A Review on Optimization-Based Motion Cueing Algorithms for Driving Simulation

Robert Jacumet, Maurice Kolff, Joost Venrooij, Markus Schwienbacher, Sebastian Wagner, Dirk Wollherr, *Senior Member, IEEE*, Marion Leibold, Daan M. Pool, *Member, IEEE*, and Max Mulder, *Senior Member, IEEE*

***Abstract*—Driving simulators are essential tools to guide automotive research and development. Their motion system requires a motion cueing algorithm (MCA) to keep the simulator motion platform within its physical boundaries, while simultaneously aiming to recreate the sensation of real vehicle motion. While traditional, filter-based approaches are still predominant, optimization-based MCAs have been at the center of MCA research for over a decade due to their ability to systematically improve motion cueing quality through explicit cost function design and constraints handling. However, despite their demonstrated advantages, these optimization-based methods have not yet achieved widespread adoption in driving simulation. This paper therefore provides a comprehensive review of optimization-based MCAs for driving simulation, categorizing and comparing algorithms, describing their key developments and core characteristics. The current limited real-time capability, lack of accurate evaluation methods, challenges in cost function design and its tuning, and the current lack of accurate future reference predictions are identified as key barriers to the practical deployment and widespread use of optimization-based MCAs. These theoretical and practical challenges are further reviewed, providing guidelines to advance the theory and application of optimization-based MCAs. Central in these advancements are a better understanding of which motions constitute a realistic motion experience, a framework allowing to compare the achieved motion fidelity of MCAs across papers, the design of the cost function focusing on human motion perception, and techniques for easing up the tuning process to swiftly reach high quality tunings for different simulators, scenarios, and use cases. We identify the need for improving the real-time capability, and providing high quality motion reference predictions using learning-based approaches on diverse datasets, along with techniques to handle existing uncertainties. Following these guidelines, new foundations for optimization-based algorithms in driving simulation can be achieved, which will significantly impact the research and development of automotive systems.**

***Index Terms*—Driving simulation, motion systems, motion cueing algorithm, model predictive control, human-in-the-loop.**

Received 4 March 2025; revised 24 September 2025; accepted 27 January 2026. Date of publication 23 March 2026; date of current version 1 July 2026. The Associate Editor for this article was I. Papamichail. *(Corresponding author: Robert Jacumet.)*

Robert Jacumet is with the Chair of Automatic Control Engineering, Technical University of Munich, 80333 Munich, Germany, and also with the Department of Research and Development, BMW Group, 80807 Munich, Germany (e-mail: r.jacumet@tum.de).

Maurice Kolff, Daan M. Pool, and Max Mulder are with the Section Control and Simulation, Delft University of Technology, 2629 HS Delft, The Netherlands (e-mail: m.j.c.kolff@tudelft.nl; m.pool@tudelft.nl; m.mulder@tudelft.nl).

Joost Venrooij, Markus Schwienbacher, and Sebastian Wagner are with the Department of Research and Development, BMW Group, 80807 Munich, Germany (e-mail: joost.venrooij@bmw.de; markus.schwienbacher@bmw.de; sebastian.ws.wagner@bmw.de).

Dirk Wollherr and Marion Leibold are with the Chair of Automatic Control Engineering, Technical University of Munich, 80333 Munich, Germany (e-mail: dw@tum.de; marion.leibold@tum.de).

Digital Object Identifier 10.1109/TITS.2026.3661819

I. Introduction

Driving simulators are invaluable tools in the development of novel automotive technologies. They provide a safe, repeatable, sustainable, cost-efficient, and controllable alternative to real vehicle testing. Typical research examples are human-machine interaction [1], human-in-the-loop testing of vehicle safety systems [2], and human factors research on automated driving and traffic safety [3]. When equipped with a dynamic motion system, the a prime goal of the driving simulator is typically to mimic the inertial motion of a real vehicle as closely as possible. Poor motion reproduction can lead to decreased realism [4], simulator sickness [5], and unwanted adaptation of control behavior [6], [7], which all negatively affect the validity of experimental outcomes. Due to the limited size and motion capabilities of even the largest and most advanced motion systems, the motion of the real vehicle can often not be reproduced directly, and imperfections and trade-offs are inevitable. Therefore, a conversion of the vehicle motion is required to keep the resulting simulator motion within the physical limits of the simulator. This conversion is done by the Motion Cueing Algorithm (MCA) [8].

Over the years, several different MCA architectures have been proposed, all with their own (dis-)advantages. A top-level classification of the different MCAs is given in Fig. 1. By far the most-used MCA is the Classical Washout Algorithm (CWA) [8], [9], [10]. It converts the reference motion through combinations of scaling, limiting, and filtering operations. However, the use of linear filters does not allow platform constraints to be accounted for. The simulator motion is therefore tuned to accommodate for the worst-case maneuver in a scenario, such that the resulting cueing is often rather conservative considering the potential of the motion system. Furthermore, its causal filters imply that predictions of the future vehicle states cannot be incorporated in the algorithm directly, limiting the simulator workspace utilization. Alternatives, such as the Adaptive Washout Algorithm (AWA) [13] and Optimal Washout Algorithm (OWA) [29], [30], [31], [32], provide some improvement, but are still filter-based and thus suffer from many of the same drawbacks as the CWA.

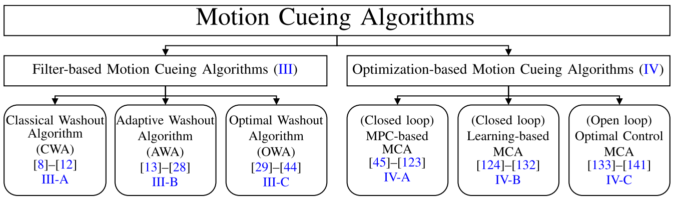


Fig. 1. Types of MCAs, along with the relevant literature included in this review, as well as the corresponding section in this work.

In recent years, enabled by enhanced real-time computing power, the focus in MCA development has moved towards optimization-based methods. An optimization-based MCA computes the simulator motions by solving an optimization problem to maximize motion fidelity, while respecting the motion constraints of the system. Through various comparative simulator studies [134], [135], [142], [143], [144], [145], [146] in which participants had to rate the motion realism of different MCAs, optimization-based algorithms, in their various forms, have been shown to be perceived as more realistic compared to filter-based algorithms. Additionally, reduced false cues, and more effective workspace utilization were achieved with optimization-based MCAs. In this category, model predictive control (MPC)-based MCAs have received the most attention, which can optimize the simulator motion including (a prediction of) the future reference motion over a prediction horizon.

Despite their great potential, MPC and other optimization-based algorithms have still not become the industry standard. Rather, it is the CWA that prevails as the dominant MCA [48]. Logically, this must result from inherent drawbacks in optimization-based algorithms that counter the benefit of their superior motion reproduction. Known issues include difficulty to prove stability, high computational effort, and algorithm and parameter tuning complexity [45], [46], [47]. To improve the algorithms in these areas, simplifications of the optimization problem are often the proposed solution, such as linearizing (part of) the kinematic relations of the motion system, decoupling Degree of Freedoms (DoFs), or reducing the size of the optimization problem by decreasing the prediction horizon. Such approaches often work, but the validity of their simplifications depends on a variety of factors, such as the use-case, the simulator, or the study purpose.

In literature, several review papers exist related to motion cueing. However, these have either focused more on describing simulator system technologies [147], [148], do not focus on optimization-based MCAs for driving simulation [149], or are more than a decade old (Fischer [150], Garrett and Best [151], Stahl et al., [8]), and, hence, do not cover the recent rapid developments of optimization-based MCAs. More recent work, such as Ghafarian et al. [152], primarily reviews driving simulator systems, along with a categorization of MCAs and their control parameter tuning. A comprehensive review of the state-of-the-art of optimization-based MCAs for driving simulation does not yet exist.

The literature review of this paper covers MCAs up to mid 2025 and is limited to driving simulation with a focus on optimization-based algorithms. Works regarding other types of simulation, mainly flight simulation, are included where they introduce foundational concepts due to the historical progression from flight to driving simulation. The literature was primarily collected from established scientific databases, including IEEE Xplore, Scopus, Web of Science, Springer Link, and Science Direct. The search was carried out by using combinations of keywords such as “motion cueing”, “driving simulation”, “motion”, “simulator”, “control”, and combinations thereof. In addition, we screened the proceedings of associated conferences (e.g. Driving Simulation Conference, IEEE SMC, IEEE ITSC) and followed a snowballing approach in which we screened the reference lists of highly cited works to ensure that recent contributions and relevant seminal works are included.

We define six MCA categories shown in Fig. 1 and assign the reviewed works to these categories. Filter-based MCAs with manually tuned and fixed parameters belong to the CWA group. AWA has parameters adapting during runtime, and filter-based MCAs with optimized fixed parameters are categorized as OWA. Optimization-based MCAs solving an infinite horizon optimal control problem with perfect knowledge of the vehicle trajectory are categorized as open-loop optimal control MCAs. In contrast, MPC-based MCAs compute the simulator controls in a closed-loop fashion, i.e., with a human driver in the loop, over a receding finite horizon. Here, linear and nonlinear MPC MCAs exist. Last, learning-based MCAs train, i.e., optimize, neural networks, either using supervised learning or reinforcement learning approaches. The different MCA types are compared regarding cueing fidelity, ease of tuning, and computational requirements.

This paper has two contributions. First, a comprehensive review of existing research in the field of optimization-based MCAs in driving simulation is provided. Second, motivated by the still dominant usage of filter-based MCAs, the paper identifies and analyzes the key challenges of optimization-based MCAs and shows possible approaches to resolve them. In addition, guidance for future research in the field is provided.

The paper is structured as follows. First, Sec. II describes key concepts in simulator kinematic control and human motion perception, which form the basis of MCA design. Section III describes and characterizes the existing filter-based approaches with an overview of their (dis)advantages. Section IV then describes optimization-based MCAs, their operating principles, and the available design choices. An in-depth review is performed on the relevant state-of-the-art problem formulations, key properties, and current solutions found in the literature. Section V then discusses the current key challenges for these optimization-based MCAs and provides guidelines for future work. Finally, Sec. VI concludes the paper.

II. Motion Cueing Preliminaries

This section describes how the motion system kinematics determine the motion to be generated by the MCA, and how this generated motion is then perceived by the driver. The proper MCA design thus requires knowledge on both.

A. Driving Simulator Motion System Kinematics

The typical goal of an MCA is to generate simulator motion that is perceived as similar as possible to the motion that

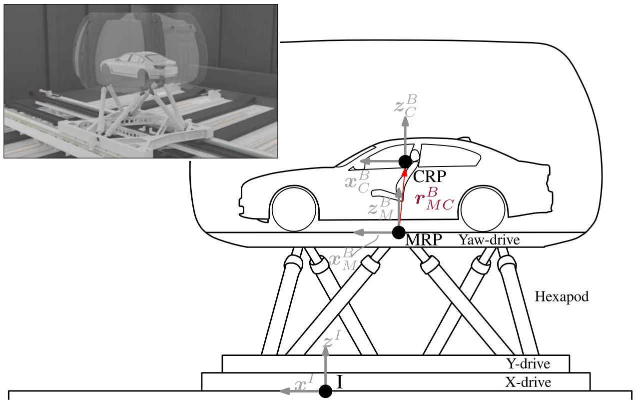


Fig. 2. Side view of the Sapphire Space simulator, showing the Cueing Reference Point (CRP) and Motion Reference Point (MRP). The motion of the simulator is defined in the simulator inertial system (I). Taken from [160].

would be perceived in a real vehicle. To this end, the MCA takes the reference motion of the simulated vehicle as an input and generates the control signals for the simulator motion (sub)systems as an output.

1) Driving Simulator Motion Systems: Many driving simulator motion systems consist of a 6-DoF hexapod [147], [152], [153], also known as *Stewart* [154] or *Stewart-Gough* [155] platform. Here, six parallel actuators move a single motion platform. Each unique actuator position defines a unique platform position. To improve the potential motion reproduction, the limited workspace of the 6-DoF hexapod can be extended using additional redundant motion subsystems. This is commonly done using linear XY-drives (e.g., FKFS's simulator [156]), yaw-drives (e.g., BMW's Sirius Vector [157]), or a combination of both (e.g., Toyota's Driving Simulator [158], Renault's ROADS [140], and BMW's Sapphire Space [157]). A side view of the latter simulator is shown in Fig. 2. More extended overviews on existing motion systems and other configurations can be found in [147], [152], and [159].

Challenging is that for each type of kinematic configuration, different system models emerge, also affecting the motion cueing formulation. For example, a *serial* robot such as a robot arm [112], in which all actuators are placed as a serial chain, has significantly different kinematic relations. Such systems oftentimes show a higher degree of nonlinearity, which further complicates system modeling and constrains real-time capability of MCAs. However, in general, by replacing the specific simulator system model and making adequate changes, MCAs, especially optimization-based algorithms, can be adapted to work on vastly different simulator configurations. Therefore, here, we focus on and describe the kinematic motion control of a typical 9-DoF system such as BMW's Sapphire Space, consisting of a XY-drive, hexapod, and yaw-drive. By setting the XY- and yaw-drive signals to zero, these descriptions furthermore also represent the highly common hexapod system.

In all kinematic configurations, the MCA controls both the rotational and translational motion of the simulator. Common practice in driving simulation, is to express the rotational motion in rotational rates and the translational motion in specific forces, rather than accelerations.

2) Specific Forces: Specific forces are the translational accelerations of the vehicle combined with gravity, i.e., what would be measured by an accelerometer. Thus, they are a measure of the total normal force acting on a driver. A car standing still on the earth has no (coordinate) acceleration, but the normal force required to counter gravity induces a specific force of magnitude g pointing upwards, with g being the gravity of earth. The human body perceives specific forces [161], [162], and not accelerations, through the otolith organs, a part of the human vestibular system located in the inner ear [163]. Therefore, specific forces are a more useful quantity than accelerations to describe the human motion perception of translational motion as generated by the simulator motion system.

In order to analyze the specific forces $\boldsymbol{f}_M^B$ acting on the Motion Reference Point (MRP) M in the simulator cabin frame B in terms of the simulator acceleration $\boldsymbol{a}^I$ in the inertial frame I, a conversion is required [160]. The specific forces can be calculated by:

$$\boldsymbol{f}_M^B = \boldsymbol{T}^{BI}(\boldsymbol{\beta})(\boldsymbol{a}^I + \boldsymbol{g}^I), \tag{1}$$

where $\boldsymbol{T}^{BI}(\boldsymbol{\beta})$ is the total rotation matrix as a function of the orientation $\boldsymbol{\beta}$ of the simulator, which thus relates the inertial and body frames. Hence, the MCA must control the simulator rotations $\boldsymbol{\beta}$ and accelerations $\boldsymbol{a}^I$ to create the desired specific forces $\boldsymbol{f}_M^B$. The rotation matrix $\boldsymbol{T}^{BI}(\boldsymbol{\beta})$ depends on the set of the three Euler angles φ, θ, and ψ, describing the roll, pitch, and yaw angles, around the x, y, and z axes, respectively. In driving dynamics, the x-axis typically points forward, y to the left (seen from the driver's perspective), and z upwards. The subsequent rotations around each angle are based intrinsically on the newly created coordinate system of the object by the previous rotation, such that the rotation sequence matters. The most common sequence is around the axes $z \to y' \to x''$ [164], i.e., the yaw rotation is applied first, followed by the pitch rotation, and then the roll rotation. The rotation matrix $\boldsymbol{T}^{BI}(\boldsymbol{\beta})$ also depends on the kinematic configuration of the simulator, i.e., in which order the rotations are defined and what components the motion system consists of. The transformations are calculated using the Euler transformation matrices, where the rotation angles are defined as counter-clockwise positive:

$$\boldsymbol{R}_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}, \quad \boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$
$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

As an example, the total sequence of rotations for the Sapphire Space simulator (see Fig. 2) are first the yaw angle ψ, pitch angle θ, and roll angle φ of the hexapod, followed by the yaw-drive with its angle ψ_d:

$$\boldsymbol{T}^{BI}(\boldsymbol{\beta}) = (\boldsymbol{R}_z(\psi)\, \boldsymbol{R}_y(\theta)\, \boldsymbol{R}_x(\varphi)\, \boldsymbol{R}_z(\psi_d))^\top. \tag{3}$$

Note that the order of the rotations depends on the kinematic configuration of the simulator [160]. If a yaw-drive would be present at the bottom rather than at the top, the yaw motion contributions of the hexapod and yaw-drive are in the same plane and can be combined as $\boldsymbol{R}_z(\psi+\psi_d)$, which simplifies the motion control. With the transformation matrix $\boldsymbol{T}^{BI}(\boldsymbol{\beta})$ known, the specific forces can be calculated as [160]:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_M^B = \boldsymbol{T}^{BI}(\boldsymbol{\beta}) \left(\begin{bmatrix} a_x^{\mathrm{XY}} + a_x^{\mathrm{hex}} \\ a_y^{\mathrm{XY}} + a_y^{\mathrm{hex}} \\ a_z^{\mathrm{hex}} \end{bmatrix}^I + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}^I \right), \tag{4}$$

where a^{hex} and a^{XY} denote hexapod and XY-drive accelerations, respectively. Due to the gravity vector $\boldsymbol{g}$, a rotation of the simulator cabin creates a sustained specific force in either the longitudinal or lateral direction. This principle can be exploited to create the illusion of a prolonged acceleration by controlling the angles $\boldsymbol{\beta}$ that specify the matrix $\boldsymbol{T}^{BI}(\boldsymbol{\beta})$, as long as the driver does not notice the accompanied cabin rotation. This is known as *tilt-coordination* [165]. The tilt-coordination technique only works if the human inside the simulator cabin cannot perceive the rotational motion of that cabin [166]. Hence, for tilt-coordination, the tilt angle must be smaller than $20\circ - 30\circ$ [167], and the tilt rate must stay below $2 - 4\,\frac{\mathrm{deg}}{s}$ [168], [169].

Note that the challenge of the MCA is to accurately control the simulator signals in the inertial frame (right hand side) to control the specific forces in the body frame, of which the conversion depends on the kinematic configuration of the simulator and its current attitude $\boldsymbol{\beta}$.

3) Rotational Rates: Apart from the otoliths, the human's vestibular system contains semicircular canals that detect rotational rates. The second task of the MCA is thus to create the desired rotational rates acting on the cabin ($\boldsymbol{\omega}^B$) by controlling the change in cabin rotation in the inertial system ($\dot{\boldsymbol{\beta}}$):

$$\boldsymbol{\omega}^B = \boldsymbol{J}(\boldsymbol{\beta})\dot{\boldsymbol{\beta}}. \tag{5}$$

The Jacobian matrix $\boldsymbol{J}(\boldsymbol{\beta})$ relates the two quantities. For the Sapphire Space, the rotational rates in the body frame $\boldsymbol{\omega}^B$ are derived from rotational rates in the inertial frame $\dot{\boldsymbol{\beta}}$ as [160]:

$$\begin{aligned} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_M^B &= \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_d \end{bmatrix} + \boldsymbol{R}_z^\top(\psi_d) \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + (\boldsymbol{R}_x(\varphi)\boldsymbol{R}_z(\psi_d))^\top \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \\ &\quad + (\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\varphi)\boldsymbol{R}_z(\psi_d))^\top \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} 0 & c_{\psi_d} & c_\varphi s_{\psi_d} & c_\theta s_\varphi s_{\psi_d} - c_{\psi_d} s_\theta \\ 0 & -s_{\psi_d} & c_{\psi_d} c_\varphi & s_\theta s_{\psi_d} + c_\theta s_\varphi c_{\psi_d} \\ 1 & 0 & -s_\varphi & c_\varphi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\psi}_d \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{\beta})\dot{\boldsymbol{\beta}} \end{aligned} \tag{6}$$

where $s_\star$ and $c_\star$ are $\sin(\star)$ and $\cos(\star)$, respectively. The equation for hexapod simulators can be derived by substituting $\psi_d = 0$ for the yaw-drive. In that case, assuming that the pitch and roll angles of the hexapod are small, a small angle approximation may be used to simplify the Jacobian matrix. However, yaw-drive angles can typically reach $\pm 180°$, such that a small angle approximation cannot be used. This results in a significant challenge for optimization-based motion cueing. Furthermore, as with the specific forces, the relations in (6) differ depending on the kinematic configuration due to differences in the rotation order, causing differences in the motion control of the MCA.

B. Reference Point Shift

The human vestibular system, containing the otoliths and semicircular canals, is located in the inner ears. As the motion is primarily perceived through the vestibular system, it is the motion of the vehicle acting at the location of the vestibular system (Cueing Reference Point (CRP), see Fig. 2) that is of prime interest. This point often differs from the point where the motion signals are applied: the Motion Reference Point (MRP). A typical choice for the MRP is the upper platform's centroid. The vector $\boldsymbol{r}_{MC}^B$ specifies the Cueing Reference Point (CRP) with respect to the MRP. The specific forces $\boldsymbol{f}_C^B$ that act on the CRP are then:

$$\boldsymbol{f}_C^B = \underbrace{\boldsymbol{f}_M^B}_{\text{(i)}} + \underbrace{\dot{\boldsymbol{S}}(\boldsymbol{\omega})\boldsymbol{r}_{MC}^B}_{\text{(ii)}} + \underbrace{\boldsymbol{S}^2(\boldsymbol{\omega})\boldsymbol{r}_{MC}^B}_{\text{(iii)}} + \underbrace{2\boldsymbol{S}(\boldsymbol{\omega})\dot{\boldsymbol{r}}_{MC}^B}_{\text{(iv)}} + \underbrace{\boldsymbol{a}_{MC}^B}_{\text{(v)}}, \tag{7}$$

where the subscript C stands for CRP, M for MRP, and $\boldsymbol{S}$ is a tensor matrix such that $\dot{\boldsymbol{T}}^{IB} = \boldsymbol{S}(\omega)\boldsymbol{T}^{IB}$ [160]. Equation (7) describes the main kinematic relation for rigid bodies. It consists of (i) the specific forces of the MRP in the body frame, (ii) the tangential acceleration, (iii) the centripetal acceleration, (iv) the Coriolis acceleration, and (v) the relative acceleration between the CRP and MRP [54]. The relation itself does not depend on the configuration of the motion system. However, the centripetal, tangential, and Coriolis acceleration depend on the rotational rates $\boldsymbol{\omega}_C^B$ of the rigid body. This difference between the CRP and MRP is ideally accounted for by the MCA, but can be neglected to simplify the control problem. Neglecting this difference can, however, lead to significantly erroneous specific force motion in the CRP [160]. For a rigid body, the rotational rates are equal for all points on that body. Thus, a reference shift, similar to that of the specific forces in (7) between the MRP and CRP, is not required.

C. Vestibular System Models

Equations (1), (5), and (7) allow to compute the specific forces and angular velocities acting at the CRP, i.e., the location of the vestibular system. Mathematical models describing the vestibular system itself can be incorporated in the MCA to take the human perception of motion into account. This can produce more realistic simulator motion [139] and remove unnecessary simulator motion. Having suitable otolith and semicircular canal models for the use in MCAs has been recognized as a major contributor to cueing quality [170]. In their review on otolith models, Asadi et al. [170] conclude that the mathematical model best suitable for use in MCAs regarding the specific force perception is given as [32]:

$$\frac{\hat{f}_\xi(s)}{f_\xi(s)} = 0.4\frac{1+10s}{(1+5s)(1+0.016s)}, \quad \xi \in \{x, y, z\}. \tag{8}$$

This transfer function takes as input the specific force f_ξ in ξ direction and outputs the sensed or perceived specific force $\hat{f}_\xi$. Similarly, in a review on semicircular canal models [171], the transfer function for rotational rate perception from [32]:

$$\frac{\hat{\omega}_\xi(s)}{\omega_\xi(s)} = 5.73\frac{80s^2}{(1+80s)(1+5.73s)}, \quad \xi \in \{x, y, z\}, \tag{9}$$

is concluded to fit best for use in MCAs. However, there is no widespread consensus on which model is the most suitable.

D. Motion Perception Thresholds

Another important aspect of human motion perception is motion perception thresholds, i.e., the minimal motion perceivable by a human. Knowledge on these thresholds can be exploited in motion cueing, e.g., for (pre-)positioning the simulator to advantageous simulator states without subjects noticing these motions. Perception threshold values depend on various factors, such as the type of motion, its direction, frequency, individuals, and whether the motion is applied on a single axis or different directions simultaneously [161]. Last, thresholds are also affected by interaction with other senses of perception [11], [172]. Typical values found in literature range between 3−6 deg/s [11], [169], [173], [174], [175] for rotational rates, 2.6−8 deg/s^2 [11], [175] for rotational accelerations and 0.05–0.17 m/s^2 [11], [168], [172] for translations.

E. Scale of Motion Reproduction

A last point of interest lies at understanding how much of the vehicle motion has to be reproduced by the simulator. Scaling factors for the reference vehicle motions were found to be “subjectively equivalent” within a wide range around 0.5 [176]. Furthermore, participants preferred motion scale factors within the range 0.4–0.75 over a scaling factor of 1 [143], [176]. Therefore, it is argued that instead of recreating the full, unscaled motion of a vehicle, it is enough or even preferred to only render a down-scaled version. This eases the challenge of recreating vehicle motions with the limited simulator workspace. One hypothesis for the preference of scaled motions is that this maintains perceptual coherence throughout the human’s multi-sensory environment perception, as visual or audio systems do not reach a level of full realism as well. Highly-reduced motion on the other hand, i.e., a scaling factor of 0 or slightly above, was judged as more sickening [143] and significantly less realistic [143], [176].

III. Filter-Based Motion Cueing Algorithms

This section gives a brief overview of filter-based MCAs. The working principles are explained along with the advantages, as well as the disadvantages, which motivate the focus on the more recent optimization-based MCAs in this review. The key publications are indicated in Fig. 3 by the orange block.

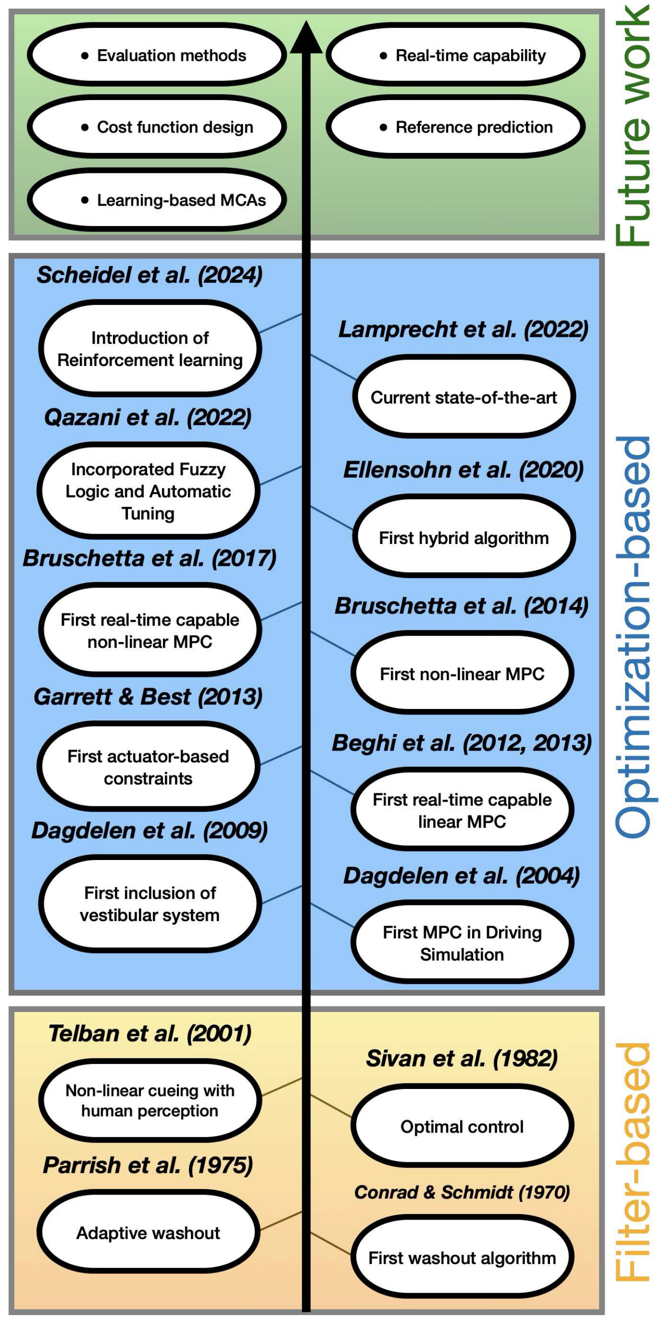


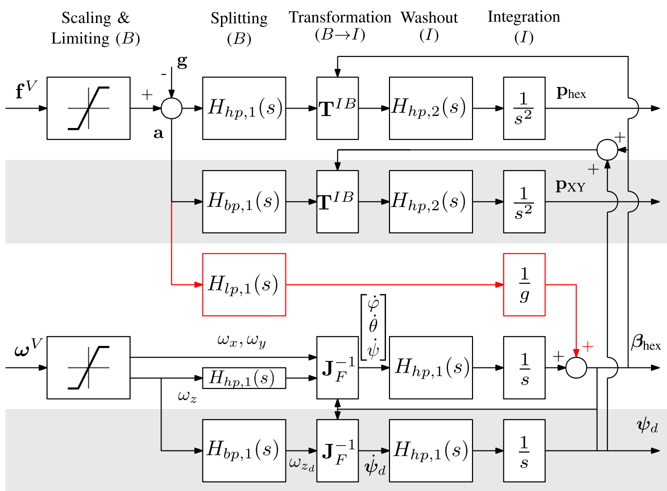
Fig. 3. Historical progression of the important milestones in motion cueing algorithms for driving simulation, with a focus on optimization-based MCAs.

A. Classical Washout Algorithm

Classical Washout Algorithms (CWAs) have been used extensively since the 1970s [9], [10] and nowadays still dominate in driving simulation.

1) Working Principle: A CWA calculates the movement commands for the simulator by applying a combination of linear scaling and filters to obtain the desired response in the six motion channels $(\boldsymbol{f}, \boldsymbol{\omega})$ that matches the motion of the real vehicle. Several steps constitute these algorithms, as indicated in Fig. 4.

First, the vehicle motions $(\boldsymbol{f}^V, \boldsymbol{\omega}^V)$ are scaled and/or limited. Scaling is a linear operation with static gain K. In limiting, values above a selected threshold are cut off [11]. Due to the limited motion workspace, most simulators can still not produce these sustained specific forces and rotational rates. Therefore, in the second step, the motion is split into low- and high-frequency components, using a high-pass filter with cut-off frequency ω_{hp}. Only the high-frequency motion is fed as desired motion to the simulator translational and rotational movement. Rotational rates ω_x and ω_y do not have to be filtered, as their magnitude is often small and most simulators are able to reproduce this motion one-to-one. To reproduce the low-frequency lateral and longitudinal specific forces, tilt-coordination can be used (red path in Fig. 4), split by a



Second-order filters

$H_{lp,2}(s) = \frac{\omega_{c,l}^2}{s^2+2\zeta\omega_{c,l}s+\omega_{c,l}^2}$

$H_{hp,2}(s) = \frac{s^2}{s^2+2\zeta\omega_{c,h}s+\omega_{c,h}^2}$

$H_{bp,2}(s) = H_{lp,2}H_{hp,2}$

First-order filters

$H_{lp,1}(s) = \frac{\omega_{c,l}}{s+\omega_{c,l}}$

$H_{hp,1}(s) = \frac{s}{\omega_{c,h}+s}$

$H_{bp,1}(s) = H_{lp,1}H_{hp,1}$

Fig. 4. Block diagram of the Classical Washout Algorithm (CWA) for a nine DoF motion system, such as the Sapphire Space. The top gray bar represents the XY-drive channel, the bottom gray bar the yaw-drive channel. The red path highlights the tilt-coordination.

low-pass filter with cut-off frequency ω_{lp}. Dividing by the gravitational constant g yields angles φ and θ that generate a sustained specific force in lateral and longitudinal direction, respectively [165].

In the case of additional redundant motion systems, such as an XY-drive, an additional band-pass response is used to reproduce the medium-frequency motion. In Fig. 4, this is illustrated by the top gray bar. For the rotational rates $\boldsymbol{\omega}$, the procedure is similar. After scaling and limiting, the yaw motion is filtered. In driving simulators, pitch and roll motion are generally not filtered, as their magnitude in driving scenarios is typically very small. In that case, a unity response can be used [177], [178]. Yaw motion is generally large in driving simulation and must be filtered. When using a yaw-drive, an extra channel is necessary (bottom gray bar in Fig. 4).

The desired simulator motion in the body frame is then transformed to the inertial frame using transformation matrices $\boldsymbol{T}^{BI}$ (see Sec. II-A). The rotational rates are converted to Cardan angle derivatives through the inverse transformation matrix $\boldsymbol{J}_F^{-1}$. After inertial-frame filtering, the angular derivatives are integrated to obtain the commanded Cardan angles.

The motion is then high-pass filtered in the inertial frame, such that the simulator is always brought back to its neutral position–from which the name 'washout' algorithm follows–preferably without the human subject noticing. Washout in the body frame is not possible, as the cross-coupling of the simulator axes would result in drift. Only where rotational motion is sufficiently small, filtering in the body frame is applicable [12]. Second-order filters are required to ensure that the response returns to zero after a step input [11]. Finally, the signals are integrated once (for rotational motion) or twice (for translational motion) to obtain the output of the algorithm, being the commanded orientations and positions. For the Sapphire Space, the translational motion consists of both the hexapod and XY-drive positions ($\boldsymbol{p}_{\text{hex}}$ and $\boldsymbol{p}_{\text{XY}}$). Note that, although not explicitly visible in Fig. 4, the described procedures are performed in each of the six motion channels (the entries of $\boldsymbol{f}$ and $\boldsymbol{\omega}$) separately, such that the parameters of the filters can be different in each channel.

2) Advantages of CWAs:

- The basic structure, using linear scaling and filters, gives the CWA a high degree of transparency. Its behavior can be deterministically determined, both in a given driving situation, and when changing parameters.
- By correctly designing the filters, the CWA is stable.
- The filters are linear, such that the same change in magnitude is obtained over the whole frequency spectrum. This is beneficial in simulations in which behavioral fidelity, or retaining the relative differences in vehicle motion between different test cases, is important.
- As calculating the transfer function responses is computationally cheap, fulfilling real-time constraints is possible.
- The CWA is relatively straightforward to implement.

3) Disadvantages of CWAs:

- The simulator states are not compared to the constraints of the platform during the simulation. This means that they must be accounted for beforehand in the tuning process [51], [56]. Using linear scaling and filters, therefore, requires that the tuning must always be done with the worst-case maneuver(s) in mind, which can lead to overly conservative simulator motions in other maneuvers.
- Tuning the CWA is labor-intensive. Finding fitting values for the large number of parameters (e.g., 21 for the 6-DoF system described in [179]) is difficult, as most parameters have no physical interpretation. This also implies that consistency in stimuli between vehicle and simulator movements cannot be guaranteed [51], [56]. The tuning procedure is often heuristic, few guidelines exist, and predicting how changes will affect the driver's perception is difficult [51], [134].
- As low-pass and high-pass filters produce phase lag and lead, respectively, there is an inherent phase shift of the simulator motion with respect to the vehicle [73], [146]. This is inevitable, as the CWA consists of causal filters, and can thus not consider future states.

B. Adaptive Washout Algorithm

Adaptive Washout Algorithms (AWAs) [14], [15], [16], [17], [18], [19], [20] address two drawbacks of the CWA: (1) the lack of simulator state feedback, and (2) the need to tune for worst-case maneuvers. The AWA follows the same principles as the CWA, but the parameters of the filters are changed in real-time during the simulation. Hence, the MCA still relies on filters that washout the motion towards the neutral state of the simulator, but becomes *closed-loop* by adapting the filter parameters based on the current workspace position. Parameters are chosen to minimize a cost function [20], typically taking into account:

- The current simulator motion state, i.e., deviation from the simulator neutral position;
- Errors between the vehicle and simulator motion;
- Differences between current and nominal parameter values.

In order to further increase workspace usage and motion fidelity of the MCA, the works [21], [22], [23], [24] obtain AWAs by adding fuzzy logic to the CWA, as fuzzy logic allows to accommodate different MCA inputs and better deal with uncertainty. Tuning of the AWA is usually done based on the expected worst-case specific forces. Steepest descent methods perform the real-time optimization. AWA has been successfully implemented in driving simulation [18], [19] and shown to increase the quality of the simulator motion [19]. Evolutionarily optimization techniques, such as the genetic algorithm (GA) or particle swarm optimization (PSO) are employed to optimize the MCA [25], [26], [27] or the fuzzy logic controller [28] in terms of workspace limitations, tilt-rate limits, and correlation and errors in the cueing signals. Due to loss of parameter transparency, tuning of the AWA is even more complicated compared to the CWA [150]. Furthermore, the consistency in cueing quality is reduced. Combined with stability issues, applications and research on AWAs have declined.

C. Optimal Washout Algorithm

To reduce the time consuming and mostly trial and error-based tuning of CWA and AWA, the Optimal Washout Algorithm (OWA) has been developed [13] and applied to driving simulation [33], [34], [35], [36], [37]. In this method, the MCA controlling the simulator is typically still a CWA. However, the filter parameters (i.e., gains, filter orders, and cut-off frequencies) are determined not through manual tuning, but through an optimization problem beforehand, i.e., offline, using a standard quadratic cost function that incorporates the human perception through a vestibular model:

$$\int_0^{T_{\mathrm{Sim}}} \big((\hat{\boldsymbol{y}}^S - \hat{\boldsymbol{y}}^V)^\top \boldsymbol{Q} (\hat{\boldsymbol{y}}^S - \hat{\boldsymbol{y}}^V) + \boldsymbol{x}^\top \boldsymbol{R} \boldsymbol{x} + \boldsymbol{u}^\top \boldsymbol{S} \boldsymbol{u}\big)\, \mathrm{d}t. \tag{10}$$

Here, $\hat{\boldsymbol{y}}^V = \begin{bmatrix} \hat{\boldsymbol{f}}^V & \hat{\omega}^V \end{bmatrix}^\top$ represents the vehicle motion (i.e., the reference) that was filtered using a mathematical vestibular model (see Subsec. II-C). Similarly, $\hat{\boldsymbol{y}}^S$ is the filtered simulator motion. Their quadratic difference is weighted using a matrix $\boldsymbol{Q}$. Values denoted with $\hat{(\cdot)}$ represent motions as perceived by the human driver. The second term penalizes excitations of the various motion subsystems states $\boldsymbol{x}$, containing the Cartesian position, velocity, and acceleration states of the simulator, weighted by $\boldsymbol{R}$. Finally, $\boldsymbol{u}$ are the inputs given to the system, which are penalized using $\boldsymbol{S}$ to reduce unnecessary control inputs. The cost function is evaluated over a segment of interest (typically, a recording of a drive) with length T_{Sim}, for which a recording of representative vehicle motion is supplied.

A problem of the above formulation is that the axes of hexapod simulators are strongly coupled in Cartesian coordinates. This makes it difficult to tune the weighting matrices and keep the simulator within workspace. Therefore, in a variation of the OWA, the simulator's actuator lengths, velocities, or accelerations are calculated and included as states $\boldsymbol{x}$ in the cost function instead of the Cartesian values, easing up the tuning process [16], [38], [39].

By including Fuzzy Logic, the OWAs in [40] and [41] show improved motion fidelity. Again, evolutionarily optimization techniques may be employed to optimize the MCA [42], [43] or the fuzzy logic parameters [40].

The main benefit of an optimal filter-based algorithm is that its tuning process is arguably easier than the tuning process for a CWA, as only the balance between the motion cue reproduction and the workspace adherence must be selected through $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{S}$. Furthermore, its explicit optimization nature generally implies better motion cueing quality than the CWA with manual tuning [32]. However, its structure gives three inherent disadvantages:

- Similar to AWA and CWA, a property of the OWA is that explicit platform constraints cannot be considered in the optimization [180]. The constraints of the system must be accounted for implicitly in the tuning process [51] through the weighting matrices.
- Because the core mechanism is still based on causal filtering, references regarding future states cannot be accounted for in the OWA, even when the OWA is supplied with a full recording of representative vehicle motion. This issue is aggravated by the resulting inherent phase shift of the simulator motion with respect to the vehicle reference motion, for which no compensation is possible [51].
- As in the CWA, a worst-case tuning must be employed [38]. An attempt at solving this issue employs adaptive optimal control [16], [34], [40] in the OWA, which allows the parameters of the filters to be varied throughout the drive. However, as mentioned by [38], this algorithm still suffers from the same disadvantages as standard adaptive algorithms, such as a high computational load and stability issues [16], [32], [172], [179].

IV. Optimization-Based Motion Cueing Algorithms

In the last decade, due to the disadvantages of filter-based MCAs, but also the advent of enhanced (real-time) computing power, research has shifted towards optimization-based methods which directly optimize for the control signals.

These algorithms can be divided into two categories: closed-loop (CL) and open-loop (OL). In open-loop driving, there is no active vehicle control by the driver. Therefore, the driver cannot affect the MCA output and subsequent simulator motion. Nevertheless, the term driver will be used in this paper to refer to the person inside the simulator, i.e., that experiences the motion, irrespective of whether they are actively driving or not. In closed-loop driving, the driver *can* actively control the vehicle, and therefore the motion of the simulator. This type of control currently represents the majority of the driving simulator use-cases. For example, less than 10% of recent studies at BMW, which operates the largest driving simulation center in the world, are performed OL. CL MPC-based, CL learning-based, and OL optimization-based MCAs are reviewed in Subsec. IV-A, IV-B and IV-C, respectively. Major milestones are shown in the blue part of Fig. 3.

A. Closed-Loop Model Predictive Control-Based MCAs

CL MPC-based MCAs must compute the simulator controls in real-time. By actively driving in the simulator, the driver controls the motions of the vehicle reference $\boldsymbol{y}^V$. A CL MCA must provide the motion system control inputs $\boldsymbol{u}$ that best recreate these motions with the limited simulator motion space at runtime. Clearly, the filter-based MCAs as presented in Sec. III are all suitable to be CL MCAs. In literature, CL MCAs are also referred to as Driver-in-the-Loop (DiL), Interactive Driving, or Online MCAs. Optimization-based MCAs that are used in a closed-loop fashion are generally based on model predictive control (MPC).

MPC iteratively solves an optimal control problem (OCP) over a finite prediction horizon N_p, producing the optimized control sequence $\boldsymbol{U} = \{\boldsymbol{u}_n\}_{n=0}^{N_p-1} = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N_p-1})$, of which only the first input $\boldsymbol{u}_0$ is applied. The prediction horizon refers to the future time steps over which predictions are made and control inputs are optimized. At every time step k, the sequence $\boldsymbol{U}$ is recalculated with updated problem values based on the current system state. To distisnguish these real-time steps from the prediction steps, we denote the prediction steps with n and omit k where it is possible and clear from the context. This way, a standard form of an MPC problem is given as:

$$\mathcal{J}^* = \min_{\boldsymbol{U}} \sum_{n=0}^{N_p-1} l(\boldsymbol{x}_n, \boldsymbol{u}_n) + l_f(\boldsymbol{x}_{N_p}) \tag{11a}$$

subject to

$$\boldsymbol{x}_{n+1} = \boldsymbol{h}(\boldsymbol{x}_n, \boldsymbol{u}_n) \qquad n = 0, \ldots, N_p - 1 \tag{11b}$$

$$\boldsymbol{u}_n \in \mathbb{U},\ \boldsymbol{x}_n \in \mathbb{X} \qquad n = 0, \ldots, N_p - 1 \tag{11c}$$

$$\boldsymbol{x}_{N_p} \in \mathbb{X}_f \tag{11d}$$

with states $\boldsymbol{x}$, system dynamics $\boldsymbol{h}$, and the cost function consisting of the stage cost l and the terminal cost l_f. The inputs and states are bound by the sets $\mathbb{U}$ and $\mathbb{X}$, respectively. A terminal constraint $\mathbb{X}_f$ is often applied to give theoretical guarantees of recursive feasibility and stability [181], [182], [183].

Advantages of MPC over non-predictive controller designs that are especially important for its application to MCAs are:

- MPC can handle state constraints such as (11c) and input constraints such as (11d) explicitly;
- Dynamics of the vestibular and simulator motion system can be explicitly taken into account in (11b);
- Due to the iterative calculation of optimal simulator input sequences, MCAs based on MPC can be online-capable;
- Knowledge on the future drive over the prediction horizon can be incorporated to leverage cueing performance.

1) Linear Time-Invariant MPC MCA: The first MPC-based MCA was introduced in 2004 [49] and later extended by the same authors [100] (see Fig. 3). Their MPC differs significantly from prevailing formulations, as it considers all motion channels independently and therefore cannot make use of tilt-coordination in the optimization. Hence, the workspace limitations are easily reached.

The basic structure of linear time-invariant (LTI) MPC MCAs, that is commonly used nowadays, was introduced by the different authors in [51], [52], and [53]. In the following, the state of the art LTI MPC MCA is presented.

MPC MCAs generally take the vehicle motion as the reference output:

$$\boldsymbol{y}^V = \begin{bmatrix} \boldsymbol{f}^V & \boldsymbol{\omega}^V \end{bmatrix}^\top = \begin{bmatrix} f_x^V & f_y^V & f_z^V & \omega_x^V & \omega_y^V & \omega_z^V \end{bmatrix}^\top. \tag{12}$$

Their goal is to calculate simulator control signals $\boldsymbol{u}$, most commonly translational accelerations and rotational velocities that lead to platform motions whose resulting rotational rates and specific forces $\boldsymbol{y}^S = \begin{bmatrix} \boldsymbol{f}^S & \boldsymbol{\omega}^S \end{bmatrix}^\top$ acting at the CRP resemble $\boldsymbol{y}^V$ as close as possible. Usually, a vestibular system model in state-space formulation described by:

$$\dot{\boldsymbol{x}}_{\text{Vest}} = \boldsymbol{A}_{\text{Vest}}\boldsymbol{x}_{\text{Vest}} + \boldsymbol{B}_{\text{Vest}}\boldsymbol{y}, \tag{13a}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C}_{\text{Vest}}\boldsymbol{x}_{\text{Vest}} + \boldsymbol{D}_{\text{Vest}}\boldsymbol{y}, \tag{13b}$$

is used to filter both $\boldsymbol{y}^V$ and $\boldsymbol{y}^S$, resulting in $\hat{\boldsymbol{y}}^V$ and $\hat{\boldsymbol{y}}^S$. The most commonly used transfer functions constituting (13) were shown in (8) and (9). This way, the MCA can take human motion perception into account and optimize for minimal *perceived* cueing errors $\hat{\boldsymbol{y}}^V - \hat{\boldsymbol{y}}^S$. Both these values are represented in the body frame of the simulator cabin.

The simulator control inputs $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{a} & \dot{\boldsymbol{\beta}} \end{bmatrix}^\top$ are given in the respective inertial frame (see Subsec. II-A) and are most often selected as accelerations for translations and velocities for rotations, as these are perceived by the vestibular system. Hence, for each translational (rotational) DoF i, $\boldsymbol{u} \in \mathbb{R}^{n_{\text{DoF}}}$ contains an entry a_i ($\dot{\beta}_i$). The simulator is modeled as an integrator chain. For each rotational DoF, the block diagonal simulator state space matrices $\boldsymbol{A}_S$ and $\boldsymbol{B}_S$ contain a block $\boldsymbol{A}_{S,i} = \begin{bmatrix} 0 \end{bmatrix}$, $\boldsymbol{B}_{S,i} = \begin{bmatrix} 1 \end{bmatrix}$, while each translational DoF shows as:

$$\boldsymbol{A}_{S,i} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{B}_{S,i} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{14}$$

The resulting linear simulator state-space model is defined by:

$$\dot{\boldsymbol{x}}^S = \boldsymbol{A}_S\boldsymbol{x}^S + \boldsymbol{B}_S\boldsymbol{u}. \tag{15}$$

The simulator state vector $\boldsymbol{x}^S$ hence contains the translational positions p_i and velocities v_i, as well as angles β_i of the simulator. As shown in Subsec. II-A, the resulting specific forces and rotational rates acting on the driver in the simulator are given by:

$$\boldsymbol{y}^S = \boldsymbol{g}\left(\boldsymbol{x}^S, \boldsymbol{u}\right) = \begin{bmatrix} \boldsymbol{T}^{BI}(\boldsymbol{\beta})(\boldsymbol{a} + \boldsymbol{g}) \\ \boldsymbol{J}(\boldsymbol{\beta})\dot{\boldsymbol{\beta}} \end{bmatrix}, \tag{16}$$

and, hence, a nonlinear function of the simulator states and control inputs. Combining the vestibular and simulator system models yields the total nonlinear system dynamics:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{x}}_{\text{Vest}} \\ \dot{\boldsymbol{x}}^S \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{\text{Vest}}\boldsymbol{x}_{\text{Vest}} + \boldsymbol{B}_{\text{Vest}}\boldsymbol{g}\left(\boldsymbol{x}^S, \boldsymbol{u}\right) \\ \boldsymbol{A}_S\boldsymbol{x}^S + \boldsymbol{B}_S\boldsymbol{u} \end{bmatrix}, \tag{17a}$$

$$\hat{\boldsymbol{y}}^S = \boldsymbol{C}_{\text{Vest}}\boldsymbol{x}_{\text{Vest}} + \boldsymbol{D}_{\text{Vest}}\boldsymbol{g}\left(\boldsymbol{x}^S, \boldsymbol{u}\right). \tag{17b}$$

Note that the only nonlinear term in the equation is $\boldsymbol{g}\left(\boldsymbol{x}^S, \boldsymbol{u}\right)$. For simulators and MCAs that allow only small $\boldsymbol{\beta}$, a small angle approximation for $\boldsymbol{T}^{BI}(\boldsymbol{\beta})$ and $\boldsymbol{J}(\boldsymbol{\beta})$ can be used. The remaining nonlinearities are removed by keeping the angles $\boldsymbol{\beta}$

constant over the prediction horizon. This allows to write the linearized equation of form:

$$\boldsymbol{y}^S = \boldsymbol{T}_1\boldsymbol{x}^S + \boldsymbol{T}_2\boldsymbol{u}, \tag{18}$$

with $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ depending on the simulator configuration. The resulting state-space representation, delivering the linearized total system dynamics of (11b), is:

$$\begin{bmatrix}\dot{\boldsymbol{x}}_{\text{Vest}}\\ \dot{\boldsymbol{x}}^S\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_{\text{Vest}} & \boldsymbol{B}_{\text{Vest}}\boldsymbol{T}_1\\ \boldsymbol{0} & \boldsymbol{A}_S\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{\text{Vest}}\\ \boldsymbol{x}^S\end{bmatrix} + \begin{bmatrix}\boldsymbol{B}_{\text{Vest}}\boldsymbol{T}_2\\ \boldsymbol{B}_S\end{bmatrix}\boldsymbol{u}, \tag{19a}$$

$$\hat{\boldsymbol{y}}^S = \begin{bmatrix}\boldsymbol{C}_{\text{Vest}} & \boldsymbol{D}_{\text{Vest}}\boldsymbol{T}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{\text{Vest}}\\ \boldsymbol{x}^S\end{bmatrix} + \boldsymbol{D}_{\text{Vest}}\boldsymbol{T}_2\boldsymbol{u}. \tag{19b}$$

Finally, the system is discretized. In order for the driver to perceive a delay-free reaction in the simulator motion to the driver-induced vehicle motion, high sampling rates of 100 Hz or above are generally aimed for [64], [75], [76], [77], [78], and [79].

In most implementations, the MPC cost function is chosen as [75], [78], [80], [81], and [82]:

$$\begin{aligned}\mathcal{J} = &\sum_{n=0}^{N_p} \underbrace{(\hat{\boldsymbol{y}}_n^S - \hat{\boldsymbol{y}}_n^V)^\top \boldsymbol{Q}(\hat{\boldsymbol{y}}_n^S - \hat{\boldsymbol{y}}_n^V)}_{\text{Tracking term}} + \underbrace{\boldsymbol{x}_n^{S,\top}\boldsymbol{R}\boldsymbol{x}_n^S}_{\text{State term}}\\ &+ \sum_{n=0}^{N_c} \underbrace{\boldsymbol{u}_n^\top \boldsymbol{S}\boldsymbol{u}_n}_{\text{Input term}} + \underbrace{\Delta\boldsymbol{u}_n^\top \boldsymbol{K}\Delta\boldsymbol{u}_n}_{\text{Change in input term}}\quad ,\end{aligned} \tag{20}$$

and minimized over the difference in lifted control inputs $\Delta\boldsymbol{u}_n = \boldsymbol{u}_n - \boldsymbol{u}_{n-1}$ to achieve a certain degree of regularity in the control signals sent to the motion system. Here, N_p and N_c describe the prediction and control horizon, respectively. The cost function in (20) contains four terms. First, the tracking aims to minimize the difference between the reference data $\hat{\boldsymbol{y}}_n^V$ and generated simulator motion $\hat{\boldsymbol{y}}_n^S$. Second, the state term penalizes deviations from the neutral state. Whereas this can function as a washout component in the optimization, it is also often included to add convexity to the cost function to speed up the optimization [184]. The third and fourth terms penalize large inputs and changes in input, respectively, resulting in a smoother optimization variable.

Constraints are chosen as workspace box constraints, respecting the limitations of the simulator:

$$\begin{aligned}&\underbrace{\begin{bmatrix}\boldsymbol{p}_{\min}\\ \boldsymbol{v}_{\min}\\ \boldsymbol{\beta}_{\min}\end{bmatrix}}_{\boldsymbol{x}_{\min}} \leq \underbrace{\begin{bmatrix}\boldsymbol{p}_n\\ \boldsymbol{v}_n\\ \boldsymbol{\beta}_n\end{bmatrix}}_{\boldsymbol{x}_n} \leq \underbrace{\begin{bmatrix}\boldsymbol{p}_{\max}\\ \boldsymbol{v}_{\max}\\ \boldsymbol{\beta}_{\max}\end{bmatrix}}_{\boldsymbol{x}_{\max}} \quad \text{for } n = 1, 2, \ldots, N_p,\\ &\underbrace{\begin{bmatrix}\boldsymbol{a}_{\min}\\ \dot{\boldsymbol{\beta}}_{\min}\end{bmatrix}}_{\boldsymbol{u}_{\min}} \leq \underbrace{\begin{bmatrix}\boldsymbol{a}_n\\ \dot{\boldsymbol{\beta}}_n\end{bmatrix}}_{\boldsymbol{u}_n} \leq \underbrace{\begin{bmatrix}\boldsymbol{a}_{\max}\\ \dot{\boldsymbol{\beta}}_{\max}\end{bmatrix}}_{\boldsymbol{u}_{\max}} \quad \text{for } n = 1, 2, \ldots, N_c,\\ &\Delta\boldsymbol{u}_{\min} \leq \Delta\boldsymbol{u}_n \leq \Delta\boldsymbol{u}_{\max} \quad \text{for } n = 1, 2, \ldots, N_c,\\ &\Delta\boldsymbol{u} = 0 \quad \text{for } n = N_c+1, \ldots, N_p.\end{aligned} \tag{21}$$

A Quadratic Programming (QP) problem is obtained by reformulation of the MPC scheme, resulting in an optimization problem of the following form [183]:

$$\begin{aligned}\arg\min_{\Delta\boldsymbol{U}} J = &\frac{1}{2}\Delta\boldsymbol{U}^\top \boldsymbol{H}\Delta\boldsymbol{U} + \Delta\boldsymbol{U}^\top \boldsymbol{h}_{\text{QP}}\\ \text{s.t. } &\boldsymbol{A}_{\text{QP}}\Delta\boldsymbol{U} \leq \boldsymbol{b}.\end{aligned} \tag{22}$$

For QPs, a range of fast and reliable real-time solvers exists [185], [186], [187]. Even without using a terminal set, the algorithm is reported to rarely experience infeasibility [51], [54].

To enable faster computations, most early works performed small simplifications of the presented LTI MPC MCA [49], [50], [51], [52], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68]. By dividing DoFs into two tilt-coordinated pairs, i.e., longitudinal/pitch and lateral/roll, as well as two single DoFs being vertical and yaw, it becomes possible to run separate MPCs with reduced complexity for the four subproblems. The issue with these simplifications is that the combined control signals from the separate MPCs may lead to constraint violations due to the approximation of independent DoFs. In the presence of a yaw-table, resulting large yaw motions do not allow to decouple longitudinal and lateral motions. To solve this, [69] and [60] optimize a 1-DoF yaw model and a complete 4-DoF tilt coordination model, where the latter deals with lateral, longitudinal, roll and pitch motions together. This leads to a linear, but time-varying, MPC MCA.

Having presented the LTI MPC MCA baseline formulation, in the following, we delve deeper into how existing research has advanced MPC-based MCAs. Specifically, we identify decoupling the vestibular systems, tilt rate limiting, stability and recursive feasibility, actuator constraints, nonlinear MPC, approaches to reduce computational load, and hybrid MCAs as important streams in literature. While these approaches and directions have managed to solve crucial issues of MPC-based MCAs to a sufficient extent, the open challenges of evaluating and comparing MCAs, cost function design and tuning, real-time capability, and vehicle motion reference prediction are detailed and discussed as key challenges in Sec. V.

2) Vestibular System and Tilt Rate Limiting: The human perception model is usually considered with coupled otolith organs and semicircular channels (see (13)). By decoupling them, i.e., modeling otoliths and semicircular channels separately in two MPCs running in parallel, angular velocities due to reference rotations can be taken into account independent of angular velocities due to tilt coordination [70]. Hence, tuning can be performed without simultaneously considering these two perception channels. Building on this principle, a tilt rate limit block can be added [71] that only constrains rotational motions due to tilt coordination, as these should stay under the human perception threshold. Reference rotations from the vehicle model should be perceivable. For this, two MPC-based MCAs are used again, where one is responsible for tilt coordination and linear specific forces, the other one for rotational motions. The prior uses the otolith organ and semicircular canal mathematical models and the latter only the semicircular canal model. However, it is still an open research question wether tilt rate limiting leads to an increase in perceived cueing quality [188], as low tilt rates may also lead to false cues.

3) Stability and Recursive Feasibility: A multitude of MPC approaches add terminal constraints and terminal cost to their MPC formulation to guarantee stability and recursive feasibility of (11) [181], [189]. The latter refers to the property that if a feasible solution to the optimization problem (11) exists at the current step, a feasible solution will also exist at subsequent steps. Guaranteeing this is a challenging task for complex driving simulator systems [83], [84]. The MPC MCA formulation in [83] uses an LQR terminal condition, which was not allowing real-time use. The MPC MCAs in [48] and [85] use the origin as the terminal set. Less restrictive terminal constraint sets are used in [49] and [50], which require the terminal state to be in the set of simulator states from which it is possible to bring the simulator to a standstill using only decelerations below the human's perception threshold. This way, false cues, i.e., cues with opposite sign to the reference motion, are avoided. As the human perception threshold on specific forces is quite low (see Sec. II-C), this type of terminal constraint set is very restrictive and limits the usage of available workspace, deteriorating the motion cueing quality. To circumvent this issue, a braking law is introduced in [83] and [84] and used in for example [58], [61], [63], [69], [79], [86], [87], which is particularly efficient with short prediction horizons. For each DoF $\xi \in \{p_x, p_y, p_z, \varphi, \theta, \psi\}$, the constraint

$$\xi_{\min} < \xi + c_v T_{\text{brake},\xi}\dot{\xi} + \frac{1}{2}T^2_{\text{brake},\xi}\ddot{\xi} < \xi_{\max} \tag{23}$$

is added, meaning that only those ξ, $\dot{\xi}$, $\ddot{\xi}$ are allowed, for which braking to a standstill within boundaries is possible. The coefficient c_v must be chosen in $0 < c_v < 0.5$ for the simulator to stay inside the workspace and allow a smooth deceleration. The time to brake $T_{\text{brake},\xi}$ is calculated as $T_{\text{brake},\xi} = \frac{v_{\max,\xi}}{a_{\text{brake},\xi}}$, where $a_{\text{brake},\xi}$ is either selected as the maximum simulator deceleration or the human perception threshold in the respective DoF. By adding (23) to the MPC, feasibility of the problem is guaranteed at the cost of a further reduced allowed workspace.

In general, the ideal terminal constraint is a Maximal Control Invariant (MCI) set, as it is the largest set that guarantees recursive feasibility [181], [182]. Hence, more workspace can be utilized and the motion of the algorithm feels less conservative. The calculation of MCI sets for MCAs is difficult, however, due to the high dimensionality of the problem [190]. Hence, the authors of [88] propose a simplification of MCI sets by using periodic-controlled invariant sets, which they derived in [89]. In essence, this presents an inner approximation of the MCI that is described by a lower number of constraints that need to be added to the optimization problem, while still guaranteeing feasibility. Nevertheless, downsides reported are conservative behavior, as well as the algorithm not working for hexapods or more complex system models with delays and nonlinearities [88]. The approach is extended in [64] to deal with system models that include delays. This is achieved by the more restrictive choice of an admissible controlled invariance set under the delay-free system dynamics as the terminal constraint set. A maximal admissible set is used in [90].

Due to the added complexity of the MPC MCA and tightened available workspace, even in the vast majority of very recent works, no stability and feasibility conditions are considered [46], [47], [48], [73], [74], [77], [78], [81], [91], [92], [93], [136]. Instead, these works rely on a sufficiently long prediction horizon to avoid stability issues.

4) Actuator Constraints: The MPC-schemes so far included rotational and translational limitations only as block constraints in Cartesian workspace coordinates, as shown in (21). The advantage of such formulations lies in their efficient handling by solvers. Additionally, simulators are often characterized by these values in their specification sheets and in research publications. However, due to a coupling of simulator DoFs in structures like the often used hexapod, workspace limitations are a function of the simulator configuration. This can lead to relatively large deviations between commonly used box constraints approximation of workspace and the real limitations of the simulator motion envelope due to limitations in joint coordinates. Effectively, this requires a downscaling of the box constraints to ensure that the actuator constraints are not reached. By adding constraints on the position $\boldsymbol{q}$, velocity $\dot{\boldsymbol{q}}$, and acceleration $\ddot{\boldsymbol{q}}$ of the actuators, i.e.,

$$\begin{aligned} \boldsymbol{q}_{\min} \leq \boldsymbol{q}_n \leq \boldsymbol{q}_{\max} \quad & \text{for } n = 1, 2, \ldots, N, \\ \dot{\boldsymbol{q}}_{\min} \leq \dot{\boldsymbol{q}}_n \leq \dot{\boldsymbol{q}}_{\max} \quad & \text{for } n = 1, 2, \ldots, N, \\ \ddot{\boldsymbol{q}}_{\min} \leq \ddot{\boldsymbol{q}}_n \leq \ddot{\boldsymbol{q}}_{\max} \quad & \text{for } n = 1, 2, \ldots, N, \end{aligned} \tag{24}$$

the actual motion platform limitations, instead of the approximated box constraints (21), are considered in the optimization. An inverse kinematic problem derives the acceleration, velocity, and position of the joints, i.e., on actuator level, based on the acceleration $\ddot{\boldsymbol{w}} = \begin{bmatrix}\boldsymbol{a} \ \ddot{\boldsymbol{\beta}}\end{bmatrix}^\top$, velocity $\dot{\boldsymbol{w}} = \begin{bmatrix}\boldsymbol{v} \ \dot{\boldsymbol{\beta}}\end{bmatrix}^\top$, and position $\boldsymbol{w} = \begin{bmatrix}\boldsymbol{p} \ \boldsymbol{\beta}\end{bmatrix}^\top$ on workspace level. For parallel robot structures, the inverse kinematics are in general described by:

$$\boldsymbol{q} = \boldsymbol{f}_{\text{IK}}(\boldsymbol{w}), \tag{25a}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_q^{-1}(\boldsymbol{w})\dot{\boldsymbol{w}}, \tag{25b}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_q^{-1}(\boldsymbol{w})\ddot{\boldsymbol{w}} + \dot{\boldsymbol{J}}_q^{-1}(\boldsymbol{w})\dot{\boldsymbol{w}}, \tag{25c}$$

where $\boldsymbol{J}_q^{-1}$ is the inverse Jacobian matrix and $\boldsymbol{f}_{\text{IK}}$ a nonlinear function. The entries of the inverse Jacobian matrix are nonlinear in the current workspace configuration $\boldsymbol{w}$ [191]. Therefore, a consequence of including (24) in the MPC formulation is that the nonlinear kinematic relations (25) between actuator space and workspace of the motion system must be computed. Solving the resulting nonlinear MPC poses a significant challenge for CL optimization, as the algorithms need to be real-time capable [67]. Before examining nonlinear MPC MCAs in the next subsection, different works that propose simplifications to solve only a linear problem when including actuator constraints are presented.

Garrett and Best [57] propose a linearization approach for hexapod-structure driving simulators. To this end, the Jacobian matrix, describing the kinematic transformation from actuator space to workspace, is kept constant such that the resulting equation for actuator velocities is linearized, giving:

$$\dot{\boldsymbol{q}}_n = \underbrace{\boldsymbol{J}_q^{-1}(\boldsymbol{w}_k)}_{\text{const. over } N_c} \dot{\boldsymbol{w}}_n. \tag{26}$$

The actuator positions are obtained by a forward Euler integration, giving the desired linear relationship:

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + T_{\text{samp}} \underbrace{\boldsymbol{J}_q^{-1}(\boldsymbol{w}_k)}_{\text{const. over } N_c} \dot{\boldsymbol{w}}_n. \tag{27}$$

with the sampling time T_{samp}. Due to their use of a decoupled system model, safety margins of $5-10\%$ had to be introduced to avoid constraint violations. Building on that, [93] extends the linearization approach to the acceleration level,

$$\ddot{\boldsymbol{q}}_n = \boldsymbol{J}_q^{-1}(\boldsymbol{w}_k)\ddot{\boldsymbol{w}}_n + \dot{\boldsymbol{J}}_q^{-1}(\boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}_k}\dot{\boldsymbol{w}}_n, \tag{28}$$

as well as to simulator systems with redundant DoFs. To ease computation, instead of (28), joint accelerations are computed through a finite difference approach in [94]:

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{\dot{\boldsymbol{q}}_{n+1} - \dot{\boldsymbol{q}}_n}{T_{\text{samp}}}. \tag{29}$$

As limitations on position level are commonly most constraining for hexapod structures [146], the MPC MCA in [54] drops constraints on velocity and acceleration level to reduce complexity.

An alternative way to handle the nonlinearities of the simulator motion system when using actuator constraints is suggested in [67] and extended by the same authors in [48]. Similar to [93], the works in [67] and [48] use linearized inverse kinematics, but then include the actuator states directly into the system model, leading to a linear time-*varying* MPC formulation, as the entries of $\boldsymbol{J}^{-1}(\boldsymbol{w}_n)$, now appearing in the augmented state matrices, change dependent on the current hexapod configuration $\boldsymbol{w}_n$ over the prediction horizon. Compared to the previous linear time-invariant MPC formulations without actuator states in the system model and hence overly conservative workspace box constraints, an increase in motion cueing quality and workspace usage is reported. Due to the high computational load of the time-varying algorithm, an application to real-time CL simulator rides in more complex scenarios shows challenging [85]. For the use in an urban driving scenario [85], the authors therefore exclude the actuator constraints again, making the problem time-invariant and hence also real-time capable. In [46], the nonlinear problem formulation for a 6-DoF driving simulator is discretized with a multiple shooting technique, linearized, and brought to a QP problem using a warm-start technique and the `ACADO` toolkit [192] to achieve real-time capability.

5) Nonlinear MPC MCA: Nonlinear MPC MCA generalizes linear MPC MCA by explicitly considering nonlinear system dynamics, nonlinear constraints, or possibly nonlinear cost functions in the optimization problem (11). Nonlinear MPC achieves better control performance for systems with nonlinearities. Few fully nonlinear MPC MCA approaches are described in literature, which, besides slight modifications to improve cueing quality, mainly differ in the used numerical methods and the complexity of the driving simulators. Nonlinear relations between DoFs of the motion system are considered for the first time in [76] (see Fig. 3). The inverse kinematics are not linearized; hence, nonlinear constraints on actuator level are used. A quadratic cost function similar to (20) was chosen with a short prediction window, and the `ACADO` framework was used to solve the nonlinear MPC problem. The algorithm was not real-time capable, but handled workspace exploitation and constraint satisfaction well considering the complex 9-DoF simulator. By performing minor simplifications and a more efficient implementation, the same authors report real-time capability for a horizon length of 0.3 s, i.e., $N_p = 30$ in [95].

In [82] a nonlinear least squares objective with linear constraints for a serial 8-DoF kinematic structure is presented, using a Real Time Iteration [193] scheme based on sequential quadratic programming implemented in `CasADi`'s `HPMPC` solver [194]. The authors of [96] provide a full nonlinear MPC formulation for a hexapod using `MAT-MPC` [195]. In [97], using a gradient-based augmented Lagrangian approach implemented in the `GRAMPC` software [196], real-time capability for a 7-DoF system is achieved. The same authors improved the algorithm in [47] by adding adaptive weights in the cost function to suppress false cues and a better prediction of the future reference to be tracked.

Instead of choosing the controlled variables on workspace level $\left(\boldsymbol{u} = \left[\boldsymbol{a}\ \dot{\boldsymbol{\beta}}\right]^\top\right)$, they can alternatively be selected on actuator level, i.e., $\boldsymbol{u} = \ddot{\boldsymbol{q}}$ [93]. This allows a direct linear incorporation of actuator limits on acceleration, velocity, and position level, as the inverse kinematics of parallel robots do not need to be computed first. Hence, the simulator motion envelope description transforms from a complex, time-variant formulation into independent and constant upper and lower boundaries for the six actuators of the hexapod. The actuator-based optimization approach presented in [93] is only applied to non-redundant simulator structures, as a description of the optimization problem for redundant structures becomes significantly more complex. Another downside is that in order to get the translational accelerations and rotational velocities in workspace coordinates required for the objective function, the direct kinematics of the parallel structure is required. Their calculation leads to a computationally expensive, nonlinear system of equations:

$$\dot{\boldsymbol{w}} = \boldsymbol{J}_q\dot{\boldsymbol{q}}, \quad \ddot{\boldsymbol{w}} = \boldsymbol{J}_q\left(\ddot{\boldsymbol{q}} - \dot{\boldsymbol{J}}_q^{-1}\dot{\boldsymbol{w}}\right), \tag{30}$$

with $\boldsymbol{J}_q$ being the same Jacobian matrix as in the inverse kinematics (25). To address this, [93] introduces two approximations of the direct kinematics, leading to a quadratic and a linear relation between actuator space and workspace, allowing for a simplified nonlinear and linear MPC formulation.

6) Approaches to Reduce Computational Load: Different approaches, apart from linearizing the simulator kinematics, have been presented to deal with the challenging topic of real-time capability in MPC MCA formulations. Previous work [63], [83], [98] has focused on designing an MCA based on explicit MPC [197]. Explicit MPC pre-computes the control law for all possible states, resulting in a piecewise affine function that can be evaluated in real-time with neglectable computational effort. Due to an exponential increase of complexity with problem dimensionality, none of the works [63], [83], [98] achieve a prediction horizon of more than $N_p = 2$ steps, even for a reduced set of used DoFs. Therefore, these existing approaches cannot compete with implicit MPCs. In

[98], an explicit MPC formulation is applied to warm-start an implicit MPC, leading to a reduction of computation times of the implicit algorithm by 30%.

As the length of the prediction horizon significantly influences the computational burden, [99] suggests a short prediction horizon MPC MCA, where exponential weighting is applied to lower the negative impact of the shorter prediction horizon on performance and stability [198]. The algorithms in [56], [59], [73], [78], and [82] use a move blocking-strategy [199], giving a non-uniform step size to reduce computational time by using a lower number of steps in the prediction horizon. Towards the end of the horizon, samples are taken less frequently, as points further away may have a higher uncertainty and can thus be considered less important in the optimization. Similarly, the step times can be geometrically increased over the prediction horizon [69]. In a different approach to deal with the real-time capability issues, the linear QP may be solved using a continuous-time recurrent neural network-based gradient method as a QP solver [77]. Last, Seefried [100] presents a method using B-splines that implicitly fulfill constraints. Even in the case of a premature stoppage of the optimization due to real-time constraints, the outcome never violates constraints. Still, real-time capability was not achieved.

7) Hybrid MCAs: Additionally, a range of so-called hybrid MCAs exists [79], [90], [136], [200], where different MCAs are combined with the goal of leveraging the advantages and canceling out the weaknesses of the respective approaches.

Most notably, in [136] (see Fig. 3), a filter-based and an open-loop optimization-based MCA are combined. For a pre-defined route to be simulated, an OL optimization-based MCA pre-computes an optimal motion cueing for a representative reference vehicle drive. In order to have a similar quality of motion cueing in specific CL drives, the results from the OL algorithm are used as a baseline for the CL simulation. Differences in driving behavior between the reference and CL driver are corrected by a filter-based MCA. The signal sent to the simulator motion system is therefore a superposition of the OL computed baseline with the corrective CL signals.

The main advantages of this hybrid approach are that no driver input prediction is necessary, which is often computationally expensive and inaccurate. Also, to some extent, the high motion cueing quality and workspace usage of not real-time capable optimization-based MCAs can be transferred to CL drives. The main downside of the presented hybrid algorithm is that it can only be used in highly controlled drives with little variance, i.e., one where the route is pre-defined, contains only single lane roads with speed limits, as well as no interaction with other road users. Otherwise, the reference drive is insufficiently representative, i.e., the CL driver will show different behavior than the reference driver, rendering the optimal motion cueing calculated by the open-loop optimization-based MCA irrelevant.

To summarize the literature on closed-loop MPC-based MCAs, Tab. I shows the key developments and main works in this field. It allows to rapidly asses the relevant components included in the respective MPC MCA (nonlinear OCP, actuator constraints, vestibular system, DOFs, subjective evaluation), as well as the prediction horizon, control rate, and wether real time capability was achieved. Additionally, the main contribution of each of the works is presented in the last column. This allows a comparative analysis based on the included objective criteria. Clearly, subjective evaluations of the various MCAs are rare, Telban and Cardullo's [32] vestibular model (VS2) is most used, stability and feasibility guarantees are not common, and real-time capability is oftentimes achieved by means of subsampling or significantly shorter control horizons.

TABLE I

COMPARISON OF KEY WORKS IN THE FIELD OF CLOSED-LOOP MPC-BASED MCAS ALONG THEIR MAIN CONTRIBUTION. NL = NONLINEAR; AC = ACTUATOR CONSTRAINTS; VS = VESTIBULAR SYSTEM; VS1: [201]; VS2: [32]; VS3: [202], [203]; SE = SUBJECTIVE EVALUATION; S/F = STABILITY/RECURSIVE FEASIBILITY; f_S: SAMPLING RATE; †: EXTENSIVE USE OF MOVE BLOCKING/SUBSAMPLING, ∗: SIGNIFICANTLY SHORTER CONTROL HORIZON

Works	NL	AC	VS	DOF	SE	S/F	RT	T_p [s]	f_S [Hz]	Main Contribution
[49]				8		S	X	?	?	First MPC-based MCA
[50]			VS1	8		S	X	?	?	First VS in MPC-based MCA
[51], [52]			VS2	6			X	0.3	100	Set MPC design standard
[83]				8		S/F	X	0.04	125	Explicit MPC MCA
[84]				8		S/F	X	0.04	125	Braking law for S/F
[57]		X	VS1	6	X		X	0.2^*	40	First actuator constraints
[56]			VS2	6			X	$8.0^\dagger$	100	VMRP for racing
[78], [116]			VS2	9			X	$8.0^\dagger$	100	VMRP for racing advanced
[69]			VS2	9		S/F	X	1.2^*	125	Separation into sub-systems
[62]			VS2	9		F		0.15	200	MPC MCA tuning procedure
[77]				8			X	0.3	100	RNN used as Solver
[70]			VS2	6			X	1.5^*	100	Decouples vestibular system
[71]			VS2	6			X	?	100	Decouples VS + tilt rate limit
[101]		X		9	X		X	1.5^*	100	Parallelization of MCA
[65], [66]			VS2	6			X	1.5^*	100	Hexarot&Gantry-Tau Models
[94]		X	VS2	9	[80]			3.0^*	100	Linearized AC
[85]			VS2	6		S/F	X	1.5^*	100	Stability & terminal condition
[48]		X	VS2	6		S/F	X	1.5^*	100	LTV MPC MCA
[114]				?			?	4^*	100	First VMRP based on a NN
[87]			VS2/3	8	X	S/F	X	$6^{*,\dagger}$	125	Compares VS2,VS3 & no VS
[105]			VS2	6			X	1.5^*	100	Fuzzy logic & GA tuning
[76]	X	X	VS2	9				?	100	First NMPC MCA
[95]	X	X	VS2	9			X	$2.9^\dagger$	83	First real-time NMPC MCA
[82]	X	X		8		F	X	0.3	100	Constraint tightening
[97]	X	X	VS3	7	[144]		X	6.0	250	Very fast computation
[93]	X	X		6				1.2^*	125	MCA with $\boldsymbol{u} = \ddot{\boldsymbol{q}}$
[92], [113]	X	X		5	[146]		X	10.0	40	Exponential Weights&VMRP
[73]	X	X		6			X	$10^\dagger$	100	VMRP with vehicle model
[115]	X	X	VS3	7			X	4.0	250	Optimal control based VMRP
[46]	X	X	VS2	6			X	0.5	100	Adaptive weight MPC MCA
[90]	X			2		F	X	?	?	Combines MPC & NMPC
[98]	X			6		S	X	0.5	100	Combines ex-&implicit MPC
[109]	X	X	VS2	6			X	0.5^*	100	GA+PSO based tuning
[47]	X	X	VS3	7	X		X	4.0	250	Most complete NMPC MCA

B. Closed-loop Learning-based MCAs

Learning based MCAs have emerged only recently and are still in their infancy. However, applying learning to MCAs is a promising direction. In the few existing approaches [124], [125], [126], [127], [128], [129], [130], [131], [132], Neural Networks (NNs) perform either partial tasks of filter-based or optimization-based MCAs, or act as a full MCA themselves. Partial tasks are taken over by NNs in [124] and [125], where a feedforward NN is trained based on the output of three differently tuned CWAs, representing good parameters for slow, medium, and fast driving. Subsequently, the trained NN replaces the filters in a CWA, enabling a fast computation without a conservative, worst-case focused CWA tuning.

Koyuncu et al. [126] use a nonlinear autoregressive network to learn the behavior of a linear OL optimization-based MCA, as these algorithms give the highest motion fidelity. The resulting fully NN-based MCA can compute motions in real-time, hence allowing CL use with an approximated model of the OL MCA. While the cueing quality was higher than with a benchmark CWA, generalization of the NN-MCA was limited. Additionally, using supervised learning, the cueing quality is limited to that of the linear OL MCA which the NN imitates. Similarly, using a differentiable predictive control framework, [130] trains a NN to mimic a nonlinear MPC MCA including a nonlinear actuator-space model.

Very recently, deep reinforcement learning (DRL) based MCAs were proposed by Scheidel et al. [127], [128], [129]. A Proximal Policy Optimization (PPO) algorithm [204] in an actor-critic implementation trains a NN prior to the study, which then generates the simulator controls in real time using only the forward propagation through the NN during CL usage. After the proof of concept [127], the work [128] adds a vestibular system, enables more general maneuvers and tunes hyperparameters using a Bayesian optimization, achieving a higher cueing quality than a benchmark CWA. In the most recent work [129], the approach is extended from a simplified 2-DoF setup to a full 6-DoF description, achieving comparable performance to an NMPC while being real-time capable. Building on [127], the authors of [131] and [132] achieve a significantly improved training speed and sample efficiency by replacing the PPO with an analytic policy gradient approach or combining the PPO with MPC.

For learning-based approaches to become an established form of MCA, these algorithms must see improvements in stability during training and application, as well as in their generalization capabilities. Additionally, so far, no reports of participant studies conducted with learning-based MCAs exist. Hence, no subjective assessment of cueing quality achieved by these MCAs is given.

C. Open-Loop Optimal Control-Based MCAs

In contrast to CL algorithms, OL algorithms do not run in real-time and therefore do not allow for the driver to actively control the system. Whereas most driving simulation experiments require CL control, there are several reasons why OL algorithms have been extensively studied in the literature.

First, especially directly after its introduction [49], MPC could not yet be employed in CL driving due to the too high computational load. Therefore, these algorithms were designed to be applied in a CL context but run OL, as it allowed for testing for the potential benefits of MPC cueing and further development of the algorithms.

Second, even in cases where CL control does run in real-time, OL can still be preferred. Examples are studies that focus on the quality evaluation of the simulator motion, i.e., for MCA research purposes. Here, OL control allows for more invasive rating methods. An example is the continuous rating method of [205], in which subjects rate the motion cueing continuously during a drive by turning a knob, resulting in rating information with a high-temporal resolution. This would not allow for CL driving simultaneously, as one hand needs to continuously control the rotating dial for rating.

Finally, within the context of quality evaluations, OL MCA can also be employed to provide an upper benchmark of the motion cueing quality. As a vehicle reference drive is used to pre-calculate the simulator motion, all future vehicle states are readily known and predicting them is not necessary. The MCAs can then solve a single OCP to compute the simulator control signals $\boldsymbol{u}_{[0:N_{\mathrm{Sim}}]}$ over the whole simulation length with N_{Sim} time steps. To this end, the whole reference vehicle motion $\boldsymbol{y}^{V}_{[0:N_{\mathrm{Sim}}]}$ is used as an input, which is taken from a pre-recorded vehicle drive in the simulation. Therefore, the calculated control signals $\boldsymbol{u}$ make optimal use of the simulator motion space, theoretically giving the best possible motion cueing [97]. Due to its knowledge of all future states, the algorithm is referred to as *Oracle* [74]. The inclusion of a full "prediction" of all future states in an Oracle MCA, compared to an CL MPC *without* reference prediction, has shown to have a large effect on the subjective ratings of test drivers [80].

As the real-time requirement is lifted, also more complex dynamics can be incorporated in the algorithm, such as the full non-linear (redundant) simulator motion system dynamics with actuator constraints and/or the vestibular system [133], [134], [135], [136], [137], [138], [139], [140]. This makes these OL optimization-based MCAs especially suitable for learning-based MCAs to learn from [126]. The drawbacks of these "full" models and the long prediction horizon is that the calculation time can be long (e.g., several hours for a 5 minute drive are reported in [135] and [142]). To mitigate this issue, [140] proposed a Fast Fourier Transform approach to efficiently compute the infinite horizon optimal control signals. A range of comparison studies have indeed shown the significant motion cueing quality increase when compared to filter-based [134], [135], [142] or MPC-based MCAs [80], [101]. Furthermore, within the Oracle context, OL optimization-based algorithms have also led to the observation of (novel) motion cueing strategies, even without explicit instruction to the algorithm to do so. Examples are pre-positioning and velocity buffering [142].

A final use-case for OL methods is where CL driving is not the goal in the first place, such as in self-driving cars or autonomous driving. In that case, OL algorithms do not restrict the experiment in any way. Research in autonomous driving currently considers investigations in driving comfort [206] as well as motion sickness [207] using driving simulators. In both cases, OL MCAs can be employed with the large benefit of providing a higher motion cueing quality and thus increasing the experiment validity.

V. Challenges of Optimization-Based MCAs and Future Work

Clearly, a wide range of optimization-based MCA formulations, ideas and solutions exist. In this section, we focus on four current challenges of optimization-based algorithms in driving simulation. The aim is to provide concrete steps for future work to improve the comparability between MCAs in order to identify and establish best practices, to advance algorithm usability, and to prioritize increased motion cueing fidelity.

These challenges are based on a critical analysis of the literature, as well as practical experience obtained at BMW's driving simulation center. The comparisons are made mostly qualitatively and are based on the presented results of the respective works from literature. Subsection V-A directly discusses the limitations when quantitatively comparing MCAs across literature, building the basis for the following analysis and discussion.

A. Comparing Motion Cueing Algorithms

Evaluating and comparing MCAs is done either regarding their structure, components, and achieved computation times, or with regards to the achieved motion cueing quality. Table I provides such an evaluation and comparison with respect to structure, components, and achieved computation times. A central issue for advancing MCAs in a targeted manner, however, results from the inability to systematically and quantitatively compare different algorithms regarding their cueing performance. This impedes the identification of improvements and best practices, therefore slowing MCA development.

Objective and subjective performance metrics exist to quantify motion cueing quality, which both come with their own (dis)advantages. On one hand, subjective evaluations rely on real simulator hardware, where a sufficiently large number of participants is asked to provide ratings of their (subjective) perception of motion realism. While this is generally the best metric regarding wether the MCA achieved its ultimate goal, i.e., providing motions perceived as realistic by study participants, performing participant studies is time consuming and very costly. This leads to only a small number of works providing a subjective evaluation of their MCA implementation, as can be seen in Tab. I. On the other hand, objective metrics, such as the correlation coefficient (CC) or the root mean square error (RSME) between the vehicle reference and simulator motion, are easy to obtain and are therefore more commonly used to evaluate MCA performance [208]. However, their disadvantage is that better values in objective criteria do not necessarily coincide with improved participant ratings of realism [104], [141], [209], and can thus not always be relied upon.

Another crucial problem regarding the comparison of MCAs is that both subjective and objective metrics do not allow a comparative, quantitative performance analysis of different MCAs across papers. Specifically, there are three major reasons:

1) No universally agreed-upon evaluation framework or standardized metrics exist and metrics such as the CC or RMSE are by far not included in all papers [4]. This means that a performance comparison of different MCAs would only be possible between papers where the same, non-standardized metrics are reported.
2) Different works use different simulators, different vehicle models, different maneuvers and driving tasks, (partly) different MCAs, different parameterizations of the MCAs, and finally different participants (age, experience, gender, etc.). This makes it generally impossible to draw conclusions on the qualitative and quantitative motion cueing quality differences achieved by different MCAs in different works. This restricts comparative analysis of cueing performance between different papers [210].
3) Even comparisons of MCAs within the same paper face challenges. Objective metrics for cueing performance do not necessarily coincide with perceived motion realism, making qualitative statements of the form "MCA 1 is better than MCA 2" difficult. Both objective and subjective metrics do not allow for quantitative statements regarding how much *more* realistic MCA 1 is perceived than MCA 2.

Based on this, our suggestions for future work are twofold. First, regarding MCA performance metrics, advanced objective metrics are necessary. These must ensure that better objective ratings coincide with better subjective participant ratings. This requires further research on human motion perception in the context of driving simulation. Until such metrics exist, authors should also perform and report subjective evaluations, as it is the participants' ratings of realism that ultimately matter. Second, to solve the issue that comparing MCAs across papers is not possible, we suggest the development of a unified benchmark framework. In such a framework, newly developed MCAs are subjected to the same representative benchmark vehicle drive, using the exact same generic model of a simulator, i.e., kinematics and motion capabilities. This would allow to compare the achieved objective cueing quality across different papers. Along with improved objective metrics, this would allow to compare MCAs regarding their ability to achieve *perceived* realism, and, therefore, would allow to accelerate MCA development.

B. Cost Function Design and Tuning

The stability, convergence speed, and ultimately the motion cueing quality depend on and are thus significantly influenced by the cost function design and the weighting parameter tuning. Furthermore, the time consuming process of tuning the weights affects the usability of the algorithms.

1) Cost Function Tuning: The most commonly used form of cost function is that shown in (20), where different works leave out some of the four terms depending on the application. Apart from selecting the terms that constitute the cost function, a key challenge is to systematically determine how these parts should be weighted with respect to each other. The tuning of the weights is a time-consuming and cumbersome task, often performed in a trial-and-error process. When choosing the weights $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{K}$ in (20), the total sum of parameters is $n_y + n_x + 2 \cdot n_u$, with n_y the number of reference tracking weights, n_x the number of state weights, and n_u the number of input weights. Thus, the challenge in the tuning further increases for increasing DoFs.

A common choice for the reference tracking matrix is $\mathbf{Q} = \text{diag}[1\ 1\ 1\ 10\ 10\ 10]$ [54], [103], [104], [137], [142]. The three latter weights for the rotational rates are chosen ten times higher, as this balances the resulting cost values. This is due to the magnitude differences when considering the root-mean square of the motion signals of specific forces (in m/s^2)

and rotational rates (in rad/s) for typical maneuvers [142]. Whereas this approach has been used as a rule of thumb with relative success, there is no direct justification for its use from a human perception standpoint.

The works [143], [205] have investigated deriving the relative importance of the motion channels to subjective ratings. For example, for typical urban driving scenarios the subjective rating is primarily caused by cueing errors in f_y, f_x, and ω_z [143], ordered by decreasing contribution. The relative importance of these channels, combined with the magnitude of the motion signals, may thus be transformed into weighting factors $\mathbf{Q}$. Although an initial experiment reported no direct improvement in subjective ratings when using the adapted weighting compared to the baseline weights, this method may still hold potential due to its close connection to human perception. However, the rating models from [143] and [205] used to determine the relative importance of motion channels appear to be dependent on the chosen scenario. This might indicate that this approach would benefit from separately determined weight factors for each scenario, rather than choosing the same baseline weighting set at all times.

The choice for the state weighting matrix $\mathbf{R}$ contributes to realizing washout behavior and thus results in a trade-off between motion cueing quality and workspace usage [73], [157]. If the prediction is accurate and long enough, no washout is needed and the weights of the state excitation term can all be set to zero [73]. Practically, a non-zero term is still beneficial to increase the convexity of the cost function [45], [157], resulting in a lower computation time. A higher weight $\mathbf{R}$ on the deviations of the simulator to the neutral position increases robustness against prediction errors and differences in drivers [73], but restricts workspace usage.

Besides choosing the weighting parameters, an investigation is necessary on how these weights are best applied. Instead of having fixed weights, in literature, adaptive MPC weights are used to achieve a smoother platform motion near its limits [46], to adapt the washout speed [106], or to optimize for improved cueing quality. To adapt the weights, the authors of [105] and [106] suggest fuzzy logic controllers.

In order to ease the cumbersome tuning process, a variety of papers suggest to apply auto-tuning approaches for weight optimization, i.e., evolutionary optimization techniques. For example, weighting matrices are obtained by a whale or a grey wolf optimization algorithm in [81] and [68], respectively. Similarly, a GA tunes the nonlinear scaling for an MPC MCA in [91] and a fuzzy controller in [105]. By employing a multi-objective GA and, in a second step, an interactive GA derives weights in [107]. Furthermore, a PSO has been shown to yield improved results compared to a GA [108]. In [109], a combination of PSO and GA is used, including human interaction to test the obtained weights. Alternatively, the weighting parameters can be obtained through an auto-tuning approach, where the optimal weights of the reference tracking matrix $\boldsymbol{Q}$ are chosen by minimizing a cost function *before* the actual optimization of the MCA takes place [74].

While a wide range of auto-tuning methodologies exists, none of these methodologies have been subjectively evaluated through a test subject study, however. A key challenge for future work is thus that, even though the tuning is significantly more important than the motion capabilities and size of the simulator [211], no commonly agreed upon tuning framework to target perceptual fidelity exists [4]. As a result, it is difficult to compare methodologies and the current cost function implementations may not be properly tuned or designed towards the human driver. Future work should therefore analyze the change in test driver ratings when employing the different cost function weights, e.g., obtained with auto-tuning approaches. Besides considering stability, computation time, and objective metrics for motion cueing quality when choosing the cost function, also the resulting subjective ratings is paramount.

2) Human-Centric Cost Function Design and Tuning: Many performance evaluations of MCAs in literature do not include any feedback or ratings from test drivers (see Tab. I). Instead, purely objective quality metrics are used [68], [85], [93], such as the root mean square error or the correlation between $\boldsymbol{y}^V$ and $\boldsymbol{y}^S$. While objective metrics correlate with the subjective ratings of test drivers [4], objective metrics have not (yet) fully replaced the subjective evaluation methods. Ultimately, it is the test drivers' subjective evaluation of cueing quality and realism that the MCA should be tuned for. Therefore, another key challenge is that optimization-based MCAs must leap towards incorporating the subjective experience of the human at center of the optimization itself.

As a basis, a variety of works have investigated how different aspects of motion cueing relate to subjective ratings given by test subjects [104], [143], [209]. Findings include that false cues are experienced more problematic than missing cues or cues with a scaling error [209]. So far, these aspects of subjective perception have not been used in the design of an optimization-based MCA, with all formulations currently not distinguishing between error types. Quadratic cost functions, optimizing for the quadratic cueing errors $(\boldsymbol{y}_n^S-\boldsymbol{y}_n^V)^\top\boldsymbol{Q}(\boldsymbol{y}_n^S-\boldsymbol{y}_n^V)$, do not capture the above considerations at all. The same holds for the automatic tuning approaches, that only pay attention to objective metrics.

Future work should therefore develop an alteration to the current cost function design, in which the subjective evaluation of the human test driver is better represented. This may be done by penalizing specific error types, penalizing motion that would be predicted to lead to high ratings (e.g., from [188]), or by adapting the cost function weights, as discussed earlier. Second, current vestibular models are adding complexity to the MCAs without validated benefit in perceived realism [87], [188]. However, model-based optimization approaches depend on the availability of a model quantity that is directly relevant to the control objective. Since the control objective here is perceived motion realism, which existing vestibular system models do not sufficiently capture, it is of utmost importance to develop perceptual models that reflect this objective. Therefore, we suggest adding more accurate and holistic models of human motion perception, including, e.g., learning and adaptation effects or visual-vestibular interaction [212], [213]. As the quality of simulator visualization is not physically limited in the same manner as simulator motion, higher coherence in visual cues creates inconsistencies between visual and motion perception. Understanding the importance of consistency in both timing and magnitude across relevant sensory cues is therefore expected to enable further improvements in motion cueing strategies [212]. Following the more recent survey on mathematical models of the vestibular system done by Momani and Cardullo [213], we suggest replacing the vestibular model by Telban and Cardullo [32] with the semicircular canal model proposed by Grant et al. [214] and the otolith model derived by Rabbitt and Damiano [215], [216], refined in [213]. Although these models are significantly more complex, they provide higher physiological accuracy and are therefore better suited for capturing the relevant perceptual effects. Furthermore, participant studies that quantify the relative contribution of existing objective criteria to perceived motion realism are expected to enable the construction of weighted composite measures that more faithfully reflect perceived motion cueing quality. Such an improved understanding and modeling of human motion perception would also allow the development of objective metrics that provide more information on the actually achieved perceived motion cueing quality. Last, future work in the direction of learning-based MCAs is advised to incorporate subjective rating data into training these MCAs. Training an MCA on subjective rating data enforces motions perceived as realistic by participants, while simultaneously decreasing the need for accurate models of human motion perception.

C. Real-Time Capability

MPC MCAs are computationally expensive. The combination of large system models (usually more than 30 states from the vestibular and the simulator model), desired control rates faster than 100 Hz (sometimes even 200 Hz or more are required), and long prediction horizons makes deploying these algorithms challenging with current computing hardware. Therefore, careful selection of the employed solver and the relevant design parameters of the MCA is needed.

1) Choice of Solver: The choice of solver plays a crucial role for the computation time [184]. In [111], performance of an interior point (IP) solver, namely `HPMPC`, and an active set active set (AS) solver, `qpOASES`, are compared. For the same prediction horizons, cueing quality was the same with both solvers. However, as the complexity of `HPMPC` is $\mathcal{O}(N(n_x + n_u)^3)$, it shows superior to `qpOASES`, whose complexity is scaling as $\mathcal{O}(N^3)$. The solver `HPIPM` [217], i.e., the successor of `HPMPC`, is employed within `acados` [218] by the same authors in their recent work [184]. Furthermore, five solvers were compared in [112], where the IP solver `Knitro-IP` [219] was identified as most suitable, along with a direct collocation scheme. IP solvers are known to be more suitable for larger optimizations, converging in fewer iterations, whereas AS-based solvers can be warm started, i.e., can exploit information from the previous solution. The complexity of `GRAMPC` scales linearly with the problem size as well, making such solvers superior for the desired long prediction horizons.

Future work should look into the application of GPU-based MPC MCA implementations, which are currently introduced in robotics [220]. Additionally, a comparison of solvers for use in MPC MCAs should be performed, paying particular attention to the demands of driving simulation. The potential improvements would allow to use more accurate models of human motion perception, extended prediction horizons and/or less expensive computation hardware, ultimately increasing motion cueing quality and usability.

2) Choice of Prediction Horizon: The prediction horizon significantly influences the computational complexity, motion cueing quality, and feasibility of the optimization [45], [181]. Mainly, a trade-off between real-time capability and motion cueing performance needs to be considered. Different suggestions for the minimum length of prediction horizons can be found in literature:

- Feasibility condition for T_p: $T_p \geq \frac{v_{\max}}{a_{\max}^{\text{break}}} \approx 0.5 - 1.5\,\text{s}$ [62].
- Based on an acceptable motion quality loss compared to an infinite horizon prediction: $2.5 - 3.5\,\text{s}$ [45].
- Computed by different evolutionary algorithms as $\approx 4.5 - 6.5\,\text{s}$ [221], [222], [223].
- At least 8 s in a racing context [56].
- Between 7 and 14 s to stop simulators using decelerations below perception thresholds from maximal speed [69].

As it is the most sophisticated approach, taking into account cueing quality and the driven maneuvers, we detail the approach presented by Katliar et al. [45]. Assuming a perfect prediction, the relationship of the cost value against the length of the prediction horizon is analyzed, giving the relation

$$\frac{c(T_p)}{c_\infty} - 1 = \left(\frac{T_p}{c_1}\right)^{-c_2}, \tag{31}$$

where c_∞ and $c(T_p)$ are the problem's cost with an infinite prediction horizon and a horizon of length T_p, respectively. The parameters c_1 and c_2 depend on the driven scenario: c_1, known as the characteristic time, represents the prediction horizon length required to achieve comparable performance to c_∞, thereby capturing the complexity of the maneuver. The largest reported values in driving simulation are approximately 3.6 s for a roundabout and 3.4 s for urban drives, hence being the most challenging according to [45]. With c_2 being around 2, a roughly quadratic decrease of normalized cost is derived as a rule of thumb. This structured approach based on empirical findings is used in, e.g., [74] to determine a suitable horizon length T_p. It is not clear, however, how well (31) can be generalized, as it is parameterized for a limited set of maneuvers, as well as a specific simulator.

In practice, however, horizon lengths are usually simply chosen as long as possible while still achieving real-time performance. As shown in Tab. I, papers that report prediction horizon lengths $T_p \geq 1$ s either solve the problem at less than 100 Hz [82], [92], [111], [113], set very low control horizons, i.e., $T_c \leq 0.1$ s [46], [65], [67], [77], [114], or use move-blocking techniques [56], [59], [69], [73].

An outstanding performance is reported in the algorithms developed by Lamprecht et al. [47], [97]. In [47], a prediction horizon of 4 seconds is applied along with a 250 Hz sampling frequency for a nonlinear MPC MCA on a 7-DOF simulator. To this end, the authors employ `GRAMPC` [196], which is based on an augmented Lagrangian formulation.

A key challenge for real-time optimization-based MCA is to further increase the prediction and control horizons to fully exploit large simulator workspaces and effectively pre-position the simulator. To do so, a critical analysis must be performed on which simplifications in simulator and human perception modeling can be done at which cost in cueing quality actually perceived by study participants. For example, [112] showed that adding a vestibular system in their NMPC increased computation times by a factor of nearly 5, at limited RMSE improvements. This understanding allows to effectively trade off cueing quality and computational effort. To this end, also the use of model order reduction techniques [224] is suggested with the goal of finding reduced order models at little quality loss, contributing to improved real-time capability.

Last, the recently proposed learning-based MCAs, where a large portion of the computation is moved from online computation to training a NN prior to the simulator study, hold great potential to circumvent real-time limitations of MPC MCA. Here, the focus will be on stable and shortened training processes to allow more widespread use.

D. Vehicle Motion Reference Prediction

One of the strengths of optimization-based MCAs, in contrast to filter-based methods, is the possibility to explicitly include and account for the *future* reference motion, i.e., the specific forces and angular accelerations that must be recreated, rather than only the current states. This information on future reference vehicle motion allows the CL optimization-based MCA to preposition the simulator, more effectively handle motion system boundaries and hence, fully exploit the available workspace [47], [51], [92], [115]. However, the future reference vehicle motion depends on the non-deterministic actions of the driver and is thus *inherently unknown* [47], [51], [116]. To use the driving simulator to its full potential, it is thus necessary to accurately predict the future reference vehicle motion, which is a major challenge.

Including accurate predictions of future vehicle states can greatly leverage the motion cueing quality. Several studies [74], [92], [111] in which the future reference motion was fully known beforehand (which is only possible in OL driving), showed a large increase in the cueing quality, which increases further with longer prediction horizons [74], [144]. Even small improvements in reference prediction quality can already lead to significant motion cueing improvements [73], showing that the inclusion of a proper reference prediction is crucial for accurate motion cueing.

Early MPC-based MCAs approximated the future reference by assuming the reference signals to stay constant over a limited prediction, i.e., $\boldsymbol{y}^V_{[1:N_p-1]} = \boldsymbol{y}^V_0$. This means that no available information from the road geometry, surrounding traffic, traffic rules, the driving task, or the current vehicle state is used. As these information sources are likely to affect the driver's future behavior, this shortcoming is often acceptable only for small prediction horizons [54]. Due to the simplicity in obtaining such predictions, however, even recent MCAs employ constant reference signals [46], [62], [64], [85], [117]. Whereas much work has been done into extending the prediction horizon as much as possible, as described in Subsec. V-C, little work exists on making the actual prediction accurate, despite its enormous potential.

Apart from the additional computational effort required to include a larger prediction horizon, one explanation for the fact that including an actual reference prediction is not often done is that the prediction itself is very difficult to make. Although the future lateral and yaw motion may be extracted from the road geometry, the longitudinal motion is more non-deterministic as to where and how strongly the driver applies the control inputs (accelerator and brake pedals). In literature, the explicit reference prediction for optimization-based MCAs has therefore only obtained a limited focus.

1) Model-Based Reference Prediction Approaches: One possible method to predict the reference motion is through the development of a prediction model for the driver behavior and subsequent vehicle motion. These methods therefore rely on modeling the driver, the vehicle, and the environment.

For example, when considering pre-defined routes on single-lane roads, lateral and yaw dynamics typically have a low mean variance between individual drives [136], [225], as the driver must follow the geometry of the road. This coincides with knowledge from vehicle dynamics modeling, where the lateral acceleration and yaw velocity can be approximated by:

$$a_y = \frac{v_x^2}{R} \quad \text{and} \quad \dot{\psi} = \frac{v_x}{R}, \tag{32}$$

respectively, where R is the curve radius and v_x the longitudinal velocity [226], which is estimated using the present speed limit. For example, the lateral specific force and yaw velocity may be predicted using a prior recorded reference drive, or as an average over a pool of drivers recorded on that route.

Whereas the lateral and yaw behavior typically follow a more pre-defined nature due to the known road geometry, the velocity and, more generally, the longitudinal motion is difficult to predict, as drivers are largely free to decide when and how strong they apply acceleration or deceleration behavior [146], [225]. Hence, a similar approach through known vehicle dynamics is not viable. The difference between the vehicle velocity and the speed limit [227], [228], as well as the current accelerator brake pedal deflections [229], may be used to predict the probability that a driver will accelerate or brake. These methods were developed for filter-based MCAs and provide the likelihood that an acceleration will occur, but do not predict the magnitude of the acceleration, nor acceleration values over time, and can thus not be directly implemented in optimization-based MCAs, where explicit motion signals (e.g., specific forces and rotational rates) over time are required.

Alternatively, it is possible to extract the reference motion through a simplified kinematic car model that tries to follow the center-line of the road chosen as the reference. With a more sophisticated nonlinear single-track model and virtual driver model, [92] achieves higher prediction quality. Additionally, a longitudinal driver input prediction is done by assuming a constant rate of change in gas and brake pedal position. To acknowledge prediction errors, the tracking error weighting matrix $\boldsymbol{Q}$ from (20) is exponentially decreased. In a recent work [47], building up on [115], the driver is assumed to act as an optimal controller on a kinematic single-track model, performing the driving task by minimizing an unknown cost function, which is then learned to predict the reference motion.

Although a wide variety of options for predicting future reference motion exist, there is still a lack of understanding of *which* information on the environment, the vehicle, or the driver provides accurate predictors. It is an open question whether it is possible to develop mathematical models that fully capture human driving behavior. In fact, a more complete multivariate time-series forecasting (including surrounding traffic and traffic rules, information on the driving task and the vehicle motion envelope, or the current vehicle state) has not been developed within the context of a CL optimization-based MCA. Given the high impact that the "perfect" predictions had in OL studies, it can be expected that a complete, multivariate prediction model for CL optimization-based MCAs will yield significant improvements over the state of the art.

Furthermore, we suggest attempting to increase the length of the prediction horizon in which a sufficiently accurate prediction is achieved. The control schemes in [227], [228], and [229] were developed for relatively small motion systems in which a very large prediction horizon had no benefit, because the potential for prepositioning was limited. Larger systems, such as BMW's Sapphire Space, need larger prediction horizons to fully exploit their motion workspace. Predicting further ahead might also reveal differences in the importance of the different *parts* of the prediction. Events further ahead might be less likely to occur or of less importance. For example, specific traffic behavior 10 seconds ahead might be less important to the current-time simulation compared to traffic behavior 5 seconds ahead. Using current methods, these would be weighted equally.

Finally, there will always be uncertainty in the reference motion, which current optimization-based MCAs in driving simulation do not incorporate in their optimization. Further research is suggested to target how to model and include the uncertainties of the future vehicle motion into the optimization-based MCA formulations. To this end, existing stochastic and robust MPC, or adaptive control strategies based on the predicted uncertainty [230] can be introduced in optimization-based MCAs.

2) Machine Learning-Based Reference Prediction: As an alternative to the model-based approaches, machine learning can be employed to predict the future reference signals. For example, in [114], the authors made use of a feedforward NN with one hidden layer to anticipate future longitudinal specific force over a horizon of 2.5 seconds. Only the past longitudinal specific force signals $\boldsymbol{f}_{x,[k-N_{\text{hist}}:k]}$ are used as an input to the NN. A slight increase in prediction quality was reported in [118] by additionally feeding the NN the position of the vehicle on the road. Fuzzy logic is added to the NNs in [119]. Due to the feedforward NN not having a feedback-loop, prediction accuracy is limited, especially for longer prediction horizons [120]. To better deal with time-series data, [96], [120] employed a recurrent neural network (RNN), which performed better than the feed-forward neural networks, but struggled with vanishing and exploding gradients [121]. The same authors proposed a non-linear auto-regressive (NAR) model with 100 time delays, leading to a higher accuracy, where advantages lay on simplicity and fast training [120]. To achieve an increased accuracy of future reference forecasting, [122], [123] apply a long short-term memory (LSTM) model, which can better capture long-term dependencies. It additionally solves the problem of exploding or vanishing gradients and memory limitations occurring in the use of RNNs [123]. Additionally, the authors of [122] propose the use of an echo state network (ESN), that showed superior results in prediction and training time compared to their LSTM approach.

Similar to most model-based approaches, these machine learning approaches implement a univariate time series prediction; i.e., single signals such as the reference longitudinal specific force or yaw rate are predicted from their historical data without consideration of other motion signals or any other knowledge. In driving simulation, the behavior and trajectories of all other agents, the exact states of the simulated vehicle, traffic rules, the complete road and road infrastructure network, as well as the driving task, are deterministic. Additionally, information such as the gas/brake pedal deflection or data from motion and eye trackers inside the mock-up might be available, whereas the earlier discussed model-based approaches (e.g., [227], [228], [229]) have shown that these indeed provide significant contributions to the future reference motion.

An outline for future research is thus the development of *multimodal* machine-learning approaches. An example for this is the Convolutional Long Short-Term Memory Neural Network (CNN-LSTM) [231] employed in [121], where the multivariate many-to-many time series prediction model takes 10 seconds of the recent historical longitudinal and heave specific force data as input, as well as longitudinal velocity data. It outputs the predicted time series values for these three signals over a time horizon of 5 seconds. Its performance exceeds that of the RNN [123] and feedforward NN with time delay [114], as well es implementations of a Random Forest Regression and Lasso Regression presented in [121]. This shows that by taking more signals as inputs to adequate machine learning approaches, prediction quality enhances.

A further limitation of the existing works is that the data used in training the various networks is often limited. Data are often collected by a single driver, driving on the same track or a very limited road network, for a duration ranging 3 – 30 minutes. Hence the previous machine learning approaches cannot be expected to generalize well, effectively only allowing trained networks to be used in the specifically trained maneuvers on the same or a very similar route. Future work is thus suggested to focus on addressing these limitations by leveraging larger and more diverse datasets, either by combining existing data sets or by employing large, readily available data sets. One of the main benefits of machine learning techniques is the possibility to develop a model on large data sets and investigate the incorporation of many predictors without the explicit investigation into a specific one. Combined, these might thus greatly enhance quality and the robustness of machine learning models and leverage the advantages of machine learning methods. By overcoming this second challenge as well, machine learning-based methods can thus provide robust and reliable tools for driver control prediction in simulation environments.

VI. Conclusion

In the last decade, the research focus of motion cueing in driving simulation has shifted from filter-based MCAs towards optimization-based MCAs. The use of optimization-based MCAs is mainly motivated by their ability to handle constraints explicitly, take simulator and vestibular dynamics into account, and incorporate information on future vehicle motion, potentially leading to improved workspace usage and motion cueing quality. Despite these benefits over filter-based approaches, MPC and other optimization-based algorithms have still not become the industry standard. This paper presented an in-depth review of optimization-based MCAs for driving simulation to answer why this is the case. Although its technical foundations have been established over the years, there is a lack of integration of what makes MPC great in theory: 1) a proper, systematic, and human-centered design of the cost function, 2) the ability to optimize for the motion in real-time, 3) the ability to explicitly include an accurate reference prediction, and 4) having proper, standardized evaluation methods. To make MPC the industry standard in driving simulation, these four challenges need to be addressed to truly make MPC stand out over filter-based approaches. By providing this literature overview, identifying these key challenges, and providing guidelines for future research, crucial advancements for optimization-based algorithms can be achieved. We hope that this work therefore spurs the development of the next generation of MCAs, ultimately enhancing the realism and utility of driving simulators across research and industry.

Appendix

Here, we shortly present MCAs that are developed for specific use-cases or present further interesting ideas and did not fit the main part.

Hvitfeldt et al. provide an adapted filter-based MCA [44] and an optimization-based MCA [102] for winter conditions. The authors in [200] propose a framework for switching between different types of MCAs in order to make the best use of their respective advantages. Schwarzhuber [232] focused on MCAs in motorsport applications, i.e., replication of motion perception in near-limit vehicle handling. In [72] the MPC cost function is extended by a term aiming to replicate the frequency content of the reference signal. Last, a framework to improve tracking errors or modeled subjective ratings during worst-case maneuvers is shown in [141].

References

[1] J. Lindner et al., "A coupled driving simulator to investigate the interaction between bicycles and automated vehicles," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 1335–1341.

[2] C. Chai, X. Zeng, I. Alvarez, and M. S. Elli, "Evaluation of responsibility-sensitive safety (RSS) model based on human-in-the-loop driving simulation," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–7.

[3] P. Bazilinskyy, L. Kooijman, D. Dodou, and J. D. Winter, "Coupled simulator for research on the interaction between pedestrians and (automated) vehicles," in *Proc. Driving Simulation Conf. Eur.*, 2020, pp. 205–206.

[4] S. Casas-Yrurzum, C. Portalés-Ricart, P. Morillo-Tena, and C. Cruz-Neira, "On the objective evaluation of motion cueing in vehicle simulations," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 3001–3013, May 2021.

[5] C. Himmels, J. Venrooij, M. Gmünder, and A. Riener, "The influence of simulator and driving scenario on simulator sickness," in *Proc. Driving Simulation Conf.*, 2022, pp. 29–36.

[6] B. J. C. Grácio, M. Wentink, and A. R. Valente Pais, "Driver behavior comparison between static and dynamic simulation for advanced driving maneuvers," *Presence Teleoperators Virtual Environ.*, vol. 20, no. 2, pp. 143–161, Apr. 2011.

[7] R. Romano et al., "An objective assessment of the utility of a driving simulator for low μtesting," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 65, pp. 34–45, Aug. 2019.

[8] K. Stahl, G. Abdulsamad, K.-D. Leimbach, and Y. A. Vershinin, "State of the art and simulation of motion cueing algorithms for a six degree of freedom driving simulator," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 537–541.

[9] B. Conrad, S. Schmidt, and J. Douvillier, "Washout circuit design for multi-degrees-of-freedom moving base simulators," in *Proc. Vis. Motion Simulation Conf.*, Sep. 1973, pp. 929–937.

[10] S. F. Schmidt and B. Conrad, "Motion drive signals for piloted flight simulators," National Aeronautics and Space Administration, Washington, DC, USA, Tech. Rep. NASA CR-1601, 1970.

[11] L. D. Reid and M. A. Nahon, "Flight simulation motion-base drive algorithms part 1. Developing and testing equations," Univ. of Toronto, Toronto, ON, Canada, Tech. Rep. 296, 1985.

[12] M. E. Bitner, "Investigation of motion base drive techniques," Naval Air Development Center, Warminster, PA, USA, Tech. Rep. SP-7037 (87), Mar. 1978.

[13] R. Sivan, J. Ish-Shalom, and J.-K. Huang, "An optimal control approach to the design of moving flight simulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-12, no. 6, pp. 818–827, Jun. 1982.

[14] R. V. Parrish, J. E. Dieudonne, R. L. Bowles, and D. J. Martin, "Coordinated adaptive washout for motion simulators," *J. Aircr.*, vol. 12, no. 1, pp. 44–50, Jan. 1975.

[15] D. Ariel and R. Sivan, "False cue reduction in moving flight simulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, no. 4, pp. 665–671, Jul. 1984.

[16] A. R. Naseri and P. R. Grant, "An improved adaptive motion drive algorithm," in *Proc. AIAA Modeling Simulation Technol. Conf. Exhib.*, 2005, pp. 1–9.

[17] M. A. Nahon, L. D. Reid, and J. Kirdeikis, "Adaptive simulator motion software with supervisory control," *J. Guid., Control, Dyn.*, vol. 15, no. 2, pp. 376–383, Mar. 1992.

[18] H. Arioui, L. Nehaoua, and H. Amouri, "Classic and adaptive washout comparison for a low cost driving simulator," in *Proc. IEEE Int. Symp. Mediterrean Conf. Control Autom. Intell. Control*, 2005, pp. 586–591.

[19] B. Aykent, D. Paillot, F. Merienne, Z. Fang, and A. Kemeny, "Study of the influence of different washout algorithms on simulator sickness for a driving simulation task," in *Proc. World Conf. Innov. Virtual Reality*, 2011, pp. 331–341.

[20] P. R. Grant and L. D. Reid, "Motion washout filter tuning: Rules and requirements," *J. Aircr.*, vol. 34, no. 2, pp. 145–151, Mar. 1997.

[21] H. Asadi, S. Mohamed, and S. Nahavandi, "Incorporating human perception with the motion washout filter using fuzzy logic control," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 6, pp. 3276–3284, Dec. 2015.

[22] H. Asadi, C. P. Lim, S. Mohamed, D. Nahavandi, and S. Nahavandi, "Increasing motion fidelity in driving simulators using a fuzzy-based washout filter," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 2, pp. 298–308, Jun. 2019.

[23] M. R. Chalak Qazani, H. Asadi, T. Bellmann, S. Perdrammehr, S. Mohamed, and S. Nahavandi, "A new fuzzy logic based adaptive motion cueing algorithm using parallel simulation-based motion platform," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2020, pp. 1–8.

[24] M. R. C. Qazani, H. Asadi, M. Rostami, S. Mohamed, C. P. Lim, and S. Nahavandi, "Adaptive motion cueing algorithm based on fuzzy logic using online dexterity and direction monitoring," *IEEE Syst. J.*, vol. 16, no. 2, pp. 1945–1953, Jun. 2022.

[25] H. Asadi et al., "A particle swarm optimization-based washout filter for improving simulator motion fidelity," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 001963–001968.

[26] C. G. Arango, H. Asadi, M. R. C. Qazani, S. Mohamed, and S. Nahavandi, "Automatic tuning of adaptive gradient descent based motion cueing algorithm using particle swarm optimisation," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2022, pp. 85–92.

[27] S. Casas, I. Coma, C. Portalés, and M. Fernández, "Towards a simulation-based tuning of motion cueing algorithms," *Simulation Model. Pract. Theory*, vol. 67, pp. 137–154, Sep. 2016.

[28] M. R. Chalak Qazani, H. Asadi, T. Bellmann, S. Mohamed, C. P. Lim, and S. Nahavandi, "Adaptive washout filter based on fuzzy logic for a motion simulation platform with consideration of joints' limitations," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12547–12558, Nov. 2020.

[29] C.-I. Huang and L.-C. Fu, "Human vestibular based (HVB) senseless maneuver optimal washout filter design for VR-based motion simulator," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2006, pp. 4451–4458.

[30] S.-H. Chen and L.-C. Fu, "An optimal washout filter design with fuzzy compensation for a motion platform," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 8433–8438, Jan. 2011.

[31] L. Nehaoua, H. Mohellebi, A. Amouri, H. Arioui, S. Espie, and A. Kheddar, "Design and control of a small-clearance driving simulator," *IEEE Trans. Veh. Technol.*, vol. 57, no. 2, pp. 736–746, Mar. 2008.

[32] R. J. Telban, F. M. Cardullo, and L. C. Kelly, "Motion cueing algorithm development: New motion cueing program implementation and tuning," State Univ. of New York, Albany, NY, USA, Tech. Rep. NASA/CR-2005-213746, 2005.

[33] W. Wu, F. Cardullo, W. Wu, and F. Cardullo, "Is there an optimum motion cueing algorithm?," in *Proc. Modeling Simulation Technol. Conf.*, Aug. 1997, pp. 23–29.

[34] R. J. Telban and F. M. Cardullo, "Motion cueing algorithm development: Human-centered linear and nonlinear approaches," State Univ. of New York, Albany, NY, USA, Tech. Rep. NASA/CR-2005-213747, 2005.

[35] R. Telban, F. Cardullo, and J. Houck, "Developments in human centered cueing algorithms for control of flight simulator motion systems," in *Proc. Modeling Simulation Technol. Conf. Exhibit*, Aug. 1999, pp. 1–11.

[36] R. J. Telban, "A nonlinear motion cueing algorithm with a human perception model," Ph.D. dissertation, Dept. Mech. Eng., Binghamton University, New York, NY, USA, 2001.

[37] M. Mayrhofer, B. Langwallner, R. Schlüsselberger, W. Bles, and M. Wentink, "An innovative optimal control approach for the next generation simulator motion platform DESDEMONA," in *Proc. AIAA Modeling Simulation Technol. Conf. Exhibit*, Aug. 2007, pp. 1–14.

[38] M. Aminzadeh, A. Mahmoodi, and M. Sabzehparvar, "Optimal motion-cueing algorithm using motion system kinematics," *Eur. J. Control*, vol. 18, no. 4, pp. 363–375, Jan. 2012.

[39] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "An optimal motion cueing algorithm using the inverse kinematic solution of the hexapod simulation platform," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 1, pp. 73–82, Mar. 2022.

[40] H. Asadi, T. Bellmann, S. Mohamed, C. P. Lim, A. Khosravi, and S. Nahavandi, "Adaptive motion cueing algorithm using optimized fuzzy control system for motion simulators," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 390–403, Jan. 2023.

[41] M. R. C. Qazani, H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "An optimal washout filter for motion platform using neural network and fuzzy logic," *Eng. Appl. Artif. Intell.*, vol. 108, Feb. 2022, Art. no. 104564.

[42] H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 238–254, Feb. 2017.

[43] H. Asadi, C. P. Lim, A. Mohammadi, S. Mohamed, S. Nahavandi, and L. Shanmugam, "A genetic algorithm–based nonlinear scaling method for optimal motion cueing algorithm in driving simulator," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 232, no. 8, pp. 1025–1038, Aug. 2018,doi: 10.1177/0959651818772940.

[44] H. Hvitfeldt, L. Drugge, and J. Jerrelind, "Motion cueing for winter test conditions," in *Advances in Dynamics of Vehicles on Roads and Tracks II*, A. Orlova and D. Cole, Eds., Cham, Switzerland: Springer, 2022, pp. 888–901.

[45] M. Katliar, K. D. Winkel, J. Venrooij, P. Pretto, and H. Bülthoff, "Impact of MPC prediction horizon on motion cueing fidelity," in *Proc. Driving Simulation Conf.*, 2015, pp. 219–222.

[46] 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, no. 4, pp. 625–647, Dec. 2020.

[47] A. Lamprecht, D. Steffen, K. Nagel, J. Haecker, and K. Graichen, "Online model predictive motion cueing with real-time driver prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12414–12428, Aug. 2022.

[48] 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 Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 10, pp. 6096–6110, Oct. 2021.

[49] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïki, "MPC based motion cueing algorithm: Development and application to the ULTIMATE driving simulator," in *Proc. Driving Simulation Conf. Eur.*, Sep. 2004, pp. 221–233.

[50] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, "Model-based predictive motion cueing strategy for vehicle driving simulators," *Control Eng. Pract.*, vol. 17, no. 9, pp. 995–1003, Sep. 2009.

[51] A. Beghi, M. Bruschetta, and F. Maran, "A real time implementation of MPC based motion cueing strategy for driving simulators," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 6340–6345.

[52] M. Baseggio, A. Beghi, M. Bruschetta, F. Maran, and D. Minen, "An MPC approach to the design of motion cueing algorithms for driving simulators," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 692–697.

[53] B. D. C. Augusto and R. J. L. Loureiro, "Motion cueing in the Chalmers driving SImulator: A model predictive control approach," Master's thesis, Dept. Signals Syst., Chalmers Univ. of Technol., Gothenburg, Sweden, 2009.

[54] F. Ellensohn, "Urban motion cueing algorithms," Ph.D. dissertation, Tech. Univ. Munich, Munich, Germany, 2019.

[55] I. A. Qaisi and A. Traechtler, "Constrained linear quadratic optimal controller for motion control of ATMOS driving simulator," in *Proc. Driving Simulation Conf.*, Paris, France, 2012, pp. 1–8.

[56] A. Beghi, M. Bruschetta, and F. Maran, "A real-time implementation of an MPC-based motion cueing strategy with time-varying prediction," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 4149–4154.

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

[58] Z. Fang and A. Kemeny, "An efficient MPC based motion cueing delay compensation algorithm for driving simulator," in *Proc. Driving Simulation Conf.*, 2015, pp. 163–172.

[59] F. Maran, M. Bruschetta, A. Beghi, and D. Minen, "Improvement of an MPC-based motion cueing algorithm with time-varying prediction and driver behaviour estimation," in *Proc. Driving Simulation Conf.*, 2015, pp. 183–190.

[60] Z. Fang, M. Tsushima, E. Kitahara, N. Machida, D. Wautier, and A. Kemeny, "Motion cueing algorithm for high performance driving simulator using yaw table," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15965–15970, Jul. 2017.

[61] Z. Fang and A. Kemeny, "An efficient model predictive control-based motion cueing algorithm for the driving simulator," *Simulation*, vol. 92, no. 11, pp. 1025–1033, Nov. 2016.

[62] M. Bruschetta, F. Maran, and A. Beghi, "A fast implementation of MPC-based motion cueing algorithms for mid-size road vehicle motion simulators," *Vehicle Syst. Dyn.*, vol. 55, no. 6, pp. 802–826, Jun. 2017.

[63] S. Munir, M. Hovd, Z. Fang, S. Olaru, and A. Kemeny, "Complexity reduction in motion cueing algorithm for the ULTIMATE driving simulator," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10729–10734, Jul. 2017.

[64] M. Soyer, S. Olaru, and Z. Fang, "MPC delay compensation based on maximal controllable sets for real-time driving simulators," in *Proc. 25th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2021, pp. 649–654.

[65] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A new Gantry-Tau-based mechanism using spherical wrist and model predictive control-based motion cueing algorithm," *Robotica*, vol. 38, no. 8, pp. 1359–1380, Aug. 2020.

[66] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "High-fidelity hexarot simulation-based motion platform using fuzzy incremental controller and model predictive control-based motion cueing algorithm," *IEEE Syst. J.*, vol. 14, no. 4, pp. 5073–5083, Dec. 2020.

[67] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A model predictive control-based motion cueing algorithm with consideration of Joints' limitations for hexapod motion platform," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 708–713.

[68] M. R. C. Qazani et al., "An MPC-based motion cueing algorithm using washout speed and grey wolf optimizer," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 1627–1633.

[69] Z. Fang, M. Tsushima, E. Kitahara, N. Machida, D. Wautier, and A. Kemeny, "Fast MPC based motion cueing algorithm for a 9DOF driving simulator with yaw table," in *Proc. Driving Simulation Conf. Eur.*, 2017, pp. 73–80.

[70] M. R. Chalak 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 *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 35–41.

[71] H. Asadi, T. Bellmann, M. C. Qazani, S. Mohamed, C. P. Lim, and S. Nahavandi, "A novel decoupled model predictive control-based motion cueing algorithm for driving simulators," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7024–7034, Jun. 2023.

[72] M. Kolff, R. Jacumet, S. Wagner, D. Wollherr, and M. Leibold, "A frequency-aware model predictive control motion cueing algorithm," in *Proc. Driving Simulation Conf. Eur. XR*, 2025, pp. 131–138.

[73] F. M. Drop, M. Olivari, M. Katliar, and H. Bülthoff, "Model predictive motion cueing: Online prediction and washout tuning," in *Proc. Driving Simulation Conf. Eur. VR*, Antibes, France, 2018, pp. 71–78.

[74] 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, no. 8, pp. 707–724, Aug. 2019.

[75] F. Maran, "Model-based control techniques for automotive applications," Ph.D. dissertation, Dept. Inf. Eng., Univ. of Padova, Padua, Italy, 2013.

[76] M. Bruschetta, F. Maran, and A. Beghi, "A non-linear MPC based motion cueing implementation for a 9 DOFs dynamic simulator platform," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 2517–2522.

[77] C. Rengifo, J.-R. Chardonnet, D. Paillot, H. Mohellebi, and A. Kemeny, "Solving the constrained problem in model predictive control based motion cueing algorithm with a neural network approach," in *Proc. Driving Simulation Conf. Eur.*, Antibes, France, 2018, pp. 63–69.

[78] M. Bruschetta, C. Cenedese, and A. Beghi, "A real-time, MPC-based motion cueing algorithm with look-ahead and driver characterization," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 61, pp. 38–52, Feb. 2019.

[79] V. Jain, A. Lazcano, R. Happee, and B. Shyrokau, "Motion cueing algorithm for effective motion perception: A frequency-splitting MPC approach," in *Proc. Driving Simulation Conf. Eur.*, Antibes, France, 2023, pp. 111–118.

[80] F. Ellensohn, J. Venrooij, M. Schwienbacher, and D. Rixen, "Experimental evaluation of an optimization-based motion cueing algorithm," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 62, pp. 115–125, Apr. 2019.

[81] M. Reza Chalak Qazani et al., "Whale optimization algorithm for weight tuning of a model predictive control-based motion cueing algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 1042–1048.

[82] M. Katliar, F. M. Drop, H. Teufell, M. Diehl, and H. H. Bulthoff, "Real-time nonlinear model predictive control of a motion simulator based on a 8-DOF serial robot," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 1529–1535.

[83] Z. Fang and A. Kemeny, "Explicit MPC motion cueing algorithm for real-time driving simulator," in *Proc. 7th Int. Power Electron. Motion Control Conf.*, Jun. 2012, pp. 874–878.

[84] Z. Fang and A. Kemeny, "Motion cueing algorithms for a real-time automobile driving simulator," in *Proc. Driving Simulation Conf.*, Paris, France, 2012, pp. 1–12.

[85] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A motion cueing algorithm based on model predictive control using terminal conditions in urban driving scenario," *IEEE Syst. J.*, vol. 15, no. 1, pp. 445–453, Mar. 2021.

[86] Z. Fang and A. Kemeny, "Review and prospects of Renault's MPC based motion cueing algorithm for driving simulator," in *Proc. Driving Simulation Conf.*, 2014, pp. 231–239.

[87] C. Rengifo, J.-R. Chardonnet, H. Mohellebi, and A. Kemeny, "Impact of human-centered vestibular system model for motion control in a driving simulator," *IEEE Trans. Hum.-Mach. Syst.*, vol. 51, no. 5, pp. 411–420, Oct. 2021.

[88] M. Soyer, S. Olaru, and Z. Fang, "A novel motion cueing algorithm based on real-time optimization and periodic invariant sets," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2020, pp. 19–24.

[89] M. Soyer, S. Olaru, K. Ampountolas, S. Scialanga, and Z. Fang, "Periodic set invariance as a tool for constrained reference tracking," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6905–6910, 2020.

[90] M. Soyer, S. Olaru, and Z. Fang, "Motion cueing control design based on a nonlinear mpc algorithm," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 341–346, 2021.

[91] H. Asadi et al., "A model predictive control-based motion cueing algorithm using an optimized nonlinear scaling for driving simulators," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1245–1250.

[92] P. Biemelt, C. Link, S. Gausemeier, and A. Trächtler, "A model-based online reference prediction strategy for model predictive motion cueing algorithms," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6082–6088, 2020.

[93] F. Ellensohn, F. Oberleitner, M. Schwienbacher, J. Venrooij, and D. Rixen, "Actuator- based optimization motion cueing algorithm," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2018, pp. 1021–1026.

[94] F. Ellensohn, M. Schwienbacher, J. Venrooij, and D. J. Rixen, "Motion cueing algorithm for a 9 DoF driving simulator: MPC with linearized actuator constraints," *SAE Int. J. Connected Autom. Veh.*, vol. 1, pp. 145–155, 2018.

[95] M. Bruschetta, F. Maran, and A. Beghi, "A nonlinear, MPC-based motion cueing algorithm for a high-performance, nine-DOF dynamic simulator platform," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 686–694, Mar. 2017.

[96] M. R. Chalak Qazani et al., "A new prepositioning technique of a motion simulator platform using nonlinear model predictive control and recurrent neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23268–23277, Dec. 2022.

[97] A. Lamprecht, J. Haecker, and K. Graichen, "Constrained motion cueing for driving simulators using a real-time nonlinear MPC scheme," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7466–7471.

[98] A. Chadha, V. Jain, A. M. R. Lazcano, and B. Shyrokau, "Computationally-efficient motion cueing algorithm via model predictive control," in *Proc. IEEE Int. Conf. Mechatronics (ICM)*, Mar. 2023, pp. 1–6.

[99] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "MPC-based motion cueing algorithm with short prediction horizon using exponential weighting," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 000521–000526.

[100] A. Seefried, "Modellprädiktive Bahnplanung für roboterbasierte Bewegungssimulatoren," Ph.D. dissertation, Universität der Bundeswehr München, München, Germany, 2021.

[101] F. Ellensohn, D. Hristakiev, M. Schwienbacher, J. Venrooij, and D. Rixen, "Evaluation of an optimization based motion cueing algorithm suitable for online application," in *Proc. Driving Simulation Conf. Eur.*, 2019, pp. 93–100.

[102] H. Hvitfeldt, L. Drugge, and J. Jerrelind, "Driver gaze model for motion cueing yaw feedback optimisation," in *Advances in Dynamics of Vehicles on Roads and Tracks III*. Cham, Switzerland: Springer, 2024, pp. 217–230.

[103] J. R. V. D. Ploeg, D. Cleij, D. M. Pool, M. Mulder, and H. Bülthoff, "Sensitivity analysis of an MPC-based motion cueing algorithm for a curve driving scenario," in *Proc. Driving Simulation Conf. Eur. VR*, Antibes, France, 2020, pp. 37–44.

[104] D. Cleij, D. M. Pool, M. Mulder, and H. Bülthoff, "Optimizing an optimization-based MCA using perceived motion incongruence models," in *Proc. Driving Simulation Conf. Eur.*, Antibes, France, 2020, pp. 53–60.

[105] M. R. Chalak Qazani, H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "A time-varying weight MPC-based motion cueing algorithm for motion simulation platform," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11767–11778, Aug. 2022.

[106] M. R. Chalak Qazani, H. Asadi, C. Arango, M. S. Hossain, and S. Nahavandi, "Adaptive washout speed model predictive control motion cueing using fuzzy logic system," in *Proc. Australas. Conf. Robot. Autom. (ACRA)*, Dec. 2022, p. 10.

[107] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Multiobjective and interactive genetic algorithms for weight tuning of a model predictive control-based motion cueing algorithm," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3471–3481, Sep. 2019.

[108] M. R. C. Qazani et al., "Weight tuning of a model predictive control motion cueing using a particle swarm optimization algorithm," in *Proc. Australas. Conf. Robot. Autom. (ACRA)*, Dec. 2022, p. 9.

[109] M. R. Chalak Qazani et al., "An optimal nonlinear model predictive control- based motion cueing algorithm using cascade optimization and human interaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9191–9202, Sep. 2023.

[110] S. Al-Serri et al., "Optimising horizons in model predictive control for motion cueing algorithms using reinforcement learning," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2024, pp. 2793–2800.

[111] 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, no. 1, pp. 9833–9839, Jul. 2017.

[112] C. G. Arango and H. Asadi, "Model predictive control motion cueing with nonlinear constraints and vestibular feedback for serial robot motion simulators," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2024, pp. 1–8.

[113] P. Biemelt and S. Henning, "A model predictive motion cueing strategy for a 5-degree-of-freedom driving simulator with hybrid kinematics," in *Proc. Driving Simulation Conf. Eur.*, 2018, pp. 79–85.

[114] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Future reference prediction in model predictive control based driving simulators," in *Proc. Australas. Conf. Robot. Automat. (ACRA)*, 2016, pp. 1–8.

[115] A. Lamprecht, D. Steffen, J. Haecker, and K. Graichen, "Optimal control based reference generation for model predictive motion cueing algorithms," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2019, pp. 203–208.

[116] F. Maran, M. Bruschetta, and A. Beghi, "Study of a real-time, MPC based motion cueing procedure with time-varying prediction for different classes of drivers," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 1711–1716.

[117] H. Hvitfeldt, J. Jerrelind, and L. Drugge, "False cue reduction in motion cueing utilising the vehicle performance envelope," in *Proc. Driving Simulation Conf. Eur. VR*, Angers, France, Sep. 2023, pp. 133–140.

[118] M. R. C. Qazani, H. Asadi, S. Mohamed, and S. Nahavandi, "Prepositioning of a land vehicle simulation-based motion platform using fuzzy logic and neural network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 10446–10456, Oct. 2020.

[119] M. R. Chalak Qazani, H. Asadi, M. Al-Ashmori, S. Mohamed, C. P. Lim, and S. Nahavandi, "Time series prediction of driving motion scenarios using fuzzy neural networks: Motion signal prediction using FNNs," in *Proc. IEEE Int. Conf. Mechatronics (ICM)*, Mar. 2021, pp. 1–6.

[120] M. R. C. Qazani, H. Asadi, C. P. Lim, S. Mohamed, and S. Nahavandi, "Prediction of motion simulator signals using time-series neural networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 3383–3392, Oct. 2021.

[121] S. Alsanwy, H. Asadi, M. R. C. Qazani, S. Mohamed, and S. Nahavandi, "Prediction of vehicle motion signals for motion simulators using a multivariate CNN-LSTM model," in *Proc. Driving Simulation Conf. Eur. VR*, Sep. 2023, pp. 151–158.

[122] M. R. C. Qazani et al., "A prediction of time series driving motion scenarios using LSTM and ESN," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2022, pp. 1592–1599.

[123] S. Alsanwy et al., "Prediction of vehicle motion signals for motion simulators using long short-term memory networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2022, pp. 34–39.

[124] M. R. Chalak Qazani et al., "A neural network-based motion cueing algorithm using the classical washout filter for comprehensive driving scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 5112–5121, Jun. 2024.

[125] M. R. C. Qazani et al., "High-fidelity learning-based motion cueing algorithm by bypassing worst-case scenario-based tuning technique," *Cognit. Robot.*, vol. 4, pp. 116–127, Jan. 2024.

[126] A. B. Koyuncu, E. Ercelik, E. Comulada-Simpson, J. Venrooij, M. Kaboli, and A. Knoll, "A novel approach to neural network-based motion cueing algorithm for a driving simulator," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 2118–2125.

[127] H. Scheidel, H. Asadi, T. Bellmann, A. Seefried, S. Mohamed, and S. Nahavandi, "A deep reinforcement learning based motion cueing algorithm for vehicle driving simulation," *IEEE Trans. Veh. Technol.*, vol. 73, no. 7, pp. 9696–9705, Jul. 2024.

[128] H. Scheidel, A. Seefried, M. R. C. Qazani, T. Bellmann, S. Nahavandi, and H. Asadi, "Deep reinforcement learning based motion cueing algorithm for universal trajectories," in *Proc. Driving Simulation Conf. Eur. VR*, Sep. 2023, pp. 181–186.

[129] H. Scheidel et al., "A nonlinear real time capable motion cueing algorithm based on deep reinforcement learning," 2025, *arXiv:2503.10419*.

[130] C. Gonzalez Arango, H. Asadi, M. Reza Chalak Qazani, and C. Peng Lim, "Learning-based approximate nonlinear model predictive control motion cueing," 2025, *arXiv:2504.00469*.

[131] X. Huang, X. Shi, P. Wang, H. Xu, X. Tang, and G. Zhang, "An analytic policy gradient-based deep reinforcement learning motion cueing algorithm for driving simulators," *IEEE Access*, vol. 13, pp. 81507–81523, 2025.

[132] X. Huang, X. Shi, P. Wang, H. Xu, X. Tang, and G. Zhang, "MPC-augmented reinforcement learning for motion cueing algorithms in driving simulators," in *Advanced Intelligent Computing Technology and Applications*, D.-S. Huang, C. Zhang, Q. Zhang, and Y. Pan, Eds., Singapore: Springer, 2025, pp. 499–510.

[133] I. Salisbury and D. J. N. Limebeer, "Optimal race car motion cueing," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 7547–7552, 2014.

[134] P. Pretto, J. Venrooij, A. Nesti, and H. H. Bülthoff, "Perception-based motion cueing: A cybernetics approach to motion simulation," in *Recent Progress in Brain and Cognitive Engineering*, vol. 5. Cham, Switzerland: Springer, 2015.

[135] J. Venrooij et al., "Comparison between filter- and optimization-based motion cueing in the daimler driving simulator," in *Proc. Driving Simulation Conf. Eur.*, Paris, France, 2016, pp. 31–38.

[136] F. Ellensohn, M. Spannagl, S. Agabekov, J. Venrooij, M. Schwienbacher, and D. Rixen, "A hybrid motion cueing algorithm," *Control Eng. Pract.*, vol. 97, Apr. 2020, Art. no. 104342.

[137] 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," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 66, pp. 29–46, Oct. 2019.

[138] Z. Fang, D. Theillier, F. Colombet, D. Wautier, and S. Régnier, "Study of motion cueing algorithm based on multi-shooting method for ROADS replaying scenario," in *Proc. Driving Simulation Conf. Eur. VR*, 2023, pp. 127–132.

[139] J. Venrooij et al., "Perception-based motion cueing: Validation in driving simulation," in *Proc. Driving Simulation Conf. Eur.*, 2015, pp. 153–161.

[140] Z. Fang, D. Wautier, and A. Kemeny, "FFT based optimal MCA for AD/ADAS driving tests," in *Proc. Driving Simulation Conf. Eur.*, 2022, pp. 119–126.

[141] R. Jacumet, J. Wetzel, S. Wagner, M. Schwienbacher, D. Wollherr, and M. Leibold, "Optimization-based motion cueing algorithm for worst case maneuvers," in *Proc. Driving Simulation Conf. Europe VR*, 2025, pp. 163–168.

[142] D. Cleij et al., "Comparison between filter- and optimization-based motion cueing algorithms for driving simulation," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 61, pp. 53–68, Feb. 2019.

[143] M. Kolff et al., "Effect of motion mismatches on ratings of motion incongruence and simulator sickness in urban driving simulations," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 115, Nov. 2025, Art. no. 103370.

[144] A. Lamprecht, D. Steffen, J. Haecker, and K. Graichen, "Comparison between a filter- and an MPC-based MCA in an offline simulator study," in *Proc. Driving Simulation Conf. Eur.*, 2019, pp. 101–108.

[145] P. Biemelt, S. Mertin, N. Rüddenklau, S. Gausemeier, and A. Trächtler, "Objective evaluation of a novel filter-based motion cueing algorithm in comparison to optimization-based control in interactive driving simulation," in *Proc. Int. Conf. Adv. Syst. Simulation (SIMUL)*, 2019, pp. 25–31.

[146] P. Biemelt, S. Böhm, S. Gausemeier, and A. Trächtler, "Subjective evaluation of filter- and optimization-based motion cueing algorithms for a hybrid kinematics driving simulator," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 1619–1626.

[147] N. Mohajer, H. Abdi, K. Nelson, and S. Nahavandi, "Vehicle motion simulators, a key step towards road vehicle dynamics improvement," *Vehicle Syst. Dyn.*, vol. 53, no. 8, pp. 1204–1226, Aug. 2015.

[148] L. Bruck, B. Haycock, and A. Emadi, "A review of driving simulation technology and applications," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 1–16, 2021.

[149] S. Casas, R. Olanda, and N. Dey, "Motion cueing algorithms: A review: Algorithms, evaluation and tuning," *Int. J. Virtual Augmented Reality*, vol. 1, no. 1, pp. 90–106, Jan. 2017.

[150] M. Fischer, "A survey of state-of-the-art motion platform technology and motion cueing algorithms," in *Proc. 2nd Motion Simulation Conf.*, Braunschweig, Germany, 2007, pp. 1–5.

[151] N. J. I. Garrett and M. C. Best, "Driving simulator motion cueing algorithms—A survey of the state of the art," in *Proc. 10th Int. Symp. Adv. Veh. Cont.*, 2010, pp. 183–188.

[152] M. Ghafarian, M. Watson, N. Mohajer, D. Nahavandi, P. M. Kebria, and S. Mohamed, "A review of dynamic vehicular motion simulators: Systems and algorithms," *IEEE Access*, vol. 11, pp. 36331–36348, 2023.

[153] J. V. Riera, S. Casas, F. Alonso, and M. Fernández, "A case study on vestibular sensations in driving simulators," *Sensors*, vol. 22, no. 15, p. 5837, Aug. 2022.

[154] D. Stewart, "A platform with six degrees of freedom," *Proc. Inst. Mech. Eng.*, vol. 180, no. 1, pp. 371–386, 1965.

[155] V. E. Gough, "Universal tyre test machine," in *Proc. FISITA 9th Int. Tech. Congr.*, 1962, pp. 117–137.

[156] G. Baumann, W. Krantz, and J. Pitz, "Evaluation of steering feel and vehicle handling in the stuttgart driving simulator," in *Proc. 5th Int. Munich Chassis Symp.*, 2014, pp. 201–215.

[157] M. Kolff, J. Venrooij, M. Schwienbacher, D. M. Pool, and M. Mulder, "Motion cueing quality comparison of driving simulators using Oracle motion cueing," in *Proc. Driving Simulation Conf. Eur.*, 2022, pp. 111–118.

[158] Q. H. N. Van and S. Ito, "On the implementation of steering controls for evaluation of automated driving systems in driving simulator," in *Proc. Driving Simulation Conf. Eur.*, 2016, pp. 165–168.

[159] M. Ottensmeier and G. Prokop, "Development of a motion control for a highly dynamic, self-propelled driving simulator," *Automot. Engine Technol.*, vol. 8, no. 1, pp. 17–42, Mar. 2023.

[160] M. Kolff, J. Venrooij, M. Schwienbacher, D. M. Pool, and M. Mulder, "The importance of kinematic configurations for motion control of driving simulators," in *Proc. IEEE 26th Int. Conf. Intell. Transp. Syst. (ITSC)*, Bilbao, Spain, Sep. 2023, pp. 1000–1006.

[161] D. C. Fitze, F. W. Mast, and M. Ertl, "Human vestibular perceptual thresholds—A systematic review of passive motion perception," *Gait Posture*, vol. 107, pp. 83–95, Jan. 2024.

[162] R. M. Mallery, O. U. Olomu, R. M. Uchanski, V. A. Militchin, and T. E. Hullar, "Human discrimination of rotational velocities," *Exp. Brain Res.*, vol. 204, no. 1, pp. 11–20, Jul. 2010.

[163] Z. Cheng and Y. Gu, "Vestibular system and self-motion," *Frontiers Cellular Neurosci.*, vol. 12, p. 456, Nov. 2018.

[164] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2006.

[165] A. M. Stratulat, V. Roussarie, J.-L. Vercher, and C. Bourdin, "Does tilt/translation ratio affect perception of deceleration in driving simulators?," *J. Vestibular Res.*, vol. 21, no. 3, pp. 127–139, Jun. 2011.

[166] D. E. Angelaki, M. Q. McHenry, J. D. Dickman, S. D. Newlands, and B. J. M. Hess, "Computation of inertial motion: Neural strategies to resolve ambiguous otolith information," *J. Neurosci.*, vol. 19, no. 1, pp. 316–327, Jan. 1999.

[167] G. Reymond and A. Kemeny, "Motion cueing in the Renault driving simulator," *Vehicle Syst. Dyn.*, vol. 34, no. 4, pp. 249–259, Oct. 2000.

[168] A. Benson, "Sensory functions and limitations of the vestibular systems," in *Perception and Control of Self-Motion*, 1st ed., Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1990.

[169] E. L. Groen and W. Bles, "How to use body tilt for the simulation of linear self motion," *J. Vestibular Res.*, vol. 14, no. 5, pp. 375–385, Nov. 2004.

[170] H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "A review on otolith models in human perception," *Behavioural Brain Res.*, vol. 309, pp. 67–76, Aug. 2016.

[171] H. Asadi, S. Mohamed, C. P. Lim, S. Nahavandi, and E. Nalivaiko, "Semicircular canal modeling in human perception," *Rev. Neurosci.*, vol. 28, no. 5, pp. 537–549, Jul. 2017.

[172] J. Telban and F. M. Cardullo, "Motion cueing algorithm development: Initial investigation and redesign of the algorithms," State Univ. of New York, Albany, NY, USA, Tech. Rep. NASA/CR-2000-209863, 2000.

[173] A. Nesti, S. Nooij, M. Losert, H. H. Bülthoff, and P. Pretto, "Roll rate perceptual thresholds in active and passive curve driving simulation," *Simulation*, vol. 92, no. 5, pp. 417–426, May 2016.

[174] P. Pretto, A. Nesti, S. Nooij, M. Losert, and H. H. Bülthoff, "Variable roll rate perception in driving simulation," in *Proc. Driving Simulation Conf.*, 2014, pp. 1–7.

[175] F. Colombet, Z. Fang, and A. Kemeny, "Tilt thresholds for acceleration rendering in driving simulation," *Simulation*, vol. 93, no. 7, pp. 595–603, Jul. 2017.

[176] A. Berthoz et al., "Motion scaling for high-performance driving simulators," *IEEE Trans. Hum.-Mach. Syst.*, vol. 43, no. 3, pp. 265–276, May 2013.

[177] P. R. Grant, "The development of a tuning paradigm for flight simulator motion drive algorithms," Ph.D. dissertation, Univ. of Toronto, Toronto, ON, Canada, Dec. 1995.

[178] D. M. Pool, P. M. T. Zaal, M. M. van Paassen, and M. Mulder, "Effects of heave washout settings in aircraft pitch disturbance rejection," *J. Guid., Control, Dyn.*, vol. 33, no. 1, pp. 29–41, Jan. 2010.

[179] M. A. Nahon and L. D. Reid, "Simulator motion-drive algorithms—A designer's perspective," *J. Guid., Control, Dyn.*, vol. 13, no. 2, pp. 356–362, Mar. 1990.

[180] I. G. Salisbury and D. J. N. Limebeer, "An algorithm for false cue reduction and prepositioning," in *Proc. Driving Simulation Conf.*, 2014, pp. 141–148.

[181] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*. London, U.K.: Springer, 2011.

[182] S. V. Raković and W. S. Levine, *Handbook of Model Predictive Control*. Cham, Switzerland: Springer, 2019.

[183] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. London, U.K.: Springer, 2009.

[184] M. Katliar, "Optimal control of motion simulators, "Optimal control of motion simulators, "Optimal control of motion simulators," Ph.D. dissertation, Dept. Microsystems Eng., Albert-Ludwigs-Universität Freiburg, Freiburg, Germany, 2020.

[185] M. S. K. Lau, S. P. Yue, K. V. Ling, and J. M. Maciejowski, "A comparison of interior point and active set methods for FPGA implementation of model predictive control," in *Proc. Eur. Control Conf. (ECC)*, Aug. 2009, pp. 156–161.

[186] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.

[187] R. M. Freund, "Solution methods for quadratic optimization," Massachusetts Inst. of Technol., Cambridge, MA, USA, Tech. Rep., 2004.

[188] M. Kolff et al., "Motion cueing in BMW's driving simulation center: Experiences versus common knowledge," in *Proc. Driving Simulation Conf. Eur. VR*, Sep. 2024, pp. 119–126.

[189] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*, vol. 245, A. Garulli and A. Tesi, Eds., London, U.K.: Springer, 1999, pp. 207–226.

[190] C. E. T. Dórea and J. C. Hennet, "(A, B)-invariant polyhedral sets of linear discrete-time systems," *J. Optim. Theory Appl.*, vol. 103, no. 3, pp. 521–542, Dec. 1999.

[191] J.-P. Merlet, *Parallel Robots*(Solid Mechanics and Its Applications). Boston, MA, USA: Kluwer Academic Publishers, 2006.

[192] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit—An open-source framework for automatic control and dynamic optimization," *Optim. Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, May 2011.

[193] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *J. Process Control*, vol. 12, no. 4, pp. 577–585, Jun. 2002.

[194] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.

[195] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "MATMPC—A MATLAB based toolbox for real-time nonlinear model predictive control," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3365–3370.

[196] T. Englert, A. Völz, F. Mesmer, S. Rhein, and K. Graichen, "A software framework for embedded nonlinear model predictive control using a gradient-based augmented Lagrangian approach (GRAMPC)," *Optim. Eng.*, vol. 20, no. 3, pp. 769–809, Sep. 2019.

[197] A. Grancharova and T. A. Johansen, *Explicit Nonlinear Model Predictive Control: Theory and Applications*, vol. 429. Berlin, Germany: Springer, 2012.

[198] L. Wang, "Use of exponential data weighting in model predictive control design," in *Proc. 40th IEEE Conf. Decis. Control*, Dec. 2001, pp. 4857–4862.

[199] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, Mar. 2004, pp. 2023–2028.

[200] R. Jacumet, S. Wagner, M. Schwienbacher, D. Wollherr, and M. Leibold, "Sequential motion cueing algorithms: Using the best algorithm in each situation," in *Proc. Driving Simulation Conf. Eur. VR*, Sep. 2024, pp. 111–117.

[201] J. L. Meiry, "The vestibular system and human dynamic space orientation," Ph.D. dissertation, Massachusetts Inst. of Technol., Cambridge, MA, USA, 1965.

[202] L. R. Young and J. L. Meiry, "A revised dynamic otolith model," *Aerosp. Med.*, vol. 39, no. 6, p. 606–608, 1968.

[203] L. R. Young and C. M. Oman, "Model for vestibular adaptation to horizontal rotation," *Aerosp. Med.*, vol. 40, no. 10, pp. 1076–1080, 1969.

[204] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," Aug. 2017, *arXiv:1707.06347*.

[205] D. Cleij, J. Venrooij, P. Pretto, D. M. Pool, M. Mulder, and H. H. Bulthoff, "Continuous subjective rating of perceived motion incongruence during driving simulation," *IEEE Trans. Hum.-Mach. Syst.*, vol. 48, no. 1, pp. 17–29, Feb. 2018.

[206] H. Bellem, M. Klüver, M. Schrauf, H.-P. Schöner, H. Hecht, and J. F. Krems, "Can we study autonomous driving comfort in moving-base driving simulators? A validation study," *Hum. Factors*, vol. 59, no. 3, pp. 442–456, May 2017.

[207] T. M. W. Talsma, O. Hassanain, R. Happee, and K. N. de Winkel, "Validation of a moving base driving simulator for motion sickness research," *Appl. Ergonom.*, vol. 106, Jan. 2023, Art. no. 103897.

[208] S. Casas, I. Coma, J. V. Riera, and M. Fernández, "Motion-cuing algorithms: Characterization of Users' perception," *Hum. Factors*, vol. 57, no. 1, pp. 144–162, Feb. 2015.

[209] D. Cleij, "Measuring, modelling and minimizing perceived motion incongruence," Ph.D. dissertation, Faculty of Aerospace Engineering, Delft Univ. of Technol., Delft, The Netherlands, 2020.

[210] J. V. Riera López, S. Casas-Yrurzum, P. Morillo-Tena, and M. Fernández Marín, "Enhancing motion perception in vehicle simulators through a heuristic-based classical washout algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 10, pp. 1–17, Oct. 2025.

[211] S. Casas, C. Portalés, P. Morillo, and M. Fernández, "To move or not to move? Analyzing motion cueing in vehicle simulators by means of massive simulations," *Virtual Reality*, vol. 24, no. 1, pp. 93–108, Mar. 2020.

[212] R. Kirollos and C. M. Herdman, "Visual–vestibular sensory integration during congruent and incongruent self-rotation percepts using caloric vestibular stimulation," *Frontiers Cirtual Real.*, vol. 4, pp. 1–13, Oct. 2023.

[213] A. Q. Momani and F. M. Cardullo, "A review of the recent literature on the mathematical modeling of the vestibular system," in *Proc. AIAA Modeling Simulation Technol. Conf.*, Jan. 2018, pp. 1–35.

[214] J. W. Grant, C. C. Huang, and J. R. Cotton, "Theoretical mechanical frequency response of the otolithic organs," *J. Vestibular Res.*, vol. 4, no. 2, pp. 137–151, 1994.

[215] R. D. Rabbitt, "Directional coding of three-dimensional movements by the vestibular semicircular canals," *Biol. Cybern.*, vol. 80, no. 6, pp. 417–431, Jun. 1999.

[216] E. R. Damiano and R. D. Rabbitt, "A singular perturbation model of fluid dynamics in the vestibular semicircular canal and ampulla," *J. Fluid Mech.*, vol. 307, pp. 333–372, Jan. 1996.

[217] G. Frison and M. Diehl, "HPIPM: A high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[218] R. Verschueren et al., "Acados—A modular open-source framework for fast embedded optimal control," *Math. Program. Comput.*, vol. 14, no. 1, pp. 147–183, Mar. 2022.

[219] R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," in *Large-scale Nonlinear Optimization*. Boston, MA, USA: Springer, 2006, pp. 35–59.

[220] E. Adabag, M. Atal, W. Gerard, and B. Plancher, "MPCGPU: Real-time nonlinear model predictive control through preconditioned conjugate gradient on the GPU," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2023, pp. 9787–9794.

[221] S. Al-Serri et al., "Implementation of the grasshopper optimisation algorithm to optimize prediction and control horizons in model predictive control-based motion cueing algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2022, pp. 3317–3323.

[222] M. R. C. Qazani, S. M. J. Jalali, H. Asadi, and S. Nahavandi, "Optimising control and prediction horizons of a model predictive control-based motion cueing algorithm using butterfly optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.

[223] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Optimizing model predictive control horizons using genetic algorithm for motion cueing algorithm," *Expert Syst. Appl.*, vol. 92, pp. 73–81, Feb. 2018.

[224] M. Hinze, J. N. Kutz, O. Mula, and K. Urban, *Model Order Reduction and Applications*, vol. 2328. Cham, Switzerland: Springer, 2023.

[225] M. Kolff, J. Venrooij, E. Arcidiacono, D. M. Pool, and M. Mulder, "Predicting motion incongruence ratings in Closed- and open-loop urban driving simulation," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 1, pp. 517–528, Jan. 2025.

[226] D. Schramm, M. Hiller, and R. Bardini, *Vehicle Dynamics: Modeling and Simulation*. Berlin, Germany: Springer, 2014.

[227] T. Chapron and J.-P. Colinot, "The new PSA Peugeot-Citroën advanced driving simulator overall design and motion cue algorithm," in *Proc. Driving Simulation Conf. North Amer.*, 2007.

[228] P. Hansson, A. Stenbeck, A. Kusachov, F. Bruzelius, and B. Augusto, "Prepositioning of driving simulator motion systems," *Int. J. Vehicle Syst. Model. Test.*, vol. 10, no. 3, pp. 288–304, 2015.

[229] J. Eppink, M. Kolff, J. Venrooij, D. M. Pool, and M. Mulder, "Probabilistic prediction of longitudinal driving behaviour for driving simulator pre-positioning," in *Proc. Driving Simulation Conf. Eur. VR*, Sep. 2023, pp. 119–126.

[230] G. Droge and M. Egerstedt, "Adaptive time horizon optimization in model predictive control," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 1843–1848.

[231] S. M. J. Jalali, S. Ahmadian, A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "Automated deep CNN-LSTM architecture design for solar irradiance forecasting," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 54–65, Jan. 2022.

[232] T. Schwarzhuber, "Motion cueing in nonlinear vehicle dynamics driving simulation," Ph.D. dissertation, Dept. Automot. Eng., Graz University of Technology, Graz, Austria, Apr. 2022.

Robert Jacumet received the M.Sc. degree from the Technical University of Munich (TUM), Munich, Germany, in 2022. He is currently pursuing the joint Ph.D. degree with the Chair of Automatic Control Engineering, TUM, and the Department of Research and Development, BMW Group, Munich. His research interests include motion cueing, human–robot interaction, driving simulation, and model-predictive control algorithms.

Maurice Kolff received the M.Sc. and Ph.D. degrees in aerospace engineering from Delft University of Technology (TU Delft), Delft, The Netherlands. He is currently a Post-Doctoral Researcher with the Section Intelligent Vehicles, TU Delft. His work focuses on the cost function design of model-predictive control in driving simulation. His research interests include motion perception, mathematical modeling, and model-predictive control algorithms.

Joost Venrooij received the M.Sc. and Ph.D. degrees (cum laude) in aerospace engineering from Delft University of Technology, Delft, The Netherlands, in 2009 and 2014, respectively. After receiving the Ph.D. degree, he was the Project Leader with the Max Planck Institute for Biological Cybernetics, Tübingen, Germany. He is currently a Research Specialist with the BMW Group, Munich, Germany. His research interests include motion cueing, driving simulation, and vehicle modeling.

Markus Schwienbacher received the Diploma degree in mechanical engineering and the Ph.D. degree in humanoid robotics from the Technical University of Munich, Munich, Germany, in 2007 and 2013, respectively. He is currently a Research Specialist with the BMW Group, Munich. His research interests include motion cueing, motion systems, mathematical modeling, optimization techniques, computer graphics, and robotics.

Sebastian Wagner received the M.Sc. degree in electrical engineering and the Dr. rer. nat. degree in computer science from the Technical University of Munich (TUM), Germany, in 2016 and 2021, respectively. After his academic career, he worked towards the homologation of ADAS functions with Simulation at the BMW Group, Munich, Germany. He is currently a Research Specialist in driving simulation with the BMW Group. His research interests include driving simulation, motion cueing, and control theory.

Dirk Wollherr (Senior Member, IEEE) received the Dipl.-Ing., Dr.-Ing., and Habilitation degrees in electrical engineering from the Technical University of Munich, Munich, Germany, in 2000, 2005, and 2013, respectively. He is currently a Senior Researcher in robotics, control, and cognitive systems. His research interests include automatic control, robotics autonomous mobile robots, human–robot interaction, and socially aware collaboration and joint action.

Marion (Nee Sobotka) Leibold received the Diploma degree in applied mathematics and the Ph.D. and Habilitation degrees in control theory from the Technical University of Munich, Munich, Germany, in 2002, 2007, and 2020, respectively. She is currently a Senior Researcher with the Faculty of Electrical Engineering and Information Technology, Institute of Automatic Control Engineering, Technical University of Munich. Her research interests include optimal control and nonlinear control theory and the applications to robotics and automated driving.

Daan M. Pool (Member, IEEE) received the M.Sc. and Ph.D. degrees (cum laude) from TU Delft, The Netherlands, in 2007 and 2012, respectively. He is currently a tenured Assistant Professor with the Control and Simulation Section, Faculty of Aerospace Engineering, TU Delft. His research interests include cybernetics, vehicle simulation, simulator-based training, and mathematical modeling, identification, and optimization techniques.

Max Mulder (Senior Member, IEEE) received the M.Sc. and Ph.D. (cum laude) degrees in aerospace engineering from Delft University of Technology (TU Delft), The Netherlands, in 1992 and 1999, respectively, for his work on the cybernetics of tunnel-in-the-sky displays. He is currently a Full Professor with the Section Control and Simulation, Faculty of Aerospace Engineering, TU Delft. His research interests include cybernetics and its use in modeling human perception and performance, and cognitive systems engineering and its application in the design of "ecological" interfaces.