Model Predictive Control-based Motion Cueing for Aircraft Upset Simulation

Master of Science Thesis

R.M.B. Bakker

May 8, 2023

TU Delft

Model Predictive Control-based Motion Cueing for Aircraft Upset Simulation

Master of Science Thesis

by

R.M.B. Bakker

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday May 8, 2023 at 12:30 PM.

Student number:	4367138	
Thesis committee:	Prof. dr. ir. M. Mulder,	TU Delft, AE, Control & Simulation
	Dr. ir. D.M. Pool,	TU Delft, AE, Control & Simulation
	Ir. O. Stroosma,	TU Delft, AE, Control & Simulation
	Dr. B. Shyrokau,	TU Delft, 3mE, Intelligent Vehicles

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Preface

This thesis report describes my research done over the past year. The thesis can be divided into two phases, a preliminary phase and a final phase. During the preliminary phase, a literature study and offline analysis were performed. During the final phase, the findings of the preliminary thesis were used to design and conduct a human-in-the-loop experiment in TU Delft's SIMONA Research Simulator. The Delft University Environment for Communication and Activation (DUECA) project developed and used for the experiment was called "MPCforUPRT".

The preliminary phase of the thesis can be found in Part III and is already graded for the course AE4020. In this part, in Chapter 1, an introduction is included that presents in detail the research motivation, objectives, and questions. The final phase of the thesis is covered by Part I and Part II, which include a scientific paper and appendices, respectively. The scientific paper contains a pre-experiment analysis, the experiment design, and the experimental findings. The appendices substantiate the work presented in the scientific paper, which includes Python code used for the analysis, experiment instructions, additional relevant experiment data, and extensive experiment results.

During the past year, I have had the unwavering support of many individuals who have made this thesis possible. First of all, to my supervisors, Olaf and Daan, I am grateful for the time and effort you devoted to giving excellent guidance, hearing me during meetings, providing me with very useful feedback, responding quickly to emails, and giving excellent support in the experiment design and preparations. Besides, I am thankful for the direction, feedback, and insights, Max, Rene, and Frank gave me during our meetings and for their reflections on my work, which significantly contributed to the quality of my work.

To my friends, I would like to extend my apologies for the time that I have been away and my appreciation for your understanding. I can assure you that you will see me a lot more often this coming year.

To Stephanie, I am eternally grateful for your love, support, and endless belief in me. Your encouragement, patience, and understanding have been a constant source of motivation, inspiration, and most of all heartwarming comfort throughout this thesis.

To conclude, I am filled with gratitude for my brother and parents, who always support me. My brother for doing nice stuff together and travelling together. My parents, for letting me do the things I want and letting me do these things how I want. I will remain thankful for this my entire life.

Roemer
May 2023

Contents

List of Figures

List of Tables

List of Abbreviations

ATPL Airline Transport Pilot License 107

AUPRTA Airplane Upset Prevention & Recovery Training Aid 106

AW Adaptive Washout 90

CFD Computational fluid dynamics 105, 106

CPL Commercial Pilot License 107

CW Classical Washout viii, x, xi, xvi, 80–82, 89–91, 94, 101, 103, 109–112, 121–129, 133, 138, 139, 147

DERP Design Eye Reference Point 86, 115

DOF Degrees of Freedom 80, 90, 101, 102, 112

EASA European Union Aviation Safety Agency ..... x, 80, 105, 107, 108, 111

FAA Federal Aviation Administration 80

FRS Flight Research Simulator 106

FSTD Flight Simulator Training Device ..... x, 80, 82, 85, 104, 105, 107–109, 111

FSTDs Flight Simulator Training Devices 80, 82, 85, 104, 105, 107, 109, 111

IATA International Air Transport Association 80, 106, 109

ICAO International Civil Aviation Organization 105–107, 109

IOS Instructor Operator Station 104

Ipopt Interior Point Optimizer 114

LGP Lower Gimbal Point ..... xiv, xv, 85, 88, 89

LOC-I Loss of Control In-Flight 80

LTI Linear Time-invariant 101

MBT Maneuver-based Training 109, 111

MCA Motion Cueing Algorithm ..... x, 80–82, 84, 89–91, 93–95, 101–103, 106, 109–114, 118, 126–129, 133

MCC Motion Control Computer 129

MFR Motion Fidelity Rating ..... viii, x, 57, 130–133

MIR Motion Incongruence Rating ..... v–viii, 58–72, 102

MISC MIsery SCale ..... x, 131–133

MPC Model Predictive Control .. viii, x, xvi, 80–82, 90, 94, 95, 98–103, 111, 112, 114, 118–121, 126–129, 133

MPCMSF Model Predictive Control Motion Simulation Framework 29

MPL Multi-crew Pilot License 107

NLP Non-linear Programming 96, 97

OEM Original Equipment Manufacturers 106, 107

OMCT Objective Motion Cueing Test 91, 94

OMSF Offline Motion Simulation Framework 29, 114

PCC Pearson Correlation Coefficient ..... viii, ix, 92, 94, 102, 121–125, 128, 141, 144

PFD Primary Flight Display 129, 130

RMSE Root Mean Square Error ..... viii, ix, 92, 94, 102, 121–125, 128, 141, 144

SBT Scenario-Based Training 109, 111

SRS SIMONA Research Simulator .. v, viii, x, 29, 38, 44, 47, 50, 53, 57, 58, 73–78, 81, 82, 84, 92, 93, 112, 114, 117, 118, 121, 126–129, 132–134

SUPRA Simulation of Upset Recovery in Aviation 104, 105, 107, 110

TU Delft Delft University of Technology ..... viii, 81, 93, 112

UGP Upper Gimbal Point ..... xv, 85, 89, 94, 114

UPRT Upset Prevention and Recovery Training 80–82, 104–112, 126, 128, 129, 133

List of Symbols

Greek symbols

α	Angle of attack
β	Angle of sideslip
μ	Sample mean
ω_b	First-order high-pass break frequency
ω_{hp}	Second-order high-pass break frequency
ω_{lp}	Second-order low-pass break frequency
ϕ	Euler roll angle
ψ	Euler yaw angle
σ	Sample standard deviation
$\hat{\boldsymbol{\alpha}}$	Reference angular acceleration
$\boldsymbol{\alpha}$	Angular acceleration vector Simulator output angular acceleration
$\hat{\boldsymbol{\omega}}$	Reference angular velocity
$\boldsymbol{\omega}$	Angular velocity vector Simulator output angular velocity
$\boldsymbol{\Omega}$	Skew-symmetric angular velocity matrix
θ	Euler pitch angle
ζ_{hp}	Second-order high-pass damping ratio
ζ_{lp}	Second-order low-pass damping ratio

Roman symbols

Δt	Sample time
ℓ	Lagrangian
$\hat{y}$	Estimated sample
$\hat{\boldsymbol{f}}$	Reference specific force vector
$\hat{\boldsymbol{y}}$	Reference inertial signal vector
$\boldsymbol{a}$	Inertial acceleration
$\boldsymbol{B}$	Base gimbal position matrix
$\boldsymbol{b}_i$	Position of base gimbal i with respect to the LGP

$\boldsymbol{f}$	Specific force vector Simulator output specific force
$\boldsymbol{g}$	Gravitational acceleration vector
$\boldsymbol{J_q}(\boldsymbol{x})$	Jacobian matrix of the actuator length with respect to the state vector
$\boldsymbol{P}$	Upper gimbal position matrix
$\boldsymbol{p}_i$	Position of upper gimbal i with respect to the UGP
$\boldsymbol{q}$	Actuator length matrix
$\boldsymbol{r}_{\boldsymbol{B},\boldsymbol{A}}$	Position of point A on body B
$\boldsymbol{R}$	Reference frame rotation matrix
$\boldsymbol{r}$	Position of the UGP with respect to the LGP
$\boldsymbol{u}$	Control input vector
$\boldsymbol{W_u}$	Control input cost weight matrix
$\boldsymbol{W_x}$	State cost weight matrix
$\boldsymbol{W_y}$	Inertial signal cost weight matrix
$\boldsymbol{x}$	State vector
$\boldsymbol{y}$	Simulator output inertial signal vector
$\mathbf{u}$	Discretised control input vector
$\mathbf{w}_\alpha$	Angular acceleration weight vector
$\mathbf{w}_\omega$	Angular velocity weight vector
$\mathbf{w}_f$	Specific force weight vector
$\mathbf{X}$	Discretised internal state vector
$\mathbf{x}$	Discretised state vector
$\mathbf{y}$	Discretised reference inertial signal vector
u_i	Discretised control input at interval i
w_u	Control input weight
w_x	State weight
x_i	Discretised state at interval i
B_i	Base gimbal i
f	Specific force
F_A	Aircraft reference frame equivalent to F_S
F_C	Aircraft reference frame equivalent to F_D
F_D	Platform reference frame
F_I	Inertial reference frame
F_{P_a}	Aircraft pilot reference frame equivalent to F_{P_s}
F_{P_s}	Simulator pilot reference frame
F_S	Simulator reference frame
i	ith interval
J	Cost

K	CW filter channel gain Number of MPC iterations
k	kth MPC iteration
N	Sample size Number of discretisation intervals Number of samples in the MPC prediction horizon
n	Load factor Vector size
p	Body angular roll rate (angular velocity around the x-axis)
P_i	Upper gimbal i
q	Body angular pitch rate (angular velocity around the y-axis) Actuator length
r	Body angular yaw rate (angular velocity around the z-axis)
t	Time variable
u	Body velocity in x-direction Control input variable
v	Body velocity in y-direction
V_f	Final or terminal cost
V_{TAS}	True airspeed
w	Body velocity in z-direction Set of optimal control decision variables
X	Internal state
x	State variable
y	True sample

Part I

AIAA Paper:

Evaluation of Optimization-based Motion Cueing for Simulating Upset and Stall Maneuvers

Evaluation of Optimization-based Motion Cueing for Simulating Upset and Stall Maneuvers

Roemer M.B. Bakker*
Delft University of Technology, Delft, Zuid-Holland, 2628 CD

Representing the aircraft with adequate motion cues during the simulation of upset and stall maneuvers remains challenging. This paper discusses the potential of optimization-based motion cueing algorithms for upset and stall simulation. An analysis is performed to compare three offline model predictive control-based algorithms with varying prediction horizon lengths and prediction strategies against a baseline classical washout algorithm. Three flight scenarios flown with a Cessna Citation II Laboratory Aircraft are used: a symmetric stall, a large roll maneuver, and a scenario with both nose-low and nose-high upsets including large bank angles. Compared to a classical washout algorithm, the analysis shows up to 147.7 % increase in objective motion cueing fidelity for two of the three optimization-based algorithms. Two algorithms, an optimization-based and the baseline classical washout, were assessed on perceived motion fidelity for the same scenarios by ten pilots in an experiment in the SIMONA Research Simulator. All ten pilots experienced identical offline generated scenarios, meaning that no control input was required during the experiment. Instead, the pilots were asked to continuously rate the motion cueing quality with a rating knob, yielding the subjective motion incongruence rating. In addition, a single motion fidelity rating, substantiated with comments, was given. A significant increase in perceived motion fidelity was found for the large roll scenario. No significant differences were found for the remaining two scenarios. The results imply, assuming that sufficient reference motion prediction correctness can be achieved, that optimization-based motion cueing algorithms have the potential to achieve significantly better motion cueing quality compared to filter-based algorithms.

Nomenclature

f	=	specific force, m/s^2
$\boldsymbol{f}$	=	specific force vector, m/s^2
f_c	=	continuous time function, -
h	=	height, m
K_{cw}	=	motion filter gain, -
p	=	roll rate, deg/s
q	=	pitch rate, deg/s
q_i	=	simulators' ith actuator length, m
$\boldsymbol{q}$	=	simulator actuator length vector, m
r	=	yaw rate, deg/s
t	=	time, s
t_s	=	sampling time, s
T_H	=	prediction horizon length, s
T_S	=	scenario length, s
$\boldsymbol{u}$	=	simulator control input, m/s^2 and deg/s
V_{IAS}	=	indicated airspeed, m/s
$V_t(x)$	=	terminal cost function, -
w	=	weight, -
$\mathbf{w}$	=	weight vector, -
$\boldsymbol{W}$	=	weight matrix, -

*M.Sc. Student, Control & Simulation, Faculty of Aerospace Engineering.

$\boldsymbol{x}$ = simulator state, m and deg and m/s and deg/s
$\boldsymbol{x}_r$ = simulator position substate, m and deg
$\boldsymbol{x}_v$ = simulator velocity substate, m/s and deg/s
$\boldsymbol{y}$ = simulator inertial signal, m/s^2 and deg/s and deg/s^2
$\hat{\boldsymbol{y}}$ = reference inertial signal, m/s^2 and deg/s and deg/s^2
α = angle of attack, deg
α_{Cr} = Cronbach's alpha, -
$\boldsymbol{\alpha}$ = angular acceleration vector, rad/s^2
ζ_{hp} = second-order high-pass motion filter damping coefficient, -
ζ_{lp} = second-order low-pass motion filter damping coefficient, -
θ = euler pitch angle, deg
ϕ = euler roll angle, deg
ψ = euler yaw angle, deg
ω = angular rate, rad/s or deg/s
$\boldsymbol{\omega}$ = angular rate vector, rad/s or deg/s
ω_b = third-order high-pass motion filter break frequency, rad/s
ω_{hp} = second-order high-pass motion filter break frequency, rad/s
ω_{lp} = second-order low-pass motion filter break frequency, rad/s

Abbreviations

CST = "Constant" model predictive control algorithm
CW = classical washout
CWO = baseline classical washout algorithm
DOF = degrees-of-freedom
FSTD = Flight Simulator Training Device
LOC-I = Loss of Control In-Flight
MCA = motion cueing algorithm
MFR = motion fidelity rating
MIR = motion incongruence rating
MPC = model predictive control
OCP = optimal control problem
OMSF = Offline Motion Simulation Framework
ORC = "Oracle" model predictive control algorithm
PCC = Pearson correlation coefficient
PFT = "Perfect" model predictive control algorithm
REF = Reference motion inertial signal
RMSE = root mean square error
SRS = SIMONA Research Simulator
UPRT = Upset Prevention and Recovery Training

I. Introduction

Accidents that are caused by Loss of Control In-Flight (LOC-I) remain one of the largest contributors to worldwide commercial aviation fatalities according to the International Air Transport Association [1]. LOC-I accidents include situations in which the flight crew was unable to maintain control of the aircraft, leading to an unrecoverable deviation from the intended flight path. Recommendations made in 2009 by the Royal Aeronautical Society's International Committee for Aviation Training in Extended Envelopes intended to decrease the number of LOC-I accidents were adopted by the International Civil Aviation Organization [2, 3] and resulted in an addition of Upset Prevention and Recovery Training (UPRT) in the pilot training curriculum as stated by European Union Aviation Safety Agency [4].

UPRT is partially performed in Flight Simulator Training Devices (FSTDs), which typically include a hexapod motion system driven by some form of a classical washout (CW) motion cueing algorithm (MCA) [5]. However, the current options in FSTDs are limited and special care should be taken to prevent negative training due to the simulator exceeding its fidelity envelope [6]. One of the difficulties of UPRT in FSTDs is representing the aircraft

motion with realistic motion cues [7]. The challenges of upset and stall motion cueing arise mainly due to the much larger amplitude and longer duration aircraft accelerations compared to normal flight, potentially leading to larger and sustained specific forces and angular rates to be cued. The first challenge is to leave sufficient motion space for other cues when sustained specific forces are cued [8]. The second challenge is the minimization of false specific force cues when large-amplitude angular motion is cued [8, 9]. The third and final challenge is to minimize the false cues that appear due to high-frequency washout of a CW MCA [9]. Research intended to improve motion cueing for upset and stall simulation [7–13] did not lead to significant industry changes or moving away from the CW MCA.

An upcoming trend in the automotive simulator research industry is the use of MCAs based on Model Predictive Control (MPC) [14–21]. This type of optimization-based MCA uses a reference motion to find the optimal simulator control input for a certain horizon and set of constraints, i.e., an optimal control problem (OCP). To do this, every time step, a reference motion prediction should be made by predicting the driver or pilot control input, or the specific forces and angular motion directly. With this prediction, the OCP is solved by the algorithm, resulting in a control input trajectory of equal length of the prediction horizon. In this way, the algorithm can anticipate future aircraft motion and better utilize the available simulator motion space, which could lead to higher motion cueing fidelity. However, this type of MCA is not yet applied to flight simulation or upset and stall simulation in particular.

The specific investigation addressed in this paper is about exploring the potential and usefulness of MPC-based MCAs for use in upset and stall simulation, with as main objective to improve motion cueing fidelity. For this, an offline MPC-based MCA is developed that is compared to a baseline CW MCA in a pre-experiment objective fidelity analysis. Since the algorithm is deployed in an offline fashion, the total duration of the reference motion is known and best-case motion cueing can be explored. Moreover, the algorithm does not need to adhere to real-time performance requirements, which largely simplifies the implementation. Three versions of the MPC-based MCA are used in the analysis, either assuming perfect knowledge or no knowledge of the reference motion. Methods to predict aircraft motion are, therefore, not included in this paper. After the analysis, an experiment with ten pilots was conducted in TU Delft's SIMONA Research Simulator (SRS), assessing the perceived motion cueing fidelity. The pilots assessed two MCAs, a baseline CW MCA and an MPC-based MCA with perfect reference motion knowledge, for three different flight test scenarios previously performed on the TU Delft/NLR PH-LAB Cessna Citation II laboratory aircraft. The first scenario was a symmetric stall including buffet and recovery. The second scenario was a large roll maneuver and the third scenario included multiple nose-high and nose-low upsets, including large roll angles.

The structure of the paper is as follows. Section II presents the algorithm design and in Section III the baseline CW and various MPC-based MCAs are introduced and analyzed. The experiment design is provided in Section IV, followed by the results and discussion in Section V and Section VI, respectively. The paper's conclusion is included in Section VII.

II. The model predictive control-based motion cueing algorithm

The software implementation of the MPC-based MCA used in this study builds upon the Offline Motion Simulation Framework (OMSF) [22, 23]. This Python-based library is designed to optimize simulator trajectories for a certain reference motion, hence, an optimal control framework. Furthermore, the software tool allows for simulator design parameters optimization, however, the latter is not relevant to this study. To efficiently solve the numerical optimization problems, the OMSF uses the direct collocation parameterization method [24] to transform a continuous-time optimal control problem (OCP) into discrete time. The OMSF depends on the CasADi symbolic numeric optimization software framework [25] in combination with the Ipopt non-linear optimization software [26]. Moreover, the Ipopt-compatible Harwell Subroutine Library high-speed solvers [27] were used in this study. Finally, the OMSF allows for integrating human-perception models, such as vestibular models that try to mimic the otoliths and semi-circular canals [28]. However, this functionality is not used during this study, because it is not yet known for what such a model may be helpful. Moreover, assumptions in these models will affect the MPC-based MCA, and not the baseline classical washout.

To use the OMSF as an offline MPC-based MCA framework for the SIMONA Research Simulator (SRS), three main steps are required. First, the offline implementation of the MPC algorithm is explained in Subsection II.A. Secondly, the geometry and workspace limitations of the system to be studied, the SRS, are defined in Subsection II.B. Thirdly, in Subsection II.C, the system definition is used to formulate the OCP that is included in the MPC algorithm.

A. Offline model predictive control scheme

The offline MPC algorithm routine requires the discretization of the reference motion scenario, $[0, T_S]$, where T_S is the scenario duration in seconds, into K intervals of the form $[t_k, t_{k+1}]$. Note that due to the offline nature of the

MPC in this study, T_S, and thus K, are known. For every kth interval, an OCP is solved. This requires discretization of the OCP with length T_H, where T_H is the so-called prediction horizon length, into N intervals of the form $[t_i, t_{i+1}]$. Note that in this study, intervals $[t_k, t_{k+1}]$ and $[t_i, t_{i+1}]$ will have an equal length in time, meaning that the intervals are perfectly aligned. After solving the OCP of the current time interval $[t_k, t_{k+1}]$, the simulator control input and resulting simulator state of the next time interval, $x_{t_{k+1}}$, are stored. The simulator state at $x_{t_{k+1}}$ is subsequently set as the initial condition for the OCP of the next time interval $[t_{k+1}, t_{k+2}]$. A schematic representation of this offline MPC routine is depicted in Figure 1. Note that when the prediction horizon length, T_H, is set to equal the scenario length, T_S, a single MPC iteration is obtained, resulting in a single and overall OCP.

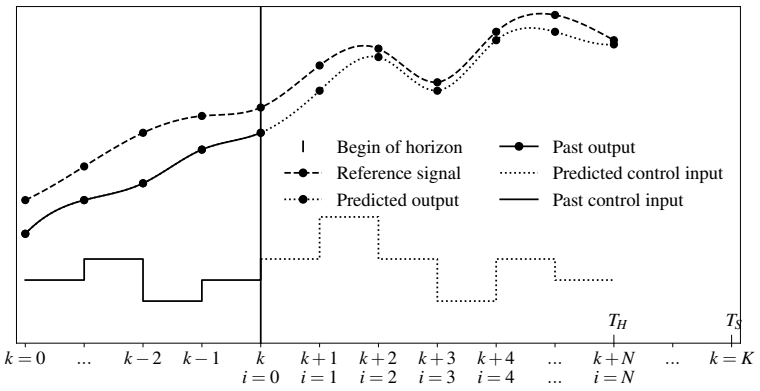


Fig. 1 Schematic representation of the offline MPC algorithm.

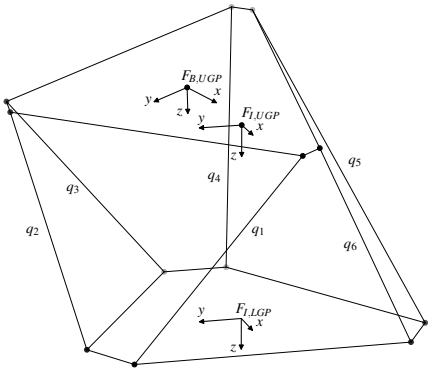


Fig. 2 Schematic hexapod system.

B. System definition

A schematic representation of the SRS's hexapod motion system geometry and reference frames is shown in Figure 2. The simulator state, control input, and output definitions used in the MPC-based MCA resemble the motion software of the SRS [29]. In this way, the offline generated simulator state and control input are used to directly playback the motion in the SRS during the experiment presented in Section IV. The simulator state captures the simulator position, attitude, velocity, and attitude rate:

$$\boldsymbol{x} = [\boldsymbol{x}_r, \boldsymbol{x}_v]^\top \in \mathbb{R}^{12} \quad \text{where} \quad \boldsymbol{x}_r = [x, y, z, \phi, \theta, \psi]^\top \quad \text{and} \quad \boldsymbol{x}_v = [\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^\top, \tag{1}$$

and where all state variables are defined in the inertial frame with its origin in the Upper Gimbal Point (UGP) of the SRS in the neutral position, $F_{I,UGP}$ in Figure 2, meaning 2.39 m above the Lower Gimbal Point (LGP) of the SRS [29]. The simulator control input consists of linear accelerations and angular accelerations:

$$\boldsymbol{u} = [\ddot{x}, \ddot{y}, \ddot{z}, \dot{p}, \dot{q}, \dot{r}]^\top, \tag{2}$$

where the linear accelerations are defined in the same inertial frame as used for the simulator state, $F_{I,UGP}$, and the angular accelerations are defined in a moving body frame with its origin attached to the UGP of the SRS, note $F_{B,UGP}$ in Figure 1. The output inertial signal as a result of the simulator motion is defined as follows:

$$\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{u}) = [\boldsymbol{f}^\top, \boldsymbol{\omega}^\top, \boldsymbol{\alpha}^\top]^\top \in \mathbb{R}^9, \tag{3}$$

where the specific forces, $\boldsymbol{f}$, the angular rates, $\boldsymbol{\omega}$, and angular accelerations, $\boldsymbol{\alpha}$, are all defined in a moving body frame with its origin attached to the Design Eye Reference Point (DERP) of the SRS, located 1.2075 m above and parallel to the UGP and thus $F_{B,UGP}$ [29].

As required for the optimal control formulation presented next in Subsection II.C, two more vectors are introduced. First, the reference inertial signal, which captures the aircraft motion that is attempted to be replicated by the simulator:

$$\hat{\boldsymbol{y}} = [\hat{\boldsymbol{f}}^\top, \hat{\boldsymbol{\omega}}^\top, \hat{\boldsymbol{\alpha}}^\top]^\top \in \mathbb{R}^9, \tag{4}$$

which is defined in a moving body frame with its origin attached to the pilot station. Secondly, the simulator actuator length vector:

$$\boldsymbol{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^\top, \tag{5}$$

defining the lengths of the six actuators, q_1 to q_6, of the SRS's hexapod motion system as also shown in Figure 2.

C. Optimal control formulation

The continuous time OCP used in the MPC-based MCA formulation is described by the following equations:

$$\underset{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)}{\text{minimize}} \quad \int_0^{T_H} \ell_c(\boldsymbol{x}(t),\boldsymbol{u}(t))dt + V_t(\boldsymbol{x}(T_H)) \tag{6a}$$

$$\text{subject to} \quad \underline{\boldsymbol{x}}_0 \leq \boldsymbol{x}(0) \leq \overline{\boldsymbol{x}}_0 \tag{6b}$$

$$\dot{\boldsymbol{x}}(t) = f_c(\boldsymbol{x}(t),\boldsymbol{u}(t)), \quad t \in [0, T_H] \tag{6c}$$

$$\underline{\boldsymbol{q}} \leq \boldsymbol{q}(t) \leq \overline{\boldsymbol{q}}, \quad t \in [0, T_H] \tag{6d}$$

$$\underline{\dot{\boldsymbol{q}}} \leq \dot{\boldsymbol{q}}(t) \leq \overline{\dot{\boldsymbol{q}}}, \quad t \in [0, T_H] \tag{6e}$$

$$\underline{\boldsymbol{x}}_{T_S} \leq \boldsymbol{x}(T_S) \leq \overline{\boldsymbol{x}}_{T_S}, \tag{6f}$$

where Eq. (6a) is the objective function which includes the running cost or Lagrangian, ℓ_c, and the terminal state cost V_t, and where Eq. (6c) contains the system dynamics, and where Eqs. (6b), (6d), (6e), and (6f) are the constraint set. The system dynamics, seen in Eq. (6c), are captured in two first-order differential equations. The first differential equation relates the substate $\boldsymbol{x}_r$ with its time derivative, the substate $\boldsymbol{x}_v$. The second differential equation relates the substate $\boldsymbol{x}_v$ with the simulator control input, $\boldsymbol{u}$, by using the time derivative of the relation between body angular rates and Euler rates. No inertial properties are included in the dynamics because sufficient SRS control input tracking is assumed.

1. Objective function

The objective function that is minimized to find the optimal control input, previously shown in Eq. (6a), consists of two parts. The first part is the running cost, called the Lagrangian:

$$\ell_c(\boldsymbol{x}(t),\boldsymbol{u}(t)) = ||\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)||^2_{\boldsymbol{W}_y} + ||\boldsymbol{x}(t)||^2_{\boldsymbol{W}_x} + ||\boldsymbol{u}(t)||^2_{\boldsymbol{W}_u} + ||\dot{\boldsymbol{u}}(t)||^2_{\boldsymbol{W}_{\dot{u}}} \tag{7a}$$

where

$$\boldsymbol{W}_y = \mathbb{1}_{9\times 9} \cdot [\mathbf{w}_f^{\circ 2}, \mathbf{w}_\omega^{\circ 2}, \mathbf{w}_\alpha^{\circ 2}]^\top \tag{7b}$$

$$\boldsymbol{W}_x = \mathbb{1}_{12\times 12} \cdot \mathrm{w}_x \tag{7c}$$

$$\boldsymbol{W}_u = \mathbb{1}_{6\times 6} \cdot \mathrm{w}_u \tag{7d}$$

$$\boldsymbol{W}_{\dot{u}} = \mathbb{1}_{6\times 6} \cdot \mathrm{w}_{\dot{u}}. \tag{7e}$$

In Eq. (7a), the right-hand side depicts the four terms of the Lagrangian. From left to right, the first term is the inertial signal incongruence cost and penalizes the difference between the reference inertial signal, $\hat{\boldsymbol{y}}$, and the simulator output inertial signal, $\boldsymbol{y}$. The inertial signal weight matrix, $\boldsymbol{W}_{\boldsymbol{y}}$, depicted in Eq. (7b), allows for individual weight setting for each inertial signal component. The remaining three terms are the simulator state cost, the control input cost, and the change of control input cost, respectively. In Eqs. (7c), (7d), and (7e), the corresponding weight matrices are shown, where all three matrices have each a single weight for all components of, respectively, the state, control input, and control input derivative.

The second term in Eq. (6a), $V_t(\boldsymbol{x}(T_H))$, is the terminal cost and penalizes the predicted simulator deviation from its neutral position at T_H. The terminal cost is described as:

$$V_t(\boldsymbol{x}(T_H)) = ||\boldsymbol{x}(T_H)||^2_{\boldsymbol{W}_{x,T_H}} \tag{8a}$$

where

$$\boldsymbol{W}_{x,T_H} = \mathbb{1}_{12\times 12} \cdot [\mathbf{w}_{x_r,T_H}^{\circ 2}, \mathbf{w}_{x_v,T_H}^{\circ 2}]^\top, \tag{8b}$$

and where the terminal state weight matrix, $\boldsymbol{W}_{x,T_H}$, consist of two weights, $\mathbf{w}_{x_r,T_H}$ and $\mathbf{w}_{x_v,T_H}$, which allows for setting different weights for the terminal simulator position and attitude, and for the terminal simulator velocity and attitude rate. This term forces the simulator back to the neutral position at the end of the OCP, which increases stability when the optimal control formulation is used in an MPC algorithm [24].

2. Constraints

Four constraints, previously presented in Eq. (6), are included in the OCP formulation. The initial condition constraint, defined in Eq. (6b), allows for setting the initial simulator state and has a key function in the MPC algorithm explained next in Subsection II.A. The actuator length and velocity constraints, seen in Eqs. (6d) and (6e), respectively, ensure that the simulator stays within its physical limits. Lastly, the final state constraint, defined in Eq. (6f), is specific to the offline MPC algorithm and defines the final state of the simulator when the end of the motion scenario is reached. This constraint ensures that the simulator will not end the scenario with high velocities, that possibly drive the simulator onto its physical limits. The numeric values used for all constraints are presented in Table 1. Note that the simulator is enforced to start in its neutral position and with zero velocity at the beginning of each scenario (see $\boldsymbol{x}_{r,0}$ and $\boldsymbol{x}_{v,0}$ in Table 1). However, the final condition values only enforce the simulator to have zero velocity and leave the end position to an arbitrary position within its available motion space (see $\boldsymbol{x}_{r,T_S}$ and $\boldsymbol{x}_{v,T_S}$ in Table 1). The actuator minimum and maximum length constraints, note $\boldsymbol{q}$ in Table 1, are set to 8 cm within the SRS's physical actuator limits [30], which is 3 cm within the SRS's actuator safety buffers [31], and 1 cm within the SRS's motion limiter software limits [29]. The maximum actuator velocity, see $\boldsymbol{q}$ in Table 1, is set to 0.75 m/s. Above this actuator velocity, the SRS's input tracking performance deteriorates significantly.

Table 1 Constraints used in the MPC formulation.

	Lower limit	**Upper limit**
$\boldsymbol{x}_{r,0}$	$[0\,\text{m}, 0\,\text{m}, 0\,\text{m}, 0\,\text{rad}, 0\,\text{rad}, 0\,\text{rad}]^\top$	$[0\,\text{m}, 0\,\text{m}, 0\,\text{m}, 0\,\text{rad}, 0\,\text{rad}, 0\,\text{rad}]^\top$
$\boldsymbol{x}_{v,0}$	$[0\,\text{m}, 0\,\text{m}, 0\,\text{m}, 0\,\text{rad}, 0\,\text{rad}, 0\,\text{rad}]^\top$/s	$[0\,\text{m}, 0\,\text{m}, 0\,\text{m}, 0\,\text{rad}, 0\,\text{rad}, 0\,\text{rad}]^\top$/s
$\boldsymbol{x}_{r,T_S}$	$[-0.981\,\text{m}, -1.031\,\text{m}, -0.636\,\text{m}, \frac{-25.9\pi}{180}\,\text{rad}, \frac{-23.7\pi}{180}\,\text{rad}, \frac{-41.6\pi}{180}\,\text{rad}]^\top$	$[1.259\,\text{m}, 1.031\,\text{m}, 0.678\,\text{m}, \frac{25.9\pi}{180}\,\text{rad}, \frac{24.3\pi}{180}\,\text{rad}, \frac{41.6\pi}{180}\,\text{rad}]^\top$
$\boldsymbol{x}_{v,T_S}$	$[0\,\text{m}, 0\,\text{m}, 0\,\text{m}, 0\,\text{rad}, 0\,\text{rad}, 0\,\text{rad}]^\top$/s	$[0\,\text{m}, 0\,\text{m}, 0\,\text{m}, 0\,\text{rad}, 0\,\text{rad}, 0\,\text{rad}]^\top$/s
$\boldsymbol{q}$	$[2.161, 2.161, 2.161, 2.161, 2.161, 2.161]^\top$ m	$[3.251, 3.251, 3.251, 3.251, 3.251, 3.251]^\top$ m
$\dot{\boldsymbol{q}}$	$[-0.75, -0.75, -0.75, -0.75, -0.75, -0.75]^\top$ m/s	$[0.75, 0.75, 0.75, 0.75, 0.75, 0.75]^\top$ m/s

III. Pre-experiment analysis

A pre-experiment analysis is performed to map out the differences between the inertial signals produced by the three algorithms described in Subsection III.C. Moreover, to compare these algorithms with a baseline CW MCA (CWO) described next in Subsection III.B and the reference motion inertial signal (REF). Moreover, the analysis serves as a substantiation of the experiment's independent measures and hypothesis, which are presented in Section IV. Two measures are used in this section to analyze the objective fidelity of the inertial signals, the Root Mean Square Error (RMSE) and the Pearson Correlation Coefficient (PCC), which are previously used for evaluating (non-linear) MCAs by [18, 32–34]. A higher PCC (min -1, max 1) indicates higher correlation in shape between the REF inertial signal and the inertial signal produced by the MCA, hence higher objective fidelity. For the RMSE, lower values indicate a lower overall error between the REF and MCA and thus higher objective fidelity.

A. Motion scenarios

1. Symmetric Stall scenario

The first scenario is a simulation-based symmetric stall from [35, 36], which was flown by an autopilot and has only motion in three degrees-of-freedom (DOF). For this study, the stall maneuver is slightly filtered using a third-order low-pass filter and zero-phase digital filtering. The reasoning behind the filtering is that the original high-frequency buffet components are for the motion system limiting motion, requiring the use of input motion scaling affecting the motion in f_x and f_z. Moreover, the high-frequency buffet motion is usually well represented by a CW MCA, and the low-frequency motion included in the stall is of more interest for this study. The course of the scenario is depicted in Figure 3. The load factor in this 15 s scenario ranges between 0.4 g and 1.2 g and a maximum angle of attack of 16 deg is reached. The idea behind this scenario is the assessment of pitch, surge, and heave motion of the different MCAs.

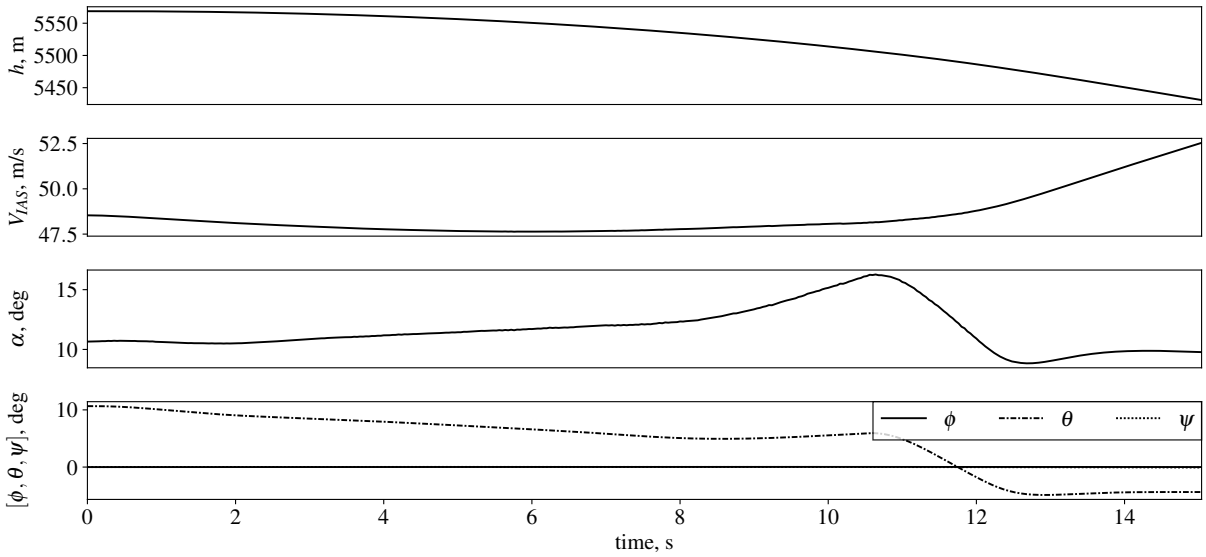


Fig. 3 Aircraft motion for the Symmetric Stall scenario.

2. Large Roll scenario

The second scenario is a flight data-based scenario that includes a single roll upset. The upset itself consisted of a negative 60 deg roll into a positive 50 deg roll, with load factors ranging from 0.7 g to 1.4 g. Before and after the upset, wings-level flight was included to increase the duration of the scenario. The idea behind this scenario is the assessment of roll, sway, and heave motion of the different MCAs. The total scenario duration is 44 s and is depicted in Figure 4.

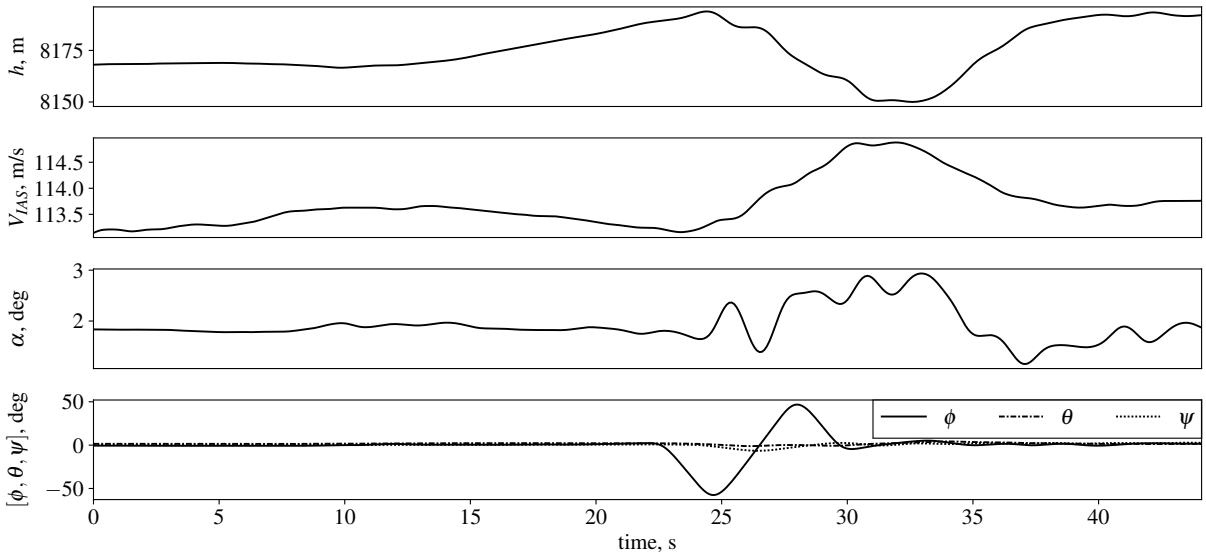


Fig. 4 Aircraft motion for the Large Roll scenario.

3. Upsets scenario

The third scenario consists of three upsets, a coordinated turn, and cruise flight. The first upset, is a 30 deg nose-up including a 125 deg roll maneuver, resulting in load factors of 0.4 g to 2.2 g. After this first upset, a 50 s coordinated turn is flown, followed by 100 s of cruise flight. To conclude the scenario, two nose-low upsets are included with,

respectively, 10 deg and 20 deg nose-low attitudes and positive 60 deg and negative 100 deg roll. Load factors during these upsets range from 0.7 g to 1.7 g. The idea behind this scenario is the assessment of the MCAs for complex six-DOF motion including sustained load factors and large roll rates. The total scenario duration is 350 s and is depicted in Figure 5.

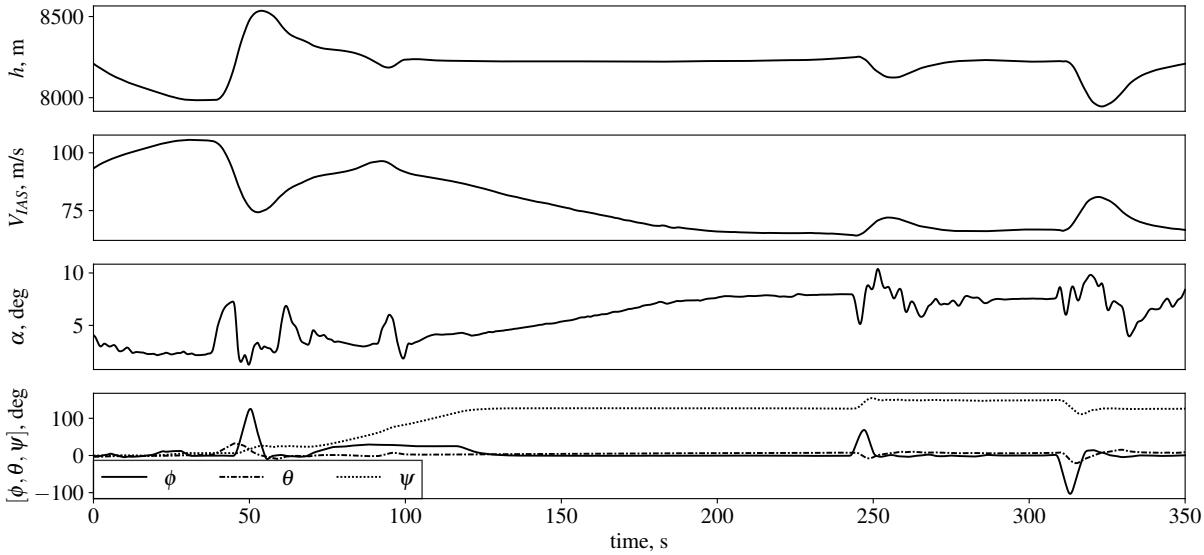


Fig. 5 Aircraft motion for the Upsets scenario.

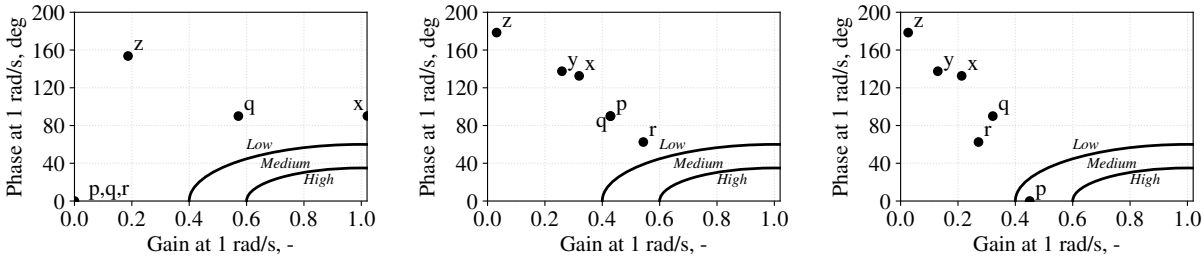
B. Baseline classical washout algorithm

The CW MCA used in this study, referenced as CWO, is the standard SRS CW algorithm and is used in an offline fashion. For each motion scenario, the CWO parameters, presented in Table 2, are optimized such that the SRS stays just within its physical limits. For all scenarios, second-order low-pass filters were used for tilt coordination for f_x and f_y. The parameters used for the Symmetric Stall scenario are from [36] and include a second-order high-pass filter for surge and pitch and a third-order high-pass filter for heave. The gains, K_{cw}, are slightly increased from 0.6 to 0.8.

The CWO parameters for the Large Roll and Upsets scenarios are based on the parameters from [10, 37] and were adapted where required. This resulted in second-order high-pass filters for surge, sway, pitch, and yaw for both the Large Roll and the Upsets scenario and for roll of the Large Roll scenario. Moreover, a third-order high-pass filter for heave for the Large Roll and the Upsets scenarios, and a first-order high-pass filter for roll for the Upsets scenario. The phase and gain values for each scenario are presented in Sinacori-Schroeder plots [38, 39], depicted in Figure 6. In Figure 6, the surge and sway values are the result of combining the high- and low-pass filters into a total specific force transfer function as defined in [10].

Table 2 CWO parameters for the three motion scenarios.

	Scenario 1 - Symmetric Stall						**Scenario 2 - Large Roll**						**Scenario 3 - Upsets**					
	x	y	z	p	q	r	x	y	z	p	q	r	x	y	z	p	q	r
K_{cw}, -	0.8	0.0	0.8	0.0	0.8	0.0	0.6	0.6	0.6	0.6	0.6	0.6	0.4	0.3	0.5	0.45	0.45	0.3
ω_{hp}, rad/s	1.2	0.0	2.0	0.0	1.0	0.0	2.0	2.0	4.0	1.0	1.0	0.7	2.0	2.0	4.0	0.0	1.0	0.7
ζ_{hp}, -	0.7	0.0	0.7	0.0	0.7	0.0	0.7	0.7	1.0	0.7	0.7	0.7	0.7	0.7	1.0	0.0	0.7	0.7
ω_b, rad/s	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.5	0.6	0.0	0.0
ω_{lp}, rad/s	2.4	0.0					2.0	2.5					2.0	2.5				
ζ_{lp}, -	0.7	0.0					0.7	0.7					0.7	0.7				



(a) The Symmetric Stall scenario. **(b) The Large Roll scenario.** **(c) The Upsets scenario.**

Fig. 6 Sinacori-Schroeder plots for the CWO parameters of the three scenarios.

C. Model predictive control-based algorithms

Three offline MPC algorithm variants are used, namely, the "Oracle", the "Perfect", and the "Constant" algorithms. The "Oracle" (ORC) has a prediction horizon length equal to the entire motion scenario. This results in a single MPC iteration, where T_H equals T_S, hence a single OCP. This, in combination with the perfect knowledge of the reference signal, will result in the best obtainable control trajectory for the motion scenario and the chosen set of parameters and weights. Note that this algorithm is not feasible in real-time, since perfect knowledge of the reference signal for the entire motion scenario is required.

The "Perfect" (PFT) assumes perfect knowledge of the reference signal for the entire prediction horizon. This will result in the best obtainable control trajectory obtained through MPC for a certain horizon length. This prediction strategy is not feasible in real-time, since it will never be possible to perfectly predict the future reference signal for the entire prediction horizon.

The "Constant" (CST) assumes knowledge of the reference signal for the current time interval only. The remainder of the horizon is then completed with a constant reference signal that is equal to the current time interval. This prediction strategy is feasible in real-time since it will only require knowledge of the reference signal for the current time interval.

The parameters used for the three algorithms, for each scenario, are summarized in Table 3. Note that the inertial signal weights, $[\mathbf{w}_f, \mathbf{w}_\omega, \mathbf{w}_\alpha]$, and the state and control input weights, w_x and w_u, are equal for all algorithm variants and are based on values from previous studies [18, 19, 23]. The control input derivative weight, $\mathrm{w}_{\dot{\boldsymbol{u}}}$, is applied reservedly and only where necessary. A terminal state weight is only applied to the simulator velocity and attitude rate and is solely required for the PFT and CST. This is due to the terminal state being equal to the final state for the ORC algorithm, hence a final state constraint is present. The sampling time, t_s, was set to 0.01 s, equal to the SRS motion control sample rate, for the ORC algorithm for all three scenarios, as this algorithm will be used in the experiment presented in Section IV. For the Large Roll and the Upsets scenario, the sample time was increased for both the PFT and CST algorithms to 0.1 s, equal to the reference motion sample rate of these two scenarios, to decrease the computation time. For the Symmetric Stall scenario, the sampling time for the PFT and CST algorithms was decreased from 0.01 s, the original reference motion sample time, to 0.05 s, to decrease the computation time, however, retain the high-frequency buffet components.

Table 3 Parameters of the different MPC algorithms for each scenario. Note that the $\cdot$ represents the unit of the respective vector denoted in the weight's subscript.

Scenario	MCA	t_s, s	T_H, s	$\mathbf{w}_f$, s^2/m	$\mathbf{w}_\omega$, s/rad	$\mathbf{w}_\alpha$, s^2/rad	w_x, $\cdot^{-2}$	w_u, $\cdot^{-2}$	$\mathrm{w}_{\dot{u}}$, $\cdot^{-2}$	$\mathbf{w}_{x_{r,T_H}}$, $\cdot^{-1}$	$\mathbf{w}_{x_{v,T_H}}$, $\cdot^{-1}$
Sym. Stall	ORC	0.01	T_S	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0	[0,0,0,0,0,0]	[0,0,0,0,0,0]
	PFT	0.05	6	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.005	[0,0,0,0,0,0]	[1,1,1,1,1,1]
	CST	0.05	2	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.005	[0,0,0,0,0,0]	[1,1,1,1,1,1]
Large Roll	ORC	0.01	T_S	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.005	[0,0,0,0,0,0]	[0,0,0,0,0,0]
	PFT	0.1	6	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.005	[0,0,0,0,0,0]	[1,1,1,1,1,1]
	CST	0.1	2	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.25	[0,0,0,0,0,0]	[1,1,1,1,1,1]
Upsets	ORC	0.01	T_S	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.005	[0,0,0,0,0,0]	[0,0,0,0,0,0]
	PFT	0.1	6	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.005	[0,0,0,0,0,0]	[1,1,1,1,1,1]
	CST	0.1	2	[1,1,1]	[10,10,10]	[1,1,1]	0.01	0.01	0.25	[0,0,0,0,0,0]	[1,1,1,1,1,1]

D. Inertial signal results

1. Symmetric Stall inertial signal

The three-DOF inertial signal, $[f_x, f_z, \omega_y]$, as a result of the four different MCAs and the reference inertial signal (REF) are shown in Figure 7. The corresponding RMSE (Figure 8a) and PCC (Figure 8b) values for all algorithms are shown in Figure 8. First, the CWO undershoots the REF's f_x due to the applied CW filter gain of 0.8, which results in a magnitude difference and an RMSE(f_x) of 0.26 m/s^2. The shape is not affected by this and is well represented by the second-order high-pass filter, leading to a PCC(f_x) of 0.88. In f_z, the inability to cue sustained deviations from -9.81 m/s^2 with a hexapod motion system becomes evident. This results in an RMSE(f_z) of 1.41 m/s^2. Apparently, the third-order high-pass filter for f_z also affects the high-frequency stall buffet shape correlation with the REF, leading to a fair PCC(f_z) of 0.39. In ω_y, at $t \approx 10.5$ s, i.e., where the stall break occurs, the CWO initially goes the wrong way to partly cue the f_x by tilt-coordination, resulting in a large RMSE(ω_y) of 2.17 deg/s and PCC(ω_y) value of 0.57.

Next, the ORC and PFT, note that these algorithms are very similar in all three DOF. This is confirmed by the up to two decimal figures identical RMSE(f_x) of 0.17 m/s^2 and equal PCC(f_x) and PCC(ω_y) values of, respectively, 0.95 and 0.85. The largest, but still minor, differences between the ORC and PFT are seen in f_z, for which the RMSE are, respectively, 1.27 m/s^2 and 1.33 m/s^2. In f_x, the ORC and PFT slightly overshoot the REF to increase the REF tracking in f_z. This results in RMSE(f_x) and PCC(f_x) values of, respectively, 0.17 m/s^2 and 0.95 for both algorithms. Similar to the CWO, in f_z, an offset is present for the ORC and PFT due to identical reasons. Compared to f_x, this results in a large RMSE(f_z) of 1.27 m/s^2 and 1.33 m/s^2 for, respectively, the ORC and PFT. Contrary to the CWO, in ω_y, the ORC and PFT show a response similar in shape to the REF, which is confirmed by the PCC(ω_y) value of 0.85 for both algorithms. This is explained by the fact that the ORC and PFT solely cue the change in f_x at $t \approx 10.5$ s by a linear acceleration in combination with an increase in f_x by tilt-coordination in the two seconds before the stall break.

Finally, the CST, this algorithm clearly performs worse in f_x than the other algorithms, resulting in the highest RMSE(f_x) of 0.43 m/s^2 and lowest PCC(f_x) of 0.29. In f_z, the shape resemblance of the CST with the REF is clearly the worst of all algorithms. This results in a negative correlation and a PCC(f_z) of -0.08. This is explained by the high-frequency buffet components that are not present at all. The CST interpreters each REF inertial signal component as a sustained cue with a duration of the entire prediction horizon, i.e., 2 s for the CST. As these cues are hard to represent with a hexapod motion system, except for sustained f_x and f_y cues that can be represented by tilt coordination, the CST does not make the effort to cue f_z. Moreover, in f_z, likewise the CWO, ORC, and PFT, the offset from the REF is present. This results in an RMSE(f_z) of 1.41 m/s^2, which is in the same order of magnitude as the other algorithms. In ω_y at $t \approx 10.5$ s, similar to the CWO, the CST initially goes the wrong way to cue f_x by tilt-coordination, resulting in an RMSE(ω_y) of 2.00 deg/s and a PCC(ω_y) of 0.38.

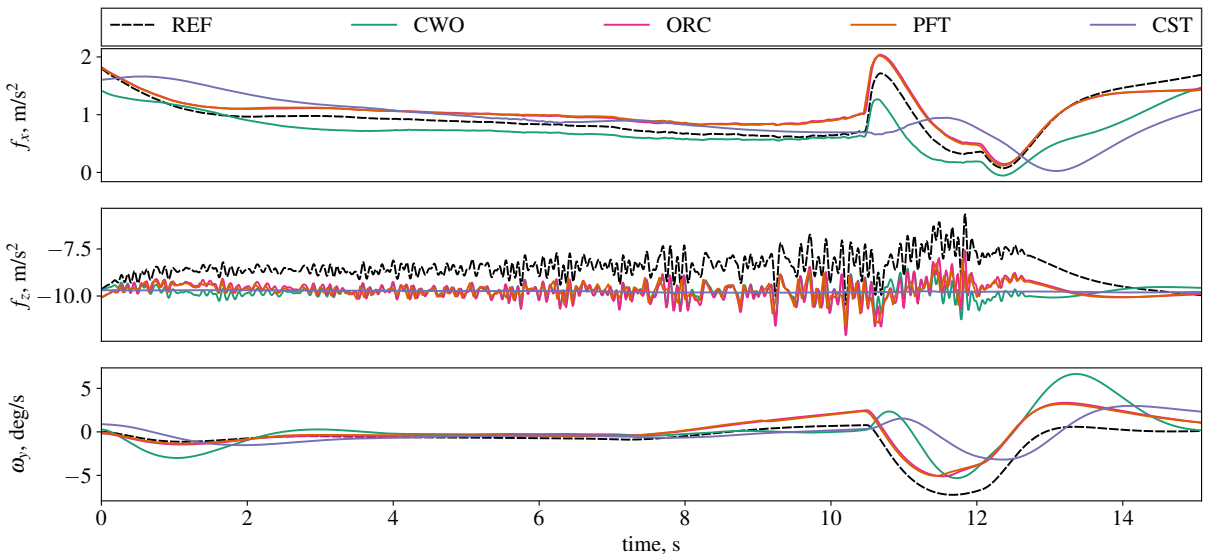
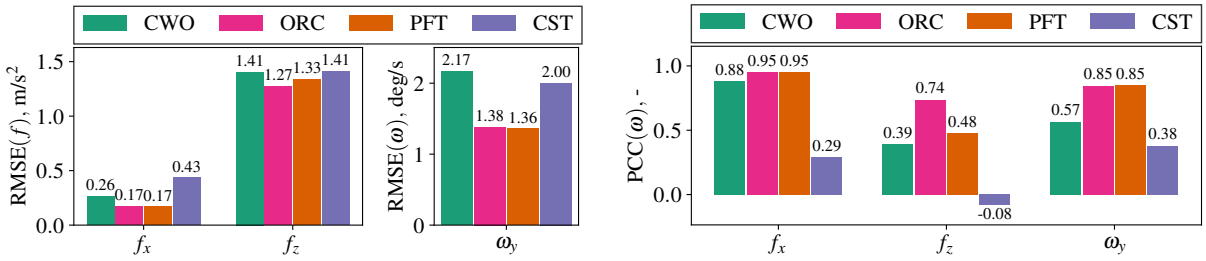


Fig. 7 Reference inertial signal and the inertial signals of the four algorithms for the Symmetric Stall scenario.



(a) RMSE. **(b) PCC.**

Fig. 8 Inertial signal RMSE and PCC for the Symmetric Stall scenario.

2. Large Roll inertial signal

Figure 9 shows the three most relevant DOF of the different algorithms' inertial signals, $[f_y, f_z, \omega_x]$, for the Large Roll scenario. In Figure 10, the Large Roll scenario's inertial signal RMSE (Figure 10a) and PCC (Figure 10b) values for all algorithms are shown. First, note the CWO algorithm, in f_y, this algorithm induces large lateral false cues due to the rolling motion in ω_x, resulting in an RMSE(f_y) of 1.00 m/s^2 and a PCC(f_y) of 0.06. In f_z, the CWO is not able to replicate the amplitudes of the REF, resulting in an RMSE(f_z) of 1.33 m/s^2. Moreover, the PCC(f_z) of -0.32 indicates a negative correlation between the CWO and REF, which may be due to the third-order high-pass filter. In ω_x, the second-order high-pass filter generates significant lead. This results in opposite-sign ω_x motion compared to the REF, which is reflected by an RMSE(ω_x) of 12.73 deg/s and a PCC(ω_x) of 0.19.

Like the Symmetric Stall scenario, the ORC and PFT are very similar. In f_y at $t \approx [18\text{ s}, 23\text{ s}]$, just before the large rolling motion, a false lateral cue is present that is induced by subtle rolling motion in ω_x. Note that this motion is so-called "pre-positioning", which is a key feature of MPC-based MCAs. Pre-positioning is explained as (opposite-sign) motion that positions the simulator in a way that it is better able to cue upcoming (large-amplitude) motion. Next to that, large lateral false cues are present to the rolling motion after the pre-positioning, resulting in an RMSE(f_y) of 0.71 m/s^2 and 0.69 m/s^2 and a PCC(f_y) of 0.28 and 0.30 for, respectively, the ORC and PFT. In f_z, the ORC and PFT are not able to follow the magnitude of the REF amplitudes, leading to an RMSE(f_z) of, respectively, 1.20 m/s^2 and 1.19 m/s^2. However, compared to the CWO, the ORC's and PFT's f_z shapes cohere significantly better, indicated by, respectively, a PCC(f_z) of 0.46 and 0.43, versus the CWO's PCC(f_z) of -0.32. In ω_x, excluding the pre-positioning mentioned earlier, the ORC and PFT almost perfectly replicate the shape of the REF motion, just with smaller amplitudes. This is well reflected by the high PCC(ω_x) values of 0.95 for the ORC and 0.94 for the PFT. The smaller amplitudes still result in a substantial RMSE(ω_x) of 6.82 deg/s and 6.84 deg/s for, respectively, the ORC and PFT.

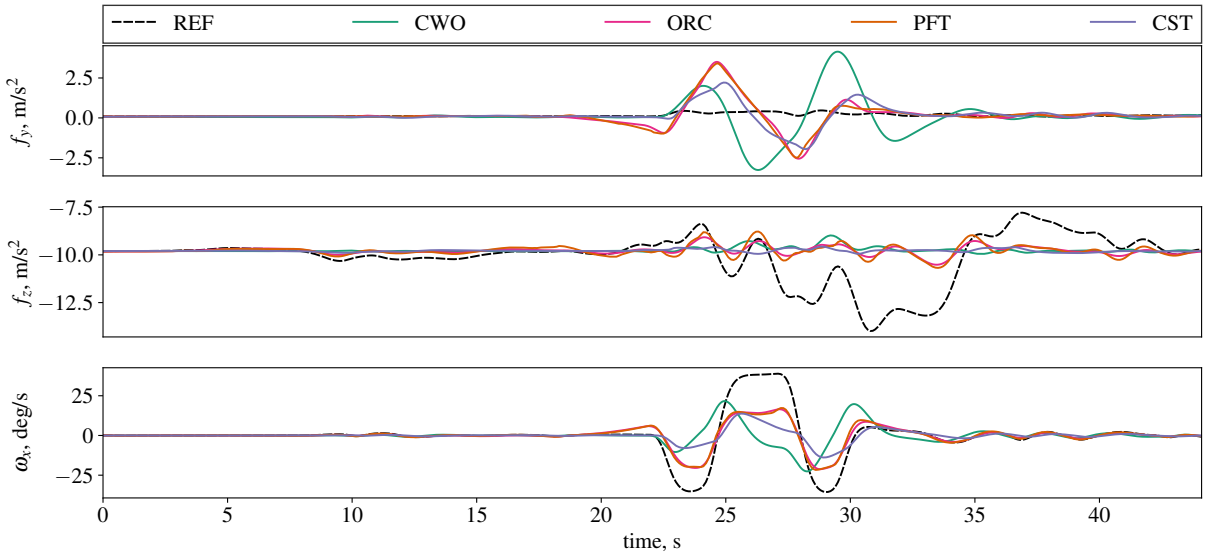
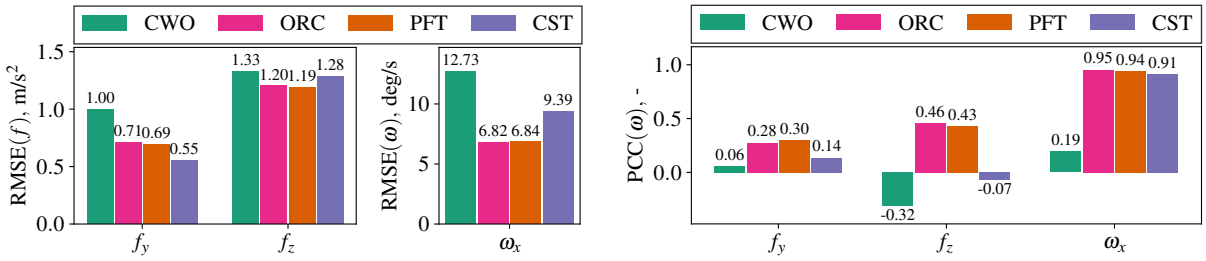


Fig. 9 Reference inertial signal and the inertial signals of the four algorithms for the Large Roll scenario.

Lastly, the CST does not show the pre-positioning motion in f_y as present by the ORC and PFT algorithms. This is explained by the fact that the CST algorithm simply does not "see" the future motion coming, as this is not present in the constant prediction. The false lateral cues are again present, however, these have a smaller amplitude compared to the ORC and PFT, resulting in a lower, thus better, RMSE(f_y) of 0.55 m/s^2. Moreover, the lateral false cues lag behind the ORC and PFT, which, apparently, results in a lower PCC(f_y) of 0.14. In f_z, the CST shows the least motion of all algorithms, likely for similar reasons explained previously for the Symmetric Stall scenario. This results in RMSE(f_z) and PCC(f_z) values of, respectively, 1.28 m/s^2 and -0.07, which are just between the CWO, and the ORC and PFT. Hence, little motion is present, however, also little false cues. In ω_x, the CST is not able to reproduce the same amplitude magnitudes as the ORC and PFT, resulting in a higher RMSE(ω_x) of 9.39 deg/s. In addition, a minor lag is visible, which was also reflected by f_y. This leads to a PCC(ω_x) of 0.91, which is lower than the ORC and PFT, however, better than the CWO.



(a) RMSE. **(b) PCC.**

Fig. 10 Inertial signal RMSE and PCC for the Large Roll scenario.

3. Upsets inertial signal

The Upsets scenario's inertial signal, $[f_x, f_y, f_z, \omega_x, \omega_y, \omega_z]$, for all four algorithms is shown in Figure 11. The Upsets scenario's inertial signal RMSEs are shown in Figure 12 and PCCs are shown in Figure 13. First, for the CWO algorithm in f_x, a large difference between the CWO and REF is present. This is caused by the CW filter gain and results in an RMSE(f_x) of 0.57 m/s^2, however, this does not affect the high PCC(f_x) of 0.98. In f_y, similar to the Large Roll scenario, large lateral false cues are present due to large rolling motion in ω_x, this results in an RMSE(f_y) of 0.65 m/s^2 and a PCC(f_y) of 0.26. Again, similar to the previous scenarios, large amplitudes of the REF f_z are not well represented, which results in an RMSE(f_z) of 2.27 m/s^2. The third-order high-pass filter also reduces the shape correlation, as a result of which the PCC(f_z) is 0.06. The first-order high-pass filter in ω_x ensures limited phase-shift and results in a PCC(ω_x) of 0.68. Compared to the REF, the amplitude magnitudes are reduced by the filter gain of 0.45, which results in an RMSE(ω_x) of 5.47 deg/s. In ω_y and ω_z, the second-order high-pass filters result in phase distortions and, thus, a PCC(ω_y) of 0.29 and a PCC(ω_z) of -0.08. Moreover, the filter gains, 0.45 and 0.3 for, respectively, ω_y and ω_z, affect the amplitudes and result in an RMSE(ω_y) of 1.47 deg/s and an RMSE(ω_z) of 1.52 deg/s. Lastly, note that the yaw motion, ω_z, during the coordinated turn at $t \approx [70\text{ s}, 120\text{ s}]$ is not present at all for the CWO.

Similar to the Symmetric Stall and the Large Roll scenarios, the ORC and PFT are again very comparable. However, the ω_z motion shows for the first time significant differences between the ORC and PFT algorithms. This is reflected by the RMSE(ω_z) of 0.85 deg/s and 1.22 deg/s and PCC(ω_z) of 0.84 and 0.65 for, respectively, the ORC and PFT. This is most likely due to the very slow dynamics present in the coordinated turn at $t \approx [70\text{ s}, 120\text{ s}]$ that is not well anticipated by the PFT due to the limited prediction horizon length of 6 s. In f_x at $t \approx 50$ s, large false cues are present for the ORC and PFT algorithms due to the ω_y motion, which, compared to the CWO, reduces the PCC(f_x) to 0.88 and 0.87 for, respectively, the ORC and PFT. Note that the RMSE(f_x) of 0.22 m/s^2 for both algorithms is better than the CWO since no filter gain is present. Looking at ω_x, the ORC and PFT algorithms first roll in the opposite direction just before all three upsets (i.e., at $t \approx 45\text{ s}, t \approx 240$ s, and $t \approx 310$ s), hence, this is pre-positioning as also observed for the Large Roll scenario. This results in false lateral f_y cues, which is represented by RMSE(f_y) of 0.46 m/s^2 and 0.45 m/s^2 for, respectively, the ORC and PFT, and a PCC(f_y) of 0.33 for both algorithms. In f_z, the ORC and PFT show more and better motion than the CWO, which is mainly confirmed by the PCC(f_z) of 0.45 and 0.33 for, respectively, the ORC and PFT, compared to the CWO's PCC(f_z) of 0.06. The ORC's and PFT's RMSE(f_z) values of, respectively, 2.22 m/s^2 and 2.21 m/s^2, reflect the magnitude deficiencies in replicating the REF's f_z motion.

Finally, the CST algorithm, in f_x, follows the REF nicely, resulting in the lowest RMSE(f_x) of all algorithms of 0.21 m/s^2. The shape correlation, PCC(f_x), between the CST and REF of 0.89 is better than the ORC and PFT (0.88 and 0.87), however, worse than the CWO (0.98). In f_y, the CST shows large lateral false cues due to motion in ω_x, which results in an RMSE(f_y) of 0.49 m/s^2 and a PCC(f_y) of 0.32. Similar to all other algorithms, the CST is not able to replicate the large amplitudes of the REF in f_z, which results in an RMSE(f_z) of 2.26 m/s^2. The CST has a PCC(f_z) of 0.27 and performs in between the CWO (0.06) and the PFT (0.33) and ORC (0.45). In ω_x, the CST does not show any pre-positioning motion and is more comparable to the CWO than the ORC and PFT. The RMSE(ω_x) of 5.46 deg/s and PCC(ω_x) of 0.77 confirm this. In ω_y, similar to all other algorithms, the CST can not replicate the REF's (sustained) amplitudes. This leads to an RMSE(ω_y) of 1.37 deg/s and PCC(ω_y) of 0.46. In ω_z, the CST is able to replicate the amplitudes of the REF, except for the more sustained motion around $t \approx 50$ s and during the coordinated turn at $t \approx [70\text{ s}, 120\text{ s}]$. This results in an RMSE(ω_z) of 1.21 deg/s and PCC(ω_z) of 0.60.

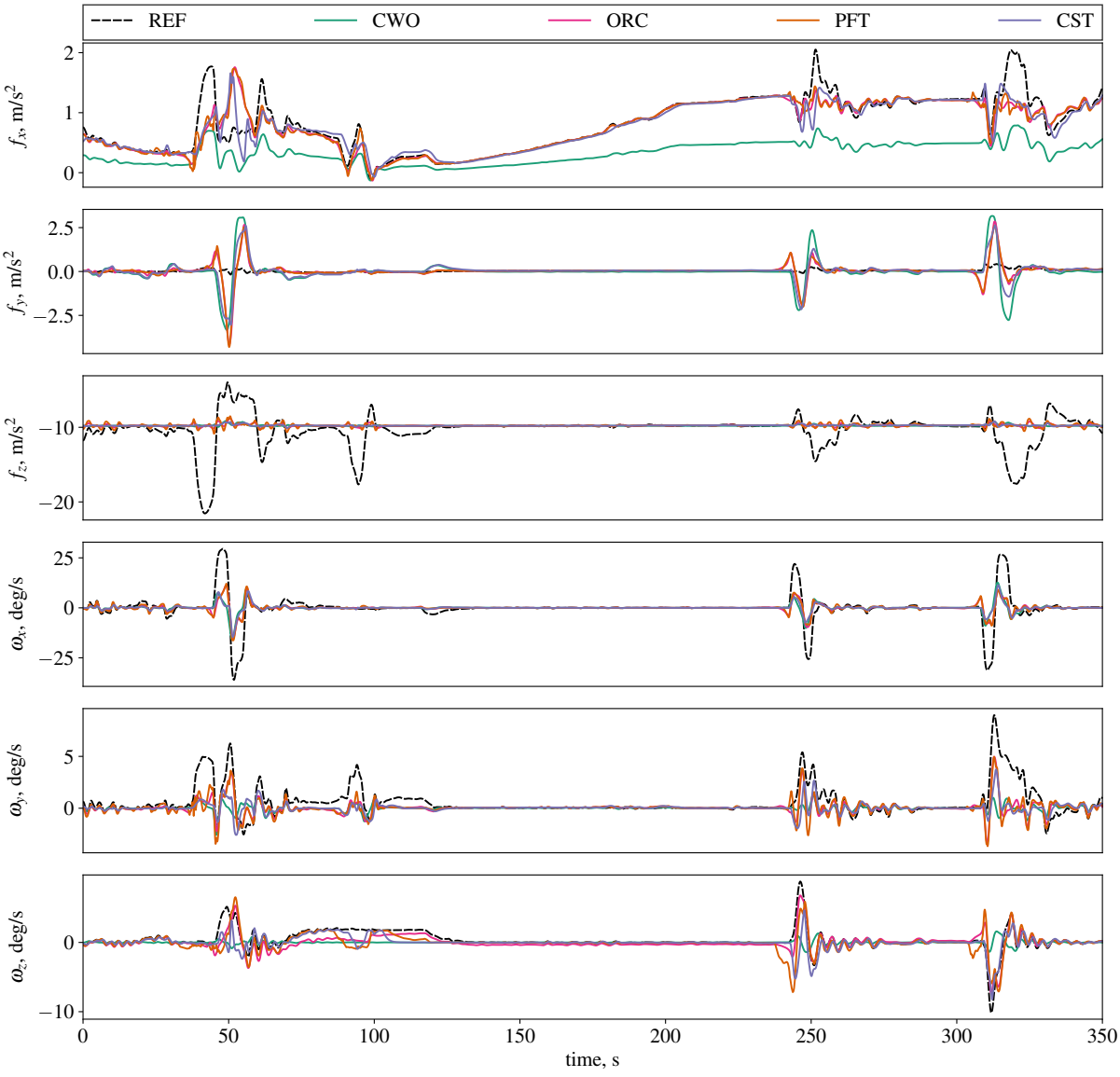


Fig. 11 Reference inertial signal and the inertial signals of the four algorithms for the Upsets scenario.

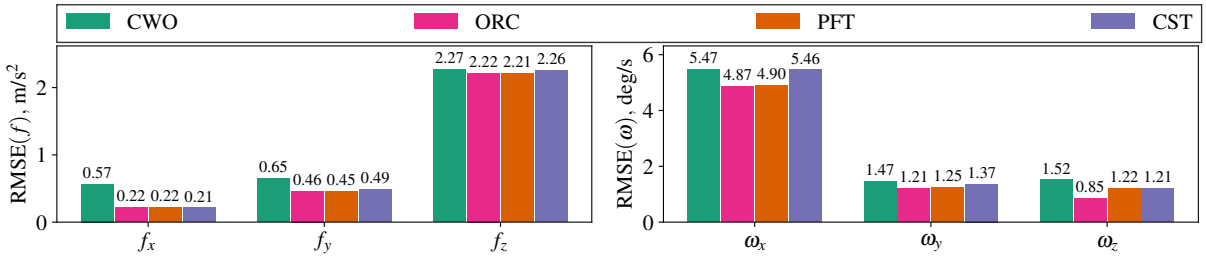


Fig. 12 Inertial signal RMSE for the Upsets scenario.

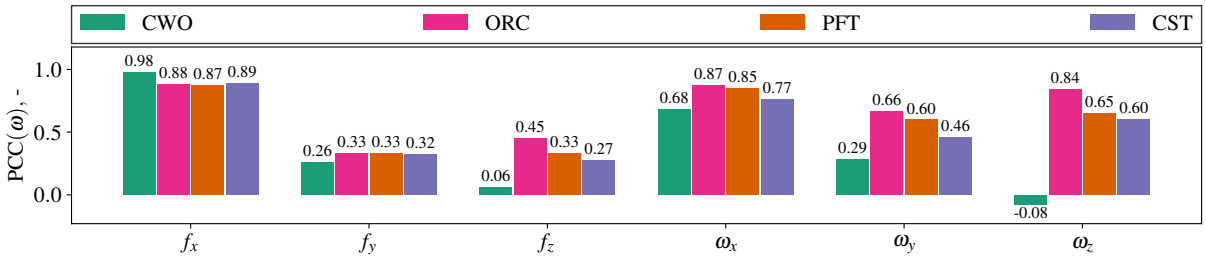


Fig. 13 Inertial signal PCC for the Upsets scenario.

E. Pre-experiment analysis main findings

Summarizing the pre-experiment analysis main findings, the ORC, as expected, results in the highest objective fidelity for seventeen of the total 24 analyzed metrics used in the analysis of the three scenarios. Note that these 24 metrics consist of three RMSEs and PCCs for both the Symmetric Stall and Large Roll scenario, i.e., in total twelve, and six RMSEs and PCCs for the Upsets scenario. To compare the algorithms, the percentage-wise differences between the algorithms are calculated per metric and then averaged. Compared to the CWO, the ORC results in an average increase in objective motion fidelity of 37.9 %, 181.7 %, 185.6 %, and 147.7 % for, respectively, the Symmetric Stall scenario, the Large Roll scenario, the Upsets scenario, and all scenarios combined. Separating the ORC's RMSEs and PCCs, respectively, an on average 27.7 % and 267.8 % increase in fidelity is found for all scenarios compared to the CWO.

The PFT, compared to the ORC, resulted in 2.5 % on average better values for six metrics: the Symmetric Stall's RMSE(ω_y); the Large Roll's RMSE(f_y), RMSE(f_z), and PCC(f_y); and the Upsets' RMSE(f_y) and RMSE(f_z). For the remaining metrics, the PFT resulted, compared to the ORC, in on average 8.7 % worse values. This suggests, at least for these scenarios, that increasing the horizon length over 6 s will not lead to a significant increase in motion quality.

The CWO resulted in an Upsets scenario's PCC(f_x) of 0.98, which is 11.4 % better than the ORC's value of 0.88 and 12.6 % better than the PFT's value of 0.87. For the remaining 23 analyzed metrics, the CWO has lower objective fidelity than the ORC (on average 56.0 %) and PFT (on average 52.2 %). Moreover, the CWO has the lowest fidelity of all algorithms for the Large Roll (six out of six metrics) and the Upsets scenarios (eleven out of twelve metrics).

The CST performs on average 38.2 % worse than the ORC and on average 34.3 % worse than the PFT, based on all 24 analyzed metrics. Moreover, the CST has the lowest fidelity of all algorithms for the Symmetric Stall scenario (four out of six metrics). Hence, the prediction quality has a large influence on the objective fidelity. For the Large Roll scenario, the lowest fidelity for the CWO is caused by the second-order filter-lead that results in phase shift and, compared to the REF, opposite-sign ω_x motion.

The only real-time feasible algorithm, the CST, which allows inter-active simulation, will not be desired for the human-in-the-loop experiment due to the essential buffet motion characteristics that are completely missing in f_z. Moreover, the CST, contrary to the ORC and PFT, does not include any pre-positioning motion, which is possibly one of the key benefits of optimization-based algorithms. Therefore, it is logical to use either the ORC or PFT algorithm in the experiment. Since these algorithms are both not real-time feasible, this will not allow for inter-active simulation in the experiment. For controlling the SRS, a sample time of 0.01 s is required. At this sample rate, it is more efficient to use the ORC instead of the PFT based on computational time. Moreover, due to the high similarities between the ORC and PFT found during the analysis, the outcome of the experiment can, most likely, be generalized between both algorithms. In conclusion, the ORC algorithm is preferred for the SRS experiment.

IV. Experiment

A. Apparatus

The experiment was conducted in the SRS, shown in Figure 14a, at the TU Delft [40]. The participants were not required to give control inputs during the experiment, meaning that no aircraft model and control instruments were required. The visual system was used to provide the outside visuals, driven by either flight or simulation data. The coordinates of the flight data-based visuals were changed to ensure high-intensity visual cues above ground, moreover, the visibility range was increased. Two displays were included, an engine display that represented the engine status and a Primary Flight Display (PFD). This PFD, found in Figure 14c, was extended to additionally present the angle of attack and load factor. Moreover, a flight freeze indicator (bottom left) and a rating bar (top) were added. The purpose of the flight freeze indicator was to inform the participants whether the experiment run was in progress. This was necessary because before and after every experiment run, the SRS was positioning itself while the outside visuals and displays were frozen. The rating bar, positioned on the upper part of the PFD, was controllable by the participants with a hand-held rotary knob, as shown in Figure 14b, and is further elaborated in Subsection IV.F.

The hydraulic motion system of the SRS was driven by offline generated control input data, linear accelerations in inertial axes, and angular accelerations in simulator body axes. To prevent simulator drift due to numerical integration errors, the offline-generated simulator state was used in a control feedback loop. To mask the moving sound of the motion system, general aircraft noises were played on a headset with passive as well as active noise cancellation.

(a) The SIMONA Research Simulator (SRS). **(b) Inside view of the SRS during the experiment.** **(c) PFD presented to the participants during the experiment.**

Fig. 14 Main apparatus of the experiment conducted in the SRS.

B. Participants

A total of ten active pilots (all male, mean age 50.4 years) participated. Five pilots had a Cessna Citation II type rating and seven pilots were commercial airline pilots (either a B737, B777, B787, A320, or A330 type rating), of which three with technical and/or flight testing experience. Seven pilots had in-aircraft experience with upsets and or stalls in either a Cessna Citation II or a medium to large passenger aircraft. The mean flight experience in years was 28.1 and the mean flight hours were about 9,000 hours. The last flight was less than two months prior to the experiment for all participants. For the first two participants, both without in-aircraft UPRT experience, only usable data for the Symmetric Stall scenario were obtained due to technical difficulties and slight adjustments made to the Large Roll and Upsets scenarios after the second participant.

C. Scenarios

Four different continuous-flight scenarios were used in the experiment, of which one scenario was used for training the participants in giving ratings. The remaining three scenarios, the Symmetric Stall, Large Roll, and Upsets, which were already analyzed in Section III, were used for the experiment measurement runs. The three measurement scenarios were designed to have distinctive motion cueing challenges. The Large Roll and the Upsets scenario were directly based on flight data (GPS, AHRS, ADC) obtained with the PH-LAB Cessna Citation II laboratory aircraft. The Symmetric Stall scenario was obtained by a Cessna Citation II autopilot simulation and was used by earlier studies [35, 36]. For every scenario, splines similar to a cosine-like windowing function with a duration of 11 s were applied to both the

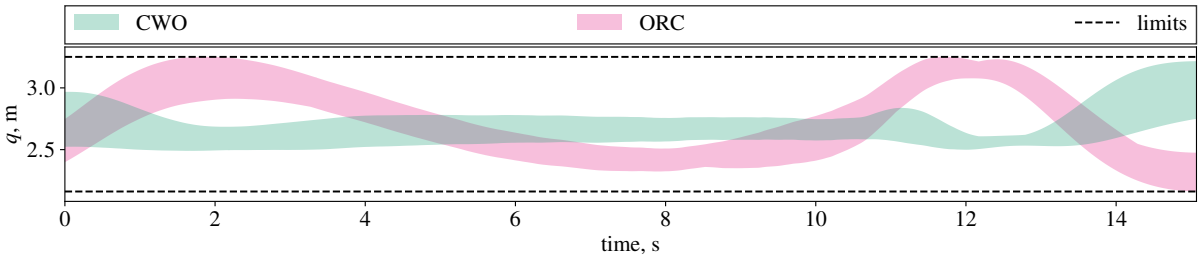
beginnings and endings. This ensured smooth transitions between measurement runs and slow pre- and post-scenario simulator positioning. During these "flight freeze" phases of the scenarios, the visuals and displays were frozen and the SRS motion system slowly positioned itself, while the flight freeze indicator was visible.

D. Independent variables

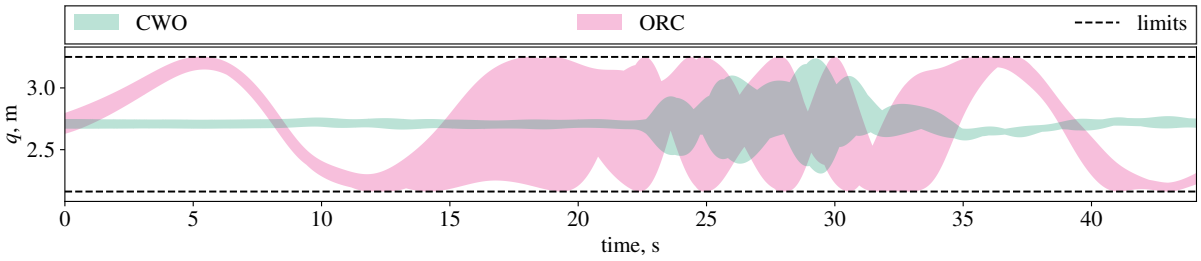
In addition to the different motion scenarios used during the experiment, the experiment tested one within-participant variable, the MCA. Two different MCAs were compared in the experiment: (1) the baseline classical washout algorithm (CWO) and (2) the model predictive control-based "Oracle" algorithm (ORC), which were, respectively, introduced in Subsection III.B and Subsection III.C and compared and analyzed in the remainder of Section III.

E. Conditions

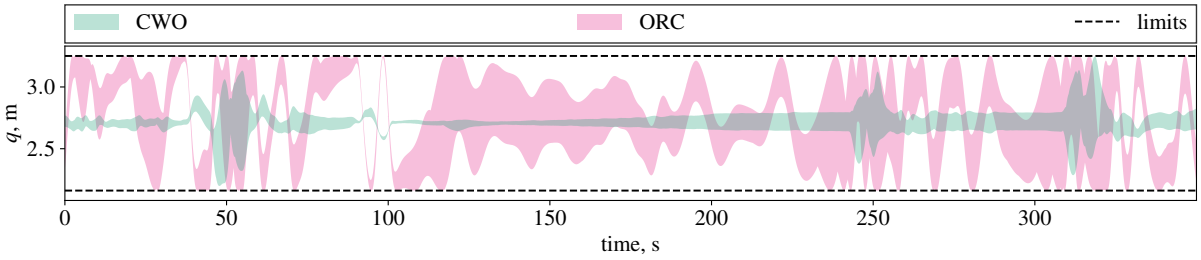
In total, six different measurement conditions are used in the experiment based on two MCAs, the CWO and ORC, and three scenarios. In Figure 15, the used SRS's actuator space, i.e., the space between the minimum and maximum actuator length, for the three scenarios and the CWO and ORC algorithms are shown. Note that the CWO algorithm has a single workspace-limiting motion for all three scenarios, namely, at $t = T_S$, $t = 29$ s, and $t = 320$ s for, respectively, the Symmetric Stall, Large Roll, and Upsets scenarios. Contrary to the CWO algorithm, the ORC algorithm uses more of the available workspace and reaches the actuator limits multiple times for each scenario.



(a) The Symmetric Stall scenario.



(b) The Large Roll scenario.



(c) The Upsets scenario.

Fig. 15 Used simulator actuator space for the CWO and ORC conditions for all three scenarios. SRS's actuator limits as implemented in the software are depicted by the dashed line. The figure's vertical axis range represents the SRS's physical actuator range.

F. Dependent measures

Two measures were collected during the experiment to compare the perceived motion between the CWO and ORC algorithms: (1) the continuous subjective motion incongruence rating (MIR) and (2) the motion fidelity rating (MFR).

The MIR was provided in real-time during the experiment runs by the experiment participants to indicate the difference between the visuals, including PFD, and the perceived motion cues as produced by the motion system. This rating method was introduced in [41] and later used in the experiments of [16, 19, 42, 43]. The MIR was provided by the participants using the hand-held rotary rating knob and visually represented with a rating bar on the PFD, as earlier depicted in Figure 14. The MIR rating ranged from 0 to 10 with 10 discrete integer steps. The color of the rating bar changed to green-yellow-orange-red, with an increasing MIR.

The MFR was a single rating provided after each experiment run. The single rating was given with the help of the MFR decision tree [44]. The focus was put on the numbering scale of the MFR, moreover, participants were instructed to disregard the performance mentions in the decision tree, as they did not actively control the simulated aircraft. To support and substantiate the MFR, comments from the participants were collected.

G. Hypothesis

Based on the analysis performed in Section III, a single hypothesis was formulated and tested for all three scenarios:

> *The ORC algorithm compared to the CWO algorithm will lead to higher perceived fidelity and results in a lower continuous MIR and post hoc MFR.*

The rationale behind the hypothesis was that the in Subsection III.E found an increase in objective motion fidelity, i.e., RMSE and PCC, for the ORC compared to the CWO, also results in an increase in perceived motion fidelity. The lowest increase in objective motion fidelity, but still on average 37.9 % increase, was found for the Symmetric stall scenario. For the Large Roll and Upsets scenarios, an increase was found of on average 181.7 % and 185.6 %, respectively.

H. Experiment procedures

Every participant was first exposed to a training scenario. This flight data-based training scenario was different from all measurement scenarios and contained two upsets and a coordinated turn. The first upset was a 20 deg nose down including a 60 deg bank maneuver, ending in a 15 deg nose-up and wings-level flight. Load factors of 0.5 g to 2.2 g were included in this upset. After this first upset, a 50 s coordinated turn was flown. To conclude the scenario, a 20 deg nose low including 90 deg roll upset was included with load factors ranging from 0.9 g to 1.5 g. The total scenario duration was 200 s excluding the flight freeze phases. The idea behind this scenario for the use as a training scenario was that motion in all six DOF was present, comprehending all motion (except buffet) that would be experienced in the measurement scenarios.

Three scenario sequences were used and are presented in Table 4. At least two repetitions of the training scenario were executed, for which both the CWO and ORC algorithms were used. If needed, more training repetitions were done. A break was included about halfway through the experiment. For the measurement scenarios, three repetitions for each condition were performed, resulting in six repetitions per scenario. The condition order, seen in Table 5, was randomized and repeated per two participants and was unknown to the participants.

Table 4 Scenario sequences used in the experiment. Balanced for every three participants.

Sequence 1:	Training	Symmetric Stall	Large Roll	Break	Upsets
Sequence 2:	Training	Large Roll	Break	Upsets	Symmetric Stall
Sequence 3:	Training	Upsets	Break	Symmetric Stall	Large Roll

The well-being of the participants during the experiment was tracked by the Misery Scale (MISC) [45], a well-known subjective rating scale for motion sickness. Prior to the experiment, it was explained that the experiment was aborted if a MISC of seven was reported, or two consecutive sixes. In addition, the participants were told that they were able to abort the experiment anytime by themselves. None of the participants became sick: for all participants, a maximum MISC of one and a median MISC of zero were reported.

Table 5 Motion cueing strategy sequences used in the experiment. Balanced for every two participants.

Sequence 1.1	CWO	ORC	CWO	ORC	ORC	CWO
Sequence 1.2	ORC	CWO	ORC	ORC	CWO	CWO
Sequence 1.3	CWO	ORC	ORC	CWO	CWO	ORC
Sequence 2.1	ORC	ORC	CWO	CWO	ORC	CWO
Sequence 2.2	ORC	CWO	CWO	ORC	CWO	ORC
Sequence 2.3	CWO	CWO	ORC	CWO	ORC	ORC

I. Data analysis

To verify the consistency of the continuous MIR data, the approach from the experiments found in [16, 19, 41–43, 46] was used. This approach calculates the consistency with Cronbach's alpha [47] for every participant, across repeated measurements. If the Cronbach's alpha was smaller than 0.7 the MIR was considered inconsistent [48]. For that specific scenario and participant, the MIR data were then excluded from the analysis. In addition, it was checked whether the participants were actively using the rating knob by visual inspection of the MIR, which was plotted in real-time during the experiment runs. The continuous MIR was averaged per participant and between participants which resulted in a mean continuous MIR. This mean continuous MIR was, thereafter, averaged over time resulting in a single mean MIR for each condition. Paired Wilcoxon signed-rank tests were used to test for significance between conditions. For the Upsets scenarios, this approach was repeated for subsets of the scenario, i.e., the individual upset maneuvers.

A similar approach was used to analyze the MFR data, starting with a consistency check by calculating Cronbach's alpha for each scenario. Due to the MFR being ordinal data, the median MFR of the three repetitions per participant was used in the paired Wilcoxon signed-rank test. Moreover, this test was applied per repetition separately. The comments used by the participants to substantiate the MFR were used to indicate preferences. This was done by introducing keywords that comprised certain characteristics of the scenario. For the Symmetric Stall scenario, these keywords were "Buffet", "Pitch/surge", and "Heave"; for the Large Roll scenario, "Onset roll" and "Sustained roll"; and for the Upsets scenario, "Pitch", "Heave", "Onset roll", and "Sustained roll". For each keyword, the comments were used to indicate a preference for either the CWO or ORC condition or no preference between the conditions.

V. Results

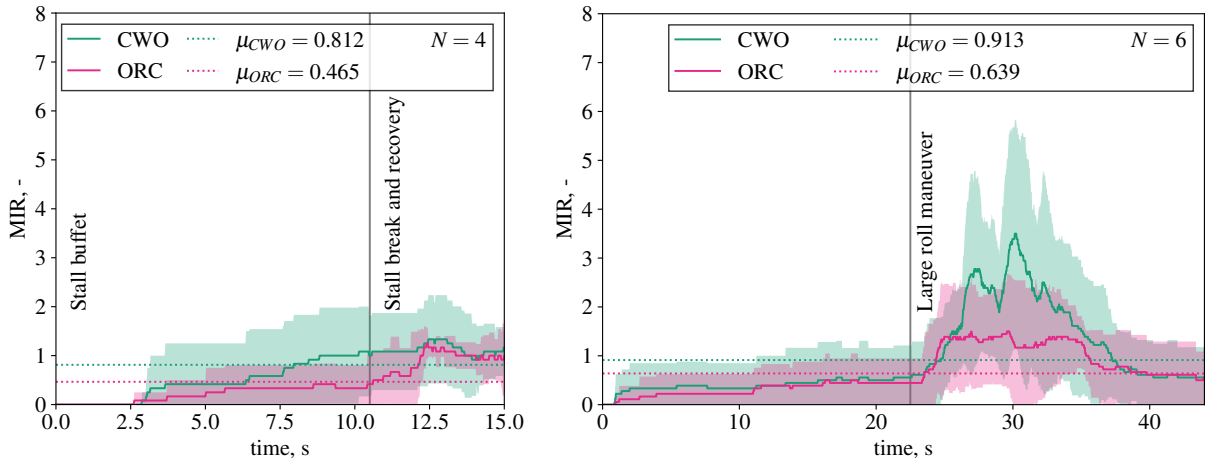
A. Continuous motion incongruence rating

Table 6 shows Cronbach's alpha, α_{Cr}, for every participant and all three scenarios, calculated from the three continuous MIR repetitions of both the CWO and ORC. For the Symmetric Stall scenario, only four participants reached $\alpha_{Cr} > 0.7$. For respectively the Large Roll scenario and the Upsets scenario, six and eight participants scored $\alpha_{Cr} > 0.7$. In line with the lowest number of participants with $\alpha_{Cr} > 0.7$, the Symmetric Stall scenario had the lowest mean α_{Cr} of 0.413, followed by the Large Roll scenario (0.579) and subsequently the Upsets scenario (0.850).

Table 6 Cronbach's alpha for the MIR.

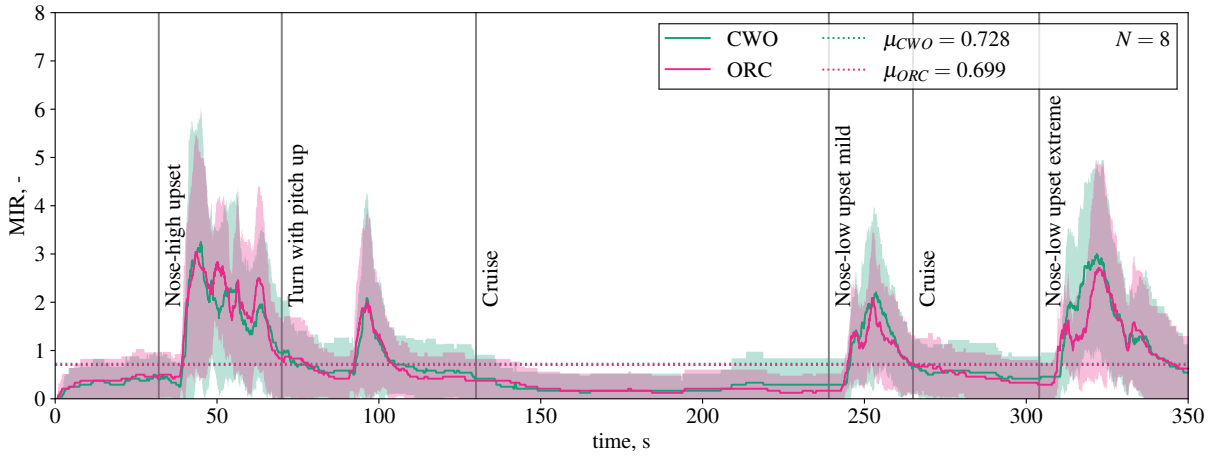
	Cronbach's alpha (α_{Cr})		
Participant	Symmetric Stall	Large Roll	Upsets
P1	0.839	-	-
P2	0.828	-	-
P3	-0.472	-0.504	0.910
P4	0.793	0.893	0.934
P5	0.784	0.864	0.884
P6	-0.095	0.186	0.803
P7	-0.025	0.849	0.751
P8	0.351	0.710	0.799
P9	0.537	0.931	0.804
P10	0.592	0.700	0.912
Mean	0.413	0.579	0.850
Standard deviation	0.438	0.466	0.064
No. participants > 0.7	4	6	8

The average continuous MIR and its standard deviation indicated with the solid lines and the shaded area, and the overall mean MIR per scenario, represented by the horizontal dotted lines, are depicted in Figure 16 for all three scenarios. Note that only the MIR results from participants with a $\alpha_{Cr} > 0.7$ are used in Figure 16. First, in Figure 16a, the average continuous MIR and overall mean MIR for the Symmetric Stall scenario are shown. In line with the pre-experiment analysis presented in Section III that showed better RMSE and PCC results for the ORC compared to the CWO except for RMSE(f_z), the ORC's overall mean MIR ($\mu_{ORC} = 0.465$) is 43 % better compared to the mean MIR of the CWO ($\mu_{CWO} = 0.812$). Moreover, the over-time increasing MIR shows that the stall buffet at $t \approx [0\text{ s}, 10.5\text{ s}]$ was rated better than the stall break and recovery at the interval $t \approx [10.5\text{ s}, T_S]$. Next, Figure 16b shows the average continuous MIR and overall mean MIR for the Large Roll scenario. Again, the MIR results are in line with the pre-experiment analysis, meaning a 30 % better mean MIR for the ORC ($\mu_{ORC} = 0.639$) compared to the CWO ($\mu_{CWO} = 0.913$). This difference mainly arises at the interval $t \approx [25\text{ s}, 35\text{ s}]$, where the aircraft is performing the roll maneuver. The parts before and after the roll maneuver do not show clear differences between the CWO and ORC. Finally, Figure 16c shows the average continuous MIR and overall mean MIR for the Upsets scenario. No clear visible differences are seen between the CWO and ORC, although again, the mean MIR of the ORC is slightly (4 %) better than the CWO's mean MIR. Thus, the large increase in objective fidelity found in Subsection III.E is not well reflected in the MIR.



(a) The Symmetric Stall scenario.

(b) The Large Roll scenario.



(c) The Upsets scenario.

Fig. 16 Mean continuous MIR for the three scenarios. Shaded areas represent the standard deviation of the mean continuous MIR.

The Wilcoxon signed-rank test results for the mean MIRs, i.e., the resulting means of averaging the three repeated measures per participant and averaging over time, of all three scenarios are summarized in Table 7. Only significant results are found for the overall mean MIR for the Large Roll scenario ($N = 6, Z = -1.992, p = 0.046$). Despite the 43 % better mean MIR of the ORC for the Symmetric Stall scenario, the small sample size ($N = 4$) may have led to insignificant results. For the Upsets scenario ($N = 8$), also the individual upset maneuvers (nose-high, nose-low mild, nose-low extreme), indicated in Figure 16c between the horizontal lines, were tested. However, no significant results were found for either the full Upsets scenario or the individual upset maneuvers. These MIR results suggest that the ORC algorithm, compared to the CWO, may lead to at least similar to significantly better-perceived motion quality.

Table 7 Paired Wilcoxon signed-rank test results for the MIR. Significant results indicated in grey.

Scenario	**Maneuver**	N	Z	p-value
Symmetric Stall	Full scenario	4	-1.826	0.068
Large Roll	Full scenario	6	-1.992	0.046
Upsets	Full scenario	8	-0.420	0.674
Upsets	Nose-high upset	8	-0.980	0.327
Upsets	Nose-low upset mild	8	-0.700	0.484
Upsets	Nose-low upset extreme	8	-0.840	0.401

B. Motion Fidelity Rating

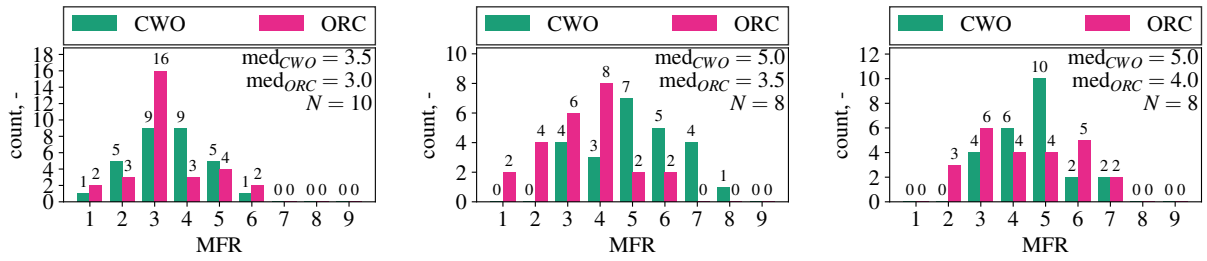
In Table 8, the α_{Cr} for all three scenarios separately as well as combined is shown, calculated based on the three repetitions of the MFR for all participants and both conditions (CWO and ORC). All three scenarios reached an $\alpha_{Cr} > 0.7$. The highest α_{Cr} was obtained for the Large Roll scenario, followed by the Symmetric Stall scenario, and lastly the Upsets scenario. As a result of this, all MFR data are used in the analysis presented next.

Table 8 Cronbach's alpha for the MFR.

Scenario	**Cronbach's alpha (α_{Cr})**	**Lower 95 % confidence**	**Upper 95 % confidence**
Symmetric Stall	0.882	0.751	0.950
Large Roll	0.933	0.844	0.974
Upsets	0.816	0.574	0.930
All	0.892	0.829	0.935

Figure 17 shows the MFR distribution, i.e., three MFRs per condition per participant, for all three scenarios. In Figure 17a, the MFR distribution for a total number of ten participants ($N = 10$) for the Symmetric Stall scenario is shown, yielding a total of 30 MFRs. For both conditions, the distribution is spread around an MFR of three and four with a minimum MFR of one (best rating) and a maximum rating of six. The ORC condition shows a clear peak at an MFR of three with a median value of three while the CWO shows an equal peak at an MFR of three and four with a median value of three and a half. Hence, little difference between the two conditions is present, with a slightly better rating for ORC. In Figure 17b, the MFR distribution ($N = 8$, 24 MFRs) for the Large Roll scenario is presented. The CWO's MFR distribution ranges from three to eight with a median value of five. The ORC condition shows a distribution between an MFR of one and six and a median of three and a half. Thus, a clear difference is seen between the CWO and ORC conditions, where the ORC is rated better. The MFR distribution ($N = 24$, 24 MFRs) for the Upsets scenario is shown in Figure 17c. The CWO distribution ranges between an MFR of three and seven with a median value of five, where also a peak in the distribution is seen. The ORC has a median MFR of four, while the distribution ranges from two to seven. No clear peak is seen for the ORC condition. Similar to the Symmetric Stall scenario, no clear difference in rating is seen between the CWO and ORC, although, the ORC is rated slightly better.

The Wilcoxon signed-rank test results for the median MFR and the three repetitions separately of all three scenarios are presented in Table 9. Only significant results are found for the Large Roll scenario's median MFR ($N = 8, Z = -2.565, p = 0.010$), meaning the median of the three repeated measurements, the Large Roll scenario's repetition one MFR ($N = 8, Z = -2.588, p = 0.010$), the Large Roll scenario's repetition two MFR ($N = 8, Z = -2.401, p = 0.016$), and the Large Roll scenario's repetition three MFR ($N = 8, Z = -2.136, p = 0.033$). These results show, similar to the MIR results, that the ORC algorithm leads to increased perceived motion quality, which is especially the case for the Large Roll scenario.



(a) The Symmetric Stall scenario. **(b) The Large Roll scenario.** **(c) The Upsets scenario.**

Fig. 17 MFR count for all three scenarios. Note that three MFR values are included per participant, due to the three repeated measurements per condition.

Table 9 Paired Wilcoxon signed-rank test results for the MFR. Significant results indicated in grey.

		Median		Repetition 1		Repetition 2		Repetition 3	
Scenario	N	Z	p-value	Z	p-value	Z	p-value	Z	p-value
Symmetric Stall	10	-1.134	0.257	-0.749	0.454	-0.378	0.705	0.000	1.000
Large Roll	8	-2.565	0.010	-2.588	0.010	-2.401	0.016	-2.136	0.033
Upsets	8	-1.414	0.157	-0.378	0.705	-1.155	0.248	-0.707	0.480

C. Participant preferences from comments

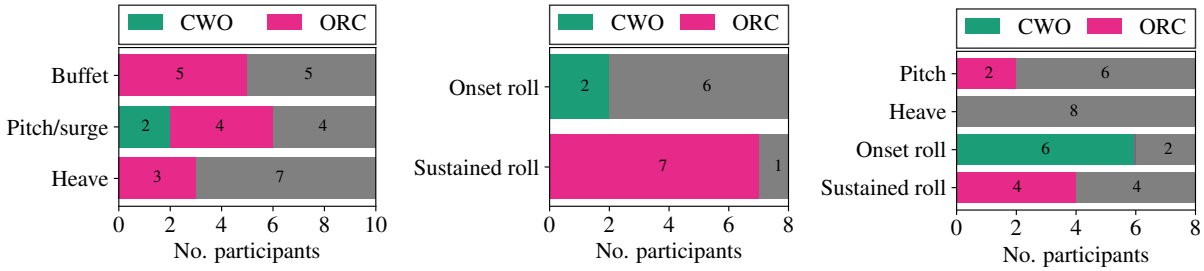
Participants' preferences were identified by analyzing the comments provided next to the MFR. As explained in Subsection IV.I, per scenario, keywords were chosen that covered certain characteristics of the respective scenario. Then, per participant and scenario, the comments were analyzed and checked whether the comment was applicable to one of the chosen keywords and whether a preference for one of the conditions was clear, or not. If no clear, or doubtful, preference was found, a reserved choice of no preference between conditions was noted. The results of this analysis are shown in Figure 18.

The results for the Symmetric Stall scenario are depicted in Figure 18a. Half of the participants, i.e., five participants, indicated a preference for the stall buffet of the ORC condition, for the other five participants, no preference was found. The participants that indicated a preference for the ORC condition mainly praised the "higher intensity" and "heavier" buffet compared to the CWO. The pitch and surge motion preferences were more divided, four participants indicated no preference between conditions. Two participants preferred the CWO over the ORC, mainly because of the "more accentuated push in the back" during the stall break. Four participants indicated that they disliked the "artificial" and "excessive push in the back" of the CWO, and preferred the "more natural" motion of the ORC. The majority of the participants, i.e., seven participants, indicated that they felt "no difference in heave" between conditions or "no variation in g-loading at all". Only three participants indicated feeling "more difference in g" or "more difference in feeling heavier and lighter" during the stall break and recovery for the ORC condition. These results indicate that the majority of participants had no preference between conditions. However, the ORC condition was still preferred over the CWO condition for all scenario characteristics. This suggests that the ORC algorithm, in line with the findings of Subsection V.A and V.B, results in at least similar to more-preferred motion.

The preferences for the Large Roll scenario are shown in Figure 18b. The initial rolling motion, i.e., the onset roll, of the CWO condition was preferred over the ORC by two participants. They indicated that they felt the ORC "initially and subtly rolling in the opposite direction" and indicated that this "revealed that the maneuver was coming". Note that this is due to pre-positioning, which was already identified in Subsection III.D. The other six participants did not indicate a preference between conditions and also did not indicate that they felt this pre-positioning motion of the ORC condition. For the sustained roll, all participants, except one that did not indicate any preference, indicated that they preferred the ORC conditions over the CWO. They commented that they felt "less side force" for the ORC condition and "too much side force" for the CWO condition. Moreover, comments were given that the side force of the CWO was "out of sync" or "did not line up" with the attitude and rolling motion of the aircraft visible on the instruments and outside visuals. Although two participants preferred the CWO's onset motion, the ORC algorithm is preferred over the CWO

for the sustained motion by seven participants. This suggests that in total the ORC was still preferred over the CWO, indicating, similar to the Symmetric Stall scenario, better motion quality.

In Figure 18c, the results for the Upsets scenario are presented. Two participants indicated a preference in pitch for the ORC condition, mainly because of "more definition and variation" in pitch. None of the participants had a pitch preference for the CWO condition. Six participants had neither a pitch preference for the CWO nor ORC condition. None of the participants indicated a preference between conditions for heave motion. Almost all participants commented that they felt "little to no motion in heave" or "negligible differences in g-loading". The onset roll motion of the CWO was preferred over the ORC by six participants. All six indicated that they felt the ORC condition "initially rolling in the wrong direction", hence, similar to the Large Roll scenario, this is due to the pre-positioning discussed earlier in Subsection III.D. Only two participants did not notice or comment on this pre-positioning of the ORC condition and did not indicate a preference. None of the participants preferred the initial rolling motion of the ORC over the CWO. The sustained roll of the ORC was preferred by four participants, they mostly indicated that they felt "less and more realistic side force". The remaining four participants did not indicate a sustained roll preference between conditions. In total, the CWO and ORC algorithms were both preferred six times, however, it can be concluded that the participants unambiguously preferred either the CWO or ORC for certain characteristics of the scenario. This suggests that both the CWO and ORC algorithms have their own distinct features that contribute to their perceived motion quality.



(a) The Symmetric Stall scenario. **(b) The Large Roll scenario.** **(c) The Upsets scenario.**

Fig. 18 Participants' preferences for all three scenarios derived from commentary. The participants that had no preference between conditions are represented by the grey bars.

VI. Discussion

The hypothesis presented in Section IV was that due to the on average 147.7 % higher objective motion fidelity of the ORC compared to the CWO found in Section III, the perceived motion fidelity would increase, leading to lower, thus better, MIR and MFR ratings. The results presented in Section V show indeed better ratings for the ORC: for the Symmetric Stall scenario a 43 % better overall mean MIR and a median MFR of three versus a three and a half for the CWO, for the Large Roll a 30 % better overall mean MIR and a median MFR of three and a half versus a five for the CWO, and for the Upsets scenario a 4 % better overall mean MIR and a median MFR of four versus a five for the CWO. However, the statistical analysis only confirmed a significant difference between the CWO and ORC for the mean overall MIR and the MFR ratings of the Large Roll scenario. Hence, only a significant increase in the perceived fidelity of the ORC with respect to the CWO was found for the Large Roll scenario, implying that the hypothesis can only be accepted for this scenario.

As stated in Section I, the minimization of false specific force cues when large-amplitude angular motion is cued and the minimization of high-frequency washout of a CW MCA were key challenges of upset and stall simulation [8, 9]. Moreover, these factors contribute to the perceived motion cueing fidelity [49]. Note that, compared to the CWO, the ORC indeed decreases the maximum amplitude of the false lateral specific forces due to rolling motion for the Large Roll scenario. However, the ORC also leads to new, but smaller, false cues due to pre-positioning. In total, the false cues are less present for the ORC, which is reflected by the lower RMSE(f_y) of 1.20 m/s^2 compared to the RMSE(f_y) of 1.33 m/s^2 of the CWO. However, a more significant improvement is seen in shape coherence: a PCC(f_y) of 0.46 compared to -0.32 of the CWO and a PCC(ω_x) of 0.95 versus 0.19 of the CWO. Moreover, only two participants noticed the pre-positioning false cues of the ORC and, therefore, preferred the CWO for onset motion. Additionally, the ORC's sustained rolling motion was preferred over the CWO by seven participants. Summarizing, it seems evident that the

phase distortion induced by the second-order filter of the CWO, which was not present for the ORC, contributed to the significant differences in perceived motion fidelity for the Large Roll scenario.

A possible explanation for why no significant results were found for the Symmetric Stall scenario, although the ORC received a 43 % better overall mean MIR, is the small sample size of $N = 4$ that passed Cronbach's alpha consistency test. Contrary to the MIR, the MFR was consistent for this scenario and all ten participants were used in the paired test. However, no significant results were found. This might be explained by the fact that both conditions were rated relatively good, which is reflected by the median MFR of three and three and a half for, respectively, the ORC and CWO. Secondly, another explanation is the fact that the lowest on average increase in objective motion fidelity of 37.9 % was found for this scenario.

Compared to the Large Roll scenario, a larger sample size of $N = 8$ was used for the overall mean MIR paired comparison of the Upsets scenario. In this case, the only 4 % better overall mean MIR of the ORC compared to the CWO must have led to insignificant results. The reason why the ORC resulted in only a 4 % better overall mean MIR can be found in the participants' preferences. Although, four participants indicated to prefer the sustained rolling motion of the ORC, the onset rolling motion of the CWO was preferred by six participants. Hence, the pre-positioning of the ORC was clearly noticeable and disliked by six out of eight participants.

The approach used to verify the MIR consistency with Cronbach's alpha resulted in the best consistency found for the Upsets scenario: all eight participants reached an $\alpha_{Cr} > 0.7$. For the Large Roll and Symmetric Stall scenarios, respectively, six out of eight and four out of ten participants reached an $\alpha_{Cr} > 0.7$. This suggests that it is easier to continuously rate a longer scenario consistently, compared to a short scenario. However, another factor that contributed largely to the consistent MIR of the Upsets scenario is the type of motion included in the scenario. The Upsets scenario contains long cruise flight sections, which ensured that the participants were rating these parts of the scenario very consistently. Hence, for the Upsets scenario, the calculated Cronbach's alpha values do not say a lot about the consistency of the rating during the upset maneuvers since these have less duration compared to the cruise flight sections. In conclusion, the Symmetric Stall scenario is too short to obtain a sufficient MIR. Moreover, Cronbach's alpha is not suitable for MIR consistency analysis if a scenario mostly consists of sections that contain motion that is not of interest to the study, which in this case was cruise flight. Developing another way to verify the consistency of the MIR is, therefore, desired.

Contrary to the MIR, Cronbach's alpha for the MFR of the Upsets scenario was the lowest ($\alpha_{Cr} = 0.816$) of all scenarios. Since the MFR was provided after each run, the long duration of almost six minutes could be the reason for this, as it is difficult, if not impossible, to remember the perceived motion quality for this period of time. The MFR was most consistent for the Large Roll scenario ($\alpha_{Cr} = 0.933$), most likely due to the relatively short duration and clear differences between both algorithms. For the Symmetric Stall scenario, the MFRs were slightly less consistent ($\alpha_{Cr} = 0.882$), although the scenario duration was even shorter. This might be due to the more similar CWO and ORC conditions. This suggests that an MFR is more suitable for scenarios with a limited duration in time, and thus not for the Upsets scenario.

Reflecting back on the scenarios, it can be concluded that all three scenarios contributed to the findings of this paper. The Symmetric Stall scenario was used for identifying possible improvement in heave, and although, three participants indicated a (subtle) preference for the (sustained) heave motion of the ORC, no significant differences were found. This suggests that optimization-based MCAs, when deployed on a motion system similar to the SRS's system, will not likely lead to improvements in (sustained) heave cues. Furthermore, this indicates that, in a future study, scenarios similar to the Symmetric Stall could be excluded. The Large Roll scenario, where the ORC was compared to the CWO with a second-order filter for roll, led to significant improvements. This indicates that the scenario was suitable for identifying differences between MCAs. In a future study, it would be interesting to compare the ORC to a CWO with a first-order filter for roll during a scenario similar to the Large Roll scenario. Contrary to the Large Roll scenario, the Upsets scenario revealed that the (roll) pre-positioning of the ORC was noticeable, the fact that three upsets were included in this scenario, hence, multiple pre-positioning in a short period of time, might have contributed to this. This might suggest that it can be useful to include multiple maneuvers in a scenario. To conclude, the Upsets scenario can be considered too difficult to cue, as it contained very extreme six-DOF motion. This makes it very hard to pinpoint (subtle) differences between MCAs.

Coming back to the MCAs used in the experiment and the analysis prior to the experiment presented in Section III, it seems right that the ORC was used in the experiment. Moreover, the outcome of the experiment can with great certainty be generalized for the PFT algorithm, since the PFT only has an on average 5.9 % decrease in objective motion fidelity compared to the ORC. The CST, with an on average 38.2 % worse objective motion fidelity than the ORC, would most likely not result in any significant results. Therefore, it can be concluded that prediction correctness is a very important

part of an MPC-based MCA. This suggests an important and interesting field of research into the short-term prediction of future aircraft motion or pilot control behavior.

The need for sufficient prediction correctness also substantiates the choice of using an offline MPC-based MCA for this study. Meeting the challenges of implementing such an algorithm in real-time, which allows for interactive simulation, seems, for the motion scenarios used in this paper, only useful when the reference motion can be predicted at a sufficient level. This suggests the usefulness of studies in which the required level of prediction correctness can be identified and studies into how this motion can be predicted.

The choice of not using extensive optimized inertial signal weights in the MPC formulation, but instead "baseline" weights from other literature seems to not have contributed to any complications during the experiment. Optimizing these weights might be more interesting when no perfect prediction correctness is available. The control input derivative weight, $w_{\dot{u}}$, was introduced for and effective for reducing the SRS's actuator velocities. The reason for this was that during the playback of the ORC condition, it was discovered that the SRS's motion system's set-point tracking performance decreases with an increasing actuator velocity and becomes substantial when the actuator's velocity limits are reached. Moreover, due to the higher actuator velocities and the increased intensity of the actuator motion during the ORC condition, a hydraulic oil flow too high for the SRS's motion system was discovered for the first time since the SRS went into operation. A drawback of increasing the $w_{\dot{u}}$ is, that it is at the expense of high-frequency motion that is replicated by the simulator. Thus, other weights, or another way, to reduce the motion intensity and actuator velocity, without compromising motion quality, should be investigated.

As mentioned in Section II, no sensory models were used in this study. However, a possible solution to reduce the motion intensity, i.e., decreasing the amount of motion produced by the simulator, is to implement sensory models that, e.g., filter out sub-threshold motion. The ORC, namely, produced a lot of motion that was not noticeable to the participants, i.e., sub-threshold motion. An example of this is the low yaw rate motion during the coordinated turn of the Upsets scenario. Sensory models might also be used for penalizing, or limiting, above-threshold pre-positioning motion. As the participants indicated that the pre-positioning was noticeable, this is worth investigating.

VII. Conclusion

This paper investigated the potential of optimization-based motion cueing algorithms for simulating upset and stall maneuvers. Three flight scenarios flown with TU Delft's laboratory aircraft were analyzed: a stall, a large roll maneuver, and a scenario with both nose-low and nose-high upsets. The pre-experiment analysis showed an up to 147.7 % average increase in objective motion fidelity for the optimization-based algorithms compared to the baseline classical washout algorithm. The experiment with ten pilots performed in TU Delft's SIMONA Research Simulator yielded higher perceived motion fidelity for the optimization-based algorithm compared to the baseline classical washout for all scenarios. However, only a significant increase in perceived motion fidelity was found for the Large Roll scenario. The pilots mainly praised the decreased lateral false cues and phase shift. Moreover, the pilots did not notice significant differences in (sustained) g-loading between the algorithms. These results indicate that optimization-based algorithms have the potential to achieve significantly better motion quality compared to filter-based algorithms if reference motion is predicted at a sufficient level.

References

[1] International Air Transport Association (IATA), "Loss of Control In-Flight Accident Analysis Report Edition 2019," Tech. rep., IATA, Montreal, 2019.

[2] International Civil Aviation Organization (ICAO), *Doc. 10011 AN/506 Manual of Aeroplane Upset Prevention and Recovery Training*, 1st ed., ICAO, 2014.

[3] International Civil Aviation Organization (ICAO), "Airplane Upset Prevention and Recovery Training Aid (REV 3)," , 2 2017.

[4] European Union Aviation Safety Agency (EASA), "Annex I to ED Decision 2019/005/R, 'AMC and GM to Part-FCL — Issue 1, Amendment 7'," , 2019.

[5] Reid, L. D., and Nahon, M. A., "Flight Simulation Motion-Base Drive algorithms: Part 1 - Developing and Testing the Equations," Tech. rep., UTIAS, 12 1985.

[6] Advani, S. K., and Schroeder, J. A., "Global implementation of upset prevention & recovery training," *AIAA Modeling and Simulation Technologies Conference*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2016. https://doi.org/10.2514/6.2016-1430.

[7] Field, J., Roza, M., and Smaili, H., "Developing upset cueing for conventional flight simulators," *AIAA Modeling and Simulation Technologies Conference*, 2012. https://doi.org/10.2514/6.2012-4948.

[8] Chung, W. W., "A preliminary investigation of achievable motion cueing in ground-based flight simulators for upset recovery maneuvers," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Honolulu, Hawaii, 2008. https://doi.org/10.2514/6.2008-6868.

[9] Zaichik, L., Yashin, Y., and Desyatnik, P., "Motion fidelity criteria for large-amplitude tasks," *AIAA Modeling and Simulation Technologies Conference*, 2009. https://doi.org/10.2514/6.2009-5916.

[10] Ko, S. F., and Grant, P. R., "Development and testing of an adaptive motion drive algorithm for upset recovery training," *AIAA Modeling and Simulation Technologies Conference*, 2012. https://doi.org/10.2514/6.2012-4947.

[11] Zaal, P. M. T., "Motion cueing for stall recovery training in commercial transport simulators," *AIAA Scitech Forum*, 2019. https://doi.org/10.2514/6.2019-0979.

[12] Zaichik, L. E., Yashin, Y. P., Desyatnik, P. A., and Smaili, H., "Some aspects of upset recovering simulation on hexapod simulators," *AIAA Modeling and Simulation Technologies Conference*, 2012. https://doi.org/10.2514/6.2012-4949.

[13] Zaichik, L. E., Yashin, Y. P., Desyatnik, P. A., and Arkhangelsky, Y. A., "Motion cueing fidelity in upset recovery simulation," *AIAA Scitech Forum*, 2019. https://doi.org/10.2514/6.2019-0711.

[14] Dagdelen, M., Reymond, G., Kemeny, A., Bordier, M., and Maïzi, N., "Model-based predictive motion cueing strategy for vehicle driving simulators," *Control Engineering Practice*, Vol. 17, No. 9, 2009, pp. 995–1003. https://doi.org/10.1016/j.conengprac.2009.03.002.

[15] Garrett, N. J., and Best, M. C., "Model predictive driving simulator motion cueing algorithm with actuator-based constraints," *Vehicle System Dynamics*, Vol. 51, No. 8, 2013, pp. 1151–1172. https://doi.org/10.1080/00423114.2013.783219.

[16] Venrooij, J., Cleij, D., Katliar, M., Pretto, P., Bülthoff, H. H., and Schöner, H.-P., "Comparison between filter-and optimization-based motion cueing in the Daimler Driving Simulator," *DSC 2016 Europe VR*, 2016.

[17] Grottoli, M., Cleij, D., Pretto, P., Lemmens, Y., Happee, R., and Bülthoff, H. H., "Objective evaluation of prediction strategies for optimization-based motion cueing," *Simulation*, Vol. 95, No. 8, 2019, pp. 707–724. https://doi.org/10.1177/0037549718815972.

[18] van der Ploeg, J. R., Cleij, D., Pool, D. M., Mulder, M., and B, H. H., "Sensitivity Analysis of an MPC-based Motion Cueing Algorithm for a Curve Driving Scenario," *DSC Europe VR*, 2020.

[19] Cleij, D., Pool, D. M., Mulder, M., and Bülthoff, H. H., "Optimizing an Optimization-Based MCA using Perceived Motion Incongruence Models," *DSC 2020 Europe VR*, 2020.

[20] Lamprecht, A., Steffen, D., Nagel, K., Haecker, J., and Graichen, K., "Online Model Predictive Motion Cueing With Real-Time Driver Prediction," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 8, 2022, pp. 12414–12428. https://doi.org/10.1109/TITS.2021.3114003.

[21] Chadha, A., Jain, V., Lazcano, A. M. R., and Shyrokau, B., "Computationally-efficient Motion Cueing Algorithm via Model Predictive Control," 2023. URL http://arxiv.org/abs/2304.03232.

[22] Katliar, M., Olivari, M., Drop, F. M., Nooij, S., Diehl, M., and Bülthoff, H. H., "Offline motion simulation framework: Optimizing motion simulator trajectories and parameters," *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 66, 2019, pp. 29–46. https://doi.org/10.1016/j.trf.2019.07.019.

[23] Katliar, M., "Optimal control of motion simulators," Ph.D. thesis, Albert-Ludwigs-Universität Freiburg, Freiburg, 2021.

[24] Rawlings, J. B., Mayne, D. Q., and Diehl, M. M., *Model Predictive Control: Theory, Computation, and Design 2nd Edition*, Nob Hill Publishing, Madison, Wisconsin, 2017.

[25] Andersson, J. A., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M., "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, Vol. 11, No. 1, 2019, pp. 1–36. https://doi.org/10.1007/s12532-018-0139-4.

[26] Wächter, A., and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57. https://doi.org/10.1007/s10107-004-0559-y.

[27] "HSL. A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk/," , 2002.

[28] Ormsby, C., *Model of human dynamic orientation*, Massachusetts Institute of Technology, 1974.

[29] Stroosma, O., "The SIMONA Research Simulator's Motion Software," , 6 2022.

[30] Advani, S., "The Kinematic Design of Flight Simulator Motion-Bases," Ph.D. thesis, Delft University of Technology, Delft, 1998.

[31] Berkouwer, W. R., Stroosma, O., Van Paassen, M. M., Mulder, M., and Mulder, J. A., "Measuring the performance of the SIMONA Research Simulator's motion system," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Vol. 2, 2005. https://doi.org/10.2514/6.2005-6504.

[32] Katliar, M., Fischer, J., Frison, G., Diehl, M., Teufel, H., and Bülthoff, H. H., "Nonlinear Model Predictive Control of a Cable-Robot-Based Motion Simulator," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 9833–9839. https://doi.org/10.1016/j.ifacol.2017.08.901.

[33] Katliar, M., Drop, F. M., Teufel, H., Diehl, M., and Bülthoff, H. H., "Real-Time Nonlinear Model Predictive Control of a Motion Simulator Based on a 8-DOF Serial Robot," *European Control Conference*, 2018. https://doi.org/10.0/Linux-x86{_}64.

[34] Khusro, Y. R., Zheng, Y., Grottoli, M., and Shyrokau, B., "MPC-Based Motion-Cueing Algorithm for a 6-DOF Driving Simulator with Actuator Constraints," *Vehicles*, Vol. 2, No. 4, 2020, pp. 625–647. https://doi.org/10.3390/vehicles2040036.

[35] Smets, S. C., de Visser, C. C., and Pool, D. M., "Subjective noticeability of variations in quasi-steady aerodynamic stall dynamics," *AIAA Scitech Forum*, 2019. https://doi.org/10.2514/6.2019-1485.

[36] Imbrechts, A., de Visser, C. C., and Pool, D. M., "Just Noticeable Differences for Variations in Quasi-Steady Stall Buffet Model Parameters," *AIAA Scitech Forum*, 2022. https://doi.org/10.2514/6.2022-0510.

[37] Grant, P. R., Moszczynski, G. J., and Schroeder, J. A., "Post-stall flight model fidelity effects on full stall recovery training," *Modeling and Simulation Technologies Conference*, 2018. https://doi.org/10.2514/6.2018-2937.

[38] Sinacori, J., "The determination of some requirements for a helicopter flight research simulation facility," Tech. rep., NASA, 9 1977.

[39] Schroeder, J. A., "Helicopter Flight Simulation Motion Platform Requirements," Tech. rep., NASA, Moffett Field, 1999.

[40] Stroosma, O., van Paassen, M. M. R., and Mulder, M., "Using the SIMONA Research Simulator for Human-machine Interaction Research," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2003. https://doi.org/doi:10.2514/6.2003-5525, URL https://doi.org/10.2514/6.2003-5525.

[41] Cleij, D., Venrooij, J., Pretto, P., Pool, D. M., Mulder, M., and Bülthoff, H. H., "Continuous subjective rating of perceived motion incongruence during driving simulation," *IEEE Transactions on Human-Machine Systems*, Vol. 48, No. 1, 2018, pp. 17–29. https://doi.org/10.1109/THMS.2017.2717884.

[42] Cleij, D., Venrooij, J., Pretto, P., Katliar, M., Bülthoff, H. H., Steffen, D., Hoffmeyer, F. W., and Schöner, H. P., "Comparison between filter- and optimization-based motion cueing algorithms for driving simulation," *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 61, 2019, pp. 53–68. https://doi.org/10.1016/j.trf.2017.04.005.

[43] van Leeuwen, T. D., Cleij, D., Pool, D. M., Mulder, M., and Bülthoff, H. H., "Time-varying perceived motion mismatch due to motion scaling in curve driving simulation," *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 61, 2019, pp. 84–92. https://doi.org/10.1016/j.trf.2018.05.022.

[44] Hodge, S. J., Perfect, P., Padfield, G. D., and White, M. D., "Optimising the yaw motion cues available from a short stroke hexapod motion platform," *Aeronautical Journal*, Vol. 119, No. 1211, 2015, pp. 1–21. https://doi.org/10.1017/S0001924000010228.

[45] Bos, J. E., MacKinnon, S. N., and Patterson, A. J., "Motion Sickness Symptoms in a Ship Motion Simulator: Effects of Inside, Outside, and No View," *Aviation Space and Environmental Medicine*, Vol. 76, 2006, pp. 1111–8.

[46] Lambooij, M., IJsselsteijn, W. A., and Heynderickx, I., "Visual discomfort of 3D TV: Assessment methods and modeling," *Displays*, Vol. 32, No. 4, 2011, pp. 209–218. https://doi.org/10.1016/j.displa.2011.05.012.

[47] Cronbach, L. J., "Coefficient alpha and the internal structure of tests," *Psychometrika*, Vol. 16, No. 3, 1951, pp. 297–334.

[48] Hair, J. F., *Multivariate Data Analysis: A Global Perspective*, 7th ed., Prentice Hall, Saddle River, 2009.

[49] Grant, P. R., and Reid, L. D., "Motion washout filter tuning: Rules and requirements," *Journal of Aircraft*, Vol. 34, No. 2, 1997, pp. 145–151. https://doi.org/10.2514/2.2158.

Part II

Appendices

Appendix A

Python framework code

In this appendix, the configuration file, `config.py`, and an example main run file, `mpcmsf_run.ipynb`, of the Model Predictive Control Motion Simulation Framework (MPCMSF), a self developed variant of the Offline Motion Simulation Framework (OMSF) [12], are included. The MPCMSF was used in this study to calculate all the simulator trajectories used in the analyses, as well as the trajectories used in the SRS experiment.

A.1. `config.py`

```
'''
Created on September 14, 2022

@author: RoemerBakker
'''
import numpy as np
import mpcmsf.simulator

### MPCMSF DIRECTORY FROM RUN DIRECTORY ###
RUN_PATH = ''

### SAVE DIRECTORY ###
SAVE_PATH = '/Users/Roemer/'

### SAVE FOLDER NAME ###
save_name = "Symmetric_Stall_ORACLE"

### DATA DIRECTORY ###
DATA_DIR = '/Users/Roemer/Documents/GitHub/thesis/thesis_data/Exp_scenarios/
 ↪mpcmsf_input'

### FILE NAME ###
FILE_NAME = '/Symmetric_Stall_mpcmsf_input.npy'

### DATA PATH ###
DATA_PATH = DATA_DIR + FILE_NAME

### PLOTTING ###
USE_TEX = False
```

```
### SIMULATOR ###
simulator = mpcmsf.simulator.SIMONA()

### WEIGHTS ###
INPUT_WEIGHT = 0.01
STATE_WEIGHT = 0.01
OUTPUT_WEIGHT = [1, 1, 1, 10, 10, 10, 1, 1, 1]
INPUT_DERIVATIVE_WEIGHT = [0.25,0.25,0.25,0.25,0.25,0.25]
TERMINAL_WEIGHT = [0,0,0,0,0,0,1,1,1,1,1,1]
weights: dict = {
    "input" : INPUT_WEIGHT,
    "input derivative" : INPUT_DERIVATIVE_WEIGHT,
    "state" : STATE_WEIGHT,
    "terminal state" : TERMINAL_WEIGHT,
    "signal" : OUTPUT_WEIGHT
}

### TRANSFORMATION UGP TO DESIRED REFERENCE POINT ###
T_DERP = np.array(
    [[1,0,0,0],
    [0,1,0,0],
    [0,0,1,-1.2075],
    [0,0,0,1]]
)
T_HP = T_DERP

### OPTIMISER SETTINGS ###
jit = False
parallelization = 'serial'
solver = {'linear_solver' : 'ma57'} # better for small problems: MPC
# solver = {'linear_solver' : 'ma97'} # better for large problems: ORACLE
order_collocation = 1

### SCENARIO SETTINGS ###
SAMPLE_TIME = 0.1
initial_prepositioning = False
T_FADE_IN = 11
T_FADE_OUT = 11

### MPC SETTINGS ###
constant_horizon = False
MPC_SAMPLE_TIME = SAMPLE_TIME # these should be equal, only this works␣
 ↪correctly!
MPC_HORIZON_LENGTH = 6
MAKE_SOLVER_SILENT = True

### CW SETTINGS ###
cw16_set = "cw16_set_Symmetric_Stall"
```

A.2. mpcmsf_run.ipynb

1 Introduction

This is an example main run file to demonstrate the functionalities of the Model Predictive Control Motion Simulation Framework (MPCMSF) and explain some of the working principles. The MPCMSF builds upon the Offline Motion Simulation Framework (OMSF), an optimal control framework for motion simulators (https://github.com/mkatliar/omsf), with two main additional functions. The first one is the ability to run MATLAB (or Simulink) models within this Python framework, for example, a Classical Washout algorithm. Secondly, the ability to use the optimal control framework as an offline Model Predictive Control (MPC) algorithm. Note that the OMSF, and thus the MPCMSF, is dependent on CasADi (https://web.casadi.org), casadi_extras (https://github.com/mkatliar/casadi_extras), Ipopt (https://coin-or.github.io/Ipopt/), and optionally the HSL solvers (https://www.hsl.rl.ac.uk/ipopt/).

1.1 Imports

First, we import some basic libraries as well as CasADi and casadi_extras. Then, we import the required stuff from the MPCMSF and, finally, the configuration file.

[1]:
```
import time
import numpy as np
import matplotlib.pyplot as plt
import casadi as cs
import casadi_extras as ct

from mpcmsf import input, output, plotting, transform, optimizer, scenario,␣
 ↪cost_analysis, mpc, util
from mpcmsf.hexapod_motion_platform import HexapodPlatform
from mpcmsf.signals import INERTIAL_SIGNAL, REFERENCE_INERTIAL_SIGNAL
import config as cfg
```

1.2 Initialization

If a LaTeX installation is present on you machine, you can set

`USE_TEX = True`

in `config.py`.

[2]:
```
if cfg.USE_TEX == True:
    plt.rcParams.update({"text.usetex": True})
```

1.2.1 Init simulator motion platform

[3]:
```
simulator = cfg.simulator
platform = HexapodPlatform(simulator=simulator)
```

```
Hexapod motion platform is initialised for SIMONA.
```

1.2.2 Init cost function

Here we initialize the cost function. The running cost (Langrangian `L`) will be passed on to the `scenario`. The terminal state weight matrix (`W_N`) and the control input derivative weight matrix (`W_udot`) will be passed on to the `optimize()` method later on.

[4]:
```
delta_y = INERTIAL_SIGNAL.cat - REFERENCE_INERTIAL_SIGNAL.cat
W = cs.diag(cfg.weights["signal"])
L = cfg.weights["input"] * cs.sumsqr(platform.input.expr) \
    + cfg.weights["state"] * cs.sumsqr(platform.state.expr) \
    + cs.mtimes([cs.transpose(delta_y), cs.mtimes(W.T, W), delta_y])

W_N = cs.diag(cfg.weights["terminal state"])
W_udot = cs.diag(cfg.weights["input derivative"])
```

1.2.3 Init scenario

[5]:
```
scenario = scenario.Scenario(platform=platform, lagrange_term=L)
scenario.headToPlatform = cfg.T_HP
```

1.2.4 Constraint initialization

Here we set the initial condition constraints, which can be either the neutral position, or an arbitrary position. Moreover, the final condition constraints, which ensure that the simulator has an arbitrary position with zero velocities at the end of the scenario.

[6]:
```
terminal_constraints = ct.struct_MX([
            ct.entry("initial", expr = scenario.initialState),
            ct.entry("final", expr = scenario.finalState)
            ])

lb_terminal = terminal_constraints()
ub_terminal = terminal_constraints()

if cfg.initial_prepositioning == True:
    lb_terminal["initial"] = simulator.initialPPStateMin
    ub_terminal["initial"] = simulator.initialPPStateMax

else:
    lb_terminal["initial"] = simulator.initialStateMin
    ub_terminal["initial"] = simulator.initialStateMax

lb_terminal["final"] = simulator.finalStateMin
ub_terminal["final"] = simulator.finalStateMax

scenario.terminalConstraint = ct.Inequality(expr=terminal_constraints,␣
 ↪lb=lb_terminal, ub=ub_terminal,␣
 ↪nominal=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])
```

1.2.5 Init optimizer

[7]:
```
optimizer = optimizer.Optimizer()
optimizer.optimizationOptions['ipopt'] = cfg.solver
optimizer.samplingTime = cfg.SAMPLE_TIME
optimizer.jit = cfg.jit
optimizer.parallelization = cfg.parallelization
optimizer.numCollocationPoints = cfg.order_collocation
optimizer.optimizationOptions['ipopt.print_level'] = 5
```

1.2.6 Load data

[8]:
```
data = input.load_npy_motion(cfg.DATA_PATH)
ref_sensory_signal = transform.recordedMotionToSensorySignal(data)

t_ref = ref_sensory_signal.time
y_ref = np.array(ref_sensory_signal.inertialSignal(t_ref))
dt_ref = t_ref[1] - t_ref[0]

result_ref = {'t': t_ref,
              'y': y_ref
              }
```

2 Classical Washout

This is the Classical Washout algorithm that is similar to the algorithm used in SRS. First, we import the run_cw16 function, that automatically imports the matlab engine and all the required Classical Washout parameters.

[9]:
```
from mpcmsf.cw16_run import run_cw16
```

```
Starting matlab.engine and initialising matlab workspace...
```

2.1 Run Classical Washout Simulink model

[10]:
```
cw_start_time = time.time()
cw_input_vector, cw_state_vector, cw_motion = run_cw16(cfg.DATA_PATH)
cw_sensory_signal = transform.recordedMotionToSensorySignal(cw_motion)
opt_time_cw = time.time() - cw_start_time
print("total cw run time: ", opt_time_cw)
```

```
Running CW16 simulink model...
total cw run time:  35.47042393684387
```

2.2 Gather results

We also need to interpolate the result to the reference motion sampling time to calculate the RMSE and PCC values and to easily plot all the results in one figure.

[11]:
```
t_cw = cw_sensory_signal.time
y_cw = np.array(cw_sensory_signal.inertialSignal(cw_sensory_signal.time))
y_cw_ref_dt = np.array(cw_sensory_signal.inertialSignal(ref_sensory_signal.
 ↪time))
cw_input_traj = cw_input_vector.T
cw_state_traj = cw_state_vector.T
cw_q_traj = transform.state_to_actuator_trajectory(simulator, cw_state_traj)
rmse_cw = cost_analysis.calculate_rmse(y_ref[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)], y_cw_ref_dt[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)])
pcc_cw = cost_analysis.calculate_pcc(y_ref[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)], y_cw_ref_dt[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)])

result_cw = {'t' : t_cw,
```

```
    'y' : y_cw,
    'u' : cw_input_traj,
    'x' : cw_state_traj,
    'q' : cw_q_traj,
    'rmse': rmse_cw,
    'pcc': pcc_cw,
    'opt_time': opt_time_cw}
```

3 Oracle

The Oracle is the single optimal control problem that optimizes the entire scenario in once.

3.1 Run optimizer

[12]:
```
oracle_start_time = time.time()
result_oracle = optimizer.optimize(scenario, [ref_sensory_signal], W_N,␣
 ↪W_udot)
opt_time_oracle = time.time() - oracle_start_time
print("total optimising time: ", opt_time_oracle)
```

```
This is OMSF trajectory optimizer. jit is OFF, parallelization is set to serial
Initializing NLP...
Initializing NLP solver. Please be patient, this may take several minutes...
Starting optimization.

******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open source code under the Eclipse Public License (EPL).
         For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

This is Ipopt version 3.12.3, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:   130434
Number of nonzeros in inequality constraint Jacobian.:   160062
Number of nonzeros in Lagrangian Hessian...:   215631

Total number of variables...:    28911
                     variables with only lower bounds:        0
                variables with lower and upper bounds:    13350
                     variables with only upper bounds:        0
Total number of equality constraints...:    24471
Total number of inequality constraints...:     8898
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:     8898
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  3.6594059e+02 1.96e+01 5.65e-03  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
  ...
  30  7.2770899e-01 3.61e-10 2.91e-10  -8.6 1.31e-04    -  1.00e+00 1.00e+00h  1

Number of Iterations...: 30
```

```
                                   (scaled)                 (unscaled)
Objective...:   7.2770898525444705e-01    7.2770898525444705e-01
Dual infeasibility...:   2.9099464958916325e-10    2.9099464958916325e-10
Constraint violation...:   3.6136152847565965e-10    3.6136152847565965e-10
Complementarity...:   3.7000137744367503e-09    3.7000137744367503e-09
Overall NLP error...:   3.7000137744367503e-09    3.7000137744367503e-09


Number of objective function evaluations             = 31
Number of objective gradient evaluations             = 31
Number of equality constraint evaluations            = 31
Number of inequality constraint evaluations          = 31
Number of equality constraint Jacobian evaluations   = 31
Number of inequality constraint Jacobian evaluations = 31
Number of Lagrangian Hessian evaluations             = 30
Total CPU secs in IPOPT (w/o function evaluations)   =      5.479
Total CPU secs in NLP function evaluations           =      8.597

EXIT: Optimal Solution Found.
OmsfNlpSolver  :   t_proc      (avg)   t_wall      (avg)    n_eval
        nlp_f  |  30.05ms (969.32us)  33.91ms (  1.09ms)        31
        nlp_g  |  71.16ms (  2.30ms)  76.06ms (  2.45ms)        31
   nlp_grad_f  | 102.94ms (  3.22ms) 123.94ms (  3.87ms)        32
   nlp_hess_l  |   5.94 s (197.93ms)   6.22 s (207.20ms)        30
    nlp_jac_g  |   2.62 s ( 81.73ms)   2.81 s ( 87.86ms)        32
        total  |  14.73 s ( 14.73 s)  15.97 s ( 15.97 s)         1
total optimising time:  20.124104976654053
```

3.2 Gather results

We also need to interpolate the result to the reference motion sampling time to calculate the RMSE and PCC values and to easily plot all the results in one figure.

[13]:
```
[sim_trajectory] = result_oracle['trajectory']
t_sim = sim_trajectory.time
y_sim = np.array(sim_trajectory.inertialSignal(sim_trajectory.time))
y_orc_ref_dt = np.array(sim_trajectory.inertialSignal(ref_sensory_signal.
 ↪time)) # needed for calculating rmse and pcc
sim_input_traj = sim_trajectory.input(sim_trajectory.time)
sim_state_traj = sim_trajectory.platformState(sim_trajectory.time)
sim_q_traj = transform.state_to_actuator_trajectory(simulator, sim_trajectory.
 ↪platformState(sim_trajectory.time))
sim_total_cost = sim_trajectory.cost(sim_trajectory.time)
sim_input_cost, sim_state_cost, sim_inertial_signal_cost = cost_analysis.
 ↪recalculate_cost_function(ref_sensory_signal, sim_trajectory, cfg.weights)
rmse = cost_analysis.calculate_rmse(y_ref[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)], y_orc_ref_dt[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)])
pcc = cost_analysis.calculate_pcc(y_ref[:,int(cfg.T_FADE_IN/dt_ref):-int(cfg.
 ↪T_FADE_OUT/dt_ref)], y_orc_ref_dt[:,int(cfg.T_FADE_IN/dt_ref):-int(cfg.
 ↪T_FADE_OUT/dt_ref)])

result_oracle = {'t' : t_sim,
    'y' : y_sim,
```

```
    'u' : sim_input_traj,
    'x' : sim_state_traj,
    'q' : sim_q_traj,
    'c' : sim_total_cost,
    'c_y': sim_inertial_signal_cost,
    'c_u': sim_input_cost,
    'c_x': sim_state_cost,
    'rmse': rmse,
    'pcc': pcc,
    'opt_time': opt_time_oracle}
```

4 MPC

The MPC is the offline MPC algorithm. We change some of the optimizer settings, so that less output is printed to the console.

[14]:
```
optimizer.optimizationOptions['ipopt.print_level'] = 0
optimizer.optimizationOptions['print_time'] = 0
```

4.1 Run optimizer

[15]:
```
mpc_start_time = time.time()
t_sim_mpc, y_sim_mpc, sim_input_traj_mpc, sim_state_traj_mpc, sim_q_traj_mpc,␣
 ↪sim_total_cost_mpc = mpc.run_scenario_with_mpc(simulator, scenario,␣
 ↪optimizer, data, W_N, W_udot)
opt_time_mpc = time.time() - mpc_start_time
print("total mpc optimising time: ", opt_time_mpc)
```

```
Starting MPC optimisation...
Input data sample time:  0.01  seconds
MPC sample time:  0.05  seconds
Total number MPC iterations:  740
Horizon lenght in seconds:  6
Horizon prediction strategy:  PERFECT
MPC ITERATION:  1 / 740          HORIZON LENGTH:  6.0 s          DURATION:
1.697 s
...
MPC ITERATION:  740 / 740          HORIZON LENGTH:  0.12 s          DURATION:
0.119 s
total mpc optimising time:  1445.113517999649
```

4.2 Gather results

We also need to interpolate the result to the reference motion sampling time to calculate the RMSE and PCC values and to easily plot all the results in one figure.

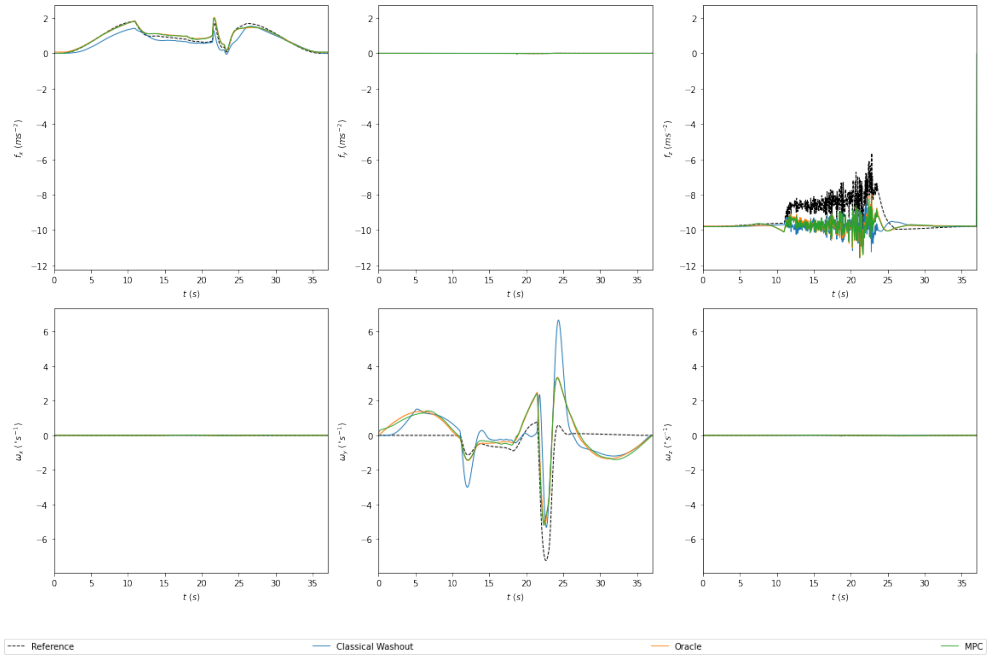
[16]:
```
y_mpc_ref_dt = util.interpolate(t_ref, t_sim_mpc, y_sim_mpc)
rmse = cost_analysis.calculate_rmse(y_ref[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)], y_mpc_ref_dt[:,int(cfg.T_FADE_IN/dt_ref):
 ↪-int(cfg.T_FADE_OUT/dt_ref)])
pcc = cost_analysis.calculate_pcc(y_ref[:,int(cfg.T_FADE_IN/dt_ref):-int(cfg.
 ↪T_FADE_OUT/dt_ref)], y_mpc_ref_dt[:,int(cfg.T_FADE_IN/dt_ref):-int(cfg.
 ↪T_FADE_OUT/dt_ref)])
```

```
result_mpc = {'t' : t_sim_mpc,
              'y' : y_sim_mpc,
              'u' : sim_input_traj_mpc,
              'x' : sim_state_traj_mpc,
              'q' : sim_q_traj_mpc,
              'c' : sim_total_cost_mpc,
              'rmse': rmse,
              'pcc': pcc,
              'opt_time': opt_time_mpc}
```

5 Plot and save

Next, the figures are created. Here we only plot a single figure which includes the inertial signal of the reference motion and the inertial signal produced by the three different algorithms. However, there are also options to plot actuator length, actuator velocity, actuator acceleration, simulator state, and simulator control input.

[17]:
```
figure_names = []
figure_names.append(plotting.
 ↪plot_inertial_signal("ref_sim_cw_inertial_signal_mpc.pdf", ["Reference",␣
 ↪"Classical Washout", "Oracle", "MPC"], t_ref, y_ref[:6,:], np.
 ↪stack((y_cw_ref_dt[:6,:], y_orc_ref_dt[:6,:], y_mpc_ref_dt[:6,:]), axis=2)))
```



5.1 Save all the results and plots

[18]:
```
output.save_optimization_result(figure_names, ref_result=result_ref,␣
 ↪oracle_result=result_oracle, mpc_result=result_mpc, cw_result=result_cw,␣
 ↪extra_name_str=cfg.save_name)
```

Appendix B

Experiment instructions

In this appendix, the experiment briefing and information sheet are presented that were used during the experiment in the SRS for informing the participants. In Figure B.1, the sheet used to inform the participants about the scenarios is shown. From the next page, the experiment briefing is presented. Note that the briefing is in Dutch.

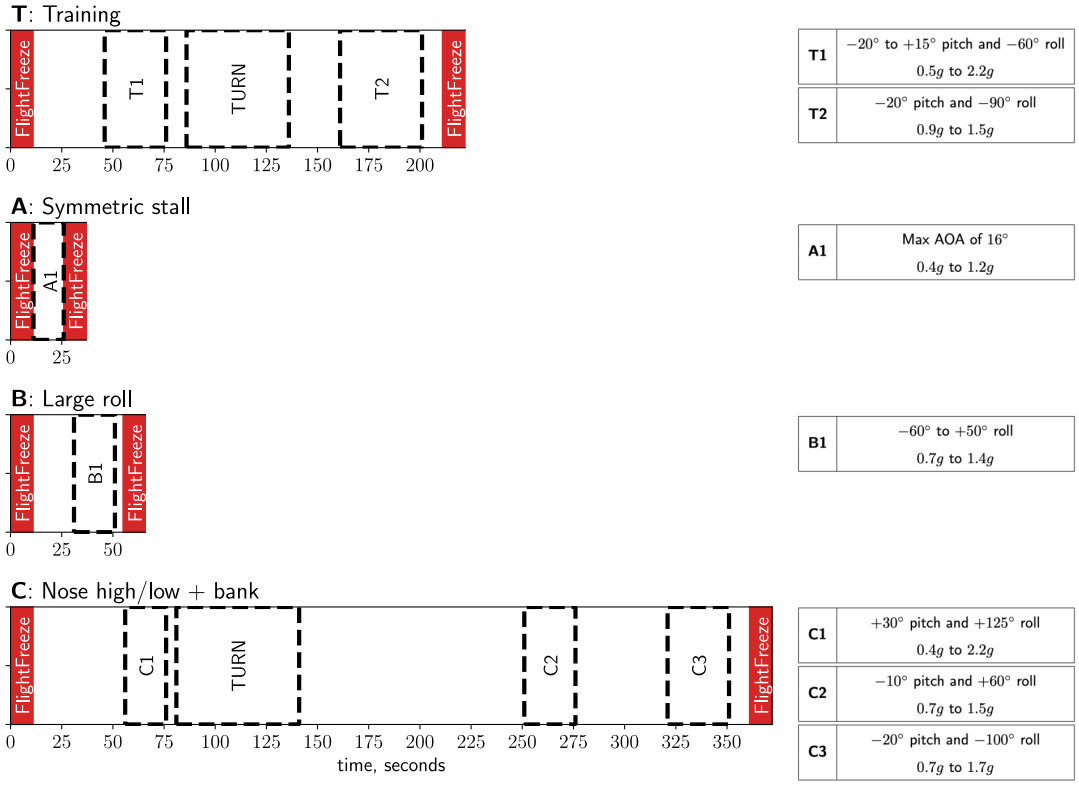


Figure B.1: Scenario definition sheet used during the experiment in the SIMONA Research Simulator (SRS).

Experiment Omschrijving

Tijdens dit experiment zijn we geïnteresseerd in hoe jij de kwaliteit van de bewegingen van de simulator ervaart.

Het experiment zal bestaan uit verschillende runs waarin elke keer een andere motion cueing strategie wordt gebruikt. De runs zullen bestaan uit vooraf gesimuleerde upset en/of stall scenario's. Welk scenario de desbetreffende run bevat zal voor de run duidelijk gemaakt worden. Tijdens de runs is het niet nodig om de simulator te besturen. Probeer je voor te stellen dat je met een collega piloot meevliegt.

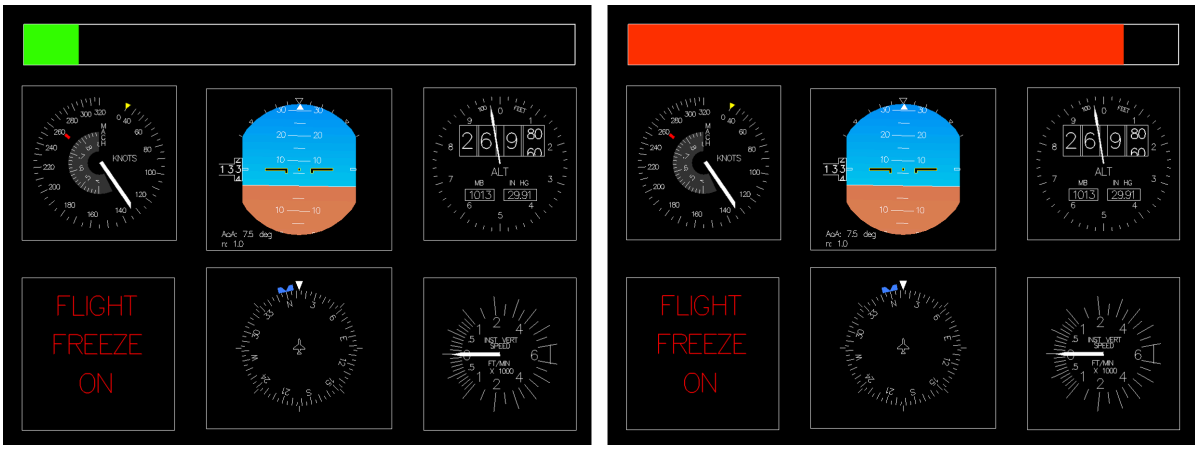
Tijdens het experiment zullen er visuals die normaal gesproken vanuit de cockpit te zien zijn afgespeeld worden. Daarnaast zijn er een primary flight display (PFD) en een engine display zichtbaar. De visuals en de PFD zullen de waarheidsgetrouwe vliegtuigbewegingen en status aangeven. Daarentegen zal de motion die je voelt nooit helemaal overeenkomen met de werkelijkheid door de gelimiteerde mogelijkheden van een hexapod simulator.

Hieronder is het beeld weergegeven wat jij zult zien tijdens het experiment vanuit de rechter stoel van de simulator. Op de PFD zijn twee extra indicatoren te zien. De eerste indicator is de flight freeze indicator, deze indicator is bedoeld om aan te geven of de experiment run begonnen is. Dit is nodig omdat de simulator voor en na elke run zichzelf voelbaar zal positioneren wat niet meegenomen moet worden in de beoordeling. De rode indicator zal verdwijnen wanneer de run begonnen is. De tweede indicator is de motion rating bar, op de volgende pagina hier meer over.

Taak tijdens de experiment runs

Tijdens de experiment runs word je gevraagd de mate van mismatch tussen de visuals en PFD en de motion die ervaart aan te geven met behulp van een draaiknop, hieronder afgebeeld, die je in je hand zult vasthouden.

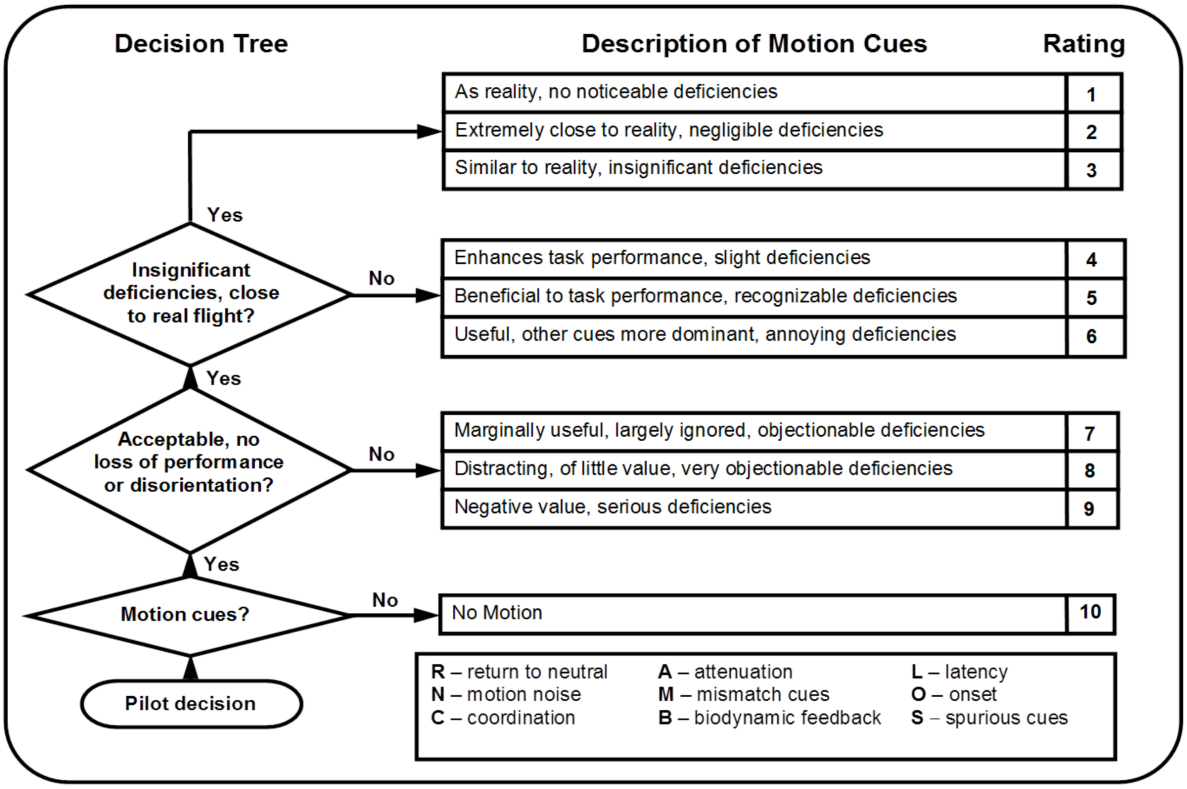
Met deze draaiknop is het mogelijk om de motion rating bar op de PFD aan te sturen. Een goede rating (simulator motion komt goed overeen met de visuals en PFD) wordt aangeven met een groene kleur. Wanneer de draaiknop met de klok meegedraaid wordt zal de motion rating bar oplopen en veranderen van kleur (groen-geel-oranje-rood). Hoe voller de bar gevuld is, hoe slechter jij vindt hoe de simulator motion overeenkomt met de visuals en PFD. De beste rating is dus een lege bar, en de slechtste rating een geheel gevulde bar. Hieronder is een goede motion rating, links, en een slechte motion rating, rechts, weergegeven.



Aan het begin van iedere run zal de rating bar leeg zijn en dus een goede rating aangeven. Het is zeer belangrijk dat gedurende de hele run, de rating continu en actief door jou zal worden gegeven.

Taak na elke experiment run

Na elke run zal er gevraagd worden om aan te geven hoe goed jij de simulator motion vond voor de gehele run. Dit wordt gedaan met behulp van de Motion Fidelity Rating (MFR), te zien in de figuur hieronder:



Verder zal er aan de hand van de onderstaande tabel, de MIsery SCale (MiSC), gevraagd worden hoe erg de wagenziekte is die je eventueel ervaart.

Symptoms		**MISC**
No problems		0
Some discomfort, but no specific symptoms		1
Dizziness, cold/warm, headache,	Vague	2
stomach/throat awareness, sweating,	Little	3
blurred vision, yawning, burping,	Rather	4
tiredness, salivation, ... but no nausea	Severe	5
	Little	6
Nausea	Rather	7
	Severe	8
	Retching	9
Vomiting		10

Zowel de MFR figuur als de MISC tabel zullen in de simulator aanwezig zijn voor referentie.

Tot slot

De verwachte duur van het experiment in de simulator is 90 minuten. Halverwege het experiment is er een lange pauze waarin je uit de simulator kunt stappen. Tussen de runs in is er ook altijd de mogelijkheid tot een pauze wanneer jij aangeeft daar behoefte aan te hebben. Na afloop van het experiment zullen wij nog een debriefing doen, er is dan ook de mogelijkheid om wat meer details te geven over ons onderzoek. Ik verwacht dat we 2 tot 2,5 uur na jouw aankomst weer klaar zullen zijn met alles. Wanneer je vooraf, tijdens of na het experiment vragen hebt, beantwoord ik die graag!

Appendix C

Experiment matrix

Table C.1: Experiment scenario and condition order matrix. Scenario and condition order are, respectively, displayed vertically and horizontally.

Participant	Scenario	Condition					
P1	Training	CWO	ORC				
	Symmetric Stall	CWO	ORC	CWO	ORC	ORC	CWO
	Large Roll	ORC	CWO	ORC	ORC	CWO	CWO
	Break						
	Upsets	CWO	ORC	ORC	CWO	CWO	ORC
P2	Training	CWO	CWO	ORC			
	Large Roll	ORC	ORC	CWO	CWO	ORC	CWO
	Break						
	Upsets	ORC	CWO	CWO	ORC	CWO	ORC
	Symmetric Stall	CWO	CWO	ORC	CWO	ORC	ORC
P3	Training	CWO	ORC				
	Upsets	CWO	ORC	CWO	ORC	ORC	CWO
	Break						
	Symmetric Stall	ORC	CWO	ORC	ORC	CWO	CWO
	Large Roll	CWO	ORC	ORC	CWO	CWO	ORC
P4	Training	CWO	ORC				
	Symmetric Stall	ORC	ORC	CWO	CWO	ORC	CWO
	Large Roll	ORC	CWO	CWO	ORC	CWO	ORC
	Break						
	Upsets	CWO	CWO	ORC	CWO	ORC	ORC
P5	Training	CWO	ORC				
	Large Roll	CWO	ORC	CWO	ORC	ORC	CWO
	Break						
	Upsets	ORC	CWO	ORC	ORC	CWO	CWO
	Symmetric Stall	CWO	ORC	ORC	CWO	CWO	ORC
P6	Training	CWO	ORC				
	Upsets	ORC	ORC	CWO	CWO	ORC	CWO
	Break						
	Symmetric Stall	ORC	CWO	CWO	ORC	CWO	ORC
	Large Roll	CWO	CWO	ORC	CWO	ORC	ORC
P7	Training	CWO	ORC				
	Symmetric Stall	CWO	ORC	CWO	ORC	ORC	CWO
	Large Roll	ORC	CWO	ORC	ORC	CWO	CWO
	Break						
	Upsets	CWO	ORC	ORC	CWO	CWO	ORC
P8	Training	CWO	ORC				
	Large Roll	ORC	ORC	CWO	CWO	ORC	CWO
	Break						
	Upsets	ORC	CWO	CWO	ORC	CWO	ORC
	Symmetric Stall	CWO	CWO	ORC	CWO	ORC	ORC
P9	Training	CWO	ORC				
	Upsets	CWO	ORC	CWO	ORC	ORC	CWO
	Break						
	Symmetric Stall	ORC	CWO	ORC	ORC	CWO	CWO
	Large Roll	CWO	ORC	ORC	CWO	CWO	ORC
P10	Training	CWO	ORC				
	Symmetric Stall	ORC	ORC	CWO	CWO	ORC	CWO
	Large Roll	ORC	CWO	CWO	ORC	CWO	ORC
	Break						
	Upsets	CWO	CWO	ORC	CWO	ORC	ORC

Appendix D

Experiment scenarios full solutions

In this appendix, the full solutions of the CWO and ORC conditions for all the experiment scenarios are presented. This includes the output inertial signal, the SRS control input, the SRS commanded state and the differences between the SRS commanded state and actual state and the differences between the SRS commanded actuator lengths and the actual actuator lengths.

D.1. The Training scenario

In Figure D.1, the output inertial signal for the Training scenario is presented and in Figure D.3 the control input. The SRS state is shown in Figure D.2 and in, respectively, Figure D.4 and Figure D.5, the commanded versus actual SRS state and actuator lengths.

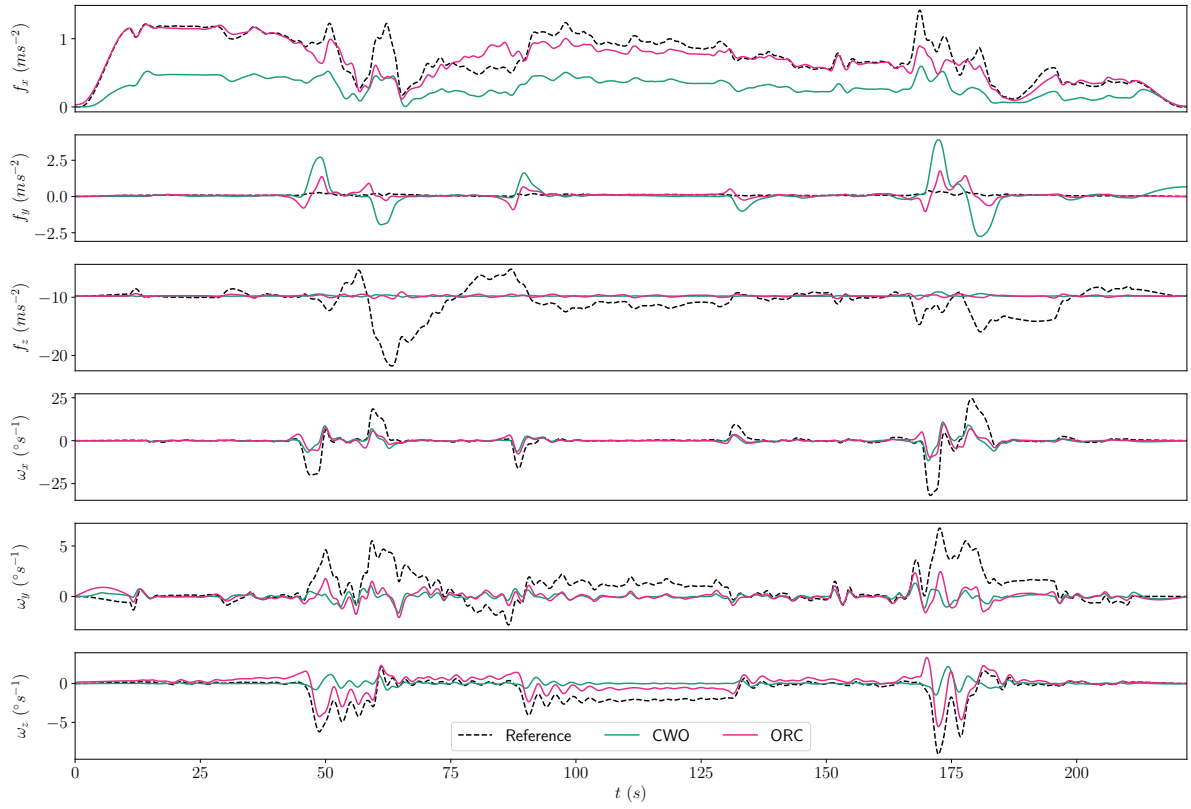


Figure D.1: Output inertial signal for the Training scenario used in the SRS experiment.

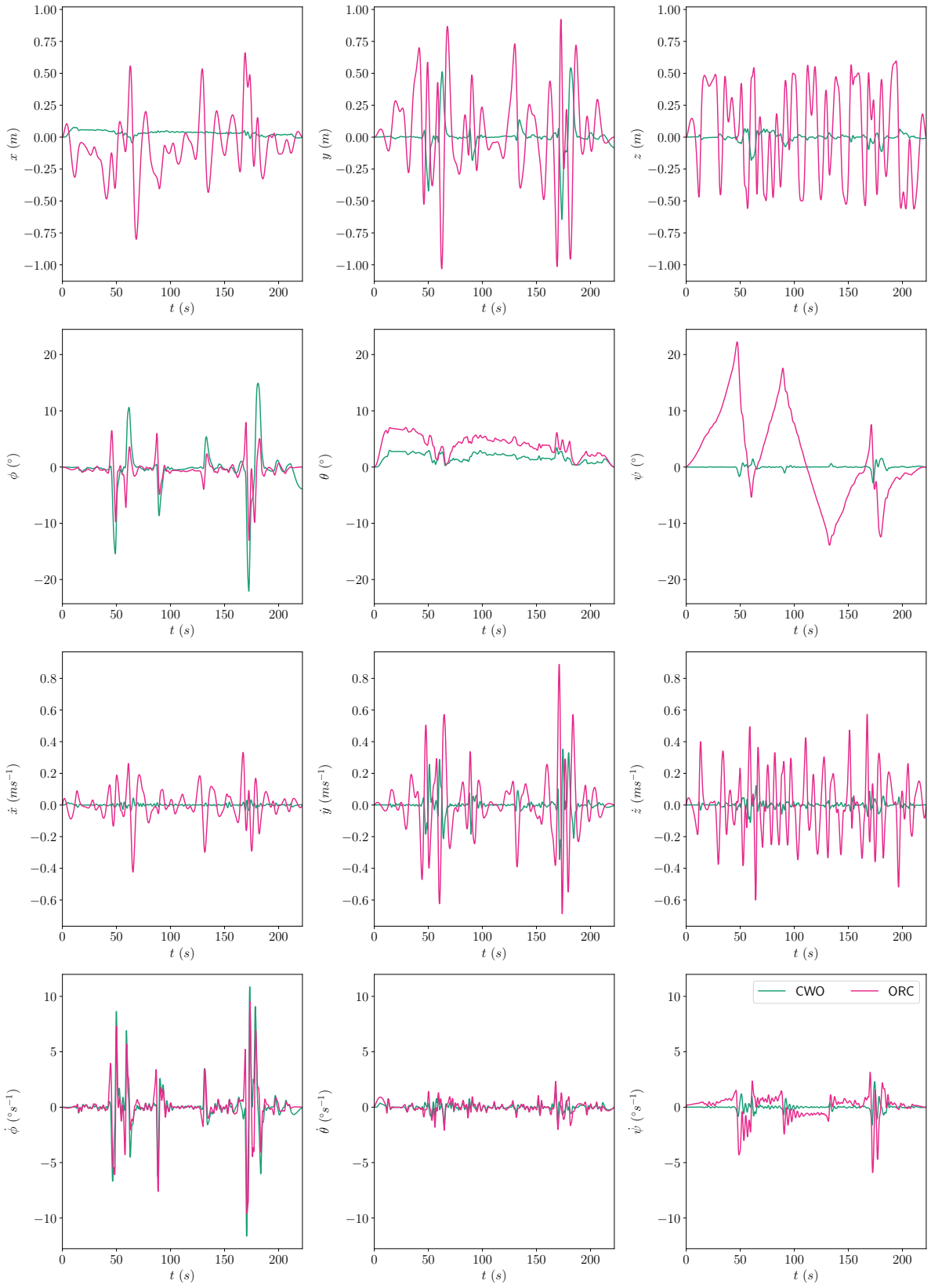


Figure D.2: Simulator state for the Training scenario used in the SRS experiment.

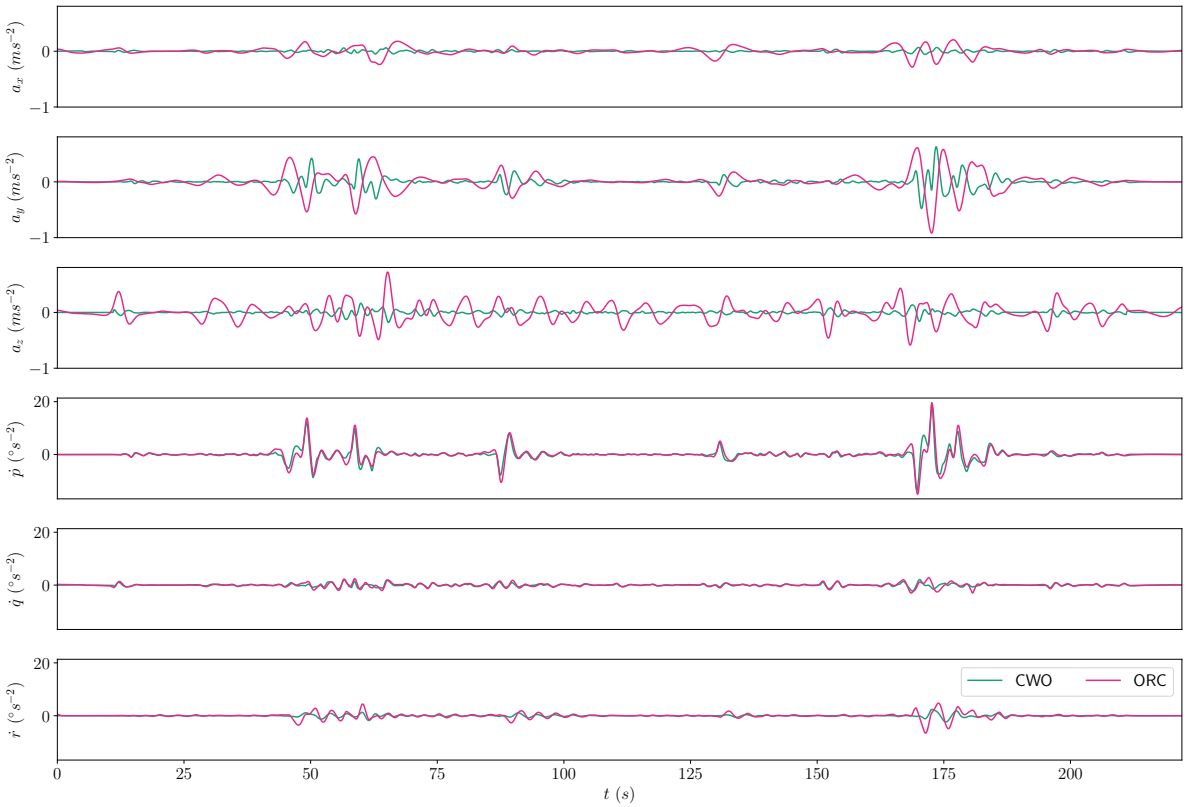


Figure D.3: Simulator control input for the Training scenario used in the SRS experiment.

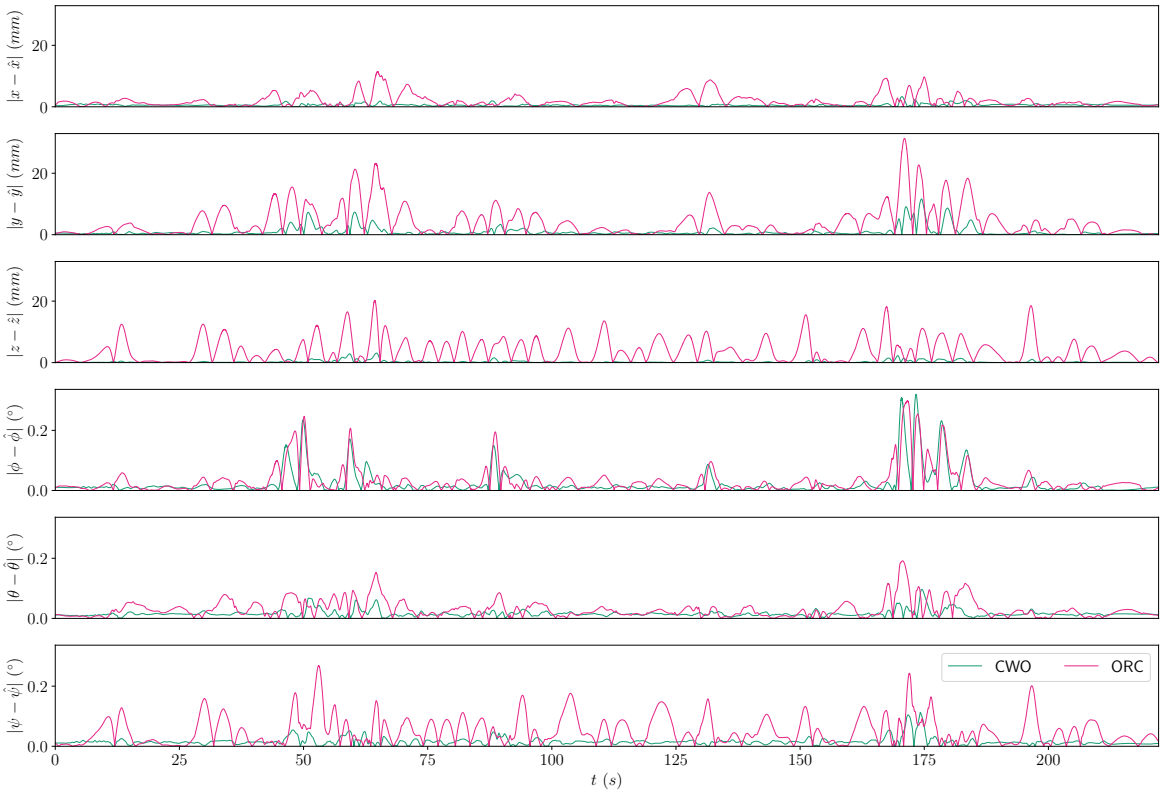


Figure D.4: Actual SRS state and calculated/commanded state difference for the Training scenario.

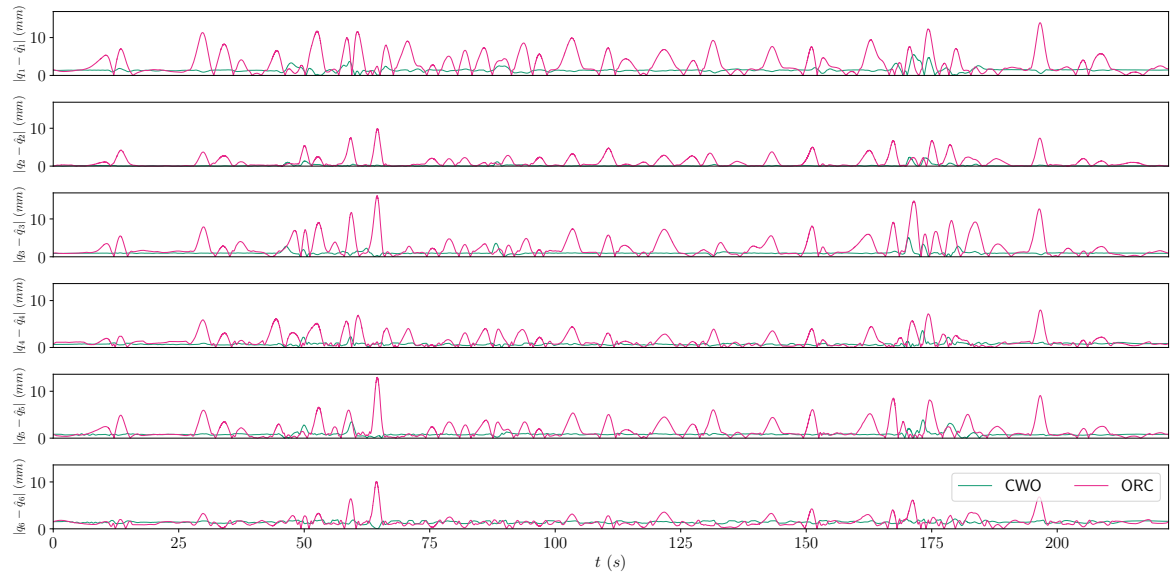


Figure D.5: Actual SRS state and calculated/commanded state difference for the Training scenario.

D.2. The Symmetric Stall scenario

In Figure D.6, the output inertial signal for the Symmetric Stall scenario is presented and in Figure D.8 the control input. The SRS state is shown in Figure D.7 and in, respectively, Figure D.9 and Figure D.10, the commanded versus actual SRS state and actuator lengths.

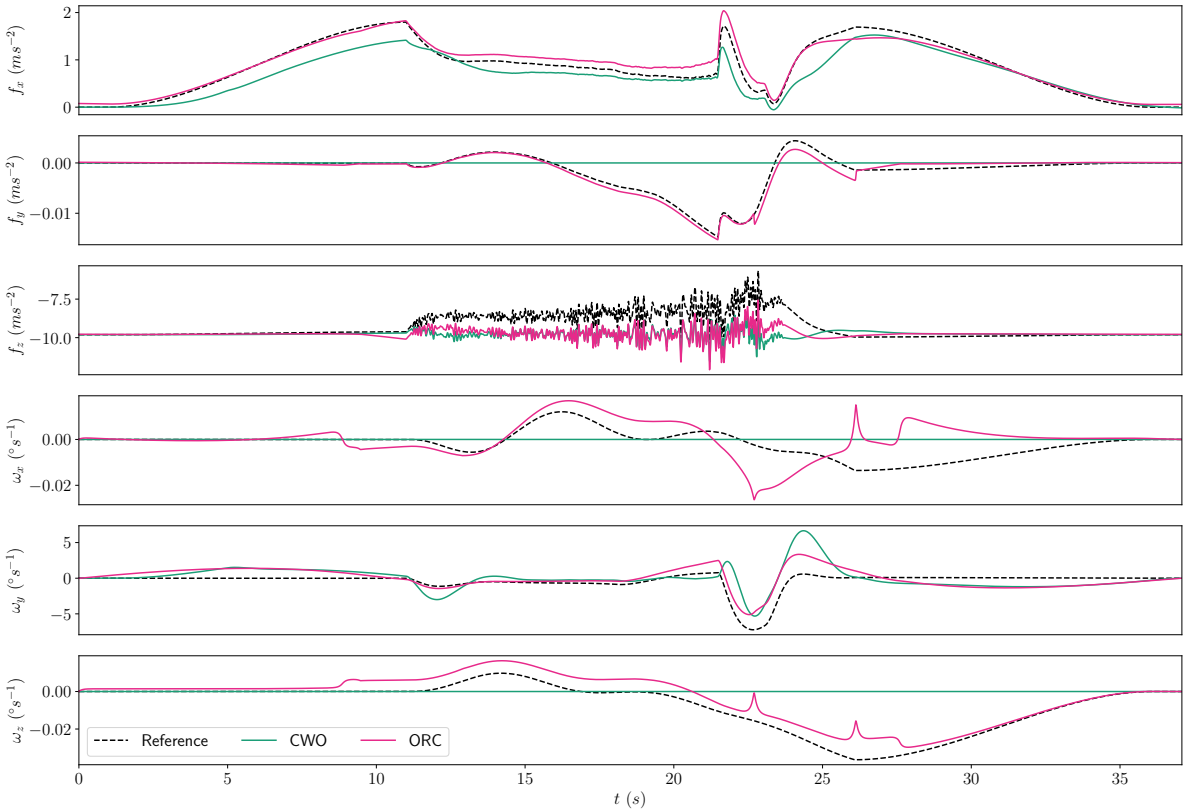


Figure D.6: Output inertial signal for the Symmetric Stall scenario used in the SRS experiment.

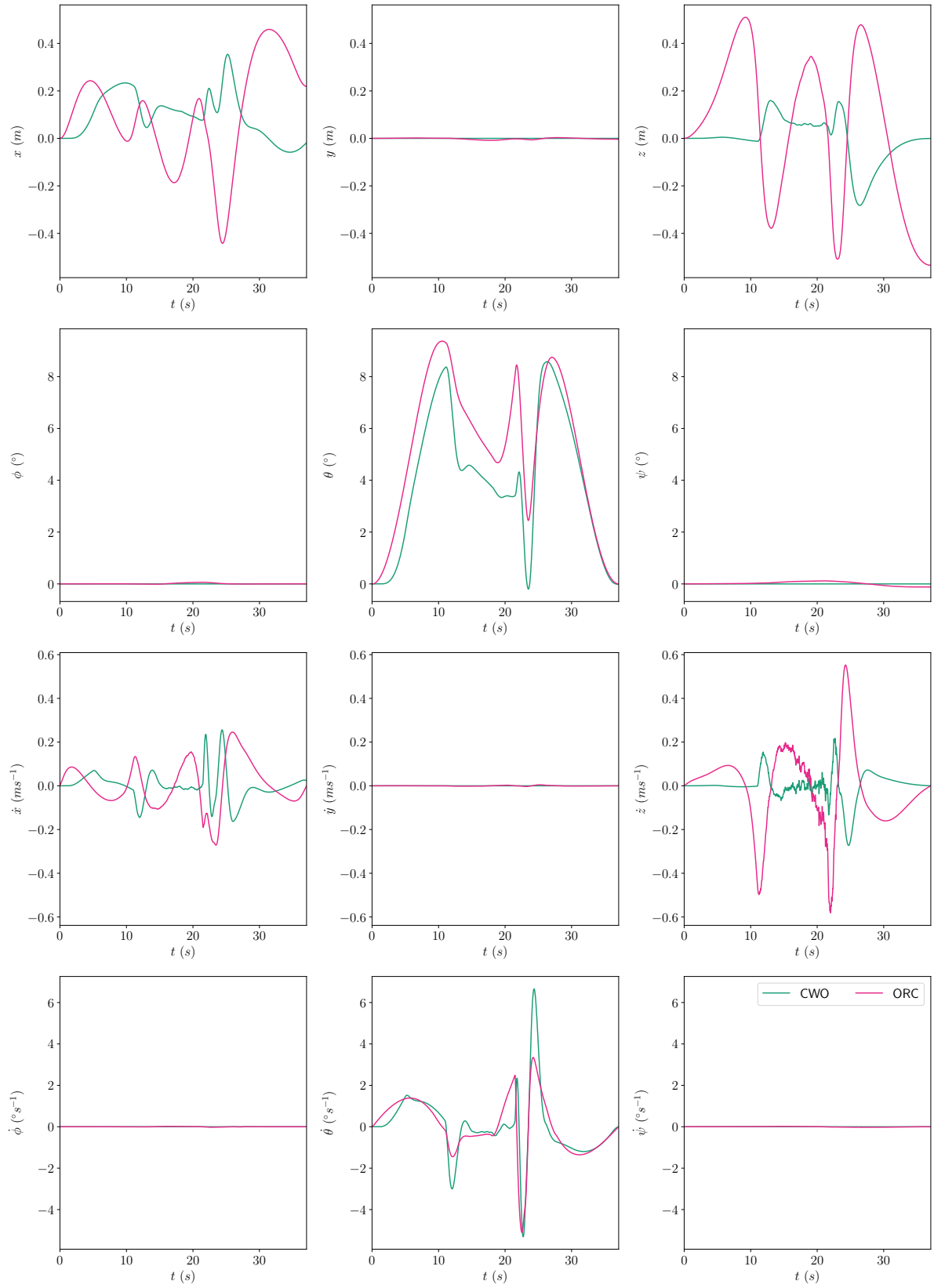


Figure D.7: Simulator state for the Symmetric Stall scenario used in the SRS experiment.

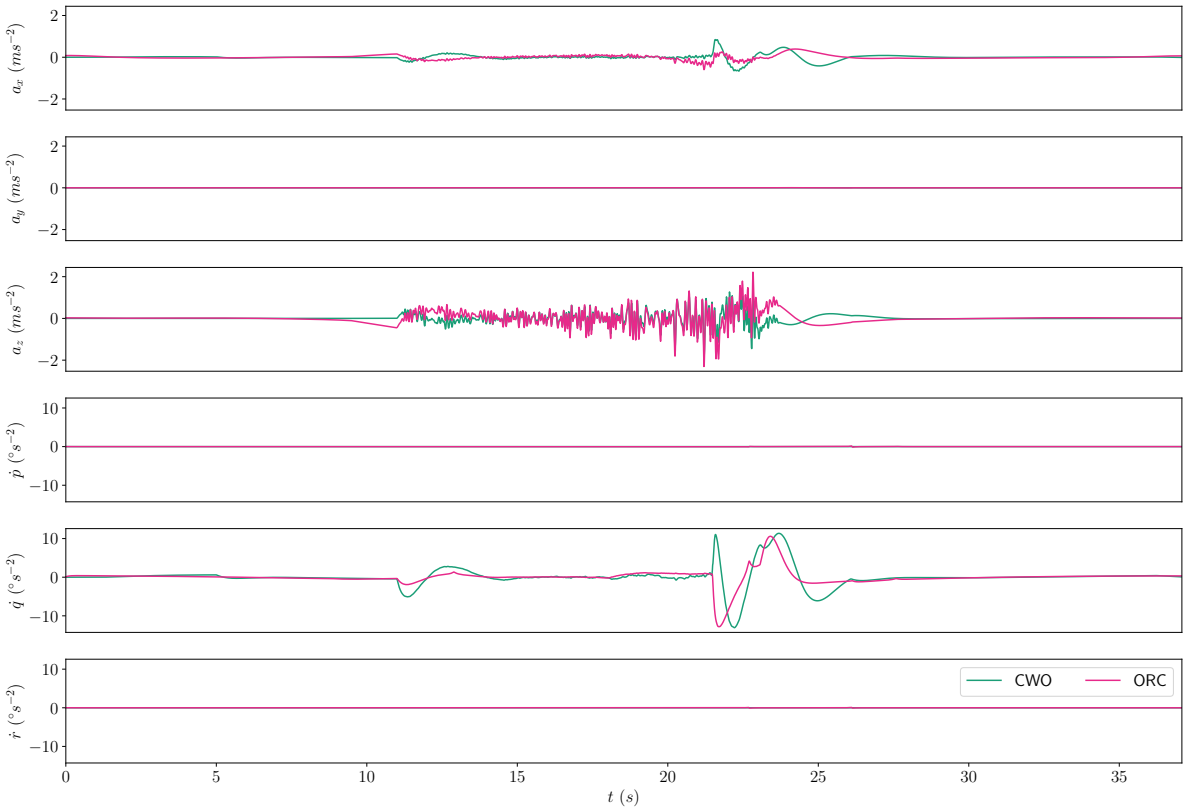


Figure D.8: Simulator control input for the Symmetric Stall scenario used in the SRS experiment.

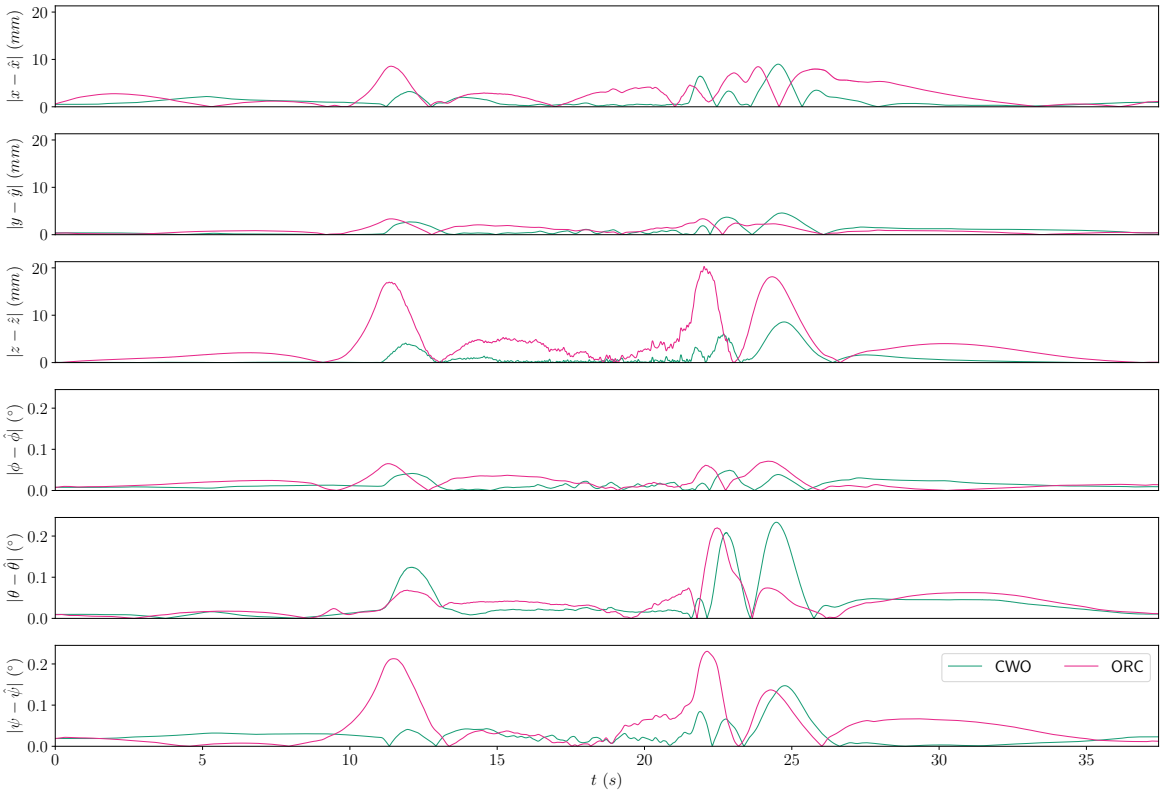


Figure D.9: Actual SRS state and calculated/commanded state difference for the Symmetric Stall scenario.

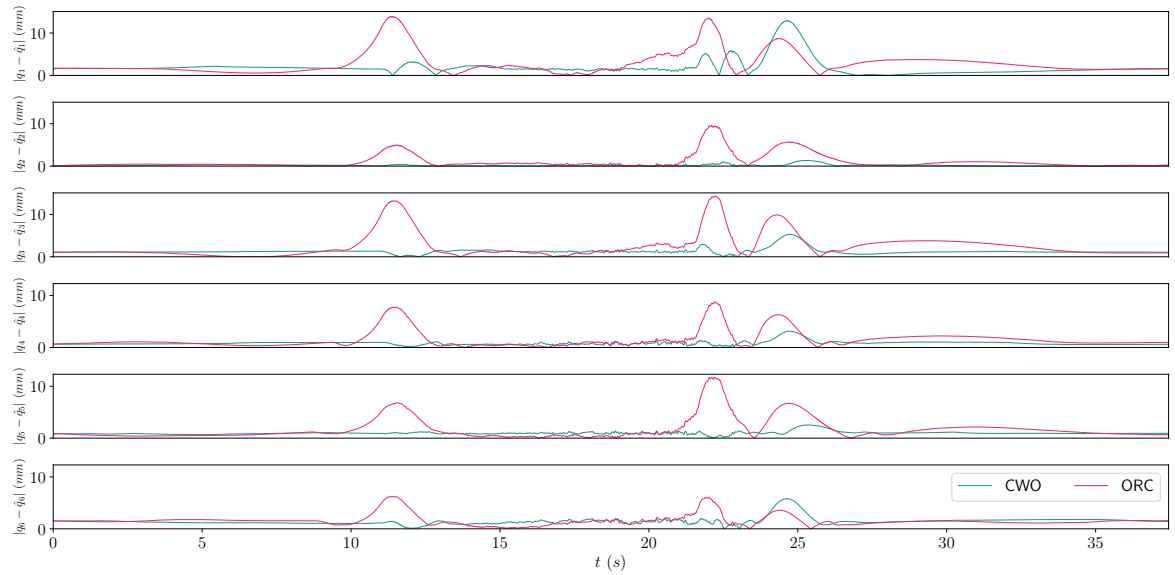


Figure D.10: Actual SRS state and calculated/commanded state difference for the Symmetric Stall scenario.

D.3. The Large Roll scenario

In Figure D.11, the output inertial signal for the Large Roll scenario is presented and in Figure D.13 the control input. The SRS state is shown in Figure D.12 and in, respectively, Figure D.14 and Figure D.15, the commanded versus actual SRS state and actuator lengths.

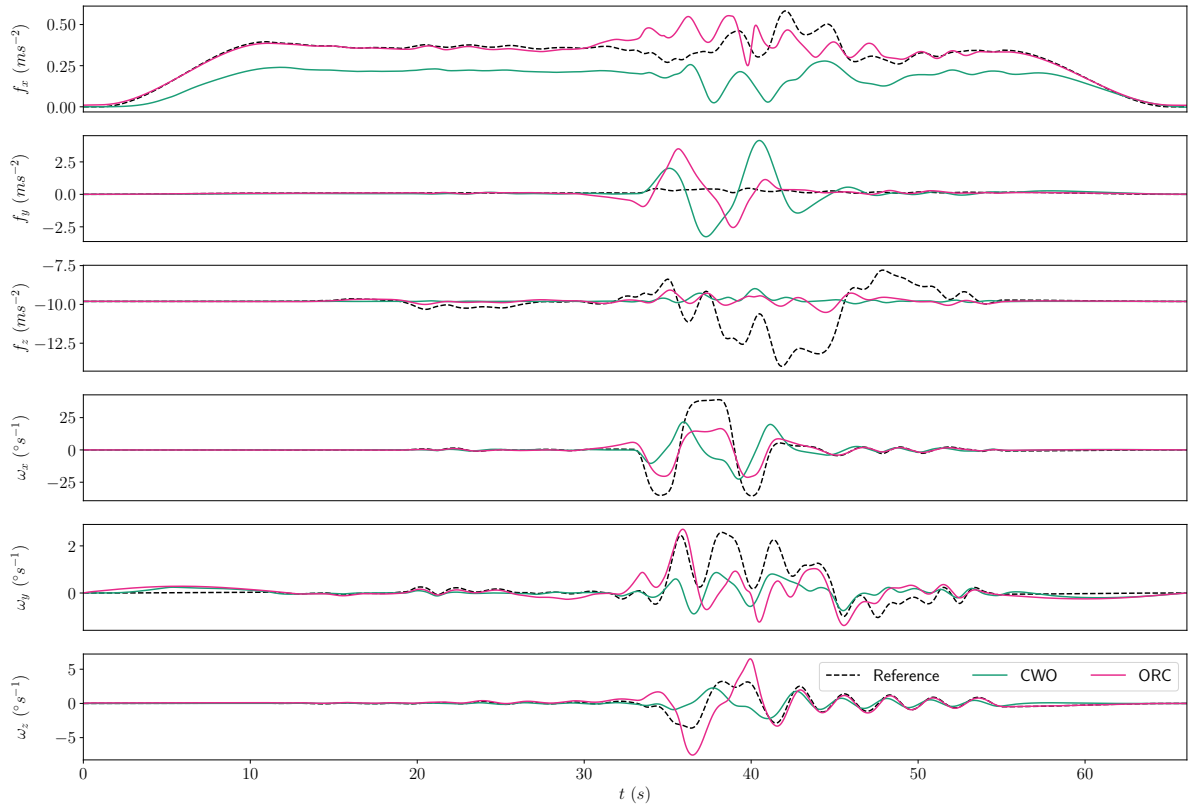


Figure D.11: Output inertial signal for the Large Roll scenario used in the SRS experiment.

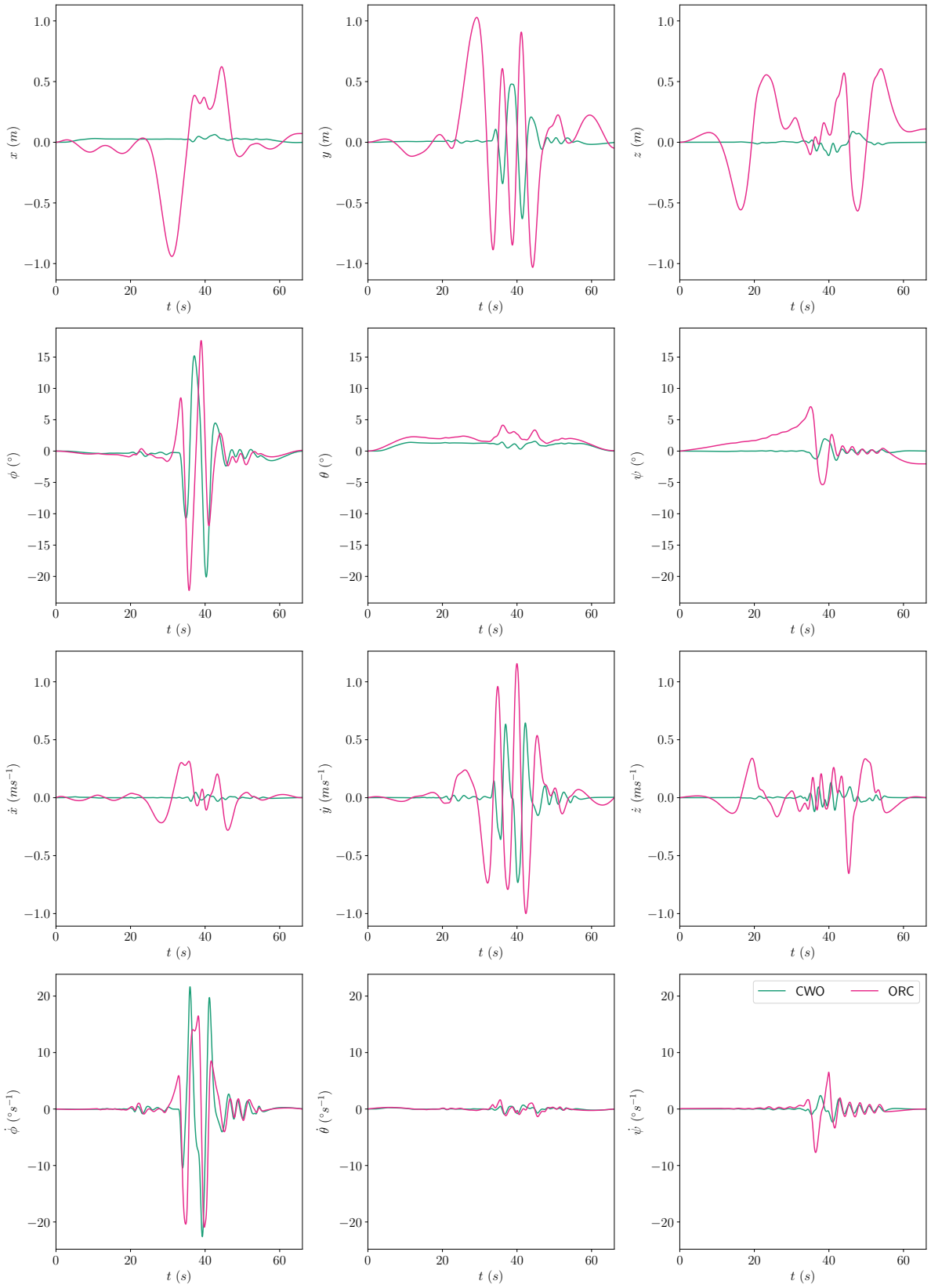


Figure D.12: Simulator state for the Large Roll scenario used in the SRS experiment.

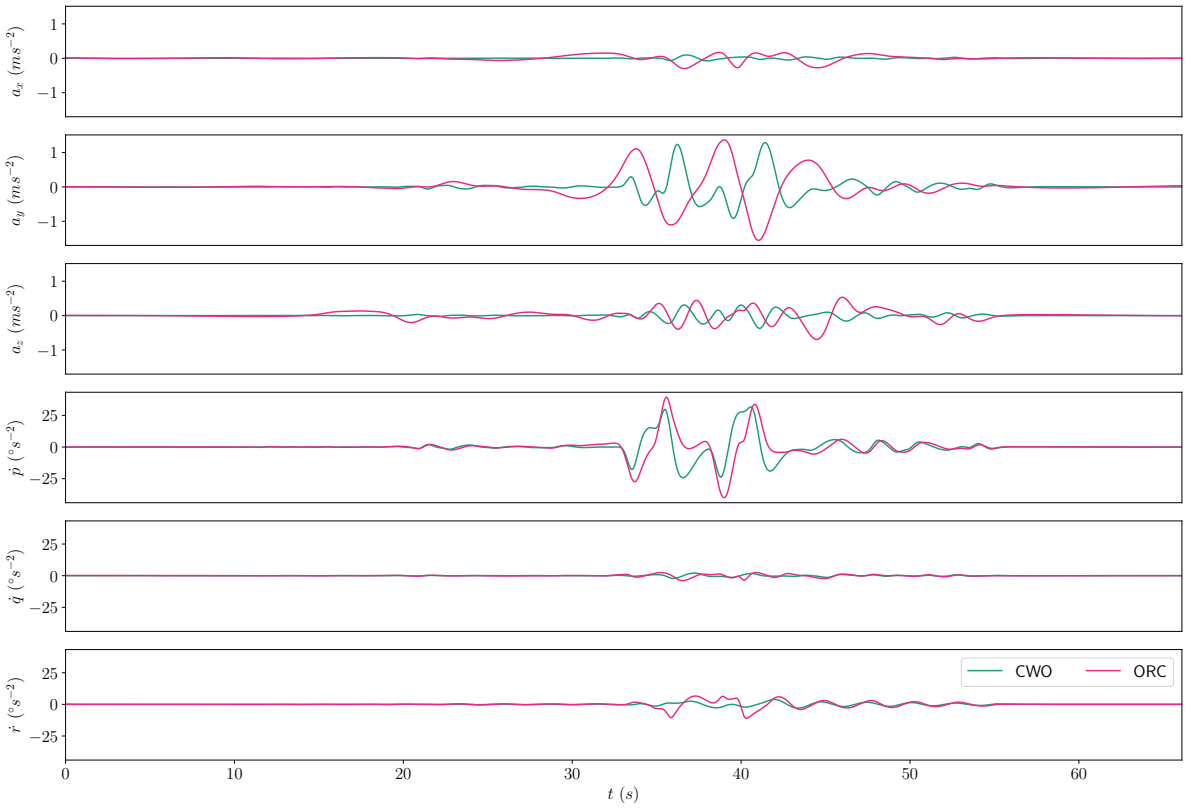


Figure D.13: Simulator control input for the Large Roll scenario used in the SRS experiment.

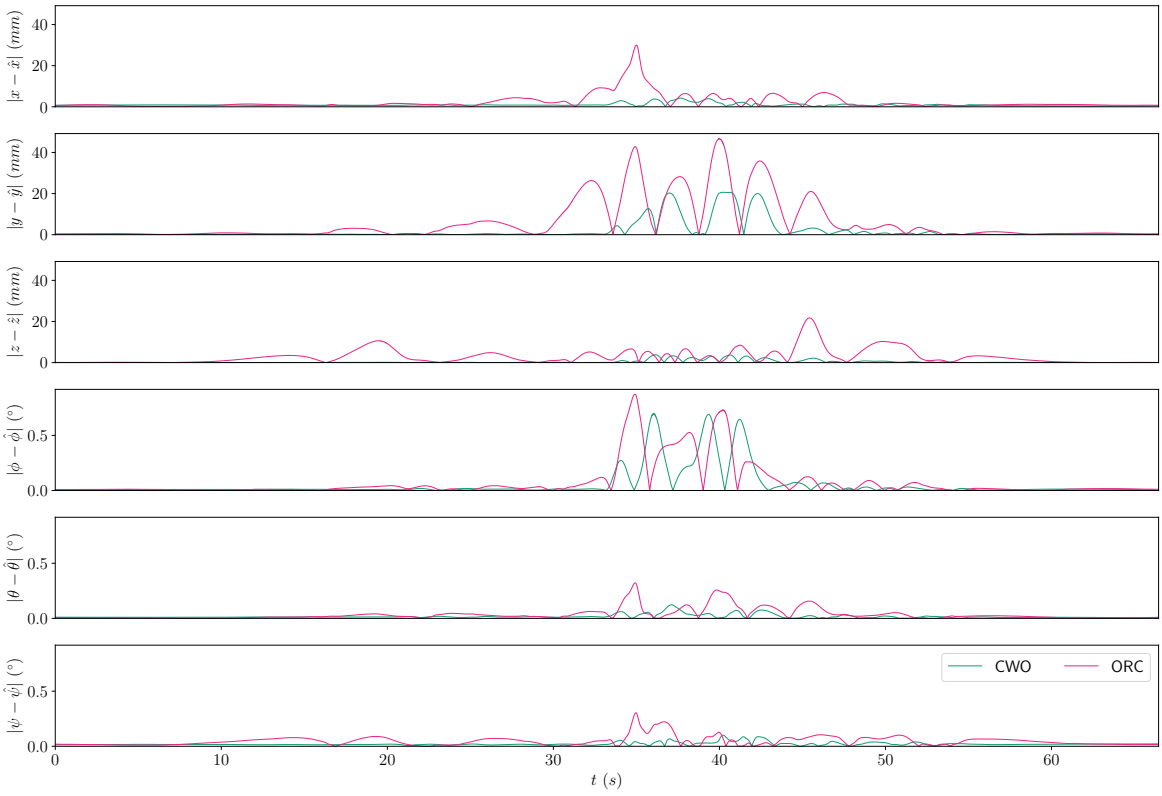


Figure D.14: Actual SRS state and calculated/commanded state difference for the Large Roll scenario.

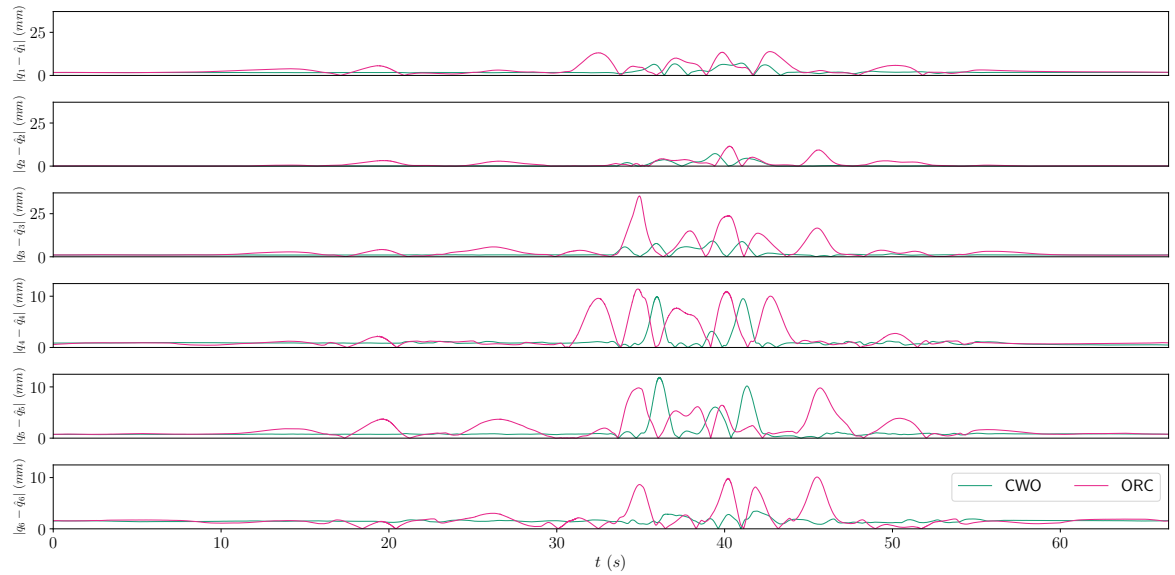


Figure D.15: Actual SRS state and calculated/commanded state difference for the Large Roll scenario.

D.4. The Upsets scenario

In Figure D.16, the output inertial signal for the Upsets scenario is presented and in Figure D.18 the control input. The SRS state is shown in Figure D.17 and in, respectively, Figure D.19 and Figure D.20, the commanded versus actual SRS state and actuator lengths.

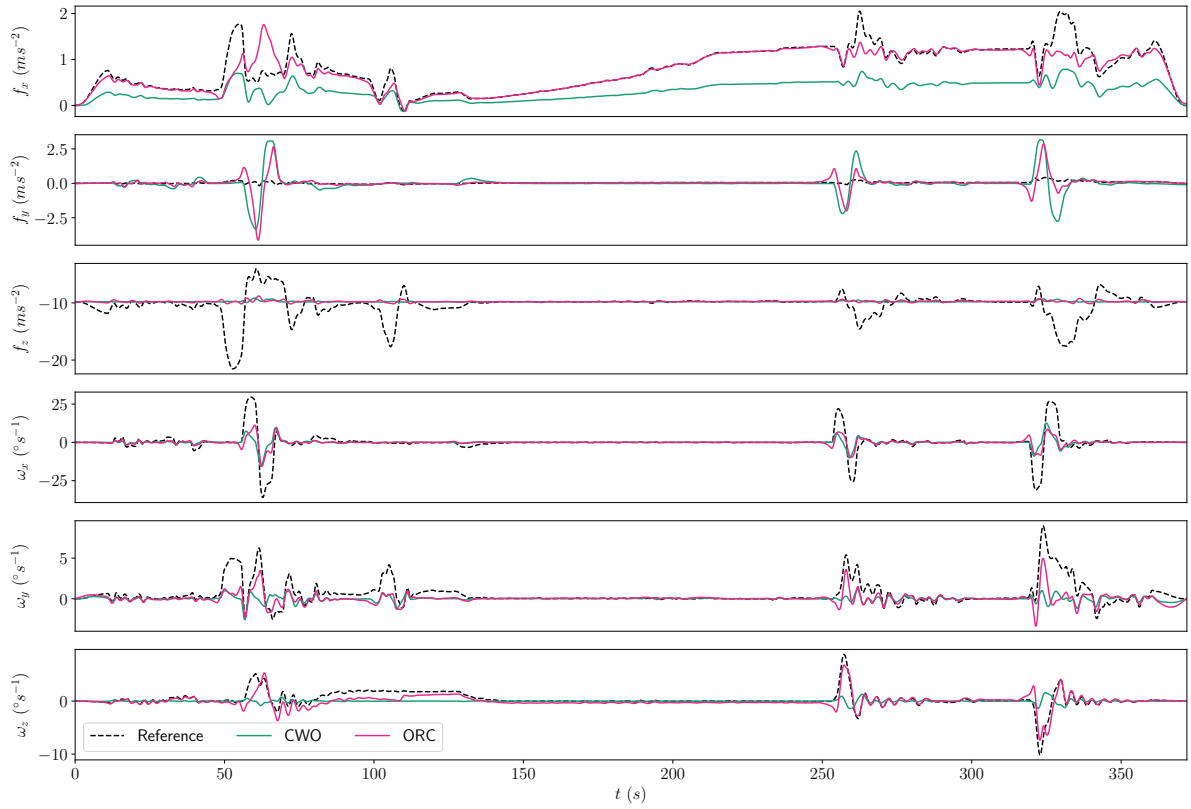


Figure D.16: Output inertial signal for the Upsets scenario used in the SRS experiment.

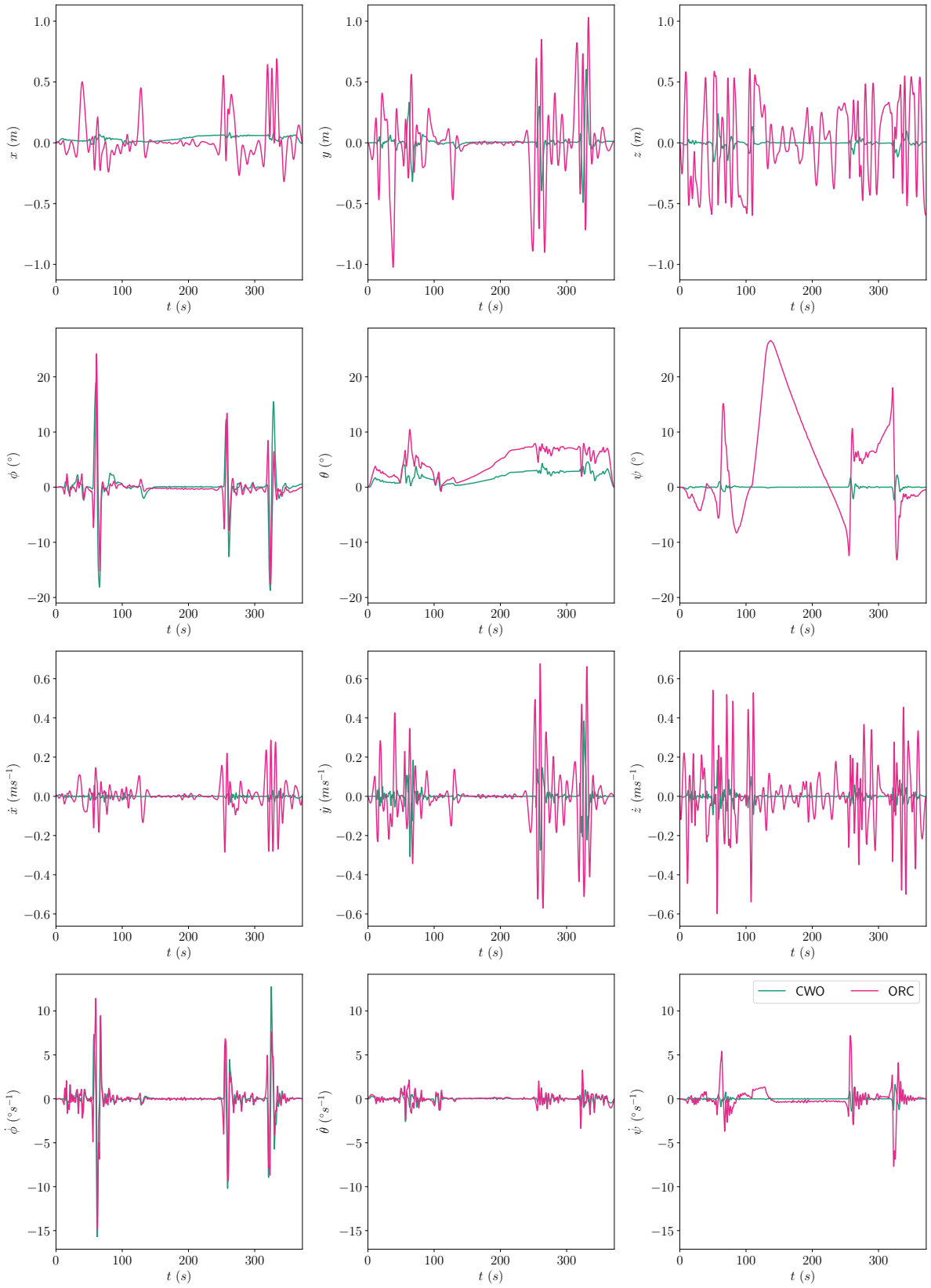


Figure D.17: Simulator state for the Upsets scenario used in the SRS experiment.

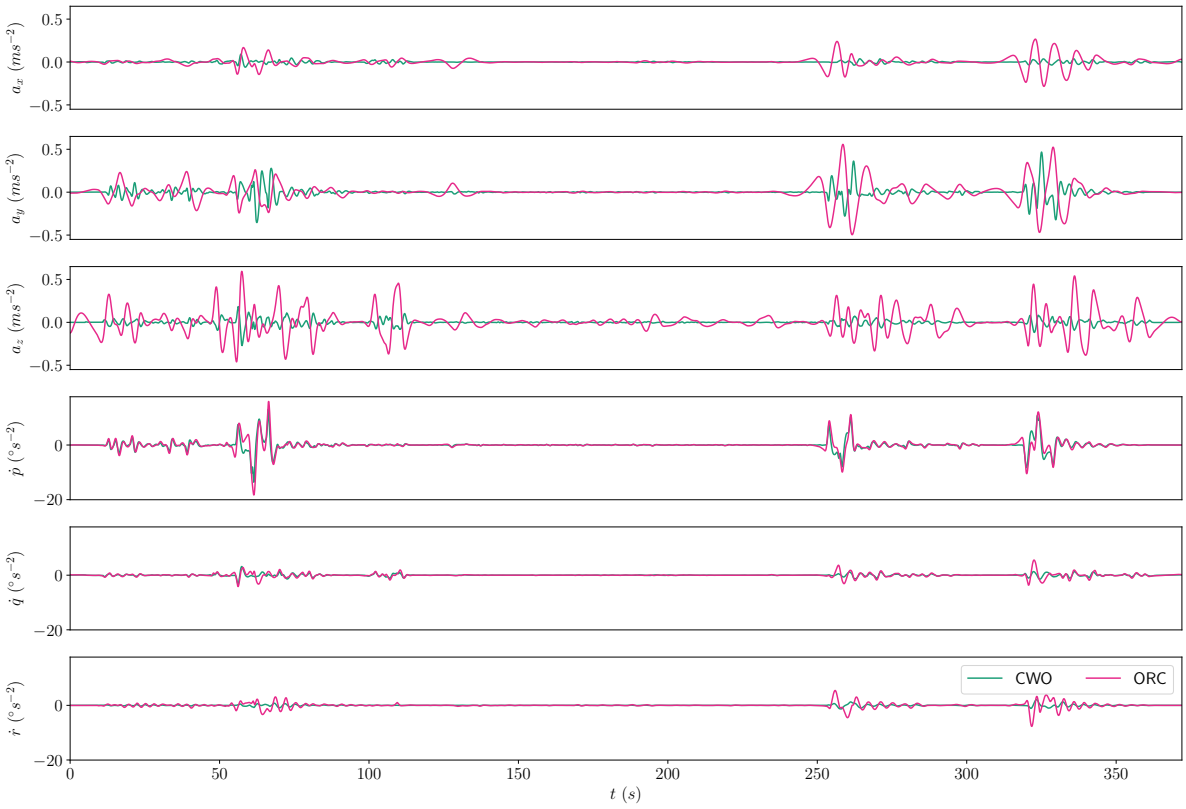


Figure D.18: Simulator control input for the Upsets scenario used in the SRS experiment.

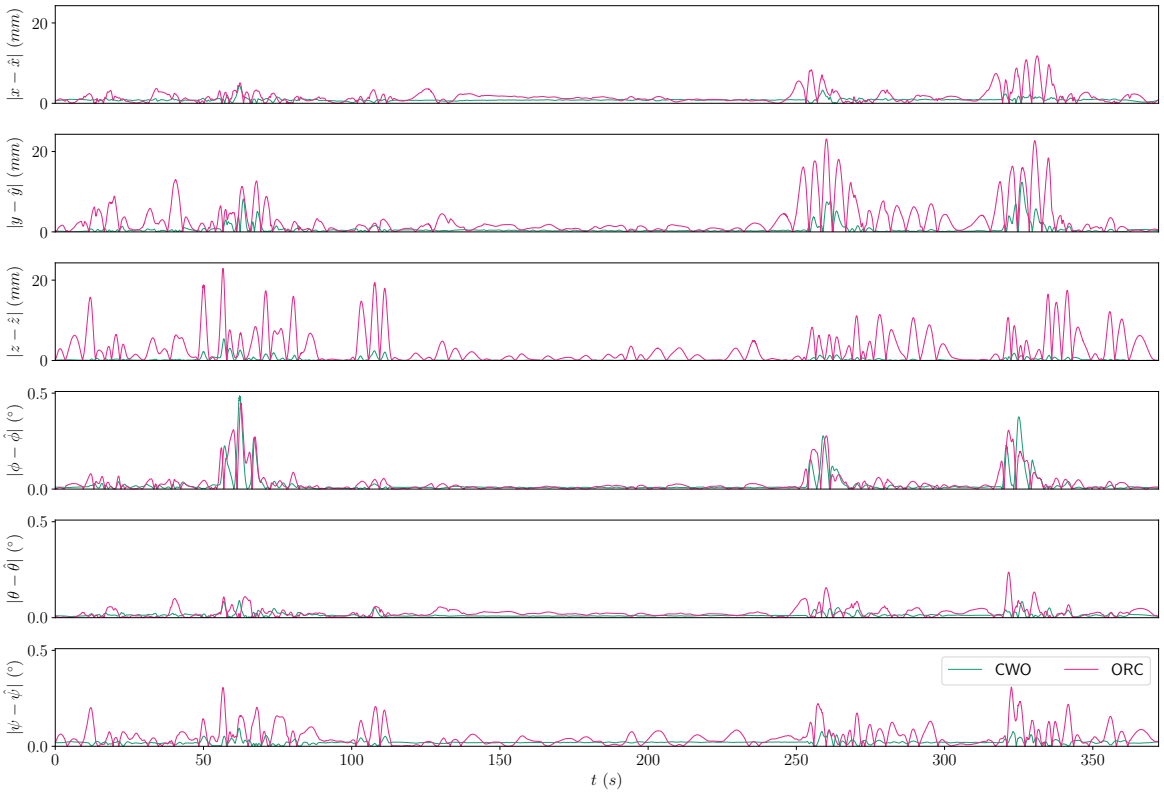


Figure D.19: Actual SRS state and calculated/commanded state difference for the Upsets scenario.

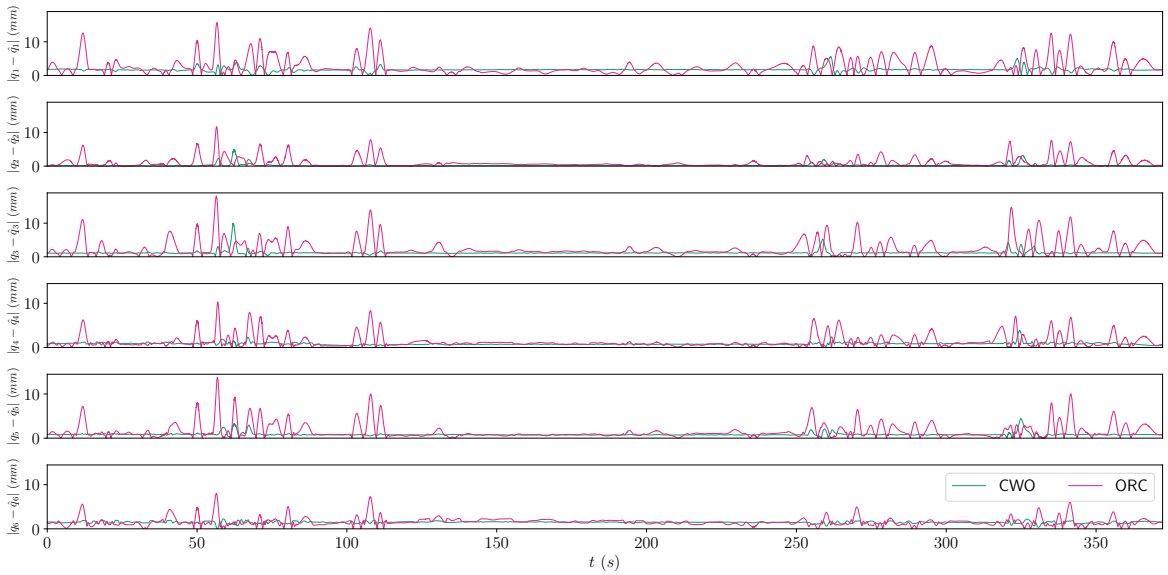


Figure D.20: Actual SRS state and calculated/commanded state difference for the Upsets scenario.

Appendix E

Experiment MFR results

In Table E.1, the MFR [6] results for all ten participants are shown. Note that the Large Roll and Upsets scenario were slightly changed after the first two participants.

Table E.1: MFR scores obtained during the SRS experiment. Note that the * indicates MFR scores that were not used in the statistical analysis, because these scores were obtained for slightly different versions of the respective scenarios.

	Training			**Symmetric Stall**						**Large Roll**						**Upsets**					
Part.	**CWO**		**ORC**	**CWO**			**ORC**			**CWO**			**ORC**			**CWO**			**ORC**		
P1	5	-	6	4	6	3	3	3	3	5*	-*	-*	-*	5*	-*	5*	5*	5*	4*	4*	3*
P2	3	2	2	5	4	4	3	4	3	2*	2*	2*	2*	2*	2*	5*	2*	2*	3*	5*	3*
P3	5	-	4	5	4	3	6	5	5	5	5	3	3	3	4	6	7	6	5	5	7
P4	4	-	2	3	3	4	3	3	3	6	5	5	4	2	2	5	5	5	4	7	4
P5	5	-	4	4	4	3	3	3	3	7	6	6	4	4	4	4	4	4	5	3	3
P6	5	-	6	5	5	5	6	5	5	4	4	4	3	4	1	5	5	5	6	6	6
P7	3	-	3	2	2	1	1	2	1	5	5	5	3	3	3	3	5	4	3	3	3
P8	7	-	6	4	3	4	3	3	4	7	7	6	5	4	5	4	5	5	5	4	4
P9	4	-	3	3	2	3	2	3	3	8	7	6	6	4	6	4	7	5	3	6	6
P10	4	-	3	2	3	2	4	3	2	3	3	3	2	2	1	3	3	3	2	2	2

Appendix F

Experiment continuous Motion Incongruence Rating results

In this appendix, the continuous Motion Incongruence Rating (MIR) [13] results obtained during the SRS experiment are included. In section F.1, the average MIR results are shown and in section F.2 until section F.11, the MIR results per participant are shown.

F.1. Average continuous MIR

In Figure F.1-F.4, the average continuous MIR per participant and for all participants are shown for, respectively, the Training, Symmetric Stall, Large Roll, and Upsets scenarios.

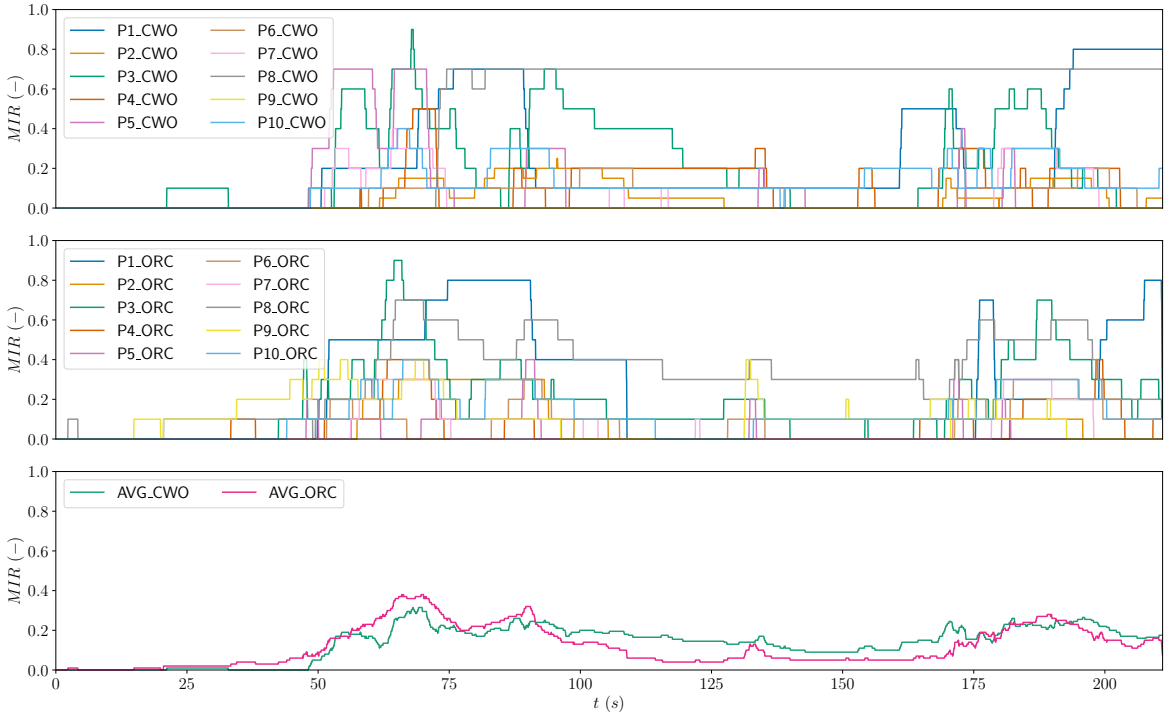


Figure F.1: Average continuous MIR for the Training scenario. The top graph shows the average continuous MIR per participant for the CWO condition, the middle graph shows the average continuous MIR per participant for the ORC condition, and the bottom graph shows for both the CWO and ORC conditions the average continuous MIR averaged over all participants.

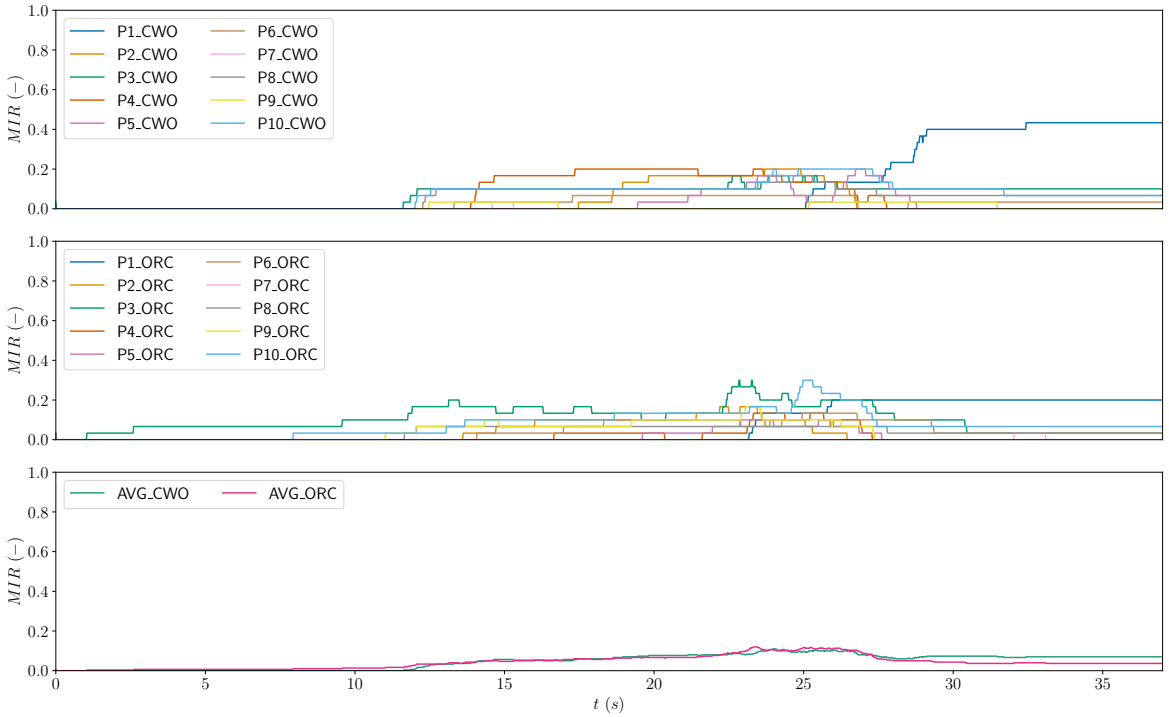


Figure F.2: Average continuous MIR for the Symmetric Stall scenario. The top graph shows the average continuous MIR per participant for the CWO condition, the middle graph shows the average continuous MIR per participant for the ORC condition, and the bottom graph shows for both the CWO and ORC conditions the average continuous MIR averaged over all participants.

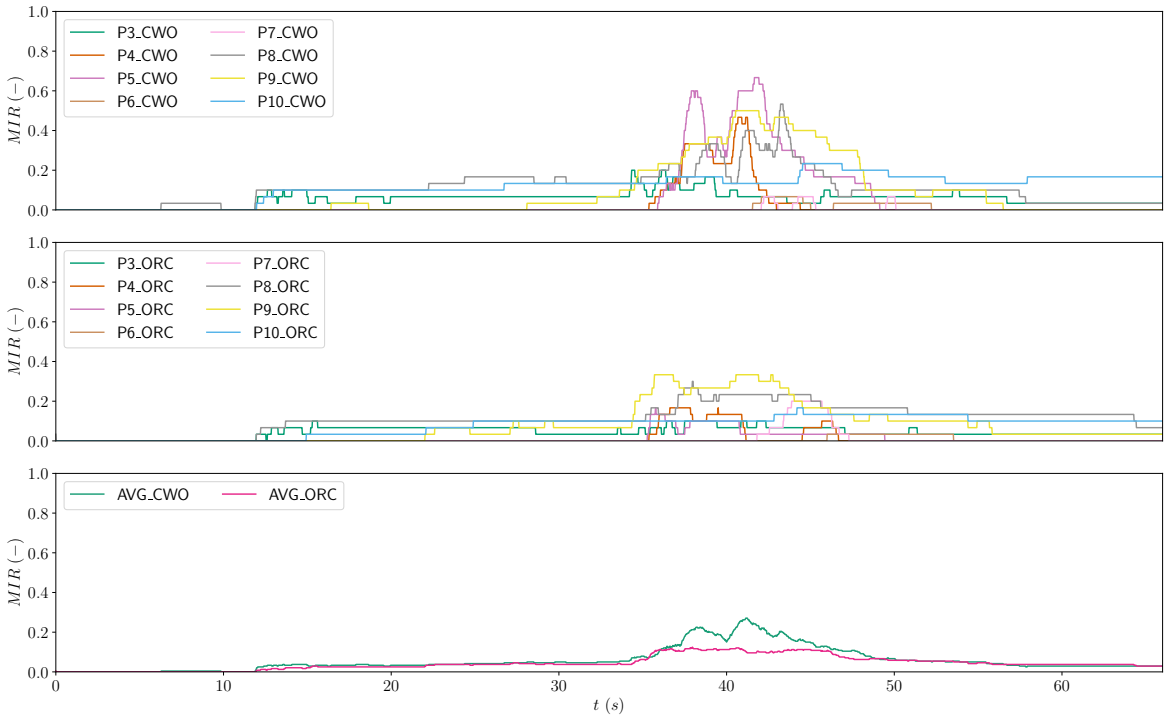


Figure F.3: Average continuous MIR for the Large Roll scenario. The top graph shows the average continuous MIR per participant for the CWO condition, the middle graph shows the average continuous MIR per participant for the ORC condition, and the bottom graph shows for both the CWO and ORC conditions the average continuous MIR averaged over all participants.

Figure F.4: Average continuous MIR for the Upsets scenario. The top graph shows the average continuous MIR per participant for the CWO condition, the middle graph shows the average continuous MIR per participant for the ORC condition, and the bottom graph shows for both the CWO and ORC conditions the average continuous MIR averaged over all participants.

F.2. Continuous MIR results for participant 1

In Figure F.5 and Figure F.6, the continuous MIR results for participant 1 are shown for, respectively, the Training scenario and the Symmetric Stall scenario.

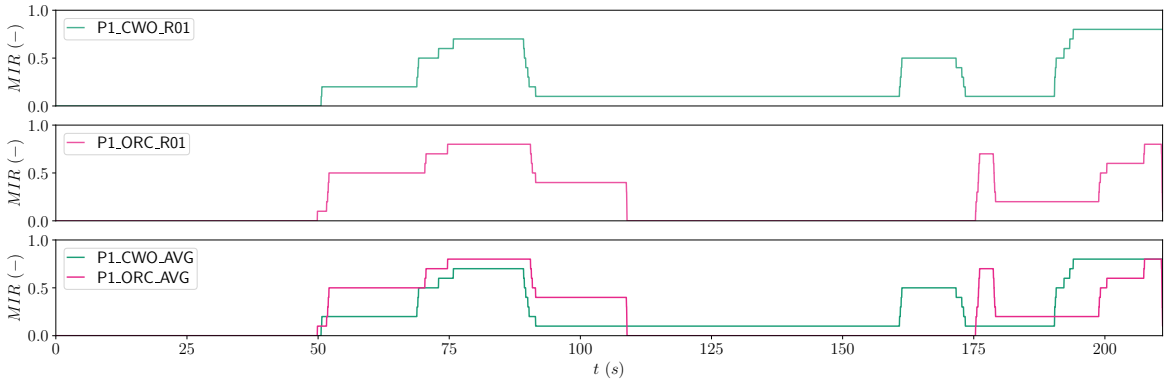


Figure F.5: Continuous MIR for participant P1 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

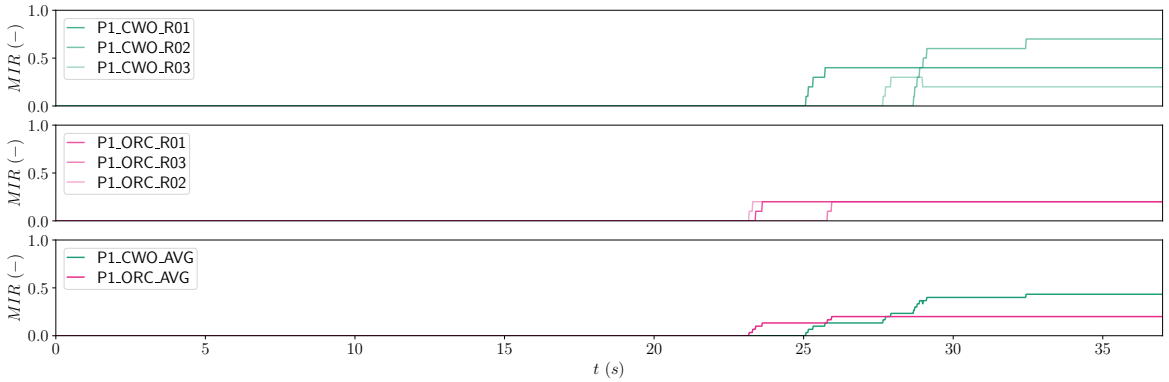


Figure F.6: Continuous MIR for participant P1 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.3. Continuous MIR results for participant 2

In Figure F.7 and Figure F.8, the continuous MIR results for participant 2 are shown for, respectively, the Training scenario and the Symmetric Stall scenario.

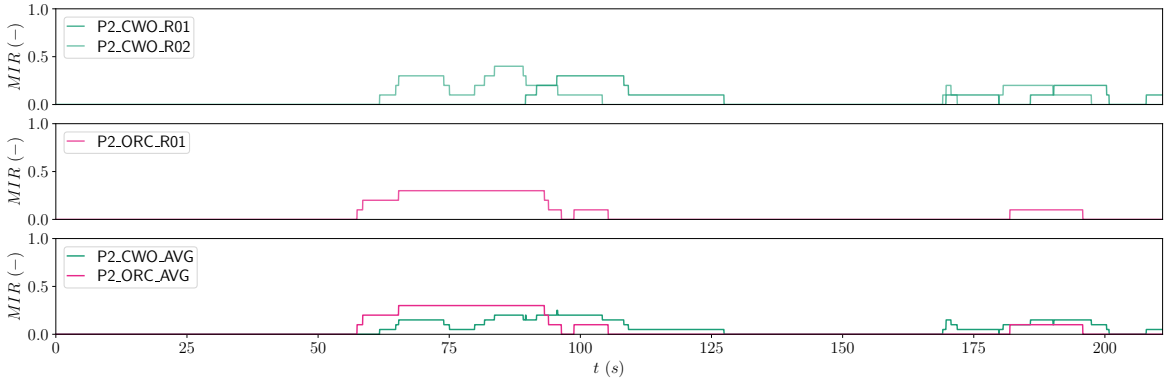


Figure F.7: Continuous MIR for participant P2 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

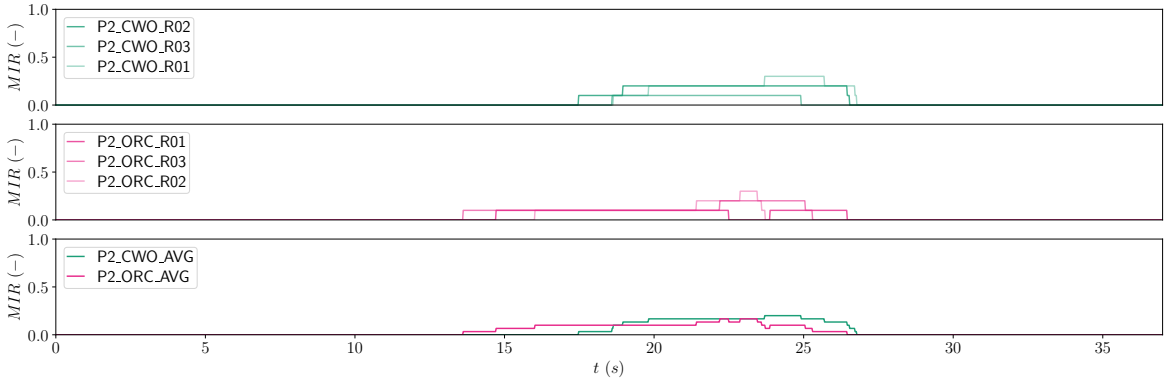


Figure F.8: Continuous MIR for participant P2 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.4. Continuous MIR results for participant 3

In Figure F.9-F.12, the continuous MIR results for participant 3 are shown.

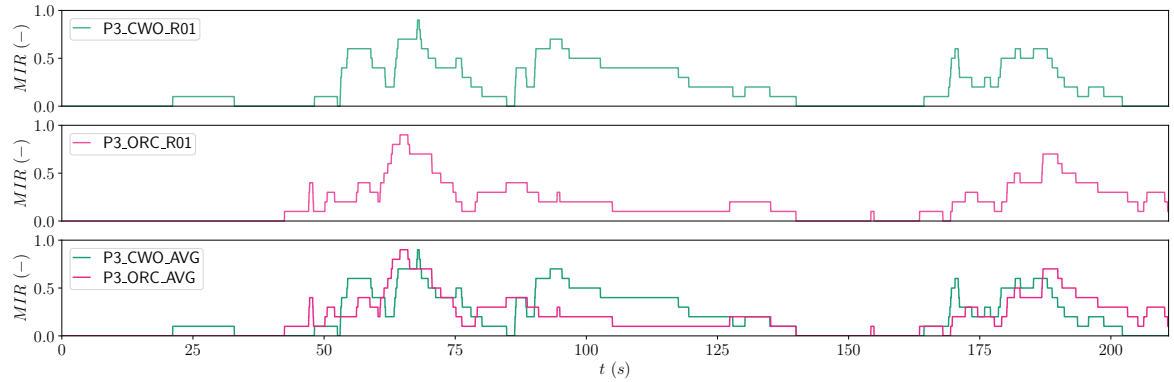


Figure F.9: Continuous MIR for participant P3 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

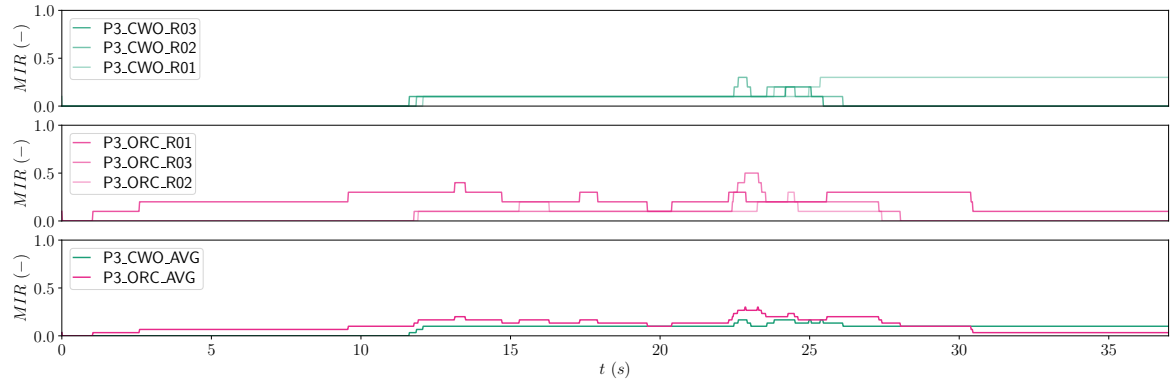


Figure F.10: Continuous MIR for participant P3 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

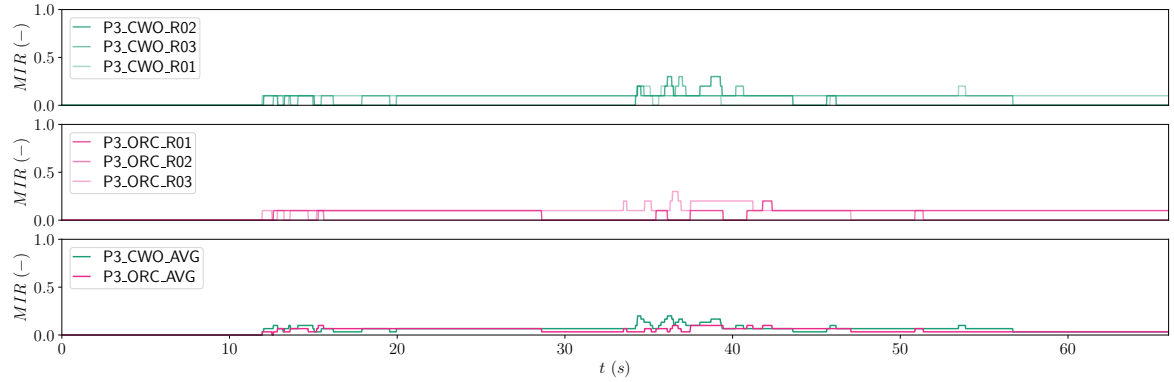


Figure F.11: Continuous MIR for participant P3 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

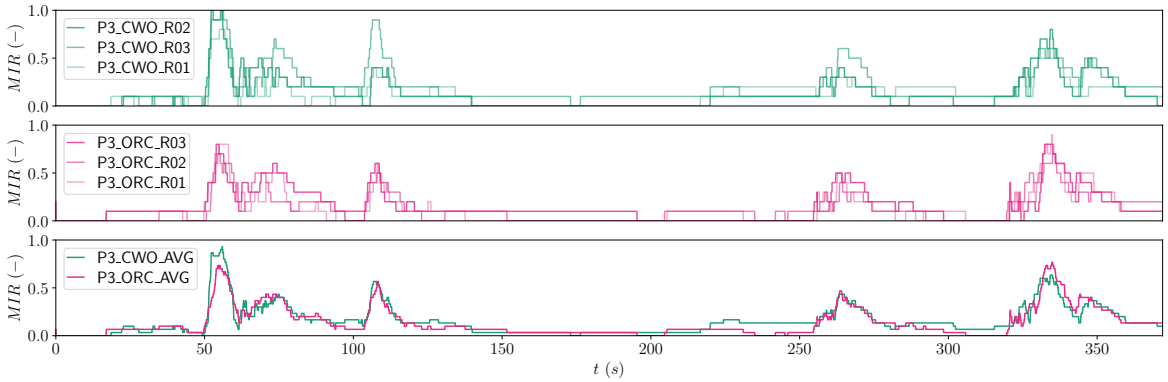


Figure F.12: Continuous MIR for participant P3 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.5. Continuous MIR results for participant 4

In Figure F.13-F.16, the continuous MIR results for participant 4 are shown.

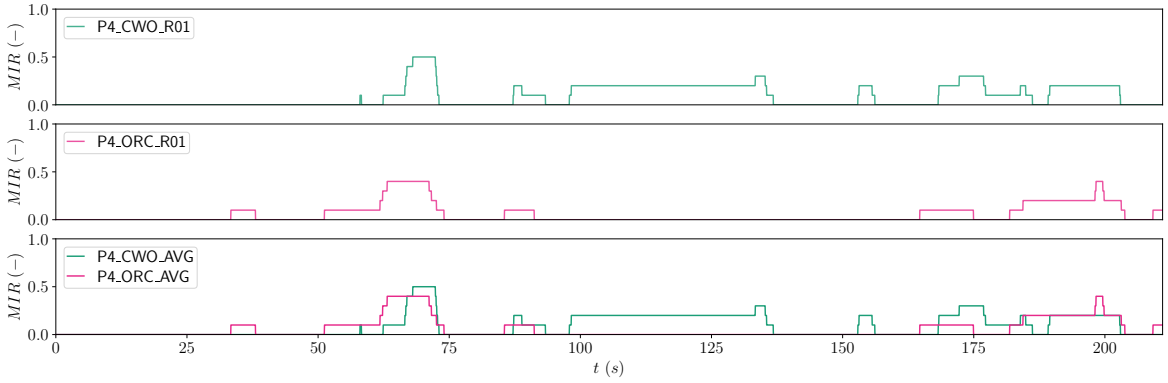


Figure F.13: Continuous MIR for participant P4 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

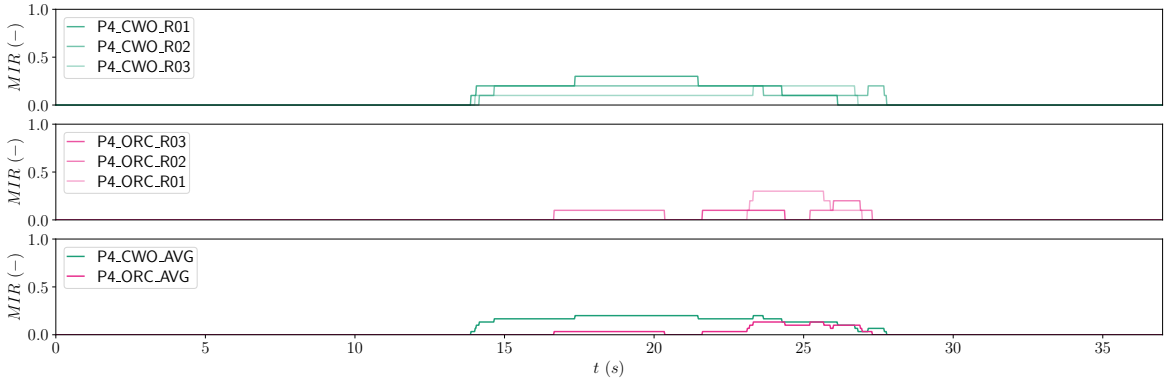


Figure F.14: Continuous MIR for participant P4 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

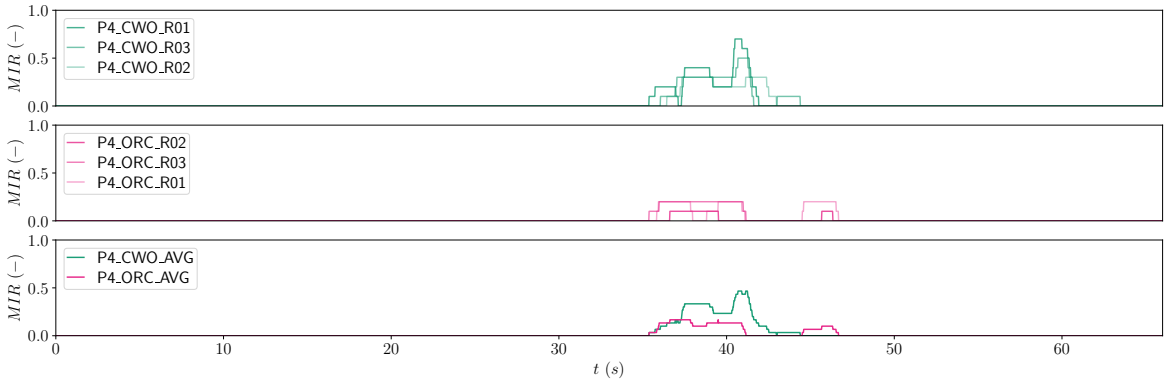


Figure F.15: Continuous MIR for participant P4 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

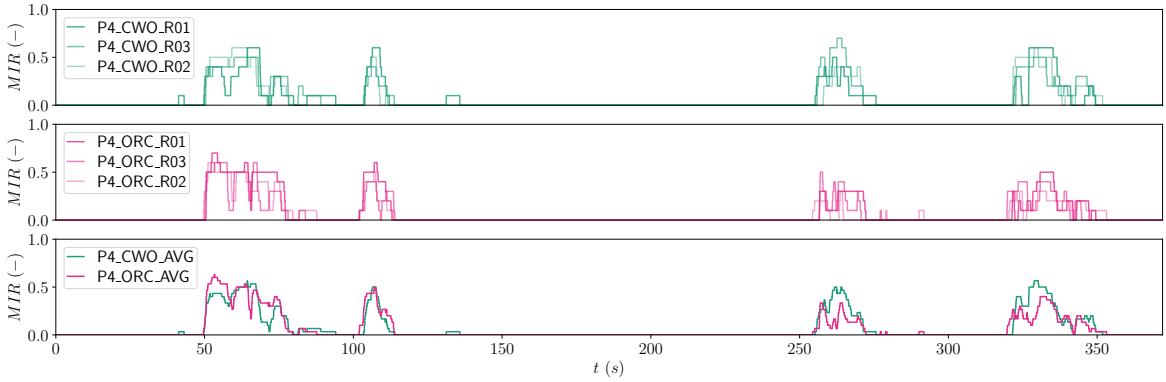


Figure F.16: Continuous MIR for participant P4 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.6. Continuous MIR results for participant 5

In Figure F.17-F.20, the continuous MIR results for participant 5 are shown.

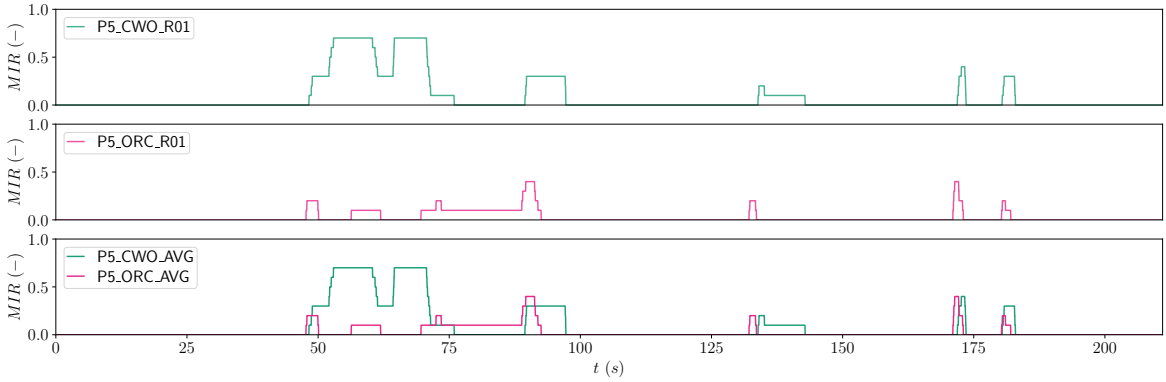


Figure F.17: Continuous MIR for participant P5 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

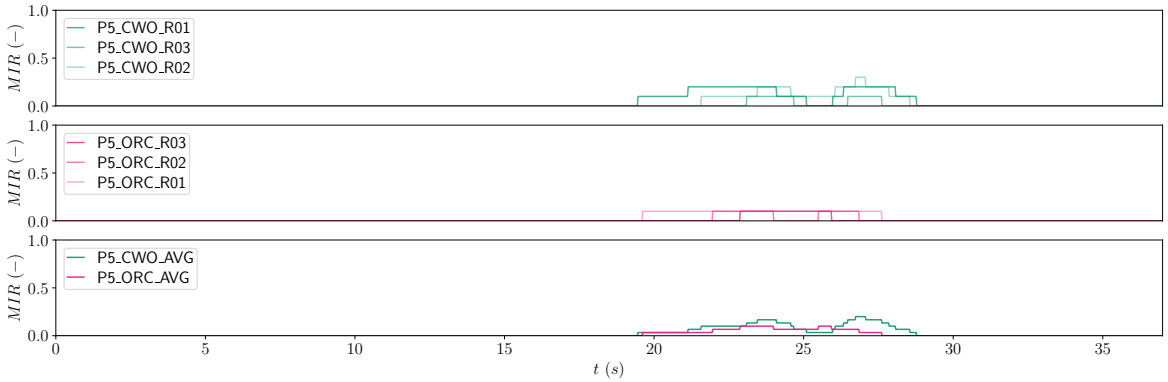


Figure F.18: Continuous MIR for participant P5 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

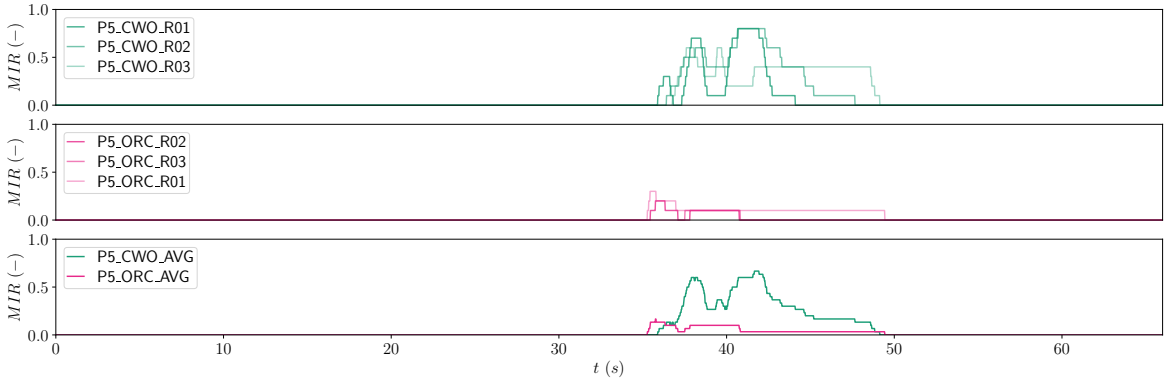


Figure F.19: Continuous MIR for participant P5 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

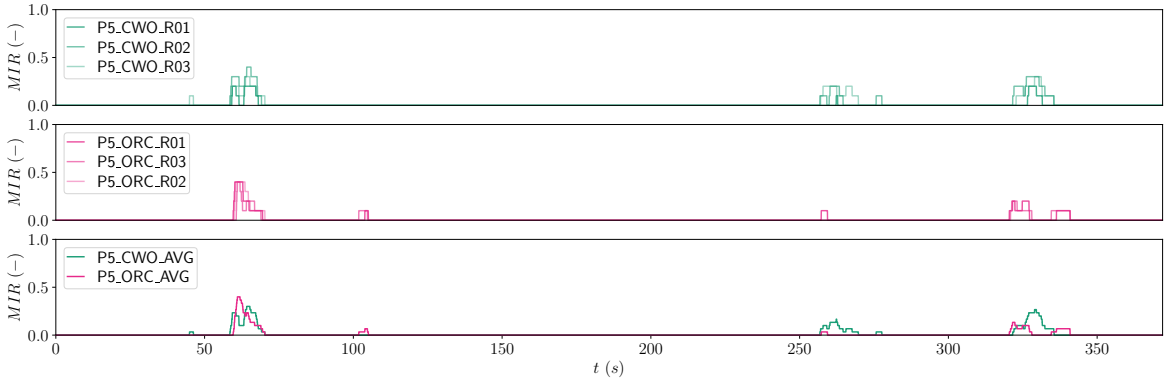


Figure F.20: Continuous MIR for participant P5 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.7. Continuous MIR results for participant 6

In Figure F.21-F.24, the continuous MIR results for participant 6 are shown.

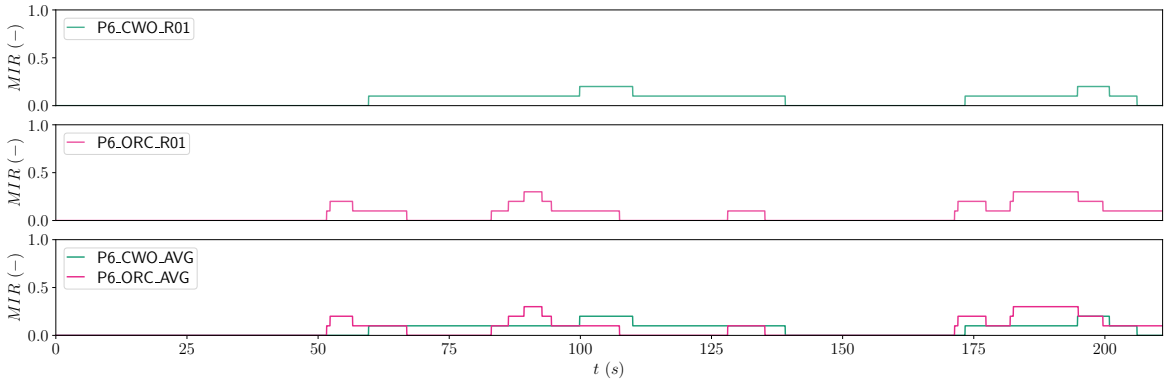


Figure F.21: Continuous MIR for participant P6 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

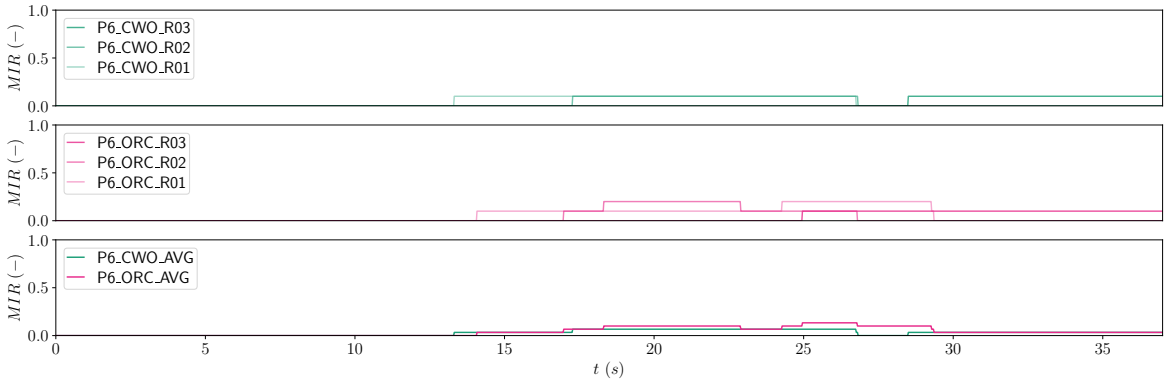


Figure F.22: Continuous MIR for participant P6 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

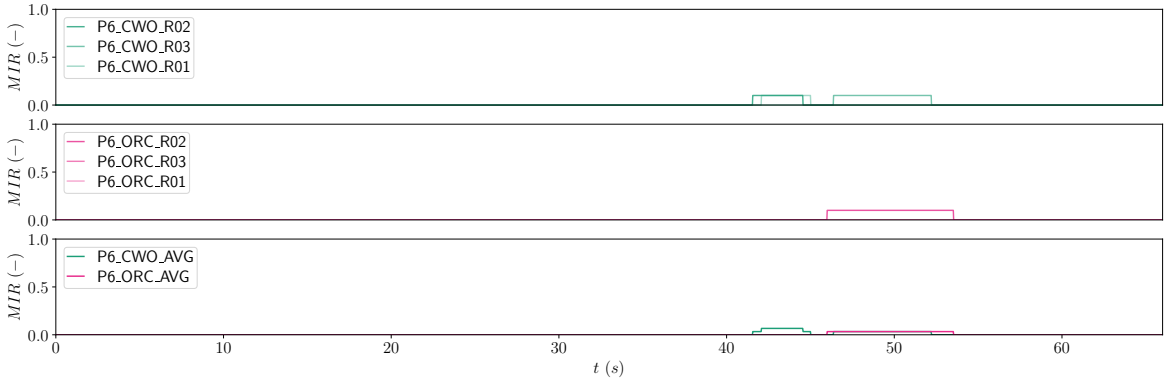


Figure F.23: Continuous MIR for participant P6 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

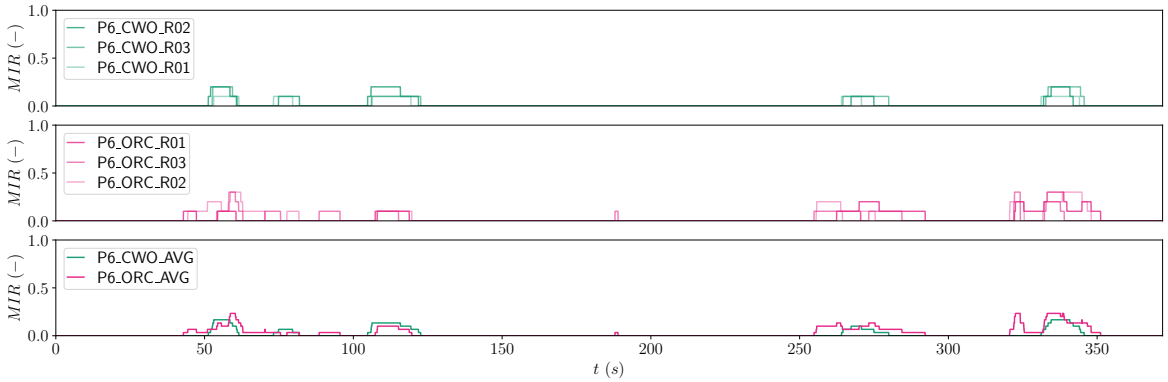


Figure F.24: Continuous MIR for participant P6 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.8. Continuous MIR results for participant 7

In Figure F.25-F.28, the continuous MIR results for participant 7 are shown.

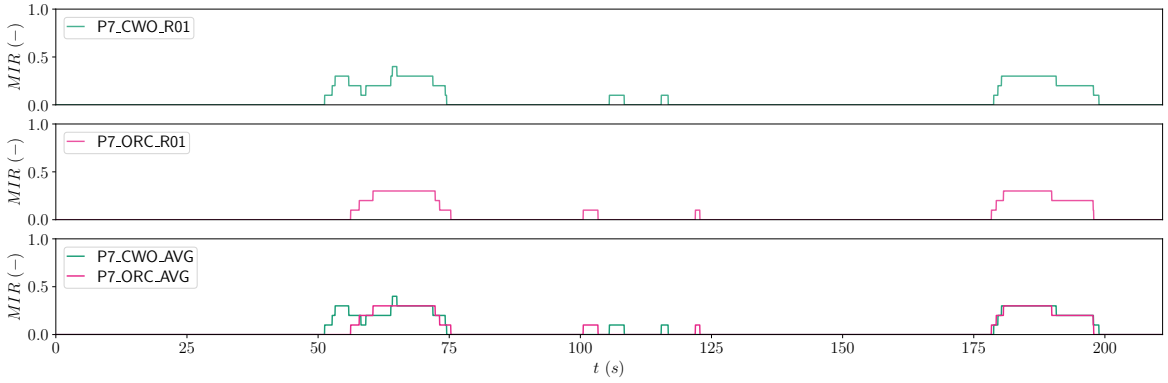


Figure F.25: Continuous MIR for participant P7 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

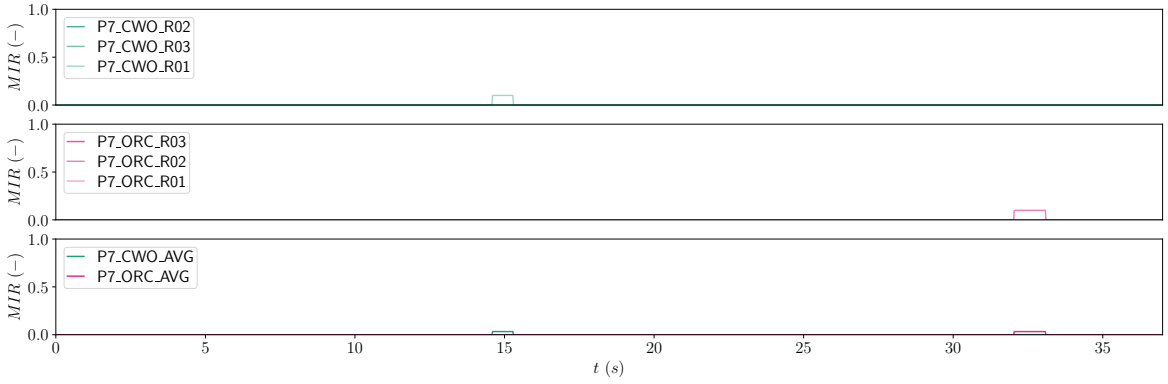


Figure F.26: Continuous MIR for participant P7 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

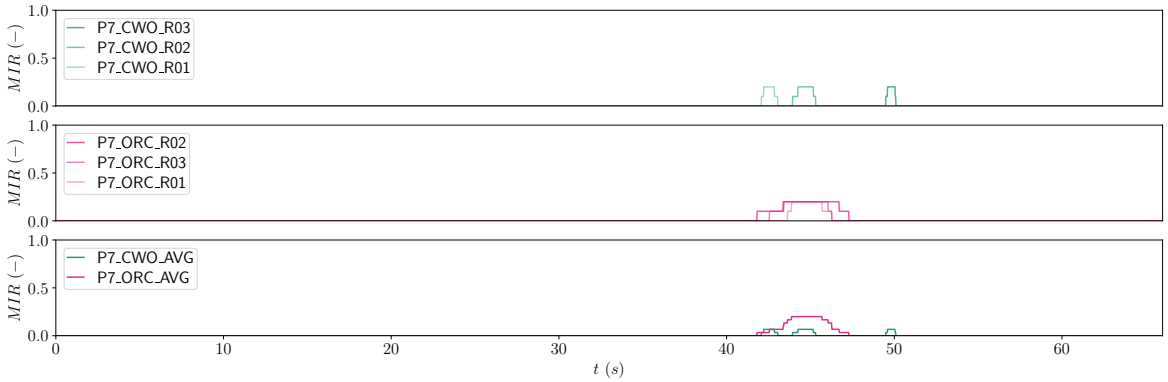


Figure F.27: Continuous MIR for participant P7 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

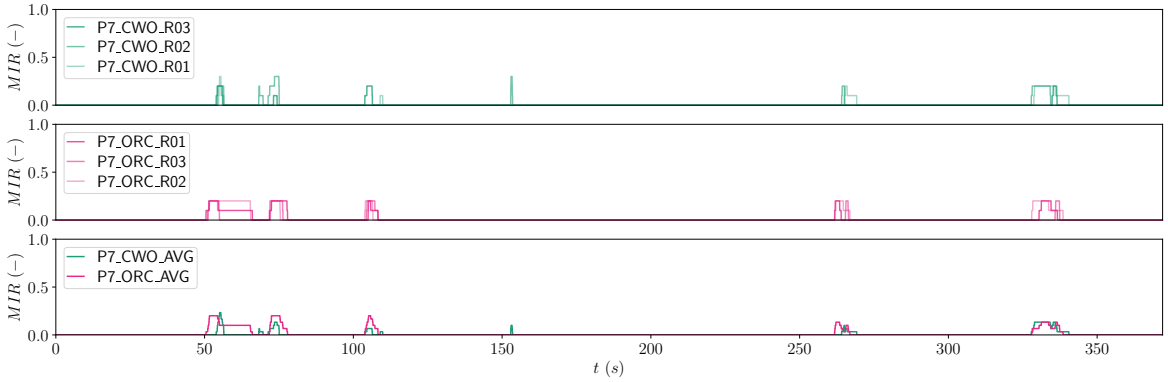


Figure F.28: Continuous MIR for participant P7 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.9. Continuous MIR results for participant 8

In Figure F.29-F.32, the continuous MIR results for participant 8 are shown.

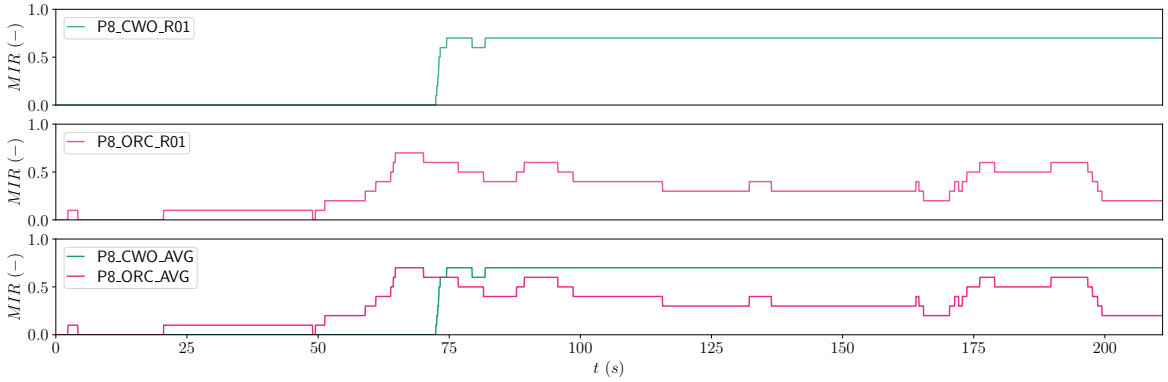


Figure F.29: Continuous MIR for participant P8 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

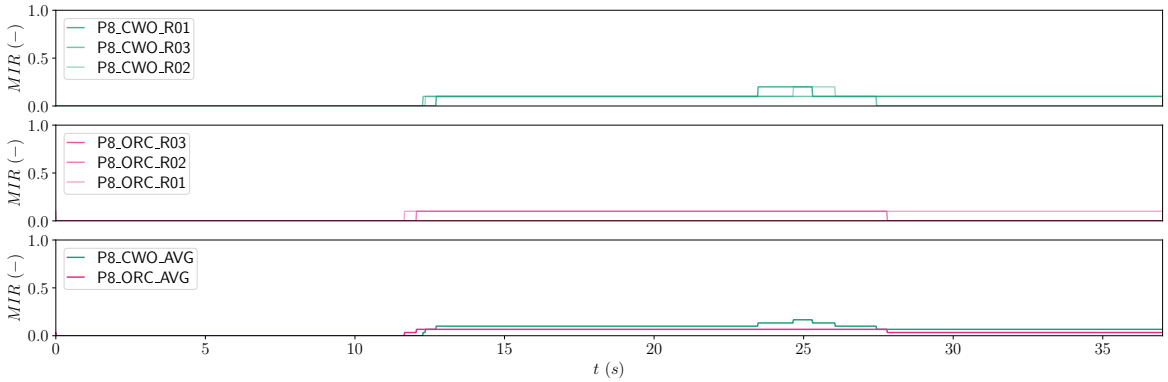


Figure F.30: Continuous MIR for participant P8 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

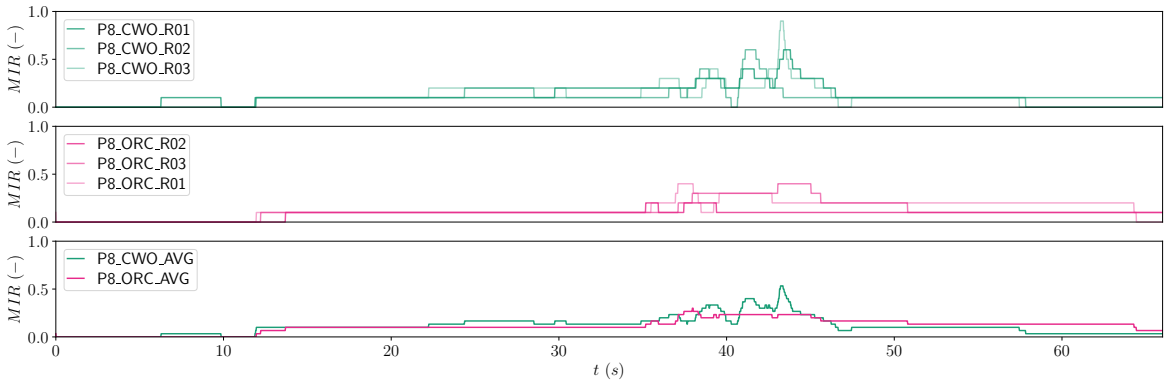


Figure F.31: Continuous MIR for participant P8 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

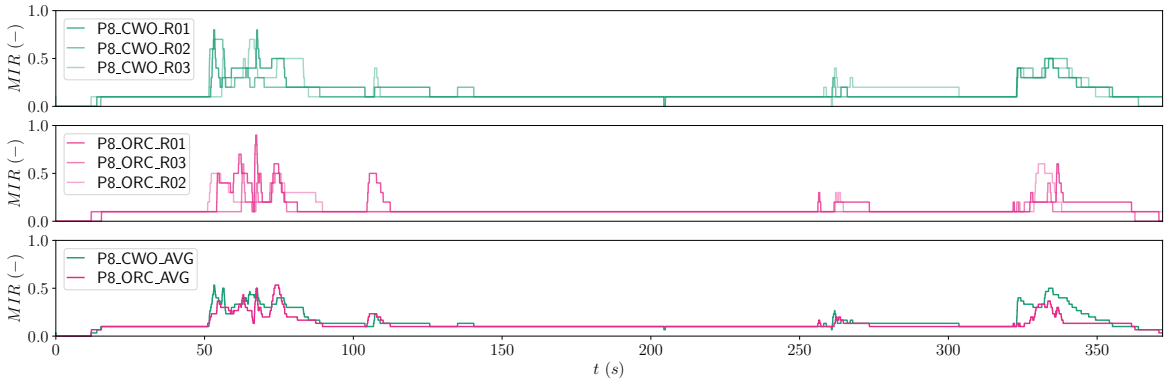


Figure F.32: Continuous MIR for participant P8 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.10. Continuous MIR results for participant 9

In Figure F.33-F.36, the continuous MIR results for participant 9 are shown.

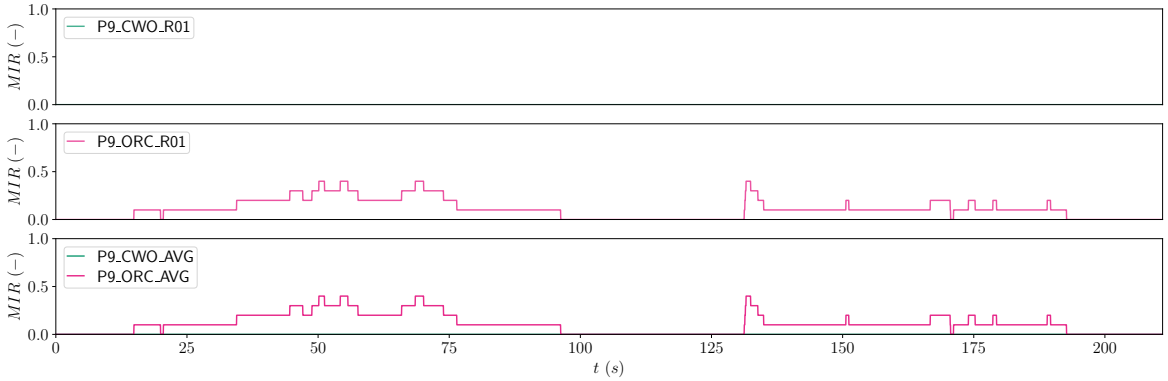


Figure F.33: Continuous MIR for participant P9 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

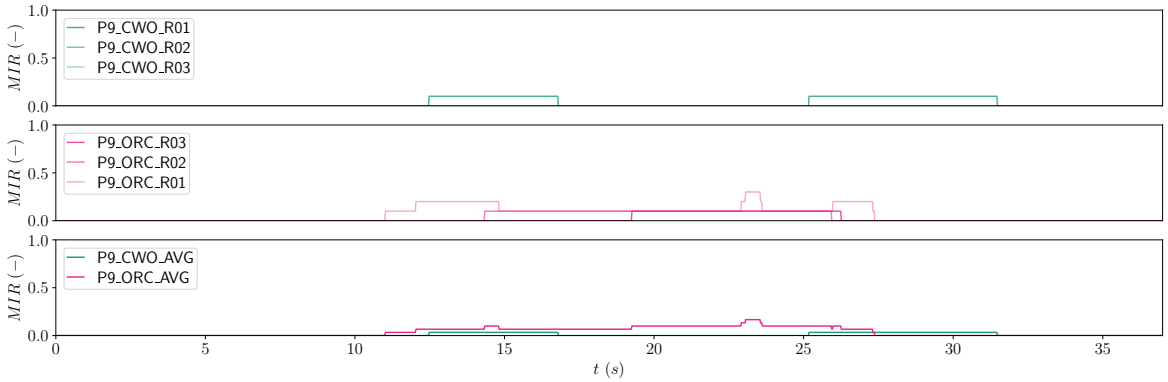


Figure F.34: Continuous MIR for participant P9 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

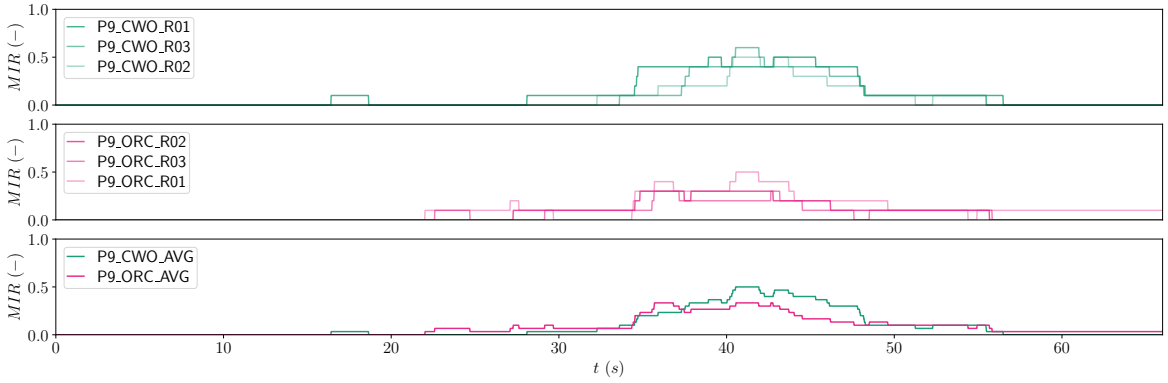


Figure F.35: Continuous MIR for participant P9 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

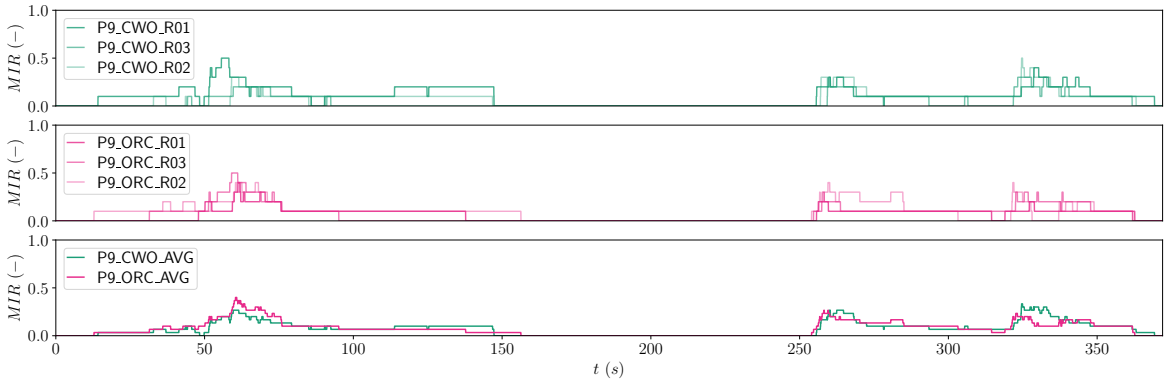


Figure F.36: Continuous MIR for participant P9 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

F.11. Continuous MIR results for participant 10

In Figure F.37-F.40, the continuous MIR results for participant 10 are shown.

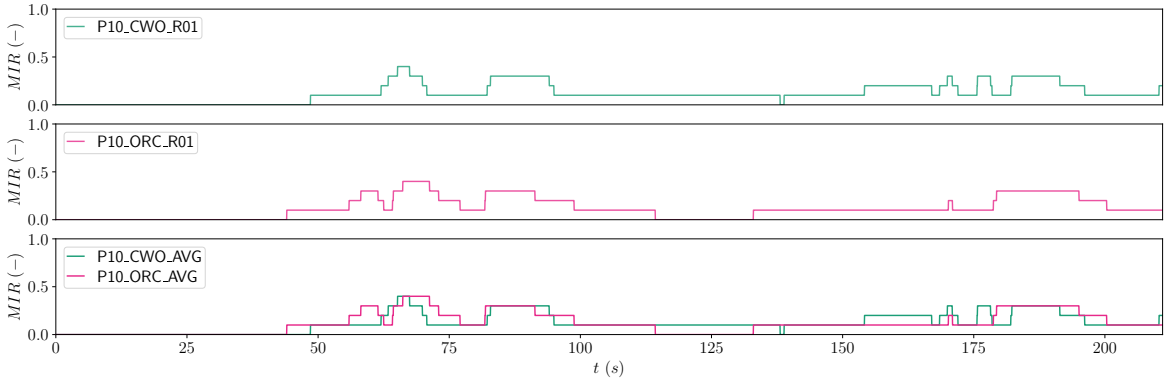


Figure F.37: Continuous MIR for participant P10 for the Training scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

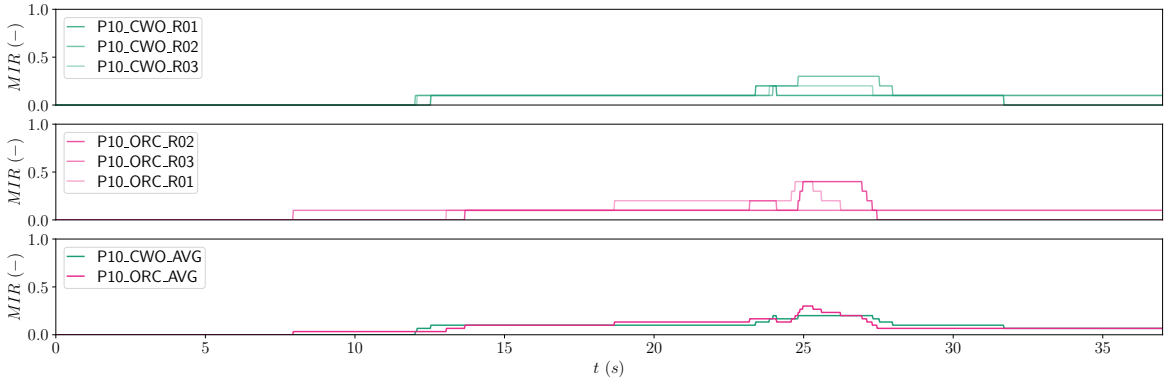


Figure F.38: Continuous MIR for participant P10 for the Symmetric Stall scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

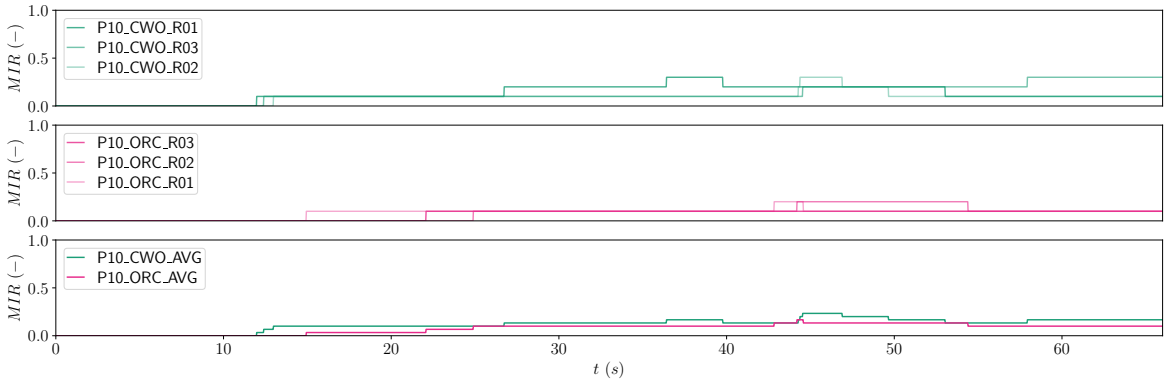


Figure F.39: Continuous MIR for participant P10 for the Large Roll scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

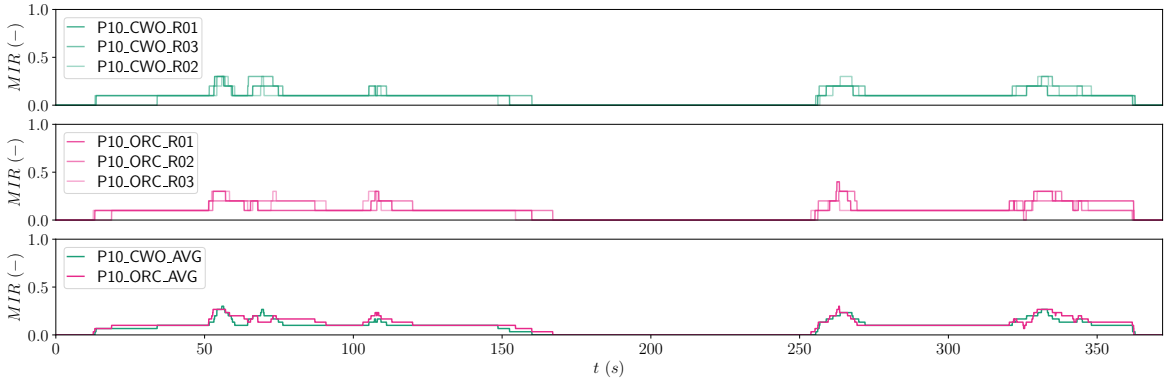


Figure F.40: Continuous MIR for participant P10 for the Upsets scenario. The top graph shows the MIR for the CWO condition, the middle graph for the ORC condition, and the bottom graph the average MIR for both the CWO and ORC conditions.

Appendix G

Experiment commentary and comment analysis results

In this appendix, the comments provided by the participant during the experiment in the SRS are presented. In Table G.1, the condition preferences derived from the comments are shown per participant. In Table G.2-G.11, the comments are presented for all ten participants.

Table G.1: Participant's condition preferences derived from comments provided during the SRS experiment runs. Grey cells indicate no preference between conditions.

Scenario	Keyword	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Symmetric Stall	Buffet		ORC		ORC	ORC			ORC	ORC	
	Pitch/surge	ORC	ORC	CWO	ORC	ORC	CWO				
	Heave	ORC		ORC						ORC	
Large Roll	Onset roll	-	-				CWO			CWO	
	Sustained roll	-	-		ORC	ORC	ORC	ORC	ORC	ORC	ORC
Upsets	Pitch	-	-	ORC		ORC					
	Heave	-	-								
	Onset roll	-	-	CWO	CWO		CWO	CWO	CWO	CWO	
	Sustained roll	-	-		ORC			ORC		ORC	ORC

Table G.2: Condition preferences of participant 1 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	3	Buffet realistich. Recovery meer sensatie nodig, mag heftiger.
		5	Late CR. Minder heftig dan run 4.
		8	Iets minder heftig dan run 7. 3.5 MFR rating.
	ORC	4	Duidelijker in recovery (vergeleken met run 3).
		6	Heftiger in voorover.
		7	Nan
Large Roll	CWO	10	Roll goed, te weinig pitch.
		-	
		-	
	ORC	9	DNF: oil level baseframe to high.
		11	DNF: oil level baseframe to high. Pitch duidelijk, MFR rating tussen 4/5.
		-	
Upsets	CWO	12	Roll goed, onset. Pitch bleef achter.
		15	Hele goede rol, bijna geen pitch. Rating had nog roder gekunt.
		16	Goede rol, pitch leek iets beter (dan run 15), maar op grote geheel niet aanwezig.
	ORC	13	Minder heftig op roll, daardoor betere pitch en balans. Beter beeld.
		14	Mooi in balans, meest in de buurt van totaal plaatje. Op het laatst minder pitch.
		17	Betere pitch onset. Roll redelijk, lag? Beste! Beste in balans, goede roll sensatie en pitch.

Table G.3: Condition preferences of participant 2 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	16	Voelde wat buffet, te weinig en duwtje in de rug.
		17	Iets betere buffet, nog steeds gek duwtje.
		19	Vorige leek beter (run 18), buffet minder duwtje in de rug merkbaarder.
	ORC	18	Nog steeds duwtje, beter dan vorige (run 17). Buffet verbetert.
		20	Beetje zelfde als vorige (run 19). Lichte buffet en duwtje in de rug.
		21	Iets meer buffet. Iets minder duwtje.
Large Roll	CWO	6	Zijwaarts goed, +/- g minder.
		7	Roll is goed. Weining g, daarom ook de rating.
		9	Roll doet het goed. G's voel je niet.
	ORC	4	Geen verschil in g's. Zijwaartse beweging voel je goed.
		5	Je voelde zelfs een beetje lichter, heel goed.
		8	Roll gewoon realistisch.
Upsets	CWO	11	Op g's na gewoon goed. Pitch traag. Roll goed. Voel inzetje, of misschien was dat vorige run 10.
		12	G aanzetjes, maar verder niet. Initiele inzet (roll) heel goed. Unloaden voel je niet.
		14	Behalve de missende g krachten geen gekke dingen.
	ORC	10	Voelde pitch slecht, niet heel realistisch door missende g's. Roll goed, pitch minder en traag. Wellicht voelbare inzet.
		13	Je voelt artificiele hobbeltjes. Je voelt hier duidelijk aanzetjes: oh nu naar links, eerst ff duwtje rechts. Misschien let ik er nu meer op, ff duwtje recht links. Eerste deel van scenario MRF 6.
		15	Rating op gekke bewegingen. Paar voelbare inzetjes: hij gaat komen. Recovery smooth, maar geen g's. Eerste deel MFR 4.

Table G.4: Condition preferences of participant 3 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	10	Iets beter, meer pitch!
		13	Pitch beter ten opzichte van heave.
		14	Pitch en heave voel ik allebei.
	ORC	9	Voelde hem naar beneden vallen. Meer naar beneden meer heave dan pitch. Mis de pitch.
		11	Pitch underrated
		12	Tandje beter, te weinig pitch
Large Roll	CWO	15	Onset te heftig. Side force groter dan in het echt, maar useful.
		18	Duidelijk minder. Meer hangen dan draaien.
		19	Goede mix van hangen en draaien.
	ORC	16	Voelen naar rechts en naar links. Dus mindere sideforce, misschien te weinig.
		17	Vond hem zelfde als vorige. Mag iets meer sideforce in.
		20	Iets meer roll cues. Allemaal de goede kant op. Minder sideforce, iets slapper.
Upsets	CWO	3	Roll beter dan pitch. Helemaal begin voel ik delay. Oppitchen voel je niet, omvallen realistisch. Mis yaw. Voelt niet heftig in de sim.
		5	Idee dat de pitch veel minder aanwezig was. Geen verschil in loadfactor. Iets minder dan de vorige (run 4), mis pitch.
		8	Teken van de roll goed. Niet counter intiutive. Te weinig Fx in cruise. Inzet bocht prima. Roll goed, optrekken niet. Roll erg goed (MFR3/4). Pitch mis ik, voel niet eens pos of negatief.
	ORC	4	Omvallen oke, geeft heftigere sensatie. 0.7g kan ook beter. Roll beweging strakker en heftiger dan run 3.
		6	Onset gedeelte beter. Sustained niet zo heel goed. Voel geen verschil in g's en lichte g verschillen. Eerst MFR 4 daarna 5.
		7	Iets rustiger dan vorige. Gevoel van achteroverhangen is aanwezig. Idee van rollen naar links, terwijl op beeld ik de andere kant op rol. Laatste twee bochten, is vervelend dus daarom hoge MFR.

Table G.5: Condition preferences of participant 4 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	5	Hetzelfde, minder dan vorige 2. Buffet minder. Op het randje, zou 4 kunnen zijn.
		6	Buffet hetzelfde: MFR 2. Mistte unloaden, attitude indicator lagged.
		8	Buffet kan overtuigender. Heel vroeg neus laten zakken, voorover in stoel.
	ORC	3	Buffet vrij realistisch. Pitch up ook. Stall break mis je.Lichter worden ook. 0.4g voel je niet.
		4	Goed, buffet beter, stevige pitch up. Neus zakken, weinig g verschil. Mist unloaden.
		7	Geen lag. Buffet goed: MFR 1. Unloading mis je. Pitch en achterover beter.
Large Roll	CWO	10	Erger dan de vorige. Sideforces to much. Pitch prima (te weinig g).
		11	Zelfde verhaal. Minder dan vorige.
		13	Stuk minder dan vorige. Uitrollen rechter bocht extreem slecht. Sideforce tegen roll richting in vervelend en veel onrealistischer.
	ORC	9	Voelde overall goed. Sideforce wat veel, voelt overdreven. Mis g.
		12	Stuk beter. Roll voelde veel realistischer, niet zo uit je stoel geslingerd.
		14	Voelde weer heel goed. Iets meer side force dan de beste. Ontbreken van de pitch, en g's.
Upsets	CWO	15	G krachten zijn er niet. Overdreven sideforces zijn meer storend. Eerste roll mis je g, sideforce veel te sterk! Ontbreken pitch. 3 pitch, 5 roll.
		16	Ongeveer hetzelfde als vorige. Verticaal afwezig. Pitch 3 roll 5. Tweede deel zelfde. Mooi richting vd bocht, upset.
		18	Onset goede kant, sideforces daarvan veel te hoog. Lateraal 6. Pitch steady iets beter. Tweede deel weer herhaling, inzet goed, daarna scheef in stoel. Ontbreken motion minder storend
	ORC	17	Opzich goed, even gevoel verkeerde teken. Op moment van inzet voel je beweging verkeerde kant. Pitch onset mis je naast g. 3 eerste deel. Beide upset ook roll verkeerde kant, als een fietser. Geen g's en pitch lijkt zelfs verkeerde kant. Onset verkeerde kant.
		19	Inzet roll voelde goed. Behoorlijk links en rechts. Richting 6. Te weinig g. Bijna 7 voor lateraal. Tweede deel wel bocht verkeerde kant inzet.
		20	Ook hier sideforce excessief: 6. Mis pitch cues en g. Ervaar het als hetzelfde. Eerst verkeerde kant op. Geen prettige cues. Pitch weinig verschil en ook onset.

Table G.6: Condition preferences of participant 5 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	15	Iets heftiger dan gewend, buffet. Longitudinaal voelde het wat minder aan.
		18	Ietsje minder fijn, door de pitch, long versnelling.
		19	Ietsje fijner. Buffet oke. Pitch down voelde natuurlijker.
	ORC	16	Longitudinaal versnelling beter. Buffet iets te heftig, vooral aan het einde.
		17	Buffet en recovery gewoon en natuurlijk aanvoelen.
		20	Buffet heftiger, pitch gewoon natuurlijk.
Large Roll	CWO	3	Hele heftige bewegingen, voel dingen die je niet in het vliegtuig voelt. Zijdelingse krachten bijv.
		5	Weer hele zware zijdelingse krachten, minder heftig dan de eerste run. Twijfel 6 of 7.
		8	Veel heftigere zijdelingse bewegingen.
	ORC	4	Weinig last van zijdelingse bewegingen. Dus minder motion voor gevoel maar ook minder storend. Voelde inrollen maar niet zijdelingse krachten.
		6	Veel beter: minder heftig en minder zijdelingse krachten.
		7	Vergelijkbaar met net, relatief weinig verschil.
Upsets	CWO	10	C1: Voelde wat natuurlijker aan. Voelt nog wel echt verschil. Long voelt oke, lat blijft lastig. Sideforces. Pitch ervaar ik een klein beetje, beperkt. G krachten mis je.
		13	C1: iets heftiger, iets meer afleiding. Iets lagere rating dan net. Verticale g is moeilijk. Iets minder: lateraal vooral. Pitch blijft lastig: merk ik niet zo.
		14	Beetje zoals net. Ik merk in z niet heel veel verschil. Met name dwarskrachten.
	ORC	9	C1: Harde inrollen viel mee. G krachten voel je een klein beetje, niet storend. C3 late rating. Beetje veel zijdelingse krachten deel 2.
		11	C1: Pitch voel ik nu wel. Iets minder dwarskrachten. Iets betere verticale g voel je aan het begin eventjes. C2 en C3 geen CR: kwam natuurlijk over.
		12	Redelijk natuurlijk. Lijk alsof je voorover hangt. Redelijk vergelijkbaar met run hiervoor. C2/3: Natuurlijk, minder heftige bewegingen.

Table G.7: Condition preferences of participant 6 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	10	Ietsje beter. Nog steeds slap. Gevoel van ietsje meer naar voren.
		11	Vergelijkbaar. Naar voren gevoel oke, niet overtuigend.
		13	Niet veel verschil tussen deze en vorige.
	ORC	9	Buffet kan zwaarder, stall break mag heftiger. Ook pitch en g. Ik verwacht een forward motion.
		12	Niet zo veel verschil. Marginaal verschil in g.
		14	Niet overtuigend. Idee van verschil. In laatste stuk meer pitch moment.
Large Roll	CWO	15	Heel aardig. Allerlaatste stuk iets te heftig. Te veel sideforce.
		16	Vergelijkbaar, laatste stuk sideforce he dit klopt niet.
		18	Weer laatste stukje te heftig qua sideforce.
	ORC	17	Mooi smooth. Daarom geen rating. Mis alsnog de g.
		19	Niet te heftig, maar idee na ijlen. Moeten nu steady zitten maar nee wobbelde nog wat na. Voel andere kant op maar zie ik ook.
		20	Was mooi, goed. Geen nazeef effect.
Upsets	CWO	5	Veel beter. Geen rare cue, alleen pitch te lafjes. Pitch te lafjes, moment en g.
		6	Vergelijkbaar met vorige, goede kant op. Niet zo veel verschil.
		8	Beter, alleen pitch nog zelfde als vorige. Geen verkeerde cues.
	ORC	3	Pitch te lafjes, reactie snelheid te laag. Eerste deel 4. 2e deel minder: roll cue verkeerde kant op voel je.
		4	Inzetten vd bocht rare cue. Verkeerde kant op. Eerst verkeerde kant.
		7	False cue, roll voel je verkeerde kant, daarna sideforce. Pitch ook weinig. 2e deel rare cue voelde ik weer.

Table G.8: Condition preferences of participant 7 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	3	Buffet iets strakker in het echt. Dip voelt natuurlijk. Vlak voor recovery dip versnelling in nose down, hoofd naar achter.
		5	Geen CR. Je voelt iets geks aan het einde bij nose down.
		8	Geen CR. Versnellen aan het einde gevoel.
	ORC	4	Geen CR. Heel erg zoeken naar verschillen. Voelde ook kleine versnelling, maar minder.
		6	Late CR. Versnelling minder. Gehele run te veel in me stoel, daarom 2.
		7	Geen CR. Zulke minieme verschillen. Niet het gevoel iets anders. Wel minder in stoel gedrukt: weet niet of dit echt zo is.
Large Roll	CWO	10	Dit was raar. Gevoel van te lang doordraaien terwijl we de andere kant op gingen. Hangen, kracht of rotatie, moeilijk te zeggen.
		13	Nee! Hier klopt helemaal niks van. Zo raar gevoel. Roll perf off. Traagheid, vertraging.
		14	Deze ook niet. Gevoel van voorlopen krachten op roll. Veel te veel in stoel gegooid.
	ORC	9	Lekker. Inrollen ging snel. Insturen in motion en visuals, dus prima. Terugrollen te snel. Roll out geen g. Transitie te smooth.
		11	Voelde beduidend beter. Kon weer op g's letten. Roll goed, 3 door g's.
		12	Gevoel tussen de een en ander. Roll performance tikkie raarder.
Upsets	CWO	15	Gebeurde veel. Idee dat de roll niet klopte, bank angle en staat/krachten klopte niet. Mis 2.2g. Eerste deel 4. Inzetten overbank mis ik 2g. Idee tweede deel roll beter. Krachten zijkant komt overeen.
		18	Roll was off. Daarvoor CR. Bank angle klopte niet op wat ik voel. Gevoel loopt achter. Gevoel van roll klopte niet, traagheid.
		19	C2 geen CR. Zo moeilijk gevoel dat het klopt, maar iets ook niet. Ik weet het niet. Teveel roll, heel lichtjes. Te veel naar links voorloopt op visual.
	ORC	16	Motion lekker behalve g. Zuiverder roll, klopte beter. Lekkerder dan vorige. Roll beter g's weinig verschil.
		17	Ook lekker. Ietsje meer heen en weer, maar heel miniem. Wel lekker voelen. Iets meer pitch zou kunnnen. Roll realistisch. CR op G's.
		20	Wel weer hetzelfde als vorige, ietsje erger. Weer minder dus beter. Meer achterover.

Table G.9: Condition preferences of participant 8 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	15	Buffet goed. In ac hogere frew. Idee dat pitch harder valt, abrupter. Deze gaat te geleidelijk. Richting een 3.
		16	Deze pitchbreak beter.
		18	Buffet perceptie goed. Nose drop goed, waarneembaar.
	ORC	17	Buffet dichterbij. Pitch break ook goed. Buffet te lang. Geen gevoel van g.
		19	Overgang buffet naar pitchbreak beter.
		20	Perceptie minder sterk, met ogen dicht. 3 of 4.
Large Roll	CWO	5	Minder pitch. Dit was even slecht. Hele grote sideforce.
		6	Pitch voel je bijna niet. Oh jongens. Te extreme sideforce. Geen gevoel van roll. Te weinig rating
		8	Pitch voel je niet. Heel veel sideforce. Te veel in herstel.
	ORC	3	Aardig. Eigenlijk niet zo erg de sideforces.
		4	Bijna zelfde als vorige. Nog steeds merkbaar.
		7	Voel pitch. Dit was interessant. Balletje alleen naar een kant.
Upsets	CWO	10	Pitch merk je niet. Veel sideforces extremer. C3 grote sideforce, ontbreekt g. Minder extreem. Gevoel gaat mee met wat je op de indicator ziet. Roll perf. Beter.
		11	G voel je niet., terugrollen teveel. Roll heel aardig, voel wel sideforces. Vooral recovery. Gs weer probleem. Het is lastig om de sideforces. Zou taak beinvloeden.
		13	Gevoel op verkeerde been door pitch. Redelijke sideslip. Niet storend maar erger door de sideforces.
	ORC	9	90 graden linker bocht goed. Pitch up voelde ik niet. 7/8 voor pitch. Roll is heel weinig. Sideforce eng->CR. Aanloop vd upset. Even naar recht, dan naar links. G ontbreekt.
		12	Hee roll naar links, maar zie ik niet. Grote sideforces niet storend. G mis ik weer. Maneuver oke. Omgekeerder illusie.
		14	Gevoel van links roll dan rechts. Niet vervelend. Zie het ook. C2 niet onaardig. Behalve g. voel nu ook yaw.

Table G.10: Condition preferences of participant 9 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	10	Voorwaarts zetje, meer voorover gevoel.
		13	Geen CR. Zetje hier zeker niet. Dit is meer de kanteling ipv surge.
		14	Je voelt hier de kist nose down. Voel je versnelling. Geen zetje. Vlak voor FF klein duwtje, acceleratie.
	ORC	9	Voelt buffet goed. Lichter worden beetje. Gevoel van doorzakken.
		11	Doorzakken delay. Klein voorwaarts zetje. Recovery redelijk goed.
		12	Timing goed. Zetje voorwaarts op goede moment.
Large Roll	CWO	15	Roll met fixed rudder. Haak effec
		18	Begon goed, initieel. Teveel sideforce erna. Opposite chasing.
		19	Overcorrectie. Delay op de voeten. Te weinig proportie qua roll rate. Of 7.
	ORC	16	Kleine time delay. Beweging zelf minder hinderlijk.
		17	Stuk betere timing. Gecoordineerd niet heel strak gebruik van voeten. Minder sideforce.
		20	Roll ingaan liep iets achter op visual. Onset niet volledig gecoordineerd.
Upsets	CWO	3	Verticale onset voel je niet. Veel sideforce. Pitch up niet te voelen in loadfactor. Doorrol boven neus sneller. Onset roll was beter nog steeds lateraal kracht.
		5	Inrollen out of sequence. Delay in roll gevoel. Lateraal grote krachten. Ook hier naijlen van laterale krachten.
		8	Onset vriendelijk. Timing beter. Groot naijlen. Roll in goed, maar daarna veel sideforce. Gevoel van inrollen goed.Haak effect, neus gaat naar buiten.
	ORC	4	Lateraal bewegingen minder uitgesproken. Roll beter aansluiten en gecoordineerd. Zakken neus meer verticale gevoel. Ook grote laterale cue, yaw damper.
		6	Dissonant roll delay. Pitch ook delay volgens mij. Ook hier time delay. Lateraal minder gebalanceerd. Verschiler in zien en voelen.
		7	Ook time delay. Onset heel vriendelijk lateraal uitzwaaien wat minder heftig. Visual achter motion. Lateraal viel mee. Out of sequences.

Table G.11: Condition preferences of participant 10 derived from comments provided during the SRS experiment runs.

Run			Comments
Symmetric Stall	CWO	5	Stuk beter. Buffet realistischer. Recovery voelde je echt. Sim kantelde echt naar voren.
		6	Te gentle. Buffet te zwak. Recovery voel je wel, maar te weinig. Weinig verschil in g.
		8	Weinig verschil met de vorige.
	ORC	3	Buffet goed. Recovery heel snel, tijdens recovery ook nog buffet. Nauwelijks verschil in g.
		4	Initieel hetzelfde. Recovery nog steeds weinig verschil in g.
		7	Recovery voelde je beter. Verschil in g, duwtje. Neus.
Large Roll	CWO	10	Slechter. Tijdens roll naar rechts voel je alleen initiele roll maar al vrij snel remmen en sideforce andere kant.
		11	Hetzelfde, laatste roll terug, lichaam helemaal naar links.
		13	Hing verkeerde kant op in de stoel. Geeft verkeerde indruk.
	ORC	9	Sideforce niet zoals echte werkelijkheid.
		12	Stuk beter . Te gentle roll rate maar niet storend. Deze voelde goed aan. Je ziet roll rate en voelt hetzelfde. Wordt niet andere kant vd stoel.
		14	Deze voelde goed aan. Je ziet roll rate en voelt hetzelfde. Wordt niet andere kant van de stoel.
Upsets	CWO	15	Initieel goed behalve g. Roll rate decelleratie gevoel. Stop rotatie en terug, tegenkracht. Positieve g voel je niet.
		16	Mis +-g. Init roll goed. Laatste roll voel je al een decelleratie. Init roll goed. Pitch en g init slecht.
		18	Heftige roll links voel je decelleratie. Hangt verkeerde kant. Eigenlijk weer hetzelfde. Recovery lijkt.
	ORC	17	Roll rate goed. Pitch kan beter init. Beter dan de vorige. Voelde iets natuurgetrouwder. +- g moet beter. Verschil in roll rate.
		19	Tot nu toe de beste. Met name overgang g voel je als rotatie. 2e gedeelte voelt goed qua roll rate, beetje gentle wel.
		20	Hij voelde goed aan. Overgang in g minder. Minder gentle dan vorige. Decelleratie roll komt overeen met visual. Weinig verschil, iets feller. Pitch kan echt beter.

Part III

Preliminary Thesis:

Model Predictive Control-based Motion Cueing for Aircraft Upset Simulation

(This part is already graded for AE4020)

Chapter 1

Introduction

Accidents that are caused by Loss of Control In-Flight (LOC-I) are currently the largest contributor to worldwide commercial aviation fatalities according to International Air Transport Association (IATA) [14]. LOC-I accidents include situations in which the flight crew was unable to maintain control of the aircraft, leading to an unrecoverable deviation from the intended flight path. Research over the past decade intended to decrease the number of LOC-I accidents resulted in an addition of Upset Prevention and Recovery Training (UPRT) in the pilot training curriculum as stated by the Federal Aviation Administration (FAA) and European Union Aviation Safety Agency (EASA) [15].

This UPRT is partially done in Flight Simulator Training Devices (FSTDs), however, the current options in FSTDs are limited and special care should be taken to prevent the introduction of negative training. Two main problems arise when UPRT is performed in FSTDs: the lack of a complete and validated aircraft model outside the normal flight envelope, that is where LOC-I often occurs [4], and secondly, the difficulty of representing the aircraft motion with realistic motion cues [16].

Current FSTDs are often equipped with a hexapod motion system (also called Stewart's Platform). Such a motion system is able to move in 6 Degrees of Freedom (DOF), however, the motion in all these DOF is limited. Especially when a so-called Classical Washout (CW) Motion Cueing Algorithm (MCA) is used, which is the case in almost all FSTDs. This type of MCA is tuned in a way that the motion system stays within limits for all training scenarios, which for many scenarios is not optimal.

An upcoming trend in the automotive simulator research industry is the use of a new type of MCA, based on Model Predictive Control (MPC) [17, 18, 19, 20, 21, 22, 23]. This algorithm uses a prediction of the driver/pilot control input to optimise the motion trajectory of the simulator over a certain horizon. In this way, more efficient use of the simulator motion space can be achieved, which could lead to higher motion cueing fidelity.

The specific investigation addressed in this study is about exploring the usefulness of this MPC-based MCA for the use in UPRT. This will be done with the help of an offline analysis, assessing the objective fidelity and a pilot/human experiment assessing the perceived fidelity. The main goal of this study is to increase the fidelity of motion cues, perceived by the pilot during UPRT, by means of implementing a new MCA based on MPC in a hexapod flight simulator.

1.1. Research Objective

The main research objective of this thesis is:

> *To achieve higher perceptual motion cueing fidelity during Upset Prevention and Recovery Training in a hexapod simulator by means of implementing a new Motion Cueing Algorithm based on Model Predictive Control and performing an offline analysis to assess the objective motion cueing fidelity, followed by a pilot-in-the-loop experiment to assess the perceptual motion cueing fidelity.*

To reach this objective, multiple sub-objectives are proposed, which are presented below.

SO 1. Perform a literature review on flight simulator motion cueing.

SO 2. Perform a literature review on MPC-based MCAs.

SO 3. Develop an offline MPC-based MCA.

SO 4. Perform a literature review on UPRT.

SO 5. Perform an offline evaluation of objective fidelity of the proposed MCA.

SO 6. Design an experiment plan for a pilot-in-the-loop experiment in the SRS.

SO 7. Perform an experiment in the SRS to evaluate the perceived motion cueing fidelity for the original and the proposed MCA.

A schematic overview of the thesis project, showing all objectives is presented in Figure 1.1. During the final thesis phase, UPRT flight test data obtained with the PH-LAB laboratory aircraft from the Delft University of Technology (TU Delft) will be used to design relevant motion scenarios for the experiment.

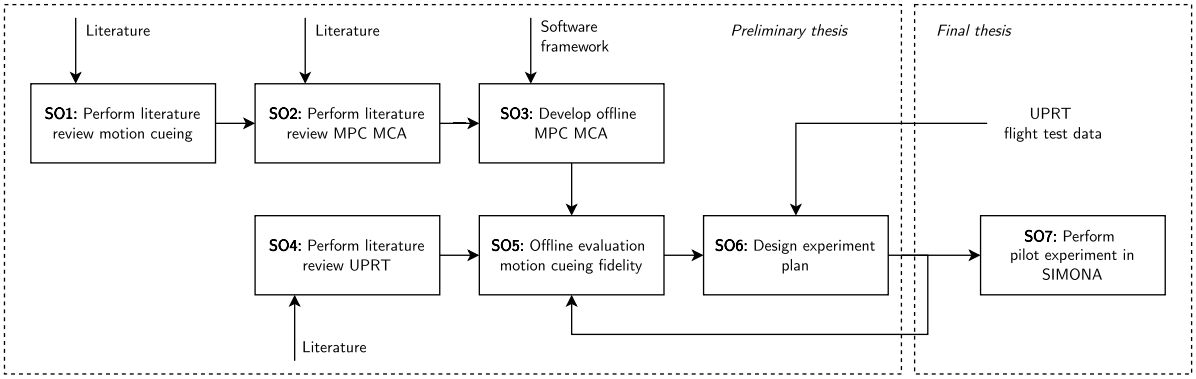


Figure 1.1: Schematic overview of the objectives of the thesis project.

1.2. Research Questions

In this section, the main research question is posed along with sub-questions. These smaller sub-questions together answer the main research question. The main research question is as follows:

> *To what extent can motion cueing fidelity during Upset Prevention and Recovery Training performed in a hexapod simulator be improved using a Motion Cueing Algorithm based on Model Predictive Control with respect to a Classical Washout Motion Cueing Algorithm?*

This research question is then further divided into five questions, which are subsequently divided into smaller sub-questions. All research questions are listed below.

RQ 1. How does motion cueing in a hexapod flight simulator work?

- **1a.** How can you define the fidelity of a flight simulator?
- **1b.** How can you define a hexapod motion system?
- **1c.** What type of motion cueing algorithms exist?
- **1d.** How can you evaluate motion cueing?

RQ 2. How do MPC-based MCAs work?

- **2a.** What is optimal control?

2b. How can a continuous-time optimal control problem be parameterised into a numerical finite-dimensional optimal control problem?

2c. What is MPC?

2d. How is MPC used for motion cueing?

RQ 3. How are FSTDs currently used for UPRT?

3a. To what extent are extended aerodynamic models required for UPRT in FSTDs?

3b. What training scenarios should be simulated for UPRT?

3c. What are the difficulties of motion cueing for UPRT?

3d. What are the limitations and shortcomings of current MCAs for UPRT?

SQ 4. How can the implementation of an MPC-based MCA during UPRT affect the **objective** motion cueing fidelity with respect to the current SIMONA CW MCA?

4a. What optimal control and MPC formulation should be used for the analysis?

4b. What is the influence of the prediction strategy on the objective motion cueing fidelity?

4c. What is the influence of the prediction horizon length on the objective motion cueing fidelity?

4d. What is the influence of the cost function weights on the objective motion cueing fidelity?

SQ 5. How can the implementation of an MPC-based MCA during UPRT affect the **perceptual** motion cueing fidelity with respect to the current SIMONA Research Simulator (SRS) CW MCA?

5a. What experiment conditions with different MCA settings should be used during the experiment?

5b. What metrics should be used to evaluate the perceived motion cueing fidelity?

5c. What type of experiment participants are required for the experiment?

5d. What data should be logged during the experiment?

The first three research questions and their sub-questions are completely answered in this report. The fourth and fifth research questions will be partly answered. For the fourth research question, additional UPRT motion scenarios will be analysed during the final thesis phase. The fifth research questions can be answered completely after the experiment is performed in the final thesis phase.

1.3. Report outline

The outline of this preliminary thesis report is as follows. To begin in Appendix 2, an introduction to motion cueing in flight simulators is presented. In this chapter, the required definitions, principles, and equations to understand and apply motion cueing are gathered. Furthermore, methods and metrics that can be used to evaluate MCAs are elaborated. Finally, the SRS is presented with its motion cueing relevant characteristics.

Thereafter, in Appendix 3, the theory about MPC and MPC-based motion cueing are discussed. Moreover, in this chapter, a literature review on how MPC-based MCAs are used in previous studies is presented.

In Appendix 4, the requirements and shortcomings of current FSTDs are elaborated. Furthermore, the training scenarios that are relevant to UPRT are identified.

Subsequently, in Appendix 5, the method, results and recommendations of the offline analysis to compare the objective fidelity of the MPC-based and CW MCAs are presented. Additionally, the stall scenario used for this analysis is described and the formulation used for the optimal control and MPC algorithms are depicted.

This report closes off with the preliminary experiment plan and design in Appendix 6. In this chapter, among others, the preliminary hypotheses, experiment variables, conditions and procedures are discussed.

Chapter 2

Motion cueing in flight simulators

In this chapter, the first research question posed in section 1.2 is answered.

How does motion cueing in a hexapod flight simulator work?

The objective of this chapter is to gather the required definitions, principles, and equations to understand and apply motion cueing for flight simulators. Furthermore, methods and metrics that can be used to evaluate MCAs are elaborated. Finally, the SRS is presented with its relevant motion cueing characteristics.

The chapter starts off with the definition of the different fidelity types and the reference frame convention used in this study in section 2.1 and Figure 2.2, respectively. In section 2.3 and section 2.4, the equations related to inertial acceleration and Euler angles are presented. Thereafter, in section 2.5, the basic working principle of motion cueing is elaborated on the basis of the human vestibular system. Subsequently, the mathematical definition of a hexapod motion system is introduced in section 2.6, after which common MCAs are presented in section 2.7. Motion cueing error definitions are given in section 2.8 and in section 2.9, methods to evaluate MCAs are elaborated. Finally, in section 2.10, the SRS with its characteristics is presented, followed by the chapter conclusions in section 2.11.

2.1. Fidelity

According to Sinacori [24], the veracity of a flight simulator is often expressed in fidelity. In practice, four types of fidelity are used, which are explained below with respect to motion cueing. A schematic overview of the different fidelity types can be seen in Figure 2.1.

Objective/engineering fidelity - Sinacori [24] and Heffley et al. [25] describe objective fidelity as the degree of similarity between the aircraft and the simulator. This type of fidelity can be applied to fidelity related to the human senses, for example, aural, visuals, motion, and haptics. For objective fidelity, it is possible to completely isolate and measure the objective fidelity (aural, visual, motion, etc.) under investigation. This means that, when, for example, the objective motion cueing fidelity is studied, the visual, aural, and haptic objective fidelity does not interfere with or influence this fidelity.

Perceptual fidelity - Sinacori [24] and Heffley et al. [25] describe perceptual fidelity as the degree of similarity between how the pilot experiences the aircraft and the simulator. This perceptual fidelity, like objective fidelity, can also be applied to the human senses. For this type of fidelity, it is *not* possible to completely isolate the perceptual fidelity (aural, visual, motion, etc.) under investigation. This means that, when, for example, the perceptual motion cueing fidelity is studied, the visual, aural, and haptic *objective* fidelity can interfere with or influence this *perceptual* motion cueing fidelity. When the perceptual motion cueing fidelity is studied, it is thus of importance that the visual, aural, and haptic *objective* fidelity remain fixed.

Behavioural fidelity - Heffley et al. [25] describe behavioural fidelity as the degree of similarity between the pilot's (control) behaviour/strategy in the aircraft and the simulator. For FSTDs, this behavioural fidelity is the most important fidelity and one should aim for the highest achievable behavioural fidelity during the design of a FSTD. Like performance fidelity, behavioural fidelity can not be applied individually to the various human senses and is always a result of all types of objective fidelity. However, it is possible to study the influence of, for example, the *objective* motion cueing fidelity on the *behavioural* fidelity by varying the objective motion cueing fidelity and fixing the other objective fidelity.

Error/performance fidelity - Heffley et al. [25] describe error fidelity as the degree of similarity between the pilot performance in the aircraft and the simulator. This performance fidelity, contrary to objective and perceptual fidelity, can also not be applied individually to the various human senses and is always a result of the objective fidelity of all human senses. However, again it is possible to study the influence of, for example, the *objective* motion cueing fidelity on the *performance* fidelity by varying the objective motion cueing and fixing the other objective fidelity.

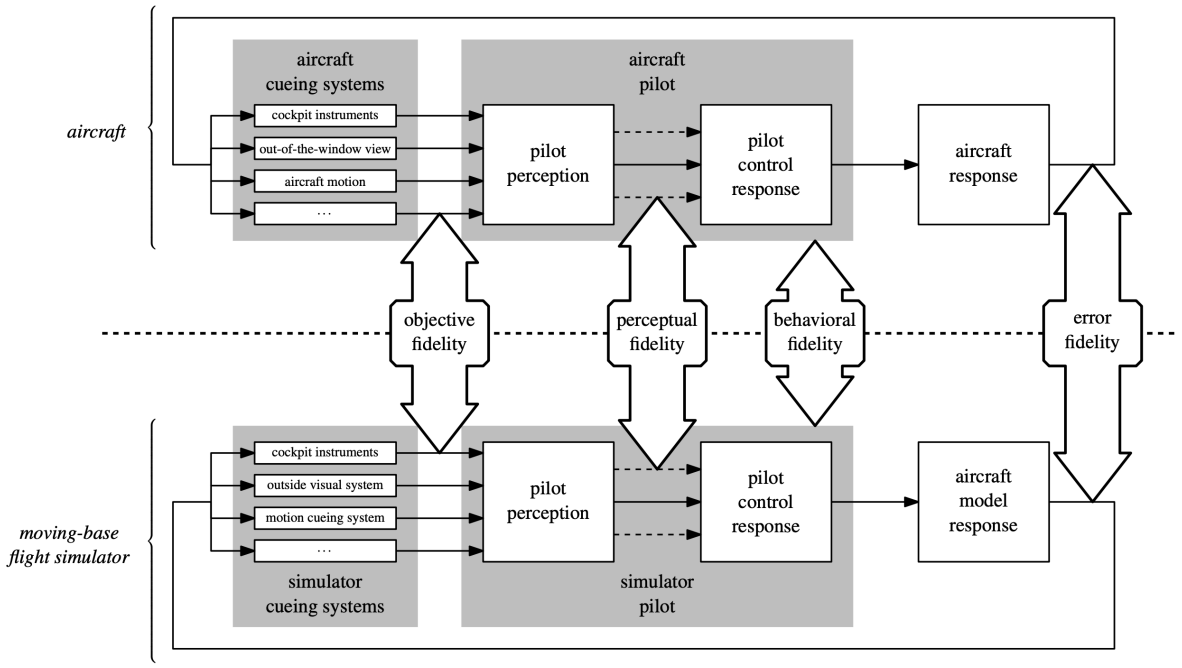


Figure 2.1: Schematic representation of the different types of flight simulator fidelity, from Pool [1].

2.2. Reference frames

During this study, the reference frame convention is used from Reid and Nahon [2]. All the simulator and equivalent aircraft reference frames are depicted in Figure 2.2. The relevant reference frames are further elaborated on below.

Inertial frame, F_I : The inertial reference frame is Earth-fixed with its z-axis aligned with the gravity vector. The origin is located in the centroid of the simulator's base platform, also known as the Lower Gimbal Point (LGP). The x-axis is pointing forwards, through the simulator projector screens when in the neutral position.

Platform frame, F_D : The platform reference frame is a moving/body frame that moves with the simulator's upper platform. The origin is located in and attached to the centroid of the simulator's upper platform, the Upper Gimbal Point (UGP). The x-axis points forward, through the simulator projector screens and the z-axis points downward. The x-y plane is parallel to the upper platform, which is parallel to the simulator cabin. The equivalent aircraft reference frame is F_C.

Simulator frame, F_S: The simulator reference frame is a moving/body frame that moves with the simulator cabin. For this study, the origin is located in and attached to the Design Eye Reference Point (DERP). The x-axis points forward, through the simulator projector screens and the z-axis points downward. The x-y plane is parallel to the upper platform and is thus parallel to reference frame F_D. The equivalent aircraft reference frame is F_A.

Pilot frame, F_{Ps}: The pilot reference frame is attached to the simulator's pilot head with its origin located at and attached to a point midway between his left and right vestibular organs. The x-axis points forward through the pilot's eyes and the z-axis downward along the spine. The x-z plane contains the pilot's vertical plane of symmetry. The equivalent aircraft reference frame is F_{Pa}. This frame is not further used in this study, instead, the simulator frame, F_S, is chosen at a convenient and fixed location, the DERP, which is assumed to be approximately equal to the pilot's head location. Meaning that in this study $F_{Ps} \approx F_S$ and F_S, defined in the DERP, will be used to assess the motion cueing fidelity.

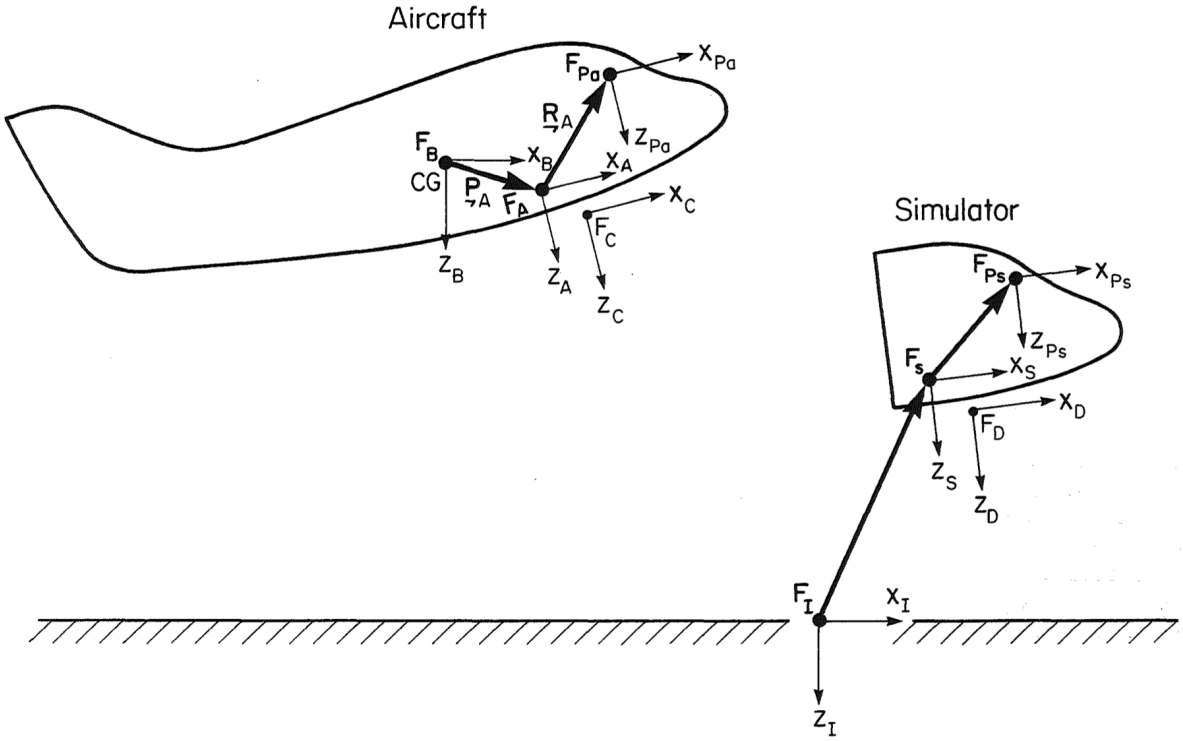


Figure 2.2: Schematic overview of the used aircraft and simulator reference frames, from Reid and Nahon [2].

2.3. Inertial acceleration

In (2.1), as defined by Reid and Nahon [2], the inertial acceleration of a point, A, on a rigid-body, B, located at a distance, $\boldsymbol{r}_{B,A}$ from the origin of a body reference frame of the body B is depicted.

$$\boldsymbol{a}_A = \boldsymbol{a}_B + [\boldsymbol{\Omega}_B\boldsymbol{\Omega}_B + \dot{\boldsymbol{\Omega}}_B]\boldsymbol{r}_{B,A} \tag{2.1}$$

Where the linear accelerations of the body, B, in an inertial frame, can be defined by the body accelerations, body angular rates, and the body velocities, as can be seen in (2.2). And where the skew-symmetric angular velocity matrix, depicted in (2.3), is used.

$$\boldsymbol{a}_B = \begin{bmatrix} \dot{u}_B & + & q_B w_B & - & r_B v_B \\ \dot{v}_B & + & r_B u_B & - & p_B w_B \\ \dot{w}_B & + & p_B v_B & - & q_B u_B \end{bmatrix} \tag{2.2}$$

$$\boldsymbol{\Omega}_B = \begin{bmatrix} 0 & -r_B & q_B \\ r_B & 0 & -p_B \\ -q_B & p_B & 0 \end{bmatrix} \tag{2.3}$$

When using the angular velocity and acceleration vectors, $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$, and the vector cross product, a more commonly known expression, shown in (2.4), is obtained for the inertial acceleration.

$$\boldsymbol{a}_A = \boldsymbol{a}_B + \boldsymbol{\alpha} \times \boldsymbol{r}_{B,A} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{B,A}) \tag{2.4}$$

2.4. Euler angles and angular velocities

Three Euler angles are used to define the orientation between two reference frames. A vector expressed in one frame, F_1, can be expressed in a different frame, F_2, with the help of these Euler angles. The rotation of a vector from F_1 to F_2 can be performed with the rotation matrix depicted in (2.5), where the attitude of F_2 with respect to F_1 is expressed with the Euler angles.

$$\boldsymbol{R}_{F_1,F_2} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2.5}$$

The angular velocity of a body expressed in an inertial reference frame can be expressed by the time derivative of the Euler angles. This angular velocity can be obtained by the angular velocity of the body in its own reference frame and the Euler angles expressing the orientation of the body frame with respect to the inertial frame. This relation is presented in (2.6). The opposite relation is obtained by the inverse of (2.6), resulting in (2.7).

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.6}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.7}$$

2.5. Human vestibular system

The human vestibular system is a sensory system located in the human head near the ears. According to Reid and Nahon [2], the vestibular system consists of the semi-circular canals and the otoliths organs and perceives the motion cues in a simulator. The three semi-circular canals, positioned approximately orthogonal to each other, are used to sense angular motion, $\boldsymbol{\omega}$, and can be interpreted as an angular velocity sensor working around three axes. The otolith organ is able to perceive specific forces, being the non-gravitational forces per unit mass. This can be defined as the vector difference between the inertial acceleration and the gravitational acceleration, as seen in (2.8) from Reid and Nahon [2].

$$\boldsymbol{f} = \boldsymbol{a} - \boldsymbol{g} \tag{2.8}$$

Since simulators, and especially hexapod simulators, have limited actuator travel and thus a limited motion space, sustained accelerations can not be simulated by accelerating the simulator cabin while staying within the motion space. However, the human vestibular system senses specific forces, and not accelerations, therefore, it is possible to use gravitational acceleration as a sustained motion cue. This is done by slowly tilting the simulator resulting in a change of specific force. In combination with a high-frequency onset acceleration of the simulator, a full motion cue can be represented, as can be seen in Figure 2.3.

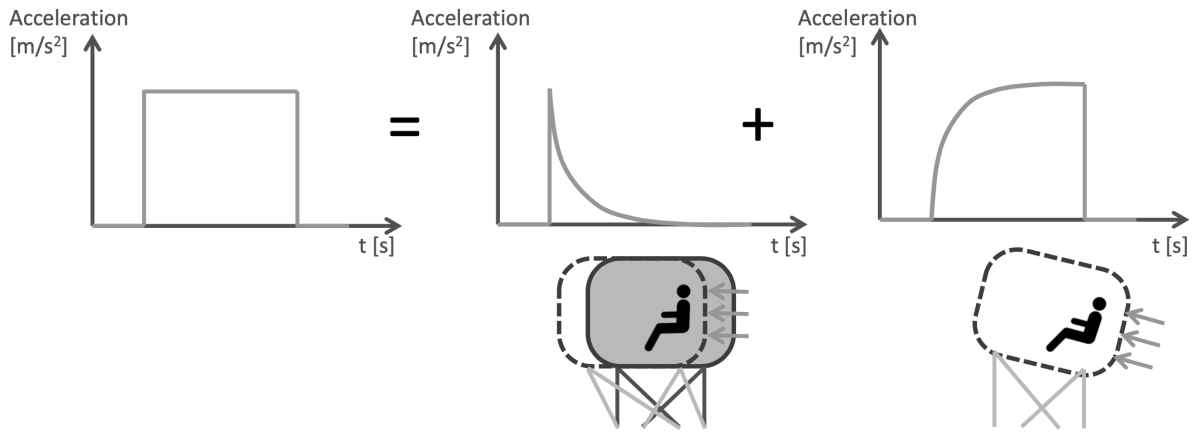


Figure 2.3: Illustration of fully complementary cueing of specific forces, where motion cues can be split into a high-frequency onset part and a low-frequency sustained part, from van Leeuwen [3].

2.6. Hexapod motion base

The naming of a hexapod simulator or motion base comes from the number of "legs", which in this case are six actuators, connecting the base and upper platform of the system. A complete schematic overview of a hexapod motion system is depicted in Figure 2.4.

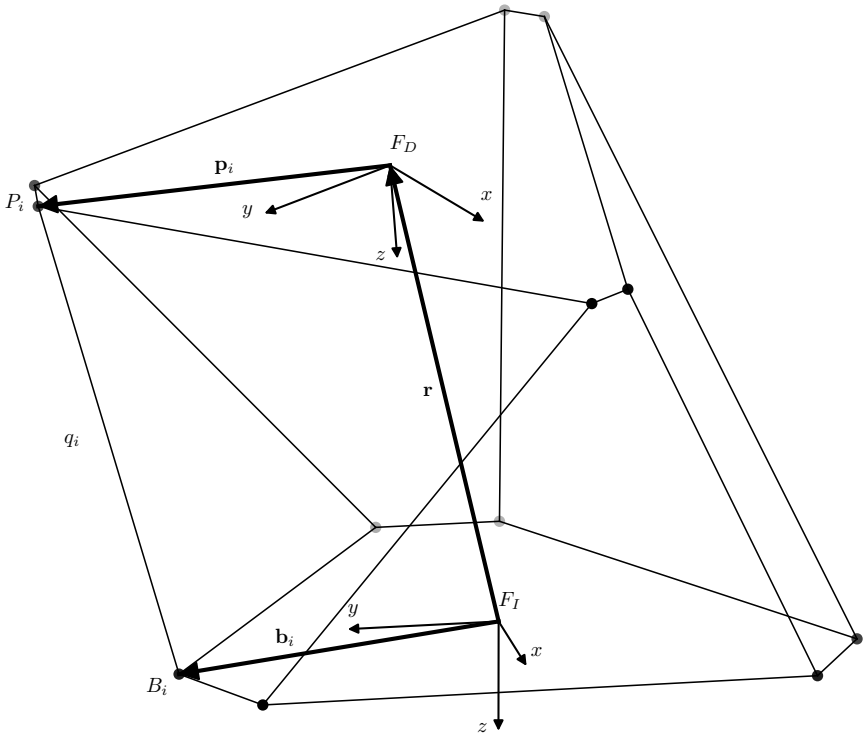


Figure 2.4: Schematic overview of a hexapod motion system.

The actuators are connected to the simulator base, a non-moving base that is fixed to the ground, with six universal joints. The fixed locations of these joints with respect to the Lower Gimbal Point (LGP) expressed in the F_I reference frame can be described by the vector presented in (2.9).

$$\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{b}_4, \boldsymbol{b}_5, \boldsymbol{b}_6]^\top \in \mathbb{R}^{18} \tag{2.9}$$

The simulator's upper platform, a moving platform, is connected to the actuator with again six universal joints. The location of these joints with respect to the Upper Gimbal Point (UGP) in the F_D reference frame is again fixed and is captured by the vector in (2.10).

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4, \boldsymbol{p}_5, \boldsymbol{p}_6]^\top \in \mathbb{R}^{18} \tag{2.10}$$

The length of the actuators determines the position and attitude of the simulator platform. The length of the actuators is described by the vector in (2.11).

$$\boldsymbol{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^\top \in \mathbb{R}^6 \tag{2.11}$$

To determine the length of the actuators, a method called inverse kinematics is used [26]. This method derives the actuator lengths from the position of the UGP with respect to the LGP, which is $\boldsymbol{r}$, the orientation of the upper platform with respect to the inertial reference frame, being $\boldsymbol{R}_{F_D,F_I}$, and the base gimbal and platform gimbal positions, $\boldsymbol{B}$ and $\boldsymbol{P}$, respectively. This relation is presented in (2.12). The individual actuator length can be derived with (2.13).

$$\boldsymbol{q} = \|\boldsymbol{r} - \boldsymbol{B} + (\boldsymbol{R}_{F_D,F_I}\boldsymbol{P}^\top)^\top\| \tag{2.12}$$

$$q_i = \|\boldsymbol{r} - \boldsymbol{b_i} + \boldsymbol{R}_{F_D,F_I p_i}\| \tag{2.13}$$

The velocity of the actuators, or the rate of change of actuator length, is the time derivative of the actuator length. These velocities are derived with the simulator state derivative and the actuator length Jacobian matrix, as seen in (2.14). The actuator length Jacobian is calculated with respect to the simulator state and can be found in (2.15).

$$\dot{\boldsymbol{q}} = \boldsymbol{J_q}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{2.14}$$

$$\boldsymbol{J_q}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial q_1}{\partial x} & \frac{\partial q_1}{\partial y} & \frac{\partial q_1}{\partial z} & \frac{\partial q_1}{\partial \phi} & \frac{\partial q_1}{\partial \theta} & \frac{\partial q_1}{\partial \psi} \\ \frac{\partial q_2}{\partial x} & \frac{\partial q_2}{\partial y} & \frac{\partial q_2}{\partial z} & \frac{\partial q_2}{\partial \phi} & \frac{\partial q_2}{\partial \theta} & \frac{\partial q_2}{\partial \psi} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\partial q_6}{\partial x} & \frac{\partial q_6}{\partial y} & \frac{\partial q_6}{\partial z} & \frac{\partial q_6}{\partial \phi} & \frac{\partial q_6}{\partial \theta} & \frac{\partial q_1}{\partial \psi} \end{bmatrix} \tag{2.15}$$

2.7. Motion cueing algorithms

Motion systems are driven by a MCA that translates an input signal, often specific forces and angular rates as a result of aircraft motion, into a simulator control input. In this study, two types of MCAs are discussed, being filter-based MCAs and optimisation-based MCAs.

2.7.1. Filter-based

Filter-based MCAs, and especially Classical Washout (CW) MCAs, are the most widely used type of MCA in flight simulators. This MCA is based on the use of linear high-pass filters for cueing the high-frequency spectrum of the specific forces and angular velocities. Furthermore, this type of MCA uses low-pass filters for cueing the low-frequency spectrum of the lateral and longitudinal specific forces by tilting the simulator. The latter is often called tilt-coordination. A schematic overview of the CW MCA is depicted in Figure 2.5.

Before a CW MCA can be used, it is required to tune the algorithm. This is because a CW MCA does not account for the physical simulator motion limits. Tuning is often done manually and a selection

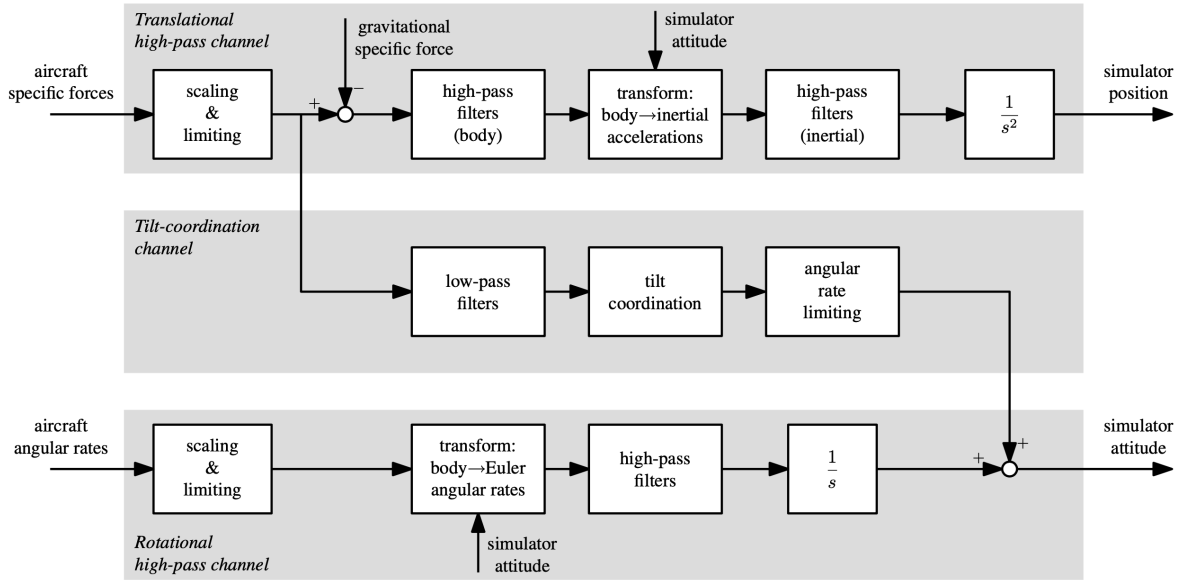


Figure 2.5: Schematic representation of the CW algorithm as defined by Reid and Nahon [2], from Pool [1]

of parameters found in Table 2.1 should be determined. These parameters are often determined in a conservative way, meaning that worst-case tuning is used to ensure that the simulator stays within its limits for all possible motion scenarios. This results in not-"worst-case" reference motion being sub-optimally represented by the algorithm.

Table 2.1: Typical CW motion filter tuning parameters.

Parameter	Description	DOFs
K	Filter gain	All
ω_{hp}	Second-order high-pass break frequency	All
ζ_{hp}	Second-order high-pass damping ratio	All
ω_b	First-order high-pass break frequency	All
ω_{lp}	Second-order low-pass break frequency	x, y
ζ_{lp}	Second-order low-pass damping ratio	x, y

Parrish et al. [27] were the first to propose an Adaptive Washout (AW) MCA. With this MCA, an attempt was made to avoid the sub-optimal tuning of the CW MCA by real-time adjusting the filter gains during simulation. While minimising a cost function that penalises the difference between the aircraft and simulator motion, the motion filter parameters are updated. However, no significant improvements were observed when a AW MCA is applied compared to a CW MCA. Moreover, the algorithm is non-linear, more complex, and sensitive to instability. Due to these reasons, this AW MCA is not commonly used.

2.7.2. Optimisation-based

During this study, two optimisation-based MCAs are discussed. The first algorithm is an optimal MCA, which is actually a trajectory optimisation problem, or an optimal control problem. This MCA is only possible to use offline when an entire reference aircraft motion scenario is available. The algorithms minimise a cost function that penalises the difference between the reference aircraft and simulator motion, resulting in a simulator control input for the entire reference motion duration. The second optimisation-based MCA is a so-called MPC-based MCA that solves smaller optimal control problems, where the size is depending on the prediction horizon length, each time step. This type of MCA could be applied offline, where the algorithm iterates through a reference motion scenario. Moreover, this

algorithm could be applied in real-time, where each time step the algorithm solves the optimal control problem using a reference aircraft motion depending on the pilot control input. This type of MCA is further elaborated in Appendix 3. A summary of the main differences of the previously mentioned MCAs can be seen in Table 2.2.

Table 2.2: Comparison of filter-based and optimisation-based MCA, based on Cleij [7]

	$\mathbf{MCA}_{FIL}$	$\mathbf{MCA}_{OPT}$	$\mathbf{MCA}_{MPC}$
Type	Filter-based	Optimisation-based	Optimisation-based
Real-time capable	Yes	No	Yes
Pilot-in-the-loop applications	Suitable	Not suitable	Suitable
Sampling rate	$\geq$100 Hz	$\geq$20 Hz	$\leq$100 Hz
Future reference	Not required	Entire trajectory	Prediction horizon
Accounting simulator limits	No, manual tuning	Yes	Yes

2.8. Motion cueing errors

Grant and Reid [28] divided motion cueing errors into three categories, false cues, scale errors, and phase errors. With respect to false cues, they found that scale and phase errors are less determinative for the reduction of perceived fidelity. The definition of the error categories is provided below.

False cue: *Type 1)* is a motion cue in the simulator that has the opposite direction of the cue in the aircraft. *Type 2)* is a motion cue in the simulator when that motion cue was not present in the aircraft. *Type 3)* is a high-frequency distortion of a sustained cue, while only a sustained cue was present in the aircraft.

Scale error: A motion cue with a scale error means that the actual motion cue in the aircraft is larger than the motion cue in the simulator. The extreme case of a scale error is a missing cue.

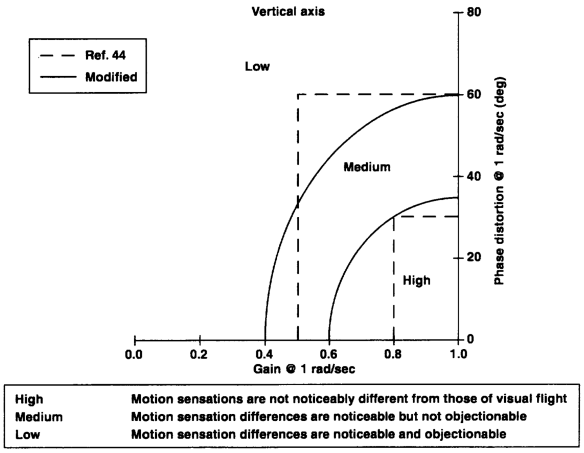
Phase error: A motion cue in the simulator that has a delay with respect to the motion cue in the aircraft. This can occur due to the MCA used and due to the limits of the simulator, for example, the actuator velocities or accelerations.

With these definitions, it is possible to more precisely identify and classify the deficiencies of an MCA. For example, this could be useful for tuning a MCA. Moreover, it could help to explain the results of objective and perceptual motion cueing fidelity analyses.

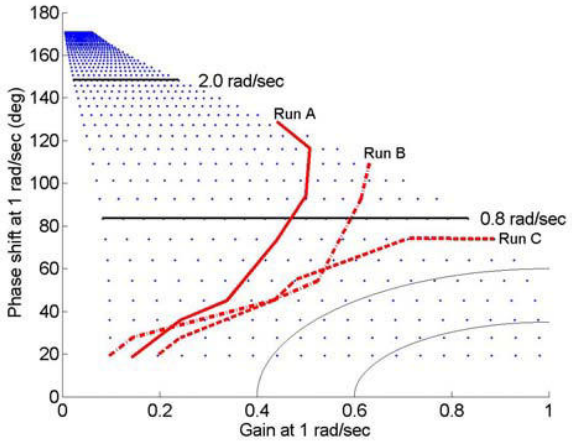
2.9. Motion cueing algorithm evaluation and tuning

Methods to evaluate MCAs are mostly developed for filter-based MCAs such as the CW MCA. The principle of these methods is based on the gain and phase distortions as a result of the CW tuning parameters. Sinacori [29] was the first to come up with a motion fidelity criterion based on this principle. The criterion indicated three fidelity regions (low, medium, high) for CW high-pass filters. These criteria can be found in Figure 2.6a by the boxes indicated by a dashed line. Schroeder [30] proposed a modified version of the Sinacori criterion, resulting in the Sinacori-Schroeder criterion depicted in Figure 2.6a. Advani and Hosman [31], applied this principle to the transfer function of the entire motion system, assuming it is linear. This resulted first in the Advani-Hosman criteria. Later, Hosman and Advani [32] developed the Objective Motion Cueing Test (OMCT), a more extensive method to compare entire motion systems of different simulators in the frequency domain.

Gouverneur et al. [33] reversed this principle and derived a method two tune the algorithm using the Sinacori-Schroeder criterion. Gouverneur displayed the possible filter settings as a result of the gain and break-frequency parameters onto the Sinacori-Schroeder criterion, resulting in Figure 2.6b. In this way, an analysis could be performed for a certain motion scenario, that results in a region of allowable CW filter settings to keep the simulator within its physical limits. In Figure 2.6b, the red lines indicate the allowable filter settings for the respective motion scenario, meaning that the settings above and to the left of these lines are possible.



(a) The Sinacori-Schroeder fidelity criterion. The dashed line shows the original fidelity criterion from Sinacori [29]. From Schroeder [30].



(b) Example Gouverneur analysis for multiple simulations runs. From Gouverneur et al. [33].

Figure 2.6: The Sinacori-Schroeder criterion and a Gouverneur analysis example using this criterion.

2.9.1. Evaluating non-linear algorithms

Non-linear motion cueing algorithms can be evaluated by comparing the simulator output inertial signal as a result of the simulator movement with respect to the reference aircraft motion. For this, metrics can be used that are suitable for comparing signals. The Root Mean Square Error (RMSE) is a measure to indicate the magnitude difference between two signals and is depicted in (2.16). Due to the squared difference in this relation, large differences between two signals have relatively more effect on the RMSE than small differences. Other studies that use the RMSE to objectively evaluate motion cueing can be found in [34, 35, 21, 36].

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N} (\hat{y}_n - y_n)^2}{N}} \tag{2.16}$$

To identify the resemblance of two signals, meaning the shape similarity of signals, the Pearson Correlation Coefficient (PCC) is used, as seen in (2.17). For this measure, only the difference in the shape of the two signals that are compared has an influence on the value. The amplitude difference, in contrary to the RMSE, is not penalised. Therefore, the PCC is a good measure to indicate smaller shape differences that are not properly reflected by the RMSE. Studies that also use the PCC to objectively evaluate motion cueing can be found in [21, 36].

$$PCC_{\hat{y}y} = \frac{1}{N-1} \sum_{n=1}^{N} \left(\frac{\hat{y} - \mu_{\hat{y}}}{\sigma_{\hat{y}}} \right) \left(\frac{y - \mu_y}{\sigma_y} \right) \tag{2.17}$$

2.10. SIMONA Research Simulator

In this study, the SRS will be the use-case simulator. The name of the simulator, SIMONA, stands for SImulation, MOtion and NAvigation which are the driving factors of the simulator's design. The main purposes of the simulator are research and education. Research topics include human perception, human performance, aircraft handling qualities, flight control system design, interface design and air traffic management. The cabin has a rough design that is not completely focused on one type of vehicle. The components of the cabin consist of a combination of off-the-shelf and custom-made parts. Touchscreen panels resemble a modern glass cockpit and can be programmed to display any kind of instrumentation and images, which contributes to a flexible cabin design. In this way, it is possible to easily switch between different aircraft types, helicopters and even cars. To further support different aircraft types, the cabin provides a yoke as well as a sidestick. The simulator has a hydraulic hexapod

motion system, which can be seen in Figure 2.7b. The outside of the SRS cabin and the collimated mirror attached to the cabin can be seen in Figure 2.7a.

(a) SRS cabin and outside of visual system.

(b) SRS motion system.

Figure 2.7: The SIMONA Research Simulator (SRS) of the Faculty of Aerospace Engineering, Delft University of Technology (TU Delft).

The motion system can be described by the parameters depicted in Table 2.3. With these geometric properties, the SRS has an available motion space defined from the neutral position that can be found in Table 2.4.

Table 2.3: SRS motion system characteristics, from Berkouwer et al. [8].

Parameter	Value
Platform radius	1650 mm
Base radius	1650 mm
Upper gimbal spacing	200 mm
Lower gimbal spacing	600 mm
Minimum actuator length	2081 mm
Maximum actuator length	3331 mm
Upper buffer length	50 mm
Lower buffer length	50 mm
Operational actuator stroke	1150 mm
Maximum actuator velocity	750 mm s^{-1}

Table 2.4: SRS motion space, from Berkouwer et al. [8].

DOF	Minimum	Maximum
Surge	−981 mm	1259 mm
Sway	−1031 mm	1031 mm
Heave	−636 mm	678 mm
Roll	−25.9°	25.9°
Pitch	−27.3°	24.3°
Yaw	−41.6°	41.6°

2.11. Chapter conclusions

This chapter is concluded with the answers to the first research question and its sub-questions and is provided below.

RQ 1. *How does motion cueing in a hexapod flight simulator work?*

The principle of motion cueing is based on the human vestibular system that senses specific forces and angular rates. A MCA is used to translate reference aircraft motion into a simulator control input that drives the actuators of the hexapod motion system.

1a. *How can you define the fidelity of a flight simulator?*

In general, four types of fidelity can be defined, being, objective, perceptual, behavioural and error fidelity. The objective fidelity indicates the difference between the aircraft and the simulator. The perceptual and behavioural fidelity indicates the difference in how the pilot perceives the simulator and behaves in the simulator. Finally, the error, or performance, fidelity indicates the difference between the aircraft response and the response of the aircraft

model used in the simulator as a result of the pilot control input.

1b. *How can you define a hexapod motion system?*
A hexapod motion system can be defined by six actuators connected two the base and upper platform by 6 gimbals on each side. Inverse kinematics is used to derive the actuator lengths from the position of the UGP location and the attitude of the upper platform.

1c. *What type of motion cueing algorithms exist?*
Two types of MCAs can be distinguished, being filter-based and optimisation-based MCAs. The CW MCA is the most used filter-based algorithm. For optimisation-based, a real-time suitable algorithm can be based on MPC.

1d. *How can you evaluate motion cueing?*
Filter-based MCAs can be evaluated by criteria such as the Sinacori-Schroeder [29], Advani-Hosman [31] or the OMCT [32]. Non-linear MCAs can objectively be evaluated by among others the RMSE and PCC.

Chapter 3

Model predictive control-based motion cueing

In this chapter, the second research question posed in section 1.2 is answered.

How do MPC-based MCAs work?

The objective of this chapter is to identify the required theory and definitions to understand optimal control and MPC-based MCAs. Furthermore, a literature review on how MPC-based MCAs are implemented is presented.

In section 3.1, the basic principles and background of optimal control problems, the main building block of MPC, are discussed. Subsequently, in section 3.2, the theory of numerical optimal control is presented to derive a mathematical description that can be deployed for use in MPC. Then, in section 3.3, the optimal control problem is taken to the real-time domain, resulting in the theory of MPC. Finally, in section 3.4, previous studies about MPC-based MCAs are investigated and chapter conclusions are presented in section 3.5.

3.1. Optimal control

Optimal control addresses the optimisation problems that arise from the control and scheduling of dynamical systems. According to Vinter [37], this field of research originated in the 1950s from among others the need of solving orbital and flight mechanics optimisation problems that arose in space programs. In the years to now, optimal control has been applied in many more areas. Examples of current use areas are process control, economics, aerospace, robotics and simulation. Vinter [37] describes optimal control as an outgrowth of the mathematical analysis of minima and maxima of functions (calculus of variations). However, compared to calculus of variations, optimal control additionally allows for new types of constraints that can be included in the mathematical problem.

The basic idea of optimal control is to find the optimum set of control inputs to a dynamical system to minimise an objective function or cost function that quantifies the difference between the desired system output and the actual system output. The desired system output can be described as a reference signal and can be seen as the input to the optimal control problem. The optimisation variables are the state of the system and control input to the system. Moreover, constraints can be imposed on the system state as well as control input.

3.2. Numerical optimal control

When applying optimal control in practice, the optimisation problem needs to be solved numerically while the system dynamics are often given in continuous-time in the form of differential equations. The general mathematical description of continuous time optimal control problems as defined by Rawlings

et al. [38] can be found in (3.1). In this equation, the time horizon is defined by $t \in [0, T]$ where T is the length of the trajectory. The state and control input trajectories are $x(\cdot)$ and $u(\cdot)$, respectively. The continuous time objective function is defined by a running cost, $\ell_c(x(t), u(t))$, also called the Lagrangian, and an optional final state cost, $V_f(x(T))$. The initial state is defined by (3.1b) and the system dynamics are described by (3.1c). Path constraints imposed on the state and control input are defined by (3.1d) and the final state constraint is defined by (3.1e).

$$\begin{aligned}
\underset{x(\cdot),u(\cdot)}{\text{minimise}} \quad & \int_0^T \ell_c(x(t), u(t))dt + V_f(x(T)) && \text{(3.1a)}\\
\text{subject to} \quad & x(0) = x_0 && \text{(3.1b)}\\
& \dot{x}(t) = f_c(x(t), u(t)), \quad t \in [0, T] && \text{(3.1c)}\\
& h(x(t), u(t)) \leq 0, \quad t \in [0, T] && \text{(3.1d)}\\
& h_f(x(T)) \leq 0 && \text{(3.1e)}
\end{aligned}$$

Since the optimal control problem in (3.1) is in continuous time, the optimisation problem has infinite-dimensional decision variables and an infinite number of constraints. To solve continuous-time optimisation problems, it should, somewhere in the solving process, be discretised.

3.2.1. Direct optimal control parameterisation

Rawlings et al. [38], depicts direct parameterisation as the most widely used method and this will also be the method used in the remainder of this study. This method first discretises the continuous-time optimal control problem into a finite-dimensional optimisation problem of the general form shown in (3.2). Thereafter, it can be solved by a Non-linear Programming (NLP) solver.

$$\begin{aligned}
\underset{w \in \mathbb{R}^{n_w}}{\text{minimise}} \quad & F(w) && \text{(3.2a)}\\
\text{subject to} \quad & G(x_0, w) = 0 && \text{(3.2b)}\\
& H(w) \leq 0 && \text{(3.2c)}
\end{aligned}$$

Single shooting, multiple shooting and collocation are approaches of direct parameterisation. As direct collocation will be the parameterisation used in the remainder of this study, it will be elaborated in this section by determining the mathematical definition. However, because direct single shooting and direct multiple shooting are the bases of direct collocation, it is needed to introduce these concepts to fully understand the latter.

3.2.2. Direct single shooting

The first step in the direct parameterisation process is to parameterise the continuous time control trajectory into a finite-dimensional vector $\mathbf{u}$. Which defines the control trajectory by the set $u(t) = \tilde{u}(t; \mathbf{u})$ for $t \in [0, T]$. Secondly, the state trajectory defined by the set $x(t) = \tilde{x}(t; \mathbf{u})$ for $t \in [0, T]$. By dividing the time horizon $[0, T]$ into N subintervals of the form $[t_i, t_{i+1}]$, the discretised control vector, $\mathbf{u} = (\mathrm{u}_0, \mathrm{u}_1, ..., \mathrm{u}_{N-1})$, with on each interval the vector $\mathrm{u}_i \in \mathbb{R}^m$ has a dimension of $n_{\mathbf{u}} = Nm$. Moreover, the initial state is defined by a discretised variable $\mathrm{x}_0 \in \mathbb{R}^n$.

With these definitions it is possible to discretise the path constraints, resulting in the definition of single shooting. The mathematical definition of single shooting is presented in (3.3). Note that also the initial (discretised) state vector, x_0, in addition to the discretised control vector ($\mathbf{u}$) are included as optimisation variables in single shooting. The objective function is still dependent on the entire discretised control vector $\mathbf{u}$. This requires sequential solving by a NLP solver. The total amount of decision variables is defined by $n_{w_{ss}} = n + Nm$.

$$\underset{\mathrm{x}_0,\mathbf{u}}{\text{minimise}} \quad \int_0^T \ell_c(\tilde{x}(t;x_0,\mathbf{u}),\tilde{u}(t;\mathbf{u}))dt + V_f(\tilde{x}(T;\mathbf{u})) \tag{3.3a}$$

$$\text{subject to} \quad \mathrm{x}_0 - x_0 = 0 \tag{3.3b}$$

$$H(\mathrm{x}_0,\mathbf{u}) \leq 0 \tag{3.3c}$$

3.2.3. Direct multiple shooting

Multiple shooting builds upon single shooting with as the main difference that now also the state trajectory is discretised in a finite-dimensional vector $\mathbf{x}$. This vector $\mathbf{x} = (\mathrm{x}_0, \mathrm{x}_1, ..., \mathrm{x}_N)$, with on each interval the vector $\mathrm{x}_i \in \mathbb{R}^n$ has a dimension of $n_{\mathbf{x}} = (N+1)n$. This allows the inclusion of the discretised state vector as an optimisation variable, which subsequently allows us to now also discretise the objective function and to include the system dynamics as optimisation constraints. A benefit of this is that it is possible to completely decouple the separate intervals, which allows parallel solving by NLP solvers. The final mathematical description of multiple shooting is presented in (3.4). The total amount of decision variables is defined by $n_{w_{ms}} = (N+1)n + Nm$.

$$\underset{\mathbf{x},\mathbf{u}}{\text{minimise}} \quad \sum_{i=0}^{N-1} \ell_i(\mathrm{x}_i,\mathrm{u}_i) + V_f(\mathrm{x}_N) \tag{3.4a}$$

$$\text{subject to} \quad \mathrm{x}_0 = x_0 \tag{3.4b}$$

$$\mathrm{x}_{i+1} = \tilde{x}_i(t_{i+1};\mathrm{x}_i,\mathrm{u}_i), \quad \text{for} \quad i = 0,...,N-1 \tag{3.4c}$$

$$H_i(\mathrm{x}_i,\mathrm{u}_i) \leq 0, \quad \text{for} \quad i = 0,...,N-1 \tag{3.4d}$$

$$h_f(\mathrm{x}_N) \leq 0 \tag{3.4e}$$

Compared to single shooting, multiple shooting has more decision variables. However, due to the possibility of parallel solving, it is in general solved faster. Moreover, the integration of the system dynamics and the minimisation of the objective function are now performed by the NLP solver, which simplifies the implementation and increases performance.

3.2.4. Direct collocation

For direct collocation, each subinterval $[t_i, t_{i+1}]$ is further divided into M collocation time intervals allowing for a control parameter and internal state parameter on each collocation interval. In contrary to multiple shooting, the resulting trajectories can be in the form of piecewise polynomials of the order $M-1$. Where the degree of precision can be controlled by the order of collocation M. The piecewise polynomials allow examining the trajectories at a sub-sample time scale.

For the mathematical definition of collocation, a new discretised internal state vector, $\mathbf{X} = [\mathrm{X}_0, ..., \mathrm{X}_{N-1}]$, with on each interval the vector, $\mathrm{X}_i = [\mathrm{x}_{i,1}, ..., \mathrm{x}_{i,M}]$, should be introduced that defines the state trajectory at the collocation time points. Subsequently, the internal state vector is included as an optimisation variable requiring the collocation conditions $G_i(\mathrm{x}_i, \mathrm{X}_i, \mathrm{u}_i)$ to be included as constraints. If interested, the full definition of the collocation conditions as defined by Rawlings et al. can be found in [38]. The full mathematical definition of direct collocation is depicted in (3.5). The total amount of decision variables is defined by $n_{w_{dc}} = ((M+1)N+1)n + Nm$.

$$\underset{\mathbf{x},\mathbf{X},\mathbf{u}}{\text{minimise}} \quad \sum_{i=0}^{N-1} \ell_i(\mathrm{x}_i, \mathrm{X}_i, \mathrm{u}_i) + V_f(\mathrm{x}_N) \tag{3.5a}$$

$$\text{subject to} \quad \mathrm{x}_0 = x_0 \tag{3.5b}$$

$$\mathrm{x}_{i+1} = F_i(\mathrm{x}_i, \mathrm{X}_i, \mathrm{u}_i), \quad \text{for} \quad i = 0, ..., N-1 \tag{3.5c}$$

$$0 = G_i(\mathrm{x}_i, \mathrm{X}_i, \mathrm{u}_i), \quad \text{for} \quad i = 0, ..., N-1 \tag{3.5d}$$

$$H_i(\mathrm{x}_i, \mathrm{X}_i, \mathrm{u}_i) \leq 0, \quad \text{for} \quad i = 0, ..., N-1 \tag{3.5e}$$

$$h_f(\mathrm{x}_N) \leq 0 \tag{3.5f}$$

3.3. Real-time optimal control: model predictive control

If the continuous-time optimal control problem shown in (3.1) is taken to the real-time domain, it can be called MPC. To do so, a time interval $[t_k, t_{k+1}]$, that determines the system's control input rate, is defined that divides the time horizon $[0, T_K]$ into K intervals. Note that for many real-time control processes, the total duration of the process, as defined by T_K, and thus K, is not known or even goes to infinity. However, in this study, it will be relevant to define the control process duration, since the MPC algorithm will be taken back to the offline domain, as further elaborated in subsection 3.3.3. Moreover, in this section, it is assumed that the length of the time interval $[t_k, t_{k+1}]$ is equal to the length of the optimal control discretisation interval $[t_i, t_{i+1}]$ as defined in section 3.2.

The basic principle of MPC is now that every current time interval $[t_k, t_{k+1}]$, an optimal control problem of the form found in (3.1) is solved. This optimal control problem determines the future control input by minimising an objective function that penalises the difference between the predicted future reference signal and the predicted future output of the system. Note that the future reference signal is a prediction and will never be entirely equal to the real future reference signal since it is dependent on environmental and possibly human factors. The length in time of the optimal control problem is called the prediction horizon and is determined by the length of $[t_i, t_{i+1}]$ (and in this case thus $[t_k, t_{k+1}]$) and the number of intervals, N, in the optimal control problem. After solving the optimal control problem of the current time interval $[t_k, t_{k+1}]$, the resulting system state of the next time interval, $x_{t_{k+1}}$, is set as the initial condition for the optimal control problem of the next time interval $[t_{k+1}, t_{k+2}]$. Note that in practice, the initial condition of the system state is often set by a combination of a measured and calculated system state, however, in this study, only the calculated system state will be used. A schematic representation of the MPC algorithm can be seen in Figure 3.1

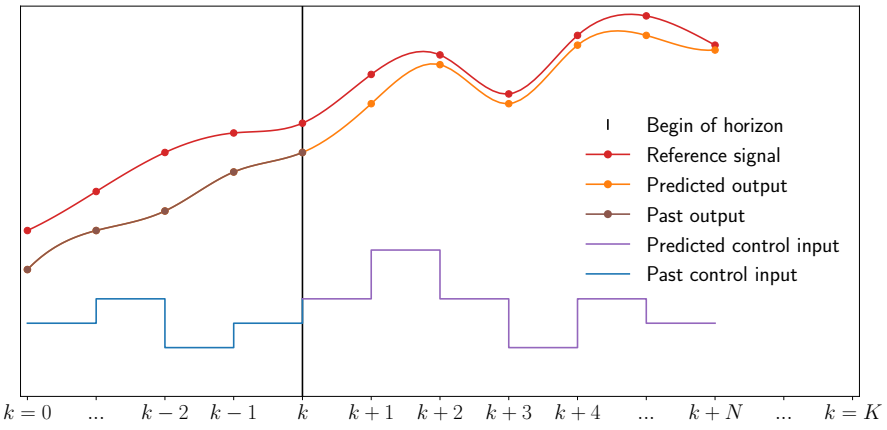


Figure 3.1: Schematic representation of MPC.

3.3.1. Stability

As MPC applied to constrained dynamical systems leads to non-linear control, Mayne et al. [39] pleads for the use of Lyapunov stability theory in order to analyse stability. They note that the value function (in this study also called cost or objective function) can be deployed as a Lyapunov equation for stability analysis of MPC of time-varying, non-linear, constrained and discrete systems. Moreover, they propose the main ingredients necessary for the open-loop optimal control problem used in MPC of non-linear systems to ensure closed-loop stability. These are first, the implementation of a terminal constraint set and secondly, the addition of a terminal cost to the value function. More recently, Rawlings et al. [38] argues that when a terminal constraint set is not necessary, for example when the initial state lies in a subset of the terminal state that is sufficiently small, it is desired to leave out the terminal constraint set in the optimal control problem formulation. This will result in optimal control problems that are easier and thus faster to solve.

3.3.2. Computational challenge

One of the major drawbacks and limiting factors of the real-time MPC implementation is the challenging computational burden that comes with it. Three main steps in the algorithm can be depicted that should be performed every time interval $[t_k, t_{k+1}]$. This time interval determines thus the maximum allowable computation time needed for every iteration of the MPC algorithm. The first step is the prediction of the future reference signal. Secondly and more challenging is the parameterisation of the optimal control problem. The last and most challenging part is solving the optimal control problem.

The computational complexity of the parameterisation and solving of the optimal control problem is mainly dependent on the size of the optimal control problem. This is subsequently determined by the prediction horizon length, expressed in the number of samples, and the complexity of the system, which is dependent on the size of the state, control input and output vectors and the complexity of the relationship between these vectors in the terms of for example non-linearities.

Increasing the time interval $[t_k, t_{k+1}]$ reduces the pressure on the calculations, moreover, decreasing the prediction horizon length decreases the computational burden. However, many dynamical systems have multiple-time-scale dynamics, meaning that both fast and slow dynamics are present in the system. If the time interval is increased, the presence of fast (high-frequency) dynamics in the reference signal and the control input trajectory and thus the system output is decreased. Furthermore, the amount of slow (low-frequency) dynamics present in the control trajectory is captured by the prediction horizon length. Christofides and Daoutidis [40] stress that not taking into account the multi-time-scale properties in the control design can lead to undesired and low-performance system output and even unstable behaviour. Additionally, Geyer [41] shows that an increase in prediction horizon length leads to better performance.

Tan et al. [42] presented a method to reduce the dimension of the optimal control problem and thus the computational complexity of the MPC algorithm while remaining to take into account both the fast and slow dynamics of a system. They proposed a prediction horizon with a non-uniform grid, meaning that in the near future the prediction horizon is discretised with a smaller time interval and in the distant future with a larger time interval. To deploy this method, care should be taken to properly use the reference signal to avoid large discretisation errors. High-frequency components in the reference signal near the end of the prediction horizon, where the discretisation grid is larger, can lead to largely fluctuating values captured by the discretised intervals, which could result in undesired or unstable behaviour. Thus, it might be required to filter high-frequency components out of the reference signal in the distant future, appropriate to the discretisation grid.

Chen et al. [43] propose a method to decrease the computational complexity of the optimal control problem while remaining the original small discretisation grid. This method, called move blocking, restricts constant control input over a number of discretisation intervals that increases over the prediction horizon. This leads to an optimal control problem with fewer decision variables while retaining the same level of sparsity, resulting in lower computational complexity.

3.3.3. Offline model predictive control

As already mentioned earlier, in this study, the MPC algorithm will be deployed in an offline fashion. In this way, the need to execute the steps discussed in subsection 3.3.2 in a time duration equal to the length of interval $[t_k, t_{k+1}]$ is factored out. This simplifies the software implementation and allows the use of interpreter programming languages, while still being to offline mimic and analyse the algorithm performance.

Restating that for the offline case, the end of the control process is known, resulting in a known number of MPC intervals K. The time t_K is then referred to as the *final* time. Additionally, the time indicating the end of the current prediction horizon, t_{k+N}, is referred to as the *terminal* time. The reason why a distinction is made between these points in time is to be able to define a *terminal* state and a *final* state on which different constraints can be imposed. Note that when $K - k \leq N$, the terminal time and final time coincide. Note that when $K - k < N$, the length of the prediction horizon, and thus the length in time of the optimal control problem is changed from N to $K - k$.

Note that for the offline case, no environmental or human influence is present on the reference signal, resulting in perfect knowledge of the reference signal. With this perfect "prediction" of the reference signal for the entire prediction horizon, the best possible control trajectory achievable by the MPC algorithm can be determined for a certain prediction horizon length. By increasing the prediction horizon length to the total length of the control scenario, only one MPC iteration is obtained. This results in a full-length optimal control problem, instead of an offline MPC algorithm. This in combination with the perfect knowledge of the reference signal, results in the optimum control trajectory to reach the desired system output. In the remainder of this study, this optimal control problem is referred to as the "Oracle".

On the contrary, for the offline MPC algorithm, it is still possible to not assume perfect knowledge of the reference signal for the prediction horizon. For example, the assumption can be made that only the reference signal at the current time interval is known, which is normally the case in control processes. The remainder of the horizon can then be filled with a reference signal based on the reference signal at the current time interval. An example of this can be a constant "prediction" for the entire horizon, based on the current time interval reference signal. This is often also often applied for (real-time) MPC implementation studies where no emphasis is put on predicting the reference signal [34, 21, 22].

To summarise, three different definitions are used in the remainder of this study that refers to either MPC with a certain prediction strategy or a full-length optimal control problem. The definitions are presented below.

Oracle Offline MPC algorithm where the prediction horizon length is equal to the entire control scenario. This results in a single MPC iteration and is thus actually an optimal control problem. This in combination with the perfect knowledge of the reference signal will result in the best obtainable control trajectory for the control scenario. This is not feasible in real-time, since perfect knowledge of the reference signal for the entire control scenario is required.

Perfect Offline MPC algorithm that assumes perfect knowledge of the reference signal for the entire prediction horizon. This will result in the best obtainable control trajectory obtained through MPC for a certain horizon length. This prediction strategy is not feasible in real-time, since it will never be possible to perfectly predict the future reference signal for the entire prediction horizon.

Constant Offline MPC algorithm that assumes knowledge of the reference signal for the current time interval only. The remainder of the horizon is then filled with a constant reference signal that is equal to the current time interval. This prediction strategy is feasible in real-time since it will only require knowledge of the reference signal for the current time interval.

3.4. Model predictive control-based motion cueing algorithms

If MPC is applied to motion cueing, the optimal control problem is solved in real-time, and each time step the optimal control problem optimises the simulator control input for the prediction horizon by

minimising a cost function. The reference motion used by the MPC MCA, if applied to flight simulation, will be the specific forces, angular rates and possibly angular accelerations as experienced by the pilot in the aircraft. The output specific forces and angular motion of the simulator are a result of the simulator state and control input. In this study, the reference aircraft motion and the output motion of the simulator are called reference inertial signal and output inertial signal, respectively.

3.4.1. Reference motion prediction

In flight simulation, specific forces and angular motion of the aircraft are obtained in real-time from an aircraft model and are dependent on the control input of the pilot. Current flight simulators, equipped with a CW MCA, only require the current output of the aerodynamic model. On the contrary, MPC-based MCAs require a reference motion for the entire prediction horizon, which will be dependent on the future control behaviour of the pilot. In this study, the focus will not be on predicting the pilot control behaviour. Luckily, more prediction strategies are possible without requiring the prediction of the future control input of the pilot. An example of this, as already mentioned in subsection 3.3.3, could be a constant prediction, equal to the current time step, over the entire prediction horizon.

3.4.2. Sensory models

Sensory models, also often called vestibular models, are models that try to simulate the human otolith and semicircular canals. The idea of these models is to obtain the specific forces and angular rates as perceived by the pilot from the actual specific forces and angular rates applied to the pilot's head. These vestibular models can be linear transfer functions as well as non-linear functions. In the coming sections, the presented MPC-based MCAs often include a sensory model. However, during this study, sensory models will not be included. The reason for this is that in this study the effect of applying a MPC-based MCA compared to CW MCA is investigated. Therefore, assumptions made in sensory models that would be included in the MPC algorithm and not in the CW algorithm, could affect the outcome of the offline analysis presented in Appendix 5 and the experiment elaborated in Appendix 6.

3.4.3. Early implementations

The increasing popularity of the use of MPC-based MCA originated from the use of the algorithm in driving simulators. The first implementation of a real-time MPC-based MCA was developed by Dagdelen et al. [17], which was implemented in the RENAULT ULTIMATE driving simulator for a 2-DOF problem for simplicity. This MPC-based MCA was designed with the simulator motion space constraints defined in Cartesian coordinates and was completely linear. Additionally, human threshold constraints were implemented for the deceleration artefacts that arise when the physical limits of the simulator were approached. The MPC-based MCA showed more usage of the available motion system during a five-minute drive. Moreover, a first experimental test showed that drivers subjectively preferred the algorithm over a CW reference.

Garret and Best [18] were the first to introduce a real-time MPC-based MCA with motion space constraints based on the actuator length limits of the hexapod motion platform instead of constraints defined in Cartesian coordinates. However, deriving the actuator lengths from the motion platform state, so-called inverse kinematics, introduces time-varying non-linearity to the system, leading to an increase in computational complexity. Therefore, Garret and Best assumed that the velocity variation with the position of the actuators was constant over the prediction horizon, resulting in a Linear Time-invariant (LTI) model. To further decrease the computational burden, the DOFs were split up into two tilt-coordinated pairs and two single-DOF channels. Furthermore, vestibular models have been included in the MPC formulation. Offline analysis of this algorithm showed better tracking of the "perceived" vehicle motion compared to a CW MCA, resulting in higher objective fidelity. Furthermore, during driver-in-the-loop experiments, the algorithm was rated better than the CW reference.

3.4.4. State-of-the-art

The state-of-the-art in MPC-based MCAs is in this report defined as non-linear and real-time feasible algorithms. Studies using this type of MCA are investigated on among others the prediction horizon length, sample time, prediction strategy used and cost function definition.

Katliar et al. [34, 35] were the first to present a real-time non-linear time-varying MPC-based MCA implemented for the cable robot simulator and the 8-DOF Robot simulator. In this algorithm, a sample time of 0.05 s is used, resulting in feasible prediction horizon lengths up to 3 s. Furthermore, the cost function includes terms on the simulator state and terminal state, the control input and the inertial signal incongruence. Katliar et al. [44] also studied the influence of the prediction horizon length on the motion cueing fidelity during offline analysis.

Grottoli et al. [20] executed an objective evaluation of an MPC-based MCA. They compared the outcome of an MPC-based MCA with a prediction strategy with perfect knowledge of vehicle trajectory and a prediction strategy with a constant prediction with the reference vehicle motion. During this study, a prediction horizon of 5 s was used in combination with a sample time of 0.1 s. The cost function included terms on the simulator state and terminal state, the control input and the difference between the reference inertial signal and the output inertial signal.

Van der Ploeg et al. [21] performed an offline sensitivity analysis on the lateral specific force error weight and the roll rate error weight, followed by an experiment. They discovered a very high correlation between the continuous perceived motion cueing fidelity rating of the experiment participants and the RMSE of the lateral specific force and roll rate cues. In this study, a short prediction horizon of 2 s was used in combination with a sample time of 0.1 s. Moreover, terms on the simulator state and terminal state, the control input and the difference between the reference inertial signal and the output inertial signal were included in the cost function.

Cleij et al. [22] performed a human-in-the-loop simulator experiment to compare two MPC-based MCAs with different cost function weight settings. The baseline weight setting was equal to the weights used in [44]. The newly proposed weights were derived with the help of a Motion Incongruence Rating (MIR) model from Cleij [7]. The cost function contained terms penalising the control input, state, terminal state, and the difference between the reference and output inertial signal. A constant prediction strategy was used and a prediction horizon length of two seconds was chosen. No sensory model was used in this study. Contrary to expectations, the majority of the participants preferred the baseline weights over the optimised weights.

Lamprecht et al. [23] recently compared a MPC-based MCA with a classical filter-based MCA. They performed a driver-in-the-loop experiment in the Daimler driving simulator. In real-time, with a sample time of 4 ms, they were able to predict the driver control input to obtain a reference inertial signal for a prediction horizon of 4 s. Moreover, during this time interval, the reference inertial signal was passed through a sensory model and the simulator control input was determined with the MPC algorithm. No terminal state cost was included in the cost function. The subjective ratings obtained during the experiment resulted in a high preference for the MPC-based MCA compared to the classical filter-based MCA.

3.4.5. Particular implementations

Qazani et al. [45] implemented a decoupled MPC-based MCA, meaning that two MPC-based MCA were used in parallel. These MPC-based MCAs were used to cue the motion sensed human semicircular canals and the human otolith organs separately. The algorithms were both linear and included linear vestibular models of the otolith and semicircular canals. Moreover, the output of the separate algorithms was fed back into each other. Offline analysis showed that the decoupled algorithm, compared to the equivalent coupled algorithm, resulted in better RMSE and PCC values and thus higher objective fidelity. Qazani et al. [46, 47] also implemented a time-varying MPC-based MCA with constraints on the joint limitations and, moreover, a linear time-varying algorithm with linearised inverse kinematics. In both studies, vestibular models were used. Furthermore, a sample time of 0.01 s and a prediction horizon of 150 samples were used. In these studies, only offline analysis was performed.

Bukal et al. [48] proposed an adaptive real-time MPC-based MCA with the motion gains included as an optimisation variable. The algorithm included linearised human perception and actuator models and was applied to driving as well as aircraft simulation. Offline analysis showed that "good" motion cueing was achievable by the algorithms provided that accurate reference motion was available. Kushro et al. [36] proposed a different version of an adaptive MPC-based MCA with the error weights increasing when reaching the bounds of the optimisation constraints. This algorithm was non-linear due to the actuator-based constraints and included linear vestibular models. Again, this algorithm was only analysed offline while using a sample time of 0.01 s and a relatively short prediction horizon of 50 samples. The results of this analysis showed fewer false and missing cues compared to a CW MCA and better reference motion tracking.

3.5. Chapter conclusions

RQ 2. *How do MPC-based MCAs work?*

If MPC is used for motion cueing, an optimal control problem is solved in real-time by optimising an objective function to find the control input of the simulator. To do this, reference vehicle motion as a result of the pilot/driver control behaviour is used, which are specific forces and angular motion.

2a. *What is optimal control?*

Optimal control is a type of optimisation problem that tries to find the optimal control input trajectory to match the system's output to a reference signal.

2b. *How can a continuous-time optimal control problem be parameterised into a numerical finite-dimensional optimal control problem?*

A continuous-time optimal control problem can be parameterised into a numerical finite-dimensional problem with the help of direct parameterisation techniques. Examples of this are single shooting, multiple shooting and collocation. Multiple shooting and collocation are preferred for MPC since these allow for parallel solving, decreasing the computation time.

2c. *What is MPC?*

MPC is a control strategy where every time interval an optimal control problem is solved that results in the system control input for the next time interval.

2d. *How is MPC used for motion cueing?*

The use of MPC for motion cueing is mainly applied for driving simulation applications and is limited to the use in research. Recently, fast real-time implementations (sample time of 4 ms) including driver prediction and sensory models are shown to be feasible for prediction horizon lengths of 4 s.

Chapter 4

Upset Prevention and Recovery Training simulation

In this chapter, the literature review for answering the fourth research question posed in section 1.2 is presented.

How are FSTDs currently used for UPRT?

The objective of this chapter is to identify what requirements and shortcomings exist for UPRT in FSTDs. Furthermore, training scenarios relevant to UPRT and useful for the analysis in Appendix 5 and the proposed experiment in Appendix 6 are investigated.

First, the requirements for the use of FSTDs for UPRT are identified in section 4.1. Thereafter, in section 4.2, the definition of the FSTD Training Envelope is given and the effect of extended aerodynamic models is investigated. Relevant training sequences and scenarios that are used in UPRT are explored in section 4.3. Subsequently, in section 4.4, the methods of how the training scenarios are used for and implemented into UPRT. Thereafter, the focus will be shifted towards motion cueing for UPRT in section 4.5, where first the importance and difficulties are discussed, followed by an overview of previous studies about motion cueing for UPRT.

4.1. Upset and stall training in flight simulators

The Simulation of Upset Recovery in Aviation (SUPRA) project by Groen et al. [49] was a project funded by the European Union with the goal of enhancing UPRT in FSTDs. They pointed out the disadvantages of in-aircraft UPRT, being safety, costs and time. The implementation of UPRT into the pilot training curriculum and these disadvantages of in-aircraft UPRT, led to FSTDs being a substantial training platform. However, the usefulness of simulator UPRT is dependent on multiple factors. Advani and Field [50] already pointed out the requirements for FSTDs a decade ago, being a representative aerodynamic model, a suitable Instructor Operator Station (IOS), relevant training scenarios, pilot control authority, stall buffet, motion feedback and upset forcing functions. While they note that most of these requirements were satisfied with minor changes to (existing) FSTDs, for example, stall buffet motion implemented as a special effect and the use of active stick shakers, the representative aerodynamic model and motion feedback requirements are more difficult to meet. This has led to research being done into the influence of aerodynamic models and motion cues during UPRT in FSTDs, which will be discussed in the coming sections [51, 52, 53, 54, 55, 56, 49, 16, 57, 58, 59, 11, 60, 61].

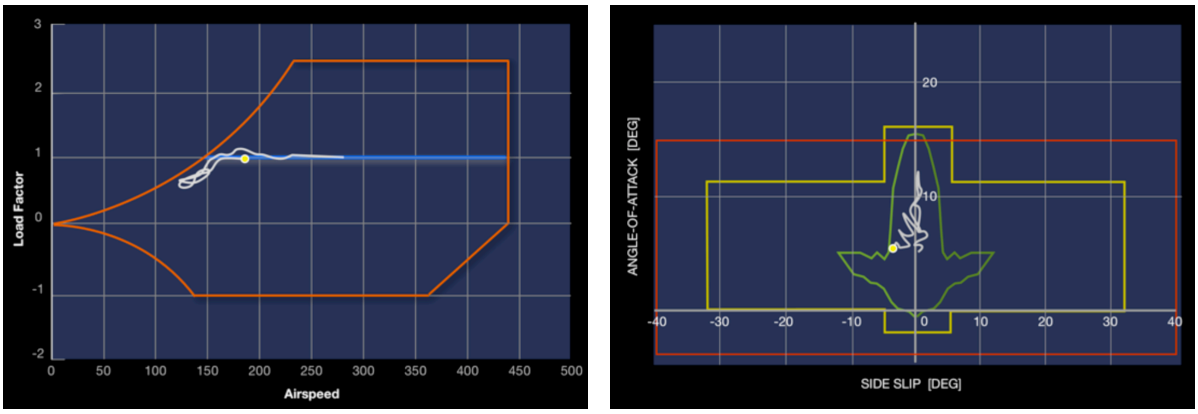
4.2. FSTD Training Envelope

Although Advani, Field and Schroeder [50, 4] stress that most upsets occur inside the normal flight envelope, recovery may lead to high load factors and large aerodynamic angles. The latter causes the

problem that the pilot, during upset recovery in a FSTD, may find him- or herself on or over the edges of the normal flight envelope. Furthermore, training to recover from aircraft stalls will always occur outside the normal flight envelope. Simulating these trajectories with an unsuitable and not-validated aircraft model can lead to unrealistic aircraft behaviour and possibly a negative transfer of training. The flight envelope can be defined by the load factor, the airspeed, the angle of attack, and the angle of sideslip and is dependent on aircraft configuration and altitude. An example flight envelope from Advani and Schroeder [4] can be seen in Figure 4.1.

For this reason, EASA published FSTD certification specification relating to the fidelity of extended aerodynamic models used for UPRT [62]. The FSTD Validation Envelope, as defined by EASA and originally by the ICAO, consists of three parts. The first part is the *Flight-test-validated region* of the aerodynamic model, which will have the highest obtainable fidelity. The second part is the *Wind tunnel and/or analytical region*, which will have lower but reasonable fidelity compared to the first region. The third part is the *Extrapolated region*, which has the lowest fidelity and within this region, there is low confidence that the model will behave as the real aircraft. The various confidence regions can be seen in Figure 4.1b.

A second definition, the FSTD Training Envelope, is defined by EASA [62] as the first two parts of the FSTD Validation Envelope, that is the *Flight-test-validated region* and the *Wind tunnel and/or analytical region*, and should thus not contain extrapolated data. For UPRT in FSTDs, EASA obliges to stay within this envelope to avoid negative transfer of training.



(a) Example V-n diagram.

(b) α-β diagram with the three aerodynamic model confidence regions.

Figure 4.1: V-n and α-β diagram with the various aerodynamic model confidence regions presents for the latter, from Advani and Schroeder [4].

4.2.1. Effect of extended aerodynamic models

During the already earlier mentioned SUPRA project, an extended aerodynamic model for the stall/post-stall flight envelope of a generic large category aircraft was developed with Computational fluid dynamics (CFD) and wind tunnel data and validated by a pilot-in-the-loop experiment [57]. The evaluation of the model was performed inside and outside the normal flight envelope. For both evaluations, guided free flight was performed, where the pilots were required to fly a set of manoeuvres, as well as scenario-based upsets, where only upset recovery was required. The model was evaluated by the pilots as "acceptable". Consistent comments provided were the model being sensitive in roll and the little phugoid/speed stability observable for normal flight manoeuvres. For the manoeuvres outside the normal flight envelope, the very realistic stall buffet and lateral-directional instability were praised and the nose-down unloading tendency was criticised.

Schroeder et al. [58] performed a pilot-in-the-loop experiment in a certified FSTD assessing the influence on the transfer of training of applying different Boeing 737 (stall) models for UPRT. The key point of this study was to investigate to what extent stall models are required to meet civil transport UPRT objectives. Four different models were used in this study, each created with different resources resulting in varying fidelity. The first model was a 737-800 FSTD model derived from a Boeing data package with

added stall buffet, which was validated until stick-shaker activation. The second model was developed by Bihrle Applied Research using CFD, scaled wind tunnel data and test pilot input. The third model was developed by Boeing using 737-800 flight test data from hundreds of fully developed stalls and turning flight stalls. Unpredictable roll-of was ensured by adding randomised asymmetric components to this model. The fourth and final model was equal to the first model with in addition only the randomised asymmetric components of the third model. The models were first assessed by nine test pilots, they concluded that the stall buffet was less present in the simulation compared to the aircraft, even after doubling the buffet response. After that, 45 airline pilots performed the experiment for only the first three models. Although no specific preference for one of the models used in the experiment was given by the airline pilots, differences in roll angles (smaller roll angles for the first model) and required recovery times (larger for the second model) were discovered. The airline pilots were all able to train to proficiency, regardless of the model used.

A very similar study was performed by Grant et al. [11]. The main differences were the use of a T-tailed turbo-prop with a stick pusher model, the control over the fidelity of the different models used in the experiment by the experimenters themselves, and the use of a Flight Research Simulator (FRS). In this study, three models were used. The first model was created by Grant et al. [59] from certification flight test data and wind-tunnel data, the second and third models were other versions of the first model with exaggerated and milder stall characteristics, respectively. The outcome of the experiment was in line with the study by Schroeder et al. [58], and the fidelity of the model had an insignificant effect on recovery performance. The pilots indicated being able to recover from a real in-aircraft stall event, independent of the model used for training. The pilots had difficulties distinguishing between the three models, except for the degree of roll-off during the stall. Although not mentioned in the paper, the outcome of the experiment could be explained by questioning whether the classical MCA used in this study was capable to translate the different models into good enough and distinguishable motion cues.

4.3. Training sequences and scenarios

In the Airplane Upset Prevention & Recovery Training Aid (AUPRTA) published by International Civil Aviation Organization (ICAO) [63] recommended training sequences by major aircraft Original Equipment Manufacturers (OEM) are presented and divided into seven groups, being:

1. Aircraft handling characteristics;
2. Upset recognition and recovery;
3. Stall;
4. Environmental factors;
5. Wake vortex;
6. Mechanical/system-induced;
7. Pilot factors.

Since not all these training sequences will be relevant or helpful to identify the importance, difficulties, and effect of motion cueing or the usefulness of a new MCA, a selection of training sequences has been made for this study. This selection includes the second and third sequences, being *Upset recognition and recovery* and *Stall*, respectively. In the coming sections, only training scenarios relevant to these training sequences will be presented. Furthermore, training scenarios related to the third and fourth training sequences, *Environmental factors* and *Mechanical/system-induced*, will only be included when used as a forcing function, for example, wind shear to put the aircraft in an upset of the second training sequence.

Training scenarios that are assigned by the ICAO to the *Upset recognition and recovery* training sequences are nose high, nose low, and spiral dive upsets. For the *Stall* training sequence, ICAO recommends a clean configuration stall, take-off stall with partial flaps, landing configuration stall, and stick pusher demonstration. IATA agrees with these recommendations and refers to the AUPRTA training sequences and scenarios in their UPRT manual [64]. In this manual, IATA stresses that these training sequences should be used in combination with the recommended UPRT of the respective aircraft OEM.

4.3.1. Scenarios required by regulations and for certification

Since the ED Decision 2019/005/R [65], EASA distinguishes between basic UPRT, advanced UPRT and type-specific UPRT. The main differences between the basic and advanced courses are that for the advanced course it is mandatory that the training exercises are executed in-aircraft and that additional instructor qualifications are required. Type-specific UPRT is dependent on the respective aircraft OEM recommendations and is often performed in FSTDs. Scenarios included in type-specific UPRT are mostly in accordance with the basic and advanced UPRT scenarios, however, then performed with the respective aircraft (FSTD). All these three UPRT courses are required for the MPL and thus Airline Transport Pilot License (ATPL) licenses. For the CPL license, only basic UPRT and type-specific (if applicable) courses are required [65]. Scenarios that are relevant to training sequences *Upset recognition and recovery* and *Stall* as defined by ICAO, that are included in the basic UPRT course are nose-high attitudes at various bank angles; nose-low attitudes at various bank angles including spiral dive; and stall events at take-off, clean, and landing configurations. Training scenarios that are included in the mandatory in-aircraft advanced UPRT course are nose high, nose low, spiral dive, stall event, and incipient spin upsets. Both these basic and advanced UPRT course descriptions are given in Amendment 7 of the EASA Part-FCL [15].

For the certification of FSTDs, EASA composed a set of training scenarios for which the FSTD should be evaluated. These scenarios are specified in the CS-FSTD(A) [62] and include a nose-high wings level aeroplane upset, a nose-low aeroplane upset, and a high bank angle aeroplane upset. Furthermore, either a stall approach and entry at wings level event or a stall approach and entry in turning flight of at least 25° bank angle event should be evaluated.

4.3.2. Scenarios included in previous studies

Advani and Field [50] derived an advisory set of important scenarios containing unusual attitudes, pre-stall, accelerated stall, nose low stall, high altitude stall, stall with yaw present, stall break, incipient spin, and developed spin. In a later study, Advani and Schroeder [4] advised additional and more specific upset scenarios to be implemented in UPRT, being pitch trim runaway during take-off, and recovery from overbank.

Fucke et al. [5], as part of the SUPRA project, categorised the UPRT scenarios into 3 categories, being unusual attitudes, approach to stall and stall scenarios, with each category containing a subset of about 4 scenarios that were analysed individually.

In an early study of Chung [51], a large roll upset of more than 100 degrees and a large pitch upset of over 50 degrees were evaluated. Liu and Grant [54] used five scenarios in their study that were derived from actual accidents, a stall with full thrust and full nose-up trim during approach, a milder stall during approach, a large roll upset, a large attitude upset due to jammed rudder, and a nose-low upset due to wind shear. Ko and Grant [55] used identical scenarios in their study in addition to a new pilot-induced stall scenario. Schroeder et al. [58] and Zaal [61] focused on stall scenarios in their experiments, whereas Schroeder et al. [58] used stall scenarios at low, medium and high altitudes.

4.3.3. Summary and definition of upset and stall scenarios

The scenarios mentioned in subsection 4.3.1 and subsection 4.3.2 are summarised with their relation to EASA regulations, FSTD certification, and previous research in Table 4.1. From this table, it is possible to derive the most important and relevant scenarios for simulator UPRT and which thus should be used for the analysis in Appendix 5 and the experiment of which the design is presented in Appendix 6.

The incipient spin scenario that is required for the advanced UPRT course and not for the basic UPRT course would not be very relevant since it is required by EASA [15] that advanced UPRT is performed in-aircraft. Moreover, this scenario is not required by EASA [62] for FSTD certification. The same goes for the developed spin scenario, which is not required by regulations or for certification. Both these scenarios are recommended by Advani and Field [50], however, they do not specify if these scenarios should be included in simulator UPRT.

The spiral dive scenario is not required for FSTD certification and is not included in any of the research presented earlier. Since it is required for the basic and advanced UPRT courses, it is considered in this study as somewhat relevant for simulator UPRT, but less than the scenarios treated next.

All the remaining scenarios (nose high, nose low, high bank, stall, accelerated stall) are required by EASA [15, 62] for the basic and advanced UPRT courses and for FSTD certification. Additionally, these scenarios are considered useful in various studies [5, 54, 50, 55, 58, 4, 61]. These scenarios are, therefore, considered very relevant and useful for simulator UPRT and thus for the analysis in Appendix 5 and the experiment presented in Appendix 6.

Table 4.1: Upset and stall scenarios and their relation to EASA regulations, FSTD certification and previous research.

Scenario	**Specified in regulations**	**Used for FSTD certification**	**Included in previous research**
Nose high	Basic & Advanced UPRT [15]	Yes [62]	Yes [51, 5, 50, 4]
Nose low	Basic & Advanced UPRT [15]	Yes [62]	Yes [51, 5, 54, 50, 55, 4]
High bank	Basic & Advanced UPRT [15]	Yes [62]	Yes [51, 5, 54, 50, 55, 4]
Spiral dive	Basic & Advanced UPRT [15]	No	No
Stall	Basic & Advanced UPRT [15]	Yes [62]	Yes [5, 54, 50, 55, 58, 4, 61]
Accelerated stall	Basic & Advanced UPRT [15]	Yes [62]	Yes [5, 50, 4]
Incipient spin	Advanced UPRT [15]	No	Yes [50]
Developed spin	Not specified	No	Yes [50]

The definitions of the scenarios mentioned in subsection 4.3.1, subsection 4.3.2 and Table 4.1 are presented below.

Nose high - Fucke et al. [5] define a nose high upset as a situation where the aircraft pitch attitude is around 30° nose up. This can be in combination with wings level, or with various bank angles between 45° and 90°.

Nose low - Fucke et al. [5] define a nose low upset as a situation where the aircraft pitch attitude is around 25° nose down. This can be in combination with wings level, with various bank angles between 45° and 90°, or with inverted bank angles larger than 90°.

High bank - Fucke et al. [5] define a high bank upset as a situation where the aircraft bank angle is between 45° and 90° in combination with a pitch angle around 0°.

Spiral dive - EASA [15] defines a spiral dive upset as a high bank angle descending turn with the aircraft in a nose-down attitude resulting in rapidly increasing airspeed and load factor.

Stall - EASA [15] defines a stall as the loss of lift caused by exceeding the aircraft's critical angle of attack. A stall can exist at any attitude and airspeed and may be recognised during the approach to stall by a stick pusher, stick shaker, stall buffeting, lack of pitch authority and/or roll control, and inability to arrest the descent rate. Stall scenarios include stall during take-off, clean configuration (high altitude), and approach.

Accelerated stall - EASA [62] defines an accelerated stall as the loss of lift caused by exceeding the aircraft's critical angle of attack during a coordinated turn of at least 25° due to the increase of load factor caused by applying a positive pitch-up moment. Due to the increased load factor, an accelerated stall occurs at an increased airspeed compared to a normal stall.

Incipient spin - EASA [15] defines an incipient spin as a transient flight condition in the post-stall regime where an initial, uncommanded roll in excess of 45° has resulted from yaw asymmetry, for example in an uncoordinated turn, during a stall and which, if recovery action is not taken, will lead rapidly to a developing spin. Prompt recovery during this incipient spin stage will normally result in an overall heading change from pre-stall conditions of not more than 180°.

Developed spin - EASA [15] defines a developed spin as a situation where the aircraft is in the post-stall regime and has achieved approximately a constant pitch attitude, yaw rate and roll rate on a descending flight path. In the transition from a stall with significant and persistent yaw to attaining a developed spin, the aircraft is likely to have rolled through at least 540°.

4.4. Training methods

There are two main methods of how the scenarios presented in section 4.3 are used for UPRT. The first method is Scenario-Based Training (SBT) and focuses on the prevention of upset scenarios. SBT makes use of dynamical scenarios, starting in normal flight, and where the upset is often triggered by external (environmental, mechanical) forces which only result in a full upset when incorrect or late measures are taken by the pilot. The training objective as defined by IATA [64] is that pilots should recognise and manage the threats and take appropriate and timely measures, using all pilot competencies.

Maneuver-based Training (MBT) is the second method and serves only upset recovery training. During this type of training, it is assumed that prevention was not successful and that the aircraft is in an upset. The training, therefore, starts after establishing the aircraft upset. IATA [64] defines the training objective of MBT for pilots to analyse the current situation, give initial control inputs, recover from the upset, and return to the original normal, stabilised flight path. Pilot competencies used during this type of training are mainly situation awareness, flight path management, manual control and multi-crew aircraft communication.

Both these methods can be used in FSTD training as well as in-aircraft training. ICAO [63] stresses that when upset training is performed in FSTDs, it is of importance that the instructor has knowledge of the FSTD deficiencies during UPRT and communicates these deficiencies towards the trainees. Basic and advanced UPRT will mainly be performed in-aircraft, in contrast to type-specific training, which will mostly be in FSTDs. During type-specific training, stall training is often treated separately from upset training and is always performed as SBT. This includes the three flight phases, normal flight, approach to stall, and fully developed stall. The focus during this training in FSTDs is on prevention and often a fully developed stall is only treated for demonstration purposes.

4.5. Motion cueing for upset scenarios

According to Ko and Grant [55], current FSTDs' motion systems are commonly driven by a CW MCA. As already explained in section 2.7, this type of MCA is a sub-optimal motion filter that is limited in reproducing the motion cues experienced in real aircraft. This raises the question of to what extent the generated motion cues are useful for UPRT, or flight simulation in general.

De Winter et al. [66] performed a meta-analysis of 24 transfer of training experiments, assessing the effectiveness of whole-body flight simulator motion. This analysis showed an overall preference for motion. The effect of motion was stronger for novice pilots compared to intermediate pilots and not present for expert pilots. Furthermore, for control tasks with external disturbances and control tasks with low or unstable dynamics, the motion had a positive effect on the transfer of training. In a later study, Zaal and Mobertz [67] used two hexapod motion conditions to compare the effect of motion on training for a multi-axis control task. Although neither the baseline nor optimised motion conditions could be unambiguously chosen as the best motion condition for this training task, the optimised condition resulted in better disturbance-rejection performance. If these results are projected onto UPRT training, a reserved conclusion can be made that motion can be effective for upsets that can be compared to disturbance tasks, and stalls, which contain unstable aircraft dynamics.

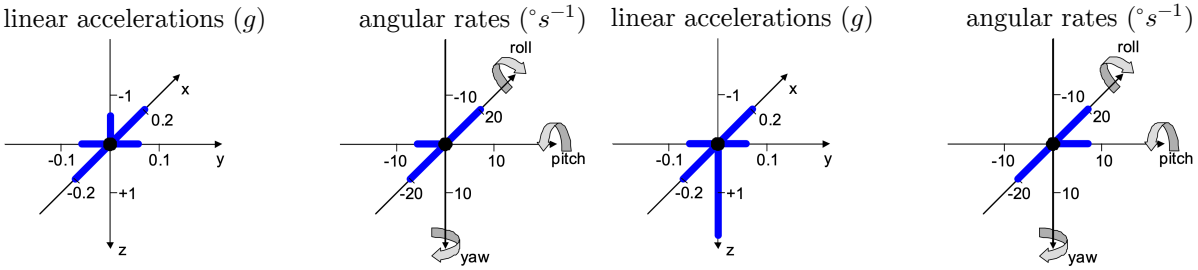
4.5.1. Challenges of upset motion cueing

Chung [51] was one of the first to publish a preliminary investigation about motion cueing for UPRT in FSTDs. He questioned whether a positive transfer of training was possible in these devices, with or without an active motion system. While not answering his posed question, he did identify motion cueing limitations for classical MCAs, related to large-amplitude UPRT motion scenarios. He identified two key problems, the first one being the cueing of sustained specific forces, taking up almost all motion space and leaving no room for other cues. Secondly, the introduction of lateral specific force false cues during large roll motions, resulting in a trade-off between angular and translational motion cues.

According to Zaichik et al. [52], the high-frequency washout of a CW MCA resulted in false opposite-sign accelerations and angular rates during large-amplitude motions. Furthermore, in line with Chung [51],

they pointed out that large angular motions introduce false specific force cues.

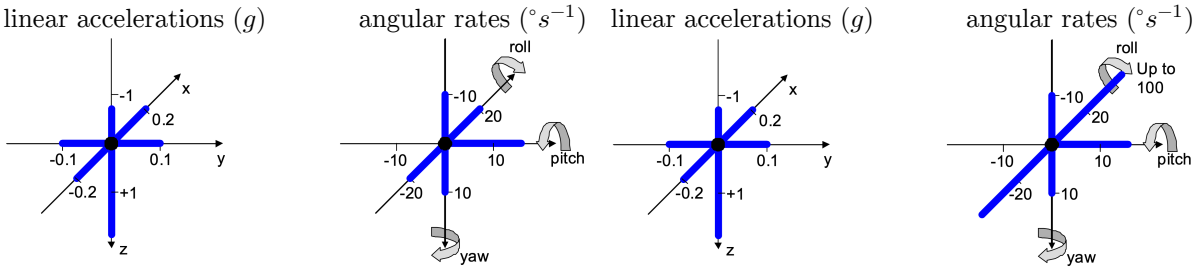
Fucke et al. [5] estimated the magnitude of the aircraft accelerations and angular rates for unusual attitude and stall upsets based on expert analysis and experience with in-flight recovery. In Figure 4.2, the estimated aircraft accelerations and angular rates for nose high and nose low upsets with various bank angles are presented for both approach and recovery. The estimated linear accelerations and angular rates for approach and recovery of stalls and accelerated stalls are displayed in Figure 4.3.



(a) Nose-high, including wings-level and bank angles between 45° and 90°.

(b) Nose low, including bank angles between 45° and 90° and inverted bank angles larger than 90°.

Figure 4.2: Estimated aircraft accelerations and angular rates for nose high and nose low upsets with various bank angles for both approach and recovery, from Fucke et al. [5].



(a) Stall

(b) Accelerated stall

Figure 4.3: Estimated aircraft accelerations and angular rates for stalls and accelerated stalls for both approach and recovery, from Fucke et al. [5].

Thus, the challenges of upset and stall motion cueing arise mainly due to the, compared to normal flight, larger amplitude and longer duration aircraft accelerations, potentially leading to larger and sustained specific forces and angular rates to be cued. The first challenge is to leave sufficient motion space for other cues when sustained specific forces are cued. The second challenge is the minimisation of specific force false cues when large-amplitude angular motion is cued. The third and final challenge is to minimise the false cues that appear due to high-frequency washout of a (CW) MCA.

4.5.2. Motion cueing algorithms used in previous studies

Zaichik et al. [56] optimised a CW MCA with the focus on g-loading (specific forces), resulting in an MCA that was positively assessed by pilots and which contributed to better pilot control activity. Furthermore, Zaichik et al. [60] analysed available motion fidelity criteria for UPRT motion cueing. They concluded that only for the upset approach, for example, approach to stall, the Sinacori-Schroeder criteria, based on phase and gain distortions, could be used.

As part of the SUPRA project, Field et al. [16] developed an extension to the CW MCA. This extension was a pre-positioning module, bringing the simulator to a not-neutral position before a large amplitude motion was expected, resulting in more available actuator travel for the simulator. Although there was no difference in pilot performance, the adjusted MCA was subjectively rated better by the pilots and objective analysis showed improved matching with onset cues. Furthermore, no negative effect on the pilot control behaviour was discovered during this experiment compared to the same training scenario in fixed-base simulation.

Ko and Grant [55] were the first to exploit an adaptive MCA for the use in UPRT. They conducted a pilot-in-the-loop experiment and concluded that false lateral specific force cues should be minimised, resulting in less angular motion. However, when the aircraft became unstable, no preference was seen due to the angular motion being a helpful cue for controlling the aircraft. Zaal [61] performed six experiments with an MCA similar to Ko and Grant. He concluded that, due to this enhanced MCA, pilot performance was increased. This included lower maximum roll angles, fewer stick shaker activations, higher minimum load factor and maximum calibrated airspeed. Although improvement in motion cueing fidelity was present, Ko and Grant and Zaal both pointed out that the UPRT motion cueing difficulties depicted by Chung were still present and that further research is required.

Wu et al. [68, 69] applied an MPC-based MCA to an UPRT motion scenario with as goal to improve the motion cueing fidelity. The results in the paper show better tracking of the reference signal compared to a classical MCA. However, only a few results are presented and little detail about the MPC formulation is given in these papers, which makes these papers less useful for this study.

4.6. Chapter conclusions

The conclusion of this chapter is provided by giving the answer to the third research question and its sub-questions and is presented below.

RQ 3. *How are FSTDs currently used for UPRT?*

FSTDs are used for generic as well as type-specific UPRT. Both upset and stall scenarios are performed in FSTDs. The various training scenarios can be implemented by two training methods, being Maneuver-based Training (MBT), mainly for recovery skills and Scenario-Based Training (SBT) for training upset prevention and training to take adequate and appropriate measures. Upset scenarios are used for MBT and SBT and stall scenarios are solely implemented as SBT.

3a. *To what extent are extended aerodynamic models required for UPRT in FSTDs?*

Extended aerodynamic models are required by regulations as specified by the EASA. Three aerodynamic model confidence regions are defined by EASA: the *Flight-test-validated region*, the *Wind tunnel and/or analytical region*, and the *Extrapolated region*. The first two regions define the FSTD training envelope. Previous studies showed that the effect of extended aerodynamic models compared to normal aerodynamic models is minimal.

3b. *What training scenarios should be simulated for UPRT?*

Two types of training scenarios can be defined, being unusual attitudes and stalls. Unusual attitudes include scenarios such as nose high, nose low, and high bank and more severe scenarios such as a spiral dive. Stall scenarios include normal stalls, accelerated stalls, and incipient spin as a result of a stall in an uncoordinated turn that eventually can develop into a full spin.

3c. *What are the difficulties of motion cueing for UPRT?*

Aircraft motion during UPRT includes accelerations and angular rates with larger amplitudes compared to normal flight. Additionally, the duration of these accelerations and angular rates can be longer, resulting in large attitude angles and sustained g-loading.

3d. *What are the limitations and shortcomings of current MCAs for UPRT?*

Current MCAs, which are mostly CW MCAs, introduce false cues during the wash-out phase of cueing large amplitudes. Furthermore, the cueing of sustained specific forces resulted in no motion space left for other cues. Finally, false lateral cues arise when a large rolling motion is cued.

Chapter 5

Evaluation of objective motion cueing fidelity during stall simulation

In this chapter, the method, results, and recommendations for answering the fourth research question posed in section 1.2 are presented.

> *How can the implementation of an MPC-based MCA during UPRT affect the **objective** motion cueing fidelity with respect to the current SIMONA Research Simulator (SRS) CW MCA?*

The objective of the analysis presented in this chapter is to get familiarised with the MPC-based MCA. Furthermore, the behaviour of the algorithm is identified by answering the sub-questions of the research question presented above. The outcome of the analysis will be used to derive substantiated preliminary experiment conditions that are included in the experiment plan, presented in Appendix 6.

The chapter starts off with the presentation of the motion scenario used for the analysis. Subsequently, the software used for this analysis is elaborated in section 5.2. In section 5.3 and section 5.4, the mathematical definitions of the optimal control and MPC formulations used are presented. Thereafter, the constraints are validated in section 5.5. The behaviour of the algorithms is presented in section 5.6, which is discussed in section 5.7. Finally, preliminary recommendations and conclusions are included in section 5.8 and section 5.9, respectively.

5.1. Citation stall scenario

For the analysis presented in this chapter, a simulated stall upset of a Cessna Citation-II is used. The aerodynamic stall model used for this simulation is developed by the stall task force group of the Delft University of Technology and is based on real flight test data obtained with the TU Delft PH-LAB laboratory aircraft. The simulation resulting in the stall scenario depicted in Figure 5.1 was created with the help of an autopilot developed by Smets et al. [70] which was later also used by Imbrechts et al. [10].

During the upset phase of the stall, the altitude hold mode controller and bank angle controller were both activated. Furthermore, a thrust setting was used that resulted in a deceleration of about one knot per second. When the angle of attack reached an angle of 16.05°, the upset recovery was triggered. During this phase, the altitude hold mode was disabled and the reference pitch angle was set to $-0.5°$. The reference pitch angle was changed to 10° after the threshold airspeed was reached, resulting at the end of the recovery phase. The resulting symmetric stall scenario only has motion in three DOFs, the specific force in x and z direction and angular motion around the y axis.

To make this motion scenario more useful for analysis, data, starting at zero and at $-9.81\,\mathrm{m\,s^{-2}}$ for the specific force in z direction, is appended at the beginning of the stall scenario with a slope of zero for the first second and ending at the first original data point of the respective DOF. Additionally,

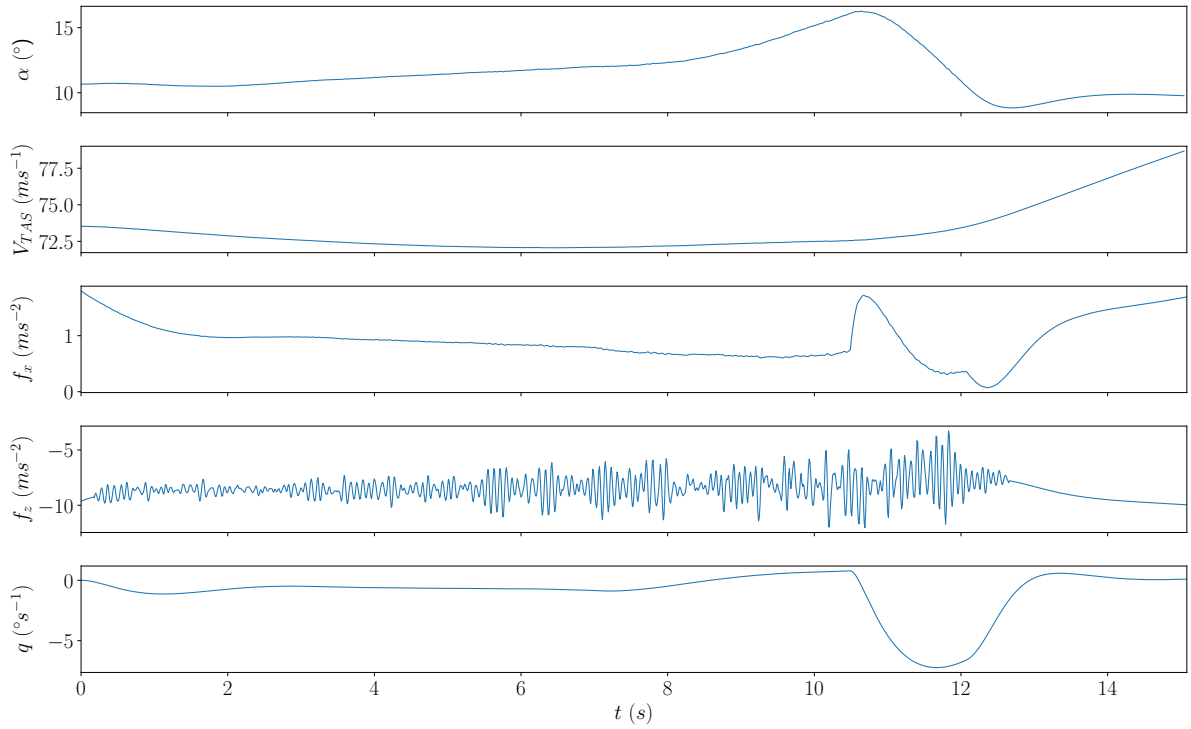


Figure 5.1: Angle of attack, true airspeed, specific force in x and z direction, and the pitch rate for the Citation stall scenario.

the scenario is extended at the end of the scenario with the same but opposite strategy. Furthermore, zero-phase digital filtering was performed to smooth the original and appended data connections and to decrease the amount of stall buffet present in the signal to simplify visual analysis. Moreover, as explained in subsection 4.5.1, the high-frequency motion cues resulting from the stall buffet will not be the limiting motion cues for the MCAs and can, therefore, safely be removed from the reference motion without affecting the outcome of the analysis. The resulting motion scenario used in the remainder of this chapter can be seen in Figure 5.2, represented with the orange line.

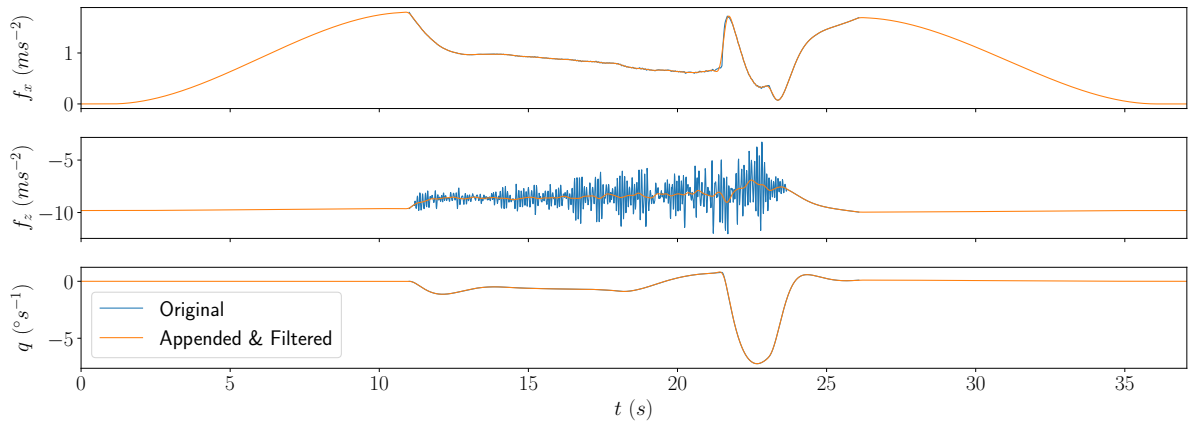


Figure 5.2: Specific force in x and z direction and the pitch rate for the Citation stall scenario including the appended and filtered stall scenario.

5.2. The Offline Motion Simulation Framework

The software used for the analysis presented in this chapter builds upon the Offline Motion Simulation Framework (OMSF) by Katliar et al. [12, 71]. This software library is designed to optimise simulator trajectories for a certain reference motion. Furthermore, the software tool allows for the optimisation of simulator design parameters, however, the latter will not be relevant during this study. To be able to efficiently solve the numerical optimisation problems, the OMSF uses the direct collocation method as defined by Rawlings et al. [38] and elaborated in section 3.2. The CasADi software framework [72] in combination with the Interior Point Optimizer (Ipopt) [73] non-linear optimisation software is used in the OMSF.

To make this software suitable for the analysis of the objective and perceptual motion cueing fidelity of the SRS, a copy of the software is created and changed to a SRS optimal control problem solver. This means that the software optimises the simulator control input to reach a certain simulator state and output. To do this, the motion system of the SRS was programmed into the OMSF with the control input, state and output definitions equal to the definition used in the SRS motion software [74]. This is further elaborated in section 5.3. Furthermore, the reference frame convention is changed to the convention used in the SRS and the possibility to use sensory models is removed throughout the OMSF as explained in subsection 3.4.2. Finally, the option to use the software as an offline MPC-based MCA is implemented, which is elaborated in section 5.4.

5.3. Optimal control problem definition: "Oracle"

Throughout this chapter, the "Oracle" definition, already explained in subsection 3.3.3, will be used to refer to the OMSF being used as a SRS optimal control problem solver, meaning that the entire reference motion scenario is used during the optimisation and that it is performed once. The naming of this definition derives from the fact that this will be the best achievable motion cueing possible with the SRS motion system, for a certain set of cost function weights. This is due to the perfect and infinite knowledge of the reference motion. In the remainder of this section, the formulation of this optimal control problem is presented.

5.3.1. Simulator state, control input and reference and output definitions

The simulator state, control input and output definitions are chosen and implemented in such a way that it resembles the motion software of the SRS. In this way, the offline generated simulator control input can be directly used to playback the motion in the SRS during the experiment. The resulting definitions are presented in (5.1) up to and including (5.5).

The simulator state vector, defined in the inertial frame with respect to the UGP in the neutral position, $F_{I,UGP}$:

$$\boldsymbol{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^\top \tag{5.1}$$

This state is split up into two sub-states, where $\boldsymbol{x}_v$ is the derivative of $\boldsymbol{x}_r$, as can be seen in (5.2).

$$\boldsymbol{x} = [\boldsymbol{x}_r, \boldsymbol{x}_v]^\top \in \mathbb{R}^{12} \tag{5.2a}$$

with

$$\boldsymbol{x}_r = [x, y, z, \phi, \theta, \psi]^\top \tag{5.2b}$$

$$\boldsymbol{x}_v = [\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^\top \tag{5.2c}$$

The reference inertial signal, describing the aircraft motion that is attempted to be replicated by the simulator, is defined in the aircraft frame, F_A:

$$\hat{\boldsymbol{y}} = [\hat{\boldsymbol{f}}^\top, \hat{\boldsymbol{\omega}}^\top, \hat{\boldsymbol{\alpha}}^\top]^\top \in \mathbb{R}^9 \tag{5.3}$$

The output inertial signal as a result of the simulator motion, defined in the simulator frame, F_S (origin of F_S in DERP):

$$\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{u}) = [\boldsymbol{f}^\top, \boldsymbol{\omega}^\top, \boldsymbol{\alpha}^\top]^\top \in \mathbb{R}^9 \tag{5.4}$$

The simulator control input, with the linear accelerations defined in the inertial frame, F_I, and the angular accelerations defined in the platform and simulator frame, F_D and F_S:

$$\boldsymbol{u} = [\ddot{x}, \ddot{y}, \ddot{z}, \dot{p}, \dot{q}, \dot{r}]^\top \tag{5.5}$$

5.3.2. System equations

The relation between the simulator control input and state can be described by two first-order differential equations. The simulator output inertial signal can be described by an output function that is dependent on the aircraft state and control input.

The first differential equation, defining the relation between the sub-states earlier seen in (5.2), is depicted in (5.6).

$$\dot{\boldsymbol{x}}_r - \boldsymbol{x}_v = 0 \tag{5.6}$$

The second differential equation relates the derivative of substate $\boldsymbol{x}_v$ to the simulator control input and current state and can be seen in (5.7).

$$\dot{\boldsymbol{x}}_v - [\ddot{x}, \ddot{y}, \ddot{z}, \ddot{\phi}, \ddot{\theta}, \ddot{\psi}]^\top = 0 \tag{5.7}$$

In (5.7), the linear accelerations, $[\ddot{x}, \ddot{y}, \ddot{z}]$ come directly from the simulator control input $\boldsymbol{u}$ (5.5). The Euler angular accelerations are derived by (5.8). This relation between the body angular accelerations and the Euler angular accelerations can be described by the derivative of the relation between the body angular rates and the Euler angular velocities as previously presented in (2.6). The resulting expression can be seen in (5.8) and includes both state and control input variables. Furthermore, the body angular rates are derived from the simulator state by the relation between the Euler angular velocities and the body angular rates depicted in (2.7).

$$[\ddot{\phi}, \ddot{\theta}, \ddot{\psi}]^\top = \begin{bmatrix} \dot{p} + \dot{q}\sin\phi\tan\theta + q\dot{\phi}\cos\phi\tan\theta + q\dot{\theta}\sin\phi\sec^2\theta \\ \quad +\dot{r}\cos\phi\tan\theta - r\dot{\phi}\sin\phi\tan\theta + r\dot{\theta}\cos\phi\sec^2\theta \\ \dot{q}\cos\phi - q\dot{\phi}\sin\phi - \dot{r}\sin\phi - r\dot{\phi}\cos\phi \\ \sec\theta\left(\dot{q}\sin\phi + q\dot{\phi}\cos\phi + q\dot{\theta}\sin\phi\tan\theta + \dot{r}\cos\phi - r\dot{\phi}\sin\phi + r\dot{\theta}\cos\phi\tan\theta\right) \end{bmatrix} \tag{5.8}$$

The output equations are used to define the simulator inertial signal output (5.4), the specific forces and angular motion produced by the simulator in the simulator frame (F_S) with its origin in the DERP. The simulator output inertial signal specific forces, $\boldsymbol{f}$ (5.4), are defined by (5.9).

$$\boldsymbol{f} = \boldsymbol{a}_S - \boldsymbol{R}_{I,S} \cdot \boldsymbol{g} \tag{5.9a}$$

where:

$$\boldsymbol{a}_S = \boldsymbol{R}_{I,D} \cdot \boldsymbol{a}_I + \boldsymbol{\alpha}_D \times \boldsymbol{r}_{D,S} + \boldsymbol{\omega}_D \times (\boldsymbol{\omega}_D \times \boldsymbol{r}_{D,S}) \tag{5.9b}$$

$$\boldsymbol{a}_I = [\ddot{x}, \ddot{y}, \ddot{z}]^\top \tag{5.9c}$$

$$\boldsymbol{\omega}_D = [p, q, r]^\top \tag{5.9d}$$

$$\boldsymbol{\alpha}_D = [\dot{p}, \dot{q}, \dot{r}]^\top \tag{5.9e}$$

The simulator output inertial signal angular rates, $\boldsymbol{\omega}$ (5.4), can be derived from the angular rates in the platform frame (F_D), found by applying (2.7), and subsequently applying (5.10). Note that during this analysis, the simulator frame is parallel to the platform frame, resulting in an identity matrix for $\boldsymbol{R}_{D,S}$.

$$\boldsymbol{\omega} = \boldsymbol{R}_{D,S} \cdot \boldsymbol{\omega}_D = \boldsymbol{\omega}_D \tag{5.10}$$

The simulator output inertial signal angular accelerations, $\boldsymbol{\alpha}$ (5.4), can directly be derived from the angular accelerations found in the simulator control input $\boldsymbol{u}$ from (5.5) with the help of (5.11).

$$\boldsymbol{\alpha} = \boldsymbol{R}_{D,S} \cdot \boldsymbol{\alpha}_D = \boldsymbol{\alpha}_D \tag{5.11}$$

5.3.3. Cost function

The cost function that is used to optimise the optimal control problem can be divided into two parts and is similar to the cost function used by Katliar et al. [12]. The first part is the inertial signal incongruence cost and penalises the squared difference between the reference inertial signal and the output inertial signal produced by the simulator. This incongruence cost can be defined by (5.12). The inertial signal weight matrix is defined by $\boldsymbol{W}_y$, as found in (5.13).

$$J_{inc}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = ||\boldsymbol{y} - \hat{\boldsymbol{y}}||^2_{\boldsymbol{W}_y} \tag{5.12}$$

$$\boldsymbol{W}_y = \mathbb{1}_{9\times 9} \cdot [\mathbf{w}_f, \mathbf{w}_\omega, \mathbf{w}_\alpha]^\top \tag{5.13}$$

The second part of the cost function is the simulator cost and penalises the simulator state and control input. This ensures washout to the neutral position of the simulator and minimum required control input to reach a certain state. The simulator cost can be defined by (5.14). The state and control input weight matrices are defined by $\boldsymbol{W}_x$ and $\boldsymbol{W}_u$, as found in (5.15) and (5.16), respectively.

$$J_{sim}(\boldsymbol{u}, \boldsymbol{x}) = ||\boldsymbol{x}||^2_{\boldsymbol{W}_x} + ||\boldsymbol{u}||^2_{\boldsymbol{W}_u} \tag{5.14}$$

$$\boldsymbol{W}_x = \mathbb{1}_{12\times 12} \cdot \mathrm{w}_x \tag{5.15}$$

$$\boldsymbol{W}_u = \mathbb{1}_{6\times 6} \cdot \mathrm{w}_u \tag{5.16}$$

The total cost is defined by the sum of the incongruence cost and the simulator cost, resulting in the expression seen in (5.17).

$$J_{total}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \boldsymbol{u}, \boldsymbol{x}) = J_{inc}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + J_{sim}(\boldsymbol{u}, \boldsymbol{x}) \tag{5.17}$$

The total cost $J_{total}(\boldsymbol{y}, \hat{\boldsymbol{y}}, \boldsymbol{u}, \boldsymbol{x})$ will serve as the Lagrangian, $\ell_c(x(t), u(t))$, as previously defined in section 3.2 and (3.1). A final state cost, $V_f(x(T))$, as found in (3.1), will not be included in the cost function for the analysis presented in this chapter.

5.3.4. Constraints

To ensure the simulator stays within its physical limits and to set initial and final conditions of the simulator state, constraints on the state, control input, and actuators are included in the optimal control formulation. The constraints are presented for the discretised state, control input and actuator vectors, $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{q}$, respectively. These discretised vectors contain the simulator state, control input and actuator length estimated at every ith subinterval $[t_i, t_{i+1}]$, for N subintervals on the time interval $[0, T]$. These constraints are presented in (5.18) up to and including (5.23). The discretisation interval is indicated with a subscript, meaning that $\mathbf{x}_i = \mathbf{x}[i]$. Lower boundaries are indicated with an underline,

for example, $\underline{\boldsymbol{x}}$, and upper boundaries are indicated with an overline, for example, $\overline{\boldsymbol{x}}$. Note that the boundaries are defined as vectors of continuous time, where the subscript indicates the point in time, meaning $\overline{\boldsymbol{x}}_t = \overline{\boldsymbol{x}}(t)$.

The state constraint for $i = 1, 2, ..., N-1$:

$$\underline{\boldsymbol{x}} \leq \mathbf{x}_i \leq \overline{\boldsymbol{x}} \tag{5.18}$$

The initial state constraint for $i = 0$:

$$\underline{\boldsymbol{x}}_0 \leq \mathbf{x}_0 \leq \overline{\boldsymbol{x}}_0 \tag{5.19}$$

The final state constraint for $i = N$:

$$\underline{\boldsymbol{x}}_T \leq \mathbf{x}_N \leq \overline{\boldsymbol{x}}_T \tag{5.20}$$

The control input constraint for $i = 0, 1, ..., N$:

$$\underline{\boldsymbol{u}} \leq \mathbf{u}_i \leq \overline{\boldsymbol{u}} \tag{5.21}$$

The actuator length constraint for $i = 0, 1, ..., N$:

$$\underline{\boldsymbol{q}} \leq \mathbf{q}_i \leq \overline{\boldsymbol{q}} \tag{5.22}$$

The actuator velocity constraint for $i = 0, 1, ..., N$:

$$\underline{\dot{\boldsymbol{q}}} \leq \dot{\mathbf{q}}_i \leq \overline{\dot{\boldsymbol{q}}} \tag{5.23}$$

5.3.5. Baseline

To start the analysis, a baseline setting for the "Oracle" optimal control problem should be formulated. The state, control input and actuator boundaries are derived from the SRS motion system characteristics. Bounds that can not be defined by the SRS motion system characteristics, since they are simply not limited by design, are set to non-limiting values. The weights of the cost function used in the baseline model are based on literature. Two settings of the angular rate weights, $\mathbf{w}_\omega$, were taken into account. First, a setting of [10,10,10] $\mathrm{rad}^2\,\mathrm{s}^{-2}$, as used by Ploeg et al. [21], Cleij et al. [22] and Katliar [71]. Secondly, a setting of [100,100,100] $\mathrm{rad}^2\,\mathrm{s}^{-2}$ used by Katliar et al. [34] and Grottoli et al. [20]. After visual inspection of the outputs produced by the two different weight settings, the setting of $10\,\mathrm{rad}^2\,\mathrm{s}^{-2}$ was chosen due to the fewer false cues present. The total list of parameters used for the baseline "Oracle" formulation can be seen in Table 5.1. The complete solution of the baseline "Oracle" SRS optimal control problem can be found in Appendix A.

Table 5.1: Parameters of the baseline "Oracle" SRS optimal control problem.

Parameter	Value
Δt_i	$0.05s$
$\boldsymbol{W_y}$	$\mathbb{1}_{9\times9} \cdot [1,1,1,10,10,10,1,1,1]^\top$
$\boldsymbol{W}_x$	$\mathbb{1}_{12\times12} \cdot 0.01$
$\boldsymbol{W}_u$	$\mathbb{1}_{6\times6} \cdot 0.01$
$\underline{\boldsymbol{x}}$	$[\underline{\boldsymbol{x}}_r, \underline{\boldsymbol{x}}_v]^\top$
$\overline{\boldsymbol{x}}$	$[\overline{\boldsymbol{x}}_r, \overline{\boldsymbol{x}}_v]^\top$
$\underline{\boldsymbol{x}}_{r,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top$
$\overline{\boldsymbol{x}}_{r,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top$
$\underline{\boldsymbol{x}}_{v,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\overline{\boldsymbol{x}}_{v,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\underline{\boldsymbol{x}}_r$	$[-0.981m, -1.031m, -0.636m, \frac{-25.9\pi}{180}rad, \frac{-23.7\pi}{180}rad, \frac{-41.6\pi}{180}rad]^\top$
$\overline{\boldsymbol{x}}_r$	$[1.259m, 1.031m, 0.678m, \frac{25.9\pi}{180}rad, \frac{24.3\pi}{180}rad, \frac{41.6\pi}{180}rad]^\top$
$\underline{\boldsymbol{x}}_v$	$[-1000m, -1000m, -1000m, -1000rad, -1000rad, -1000rad]^\top s^{-1}$
$\overline{\boldsymbol{x}}_v$	$[-1000m, -1000m, -1000m, -1000rad, -1000rad, -1000rad]^\top s^{-1}$
$\underline{\boldsymbol{x}}_{r,T}$	$[-0.981m, -1.031m, -0.636m, \frac{-25.9\pi}{180}rad, \frac{-23.7\pi}{180}rad, \frac{-41.6\pi}{180}rad]^\top$
$\overline{\boldsymbol{x}}_{r,T}$	$[1.259m, 1.031m, 0.678m, \frac{25.9\pi}{180}rad, \frac{24.3\pi}{180}rad, \frac{41.6\pi}{180}rad]^\top$
$\underline{\boldsymbol{x}}_{v,T}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\overline{\boldsymbol{x}}_{v,T}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\underline{\boldsymbol{u}}$	$[-10m, -10m, -10m, -2rad, -2rad, -2rad]^\top s^{-2}$
$\overline{\boldsymbol{u}}$	$[10m, 10m, 10m, 2rad, 2rad, 2rad]^\top s^{-2}$
$\underline{\boldsymbol{q}}$	$[2.131, 2.131, 2.131, 2.131, 2.131, 2.131]^\top m$
$\overline{\boldsymbol{q}}$	$[3.281, 3.281, 3.281, 3.281, 3.281, 3.281]^\top m$
$\underline{\dot{\boldsymbol{q}}}$	$[-0.75, -0.75, -0.75, -0.75, -0.75, -0.75]^\top m s^{-1}$
$\overline{\dot{\boldsymbol{q}}}$	$[0.75, 0.75, 0.75, 0.75, 0.75, 0.75]^\top m s^{-1}$

5.4. Model Predictive Control definition

To take the optimal control problem one step further and change it into an offline MPC-based MCA, it is required to introduce additional parameters, make changes to the constraints, and define prediction strategies.

First, the time interval $[0, T]$ is divided into K subintervals of the form $[t_k, t_{k+1}]$. Every kth interval, the MPC algorithm creates a smaller reference inertial signal on the time interval $[t_k, t_{k+N}]$ existing of N subintervals of the form $[t_i, t_{i+1}]$, resulting in an optimal control problem of the form found in (3.5). The resulting new parameters are the MPC sample time, Δt_k, and secondly, the prediction horizon length, N, which is equal to the number of discretisation intervals in the optimal control problem. It is important to note that $\Delta t_k = \Delta t_i$, meaning that the subintervals for k and the subintervals for i are perfectly aligned. For reference and visualisation of the interval definitions, Figure 3.1 can be used.

5.4.1. MPC constraints

In (5.24) to (5.27), the constraints for the MPC formulation are found. It is essential to note that it is necessary to define the constraint boundaries in either continuous time, for example, $\underline{\boldsymbol{x}}_t = \underline{\boldsymbol{x}}(t)$, or discrete time, for example, $\underline{\mathbf{x}}_{k,i} = \underline{\mathbf{x}}[k, i]$.

The changes applied to the constraints regard among others the initial state condition constraint. Since

there are now K smaller optimal control problems to be solved, K initial state conditions should be set. Meaning that for $k = 0$ and $i = 0$:

$$\underline{\boldsymbol{x}}_0 \leq \mathbf{x}_{k,i} \leq \overline{\boldsymbol{x}}_0 \tag{5.24}$$

And for $k = 1, 2, ..., K-1$ and $i = 0$:

$$\underline{\mathbf{x}}_{k-1,1} \leq \mathbf{x}_{k,i} \leq \overline{\mathbf{x}}_{k-1,1} \tag{5.25}$$

Additionally, terminal state constraints are introduced, defined as the final conditions of the smaller optimal control problems at interval $[k = k, i = N]$ and where this same interval is not equal to the interval $[k = K,\ i = 0 = N]$. Meaning that for $k = 0, 1, ..., K-N-1$ and $i = N$:

$$\underline{\mathbf{x}}_{k,N} \leq \mathbf{x}_{k,i} \leq \overline{\mathbf{x}}_{k,N} \tag{5.26}$$

The final state constraints are subsequently defined as the final condition of the entire MPC problem, thus at time T. Meaning that for $k = K-N, ..., K$ and $i = N$:

$$\underline{\boldsymbol{x}}_T \leq \mathbf{x}_{k,i} \leq \overline{\boldsymbol{x}}_T \tag{5.27}$$

Note that for the normal (not initial, terminal or final) state constraints, the control input constraint and actuator constraints nothing changes, since these constraints will only be used within the smaller optimal control problems.

5.4.2. MPC prediction strategies

The last requirement for the MPC formulation is the introduction of the prediction strategies. Note the difference between the discretised reference inertial signal vector, $\hat{\mathbf{y}}_{k,i} = \hat{\mathbf{y}}[k, i]$, indicating the prediction horizon, and the continuous time reference inertial signal vector, $\hat{\boldsymbol{y}}_t = \hat{\boldsymbol{y}}(t)$, indicating the reference inertial signal for the entire reference motion. For now, two strategies are proposed. The first strategy assumes perfect knowledge of the reference inertial signal for the entire prediction horizon length and is named the "Perfect" prediction strategy, as also elaborated in subsection 3.3.3. This strategy is mathematically defined by (5.28). The "Perfect" prediction strategy means that for $k = 0, 1, ..., K$:

$$\hat{\mathbf{y}}_{k,i=0,...,N} = \hat{\boldsymbol{y}}_{[t_k,...,t_{k+N}]} \tag{5.28}$$

The secondly proposed prediction strategy is the "Constant" prediction strategy and is based on only knowledge of the reference inertial signal of the current point in time, t_k, as also elaborated in subsection 3.3.3. This strategy is mathematically defined by (5.29). The "Constant" prediction strategy means that for $k = 0, 1, ..., K$:

$$\hat{\mathbf{y}}_{k,i=0,...,N} = \hat{\boldsymbol{y}}_{t_k} \tag{5.29}$$

5.4.3. MPC baseline

With the complete baseline MPC formulation, Table 5.2 is created with the additionally required parameters with respect to the parameters presented in Table 5.1.

Table 5.2: Parameters of the baseline MPC formulation

Parameter	Value
Δt_k	$0.05s$
N	200
$\underline{\boldsymbol{x}}_{r,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top$
$\overline{\boldsymbol{x}}_{r,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top$
$\underline{\boldsymbol{x}}_{v,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\overline{\boldsymbol{x}}_{v,0}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\underline{\mathbf{x}}_{r,k,N}$	$[-0.981m, -1.031m, -0.636m, \frac{-25.9\pi}{180}rad, \frac{-23.7\pi}{180}rad, \frac{-41.6\pi}{180}rad]^\top$
$\overline{\mathbf{x}}_{r,k,N}$	$[1.259m, 1.031m, 0.678m, \frac{25.9\pi}{180}rad, \frac{24.3\pi}{180}rad, \frac{41.6\pi}{180}rad]^\top$
$\underline{\mathbf{x}}_{v,k,N}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\overline{\mathbf{x}}_{v,k,N}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\underline{\boldsymbol{x}}_{r,T}$	$[-0.981m, -1.031m, -0.636m, \frac{-25.9\pi}{180}rad, \frac{-23.7\pi}{180}rad, \frac{-41.6\pi}{180}rad]^\top$
$\overline{\boldsymbol{x}}_{r,T}$	$[1.259m, 1.031m, 0.678m, \frac{25.9\pi}{180}rad, \frac{24.3\pi}{180}rad, \frac{41.6\pi}{180}rad]^\top$
$\underline{\boldsymbol{x}}_{v,T}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$
$\overline{\boldsymbol{x}}_{v,T}$	$[0m, 0m, 0m, 0rad, 0rad, 0rad]^\top s^{-1}$

5.5. Validation of the constraints

The initial and final state constraints and the actuator length and velocity constraints can be validated by inspecting the solution to the "Oracle" optimal control problem. First, it is checked that the initial value is zero for all state variables. Furthermore, it is checked that the final condition is zero for the $\boldsymbol{x}_v$ variables and arbitrary for the $\boldsymbol{x}_r$ variables. From this, it can be concluded that the initial and final state constraints are working correctly. If interested, the figure used for this validation can be found in Figure A.2.

The actuator length and velocity constraints are validated by inspecting Figure 5.3 and Figure 5.4. From Figure 5.3, it can be concluded that the actuator length constraint is working correctly since the actuator length reaches its limits and stays within these limits. For the actuator velocity, the limits of $75\,\mathrm{cm\,s^{-1}}$ are not reached during the "Oracle" baseline solution. Therefore, only for this constraint validation, the actuator velocity limit is set to $25\,\mathrm{cm\,s^{-1}}$ resulting in Figure 5.4. From this figure, it can be concluded that the actuator velocity constraint is working properly.

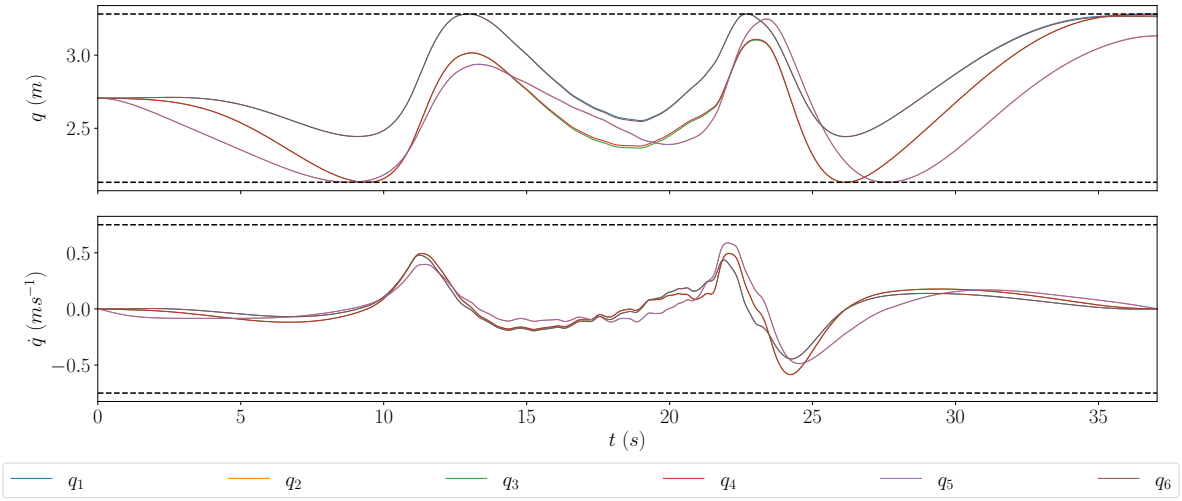


Figure 5.3: Actuator length and velocity for the "Oracle" baseline condition, with the original actuator velocity limit of $75\,\mathrm{cm\,s^{-1}}$. The actuator length limit is shown by the black dashed line.

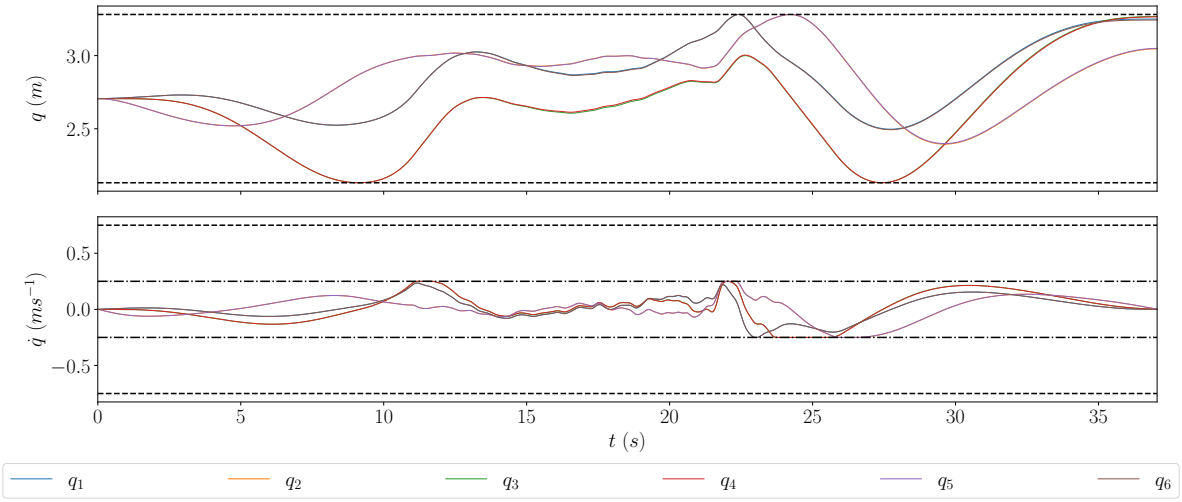


Figure 5.4: Actuator length and velocity for the "Oracle" condition, after lowering the actuator velocity limit to $25\,\mathrm{cm\,s^{-1}}$. The original actuator velocity limit is shown by the black dashed line, lower actuator velocity limit is shown by the black dashed-dotted line.

5.6. Results

The results presented in this section are related to the prediction strategy, the prediction horizon length, and the inertial signal weights used in the cost function. The "Oracle" results and the MPC with "Perfect" and "Constant" prediction strategies results are compared to the CW used in the SRS. For this, two different CW settings are used, being the motion filter parameters used by Imbrechts et al. [10], defined as CW_I, and the parameters used by Grant and Schroeder [11], defined as CW_G. For the parameters used by Grant and Schroeder [11], the pitch and roll fidelity are close to high-fidelity and the translational motion is low-fidelity according to the Sinacori-Schroeder motion fidelity criteria [30]. Compared to the parameters used by Grant and Schroeder [11], the parameters used by Imbrechts et al. [10] result in higher fidelity for translational motion and lower fidelity for angular motion. Both these CW settings, CW_I and CW_G, can be found in full in Appendix B.

5.6.1. Influence of the prediction strategy

The influence of the prediction strategy is studied by comparing the "Oracle", the "Perfect" and the "Constant" baseline conditions with the two CW settings mentioned previously. In Figure 5.5, the RMSE and PCC are shown for the various conditions. In this figure, it can be seen that the "Oracle" and "Perfect" baseline conditions are very similar. Furthermore, these conditions outperform all other conditions except for the angular rates PCC, where the two CW conditions are slightly better. Looking at Figure 5.6, these results can be visibly validated, in f_x the "Oracle" and "Perfect" conditions are more close to the reference signal due to more pitch up tilt of the simulator. Additionally, in f_z, the shape of the "Oracle" and "Perfect" output signal is more similar to the reference and shows larger amplitudes. Finally, the pitch rate signal shape of the "Oracle" and "Perfect" conditions is particularly different from the reference around $t = 20s$ and during the final 20 seconds of the scenario. This could explain the higher PCC value for the angular rates of the two CW references. However, less phase shift is present in the pitch rate for the "Perfect" condition compared to the CW references, as found at $t = 22s$ in Figure 5.6.

In Figure 5.5, for the "Constant" condition, only the RMSE of the specific forces is better than the CW references and it is slightly worse than the "Oracle" and "Perfect" conditions. The "Constant" RMSE for the angular rates and the PCC for the specific forces and angular rates are the worst of all conditions. This can be validated by inspecting Figure 5.6, the reasonable RMSE for the specific forces is due to the good pitch tilt coordination resulting in a decent f_x. However, the shape and amplitudes of the "Constant" f_z signal are very bad since there is almost no motion, resulting in the low PCC value for

the "Constant" specific forces. Finally, for the angular rates, more delay (at $t = 13s$) and false cues (at $t = 22s$) can be identified, resulting in the low RMSE and PCC values for the angular rates.

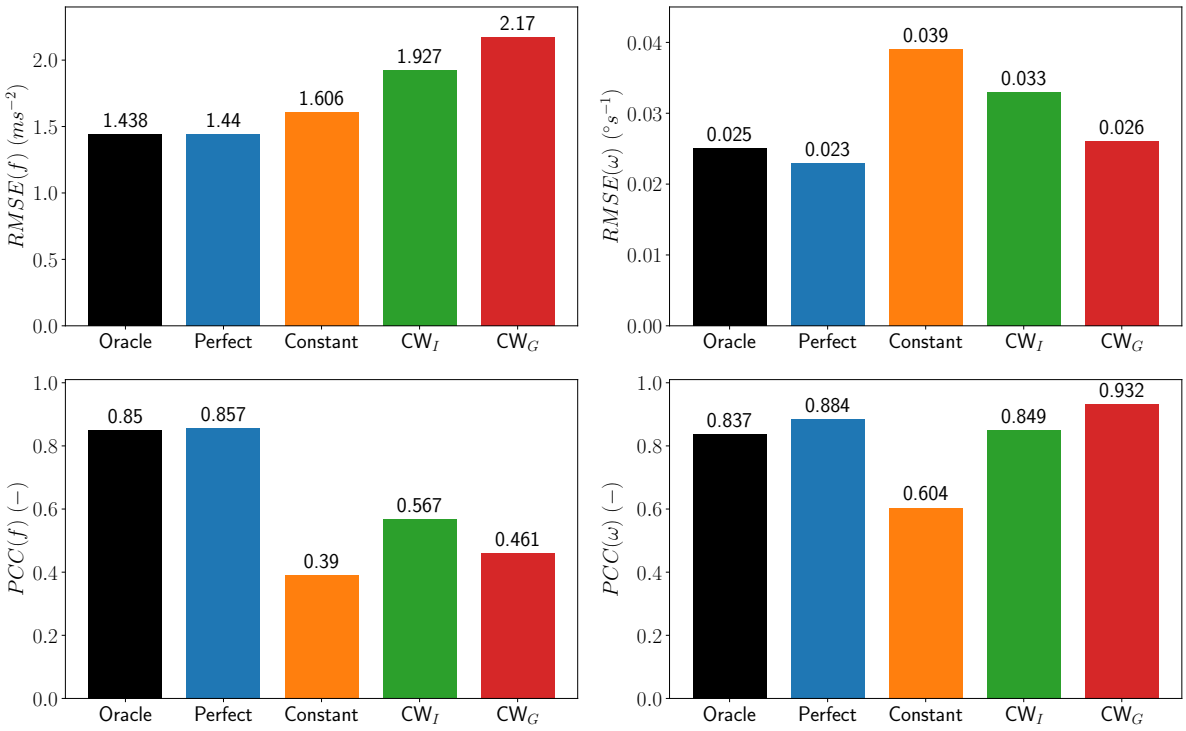


Figure 5.5: RMSE and PCC for the inertial signal specific forces and angular rates as a result of different prediction strategies. For the RMSE, the sum of the three RMSE values of the specific forces and of the angular rates are presented. For the PCC, the average of the three PCC values is presented.

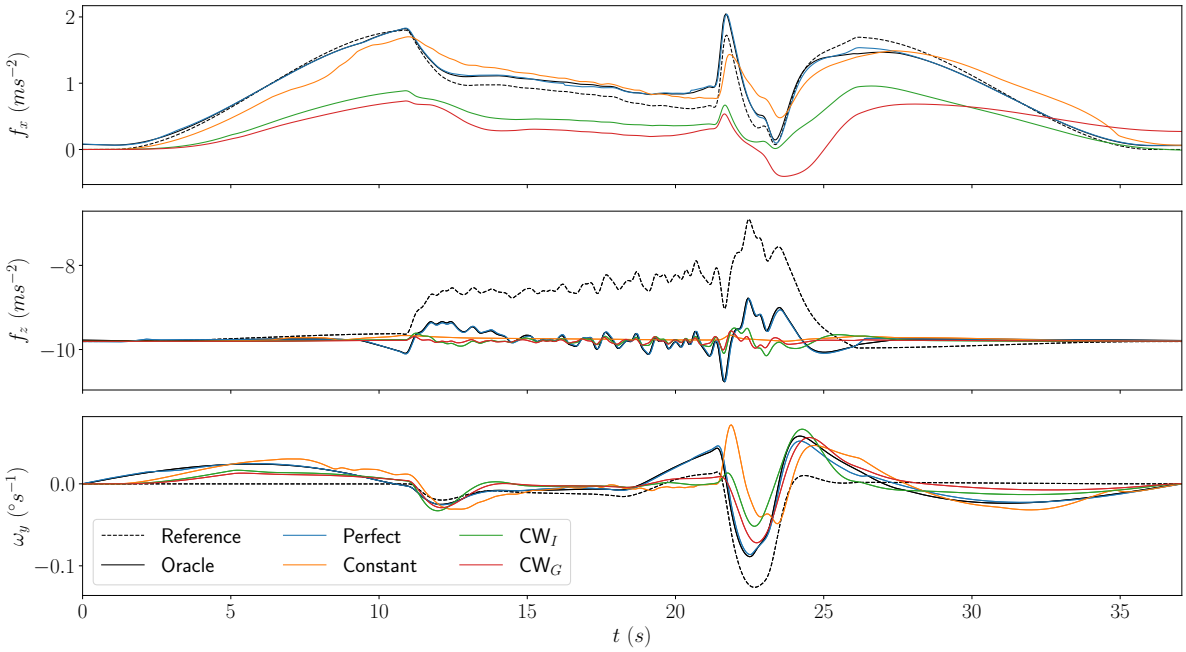


Figure 5.6: Reference inertial signal and the output inertial signal for the "Oracle", the "Perfect", the "Constant" and the two CW conditions.

5.6.2. Influence of the prediction horizon length

The influence of the prediction horizon length is studied by comparing horizon length ranging from 0.5 s to 10 s and comparing the RMSE and PCC with the "Oracle" condition. This can be seen in Figure 5.7. From this figure, it can be concluded that increasing the horizon length over 6 s has very little influence on the RMSE and PCC values. Furthermore, the angular rates' RMSE and PCC for the "Constant" condition, get worse with an increasing horizon length. For the "Perfect" condition, the angular rates' RMSE and PCC are also best for a horizon of 0.5 s, slightly worse for a horizon of 1 s to 5 s and finally, close to the "Oracle" condition for a horizon of 6 s to 10 s.

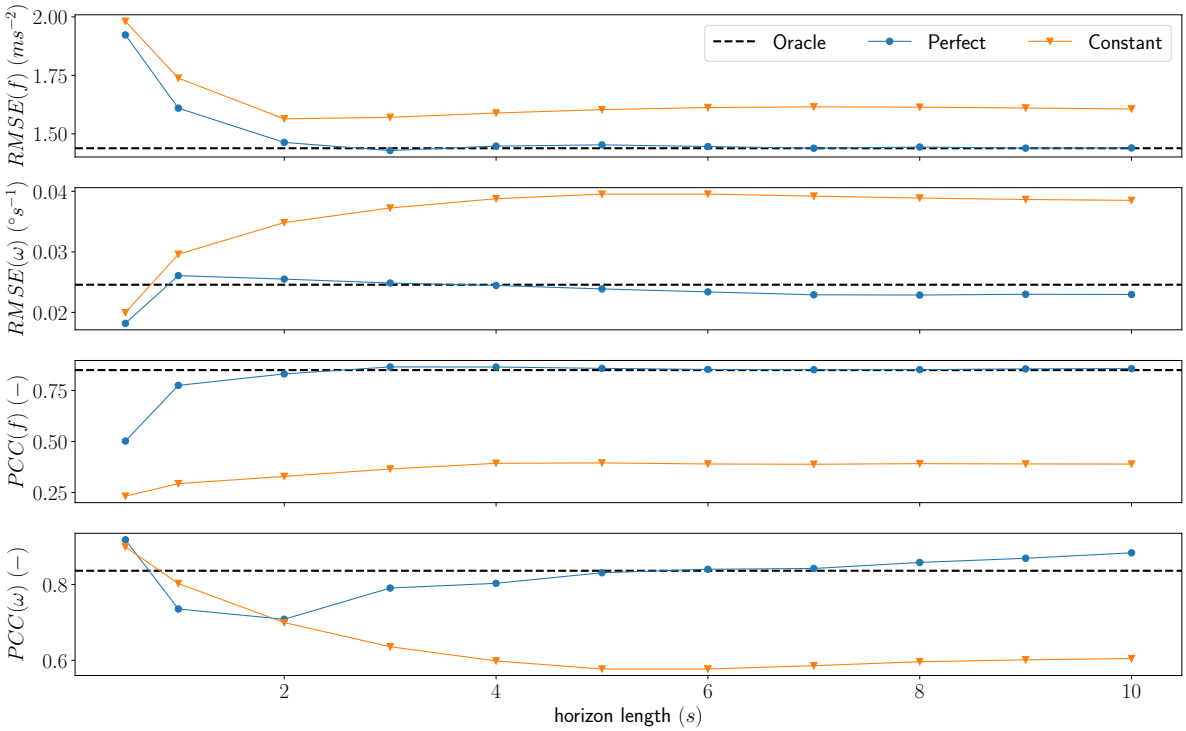


Figure 5.7: RMSE and PCC for different prediction horizon lengths, for the "Perfect" and "Constant" conditions.

5.6.3. Influence of the specific force and angular rate weights

The influence of the specific force and angular rate weights are studied to check whether the angular rates of an "Oracle" condition can outperform the CW references. Secondly, a weight setting is looked for that results in better specific force performance than the baseline, while the angular rates remain reasonable.

This was done by performing two sensitivity analyses, the first analysis was done by comparing the results obtained by ranging all three specific forces weights, captured in the vector $\mathbf{w}_f$, from $1\,\mathrm{m^2s^{-4}}$ to $8\,\mathrm{m^2s^{-4}}$ with weight increment steps of $1\,\mathrm{m^2s^{-4}}$. The output inertial signal, RMSE, and PCC were compared for all the different weights. During this comparison, the output inertial signal was visually analysed and assessed on reference inertial signal tracking performance and the amount and size of false cues. Thereafter, this was validated by the RMSE, indicating tracking performance, and the PCC, indicating false cues. If interested, the figures used for this analysis can be found in Appendix D, containing in Figure D.2 the reference and output inertial signals and in Figure D.1 the RMSE and PCC values. This analysis resulted in a chosen specific force weight setting of $2\,\mathrm{m^2s^{-4}}$ and is referred to as the "Oracle$_f$" condition, as shown in Figure 5.8 and Figure 5.9. From Figure 5.8, it can be concluded that the specific forces are performing better than the baseline "Oracle" condition, which is validated by looking at Figure 5.9. The angular rates RMSE values are very similar to the "Oracle" baseline condition, however, the angular rate PCC is clearly worse. In Figure 5.9, a reason for this can be found

at $t = 27$, where a false cue is present for the pitch rate of the "Oracle$_f$" condition.

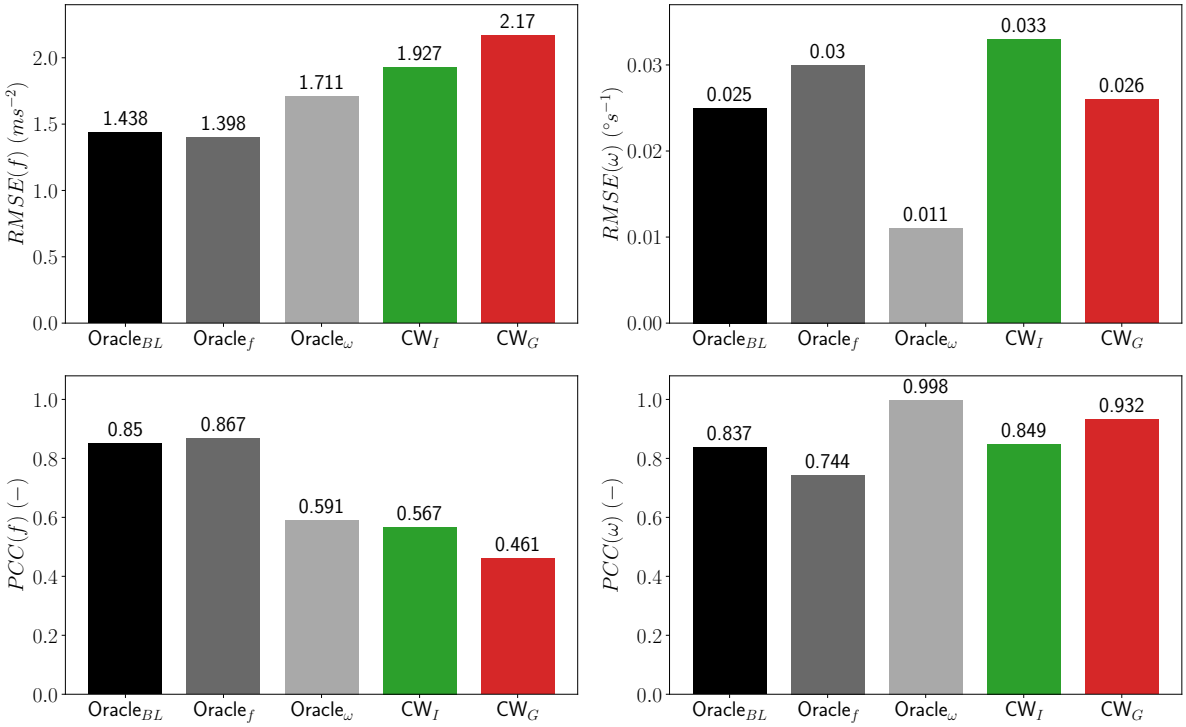


Figure 5.8: RMSE and PCC for the inertial signal specific forces and angular rates as a result of different weight settings. For the RMSE, the sum of the three RMSE values of the specific forces and of the angular rates are presented. For the PCC, the average of the three PCC values is presented.

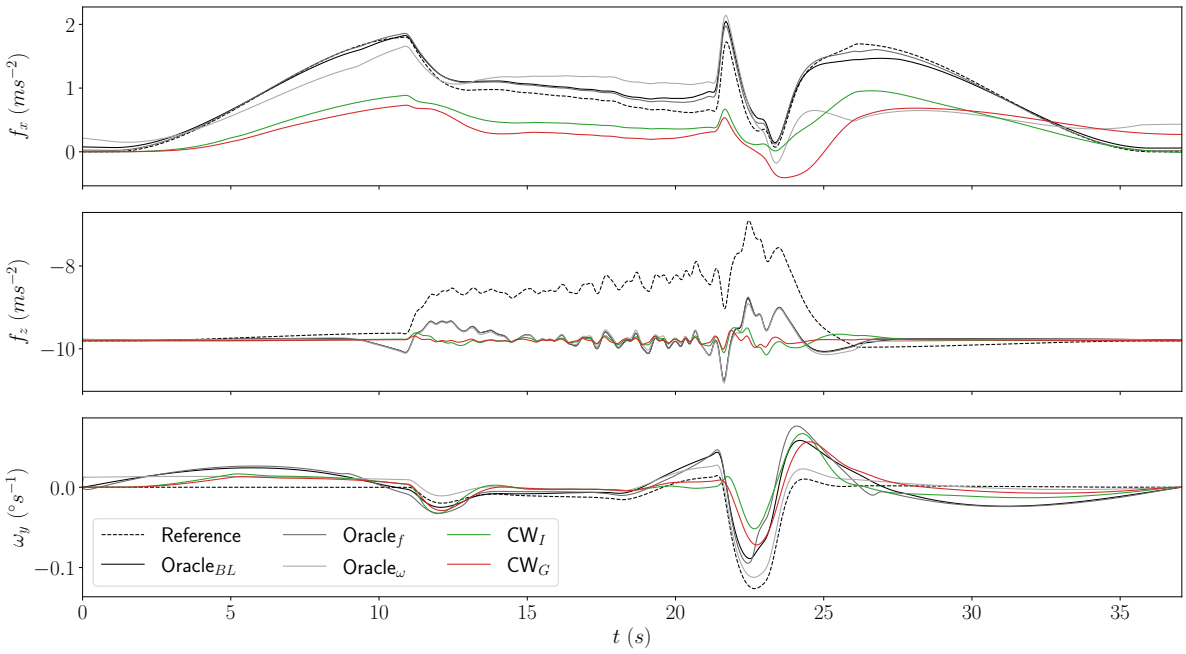


Figure 5.9: Reference inertial signal and the output inertial signal for the "Oracle" with three different weight settings and the two CW conditions.

During the second sensitivity analysis, the results were compared by ranging all three angular rate weights, captured in the vector $\mathbf{w}_\omega$, from $10\,\mathrm{rad^2 s^{-2}}$ to $100\,\mathrm{rad^2 s^{-2}}$, with weight increment steps of $10\,\mathrm{rad^2 s^{-2}}$. The results were analysed with an equal approach as the approach used for the analysis of

the specific force weights. If interested, the figures used for this analysis can be found in Appendix D, containing in Figure D.6 the reference and output inertial signals and in Figure D.5 the RMSE and PCC values. The analysis resulted in the "Oracle$_\omega$" condition with angular rate weights of $60\,\mathrm{rad^2 s^{-2}}$, as found in Figure 5.8 and Figure 5.9. The angular rates now outperform both CW references, while the specific forces remain also better than the CW references. The "Oracle$_\omega$" performs slightly worse for the specific forces compared to the baseline "Oracle" and the "Oracle$_f$", which is mainly due to the larger error of f_x, as visible in Figure 5.9.

5.7. Discussion of the results

The results are discussed in the same order used in section 5.6, being first the prediction strategy, then the prediction horizon length, and finally, the specific force and angular rate weights.

5.7.1. Prediction strategy

An explanation of why the "Oracle" and "Perfect" conditions are very similar (see Figure 5.5 and Figure 5.6) could be that both conditions have perfect knowledge of the reference signal. Furthermore, the prediction horizon length of the "Perfect" condition is 10 s, which is apparently long enough to fully utilise the available motion space for onset cues and possible simulator pre-positioning for this reference motion scenario.

The reason why more pitch tilt coordination is present for the "Oracle", "Perfect" and "Constant" conditions compared to the CW references (see f_x in Figure 5.6) could be explained by the CW gain, K_{f_x}. This gain is 0.5 and 0.4 for, respectively, the CW$_I$ and CW$_G$ conditions, meaning that the CW references will never try to replicate the f_x signal for more than 50% of the original reference inertial signal magnitude.

The "Constant" condition has almost no motion in heave (see f_z in Figure 5.6), the non-sustained or onset amplitude variations in f_z are not present at all. This is explainable by the fact that there is a terminal constraint present, constraining the state velocities and angular rates to be zero at the end of the prediction horizon. This means that every acceleration in a certain direction should be counteracted by an acceleration in the opposite direction. Hence, this will result in the same absolute error as when no acceleration at all is applied since the "Constant" condition demand a constant acceleration over the entire horizon. Combining this with the fact that the control input is also slightly penalised, the no-acceleration scenario is preferred by the algorithm, leading to almost no heave motion. When there would be no terminal constraint, it will not solve this problem completely, since the same phenomenon will be present due to the limited actuator lengths and velocity. However, better performance in heave is expected.

An explanation of the large false cues in the pitch rate for the "Constant" condition (see ω_y at $t = 22s$ in Figure 5.6), could be that the simulator is very good at reproducing sustained, thus constant, f_x cues. This is done by tilt coordination, thus the algorithm continuously follows the f_x signal by tilting the simulator in pitch. The algorithm probably prefers to correctly cue the f_x over the pitch rate due to the chosen weights in the cost function. This results in false cues in the pitch rate signal when the f_x changes rapidly. An explanation why this false cue at $t = 22s$ is not present for the "Perfect" condition can be explained by the fact that, in contrary to the "Constant" condition, the "Perfect" condition can pre-position itself (see ω_y at $t = 18s$ until $t = 22s$ in Figure 5.6), since it exactly knows up to 10 s in front what motion is coming.

5.7.2. Prediction horizon length

Why very short horizon lengths (0.5 s) are good for tracking the angular rate signals (see Figure 5.7), could be explained by the fact that the angular onset cues are better reproducible by the simulator than the specific force onset cues. This results in the algorithm preferring cueing onset angular rates at the cost of lower fidelity specific forces.

The reason why increasing the prediction horizon length over 6 s has little influence (see Figure 5.7)

could be that the reference inertial signal does not contain motion cues that need longer simulator pre-positioning of 6 s. Secondly, it could be that the size of the motion system, meaning the actuator range, and the actuator velocity limit are an explanation of this. A larger motion system with higher actuator velocity limits could possibly benefit from longer knowledge of the reference signal because it can simply reproduce more of the reference signal. It can better and longer cue onset-specific forces resulting in higher actuator velocities and has more space for pre-positioning.

5.7.3. Specific force and angular rate weights

For the specific force sensitivity analysis, possibly better results are possible. The chosen weight setting "Oracle$_f$", with specific force weights of $2\,\mathrm{m^2s^{-4}}$, is namely the first of seven weight increment steps of $1\,\mathrm{m^2s^{-4}}$ from the baseline condition used in the sensitivity analysis.

5.8. Preliminary recommendations

In this section, recommendations are given that can be taken into account while designing the experiment in Appendix 6 and for future or additional analyses similar to the analysis presented in this chapter.

The first recommendation is to either choose the "Oracle" or the "Perfect" condition as experiment conditions due to the large similarity. The "Perfect" condition would then be favoured due to the fact that it is more realistic.

Secondly, the "Constant" condition should be improved by trying first to remove the terminal constraint. If this will lead to unstable behaviour, an additional terminal state weight could be added to the cost function. If the "Constant" prediction is still not performing well, a different prediction strategy should be proposed. This could be for example a strategy where the first half of the prediction horizon is the original constant prediction, and the second half of the horizon is the opposite of that prediction. If this still leads to reaching the actuator velocity constraint, a prediction strategy could be chosen where the reference signal is for the first part of the horizon the original constant prediction strategy, the second part the opposite, and the rest of the horizon zero. Of course, this should be implemented with smooth and "continuous" transitions.

Thirdly, one prediction horizon length should be chosen between 6 s and 10 s, because this range is computationally feasible and does not have an influence on the motion cueing fidelity. Furthermore, the influence of making the prediction horizon non-uniform should be investigated. This might result in less computationally heavy problems and this is more realistic since it is often used in practice.

Fourthly, the specific force weight sensitivity analysis grid should be made finer, resulting in a more precise analysis. This could also be done for the angular rate weight analysis, once the optimal or desired weight is approximately reached.

Finally, it could be beneficial to perform spectral analysis. This could for example identify the lack of heave motion for the "Constant" prediction without having to visualise the inertial signal output. This would mainly be useful when a full stall buffet is present, which makes visual analysis of the inertial signal difficult, contrary to the filtered buffet in this chapter.

5.9. Chapter conclusions

The answer to the fourth research question and the answers to the accompanying sub-questions are provided below to conclude this chapter.

SQ 4. *How can the implementation of an MPC-based MCA during UPRT affect the objective motion cueing fidelity with respect to the current SRS CW MCA?*
The "Oracle" and "Perfect" conditions both show better and more tilt coordination, resulting in an increase in longitudinal motion cueing fidelity. Moreover, the "Perfect" condition shows less phase shift (delay) and more resemblance in shape with the reference motion. The optimal control problem can be tuned ("Oracle$_\omega$") to outperform the reference CW filters for all used measures.

The "Constant" prediction shows promising longitudinal performance, however, in heave and pitch, it performs clearly worse than the CW references.

4a. *What optimal control and MPC formulation should be used for the analysis?*
The state and control input definitions used in the optimal control and MPC formulation match the definitions used in the SRS motion software [74]. The system equations are two first-order implicit differential equations relating the control input and the state. Constraints are imposed on the actuator lengths and velocities and the initial, current, terminal and final state.

4b. *What is the influence of the prediction strategy on the objective motion cueing fidelity?*
The prediction strategy has a large influence on the motion cueing quality. Both the "Oracle" and "Perfect" conditions outperform the reference CW motion filters. Moreover, these conditions are very much alike, using both conditions in the pilot experiment would therefore not be recommended. The "Constant" condition performs worse than the CW references, mainly due to the very little motion in heave and a large false cue in pitch due to fast tilt coordination.

4c. *What is the influence of the prediction horizon length on the objective motion cueing fidelity?*
The prediction horizon length has little influence on the motion cueing quality if the length is increased from 6 s to 10 s. It is recommended to use one prediction horizon length for the remainder of this study.

4d. *What is the influence of the cost function weights on the objective motion cueing fidelity?*
The cost function weights are very decisive for the output of the MCA. The analysis showed that it is possible to derive substantially different specific force and angular rate weight settings without introducing large false cues. In the pilot experiment, multiple weight conditions are recommended. Lastly, the specific force weights are a factor of 10 to 100 more effectual than the angular rate weights, meaning that a specific force weight sensitivity analysis should be performed with a much smaller weight grid.

Chapter 6

Experiment plan

In this chapter, the (tentative) plan and design of the proposed experiment for answering the fifth research question posed in section 1.2 are presented.

> *How can the implementation of an MPC-based MCA during UPRT affect the* ***perceptual*** *motion cueing fidelity with respect to the current SRS CW MCA?*

The experiment plan will answer the accompanying sub-questions that were also presented in section 1.2. This chapter starts with the experiment objective and hypothesis in section 6.1 and section 6.2, respectively. Thereafter, in section 6.3, the experiment design, including the used apparatus and different experiment variable definitions, is introduced. This is followed by the experiment conditions, participants (matrix), and procedures in section 6.4, section 6.5, and section 6.6. Subsequently, in section 6.7, the approach for post-experiment data analysis is presented. To conclude this chapter, the posed research questions and the corresponding sub-questions are succinctly answered in section 6.8.

6.1. Experiment objective

The objective of the experiment is to evaluate the perceived motion cueing fidelity for the newly proposed MCA. Furthermore, the influence of the cost function weights and the prediction strategy are investigated. Finally, the experiment serves as validation of the objective motion cueing fidelity assessment presented in Appendix 5 and as an investigation towards a potential correlation with the perceived motion cueing fidelity as a result of the experiment.

6.2. Hypotheses

The following preliminary hypotheses are derived from the analyses performed in Appendix 5:

1. The CW MCA will result in a lower perceived motion cueing fidelity compared to the MPC-based MCA (MCAs specified in subsection 6.3.2).
2. The perceived motion cueing fidelity will increase under increasing prediction correctness of the reference inertial signal.
3. The perceived motion cueing fidelity will be higher for the balanced weight setting compared to the specific force or angular weight settings (weight settings specified in subsection 6.3.2).
4. The perceived motion cueing fidelity will be higher for the specific force weight setting compared to the angular weight settings (weight settings specified in subsection 6.3.2).
5. The perceived motion cueing fidelity ratings will be in line with the objective motion cueing fidelity defined by the RMSE and PCC.

6.3. Experiment design

The subjects of the experiment will participate in a *passive* simulated aircraft upset scenario presented in a flight simulator and will be asked to rate the incongruity between the visual images, including an Primary Flight Display (PFD) extended with the angle of attack and load factor indication, and the motion they experience. This incongruence rating will then be used to evaluate the proposed MCA, compared to a baseline CW MCA. Furthermore, the outcome of the experiment is used to identify a possible correlation between the offline evaluation of the objective motion cueing fidelity and the perceived motion cueing fidelity.

6.3.1. Apparatus

For the experiment, the SIMONA Research Simulator (SRS) will be used. Since it will be a passive experiment, that means no control inputs, no aircraft model and control instruments are required. The visual system will be used and will provide the outside visuals, driven by offline-generated flight path data. Furthermore, a PFD will be used that will be extended to additionally present the angle of attack, load factor and a flight freeze indicator. The flight freeze indicator shows whether the upset scenario is in progress, this is necessary since the simulator will be moving before and after the upset scenario to correctly position itself. The motion system will be driven by offline generated input data, linear accelerations in inertial axes and angular accelerations in simulator body axes, which is transferred to the Motion Control Computer (MCC) with the help of a direct motion filter. If too much simulator drift is experienced during testing, another simulator control strategy with simulator state feedback should be implemented. Finally, if the upset scenarios are of sufficient duration, a continuous rating device that can be operated by the subjects will be required.

6.3.2. Independent variables

Motion cueing algorithm: Two different MCAs will be used in the experiment. **a)** *CW MCA*: this will be the baseline motion case, using the SRS CW16 algorithm. The tuning will be similar to Grant and Schroeder [11], Imbrechts et al. [10] or modified tuning of these references and will be depending on the motion scenarios. A modified tuning can be derived with the help of the Gouverneur analysis [33] mentioned in section 2.9. A single CW tuning will be used for all the motion scenarios. **b)** *MPC-based MCA*: this will be the newly proposed MCA as explained in Appendix 3 and used in Appendix 5. Additionally, a terminal state cost term will be included in the cost function.

Cost function weights: Three different weight settings will be used in the experiment and will be the same for the different motion scenarios. **a)** *Specific force*: this includes weights optimised to represent specific forces. **b)** *Angular motion*: this include weights optimised to represent angular motion. **c)** *Balanced*: this includes weights optimised to represent specific forces equally as angular motion.

Prediction strategy: Two different prediction strategies are proposed for the experiment. **a)** *Perfect*: this prediction strategy implies perfect knowledge of the entire reference trajectory. As shown in Appendix 5, with sufficient prediction horizon length, the result of this prediction strategy is very similar to the result of the "Oracle". **b)** *Constant(+)*: this prediction strategy assumes knowledge of the reference signal for the current time interval only. The remainder of the horizon is then filled with a constant reference signal that is equal to the current time interval. If necessary, this strategy should be adjusted to a strategy where only a part of the horizon is constant, then constant but negative/opposite and for the remainder of the horizon zero.

Upset/motion scenario: The number of upset scenarios used in the experiment will be dependent on the upset scenarios that will be obtained from the UPRT flight test data during the final phase of the thesis. As elaborated in Appendix 4, both upsets and stalls are relevant to UPRT. Therefore, it will be desirable to include both upsets and stalls in the motion scenarios. It could also be that the different upset scenarios will be appended to each other, to obtain motion scenarios with a longer duration. The proposed method will then be one motion scenario with multiple upsets and

one motion scenario with multiple stalls.

6.3.3. Dependent variables

Two dependent variables will be acquired:

Continuous subjective rating of perceived motion incongruence: This is a continuous rating provided in real-time by the subjects of the experiment to indicate the difference between the visuals, including the extended PFD, and the provided motion cues. This variable was introduced by Cleij et al. [13] and later used in experiments found in [19, 75, 76, 22].

Overall subjective rating of perceived motion incongruence: This is a single rating provided after each experiment condition. The single rating will be given with the help of the Motion Fidelity Rating (MFR) from Hodge [6].

6.3.4. Control variables

The control variables during the experiment conditions are listed below.

- Prediction horizon length N
- Sample time Δt_n and Δt_k
- Constraints
- Simulator cost weights $\boldsymbol{W_u}$ and $\boldsymbol{W_x}$ (cost on control input and state deviation from neutral)

6.4. Experimental conditions

In Table 6.1, the preliminary experiment conditions are summarised. To check the consistency of the dependent variables, it will be necessary to perform multiple repetitions for each experiment condition. Cleij et al. [22] successfully used three repetitions per condition and this will most likely be the number of repetitions used in this experiment.

Table 6.1: Preliminary experiment conditions.

Condition	Scenario	MCA	Prediction	Weights
1	Upset	CW	-	-
2	Upset	MPC	Perfect	Specific forces
3	Upset	MPC	Perfect	Angular motion
4	Upset	MPC	Perfect	Balanced
5	Upset	MPC	Constant(+)	Specific forces
6	Upset	MPC	Constant(+)	Angular motion
7	Upset	MPC	Constant(+)	Balanced
8	Stall	CW	-	-
9	Stall	MPC	Perfect	Specific forces
10	Stall	MPC	Perfect	Angular motion
11	Stall	MPC	Perfect	Balanced
12	Stall	MPC	Constant(+)	Specific forces
13	Stall	MPC	Constant(+)	Angular motion
14	Stall	MPC	Constant(+)	Balanced

6.5. Experiment participants and matrix

Commercial airline pilots will be asked to participate in the experiment. The pilots will serve as within-participants. The experiment conditions will be provided to the participants in a balanced and alternating order, an example of this is shown in Table 6.2.

Table 6.2: Example of an experiment participant and condition matrix.

Participant	Trial 1	Trial 2	Trial 3	Trial ...
1	1	2	3	4
2	2	3	4	1
3	3	4	1	2
4	4	1	2	3
5	4	3	2	1
6	3	2	1	4
7	2	1	4	3
...	1	4	3	2

6.6. Experiment procedures

The experiment will start with a briefing, followed by one or more training trials. Between some of the experiment trials, one or more breaks will be included. The experiment is concluded with a debriefing.

6.6.1. Briefing

During the briefing of the experiment, an introduction to the experiment will be given that includes the general purpose of the experiment and experiment procedures. It is important to not show what the hypotheses are during the briefing. Furthermore, the continuous incongruence rating, the MFR (Figure 6.1) and MIsery SCale (MISC) (Table 6.3) will be explained.

Table 6.3: 11-point MIsery SCale (MISC), from Bos et al. [9].

Symptoms		MISC
No problems		0
Some discomfort, but no specific symptoms		1
Dizziness, cold/warm, headache, stomach/throat awareness, sweating, blurred vision, yawning, burping, tiredness, salivation, ... but no nausea	Vague	2
	Little	3
	Rather	4
	Severe	5
Nausea	Little	6
	Rather	7
	Severe	8
	Retching	9
Vomiting		10

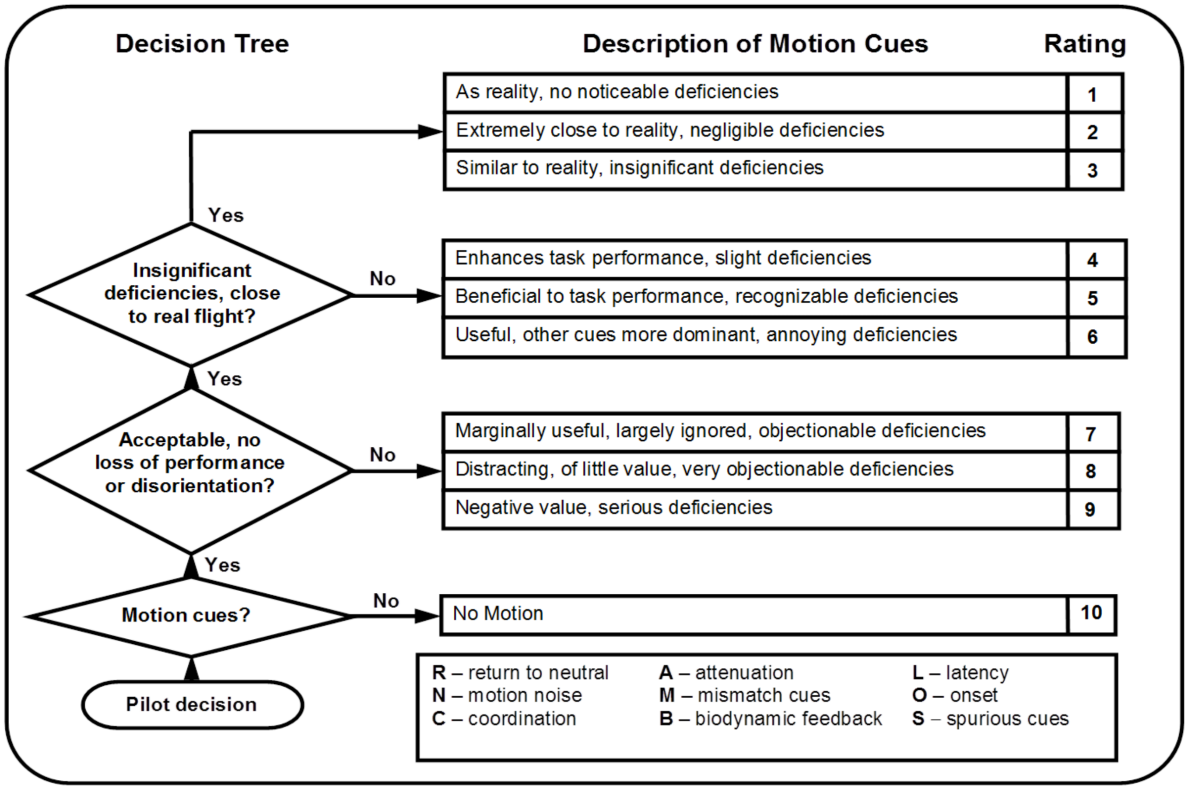


Figure 6.1: Motion Fidelity Rating (MFR), from Hodge et al. [6].

6.6.2. Training

Training is performed by using experiment conditions, similar to the real experimental conditions. This training will be used to make the participants familiar with the use of the continuous rating scale. Furthermore, the participants will be requested to fill in the MFR and MISC ratings. In this way, experiment participants' internal subjective motion incongruence grading scale can be identified. A training procedure used by Cleij et al. [22] consisted of two runs for each experiment condition. In the first run, the participants were asked to indicate the minimum and maximum experienced mismatch and to give an overall rating. In the second run, the participants were asked to give a continuous rating and give the overall rating.

6.6.3. Communication during the experiment

During the experiment, the SRS audio system will be used to communicate with the participants. Before each trial, a confirmation of readiness is requested and a reminder is given to use the continuous rating scale. During the trials, except for the training trials, no communication is preferred. After each trial, some time is given to fill in the MFR and MISC. Finally, the MISC rating is discussed and a decision is made to continue with the next trial or abort the experiment. The maximum MISC rating will be three.

6.7. Experiment data analysis

During the experiment, the independent variables are logged and sorted with respect to the participant numbers. To identify possible deficiencies during the experiment, the SRS motion logger will be used to log the input and output motion data of the SRS. For the analysis of the independent variables, most likely non-parametric tests will be used due to the limited number of experiment participants. Non-parametric tests suitable for within-participant experiments are, for example, the Friedman Test

and Mood's Median Test.

To check the consistency of the continuous rating, the approach from the experiments found in [77, 19, 13, 75, 76, 22] is used. This approach calculates the consistency with Cronbach's alpha [78]. If the Cronbach's alpha is smaller than 0.7 the ratings are considered inconsistent [79] and excluded from the analysis.

6.8. Chapter conclusions

SQ 5. *How can the implementation of an MPC-based MCA during UPRT affect the* ***perceptual*** *motion cueing fidelity with respect to the current SRS CW MCA?*
To evaluate the perceived motion fidelity for the MPC-based MCA in comparison with the SRS CW MCA, a passive experiment in the SRS will be performed.

5a. *What experiment conditions with different MCA settings should be used during the experiment?*
Two types of MCAs, CW and MPC-based. Three cost function weight settings, optimised for specific forces, optimised for angular rates, and finally, a balanced setting will be used. Two prediction strategies, one with perfect knowledge of all samples of the prediction horizon and one with only knowledge of the current sample of the prediction horizon.

5b. *What metrics should be used to evaluate the perceived motion cueing fidelity?*
The continuous subjective rating of perceived motion incongruence and the MFR.

5c. *What type of experiment participants are required for the experiment?*
Experienced commercial airline pilots are desired to participate in the experiment.

5d. *What data should be logged during the experiment?*
During the experiment, the continuous subjective motion incongruence rating, the MFR, the input and output motion data of the SRS, and the MISC will be logged.

Appendix A

Baseline "Oracle" SIMONA Research Simulator optimal control solution

In this appendix, the full solution of the baseline "Oracle" SIMONA Research Simulator (SRS) optimal control problem as described in section 5.3 is presented for the Citation stall scenario from section 5.1. In Figure A.1, the resulting simulator output, the inertial signal, is compared to the reference inertial signal. In Figure A.2 and Figure A.3, the resulting simulator state and control input are presented, respectively. The actuator length, velocity and acceleration are presented in Figure A.4. Finally, the cost function components and total cost are depicted in Figure A.5.

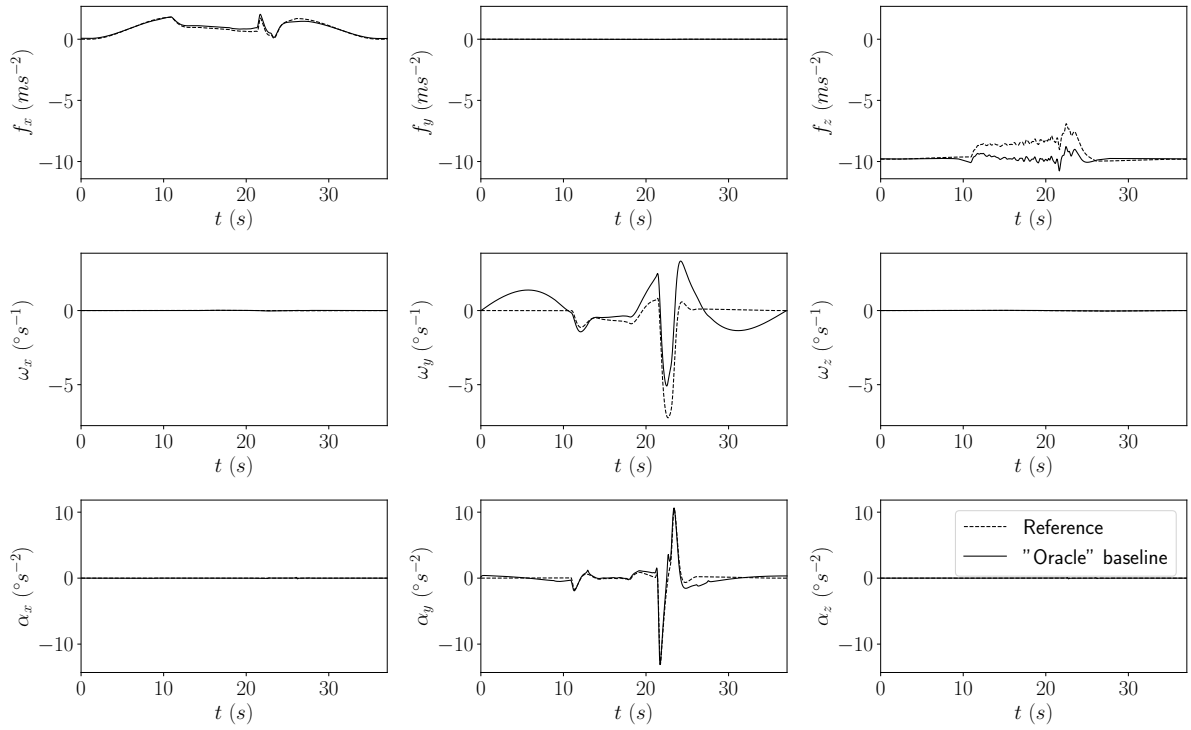


Figure A.1: Baseline "Oracle" output inertial signal and reference inertial signal.

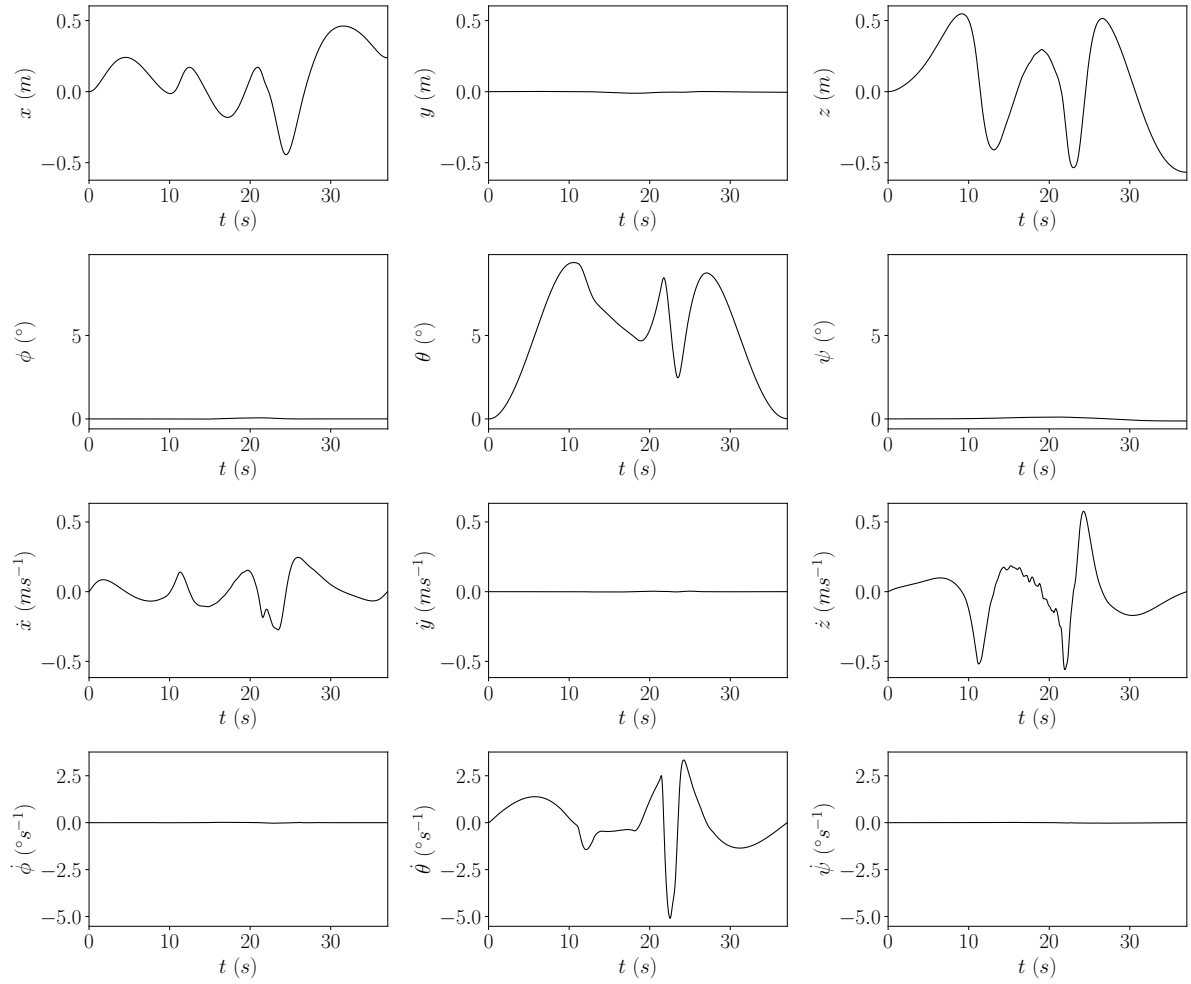


Figure A.2: Baseline "Oracle" state.

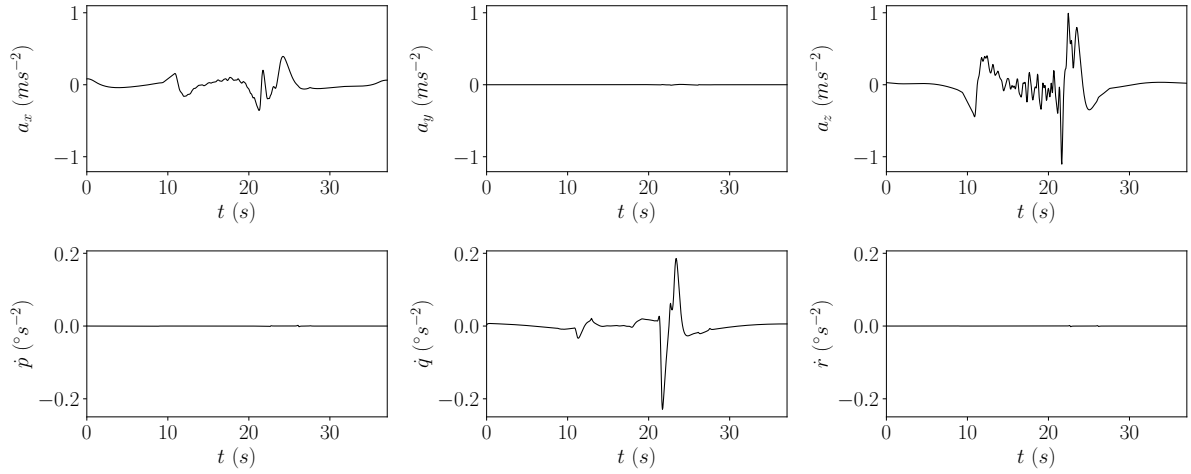


Figure A.3: Baseline "Oracle" control input.

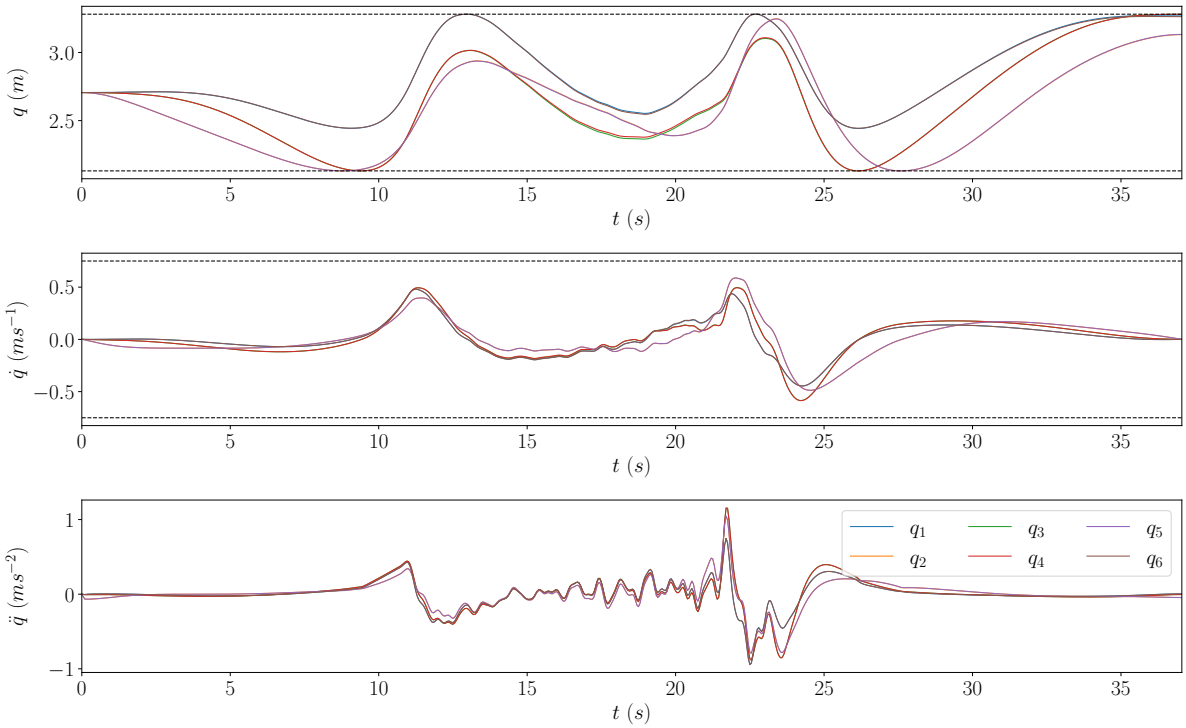


Figure A.4: Baseline ”Oracle” actuator length, velocity and acceleration.

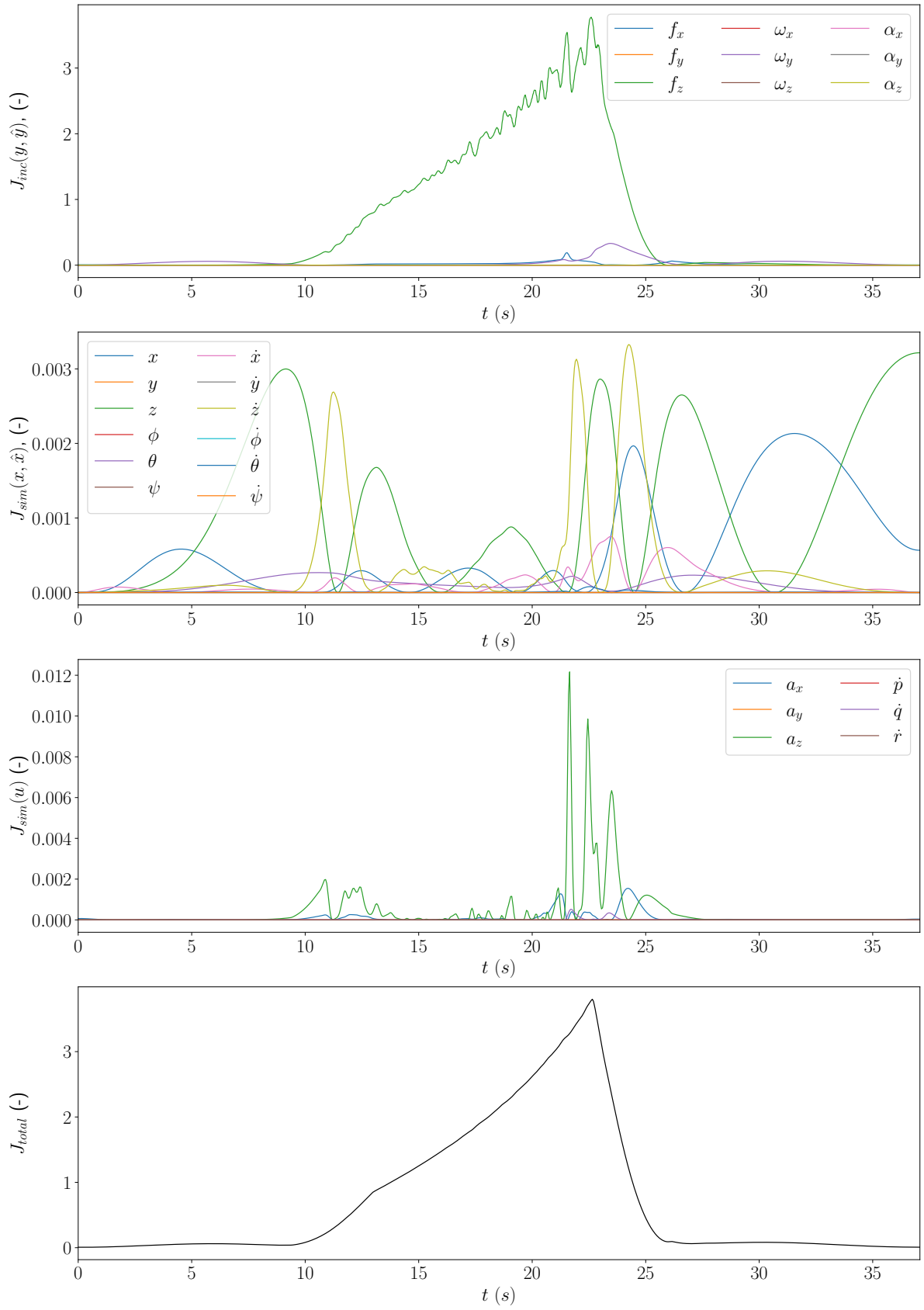


Figure A.5: Baseline "Oracle" output, state, input and total cost.

Appendix B

Reference Classical Washout motion filter settings

In this appendix, the Classical Washout (CW) motion filter settings used in Appendix 5 as reference are presented. In Table B.1 and Table B.2, the filter settings used by Imbrechts et al. [10] and Grant and Schroeder [11] are presented, respectively.

Table B.1: CW filter settings used by Imbrechts et al. [10]

	x	y	z	p	q	r
K	0.5	0.0	0.5	0.0	0.5	0.0
ω_{hp}	1.2	0.0	2.0	0.0	1.0	0.0
ζ_{hp}	0.7	0.0	0.7	0.0	0.5	0.0
ω_b	0.0	0.0	0.3	0.0	0.0	0.0
ω_{lp}	2.4	0.0				
ζ_{lp}	0.7	0.0				

Table B.2: CW filter settings used by Grant and Schroeder [11].

	x	y	z	p	q	r
K	0.4	0.3	0.5	0.6	0.6	0.3
ω_{hp}	2.0	2.0	4.0	0.6	0.5	0.7
ζ_{hp}	0.7	0.7	1.0	0.0	0.0	0.7
ω_b	0.0	0.0	0.5	0.0	0.0	0.0
ω_{lp}	2.0	2.5				
ζ_{lp}	0.7	0.7				

Appendix C

Full results for the prediction strategy analysis

In this appendix, the full solution of the prediction strategy analysis performed in section 5.6 is presented. In Figure C.1, the resulting simulator output inertial signal for the various prediction strategies is compared with the reference inertial signal, the "Oracle" baseline and the reference CW results. In Figure C.2 and Figure C.3, the resulting simulator state and control input are presented, respectively.

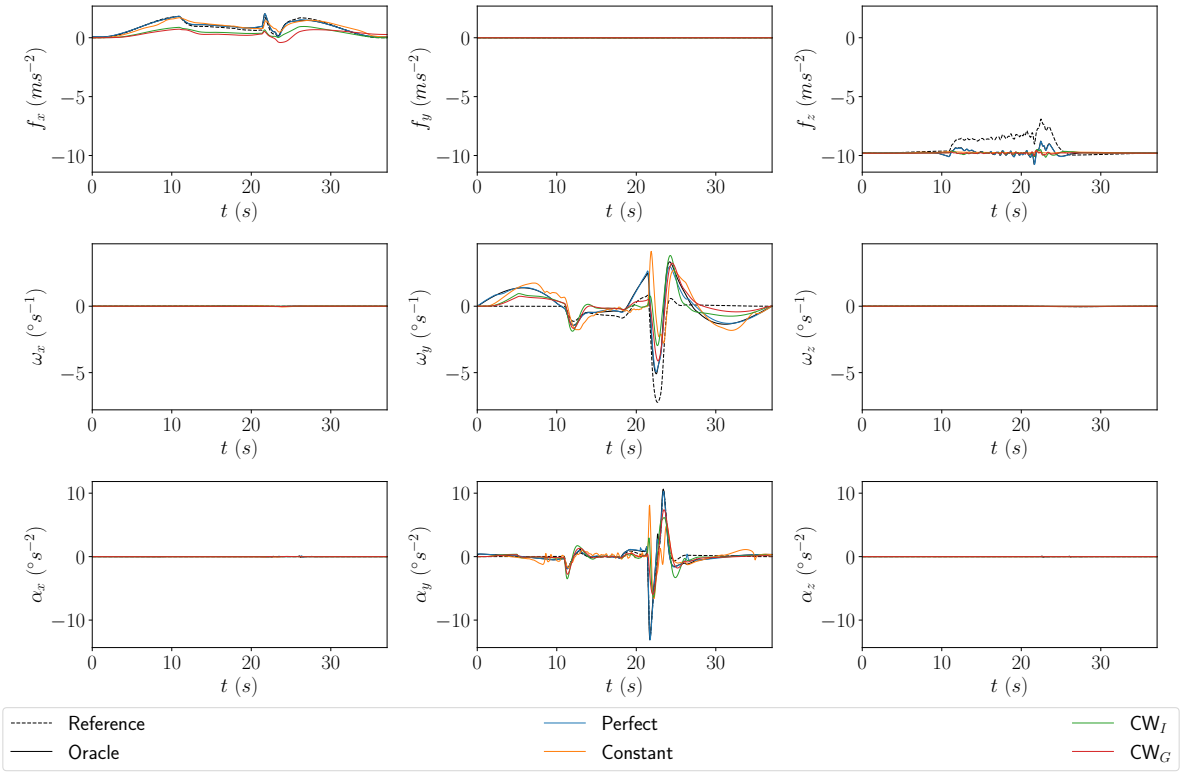


Figure C.1: Inertial signal output for the "Oracle", "Perfect", "Constant", "CW_I", "CW_G" conditions.

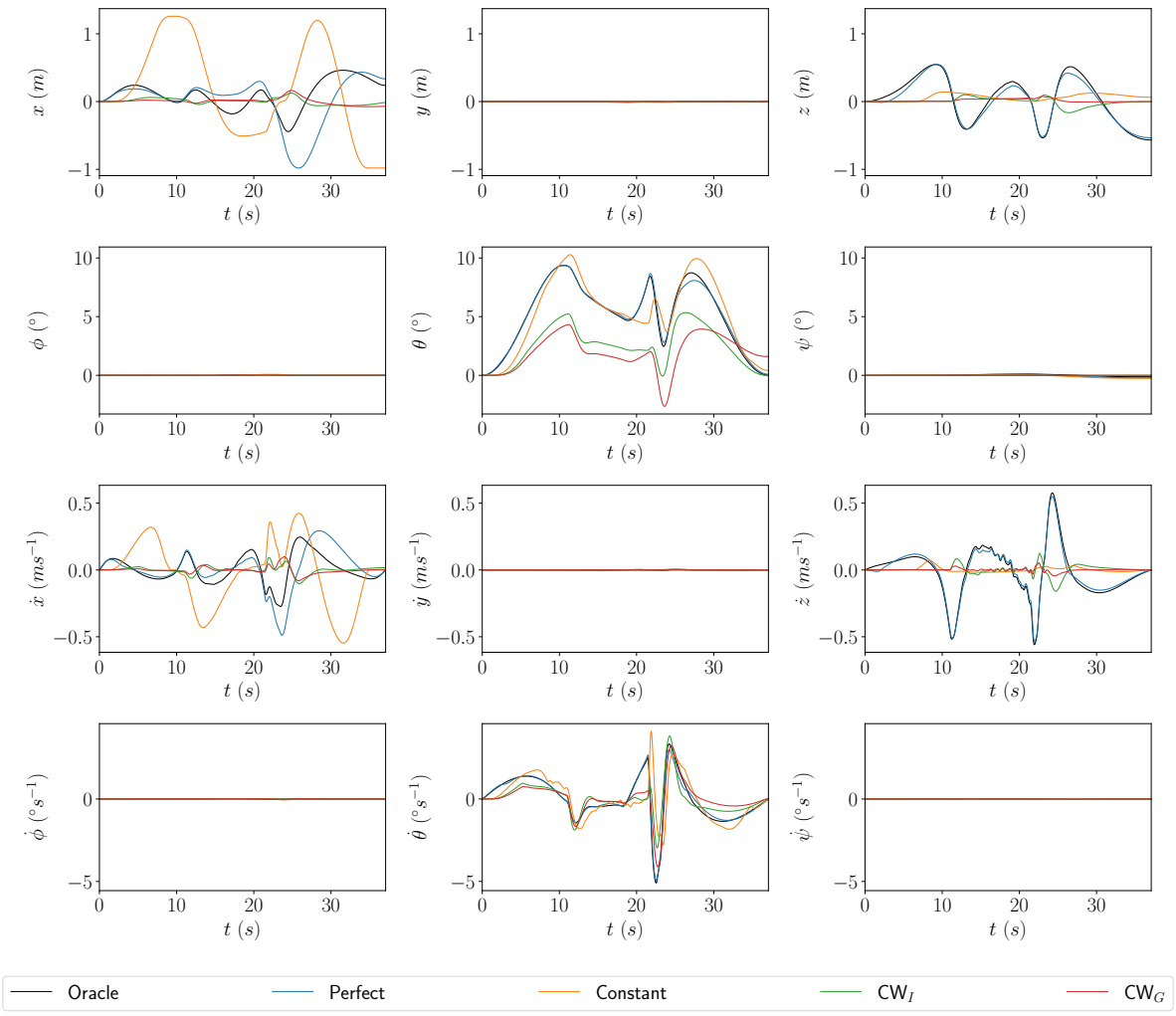


Figure C.2: State for the "Oracle", "Perfect", "Constant", "CW$_I$", "CW$_G$" conditions.

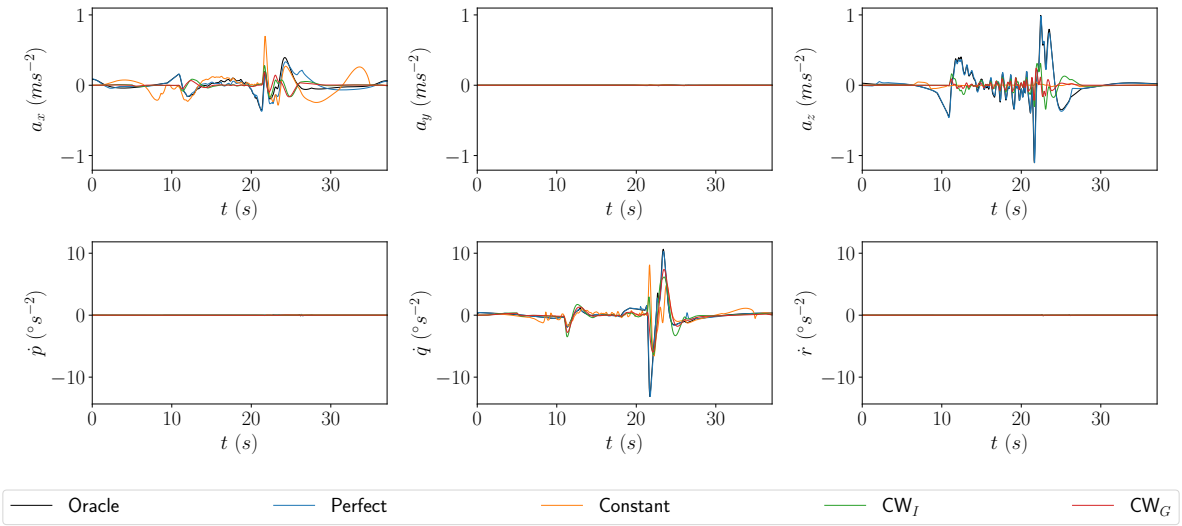


Figure C.3: Control input for the "Oracle", "Perfect", "Constant", "CW$_I$", "CW$_G$" conditions.

Appendix D

Full results for the specific force and angular rate weight sensitivity analysis

In this appendix, the full solution of the specific force and angular rate weight sensitivity analysis performed in section 5.6 is presented.

D.1. Specific force weight sensitivity analysis results

The RMSE and PCC for various specific force weights are presented in Figure D.1. The simulator output inertial signal for the various specific force weights and the reference inertial signal are depicted in Figure D.2. In Figure D.3 and Figure D.4, the resulting simulator state and control input are presented, respectively.

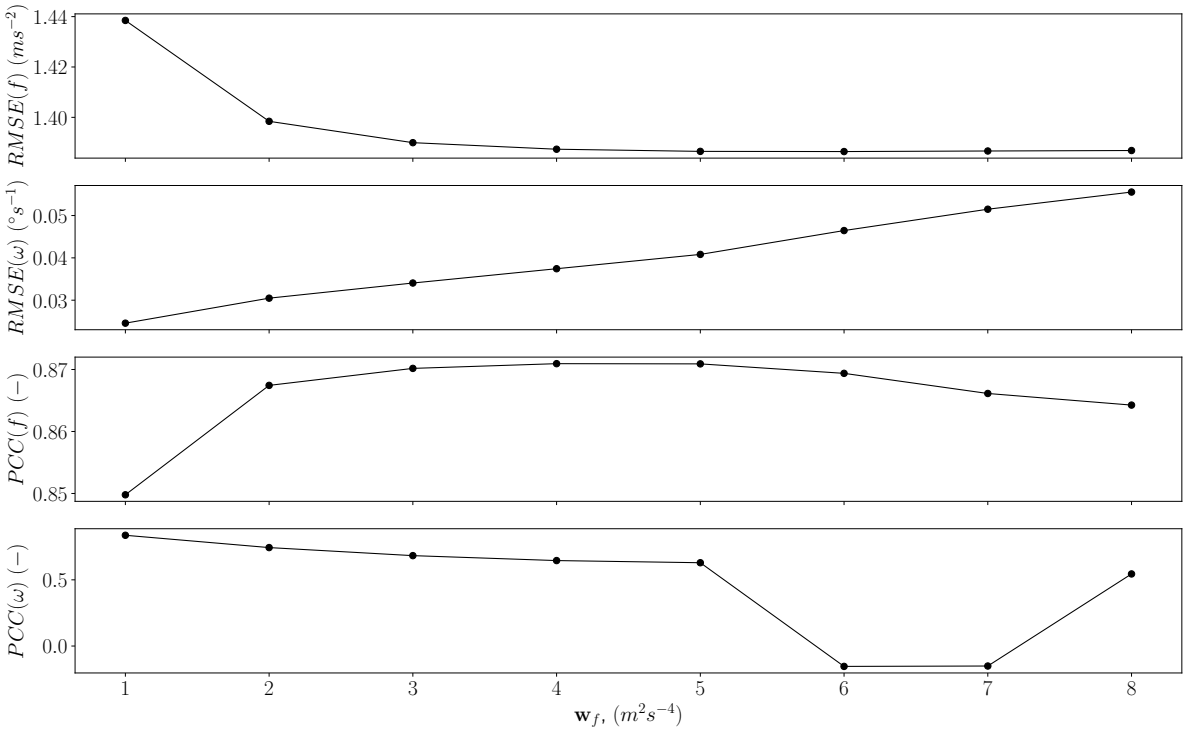


Figure D.1: RMSE and PCC for various specific force weights.

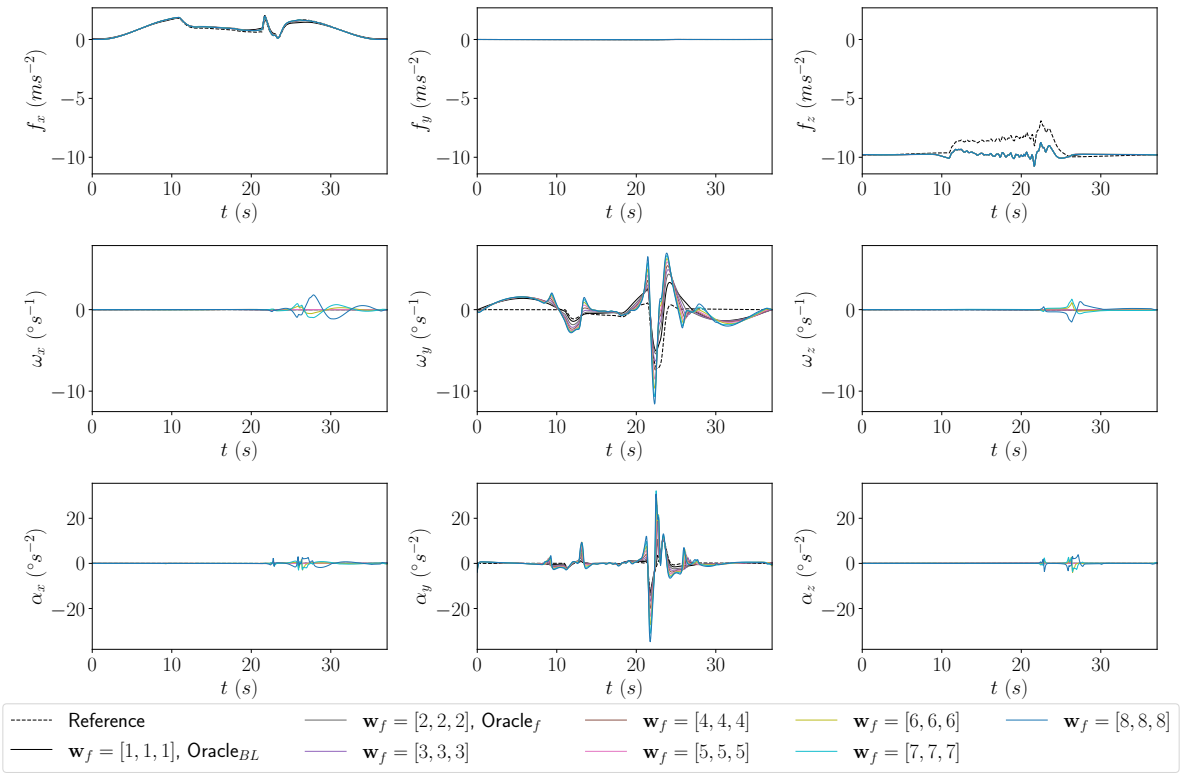


Figure D.2: Inertial signal output for varying specific force weights.

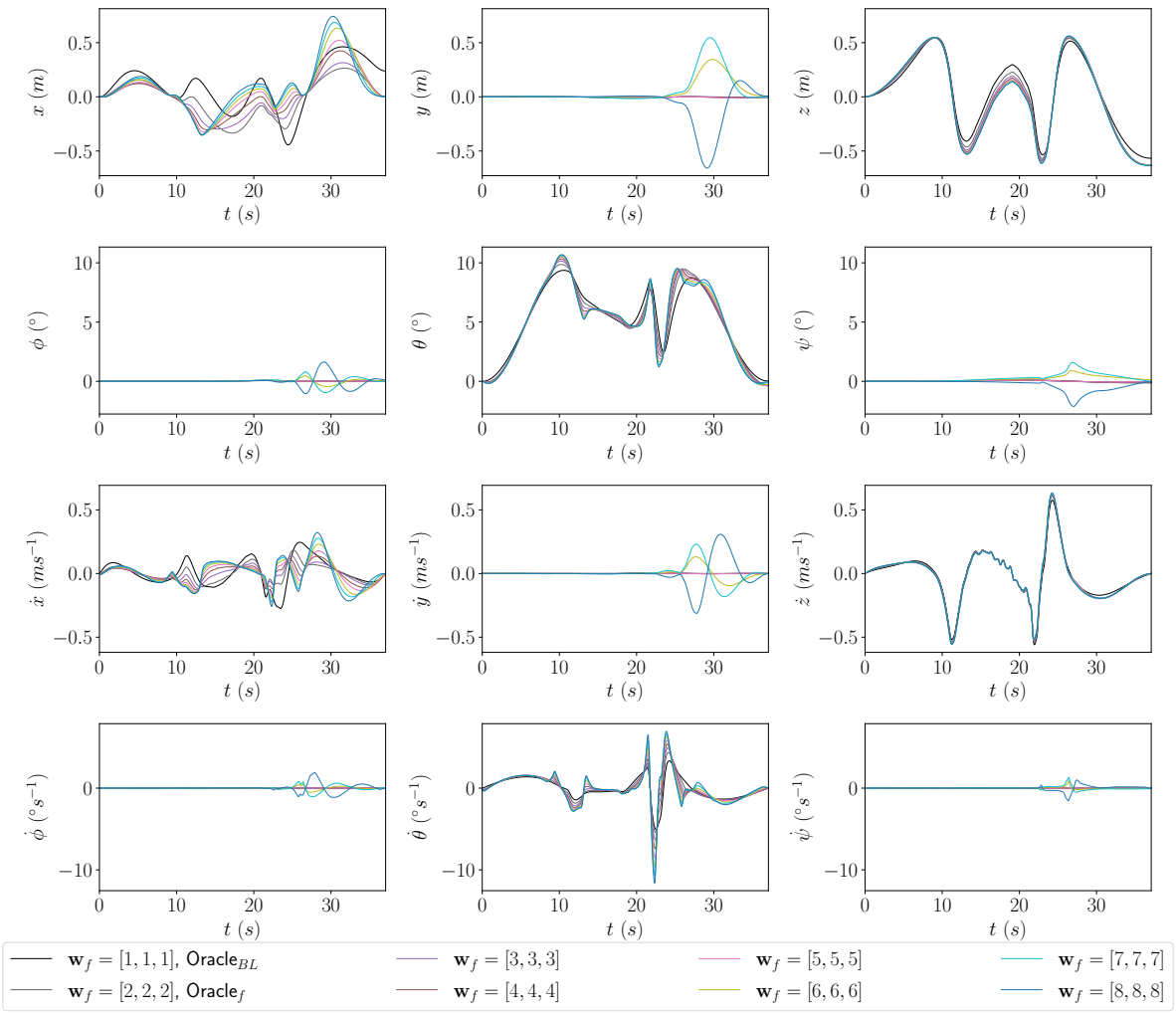


Figure D.3: State for varying specific force weights.

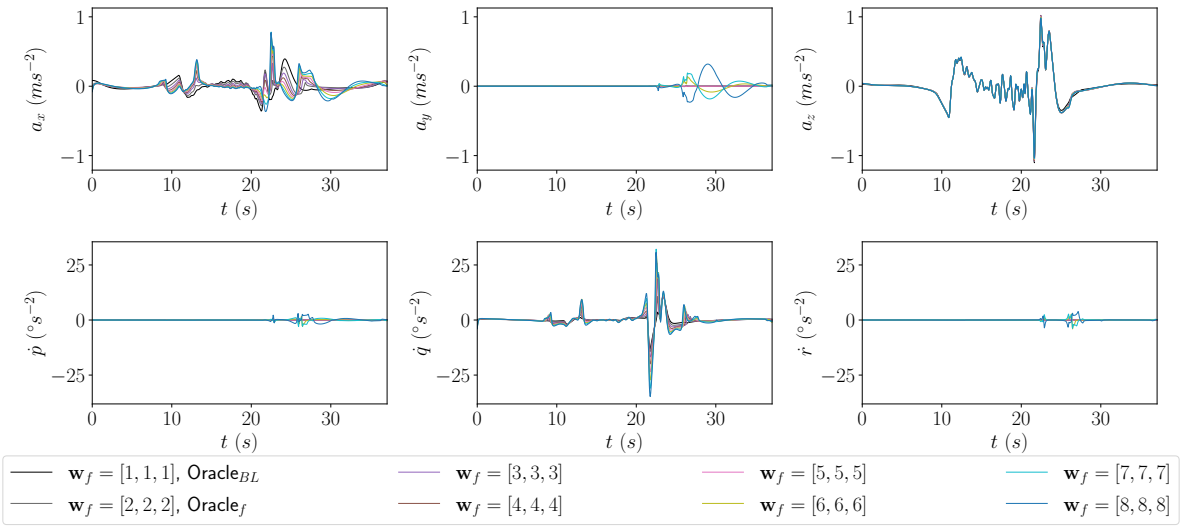


Figure D.4: Control input for varying specific force weights.

D.2. Angular rate weight sensitivity analysis results

The RMSE and PCC for various angular rate weights are presented in Figure D.5. The simulator output inertial signal for the various angular rate weights weights and the reference inertial signal are depicted in Figure D.6. In Figure D.7 and Figure D.8, the resulting simulator state and control input are presented, respectively.

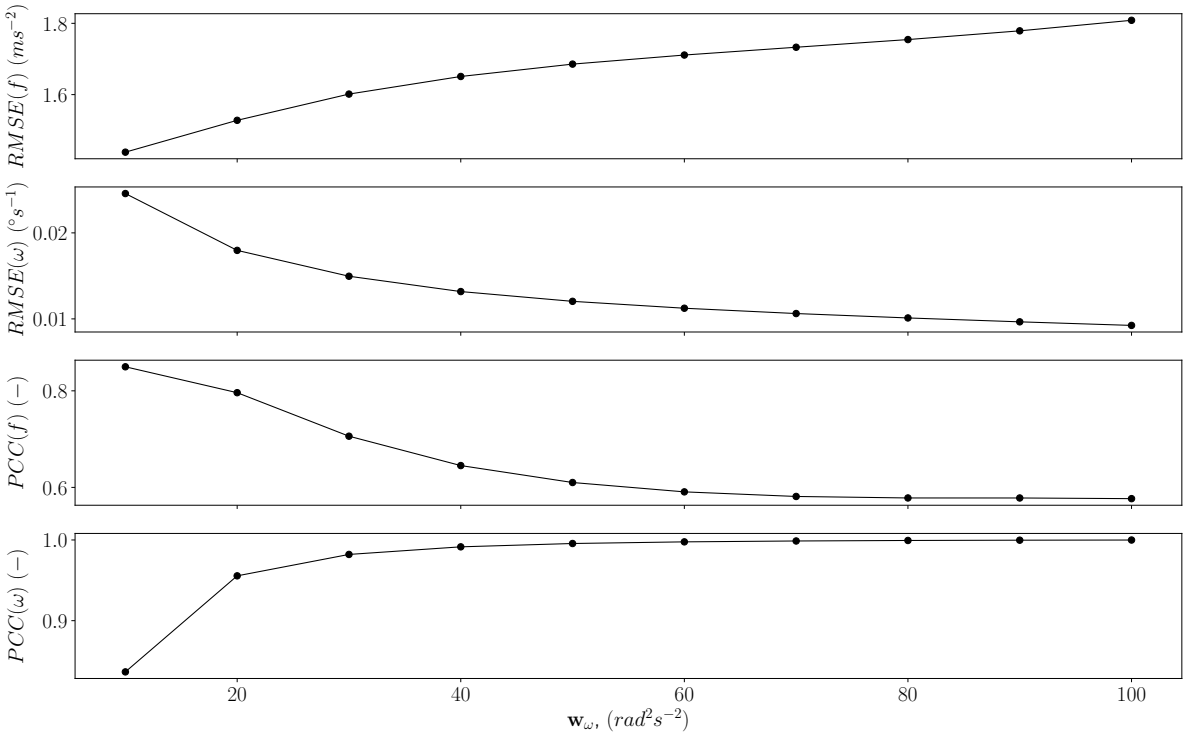


Figure D.5: RMSE and PCC for various angular rate weights.

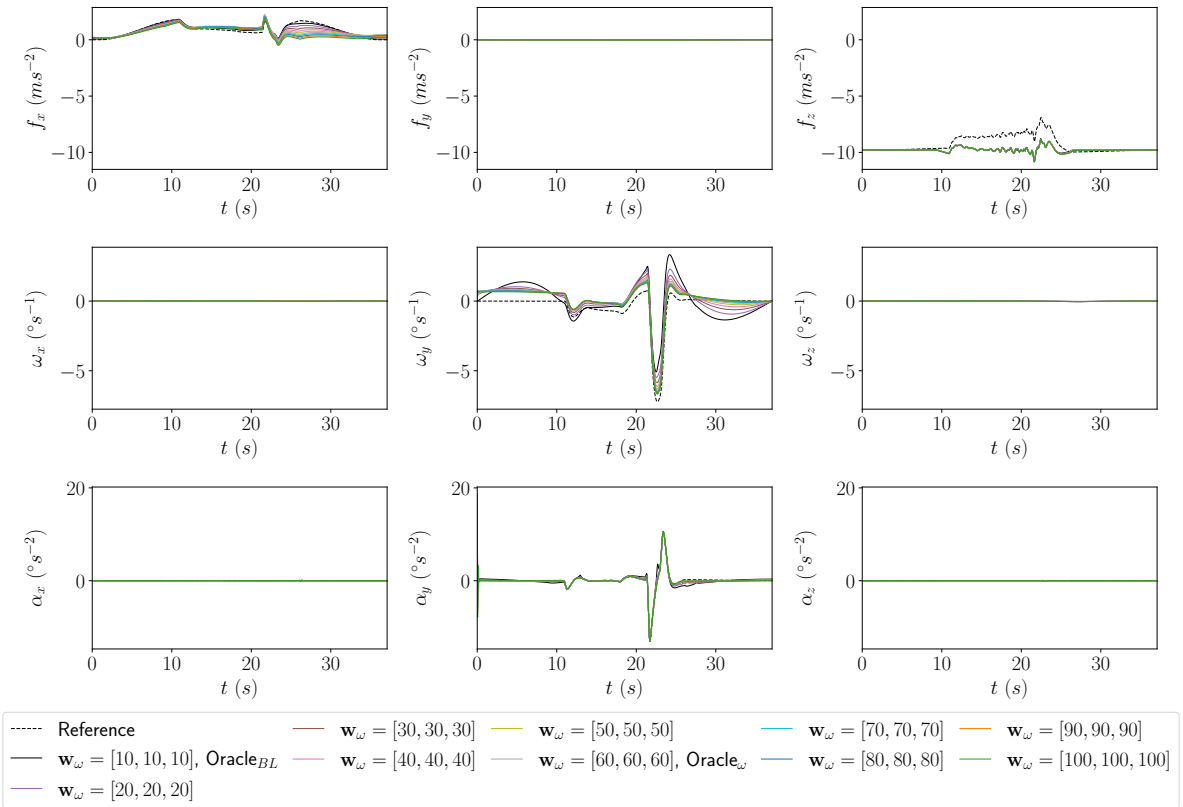


Figure D.6: Inertial signal output for varying angular rate weights.

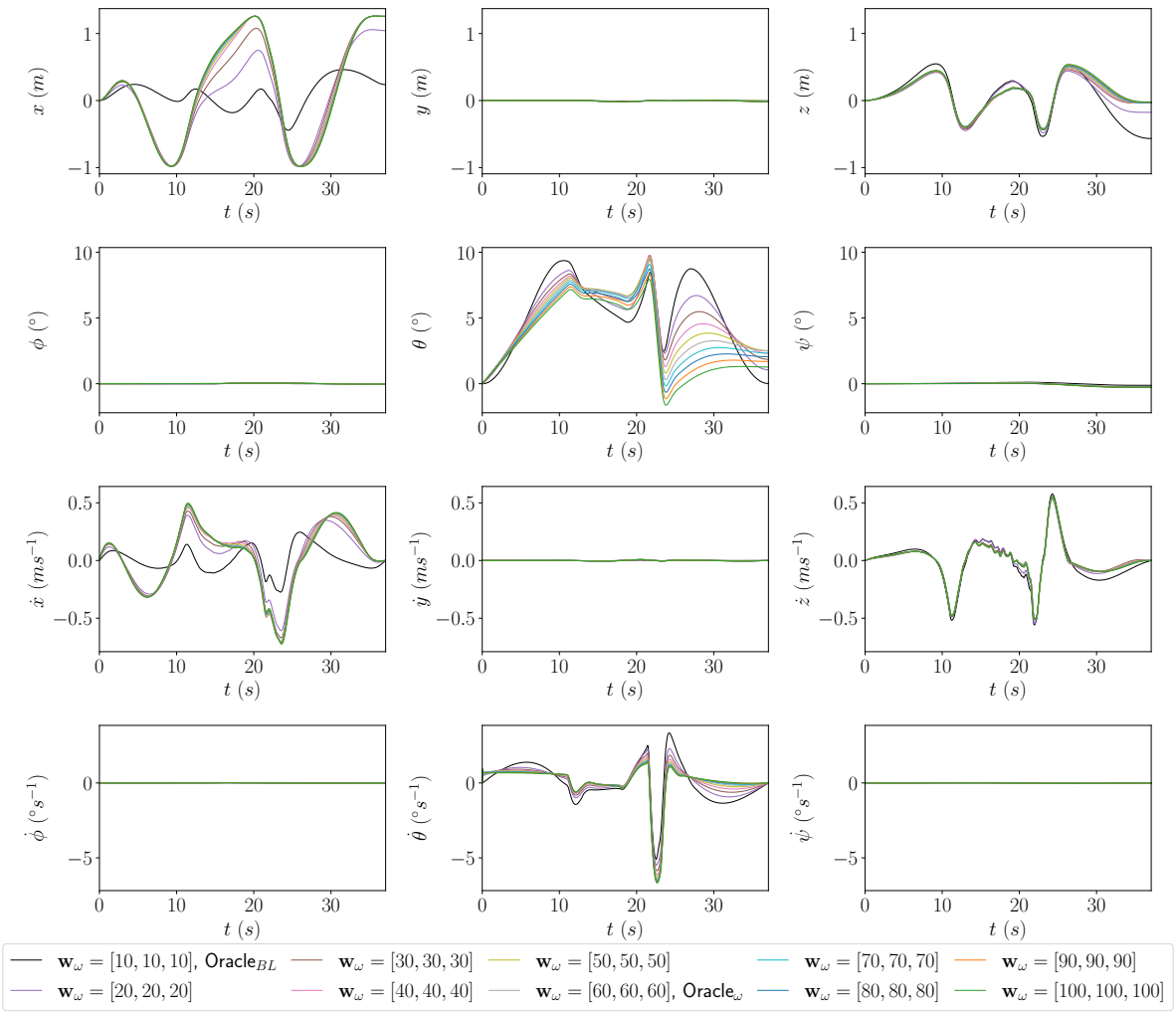


Figure D.7: State for varying angular rate weights.

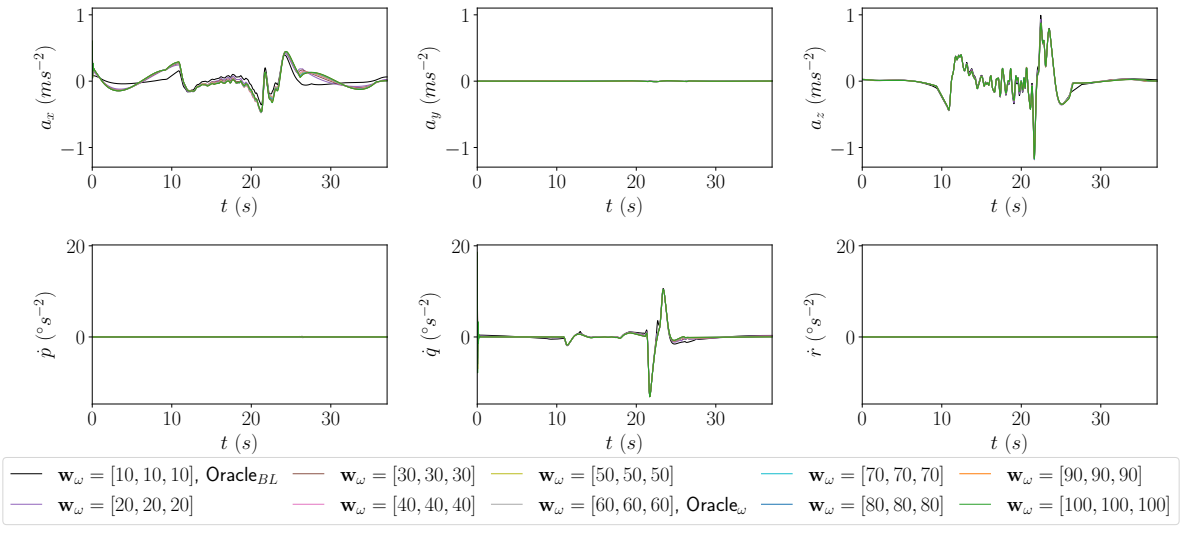


Figure D.8: Control input for varying angular rate weights.

D.3. Full results for the baseline, specific force, angular rate, and reference classical washout conditions

In this section, the chosen conditions resulting from the specific force and angular rate sensitivity analyses are compared to the baseline and CW conditions. The simulator output inertial signal for the various conditions and the reference inertial signal are depicted in Figure D.9. In Figure D.10 and Figure D.11, the resulting simulator state and control input are presented, respectively.

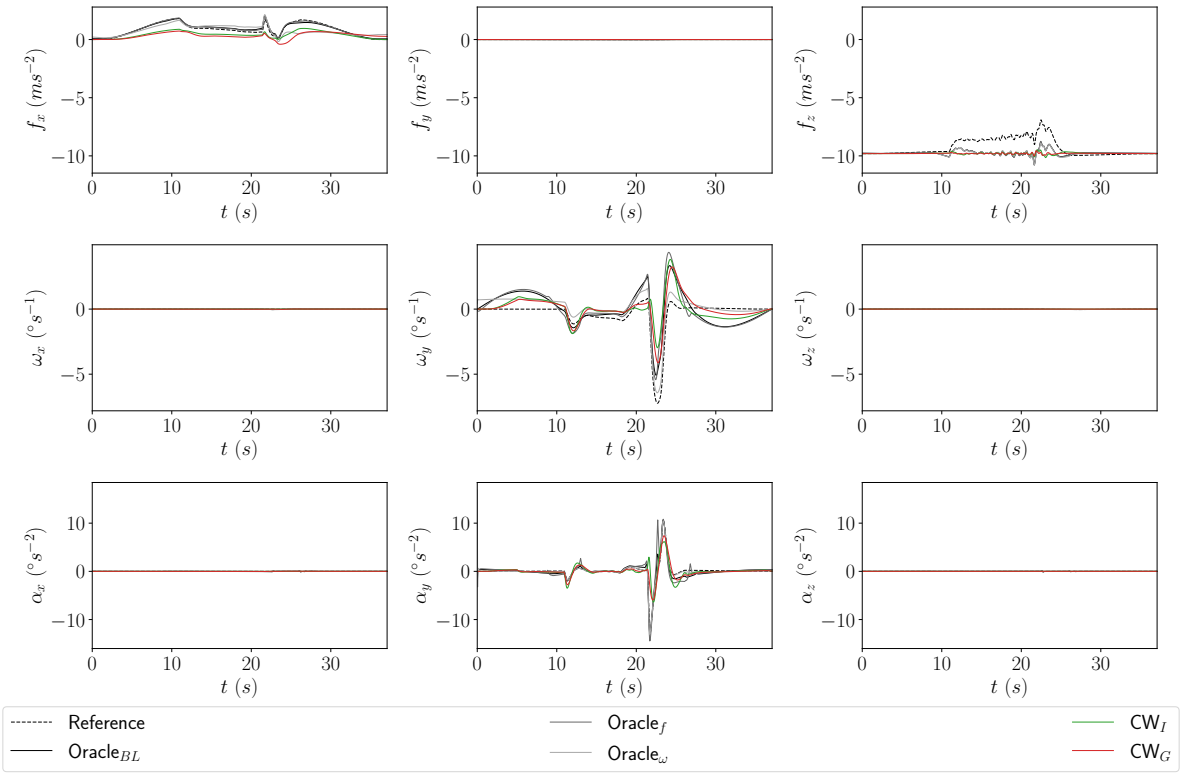


Figure D.9: Inertial signal output for the "Oracle$_{BL}$", "Oracle$_f$", "Oracle$_\omega$", "CW$_I$", "CW$_G$" conditions.

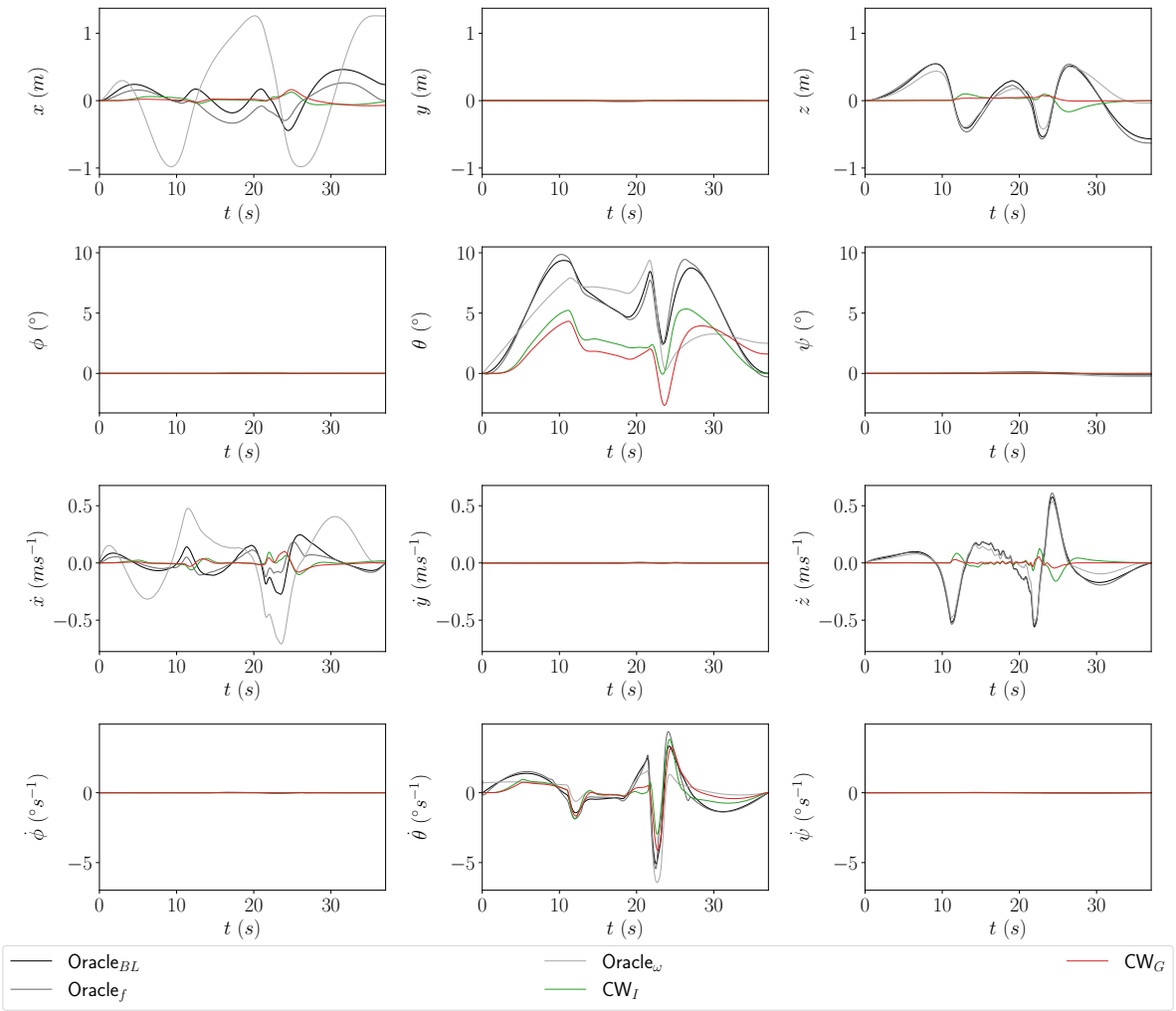


Figure D.10: State for the "Oracle$_{BL}$", "Oracle$_f$", "Oracle$_\omega$", "CW$_I$", "CW$_G$" conditions.

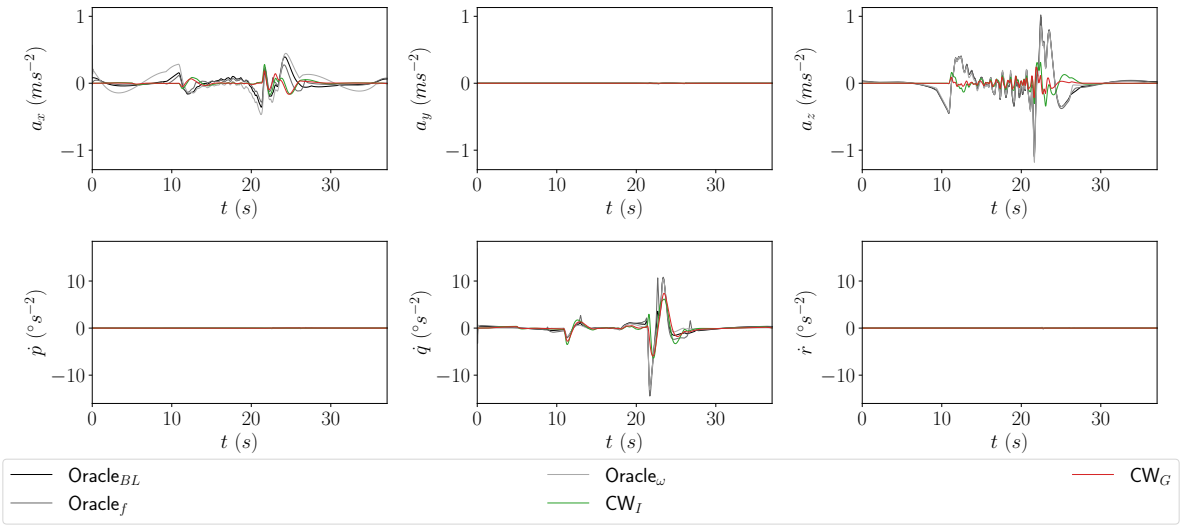


Figure D.11: Control input for the "Oracle$_{BL}$", "Oracle$_f$", "Oracle$_\omega$", "CW$_I$", "CW$_G$" conditions.

Bibliography

[1] D. M. Pool, *Objective evaluation of flight simulator motion cueing fidelity through a cybernetic approach.* PhD thesis, Delft University of Technology, Delft, 2012.

[2] L. D. Reid and M. A. Nahon, "Flight Simulation Motion-Base Drive algorithms: Part 1 - Developing and Testing the Equations," tech. rep., UTIAS, 12 1985.

[3] T. Van Leeuwen, "Simulator motion cueing error detection using a wavelet-based algorithm," tech. rep., Delft University of Technology, Delft, 2017.

[4] S. K. Advani and J. A. Schroeder, "Global implementation of upset prevention & recovery training," in *AIAA Modeling and Simulation Technologies Conference*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2016.

[5] L. Fucke, V. Biryukov, M. Grigorev, V. Rogozin, E. Groen, M. Wentink, J. Field, B. Soemarwoto, N. Abramov, M. Goman, and A. Khrabrovitt, "Developing scenarios for research into upset recovery simulation," in *AIAA Modeling and Simulation Technologies Conference 2010*, American Institute of Aeronautics and Astronautics Inc., 2010.

[6] S. J. Hodge, P. Perfect, G. D. Padfield, and M. D. White, "Optimising the yaw motion cues available from a short stroke hexapod motion platform," *Aeronautical Journal*, vol. 119, pp. 1–21, 1 2015.

[7] D. Cleij, *Measuring, modelling and minimizing perceived motion incongruence for vehicle motion simulation.* PhD thesis, Delft University of Technology, 2020.

[8] W. R. Berkouwer, O. Stroosma, M. M. Van Paassen, M. Mulder, and J. A. Mulder, "Measuring the performance of the SIMONA Research Simulator's motion system," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, vol. 2, 2005.

[9] J. E. Bos, S. N. MacKinnon, and A. J. Patterson, "Motion Sickness Symptoms in a Ship Motion Simulator: Effects of Inside, Outside, and No View," *Aviation Space and Environmental Medicine*, vol. 76, pp. 1111–8, 2006.

[10] A. Imbrechts, C. C. de Visser, and D. M. Pool, "Just Noticeable Differences for Variations in Quasi-Steady Stall Buffet Model Parameters," in *AIAA Scitech Forum*, 2022.

[11] P. R. Grant, G. J. Moszczynski, and J. A. Schroeder, "Post-stall flight model fidelity effects on full stall recovery training," in *Modeling and Simulation Technologies Conference*, 2018.

[12] M. Katliar, M. Olivari, F. M. Drop, S. Nooij, M. Diehl, and H. H. Bülthoff, "Offline motion simulation framework: Optimizing motion simulator trajectories and parameters," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 66, pp. 29–46, 10 2019.

[13] 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*, vol. 48, pp. 17–29, 2 2018.

[14] International Air Transport Association (IATA), "Loss of Control In-Flight Accident Analysis Report Edition 2019," tech. rep., IATA, Montreal, 2019.

[15] European Union Aviation Safety Agency (EASA), "Annex I to ED Decision 2019/005/R, 'AMC and GM to Part-FCL — Issue 1, Amendment 7'," 2019.

[16] J. Field, M. Roza, and H. Smaili, "Developing upset cueing for conventional flight simulators," in *AIAA Modeling and Simulation Technologies Conference*, 2012.

[17] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, "Model-based predictive motion cueing strategy for vehicle driving simulators," *Control Engineering Practice*, vol. 17, pp. 995–1003, 9 2009.

[18] N. J. Garrett and M. C. Best, "Model predictive driving simulator motion cueing algorithm with actuator-based constraints," *Vehicle System Dynamics*, vol. 51, pp. 1151–1172, 8 2013.

[19] J. Venrooij, D. Cleij, M. Katliar, P. Pretto, H. H. Bülthoff, and H.-P. Schöner, "Comparison between filter-and optimization-based motion cueing in the Daimler Driving Simulator," in *DSC 2016 Europe VR*, 2016.

[20] M. Grottoli, D. Cleij, P. Pretto, Y. Lemmens, R. Happee, and H. H. Bülthoff, "Objective evaluation of prediction strategies for optimization-based motion cueing," *Simulation*, vol. 95, pp. 707–724, 8 2019.

[21] J. R. van der Ploeg, D. Cleij, D. M. Pool, M. Mulder, and H. H. B, "Sensitivity Analysis of an MPC-based Motion Cueing Algorithm for a Curve Driving Scenario," in *DSC Europe VR*, 9 2020.

[22] D. Cleij, D. M. Pool, M. Mulder, and H. H. Bülthoff, "Optimizing an Optimization-Based MCA using Perceived Motion Incongruence Models," in *DSC 2020 Europe VR*, 2020.

[23] A. Lamprecht, D. Steffen, K. Nagel, J. Haecker, and K. Graichen, "Online Model Predictive Motion Cueing With Real-Time Driver Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 12414–12428, 8 2022.

[24] J. B. Sinacori, "Piloted Aircraft Simulation Concepts and Overview," tech. rep., Systems Technology, Inc, Mountain View (CA), 1978.

[25] W. F. Heffley, R. K. Clement, R. F. Ringland, W. F. Jewell, H. R. Jex, D. T. McRuer, and V. E. Carter, "Determination of Motion and Visual System Requirements for Flight Training Simulators," tech. rep., U.S. Army Research Institute for the Behavioral and Social Sciences, 1981.

[26] S. Advani, *The Kinematic Design of Flight Simulator Motion-Bases*. PhD thesis, Delft University of Technology, Delft, 1998.

[27] R. V. Parrish, J. E. Dieudonne, R. L. Bowles, and D. J. Martin, "Coordinated Adaptive Washout for Motion Simulators," *Journal of Aircraft*, vol. 12, pp. 44–50, 1 1975.

[28] 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.

[29] J. Sinacori, "The determination of some requirements for a helicopter flight research simulation facility," tech. rep., NASA, 9 1977.

[30] J. A. Schroeder, "Helicopter Flight Simulation Motion Platform Requirements," tech. rep., NASA, Moffett Field, 1999.

[31] S. K. Advani and R. Hosman, "Towards Standardising High-Fidelity Cost-Effective Motion Cueing in Flight Simulation," in *Royal Aeronautical Society Conference on: Cutting Costs in Flight Simulation. Balancing Quality and Capability*, (London), 2006.

[32] R. Hosman and S. Advani, "Design and evaluation of the objective motion cueing test and criterion," *Aeronautical Journal*, vol. 120, pp. 873–891, 5 2016.

[33] B. Gouverneur, J. B. Bob Mulder, M. M. René van Paassen, O. Stroosma, and E. J. Field, "Optimisation of the SIMONA research simulator's motion filter settings for handling qualities experiments," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, American Institute of Aeronautics and Astronautics Inc., 2003.

[34] M. Katliar, J. Fischer, G. Frison, M. Diehl, H. Teufel, and H. H. Bülthoff, "Nonlinear Model Predictive Control of a Cable-Robot-Based Motion Simulator," *IFAC-PapersOnLine*, vol. 50, pp. 9833–9839, 7 2017.

[35] M. Katliar, F. M. 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 *European Control Conference*, 2018.

[36] Y. R. Khusro, Y. Zheng, M. Grottoli, and B. Shyrokau, "MPC-Based Motion-Cueing Algorithm for a 6-DOF Driving Simulator with Actuator Constraints," *Vehicles*, vol. 2, pp. 625–647, 12 2020.

[37] R. B. Vinter, *Optimal Control*. Boston: Birkhäuser, 2000.

[38] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design 2nd Edition*. Madison, Wisconsin: Nob Hill Publishing, 2017.

[39] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[40] P. D. Christofides and P. Daoutidis, "Feedback control of two-time-scale nonlinear systems," *International Journal of Control*, vol. 63, pp. 965–994, 3 1996.

[41] T. Geyer, "Computationally efficient model predictive direct torque control," *IEEE Transactions on Power Electronics*, vol. 26, no. 10, pp. 2804–2816, 2011.

[42] C. K. Tan, M. J. Tippett, and J. Bao, "Model predictive control with non-uniformly spaced optimization horizon for multi-timescale processes," *Computers and Chemical Engineering*, vol. 84, pp. 162–170, 1 2016.

[43] Y. Chen, N. Scarabottolo, M. Bruschetta, and A. Beghi, "Efficient move blocking strategy for multiple shooting-based non-linear model predictive control," *IET Control Theory and Applications*, vol. 14, pp. 343–351, 1 2020.

[44] M. Katliar, K. de Winkel, J. Venrooij, P. Pretto, and H. Bülthoff, "Impact of MPC Prediction Horizon on Motion Cueing Fidelity," in *DSC 2015 Europe*, 2015.

[45] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A decoupled linear model predictive control-based motion cueing algorithm for simulation-based motion platform with limitted workspace," in *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2019-February, pp. 35–41, Institute of Electrical and Electronics Engineers Inc., 2 2019.

[46] M. R. Qazani, H. Asadi, and S. Nahavandi, "A Model Predictive Control-Based Motion Cueing Algorithm with Consideration of Joints' limitationsfor HexapodMotion Platform," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, (Bari), IEEE, 10 2019.

[47] M. R. C. Qazani, H. Asadi, S. Khoo, and S. Nahavandi, "A Linear Time-Varying Model Predictive Control-Based Motion Cueing Algorithm for Hexapod Simulation-Based Motion Platform," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, pp. 6096–6110, 10 2021.

[48] A. S. Bukal, B. Haycock, and P. R. Grant, "An adaptive model predictive control based motion drive algorithm," in *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2019.

[49] E. Groen, W. Ledegang, J. Field, H. Smaili, M. Roza, L. Fucke, S. Nooij, M. Goman, M. Mayrhofer, L. Zaichik, M. Grigoryev, and V. Biryukov, "SUPRA - Enhanced upset recovery simulation," in *AIAA Modeling and Simulation Technologies Conference 2012*, 2012.

[50] S. J. Advani and J. N. Field, "Upset Prevention and Recovery Training in Flight Simulators," tech. rep., National Aerospace Laboratory NLR, Amsterdam, 2011.

[51] W. W. Chung, "A preliminary investigation of achievable motion cueing in ground-based flight simulators for upset recovery maneuvers," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, (Honolulu, Hawaii), 2008.

[52] L. Zaichik, Y. Yashin, and P. Desyatnik, "Motion fidelity criteria for large-amplitude tasks," in *AIAA Modeling and Simulation Technologies Conference*, 2009.

[53] F. Liu and P. R. Grant, "Ground-based simulation of airplane upset recovery," in *AIAA Modeling and Simulation Technologies Conference*, American Institute of Aeronautics and Astronautics Inc., 2009.

[54] F. Liu and P. R. Grants, "Ground-based simulation of airplane upset recovery using an enhanced aircraft model," in *AIAA Modeling and Simulation Technologies Conference 2010*, American Institute of Aeronautics and Astronautics Inc., 2010.

[55] S. F. Ko and P. R. Grant, "Development and testing of an adaptive motion drive algorithm for upset recovery training," in *AIAA Modeling and Simulation Technologies Conference*, 2012.

[56] L. E. Zaichik, Y. P. Yashin, P. A. Desyatnik, and H. Smaili, "Some aspects of upset recovering simulation on hexapod simulators," in *AIAA Modeling and Simulation Technologies Conference*, 2012.

[57] N. B. Abramov, M. G. Goman, A. N. Khrabrov, E. N. Kolesnikov, L. Fucke, B. Soemarwoto, and H. Smaili, "Pushing ahead - SUPRA airplane model for upset recovery," in *AIAA Modeling and Simulation Technologies Conference 2012*, 2012.

[58] J. A. Schroeder, J. Bürki-Cohen, J. A. Volpe, D. A. Shikany, D. R. Gingras, and P. Desrochers, "An Evaluation of Several Stall Models for Commercial Transport Training," in *AIAA Modeling and Simulation Technologies Conference*, 2014.

[59] P. R. Grant, S. Liu, Z. Z. Luo, and G. Moszczynski, "Development of post-stall flight models from certification flight test and wind-tunnel data," in *AIAA Modeling and Simulation Technologies Conference, 2017*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2017.

[60] L. E. Zaichik, Y. P. Yashin, P. A. Desyatnik, and Y. A. Arkhangelsky, "Motion cueing fidelity in upset recovery simulation," in *AIAA Scitech Forum*, 2019.

[61] P. M. T. Zaal, "Motion cueing for stall recovery training in commercial transport simulators," in *AIAA Scitech Forum*, 2019.

[62] European Union Aviation Safety Agency (EASA), "Certification Specifications for Aeroplane Flight Simulation Training Devices 'CS-FSTD(A)'," 2018.

[63] International Civil Aviation Organization (ICAO), "Airplane Upset Prevention and Recovery Training Aid (REV 3)," 2 2017.

[64] International Air Transport Association (IATA), "Guidance Material and Best Practices for the Implementation of Upset Prevention and Recovery Training, 2nd Edition," tech. rep., IATA, Montreal, 2018.

[65] European Union Aviation Safety Agency (EASA), "Explanatory Note to Decision 2019/005/R," 2019.

[66] J. C. de Winter, D. Dodou, and M. Mulder, "Training Effectiveness of Whole Body Flight Simulator Motion: A Comprehensive Meta-Analysis," *International Journal of Aviation Psychology*, vol. 22, pp. 164–183, 4 2012.

[67] P. M. T. Zaal and X. Mobertz, "Effects of Motion Cues on the Training of Multi-Axis Manual Control Skills," in *AIAA Modeling and Simulation Technologies Conference*, 6 2017.

[68] D. Wu, Z. Zhang, and J. Zhao, "Enhance Upset Prediction and Recovery Training Motion Cueing Using Switched Model Predictive Control and Pilot Model," in *Proceedings of 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology, ICCASIT 2020*, pp. 1083–1087, Institute of Electrical and Electronics Engineers Inc., 10 2020.

[69] D. Wu, J. Zhao, and H. Gu, "A Time Varying Model Predictive Motion Cueing Algorithm for Full Flight Simulator Upset Recovery Training," in *Proceedings of 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology, ICCASIT 2021*, pp. 1127–1131, Institute of Electrical and Electronics Engineers Inc., 2021.

[70] S. C. Smets, C. C. de Visser, and D. M. Pool, "Subjective noticeability of variations in quasi-steady aerodynamic stall dynamics," in *AIAA Scitech Forum*, 2019.

[71] M. Katliar, *Optimal control of motion simulators*. PhD thesis, Albert-Ludwigs-Universität Freiburg, Freiburg, 2021.

[72] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 3 2019.

[73] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 5 2006.

[74] O. Stroosma, "The SIMONA Research Simulator's Motion Software," 6 2022.

[75] 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," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 61, pp. 53–68, 2 2019.

[76] 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*, vol. 61, pp. 84–92, 2 2019.

[77] M. Lambooij, W. A. IJsselsteijn, and I. Heynderickx, "Visual discomfort of 3D TV: Assessment methods and modeling," *Displays*, vol. 32, pp. 209–218, 10 2011.

[78] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, 1951.

[79] J. F. Hair, *Multivariate Data Analysis: A Global Perspective*. Saddle River: Prentice Hall, 7th ed., 2009.