AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-0978

Comparison of Classical and Optimization-based Motion Cueing for Simulating Aircraft Upset Maneuvers

Roemer M.B. Bakker*
Royal Netherlands Aerospace Centre, Amsterdam, Noord-Holland, 1059 CM

Olaf Stroosma†, Daan M. Pool‡, Marinus M. van Paassen§, and Max Mulder¶
Faculty of Aerospace Engineering, Delft University of Technology, 2600 GB Delft, The Netherlands

Providing adequate simulator motion cues for simulated upset and stall scenarios remains challenging. This paper evaluates the potential of novel optimization-based motion cueing algorithms for upset and stall simulation. An offline analysis is performed to compare three Model Predictive Control (MPC) algorithms with varying prediction horizon lengths and prediction strategies (i.e., "Oracle", "Perfect", and "Constant") against two baseline classical washout algorithm implementations from literature. The analysis is performed for a symmetric stall scenario flown with TU Delft's Cessna Citation II laboratory aircraft. Overall, the analysis shows that the objective motion cueing quality expressed in terms of the Root Mean Square Error (RMSE) improves by 29.8% (for specific forces) and 18.7% (for rotational velocities) with the "Oracle" and "Perfect" MPC implementations compared to the reference classical washout results. For the "Constant" MPC algorithm, which in fact does not include any explicit prediction across the MPC algorithm's future prediction window, only a marginal improvement in motion quality was found. Overall, these results imply that, assuming a sufficient future reference motion prediction can be achieved, optimization-based motion cueing algorithms have the potential to provide significantly better motion cueing quality compared to classical motion cueing algorithms.

Nomenclature

f	=	specific force, m/s^2	$\mathbf{w}$	=	weight vector, -
$\boldsymbol{f}$	=	specific force vector, m/s^2	$\boldsymbol{W}$	=	weight matrix, -
f_c	=	continuous time function, -	$\boldsymbol{x}$	=	simulator state, m and deg and m/s and deg/s
h	=	height, m	$\boldsymbol{x}_r$	=	simulator position substate, m and deg
J	=	cost function, -	$\boldsymbol{x}_v$	=	simulator velocity substate, m/s and deg/s
K_{cw}	=	motion filter gain, -	$\boldsymbol{y}$	=	simulator inertial signal, m/s^2 and deg/s and deg/s^2
q	=	pitch rate, deg/s	$\hat{\boldsymbol{y}}$	=	reference inertial signal, m/s^2 and deg/s and deg/s^2
q_i	=	simulator's ith actuator length, m	α	=	angle of attack, deg
$\boldsymbol{q}$	=	simulator actuator length vector, m	$\boldsymbol{\alpha}$	=	angular acceleration vector, rad/s^2
t	=	time, s	ζ_{hp}	=	second-order high-pass motion filter damping, -
t_s	=	sampling time, s	ζ_{lp}	=	second-order low-pass motion filter damping, -
T_H	=	prediction horizon length, s	ω	=	angular rate, rad/s or deg/s
T_S	=	scenario length, s	$\boldsymbol{\omega}$	=	angular rate vector, rad/s or deg/s
$\boldsymbol{u}$	=	simulator control input, m/s^2 and deg/s	ω_b	=	first-order high-pass motion filter frequency, rad/s
V_{IAS}	=	indicated airspeed, m/s	ω_{hp}	=	second-order high-pass motion filter frequency, rad/s
w	=	weight, -	ω_{lp}	=	second-order low-pass motion filter frequency, rad/s

*Research Engineer, Aerospace Operations division, Training and Simulation department, roemer.bakker@nlr.nl
†Senior Researcher, Control and Operations department, Control and Simulation section, o.stroosma@tudelft.nl, Senior Member AIAA.
‡Assistant Professor, Control and Operations department, Control and Simulation section, d.m.pool@tudelft.nl, Associate Fellow AIAA.
§Associate Professor, Control and Operations department, Control and Simulation section, m.m.vanpaassen@tudelft.nl.
¶Professor, Control and Operations department, Control and Simulation section, m.mulder@tudelft.nl, Associate Fellow AIAA.

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]. These resulted in the current Upset Prevention and Recovery Training (UPRT) syllabus that is part of the pilot training curriculum, as formalized by the European Union Aviation Safety Agency [4] and Federal Aviation Administration [5].

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) [6]. 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 [7]. One of the difficulties of UPRT in FSTDs is providing realistic motion cues in the simulator [8]. The challenges of upset and stall motion cueing arise mainly due to the much larger amplitude and longer duration aircraft cues compared to normal flight, potentially leading to larger and sustained specific forces and angular rates needing to be presented. The first challenge is to leave sufficient motion space for other cues when sustained specific forces are cued [9]. The second challenge is the minimization of false specific force cues when large-amplitude angular motion is cued [9, 10]. The third and final challenge is to minimize the false cues that inevitably occur due to high-frequency washout of a CW MCA [10]. Research intended to improve motion cueing for upset and stall simulation [8–14] has so far not led to significant industry changes or moving away from the CW MCA.

A current trend in the automotive simulator research industry is the use of MCAs based on Model Predictive Control (MPC) [15–22]. 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 motions and better utilize the available simulator motion space, thereby potentially enhancing motion cueing fidelity. However, this type of MCA is not yet applied to flight simulation or upset and stall simulation in particular.

(a) Cessna Citation II laboratory aircraft (PH-LAB). **(b) The SIMONA Research Simulator (SRS).**

Fig. 1 TU Delft's main research facilities considered in this paper.

This paper evaluates the potential and usefulness of optimization-based MCAs that implement MPC for use in upset and stall simulation, with as its main objective to improve motion cueing fidelity. In this paper, this is done for the example test case of replicating quasi-steady symmetric stall maneuvers flown with TU Delft's Cessna Citation II laboratory aircraft, see Fig. 1a in TU Delft's SIMONA Research Simulator (SRS), see Fig. 1b. For this, an offline MPC-based MCA is implemented, whose resulting simulator motion is compared to two reference CW MCAs in an objective motion cueing fidelity analysis. Since this approach solves the MPC problem in an offline fashion, as the total duration of the reference motion is known, the most optimal motion cueing improvements attainable with optimization-based cueing are explored. Three versions of the MPC-based MCA are compared in the analysis: two algorithms that assume perfect knowledge of the future reference motion for either the entire scenario ('Oracle') or

across a limited 6-second prediction horizon ('Perfect'), while the third assumes no changes in the future reference ('Constant'). The root mean square error (RMSE) and Pearson's correlation coefficient (PCC) [19, 23, 24] are used as objective motion cueing metrics for comparing the motion cueing performance of the different tested classical and optimization-based motion cueing algorithms.

The structure of the paper is as follows. First, Section II introduces the stall scenario that is evaluated in this paper. Section III presents the baseline CW and the different MPC-based motion cueing algorithms and their parametrization. The offline analysis results are presented in Section IV and the work's implications, main conclusions, limitations, and next steps are discussed in Section V. The paper's main conclusions are summarized in Section VI.

II. Upset Flight Test Data

The analysis presented in this paper focuses on one specific upset maneuver of a Cessna Citation II aircraft, which will also be tested in an upcoming pilot-in-the-loop simulator experiment. The maneuver is based on real flight test data obtained with TU Delft's PH-LAB laboratory aircraft. This maneuver is a symmetric stall maneuver, as previously considered in [25, 26], which only includes motion in the three symmetrical degrees-of-freedom (DOF); i.e., pitch, vertical (heave) and horizontal-translational (surge). A selection of aircraft motion states across the stall scenario is depicted in Fig. 2. For this study, the stall maneuver's specific force time traces are filtered using a third-order low-pass filter, applied through zero-phase digital filtering, see $f_{x,filt}$ and $f_{z,filt}$ in Fig. 2. The reasoning for implementing this filtering is that the original high-frequency buffet components are structurally taxing on the motion system, necessitating 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 assessing the potential benefits of MPC-based MCAs in this study. The filtered load factor in this 15-second scenario is seen to range between 0.4 g and 1.2 g and a maximum angle of attack of 16 deg is reached. Not shown in Fig. 2, the stall maneuver is extended for 11 s at its beginning to position the simulator smoothly into the maneuver begin state. Similarly, after the maneuver, 11 s are added to transition the simulator into its neutral state. In this paper, this scenario is used to compare the pitch, surge, and heave motion obtained with the different tested MCAs.

III. Motion Cueing Algorithms

A. Classical Motion Cueing Algorithm

As a reference for the motion cueing quality obtained with optimization-based algorithms with different prediction strategies, their results are compared to motion signals obtained with representative Classical Washout (CW) algorithms previously used for similar maneuvers. In this paper, two different CW settings are considered:

- CW_I: the CW parameters originally used by Imbrechts et al. [26]
- CW_G: the CW parameters originally used by Grant and Schroeder [27]

The CW parameter settings corresponding to CW_I and CW_G are listed in Table 1 and 2, respectively. For the parameters used by Grant and Schroeder [27], the pitch and roll fidelity are close to high-fidelity and the translational motion is low-fidelity according to the Sinacori-Schroeder motion fidelity criterion [28]. Compared to the parameters used by Grant and Schroeder [27], the parameters used by Imbrechts et al. [26] (CW_I) result in higher fidelity for translational motion and lower fidelity for the angular pitching motion.

Table 1 CW filter settings used by Imbrechts et al. (CW_I) [26]

	x	y	z	p	q	r
K_{cw}	0.5	-	0.5	-	0.5	-
ω_{hp}	1.2	-	2.0	-	1.0	-
ζ_{hp}	0.7	-	0.7	-	0.5	-
ω_b	0.0	-	0.3	-	0.0	-
ω_{lp}	2.4	-				
ζ_{lp}	0.7	-				

Table 2 CW filter settings used by Grant and Schroeder (CW_G) [27].

	x	y	z	p	q	r
K_{cw}	0.4	0.3	0.5	0.6	0.6	0.3
ω_{hp}	2.0	2.0	4.0	0.0	0.0	0.7
ζ_{hp}	0.7	0.7	1.0	0.0	0.0	0.7
ω_b	0.0	0.0	0.5	0.6	0.5	0.0
ω_{lp}	2.0	2.5				
ζ_{lp}	0.7	0.7				

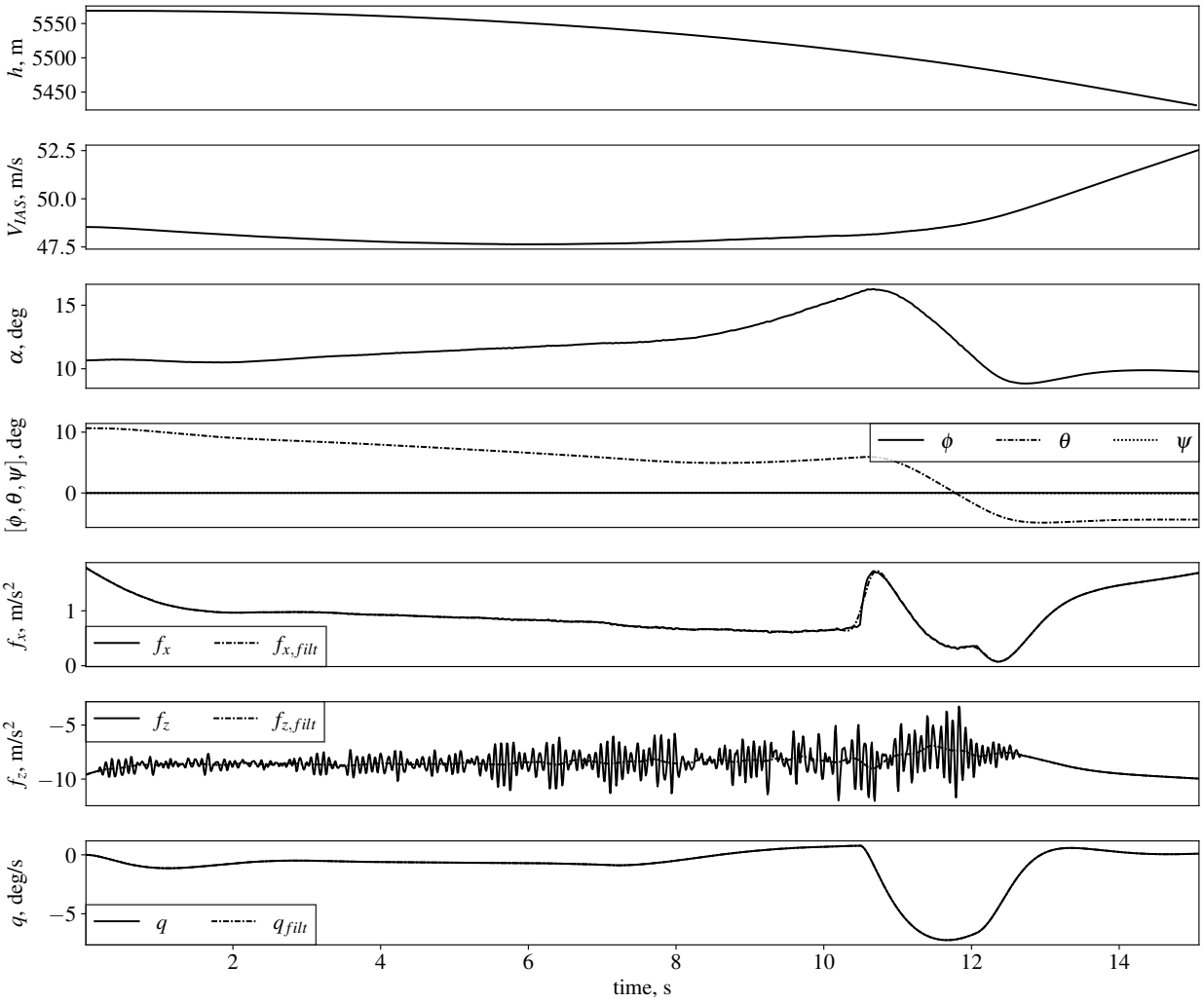

Fig. 2 Altitude, true airspeed, Angle of Attack, attitude, specific force in x and z direction, and the pitch rate for the Citation stall scenario.

B. Optimization-based Motion Cueing Algorithm

1. Hexapod motion system definition

In this paper, we investigate MPC-based MCAs implemented for a hexapod motion system as depicted in Fig. 3. The naming of a hexapod simulator or motion base comes from the six parallel 'legs', or actuators, that connect the system's fixed base platform and moving upper platform. The fixed locations of the platform's lower joint locations are typically defined with respect to the Lower Gimbal Point (LGP) and the $F_{I,LGP}$ reference frame, see Eq. (1):

$$\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{b}_4, \boldsymbol{b}_5, \boldsymbol{b}_6]^\top \in \mathbb{R}^{18} \tag{1}$$

The simulator's upper moving platform is also connected to all six actuators with universal joints. The location of these joints with respect to the Upper Gimbal Point (UGP) in the moving $F_{B,UGP}$ reference frame is again fixed and is defined by the vector in Eq. (2). 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 Eq. (3).

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4, \boldsymbol{p}_5, \boldsymbol{p}_6]^\top \in \mathbb{R}^{18} \tag{2}$$

$$\boldsymbol{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^\top \in \mathbb{R}^6 \tag{3}$$

To determine the length of the actuators, a method called inverse kinematics is used [29]. This method derives the actuator lengths from the position of the UGP with respect to the LGP (here defined as $\boldsymbol{r}$), the orientation of the upper platform with respect to the inertial reference frame (rotation matrix $\boldsymbol{R}_{F_{B,UGP},F_{I,LGP}}$), and the base gimbal and platform gimbal positions ($\boldsymbol{B}$ and $\boldsymbol{P}$, respectively). This relation is presented in Eq. (4), whole the individual actuator lengths are given by Eq. (5). A potential benefit of MPC-based cueing is that workspace constraints (i.e., on actuator positions $\boldsymbol{q}$, velocities and accelerations) can be explicitly accounted for in calculating the optimal cueing solution. This implies that an MPC-based MCA requires an explicit model of the motion system's kinematics and constraints.

$$\boldsymbol{q} = \|\boldsymbol{r} - \boldsymbol{B} + (\boldsymbol{R}_{F_{B,UGP},F_{I,LGP}}\boldsymbol{P}^\top)^\top\| \tag{4}$$

$$q_i = \|\boldsymbol{r} - \boldsymbol{b_i} + \boldsymbol{R}_{F_{G,UGP},F_{I,LGP}}p_i\| \tag{5}$$

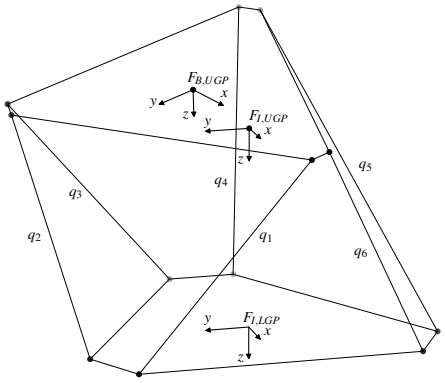

Fig. 3 Schematic hexapod system.

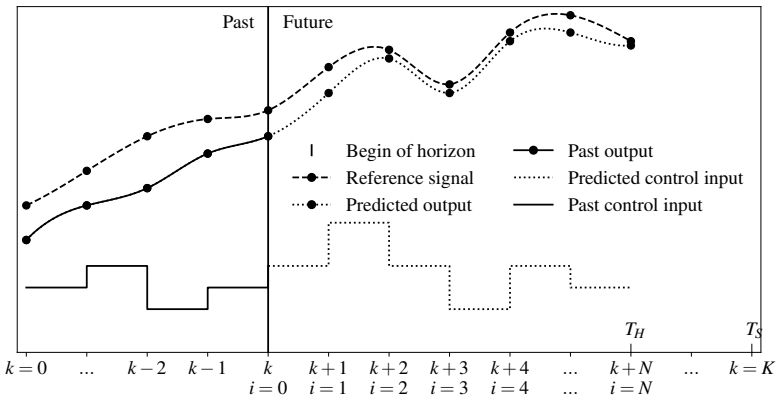

Fig. 4 Schematic representation of the offline MPC algorithm.

2. MPC algorithm

The basic principle of MPC-based MCAs is that every cycle k of the algorithm's sampling clock, an optimal control problem is solved to calculate an optimal motion system command to be applied in the current time interval $[t_k, t_{k+1}]$. This optimal control problem determines the this control input by minimizing an objective function that penalizes the difference between the predicted future reference signal and the predicted future output of the system, see Fig. 4. Note that the future reference signal is, especially for real-time MPC-based MCAs, often a prediction and may never be truly equal to the real future reference signal, as it is dependent on environmental factors and human pilot inputs. The length in time of the future window accounted for in the optimal control problem is called the prediction horizon T_H. After solving the optimal control problem of the current time interval k, the resulting system state of the next time interval $k+1$ becomes the initial condition for the optimal control problem solved to obtained the motion command for the next time interval $[t_{k+1}, t_{k+2}]$.

The software implementation of the MPC-based MCA used in this paper was developed using the open-source Offline Motion Simulation Framework (OMSF) [30, 31]. 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 parametrization method [32] to transform a continuous-time optimal control problem (OCP) into discrete time. The OMSF depends on the CasADi symbolic numeric optimization software framework [33] in combination with the Ipopt non-linear optimization software [34]. Moreover, the Ipopt-compatible Harwell Subroutine Library high-speed solvers [35] were used in this study.

3. Cost function

The cost function that is used to optimize the optimal control problem can be divided into two parts and is similar to the cost function used by Katliar et al. [30]. The first part is the inertial signal incongruence cost for the specific forces f, rotational rates ω, and rotational accelerations α. It penalizes the squared difference between the reference inertial signal $\hat{y}$ and the output inertial signal produced by the simulator y. This incongruence cost can be defined by Eq. (6). The inertial signal weight matrix is defined by W_y, as found in Eq. (7).

$$J_{inc}(y, \hat{y}) = ||y - \hat{y}||^2_{W_y} \tag{6}$$

$$W_y = \mathbb{1}_{9\times 9} \cdot [\mathbf{w}_f, \mathbf{w}_\omega, \mathbf{w}_\alpha]^\top \tag{7}$$

The second part of the cost function is the simulator cost and penalizes 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 Eq. (8). The state and control input weight matrices are defined by W_x and W_u, as found in Eq. (9) and Eq. (10), respectively.

$$J_{sim}(u, x) = ||x||^2_{W_x} + ||u||^2_{W_u} \tag{8}$$

$$W_x = \mathbb{1}_{12\times 12} \cdot \mathrm{w}_x \tag{9}$$

$$W_u = \mathbb{1}_{6\times 6} \cdot \mathrm{w}_u \tag{10}$$

The total optimized cost is defined by the sum of the incongruence cost and the simulator cost, resulting in the expression seen in Eq. (11).

$$J_{total}(y, \hat{y}, u, x) = J_{inc}(y, \hat{y}) + J_{sim}(u, x) \tag{11}$$

4. Tested prediction types

In this paper, three offline MPC algorithm variants are tested, varying in the extent to which they can use future motion cue signals in their optimization. In this paper, we use the symbol T_H to refer to the MPC's prediction horizon length, while T_S indicates the total scenario length, see also Fig. 4. The three tested prediction types are referred to as the "Oracle", the "Perfect", and the "Constant" algorithms:

- The "Oracle" (ORC) has full knowledge of the entire reference signal from the flight test data, and employs 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 a given motion scenario and the chosen set of parameters and weights. Note that this algorithm is not feasible in an interactive, real-time situation, since perfect knowledge of the reference signal for the entire motion scenario is generally not available.
- The "Perfect" (PFT) assumes perfect knowledge of the reference signal for the entire prediction horizon, T_H, where T_H is shorter than the scenario duration. This will result in the best obtainable control trajectory obtained through MPC for a certain horizon length. This prediction strategy is not feasible in an interactive simulation, 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 that in the current time interval. Contrary to the first two variants, this prediction strategy is feasible in an interactive setting [23], since it will only require knowledge of the reference signal for the current time interval.

5. MPC parameter settings

MPC problems are well-known to have a large number of parameter settings that require tuning and can greatly influence the algorithm's optimal solution. For the evaluation of the three MPC-based MCAs in this paper, these parameter settings were based on prior work (i.e., literature) and knowledge of the SRS's motion system characteristics where possible. A formal optimization of the MPC parameters is considered important future work. The total list of parameters used in this paper is listed in Table 3.

Table 3 MPC-based MCA parameter settings.

Setting (and applicable algorithm)	Parameter	Value
Prediction horizon settings	Δt_i	0.05 s
("Perfect", "Constant")	N	200
Weight settings ("Oracle", "Perfect", "Constant")	W_y	$\mathbb{1}_{9\times9} \cdot [1, 1, 1, 10, 10, 10, 1, 1, 1]^\top$
	W_x	$\mathbb{1}_{12\times12} \cdot 0.01$
	W_u	$\mathbb{1}_{6\times6} \cdot 0.01$
Simulator state constraints	$\underline{x}$	$[\underline{x}_r, \underline{x}_v]^\top$
("Oracle", "Perfect", "Constant")	$\overline{x}$	$[\overline{x}_r, \overline{x}_v]^\top$
Initial state constraints	$\underline{x}_{r,0}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top$
("Oracle", "Perfect", "Constant")	$\overline{x}_{r,0}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top$
	$\underline{x}_{v,0}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top s^{-1}$
	$\overline{x}_{v,0}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top s^{-1}$
Final state constraints	$\underline{x}_{r,T}$	$[-0.981\text{ m}, -1.031\text{ m}, -0.636\text{ m}, \frac{-25.9\pi}{180}\text{ rad}, \frac{-23.7\pi}{180}\text{ rad}, \frac{-41.6\pi}{180}\text{ rad}]^\top$
("Oracle", "Perfect", "Constant")	$\overline{x}_{r,T}$	$[1.259\text{ m}, 1.031\text{ m}, 0.678\text{ m}, \frac{25.9\pi}{180}\text{ rad}, \frac{24.3\pi}{180}\text{ rad}, \frac{41.6\pi}{180}\text{ rad}]^\top$
	$\underline{x}_{v,T}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top s^{-1}$
	$\overline{x}_{v,T}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top s^{-1}$
State, control input and actuator constraints ("Oracle", "Perfect", "Constant")	$\underline{x}_r$	$[-0.981\text{ m}, -1.031\text{ m}, -0.636\text{ m}, \frac{-25.9\pi}{180}\text{ rad}, \frac{-23.7\pi}{180}\text{ rad}, \frac{-41.6\pi}{180}\text{ rad}]^\top$
	$\overline{x}_r$	$[1.259\text{ m}, 1.031\text{ m}, 0.678\text{ m}, \frac{25.9\pi}{180}\text{ rad}, \frac{24.3\pi}{180}\text{ rad}, \frac{41.6\pi}{180}\text{ rad}]^\top$
	$\underline{x}_v$	$[-1000\text{ m}, -1000\text{ m}, -1000\text{ m}, -1000\text{ rad}, -1000\text{ rad}, -1000\text{ rad}]^\top s^{-1}$
	$\overline{x}_v$	$[1000\text{ m}, 1000\text{ m}, 1000\text{ m}, 1000\text{ rad}, 1000\text{ rad}, 1000\text{ rad}]^\top s^{-1}$
	$\underline{u}$	$[-10\text{ m}, -10\text{ m}, -10\text{ m}, -2\text{ rad}, -2\text{ rad}, -2\text{ rad}]^\top s^{-2}$
	$\overline{u}$	$[10\text{ m}, 10\text{ m}, 10\text{ m}, 2\text{ rad}, 2\text{ rad}, 2\text{ rad}]^\top s^{-2}$
	$\underline{q}$	$[2.131, 2.131, 2.131, 2.131, 2.131, 2.131]^\top m$
	$\overline{q}$	$[3.281, 3.281, 3.281, 3.281, 3.281, 3.281]^\top m$
	$\underline{\dot{q}}$	$[-0.75, -0.75, -0.75, -0.75, -0.75, -0.75]^\top m s^{-1}$
	$\overline{\dot{q}}$	$[0.75, 0.75, 0.75, 0.75, 0.75, 0.75]^\top m s^{-1}$
Interval final state constraints	$\underline{\mathbf{x}}_{r,k,N}$	$[-0.981\text{ m}, -1.031\text{ m}, -0.636\text{ m}, \frac{-25.9\pi}{180}\text{ rad}, \frac{-23.7\pi}{180}\text{ rad}, \frac{-41.6\pi}{180}\text{ rad}]^\top$
("Perfect", "Constant")	$\overline{\mathbf{x}}_{r,k,N}$	$[1.259\text{ m}, 1.031\text{ m}, 0.678\text{ m}, \frac{25.9\pi}{180}\text{ rad}, \frac{24.3\pi}{180}\text{ rad}, \frac{41.6\pi}{180}\text{ rad}]^\top$
	$\underline{\mathbf{x}}_{v,k,N}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top s^{-1}$
	$\overline{\mathbf{x}}_{v,k,N}$	$[0\text{ m}, 0\text{ m}, 0\text{ m}, 0\text{ rad}, 0\text{ rad}, 0\text{ rad}]^\top s^{-1}$

The MPC prediction horizon was implemented with a time step of 0.05 and a maximum number of 200 samples (i.e., 10 s) for the "Perfect" and "Constant" algorithms. The weights used for the different MPC cost function terms were based on literature [19, 20, 31]. The used state, control input and actuator constraints are directly derived from the SRS's motion system characteristics. Constraints that are not relevant to the current implementation for the SRS motion system, as the solution will not be limited by its design, are set to non-limiting values (e.g., 1000 rad).

C. Evaluation Metrics

In this paper, we focus on evaluating the motion cueing quality attained by the different tested algorithms from an objective perspective, i.e., by quantitatively comparing the simulator output inertial signal as a result of the simulator movement with respect to the reference aircraft signals. For this, we use two objective metrics – the root mean square error (RMSE) and Pearson's correlation coefficient (PCC) – that were are also used for comparing motion signals in previous work [19, 23, 24, 36]. The RMSE is a measure to indicate the magnitude difference between two signals and is depicted in Eq. (12). Due to the squared difference in this relation, large differences between two signals have relatively more effect on the RMSE than small differences.

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^{N} (\hat{y}_k - y_k)^2}{N}} \tag{12}$$

To identify the resemblance of two signals, meaning the shape similarity of signals, the PCC is used, as seen in Eq. (13). 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 penalized. 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 [19, 24].

$$\text{PCC} = \frac{1}{N-1}\sum_{k=1}^{N}\left(\frac{\hat{y}_k - \mu_{\hat{y}}}{\sigma_{\hat{y}}}\right)\left(\frac{y_k - \mu_y}{\sigma_y}\right) \tag{13}$$

IV. Results

A. Motion Reproduction

The influence of the prediction strategy applied for the optimization-based algorithm is analyzed by comparing the "Oracle", the "Perfect" and the "Constant" baseline conditions with the two CW settings defined in Section III.A. In Fig. 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 result in very similar objective motion cueing quality. Furthermore, these conditions outperform all other conditions except for the pitch rate PCC, where the two CW conditions are marginally better. For the "Oracle" and "Perfect" MPC MCAs, the RMSE values for the specific forces and rotational velocities are, however, found to be 29.8% and 18.7% lower than for the CW algorithms. Looking at Fig. 6, which shows the f_x, f_z and q time traces for all MCAs compared to the reference aircraft data, these results can be visibly validated. For f_x the "Oracle" and "Perfect" conditions are closer to the reference signal due to increased 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 $t = 20$ s and during the final 20 seconds of the scenario. This explain the higher (better) 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 for example seen at $t = 22$ s in Fig. 6.

For the "Constant" condition, Fig. 5 shows that 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 tested MCAs. This can be validated by inspecting Fig. 6, which shows that 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 amplitude of the "Constant" f_z signal are very bad since there is almost no motion, resulting in a low PCC value for the "Constant" specific forces. Finally, for the angular rates, more delay (at $t = 13$ s) and false cues (at $t = 22$ s) can be identified, resulting in the low RMSE and PCC values for the angular rates.

B. Motion Workspace Utilization

A crucial benefit of MPC-based MCAs over the CW is that MPC enables optimized maximum used of the available simulator workspace at any given moment during a simulated scenario. For a CW, the scaling and filtering applied to the complete scenario's data is always tuned to ensure the singular worst-case instance of vehicle motion that occurs somewhere in the scenario does not cause workspace excursions. To directly compare the simulator workspace utilization between the different CW and MPC MCAs analyzed in this paper, Fig. 7 shows the envelopes of actuator displacement as a function of scenario simulation time. The range of values captured by the presented envelopes show the maximum and minimum extensions across all six hexapod actuators. Overall, Fig. 7 shows that for all three MPC MCAs the maximum actuator displacement constraints – indicated with the horizontal black dashed lines – are respected at all times, but that the optimization truly makes use of the full available workspace; at multiple moments during the simulated stall scenario are the maximum allowed actuator displacements (almost) reached. This is a stark contrast with the workspace utilization by the CW_I and CW_G MCAs, which show much reduced actuator displacements, except around $t = 26$ s where the stall recovery is performed.

Consistent with Fig. 6, Fig. 7 further shows highly similar workspace utilization and simulator movement for the "Oracle" and "Perfect" MPC algorithms, which only differ at the recovery initiation. The solution obtained from the "Constant" MPC algorithm is seen to result in very different utilization of the workspace; this shows the direct sensitivity of MPC-based MCA results with respect to the quality of their internal prediction over T_H.

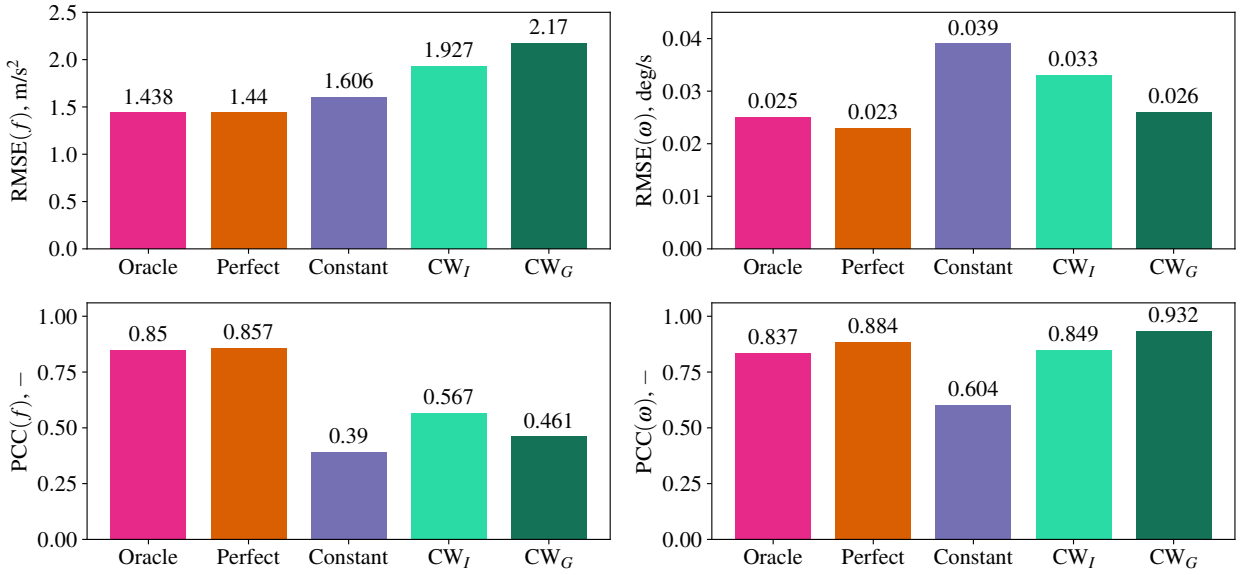

Fig. 5 RMSE and PCC for the inertial signal specific forces and angular rates as a result of different prediction strategies. For the RMSE, the sum of the three RMSE values of the specific forces and of the angular rates are presented. For the PCC, the average of the three PCC values is presented.

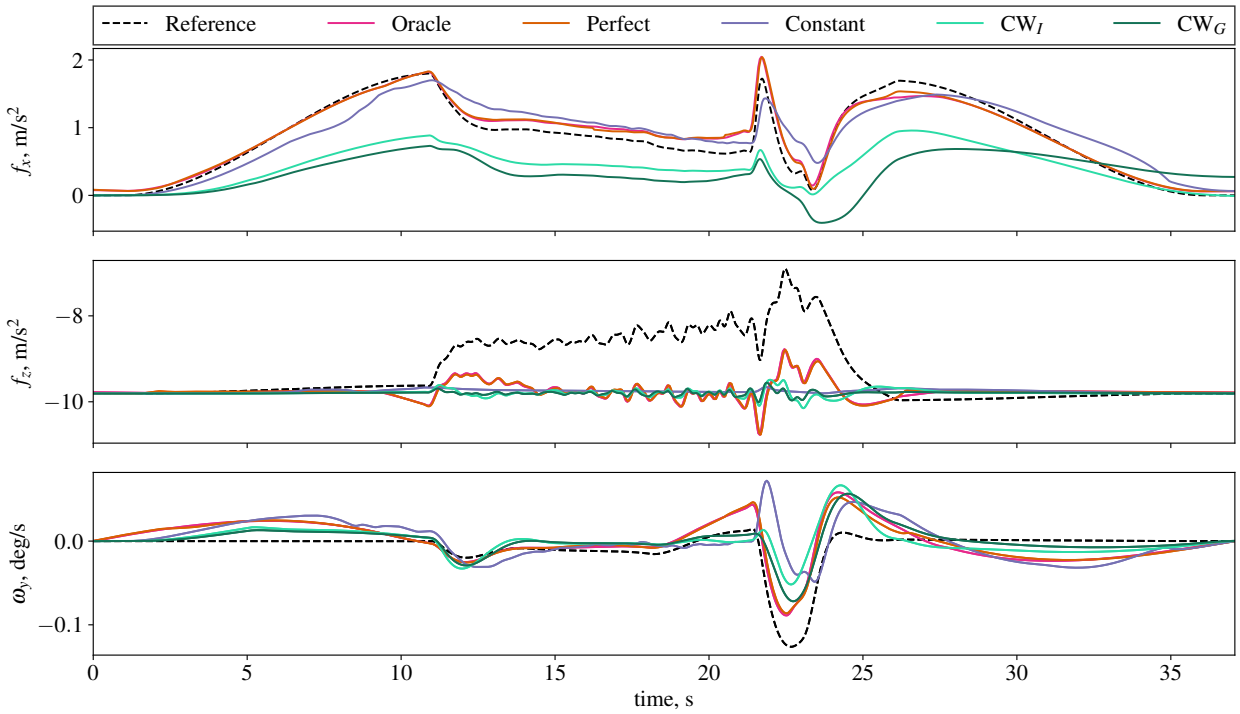

Fig. 6 Reference inertial signal and the output inertial signal for the "Oracle", the "Perfect", the "Constant" and the two CW conditions.

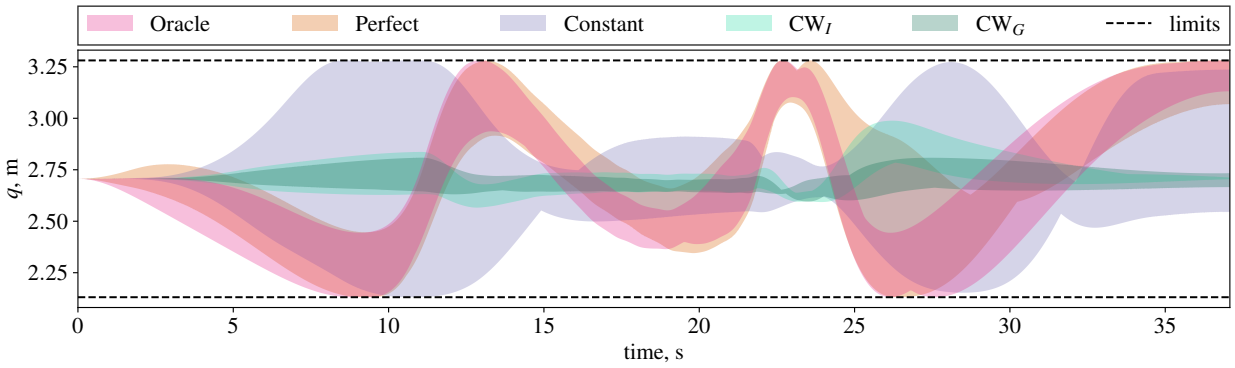

Fig. 7 Used actuator space for the "Oracle", the "Perfect", the "Constant" and the two CW conditions.

C. MPC Optimization Analysis

To provide more insight into how MPC-based MCAs obtain an optimal motion cueing outcome at every simulation time step using their cost function as defined by Eq. (11), Fig. 8 shows an explicit break down of the MPC's cost function for the "Oracle" MCA's final result. The top figure in Fig. 8 shows the different contributions to the 'incongruence cost' defined in Eq. (6). The second and third figures show the two components of the 'simulator cost' defined in Eq. (8), i.e., corresponding to state deviations (x) and the provided control inputs (u), respectively. Finally, the bottom figure in Fig. 8 shows the final result of Eq. (11), i.e., the total MPC cost attained throughout the scenario.

Overall, Fig. 8 shows that the total MPC cost increases throughout the simulated stall scenario, reaching its peak at the moment when the stall recovery is initiated. This is expected, as especially the required vertical (heave) motion amplitude increases during this scenario and can only be replicated on the SRS's motion system with limited success. Due to this, the total overall internal quality of the solution calculated by the MPC (J_{total}) increases. Furthermore, Fig. 8 also shows that the main contributor to the total MPC cost for this scenario is indeed the incongruence cost for replication of f_z (top figure, green time trace). The other cost terms are seen to all be orders of magnitude smaller throughout the scenario. Naturally, a different tuning of the MPC weights would enable placing more emphasis on other cost function terms, which would increase the magnitude of their contribution compared to the results shown in Fig. 8.

D. MPC Prediction Horizon Sensitivity

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 Fig. 9. 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 to 5 seconds and finally, close to the "Oracle" condition for a horizon of 6 to 10 seconds.

V. Discussion

This paper focused on the offline comparison of two different Classical Washout (CW) and three different Model Predictive Control (MPC) motion cueing algorithms (MCAs) for a symmetric stall scenario. Objective metrics for the degree of motion reproduction and the utilization of the available workspace – as also considered in MPC MCA cost functions – were considered for this comparison. The three different MPC implementations – "Oracle", "Perfect", and "Constant" – varied in the 'quality' of their knowledge of the future vehicle motion and their MPC prediction horizon implementation. Furthermore, an explicit analysis was included to verify the effect of choosing a certain MPC horizon length for the "Perfect" and "Constant" algorithms.

The analysis in this paper showed very similar results for the "Oracle" and "Perfect" MPC implementations, especially with a horizon length larger than 5 s for "Perfect", see Figure 5 and Figure 6. Both algorithms use perfect knowledge of the reference vehicle motion, but the "Perfect" MPC does so over a limited horizon. The similarity in result for both cases indicates that for the current scenario, the maximum tested horizon length of 10 s for "Perfect" is clearly sufficient for enabling the MPC to fully utilize the available motion space for onset cues and possible pre-positioning. The reason

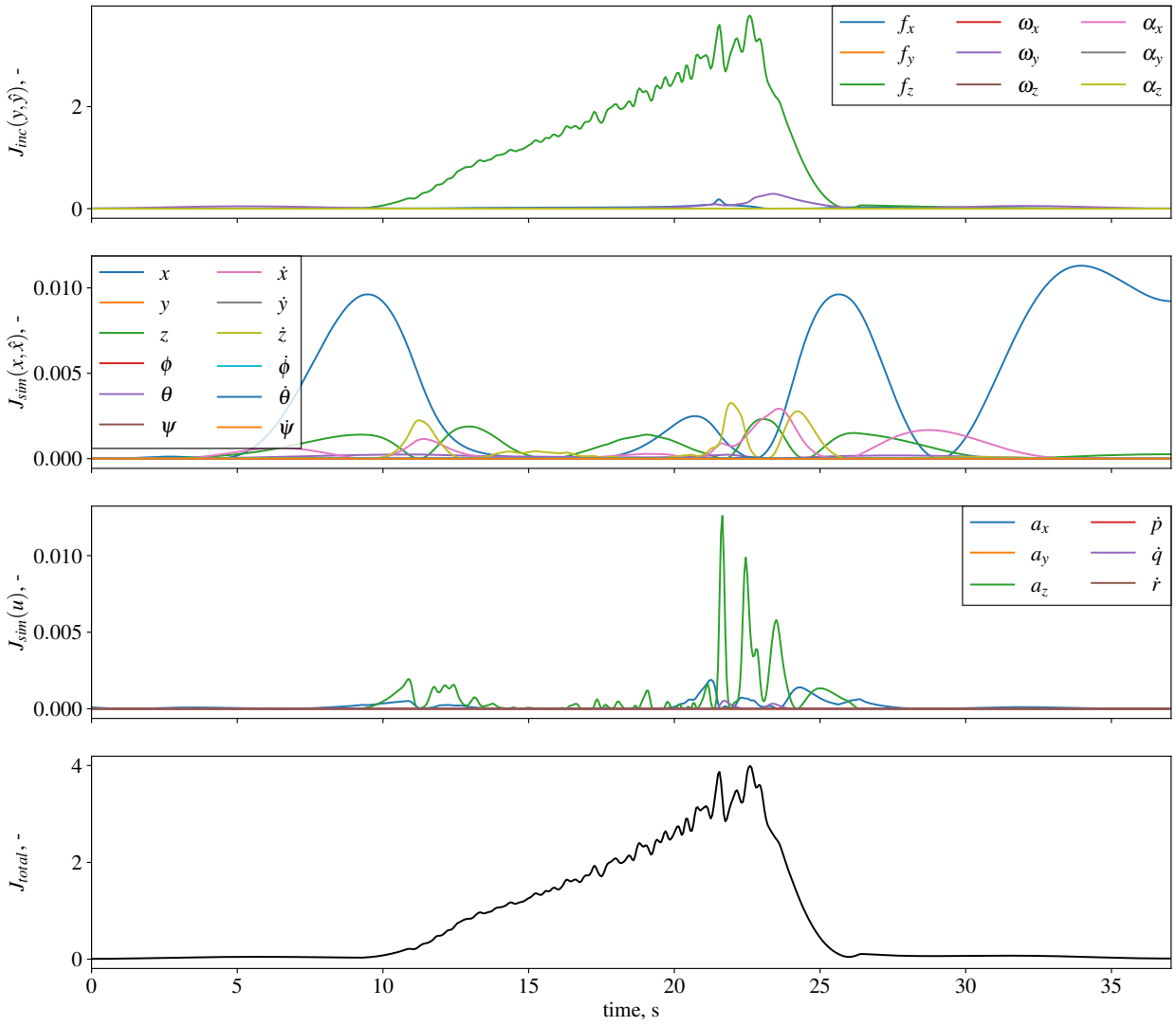

Fig. 8 Inertial signal incongruence, simulator state, simulator input, and total cost for the "Oracle" condition.

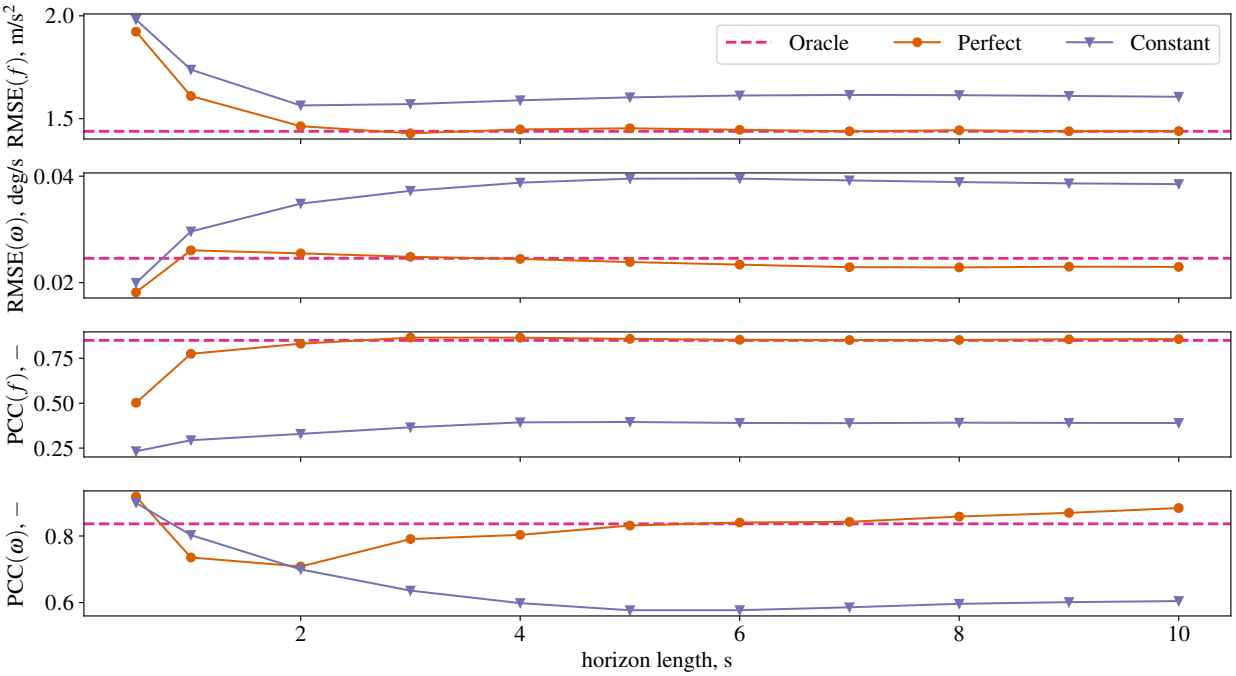

Fig. 9 RMSE and PCC for different prediction horizon lengths, for the "Perfect" and "Constant" conditions.

why increasing the prediction horizon length over 5 s has little influence (see Figure 9) is that for the chosen scenario the reference inertial signals do not contain motion cues that need longer simulator pre-positioning. The size of the motion system, meaning the available actuator range, and the actuator velocity limit further contribute to 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.

For reference, the "Constant" MPC algorithm, which assumes the same aircraft rotational accelerations and specific forces across the complete prediction horizon, was also included in the comparison made in this paper. This assumption is seen to directly cause a lack of any pre-positioning action (as observed for "Oracle" and "Perfect") for the "Constant" MPC. Furthermore, it is found that this "Constant" MPC reproduces almost no motion in heave (see f_z in Figure 6). This is explained by the (required) implementation of a terminal constraint, which forces the simulator state velocities and angular rates to be zero at the end of the prediction horizon. The "Constant" MPC mostly optimizes for a match of f_x, also by making use of considerable pitch tilt rates, see ω_y at $t = 22$ s in Figure 6. Alleviating the terminal constraint likely would improve, but not fully remove, these drawbacks for the "Constant" MPC algorithm.

For the considered symmetric stall scenario, all three MPC algorithms show larger pitch rates, and hence more tilt coordination, than the two reference CW MCAs. The reason why more pitch tilt coordination is present for the "Oracle", "Perfect" and "Constant" algorithms can be explained by the considered CW forward specific force gain, K_{f_x}. This gain is set to 0.5 and 0.4 for the CW_I and CW_G conditions, respectively, meaning that *at most* the CW CWAs will try to replicate the f_x signal for more than 60% of the original vehicle's inertial signal magnitude. This uniform scaling of specific forces directly at the input is not performed for all MPC MCAs.

This paper presented a comparison of the motion cueing quality for three different MPC algorithms that all used the same parameter (MPC weight) settings. A more precise sensitivity analysis of all weight parameters would be expected to result in a different MPC cost breakdown as visualized in Fig. 8 and a further (minor) improvement of the presented MPC results. Due to the large number of (correlated) weight parameters that need to be set, developing objective MPC parameter tuning methodologies that can for example be applied to optimize MPC motion cueing for different scenarios, is a crucial next step for MPC motion cueing algorithm research.

Overall, for the considered symmetric stall scenario, a consistent improvement in motion cueing accuracy of 29.8% and 18.7% for RMSE(f) and RMSE(ω), respectively, was observed in our analysis. To also achieve this improved cueing quality with an online MPC, an accurate prediction of the future vehicle motion along a prediction horizon between 5-10 s would be needed. Furthermore, the influence of making the prediction horizon non-uniform in terms

of its time resolution should be investigated. This may result in a less computationally heavy MPC problem, that is more readily implemented in practice. Furthermore, the extent to which this clear difference in objective motion cueing quality between MPC and CW MCAs also translates to a noticeable difference for pilots needs to be verified with a pilot-in-the-loop evaluation. This is the crucial next step we aim to pursue in support of the further development of optimization-based MCAs for UPRT simulation.

VI. Conclusions

This paper investigates the potential of optimization-based motion cueing algorithms for simulating aircraft upset and stall maneuvers in moving-based flight simulators. In preparation for an upcoming pilot-in-the-loop evaluation in TU Delft's SIMONA Research Simulator of this novel approach to simulator motion cueing, this paper presents an offline simulation analysis to compare motion cueing performance between optimization-based algorithms with different prediction implementations and two classical washout algorithm settings from previous experiments. For this analysis, objective metrics – i.e., the root mean square error (RMSE) and Pearson's correlation coefficient (PCC) – are used to compare the match between simulator motion signals and vehicle data for a simulated stall maneuver. The results show that especially the "Oracle" or the "Perfect" optimization-based algorithms – which both assume perfect knowledge of the future vehicle motion along the prediction horizon – show improved tracking of both the vehicle specific forces and rotational rates (29.8% and 18.7% lower RMSE, respectively) compared to the classical filter-based algorithms. Furthermore, for algorithms with a finite prediction horizon (i.e., "Perfect" and "Constant"), the horizon length should be chosen between 6 and 10 seconds, as this range is computationally feasible and does not limit motion cueing fidelity. Overall, the presented 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] Federal Aviation Administration, "AC 120-111 - Upset Prevention and Recovery Training - with Change," , 8 2020.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

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

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.