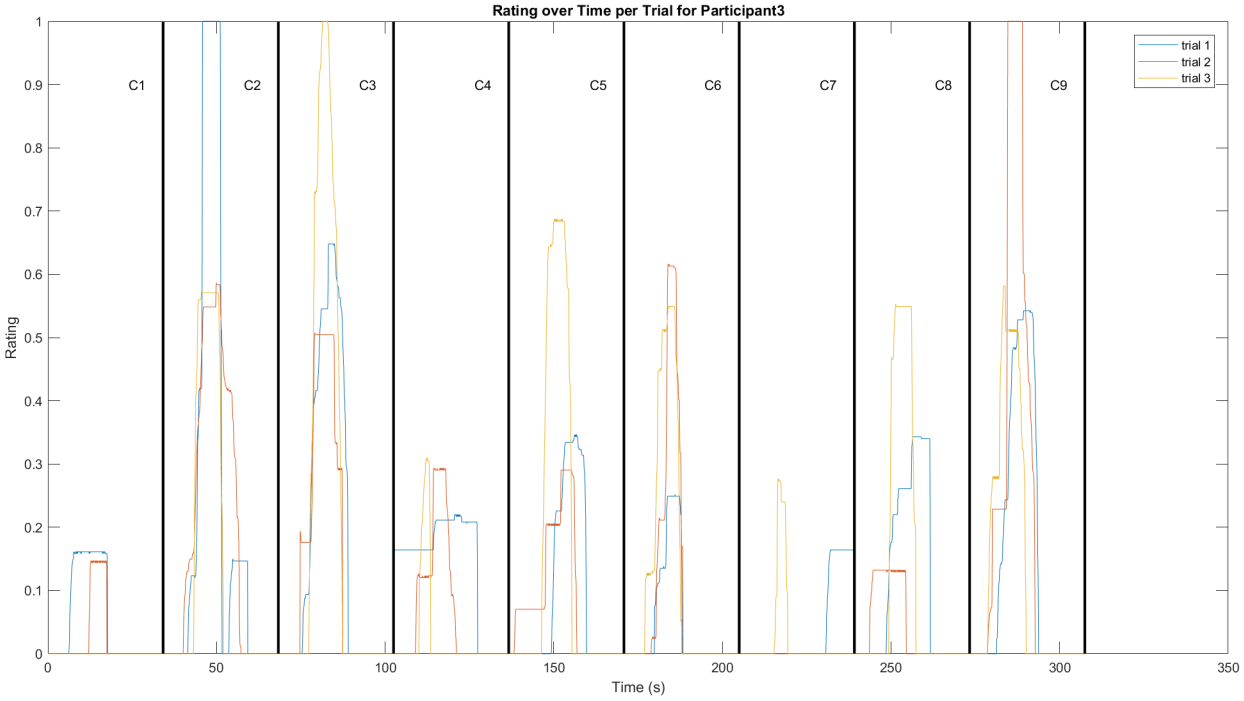


Figure D.5: Rating over time per trial for participant 3

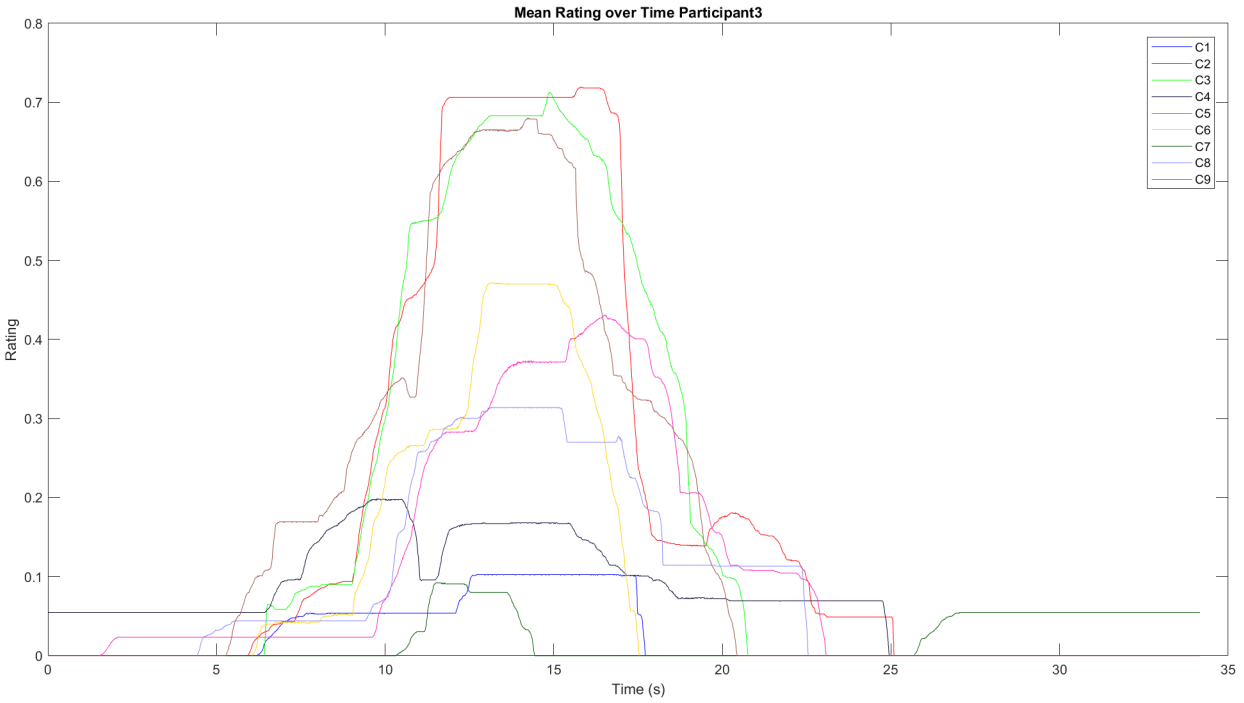


Figure D.6: Mean rating over time for participant 3

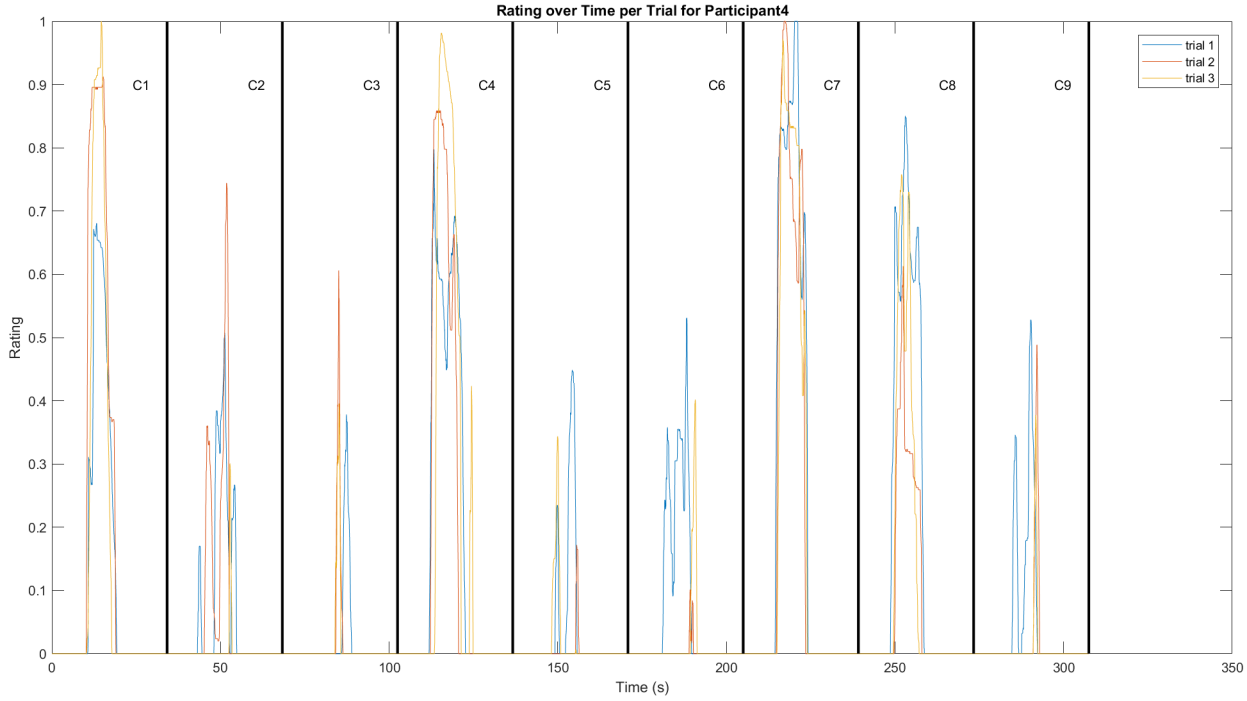


Figure D.7: Rating over time per trial for participant 4

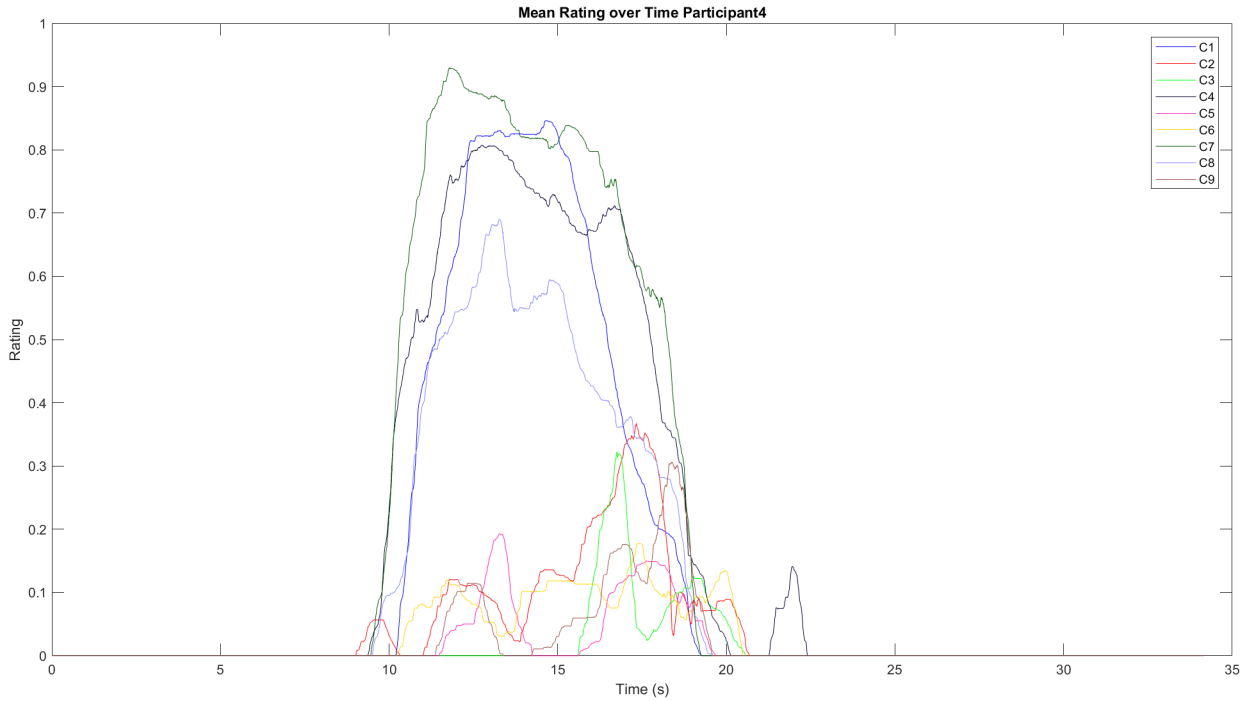


Figure D.8: Mean rating over time for participant 4

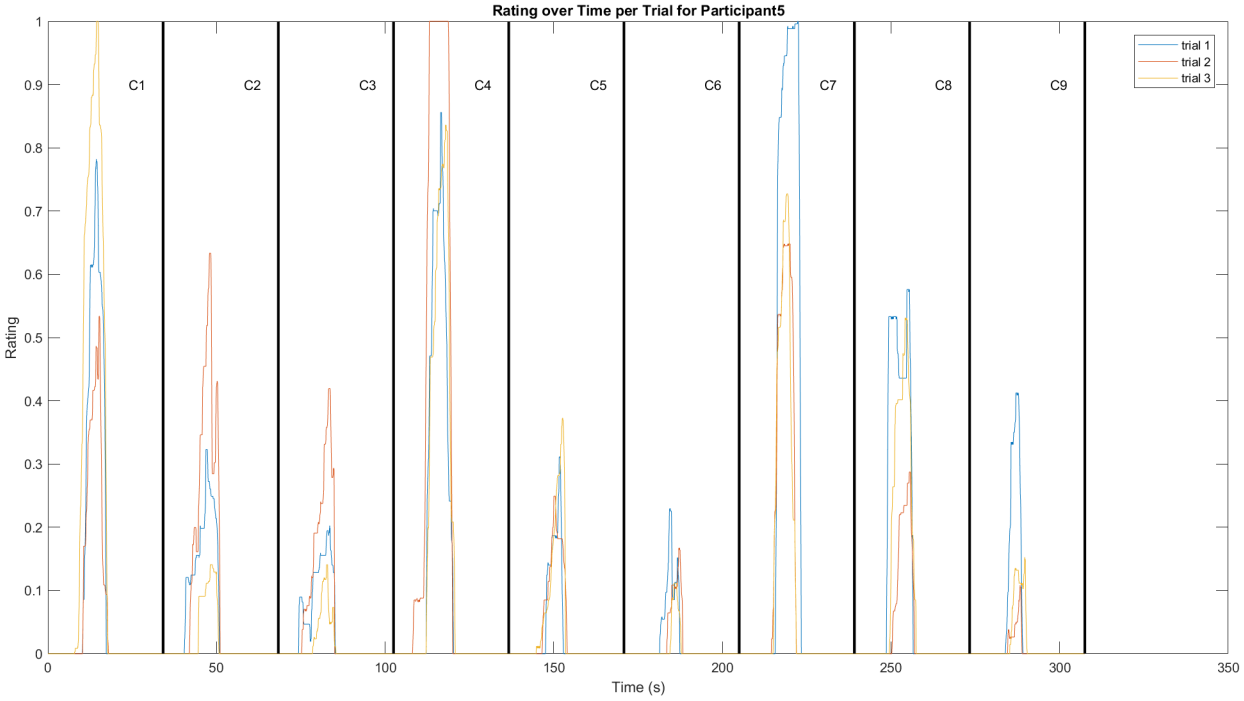


Figure D.9: Rating over time per trial for participant 5

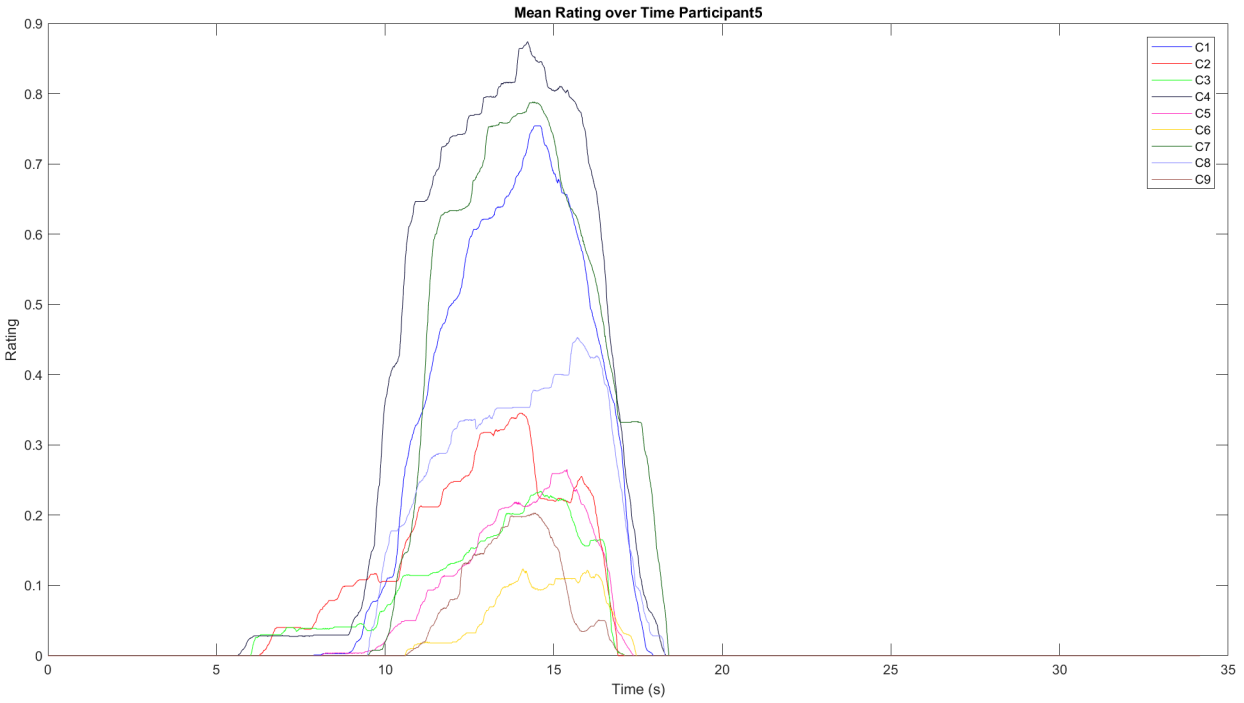


Figure D.10: Mean rating over time for participant 5

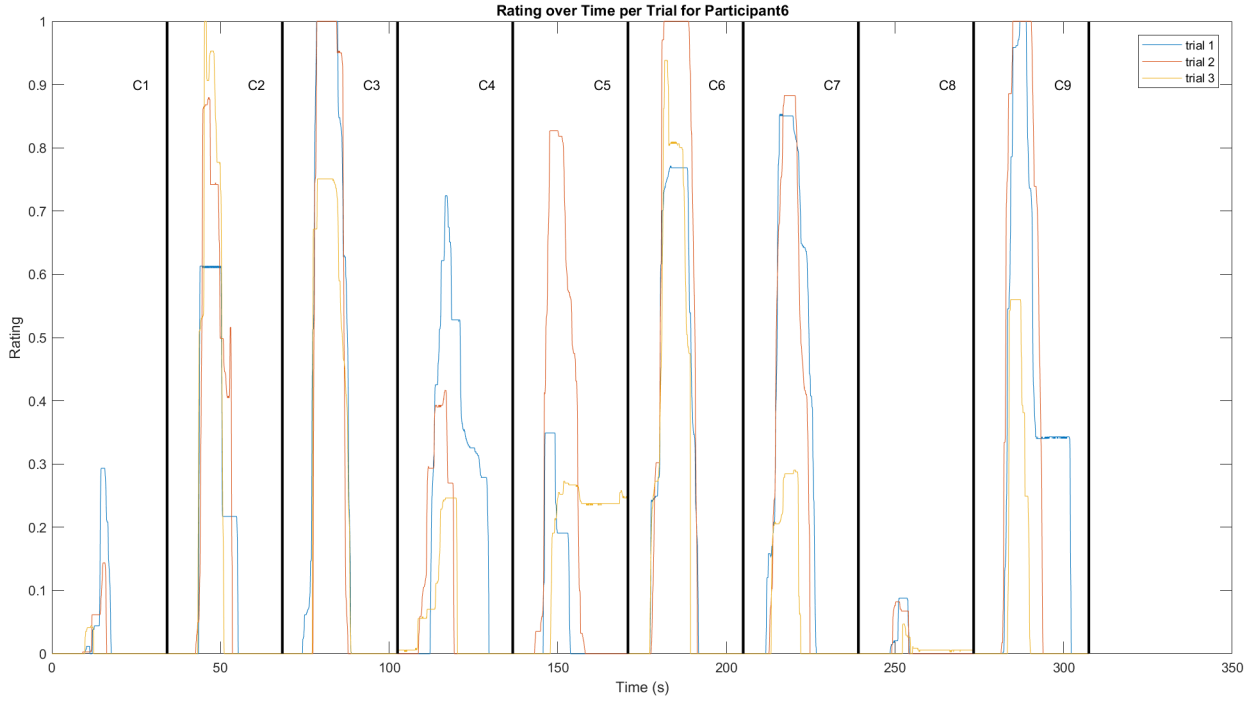


Figure D.11: Rating over time per trial for participant 6

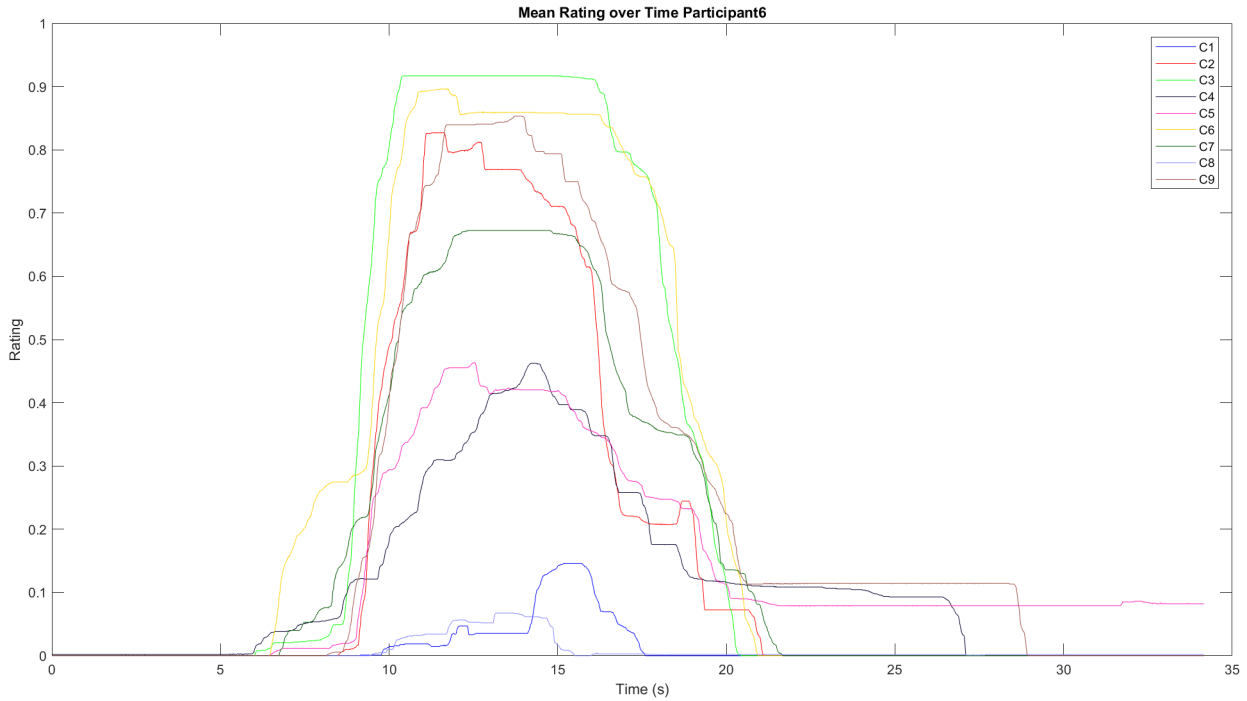


Figure D.12: Mean rating over time for participant 6

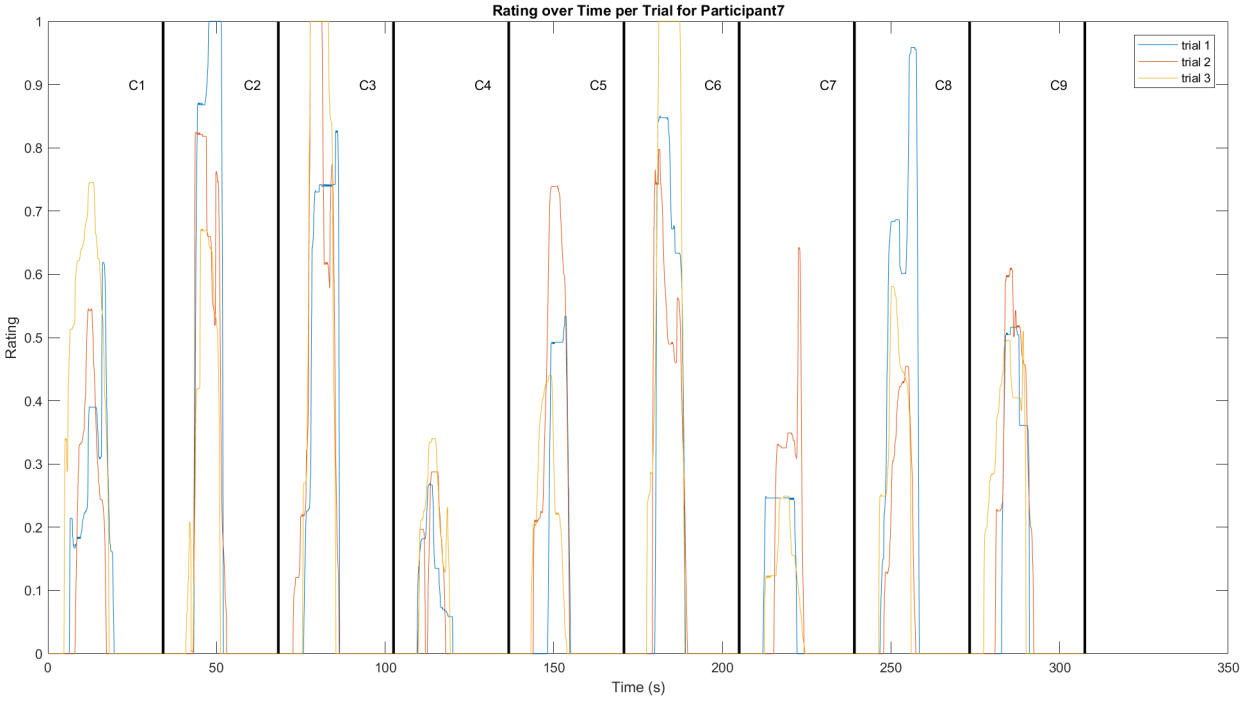


Figure D.13: Rating over time per trial for participant 7

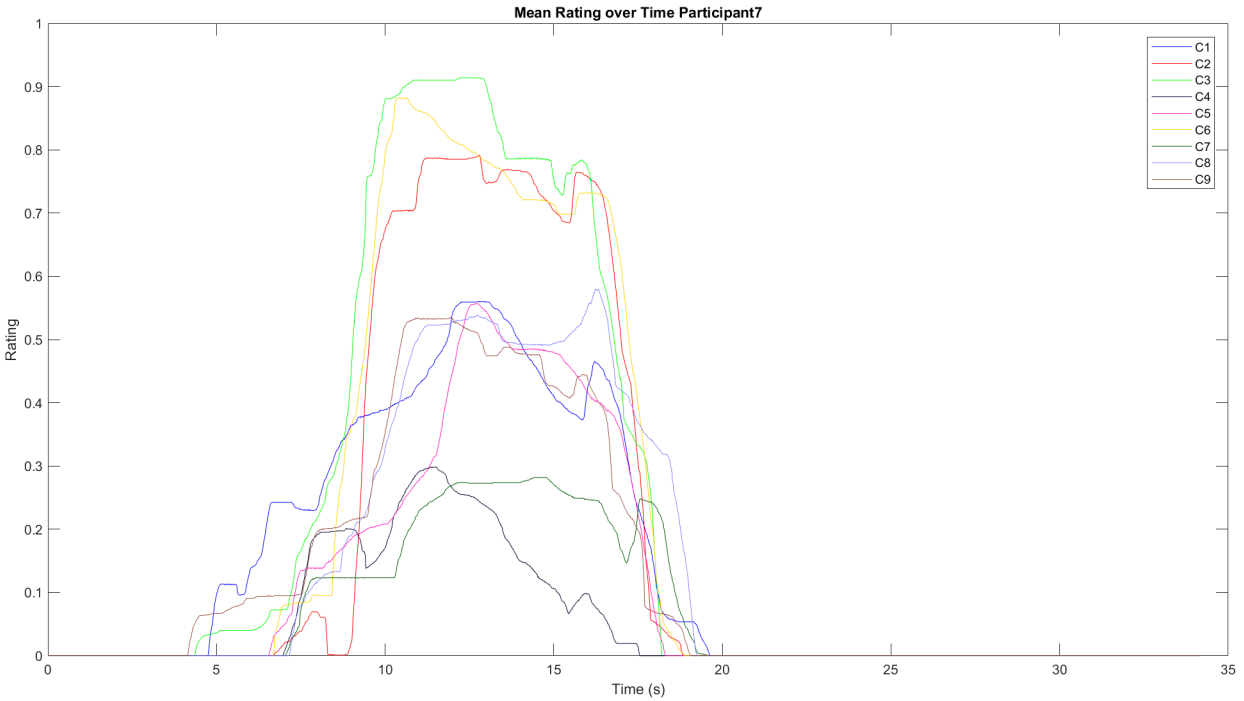


Figure D.14: Mean rating over time for participant 7

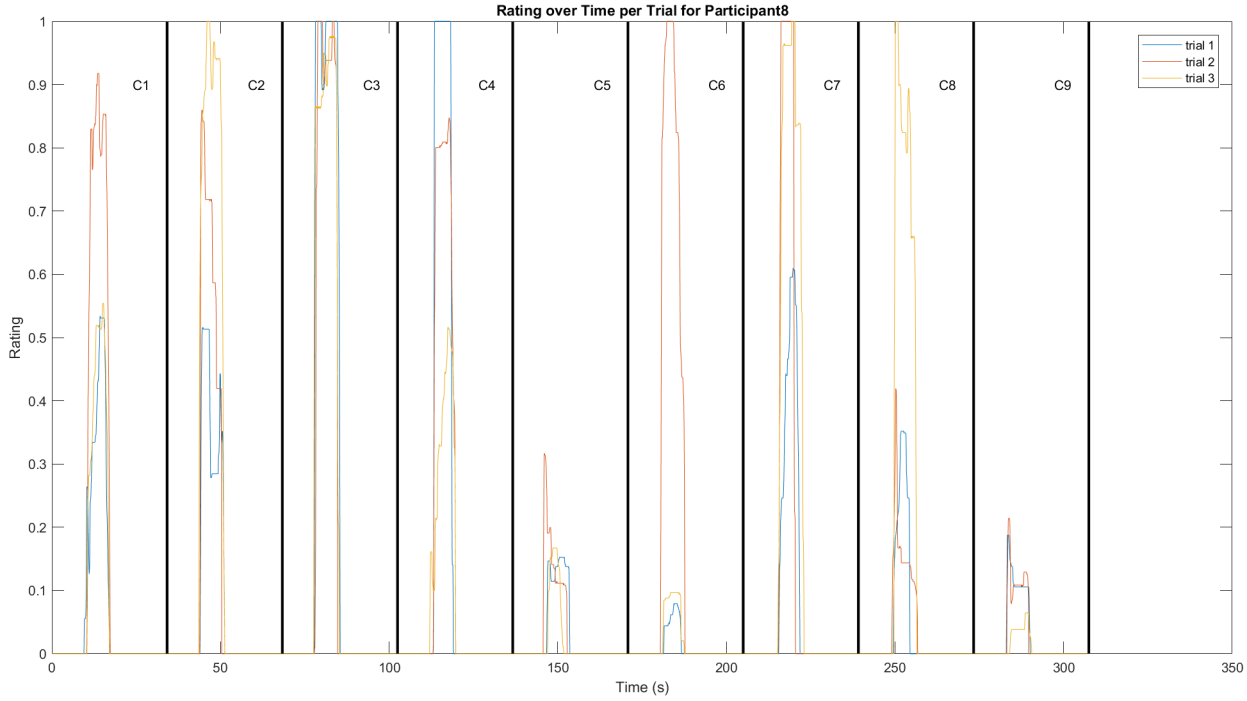


Figure D.15: Rating over time per trial for participant 8

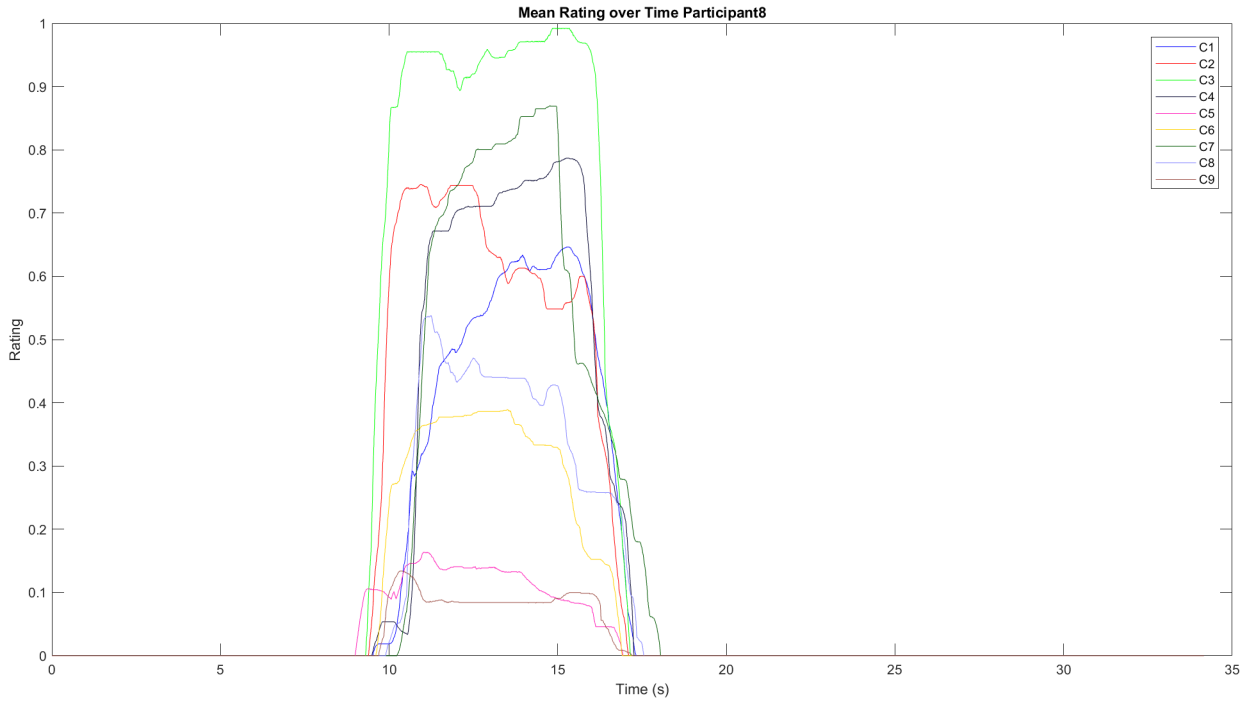


Figure D.16: Mean rating over time for participant 8

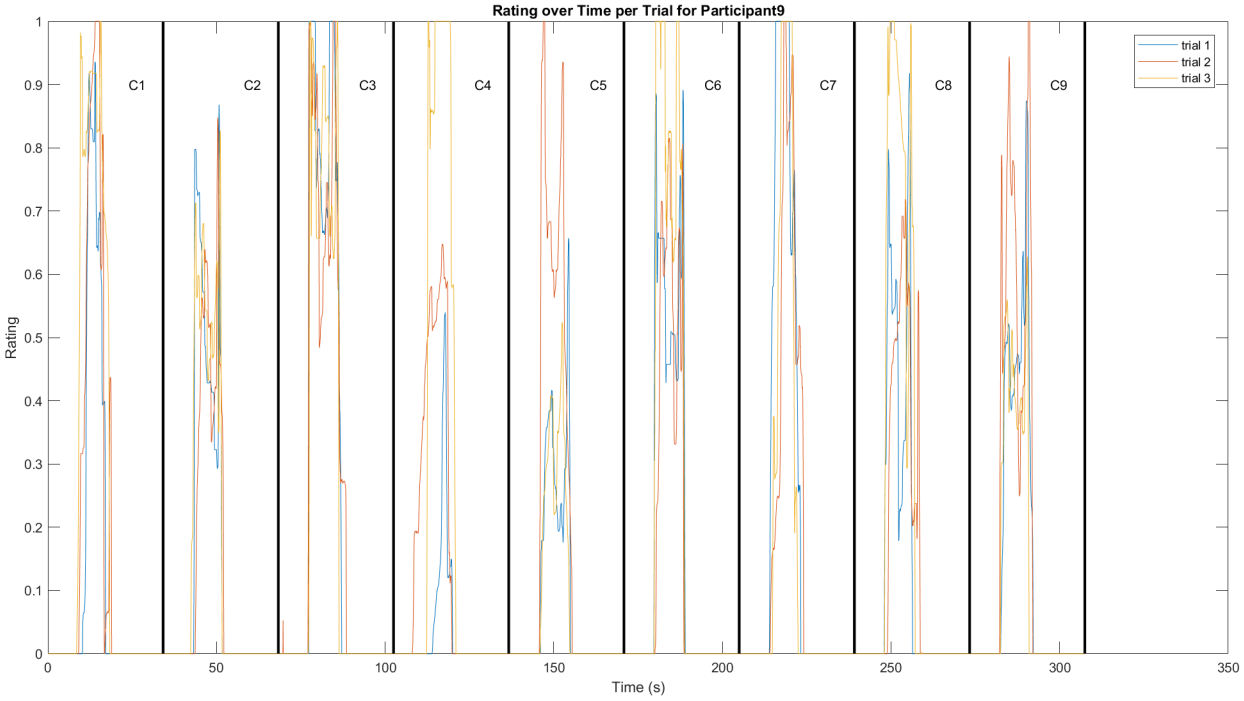


Figure D.17: Rating over time per trial for participant 9

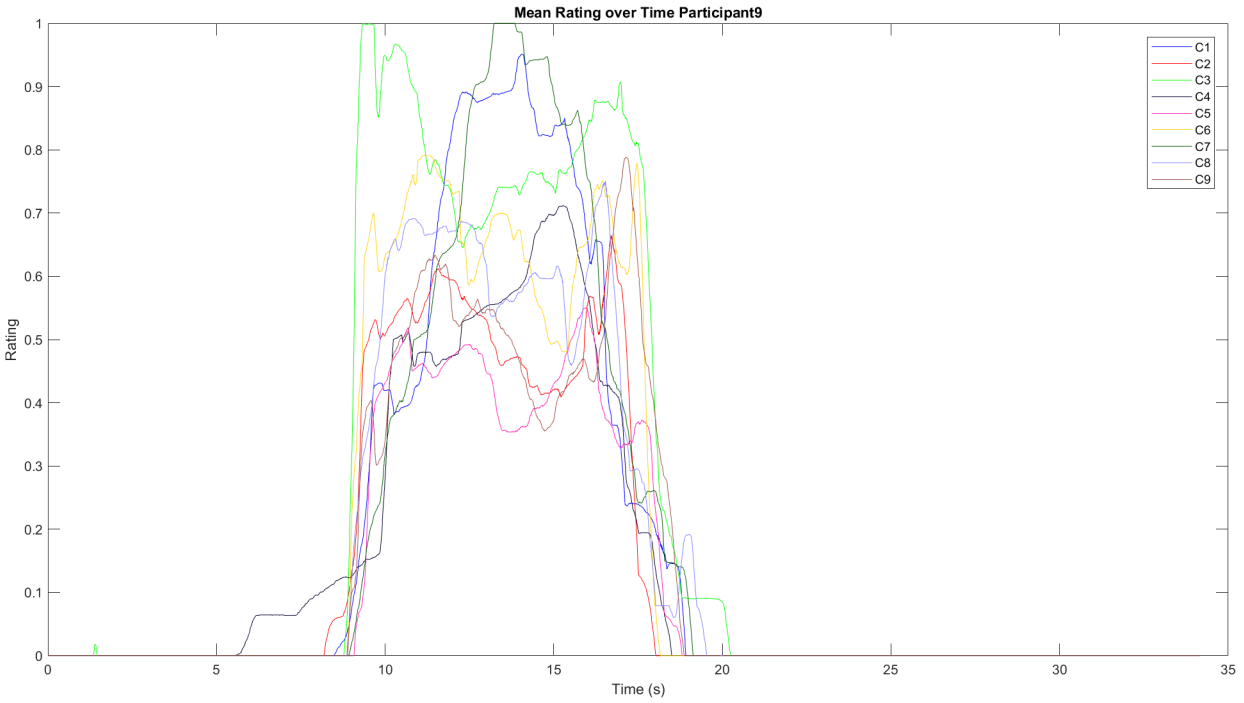


Figure D.18: Mean rating over time for participant 9

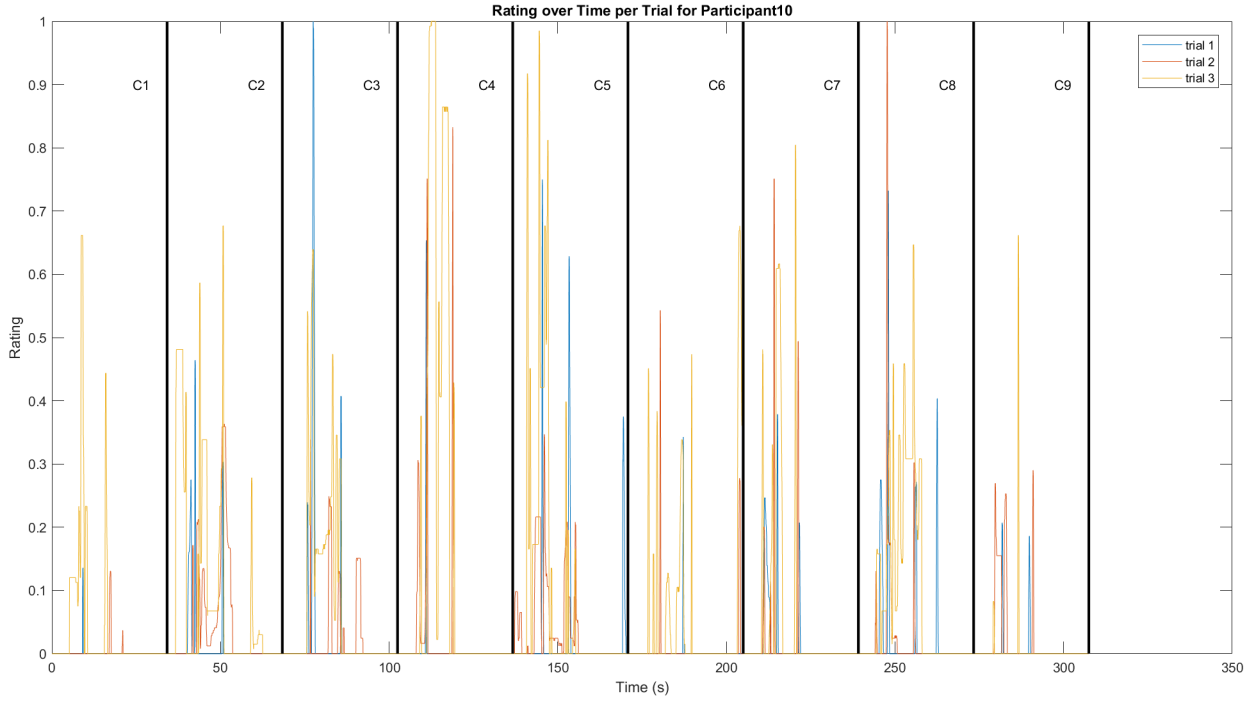


Figure D.19: Rating over time per trial for participant 10

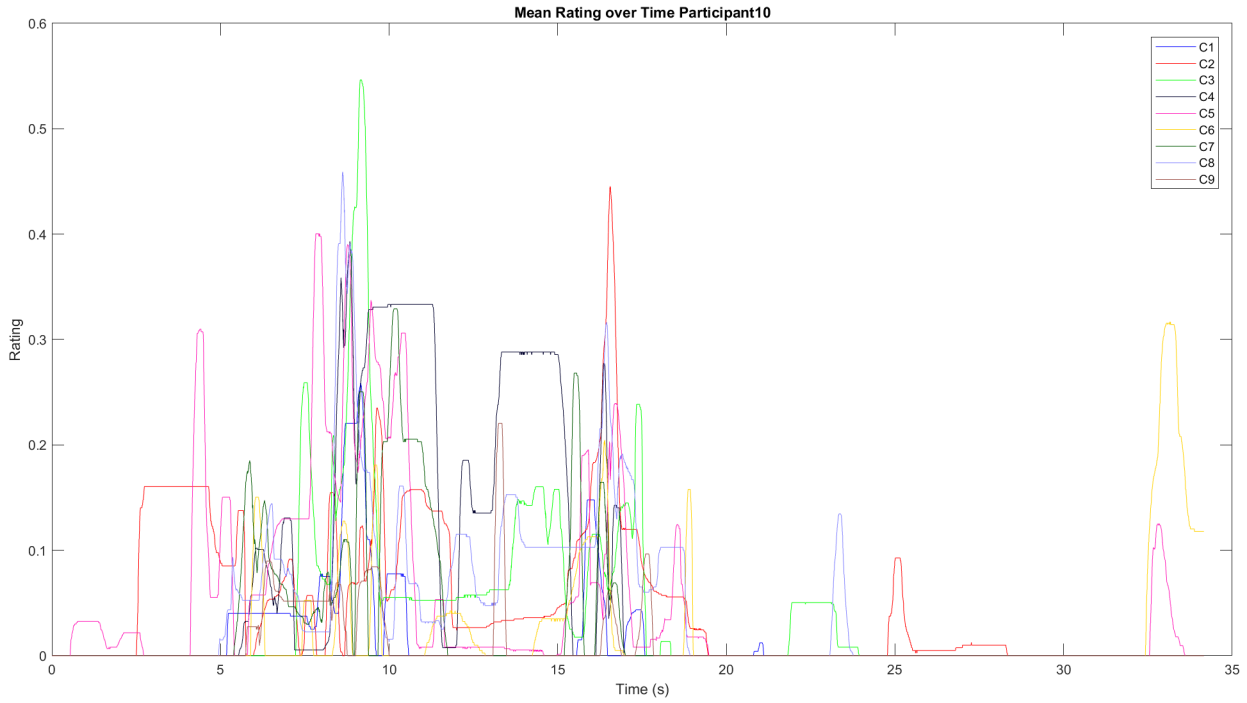


Figure D.20: Mean rating over time for participant 10

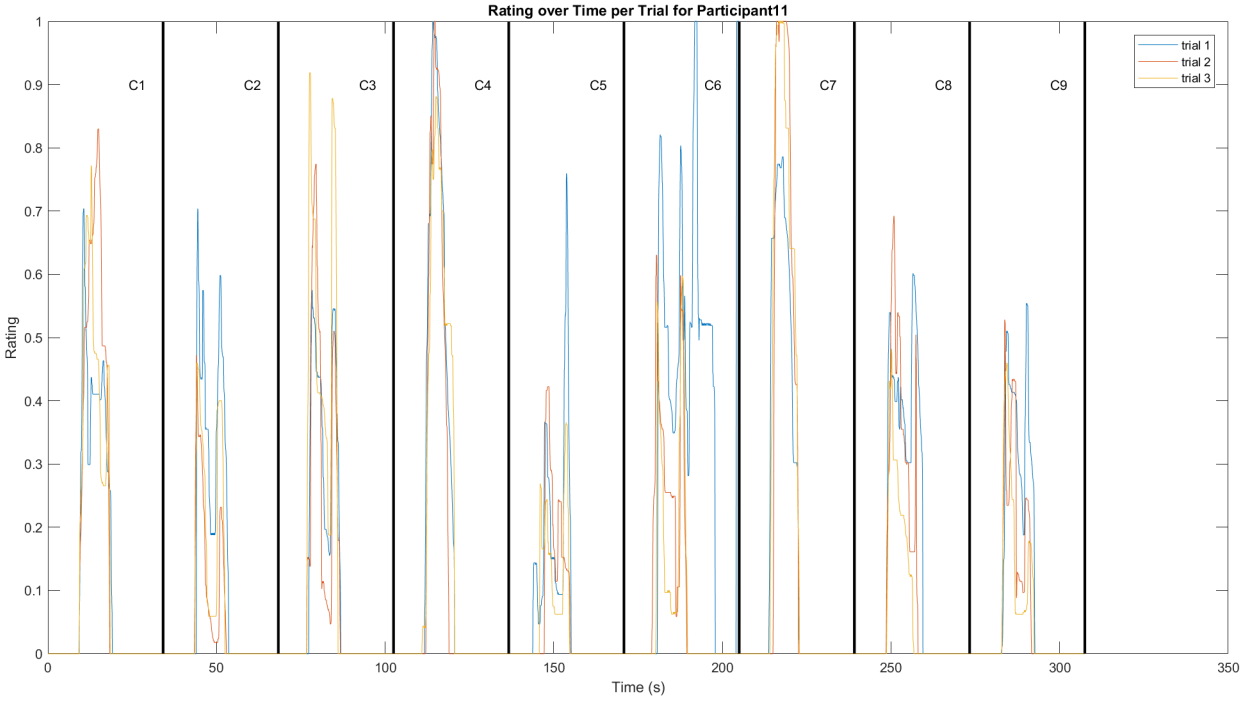


Figure D.21: Rating over time per trial for participant 11

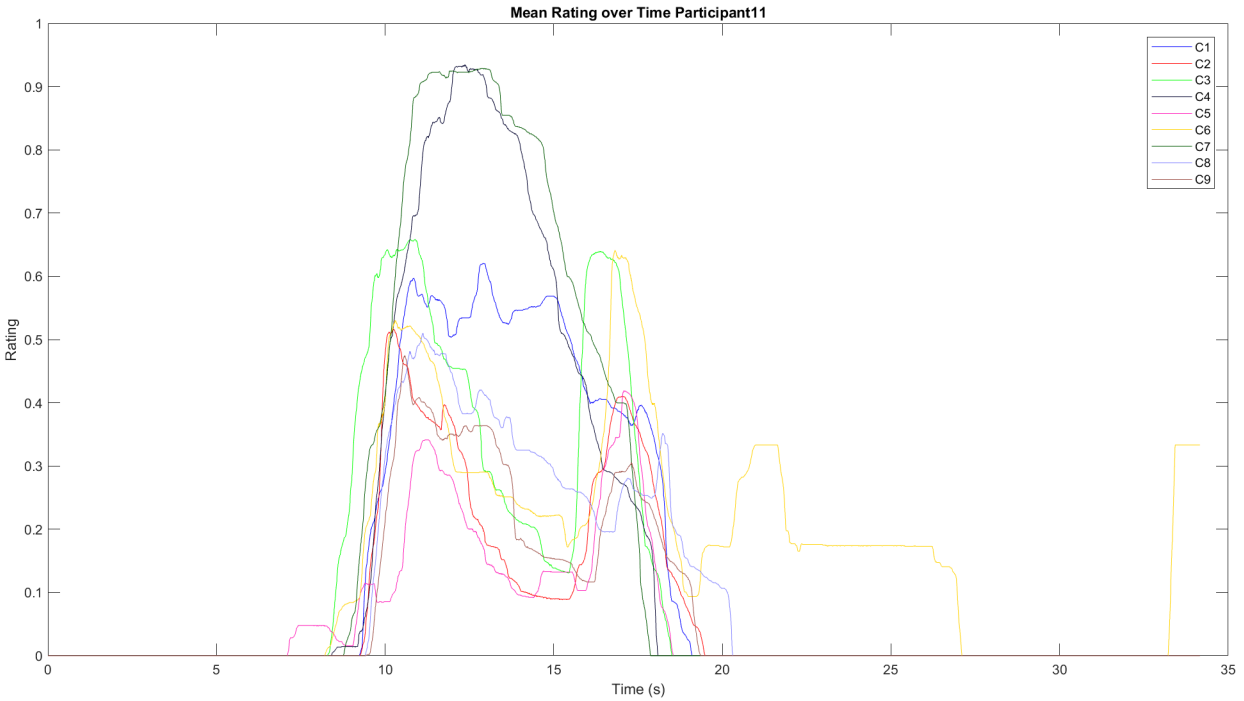


Figure D.22: Mean rating over time for participant 11

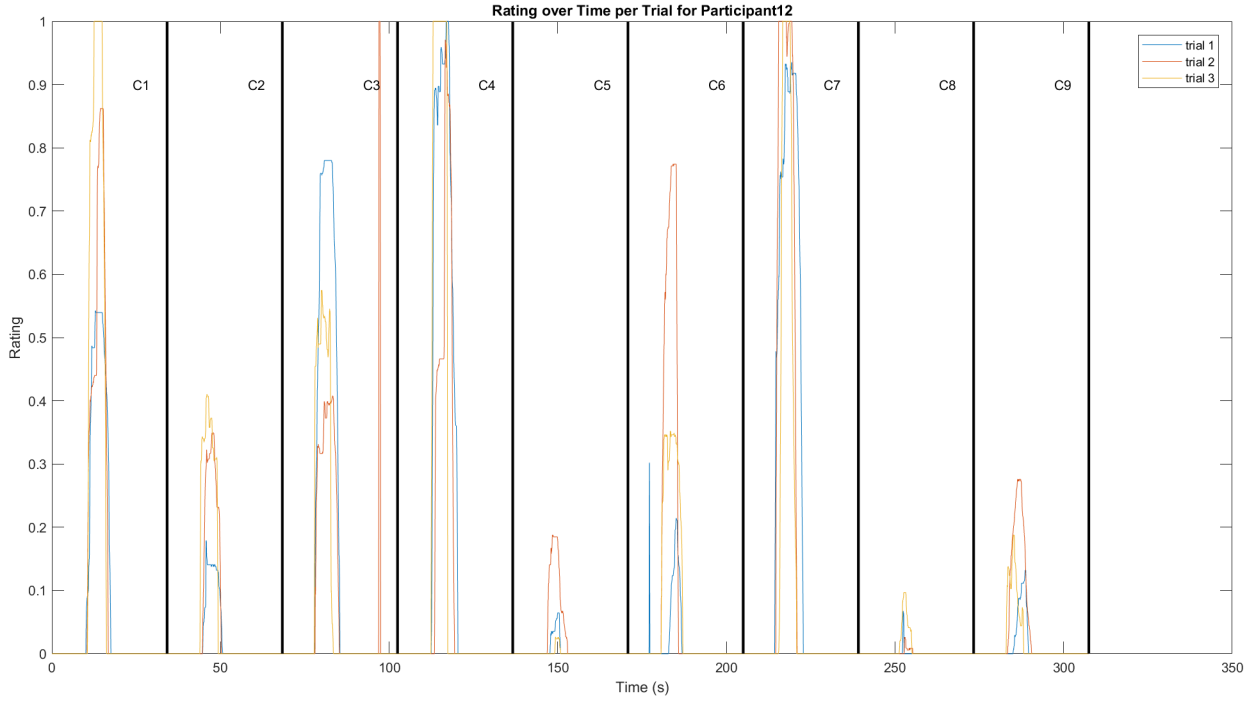


Figure D.23: Rating over time per trial for participant 12

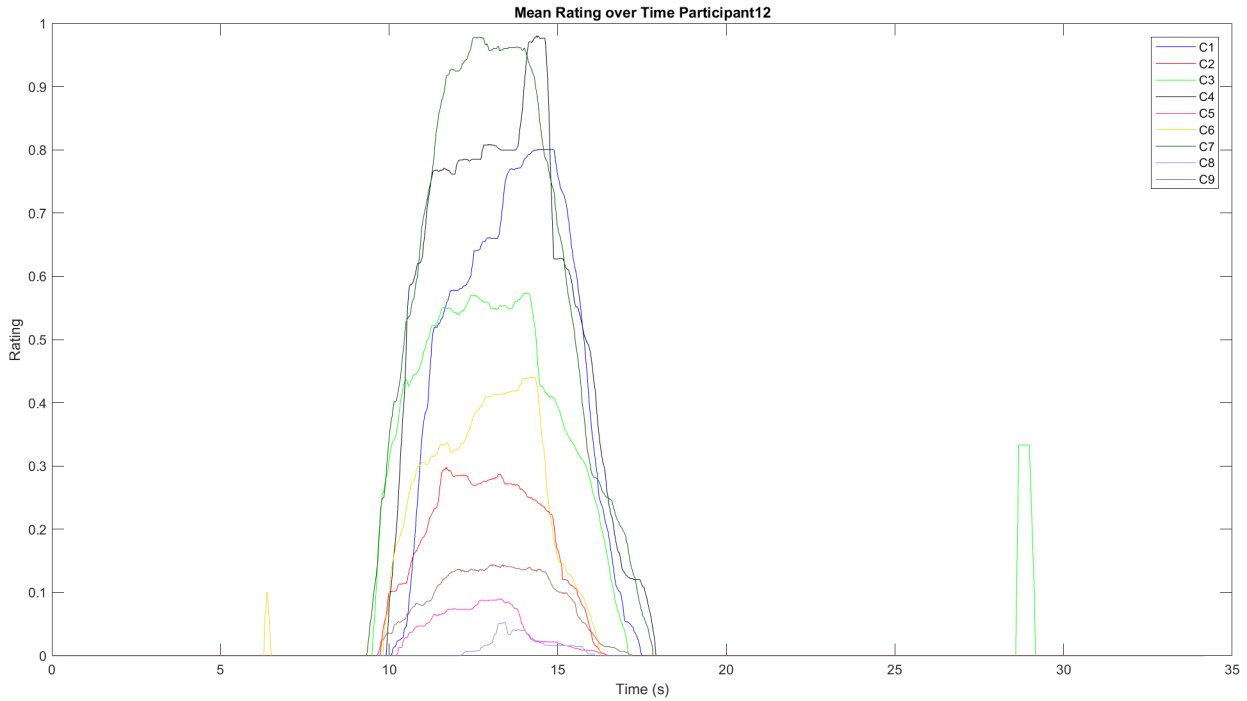


Figure D.24: Mean rating over time for participant 12

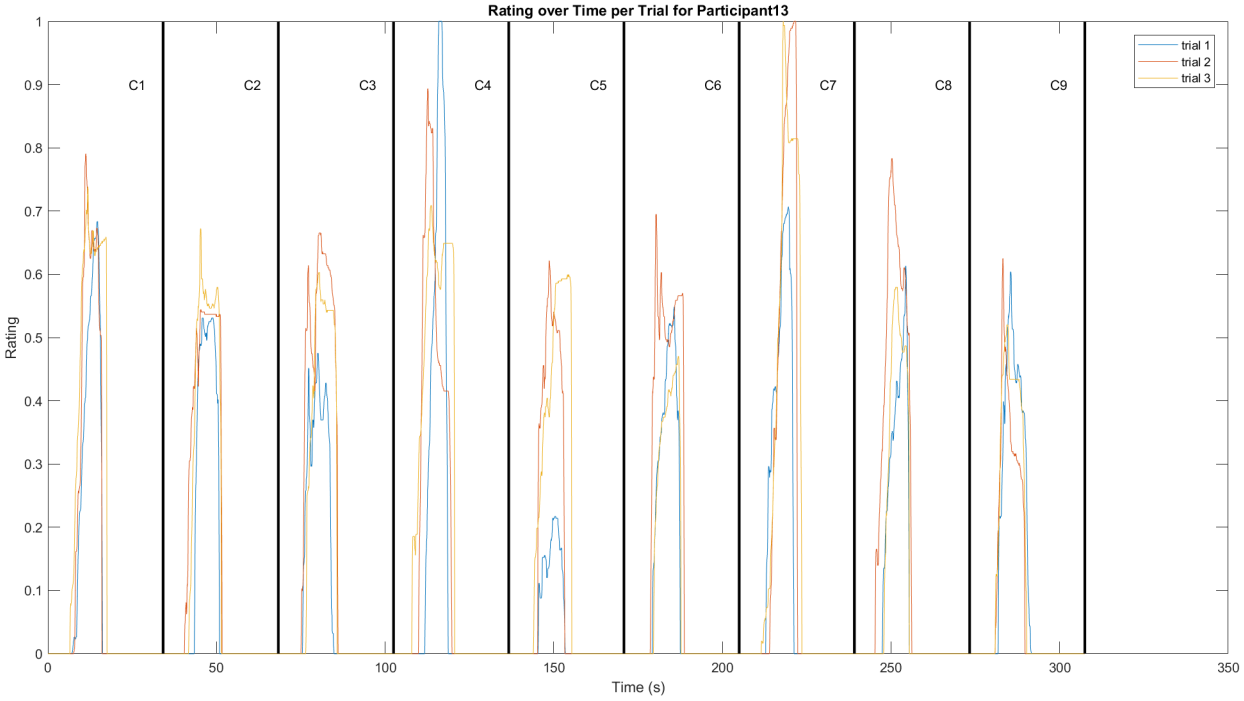


Figure D.25: Rating over time per trial for participant 13

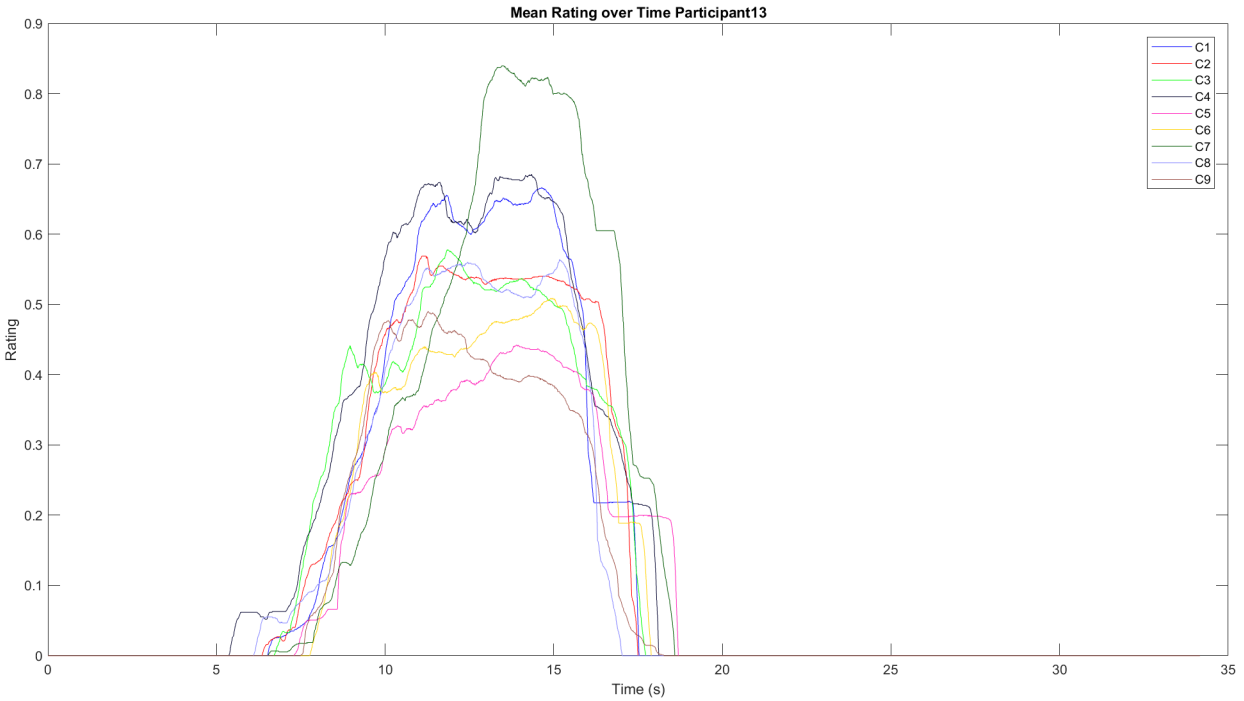


Figure D.26: Mean rating over time for participant 13

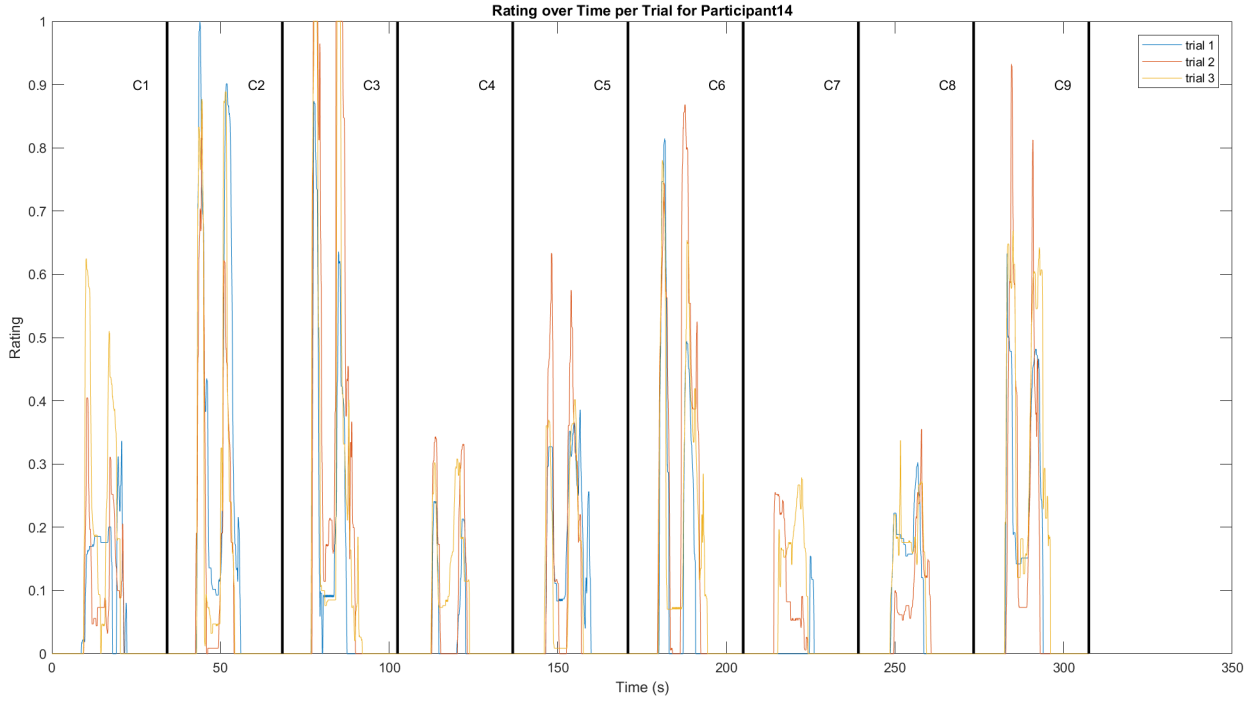


Figure D.27: Rating over time per trial for participant 14

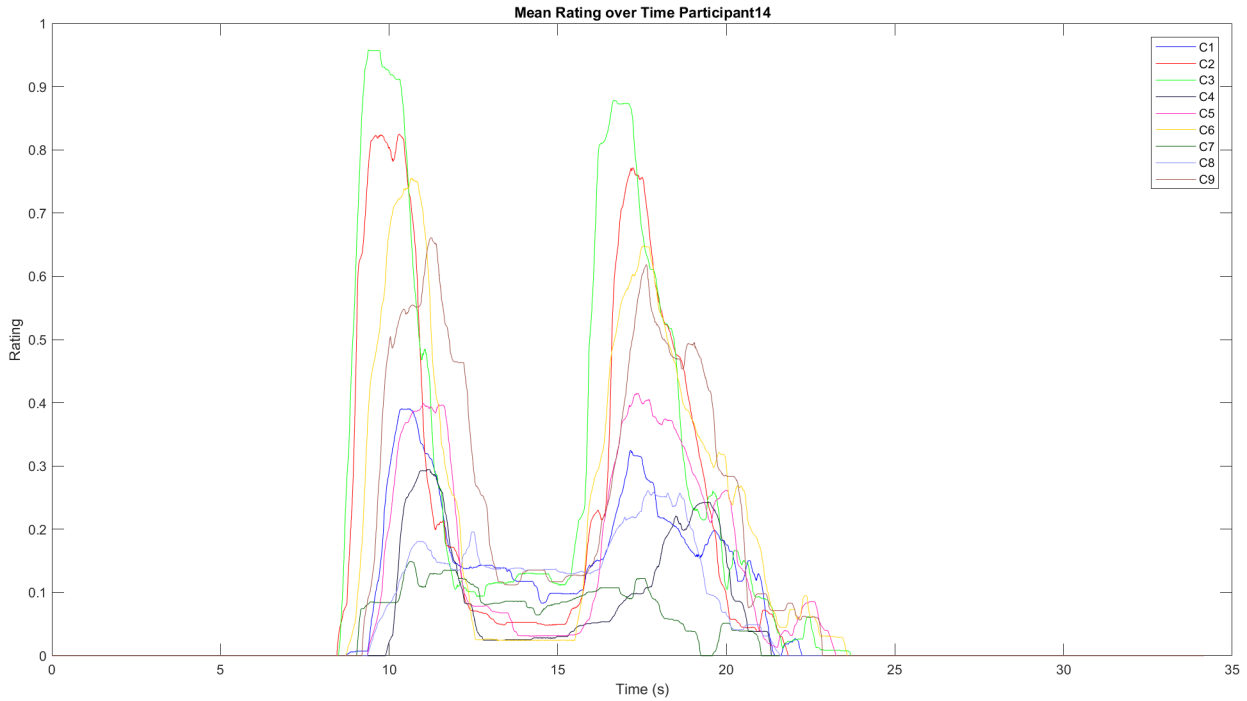


Figure D.28: Mean rating over time for participant 14

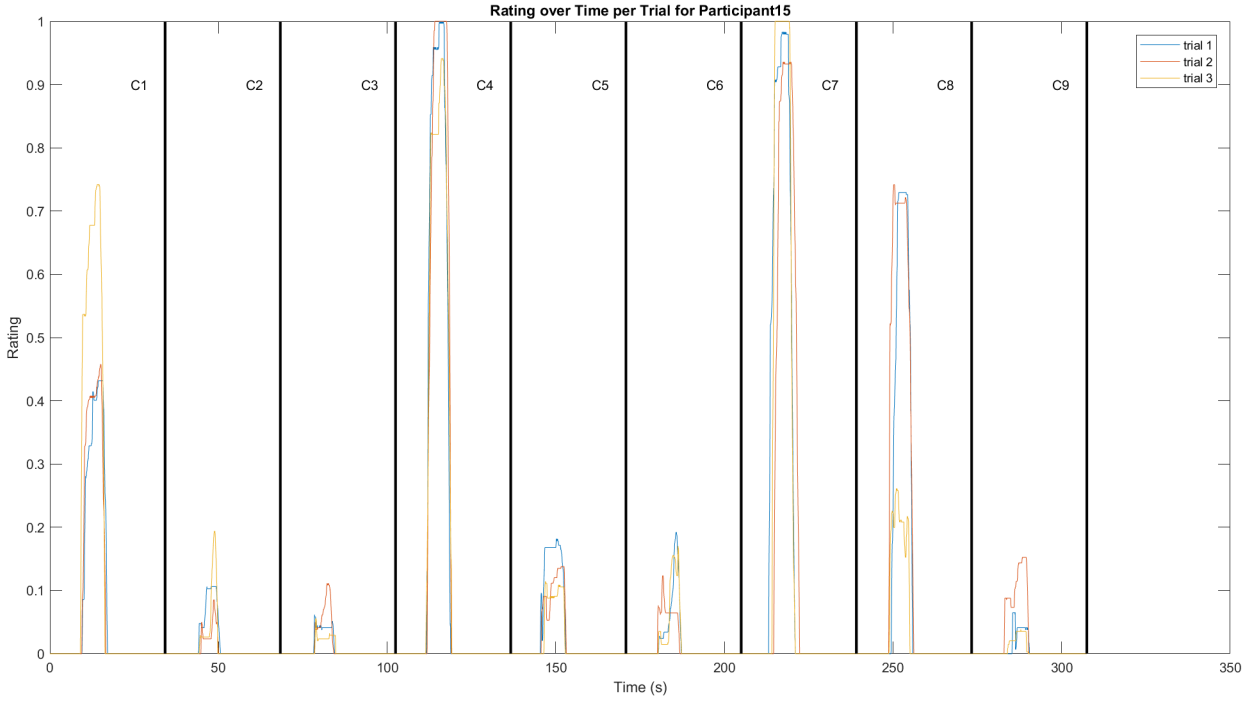


Figure D.29: Rating over time per trial for participant 15

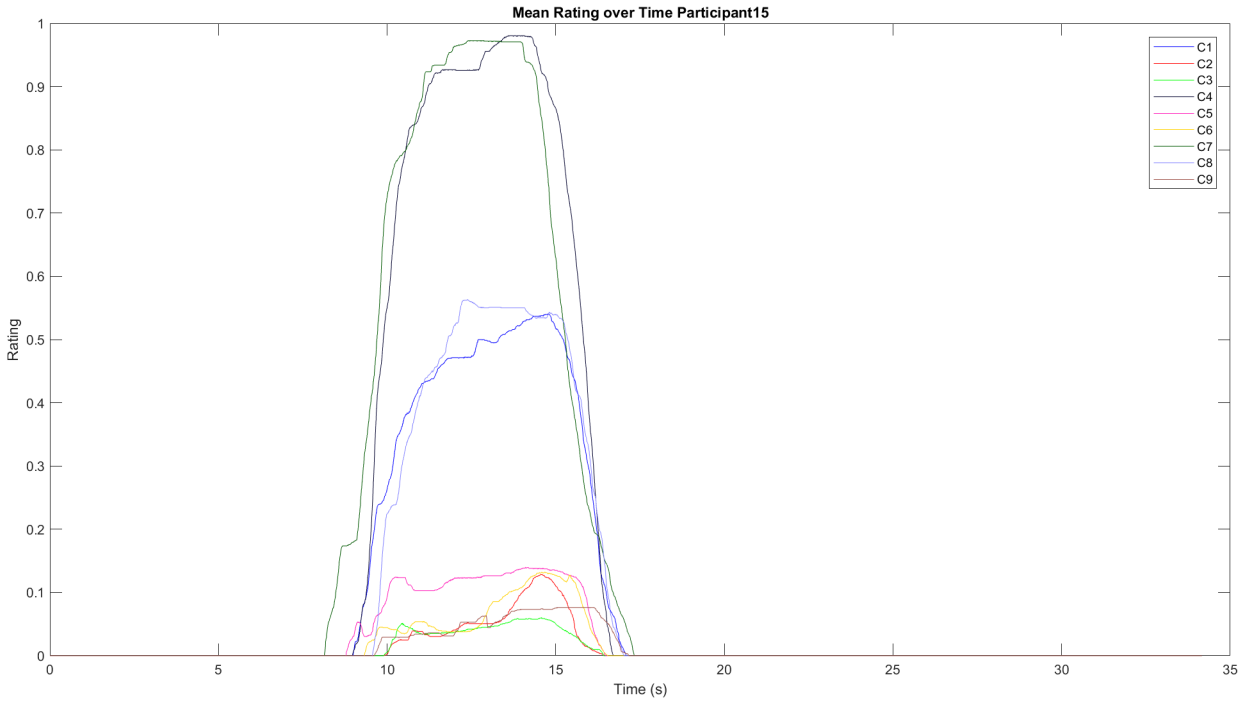


Figure D.30: Mean rating over time for participant 15

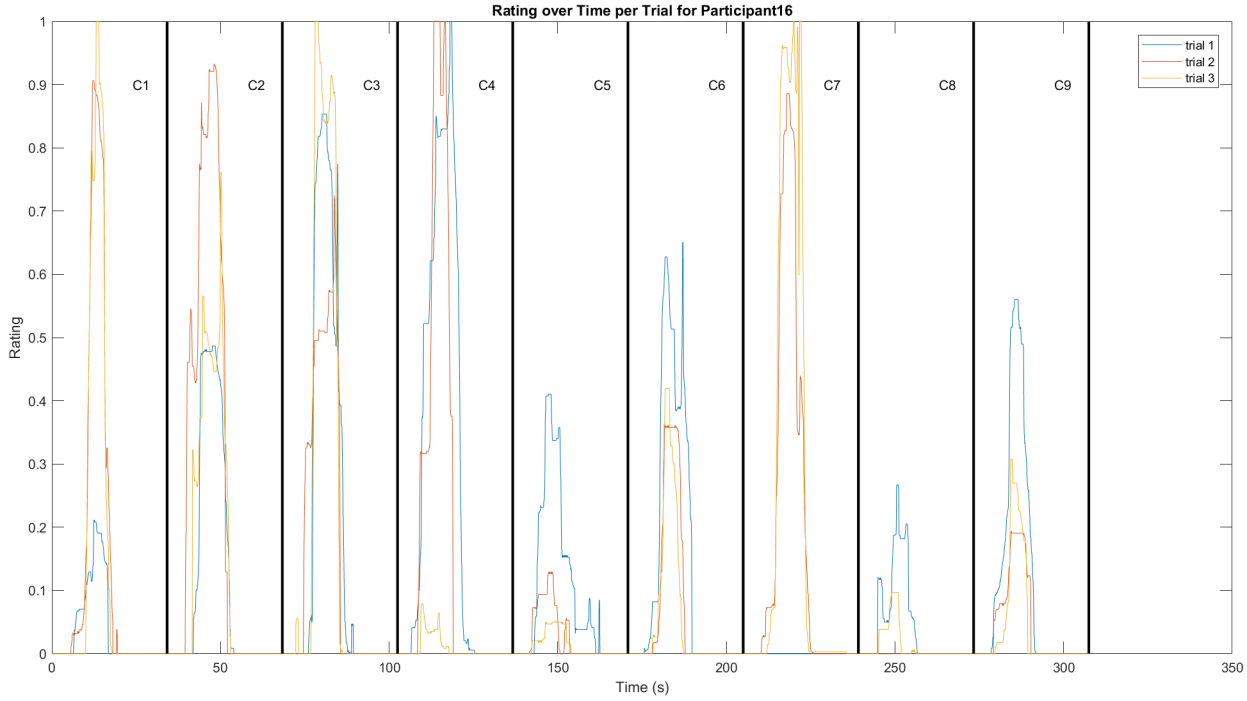


Figure D.31: Rating over time per trial for participant 16

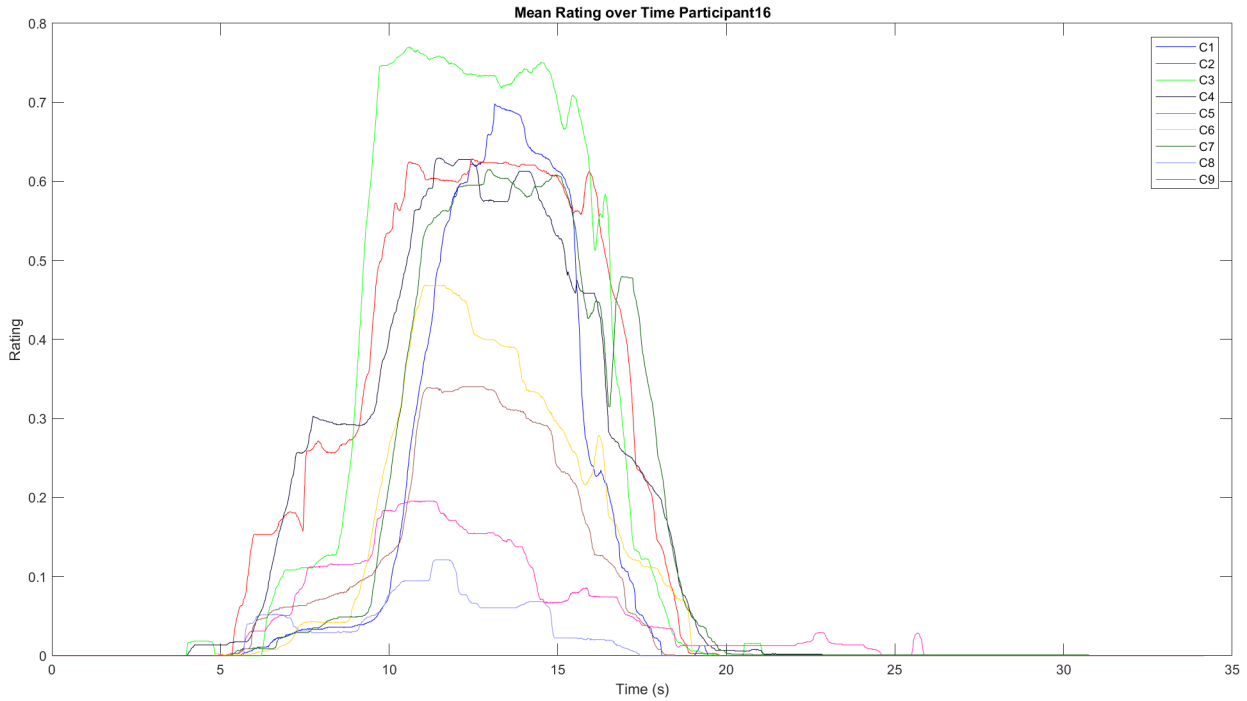


Figure D.32: Mean rating over time for participant 16

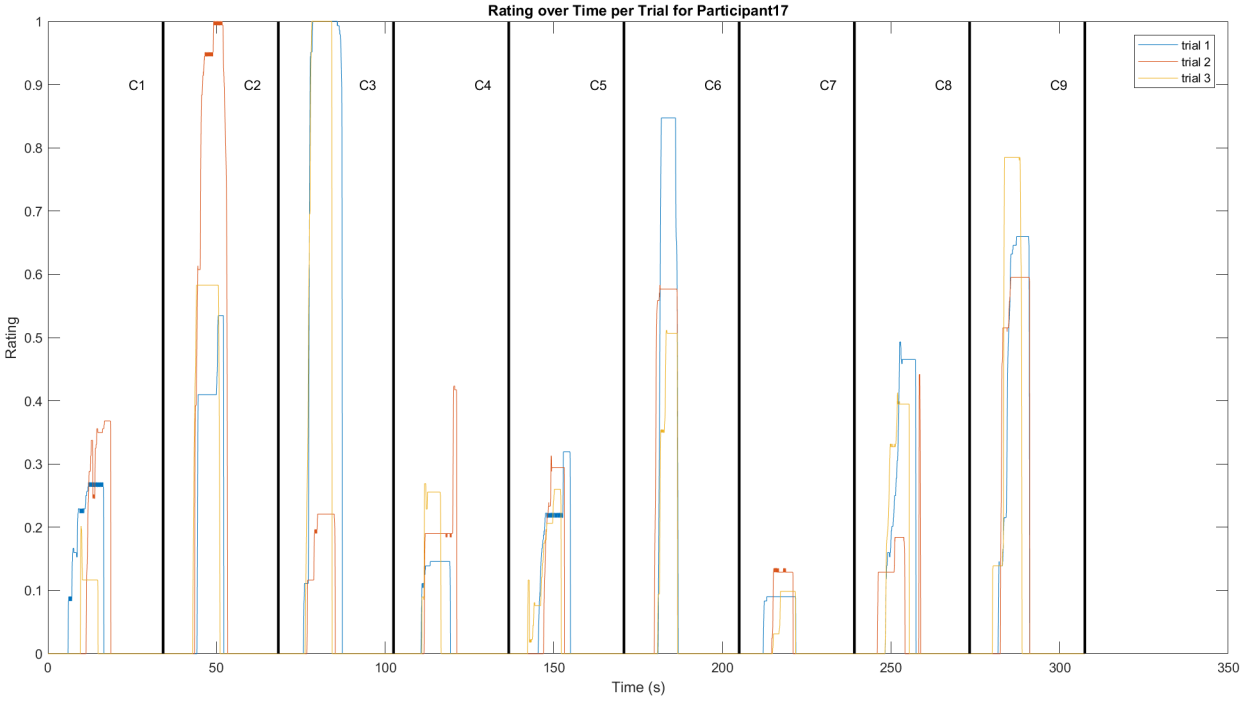


Figure D.33: Rating over time per trial for participant 17

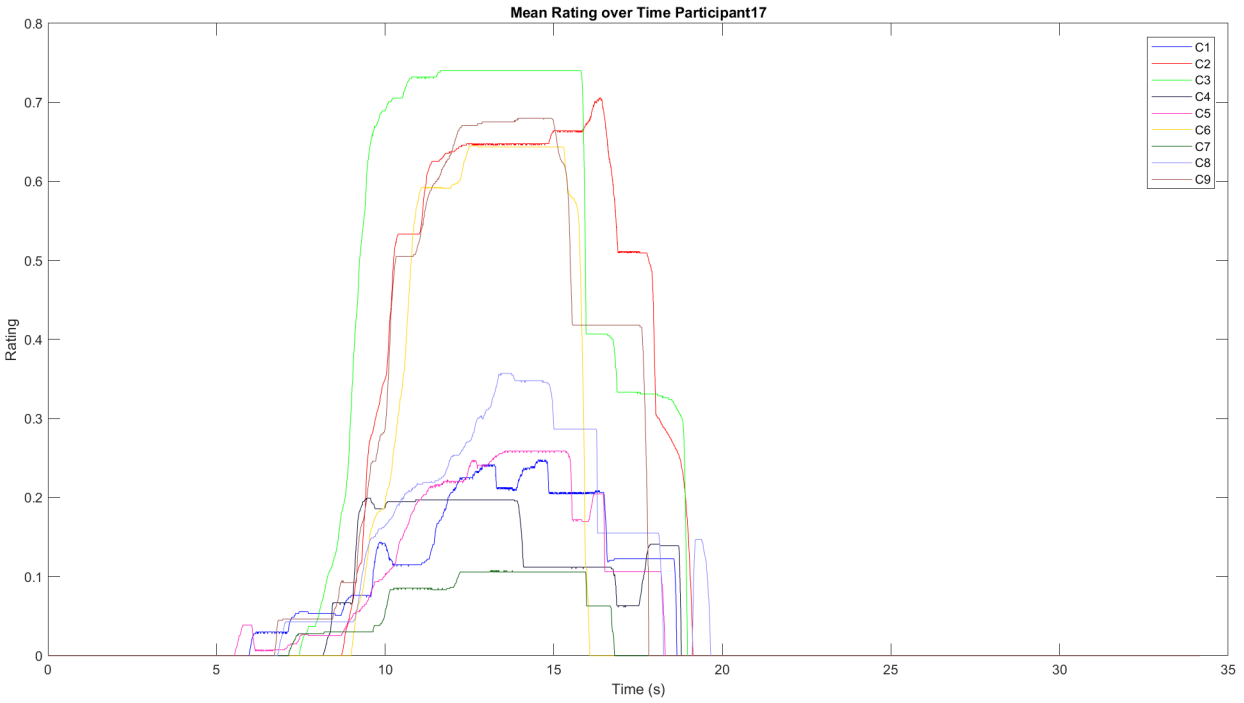


Figure D.34: Mean rating over time for participant 17

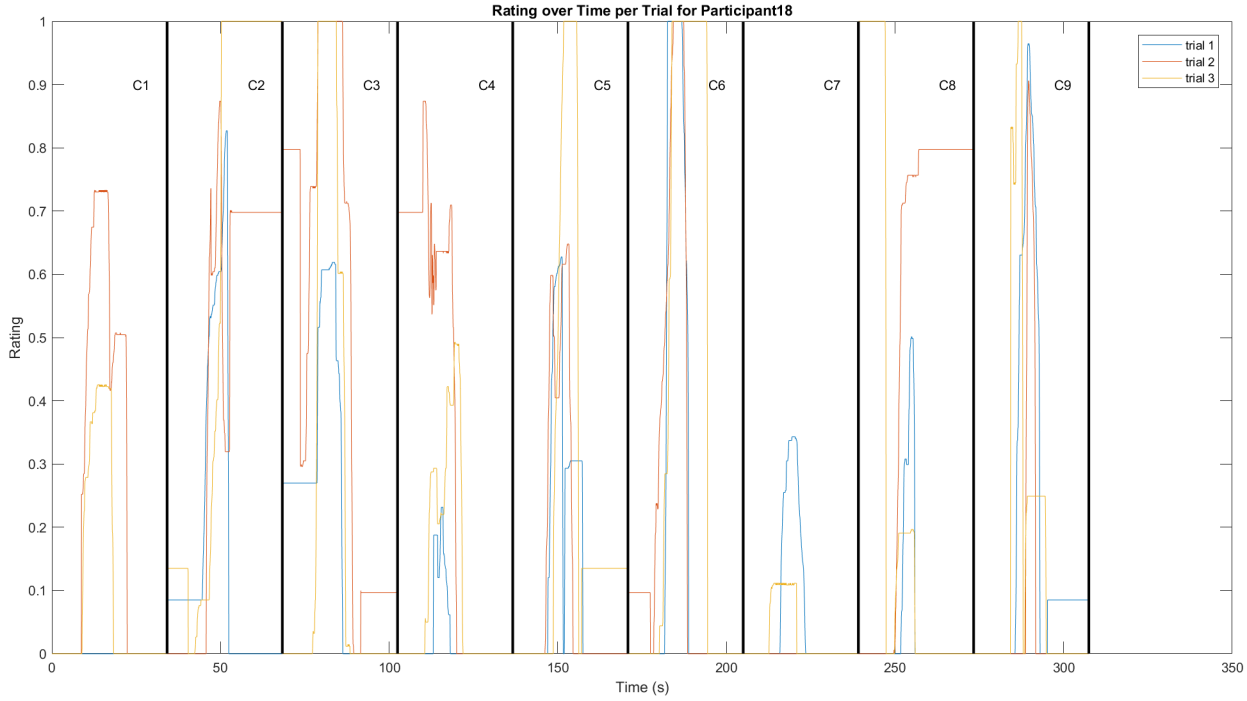


Figure D.35: Rating over time per trial for participant 18

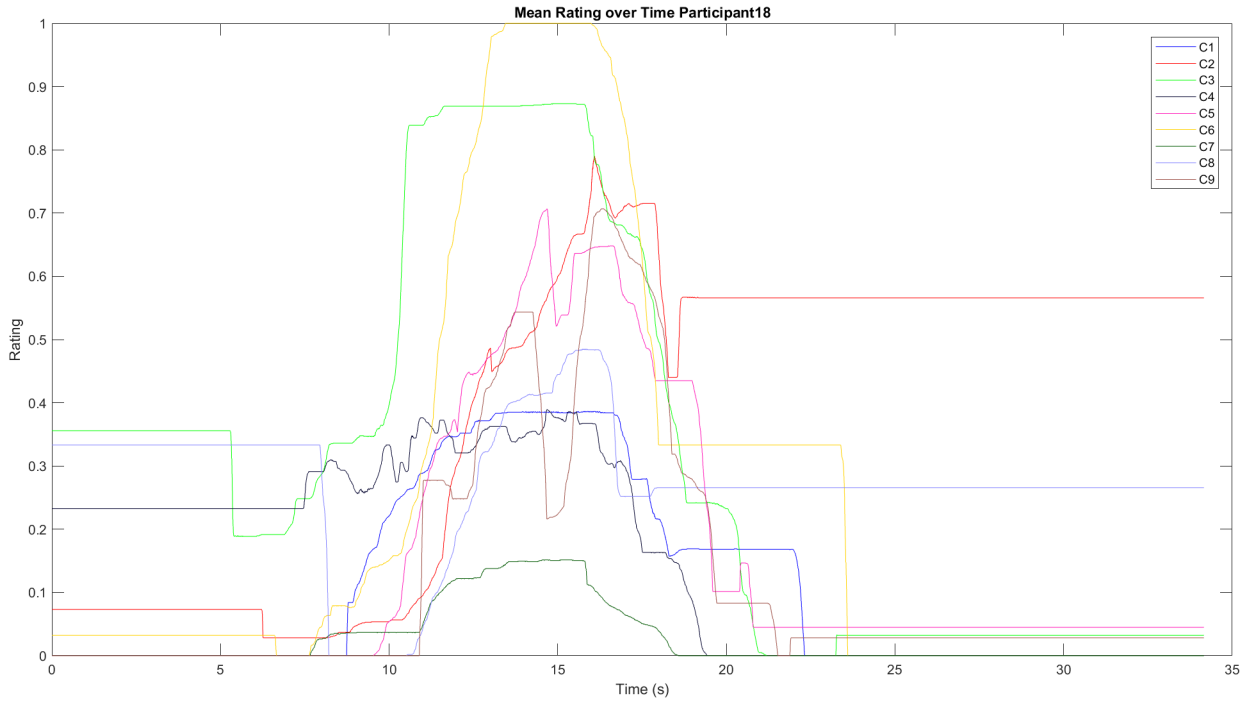


Figure D.36: Mean rating over time for participant 18

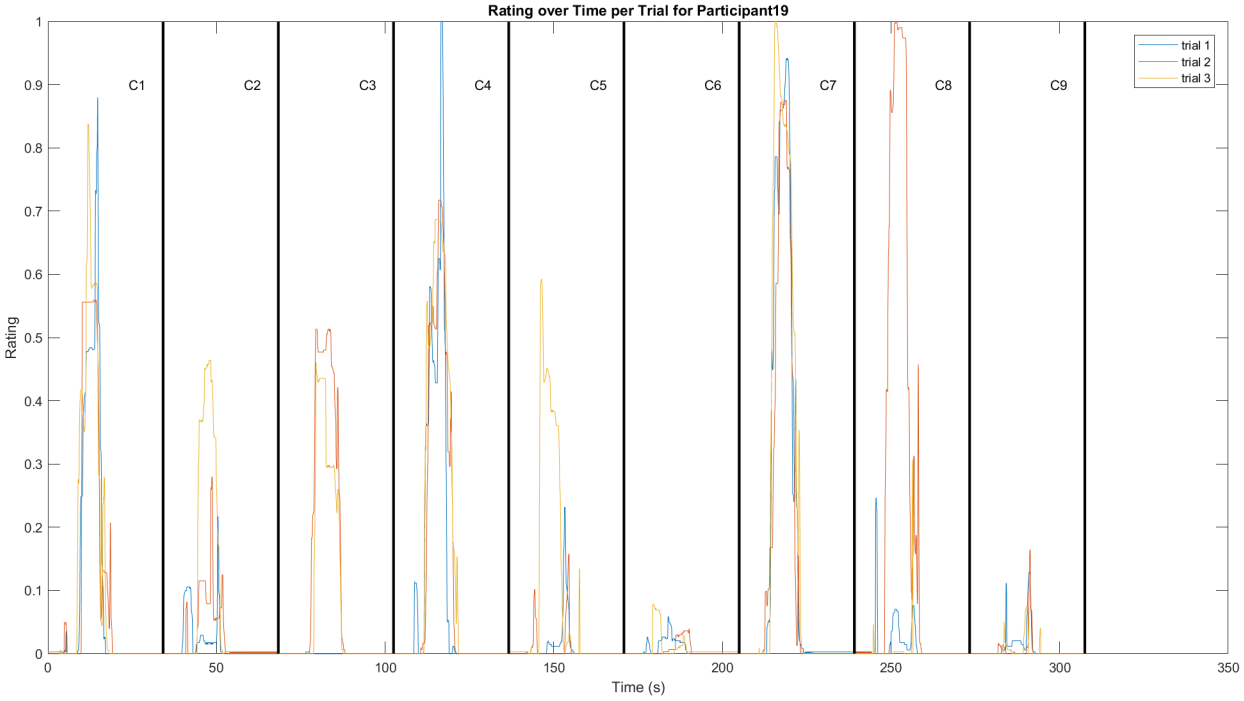


Figure D.37: Rating over time per trial for participant 19

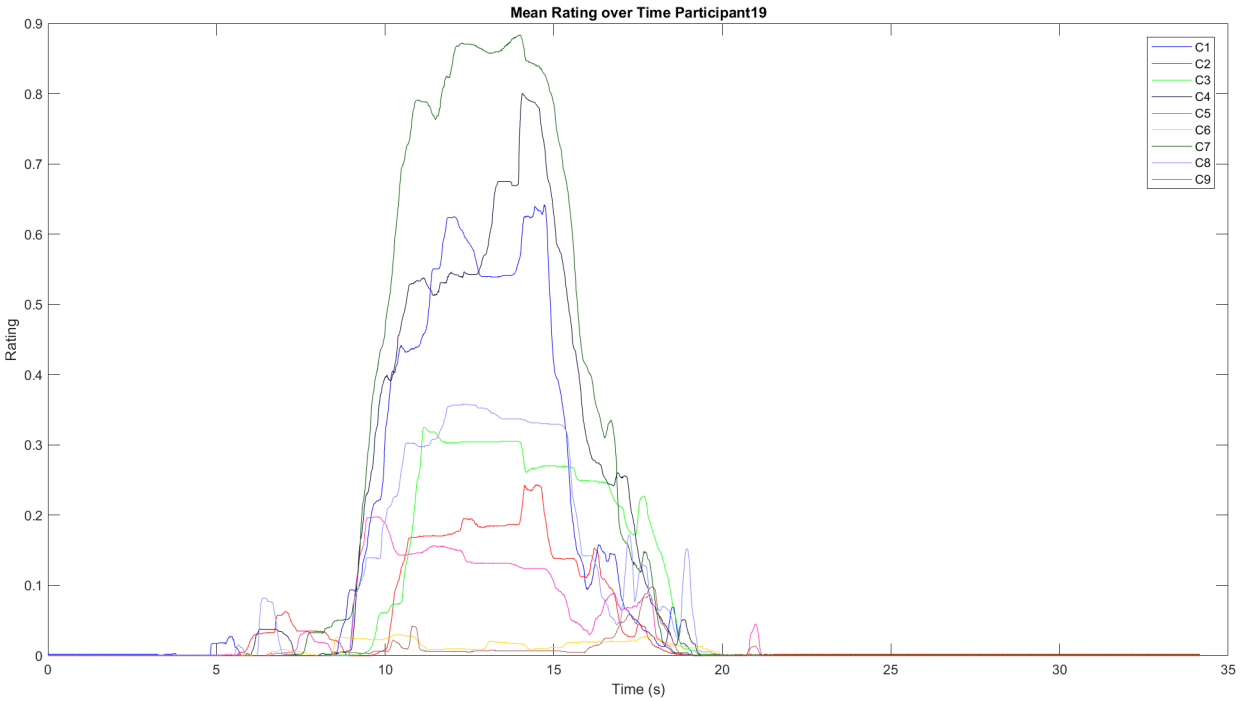


Figure D.38: Mean rating over time for participant 19

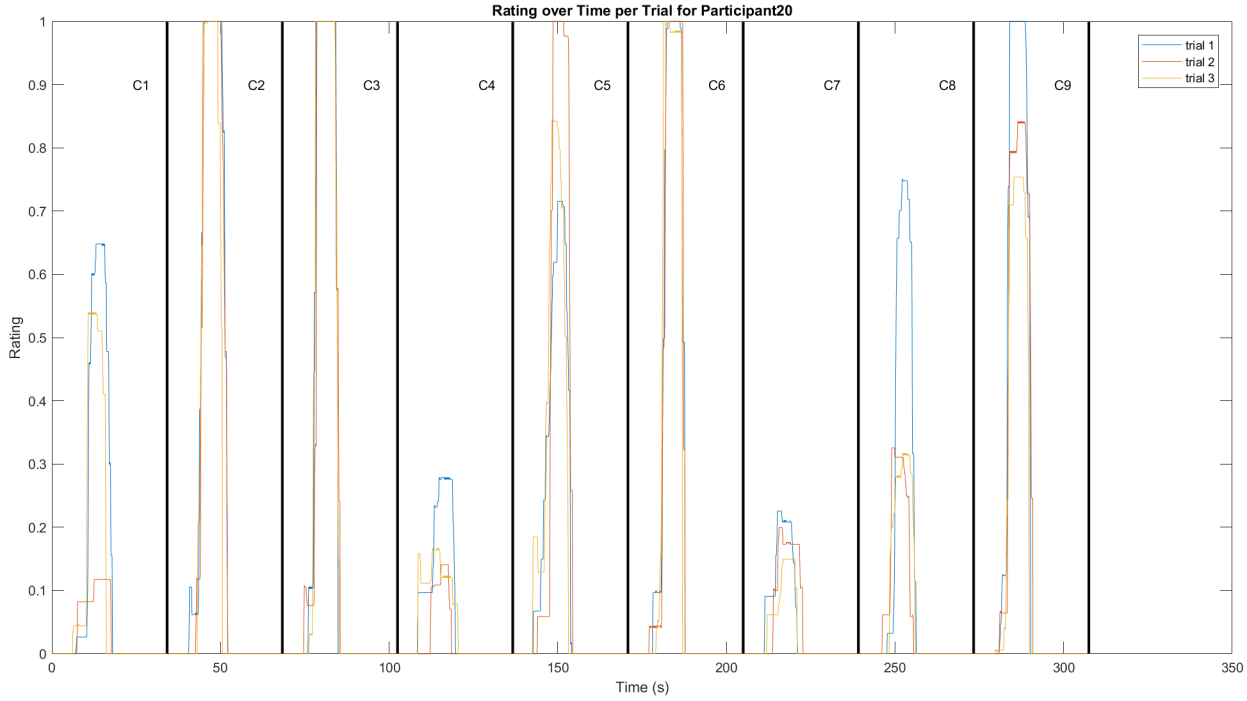


Figure D.39: Rating over time per trial for participant 20

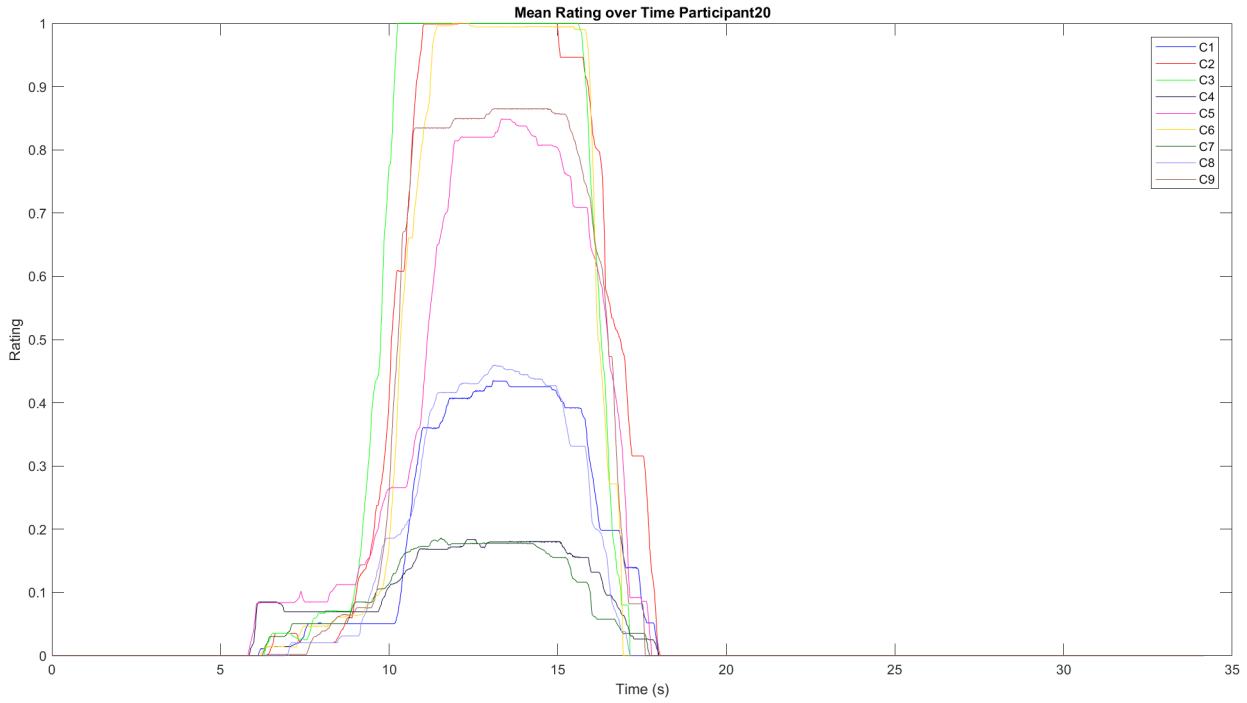


Figure D.40: Mean rating over time for participant 20