False cue influence on motion cue quality for 10 motion cueing algorithms

Duc-An Pham and Duc-Toan Nguyen
School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

Abstract
Motion simulators are becoming increasingly popular for many applications in which human sensation is important to replicate and optimize target motions. For the emulation of the perceived human acceleration, motion cueing algorithms (MCAs) have been proposed in the literature that mimics the motion sensation by a combination of actual acceleration and tilted gravity effects, termed g-force or specific force. However, their relative performance has not yet been analyzed. This paper reviews existing families of MCAs and compares their performance for a simple offline S-shaped planar test trajectory featuring only lateral acceleration. The comparison is carried out both numerically using two previously published objective measures, the “performance indicator” of Pouliot, Gosselin, and Nahon, and the “good criterion” of Schmidt, as well as subjectively by a preliminary passenger rating on a real motion platform—RoboCoaster testbed. The results show that (a) the novel optimizing MCA group exploits more effectively the workspace of the motion platform than the traditional MCA group for reducing false cue with small scale error and shape errors, (b) path-dependent tuning of MCA parameters may improve motion sensation, (c) average subjective ratings can be made to correlate well with the “good criterion” when expanded with a penalty for false angular velocity cues, and (d) the scale error of specific force seems to play the most important role to the evaluation of test subject on the motion cue quality. However, still a strong variance in subjective ratings was observed, making further research necessary.

Keywords
Motion cueing algorithm, false cues, objective assessment, driving simulator

Corresponding author:
Duc-Toan Nguyen, School of Mechanical Engineering, Hanoi University of Science and Technology, 1A-Dai Co Viet, Hai Ba Trung, Hanoi 100000, Vietnam.
Email: toan.nguyenduc@hust.edu.vn

Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
Introduction

Driving simulation systems are useful and affordable system applying on both training and researching human driving behavior. The systems have first used the parallel robot named Stewart platforms to generate the supporting motion. For reducing these costs and, most notably, extending the motion workspace, new motion platforms based on serial robots have been recently developed, enlarging the applicability of the simulating process. One example of these serial-robot ride simulators is the KUKA Robocoaster, used in the entertainment industry as well as for research purposes. Common to all motion platforms, including serial robots, is that exact target motions cannot always be precisely reproduced due to the limited workspace. In these cases, the motion cueing algorithms (MCAs) play an important role in mincing the benefit motion sensations that can be reproduced by motion platforms in the restricted available workspace. MCAs divide the target acceleration into the high-and low-frequency parts. While the high-frequency (HF) parts are reproduced by the transient linear acceleration of the motion platform, the low-frequency (LF) parts of acceleration are simulated by the tilt coordination technique. The principle of the technique is to tilt the motion platform a suitable angle to treat the human vestibular system. However, the interaction between the threshold of human motion perception and the restricted available workspace of motion platforms often leads to false cues. The negative cues induce simulator sickness because the visual, vestibular, motion perception have no consistency information. For example, when perception thresholds for tilt-angle g-force emulations are exceeded, drivers can feel the rotational motions that do not exist in the target motion. Recently, various MCAs have still been developed for different motion platform, thus, for a specific simulation task, selecting the best MCA and tuning its parameters are still the open problems.

Existing MCAs employ different techniques to blend tilt-angle g-force and proper motion acceleration. At the origin of MCAs, classical MCA (or named washout filter) was proposed by Conrad and Schmidt consisting of a low-pass filter (LP) cross-coupled with a high-pass filter (HP), whose parameters were manually adjusted offline. Due to its simplification in tuning and deploying, this algorithm has been broadly used in commercial driving simulation systems. However, the ineffective exploitation of the available workspace is the weakness of the algorithm. Besides, the classical MCA generates false cues due to the linear characteristics of the washout filters. For better exploitation of the serial robot workspace, such as the KUKA Robocoaster, the classical MCA was modified into “cylindrical classical algorithm” by Giordano et al. The authors implemented the washout filters in a defined cylindrical coordinate by which the end-effector can move along both the linear and circular motion. Therefore, the cylindrical MCA can generate larger lateral accelerations than the classical MCA. In order to adjust parameters of washout filters in realtime, the “adaptive washout filter” was introduced. The cost function of the adaptive method uses the steepest descent technique that is to minimize motion cue errors and to restrict linear movements of motion platform deviation of the filtered acceleration signal from the target signal.
The adaptive MCAs decreases the false cues compared with the classical MCA, however, the stability of adaptive MCAs significantly depends on its parameters. Further development has been to design MCAs that minimize a cost function involving the assumed perception error at the vestibular system and energy terms over the complete track. This family of methods, firstly introduced by Sivan et al., solves the stochastic optimization problem to find higher-order washout filters for four couple motions. The Optimal MCAs were later improved by Nahon and Reid and Telban et al. who use different mathematical models of vestibular systems. Zywiol and Romano considered on the tracking problem of simulated and target signal to propose a time backward optimal MCA. The algorithm solves the tracking problem with the linear model of tilt-coordination in offline mode to find the best exploitation of the available workspace for both translational and rotational motion. More recently, also novel model predictive control (MPC) was applied to the problem of motion cueing. The algorithm takes the various restriction into account, such as workspace limits, the motion perception thresholds (angular velocity, perception of linear acceleration), etc. These algorithms show substantial improvements in motion sensation fidelity and are able to fill better the available workspace, but still, have some limitations in terms of computational efficiency and avoidance of false cues. Khusro et al. proposed a nonlinear MPC solution that uses the Stewart platform’s nonlinear kinematics into the MPC algorithm to improve cueing fidelity and maximize workspace. The author also employed the adaptive weights tuning to smooth the platform’s movement toward its physical limits. Katliar et al. developed an MPC-based MCA for a Cable-Robot-based motion simulator that includes motion platform actuation. Their key conclusion was that an MPC-based MCA with a complex model can be executed in real time with the right software and numerical approaches. Katliar et al. used a model predictive controller (MPC) for real-time control of a motion simulator based on an 8-degree-of-freedom serial robot. By moving the human participant sitting within the cabin positioned at the end effector, the controller aims to properly duplicate six reference signals (accelerations and angular velocities in the body frame of reference) acquired from a simulated or actual vehicle. Besides, the use of a nonuniform weighting strategy to stabilize the MCA using MPC with a short prediction horizon and improved weighting modification is proposed in the study of Mohammadi et al. The author mentioned that reduced prediction horizons are beneficial because they reduce computing strain, but they also move the system toward instability. Moreover, Asadu et al. proposed a method based on compensators and generic algorithm that has the goal of minimizing the fitness function, which can lead to greater fidelity motions and better solutions. The following factors are considered when creating the fitness function such as minimizing the translational and rotational human sensation error between the real and simulator drivers; maximizing the correlation coefficient; minimizing the angle and linear displacements; minimizing the acceleration, velocity, and displacement of the platform; and minimizing sensation error fluctuations.
The objective of the present work is (1) to review currently available MCAs for the families of classical, adaptive, optimal control, and model-predictive control algorithms in offline mode; (2) to numerically compare their performance for a simple planar S-shaped test ride using two existing objective measures for perception evaluation, the “performance indicator” of Pouliot et al.,19 and the “good criterion” of Fischer20; and (3) to correlate the thus obtained objective measures with preliminary subjective ratings on a real simulator ride.

The rest of the paper is structured as follows. First, the testbed is described in Section 2. Section 3 then reviews briefly all existing MCAs, whose perception quality on the test ride is evaluated numerically in Section 4 and by preliminary subjective ratings in Section 5. The results finally are discussed in the conclusions section.

**Driving simulator based on KUKA robocoaster KR500**

The motion platform based on KUKA robocoaster is applied for a driving simulator due to its flexibility in rotational motions that are restricted in the steward motion platform. For example, the motion platform in the Laboratory for Mechanics and Robotics (LMR) at University of Duisburg-Essen uses a KUKA KR500/1 TÜV Robocoaster robot, shown in Figure 1. The motion platform can lift driver seating on a Maurer Söhne rollercoaster seat at the robot end effector. Besides, the head-mounted display unit—Visette45—provides the virtual reality of a 3D environment to the driver. The trajectory of the motion platform is generated by motion cueing algorithms and then transferred to the robot’ controller. The virtual reality environment is simultaneously computed to the trajectory of the robot in the computer visualizing environment, then is delivered in stereo 3D to the head-mounted display. The physical limits of the robot’s joints are shown in Table 1. When compared to the Stewart platform, the KUKA robocoaster can reproduce a larger lateral motion by utilizing its enormous circular motion due to the large rotating space of the base joint (joint 1). On the other hand, the simulation of vertical motion is still restricted in a range of 0.127 m.

**Review of existing MCAs**

There exists a great variety of motion cueing algorithms (MCAs) in the current literature. They are mainly devoted to mimic motion perception by a superposition of actual passenger acceleration with tilted gravity vector effects, termed the specific force. In a driving simulator, the motion cues are dominantly sensed by the vestibular system that perceives translational and rotational motion represented by specific force and angular velocity, respectively.22 This section reviews the principal MCA families with particular interest on the transversal accelerations $P a_v = [0, a_{vy}, 0]^T$ in body-fixed coordinates, where $x, y, z$ are the “front,” “lateral,” and “up” coordinates, respectively, and the target angular velocity is $P \omega_v = [\omega_{vx}, \omega_{vy}, \omega_{vz}]^T = [0, 0, 0]^T$. The rotation about the front axis is measured by
the roll angle $\phi$, and the linear acceleration is measured at a reference point “V” in the cabin, which is assumed to be located at the passenger’s head “P.”

**Classical MCA with cylindrical coordinates (CL)**

This algorithm (Figure 2), introduced by Giordano et al.,\(^1\) applies the standard classical washout algorithm to cylindrical coordinates. Hereby, absolute Cartesian
coordinates \( \mathbf{p} = [X, Y, Z]^T \) of the cabin (\( Z \) is “up”) are transformed to cylindrical coordinates according to equation (1).

\[
\xi = [R, \alpha, Z]; \text{with } R = \sqrt{X^2 + Y^2}; \tan(\alpha) = \frac{Y}{X}.
\]

At the robot base, a vertically rotating frame \( K_W \) is attached such that the cabin reference point always lies in its \( x/z \) plane. The cabin orientation is measured with respect to this frame by pitch and roll rotations \( \theta_W, \varphi_W \), in that order, collected in the vector \( \beta_W \). The target acceleration \( \mathbf{a}_v \) is first transformed to \( K_W \) by a rotation matrix \( \mathbf{w}_R \) and then to cylindrical accelerations \( \mathbf{\tilde{\xi}} \) by the linear mapping \( C(\xi) = \text{diag}\{1, \frac{1}{R}, 1\} \). These are then fed into the three channels of linear high-pass filters (equation (2)) yielding high-frequency components \( \mathbf{\tilde{\xi}}_{HP} \). Due to the circular motion, centripetal and Coriolis accelerations \( -R\ddot{\alpha}_{HP}^2 \) and \( 2\ddot{R}\dot{\alpha}_{HP} \), respectively, are generated which must be compensated by cabin pitch and roll tilt angle contributions \( \beta_T = -\arcsin\frac{\mathbf{a}_v \cdot \mathbf{g}}{g}, \varphi_T = \arcsin\frac{\mathbf{a}_v \cdot \mathbf{g} \cos \theta}{g} \), respectively, collected in vector \( \mathbf{\beta}_T \). These are added to the result of the low-pass filter \( LP(s) = 1 - HPa(s) \) applied to the target acceleration. The final cabin pitch/roll angles \( \beta_w \) are assumed to be the sum of \( \beta_T \) and the pitch/roll angles \( \beta_{HP} \) from the time integral of the high-pass filter (equation (3)) applied to the target angular velocity transformed to cabin coordinates. Prepending the rotation of the frame \( K_W \) to the pitch/roll angles \( \beta_w \) as a yaw angle gives the global orientation \( \beta_s \) of the cabin as yaw/pitch/roll angles (in that order). The filter parameters for all channels used in our comparison are shown in Table 2, where \( K \) is the proportional gain, \( \zeta \) is the damping ratio, \( \omega_n \) is the natural frequency and \( \omega_b \) is the break frequency.

\[
HPa(s) = \frac{Ks^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \cdot \frac{s}{s + \omega_b},
\]

\[
HP\omega(s) = \frac{Ks^2}{(s^2 + 2\zeta \omega_n s + \omega_n^2)}
\]

Figure 2. Control diagram of classical MCA in cylindrical coordinate.
Adaptive MCA with cylindrical coordinates (AD)

The algorithm (Figure 3) is developed from the classical MCA with cylindrical coordinates by using a steepest descent method in an adaptive block “ADAPT” to adapt the parameter set \(f = (K, \omega_n, \zeta)\) of the high-pass filter \(HP_2 = \frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{s + \omega_n}\) in real time to minimizing the following cost function (equation (4)).

\[
J = \frac{1}{2} \left[ w_a (\xi_F - \xi_{HP})^2 + w_s \xi_{HP}^2 + w_t \xi_{HP}^2 + w_k (K - K_0)^2 + w_\zeta (\zeta - \zeta_0)^2 + w_\omega (\omega_n - \omega_{n0})^2 \right]
\]  

(4)

Here, \(\xi_F\) is the filtered acceleration through \(HP_1 = \frac{s^2 + \omega_n}{s + \omega_n}\), \(K_0, \zeta_0, \omega_{n0}\) are the reference anchor values of the filter parameters, and \(w_{(\cdots)}\) are weighting parameters (Table 3), which penalize the deviation between the target and the simulated acceleration, and restrain the cabin’s movement.

Optimal washout MCAs (OpS, OpNR, Tel-YoMe, and Tel)

The problem was first raised by Sivan et al.\(^7\) The authors defined the linear quadratic optimal control for tilt coordination problem to find higher-order filters \(W(s)\) (Figure 4). The filters minimize the cost function (equation (5)), and the state space model (equation (6)).
Here, \( E \{ \} \) is the mean estimator, and \([t_0, t_f]\) is the time-simulation time. The output vector \( y = [e, x_c]^T \) contains the sensation error \( e(t) = [\Delta \hat{f}_y, \Delta \hat{\omega}_y]^T \) of the specific force and angular velocity, respectively, and the \( x_c \) simulator state variables. It is important to note that the state variable \( x \), system inputs \( u \) and \( x_c \), as well as the weighting matrices \( R = \text{diag} \{ r_1, r_2 \} \), \( Q = \text{diag} \{ q_1, q_2 \} \), and \( R_c = \text{diag} \{ r_{c1}, \ldots, r_{c4} \} \), differ between methods and are given in Tables 4 and 5, respectively. For example, in the basic approach Sivan (equation (6)) defined the state variable: \( x = [x_v, x_s, x_n]^T \), where the state variables in the vehicle \( x_v = [x_v^{oto}, x_v^{sc}, \int a_{vy}, \int a_{vy}]^T \) with otolith state variable \( x_v^{oto} = [\hat{f}_y - G_B f_{vy}] \), and semi-circular state variable \( x_v^{sc} = [\hat{\omega}_y - G_S B_S \varphi_y, \int \hat{\omega}_y]^T \). Similarly, in term of the simulator state variables \( x_s \) is defined as \( x_v \) (with subscript “v” is replaced by “s”). Note that \( x_n = [a_{vy}, \varphi_v] \) and \( H_n \) are the state of white noise filter and state-space noise, respectively. The matrices in the first equation of the dynamic model in (equation (6)) are represented as equation (7).
\[ A = \begin{bmatrix} A & 0 & B_c \\ 0 & A & 0 \\ 0 & 0 & A_n \end{bmatrix}, B = \begin{bmatrix} 0 \\ B \\ 0 \end{bmatrix}, \text{ and } H = \begin{bmatrix} 0 \\ 0 \\ H_n \end{bmatrix} \] (7)

while, the second equation has the matrices as equation (8).

\[ C = \begin{bmatrix} -C & C & -D \\ 0 & C_c & 0 \end{bmatrix}, D = \begin{bmatrix} D \end{bmatrix}. \] (8)

Moreover, the state matrix \( A_n = \text{diag}\{-\gamma_1, -\gamma_2\} \) and input matrix \( B_n = \text{diag}\{\gamma_1, \gamma_2\} \) are used for modeling white noise transmission, \( C_c = [0, 1] \), while the matrices \( \bar{A}, \bar{B}, \bar{C}, \bar{D} \) integrate the sub-matrices of vestibular dynamic model as \( \bar{A}, \bar{B}, \bar{C}, \bar{D} \) depending on the method. For example, the basic approach of Sivan et al.\(^7\) (OpS) uses equation (9).

\[ \bar{A} = \begin{bmatrix} \bar{A} & 0 \\ 0 & A_c \end{bmatrix}, \bar{B} = \begin{bmatrix} \bar{B} \\ B_c \end{bmatrix}, \bar{C} = \begin{bmatrix} \bar{C} \\ 0 \end{bmatrix}, \bar{D} = \bar{D}, \] (9)

with equation (10).
Table 6. The otolith parameters (sway direction).

|         | \( A_0 (s^{-1}) \) | \( B_0 (s^{-1}) \) | \( B_1 (s^{-1}) \) | \( G_0 (s^2/m) \) | \( \delta_0 (m/s^2) \) |
|---------|---------------------|---------------------|---------------------|---------------------|---------------------|
| OpS\(^7\) | 0.076               | 0.19                | –                   | 2.16                | 0.47                |
| OpRN\(^4\) | 0.076               | 0.19                | –                   | 5.86                | 0.17                |
| Tel-YoMe\(^8\) | 0.076               | 0.19                | 1.5                 | 8.82                | 0.17                |
| Tel\(^8\) | 0.1                 | 0.2                 | 62.5                | 294.12              | 0.17                |

"-": the abandoned parameters.

Table 7. Semicircular parameters (roll angle).

|         | \( \tau_1 (s) \) | \( \tau_1 (s) \tau_1 (s) \) | \( \tau_0 (s) \) | \( \tau_0 (s) \) | \( G_s (s^2/rad) \) | \( \delta_s (\degree/s) \) |
|---------|------------------|-----------------------------|-----------------|-----------------|---------------------|---------------------|
| OpS\(^7\) | 5.9              | 0.003                       | –               | –               | 233                 | 1.45                |
| OpRN\(^4\) | 6.1              | 0.1                         | –               | –               | 118.55              | 3                   |
| Tel-YoMe\(^8\) | 5.73             | 0.005                       | 80              | 0.06            | 28.6479             | 2                   |
| Tel\(^8\) | 5.73             | 0.005                       | 80              | 0.06            | 28.6479             | 2                   |

\[
\begin{align*}
\tilde{A} &= \begin{bmatrix} -B_0 & 0 & 0 \\ 0 & -a_S & 1 \\ 0 & -b_S & 0 \end{bmatrix}, \quad
\tilde{B} = \begin{bmatrix} G_O(A_0 - B_0) & -G_Og(A_0 - B_0) \\ 0 & a_S b_S G_S \\ 0 & -b_S^2 G_S \end{bmatrix}, \\
\tilde{C} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad
\tilde{D} = \begin{bmatrix} G_O & -G_Og \\ 0 & G_S b_S \end{bmatrix}, \quad
A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad
B_c = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.
\end{align*}
\]

(10)

Here, \( A_0, B_0, G_O, G_S \) are the physical parameters of the dynamic models of vestibular system (Tables 6 and 7), and \( a_S = (\tau_1 + \tau_2)/\tau_1 \tau_2 \) and \( b_S = 1/\tau_1 \tau_2 \).

Telban et al.\(^8\) (Tel) extended these models to higher-order filters

\[
scc(s) = G_S \frac{\tau_0 \tau_1 s^2(1 + \tau s)}{(\tau_1 s + 1)(\tau_2 s + 1)(\tau_0 s + 1)}
\]

(11)

and

\[
oto(s) = G_O \frac{(s + A_0)}{(s + B_0)(s + B_1)},
\]

(12)

for the semicircular and otolith system, also using a Young-Meiry vestibular model (Tel-YoMe). An intermediate filter complexity was introduced by Nahon and Reid\(^6\) (OpNR). Tables 6 and 7 summarize the parameters used in these optimal washout signals, where missing parameters mean that this order is not present in the transfer function. The scalar elements \( W_{ij} \) of the washout filter in Figure 4 are then determined in offline mode as the components of the \( 2 \times 2 \) matrix.
\[ W(s) = - \left( I + F_2(sI - A_s)^{-1}B_s \right)^{-1} \cdot \left( F_1(sI - A_r)^{-1}B_r + F_3 \right), \]  
(13)

where \( F_i, i = 1, \ldots, 3 \) are the submatrices corresponding to the subvectors \( x_r, x_s, x_n \) of the feedback matrix in the expression \( u = -Fx \) obtained as

\[ F = [F_1, F_2, F_3] = R_2^{-1}(B^T P + R_{12}^T), \quad R_1 = C^T Q C, \quad R_{12} = C^T Q D, \quad R_2 = R + D^T Q D. \]  
(14)

Note that the solution of the algebraic Riccati equation \( P \) is the unique and non-negative.

\[ P(A - BR_2^{-1}R_{12}^T) + (A - BR_2^{-1}R_{12}^T)^T P + R_1 - R_{12}R_2^{-1}R_{12}^T - PBR_2^{-1}B^T P = 0 \]  
(15)

The washout filter works by first transforming the target acceleration \( a_{vy} \) through the block “\( TF \)” to inertial coordinates and then filtering it through \( W_{11} \) to yield the simulator acceleration \( a_{sy} \) that passes through the integrator block to produce the simulator movement \( S_y \). Note that, the target acceleration \( a_{vy} \) is first filtered by \( W_{12} \)—tilted coordination filter—and then the filtered part is summed with the filtered roll target angle through \( W_{22} \) to produce the roll tilt angle \( \phi_s \). The cross-filter \( W_{21} \) commonly has an insignificant gain, thus it can be neglected.

**Offline optimal control algorithms (ZyRo and ZyRo*)**

The algorithm was first introduced by Zywiol and Romano,\(^9\) solves the tilt coordination as an optimal output tracking problem, whose cost functional (equation (16)) is minimized by the appropriate time-variant input vector (controlling vector) \( u = [u_1, u_2]^T \) with \( u_1 \) representing for specific force at the pilot’s head, and \( u_2 \) representing for angular velocity. The input components \( u_1, u_2 \) are passed through low-pass filter (PT1) to avoid the promptly unpractical change of the specific force and the angular velocity, respectively. The objective is that the output signal \( y = [f_{sy}, S_y]^T \) containing the specific force and the lateral displacement of the cabin tracks the target output vector \( y_{ref} = [f_{sy}, 0]^T \) as close as possible while “pulling” the cabin to the home configuration. Let \( x = [f_{sy}, \phi_s, \phi_s, S_y, S_y] \) be the state variables and \( A, B, C \) be the corresponding system matrices of the control system. The control input vector can be computed as equation (17).

\[ J = \int_{t_0}^{t_f} \left\{ y - y_{ref} \right\}^T Q \left\{ y - y_{ref} \right\} + u^T R u \} dt \]  
(16)

\[ u = -R^{-1}[B^T U x + B^T S], \]  
(17)

where matrix \( U \), which is the positive semi-definite, is the solution of the following differential Riccati equation (equation (18)).
\[ U = \frac{1}{C \alpha} \]

Note that \( S \) in (equation (17)) is the solution of co-state equations, and it is solved in backward time as in the equation (19).

\[ \dot{S} = -\left[A^T - U B R^{-1} B^T\right] S + C^T Q y_{ref} \quad (19) \]

The components of weighting matrices \( Q = \text{diag}\{q_1, q_2\} \), \( R = \text{diag}\{r_1, r_2\} \) are tuned for tracking purposes of specific force and returning to the home position. On the other hand, the first break frequencies of low-pass filters (PT1) \( c \) and \( \gamma \) from Figure 5 are tuned for smooth response of angular velocity and specific force. The parameters are summarized in Table 8 for the original paper of Zywiol and Romano\(^9\) (termed “ZyRo”) as well as for an alternative selection of values tuned for the motion platform based on KUKA robocoaster.

### Table 8. Parameters of ZyRo algorithm.

|       | \( \gamma (s^{-1}) \) | \( c (s^{-1}) \) | \( q_1 \) | \( q_2 \) | \( r_1 \) | \( r_2 \) |
|-------|------------------------|-----------------|--------|--------|--------|--------|
| ZyRo  | 20                     | 10              | 1      | 0.005  | 4      | 1      |
| ZyRo* | 20                     | 100             | 199    | 0.4    | 0.01   | 199    |

\[ \dot{U} = -UA - A^T U - C^T QC + UBR^{-1} B^T U \quad (18) \]

### Model predictive control algorithms (MPC and MPC\(^*\))

MPC algorithms base on model predictive control for the tilt coordination problem. The MPC is the multivariable optimization method that generates the best future behavior of a plant output by observing possible responses in a future fixed interval (prediction horizon). The possible responses are computed with corresponding parametrized future inputs over the prediction horizon while respecting the defined constraints. Therefore, the MPC algorithms can deal with both
simulator limitations and human motion perception thresholds. Moreover, the algorithm can improve the motion cueing quality. There exist implicit MPC\textsuperscript{10} based on an online optimization technique with Quadratic Programing (QP) as well as explicit MPCs\textsuperscript{12} based on an offline optimization technique.

**Implicit MPC motion cueing algorithms.** With the aim of reproducing the target specific force while keeping the simulator cabin inside the simulator workspace, Augusto and Loureiro\textsuperscript{10} designed an implicit MPC algorithm by the minimization of the cost functional (equation (20)).

\[
J(\Delta u) = [y - y_{\text{ref}}]^T Q [y - y_{\text{ref}}] + u^T S u + \Delta u^T R \Delta u,
\]  

\text{(20)}

where $y$, $y_{\text{ref}}$, $u$, and $\Delta u$ are the output vector, its reference values, the input vector, and its increment, respectively. Besides, $Q = \text{diag}\{q_1, \ldots, q_n\}$, $S = \text{diag}\{s_1, \ldots, s_n\}$, and $R = \text{diag}\{r_1, \ldots, r_n\}$ are weighting matrices. Augusto and Loureiro\textsuperscript{10} developed the MPC algorithm for four operating models: pitch/surge, roll/sway, heave, and yaw. Here, the lateral mode (roll/sway), which will be termed MPC in the following (Figure 6), are considered. The control vector consists of three inputs $u = [\dot{\varphi}_x, \dot{\varphi}_r, a_{sy}]^T$ denoting the tilt roll rate for simulating specific force, the actual vehicle roll rate, and the simulated acceleration, respectively. For the output tracking problem, the solution of the MPC is the optimized control signal $u$ such that the output vector $y = [\hat{\omega}_{sx}, \varphi_x, \hat{f}_{sy}, S_y, v_y]^T$ comprising of the sensation of roll angular velocity, the cabin roll angle, the lateral sensation of specific force, the velocity, and position of the cabin, respectively, tracks the corresponding reference signals $y_{\text{ref}} = [\hat{\omega}_{sx}, 0, f_{sy}, 0, 0]^T$. The matrix $H$ in Figure 6 maps the tilt angle $\dot{\varphi}_x$ and simulated acceleration $a_{sy}$ to the simulated specific force, $f_{sy}$, $H = [-g, 1]$. While applying the MPC algorithm with original parameters (Table 9 with $p, m$ are the number of time step for horizontal and control prediction, respectively), it turned out that the motion had a tendency to dephase with respect to the target specific

**Figure 6.** Original control system of MPC algorithm.
force and to undulate in the angular velocity. Note that an unstable, jerky motion of the cabin could exist if the weighting matrices are selected to achieve the desired specific force fidelity. To avoid these problems, an alternative MPC algorithm termed MPC* (Figure 7) are proposed. The MPC has two inputs and insert, as in the optimal control algorithm,\(^9\) two first-order low-pass filters in both the translational and rotational input channel (Figure 7, Table 9) to smooth the input signals. The break frequency of two low-pass filters were selected as in ZyRo* algorithm. Likewise, the output vector was reduced to \(y = [\dot{\omega}_{sv}, S_y, f_{sy}]^T\) with reference values \(y_{ref} = [\dot{\omega}_{sv}, 0, f_{sy}]^T\). Moreover, while the original MPC approach uses the extended linearized vestibular model proposed in Reid and Nahon,\(^4\) we used the reduced model proposed therein without with additional time parameters \(B_1 = 1.5(1/s)\) and \(\tau_a = 30(s)\) (Tables 6 and 7) to reduce computational effort. Besides, the horizon \(p\) and control prediction \(m\) parameters were chosen larger than those of original MPC to improve the tracking quality.

**Explicit MPC motion cueing algorithm.** Based on the multi-parameter programing toolbox (MPT), developed by the Automatic Control Laboratory of ETH, Zürich, and the solution for the tracking problem, Fang and Kemeny designed the MCA based on an explicit MPC for real-time driving simulators (Figure 8), in which the optimized input vector minimizes the cost function (equation (21)) in the prediction horizon \(N\). Here, \(u_i, x_i,\) and \(y_i\) denote the \(i\)th predicted input vector, predicted state

### Table 9. Weighting values of MPC.

|        | \(s_i\) | \(r_i\)  | \(q_i\)  | \(p\)   | \(m\)  |
|--------|---------|----------|----------|---------|--------|
| MPC\(^4\) | 1|20|1 | 1|8|0|0|1 | 10000|20|10|10|15 | 18 | 6 |
| MPC\(^*\) | 30|0|1 | 9000|0|1 | 50|5|99 | 400 | 200 |

![Figure 7. Control system of MPC* algorithm.](image)
vector, and predicted output vector, respectively; \( x_N \) represents the final state vector and \( y_{\text{ref},i} \) stand for the reference vector; the weighting matrices are diagonal matrices such as \( Q_N = \text{diag}\{q_1, \ldots, q_n\} \), \( Q_x = \text{diag}\{q^x_1, \ldots, q^x_n\} \), \( R = \text{diag}\{r_1, \ldots, r_n\} \), and \( Q_y = \text{diag}\{q^y_1, \ldots, q^y_n\} \).

\[
J_N(x_k) = \min_{u_0, u_1, \ldots, u_{N-1}} x_N^T Q_N x_N + \sum_{i=0}^{N-1} u_i^T R u_i 
+ \sum_{i=0}^{N-1} x_i^T Q x_i + \sum_{i=0}^{N-1} (y_i - y_{\text{ref},i})^T Q_y (y_i - y_{\text{ref},i})
\]  

(21)

For smoothing motion in the available simulator’s workspace, Fang and Kemeny used a braking control law which avoids the jerky stop at the limit position (equation (24)). Starting from state-space model (equations (22) and (23))

\[
x_k = \begin{bmatrix} S_{yk} \\ v_{yk} \\ \dot{\varphi}_k \\ y_{\text{ref},k} \end{bmatrix}, \quad A_d = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0.5T_s^2 & 0 \\ T_s & 0 \\ 0 & 0.5T_s^2 \\ 0 & T_s \end{bmatrix}
\]  

(22)

\[
C_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & g & 0 \end{bmatrix}, \quad D_d = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad u_k = \begin{bmatrix} a_k \\ \dot{\varphi}_k \end{bmatrix}
\]  

(23)

the braking law is defined as

**Figure 8.** Control diagram of the explicit MPC.
which subject to equation (25)

\[
\begin{align*}
|S_{yk} + c_v T_1 S_{yk} + 0.5 c_u S_{yk}^2 T_1^2| &< 0.5 S_{y,\text{max}} \\
|\varphi_k + c_v T_2 \varphi_k + 0.5 c_u \varphi_k^2 T_2^2| &< 0.5 \varphi_{\text{max}}
\end{align*}
\]

(24)

\[
\begin{align*}
|x_{1k}| &\leq S_{y,\text{max}} |x_{2k}| \leq v_{y,\text{max}} |u_{1k}| \leq a_{y,\text{max}} \\
|x_{3k}| &\leq \varphi_{\text{max}} |x_{4k}| \leq \omega_{\text{max}} |u_{2k}| \leq \dot{\omega}_{\text{max}},
\end{align*}
\]

(25)

where $\varphi_k$, $\dot{\varphi}_k$, $v_{yk}$, and $S_{yk}$, represent for the discrete state variables of tilted angle, tilted rate, velocity, and position of the driving simulator, respectively. The reference vector $y_{\text{ref}}, k$ (Figure 8) is selected for pulling the simulator to its home position while tracking the target specific force $f_{vy}$. Similarly, the discrete acceleration $a_{sy}$ and angular acceleration $\varphi$ are $a_k$ and $\varphi$, respectively. The limits values $\{\cdot\}_{\text{max}}$, are defined as the absolute maximum values of the simulator’s motion quantities such as position, velocity, lateral acceleration, and the motion perception threshold: angular velocity, and angular acceleration. The sample time of the discrete dynamic model is $T_s$, and the parameters of the braking control law including $c_v$, $c_u$, $T_1$, and $T_2$ determine the smoothing approach of the simulator to its workspace limits. Additionally, the important condition for selecting possible parameters in order to avoid overshoots in equation (24) is recommend as $(c_v^2/2c_u)>1$. Starting from the original parameters published by Fang and Kemeny,\textsuperscript{12} the values of parameters are selected to gain feasible simulations for tracking specific force on the Robocoaster, obtaining the “exMPC” algorithm (Table 10).

### Numerical comparison of MCA families

In order to compare the perceived quality of outcomes of the different MCA families, a simple virtual ride at a constant speed of 6 m/s along a planar left-right S-shaped trajectory with smoothed transitions was chosen (thick lines in Figure 9), featuring purely lateral accelerations. Hereby, two cases were considered. In case 1, the original target lateral acceleration was fed into the 11 reviewed MCAs. In case 2, for a preparatory subjective test, which verifies the correlation between the

| Model | Limit values | Parameters of law brake |
|-------|--------------|------------------------|
| $T_s$ | 0.125 (s)    | $S_{y,\text{max}}$ 1.5 (m) | $c_u$ 1.5 (s) |
| $q_{d}$ | 4/0.01/299  | $v_{y,\text{max}}$ 10 (m/s) | $c_v$ 1 (s) |
| $r_i$ | 0.1/2       | $a_{y,\text{max}}$ 14 (m/s$^2$) | $T_1$ 2 (s) |
| $q_{yi}$ | 1/1/1      | $\varphi_{\text{max}}$ 0.5 (rad) | $T_2$ 1 (s) |
| $N$  | 5           | $\omega_{\text{max}}$ 0.25 (rad/s) | – |
| $Nc$ | 5           | $\omega_{\text{max}}$ 10 (rad/s$^2$) | – |

### Table 10. Explicit MPC parameters as tuned for the KUKA robocoaster motion platform.
numerical criteria, and passengers’ rates according to their motion perception, the target accelerations were scaled according to the rules a) the range of simulated motion is within the workspace of the simulators, b) the $k_y = [0.4, 1]$ (equation (27)). The range of $k_y$ was referred to the scale factor experiment of Berthoz et al.\textsuperscript{23} The algorithm MPC was removed from this case as it featured a too large phase shift with respect to the original signal in the specific force.

Figure 9(a) for case 1 shows that most signals follow well the target specific force, except the original MPC, which displays a strong phase shift. On the other hand, Figure 9(b) shows that all MCAs violate the threshold level for the angular velocity, beyond which the tilt rotation is sensed as such and not as an increase of lateral g-force. Hereby, the most severe angular velocity threshold violation is reached by the exMPC algorithm. By scaling down the target accelerations (Figure 10), the angular velocity violations could be reduced for all MCAs except the exMPC algorithm, but the target specific force was not reached by all MCAs except MPC*. 

Figure 9. Results of MCA families of lateral specific forces and angular velocity for the test track – case 1: (a) simulated lateral specific forces and (b) simulated angular velocities.
For the motion perception evaluation, only a few objective measures exist in the literature. A comprehensive objective measures were proposed by the “performance indicator” of Pouliot et al.\textsuperscript{19} and the “good criterion” of Fischer.\textsuperscript{20}

The “performance indicator,” $\lambda_1$ and $\lambda_2$ (equation (26)), analyzes the quality of the simulated signals. The first indicator $\lambda_1$ is the normalized average of the square error between the target and simulated signals, while the second indicator $\lambda_2$ is the normalized average square error of the rates of change. Each indicator combines two sub-indicator indexed “$f$” for the specific force, and indexed “$\omega$” for the angular velocity. The two sub-indicators are normalized by $a_{\text{max}}$ and $\omega_{\text{max}}$ (equation (26)) which represent the maximal acceleration and the maximal angular velocity of the simulator, respectively.

\[
\lambda_1 = 100 \cdot \left( \frac{\lambda_{1,f}}{a_{\text{max}}} + \frac{\lambda_{1,\omega}}{\omega_{\text{max}}} \right) ; \\
\lambda_2 = 100 \cdot \left( \frac{\lambda_{2,f}}{a_{\text{max}}} + \frac{\lambda_{2,\omega}}{\omega_{\text{max}}} \right)
\]  

Figure 10. Results of MCA families of lateral specific forces and angular velocity for the test track – case 2: (a) simulated lateral specific forces and (b) simulated angular velocities.
Further, the “good criterion,”\textsuperscript{20} separates the translational and rotational sub-indicators into a “scale” component (index “sc”) for estimating strong stimulation of simulated signals, and a “shape” component (index “sh”) for predicting the false cues in the simulated signals. We describe here the scale and shape components for the specific force that applies analogously to the rotational part. Considering on the time $t_j$ in the $N + 1$ time sample, let $f_{si}^j, f_{vi}^j$ be the simulated and target (“vehicle”) specific forces, respectively, for a the coordinate $i \in \{x,y,z\}$. Regarding the simulated and target specific force, the global scaling factor is defined as equation (27)
\[
k_i = \left( \sum_{j=0}^{N} |f_{ij}^f| \right) / \left( \sum_{j=0}^{N} |f_{ij}^v| \right), \quad i \in \{x, y, z\}. \tag{27}
\]

At a sample point, the different components are equation (28).

\[
\Delta f_{i,sc}^j = f_{ij}^f(1 - k_i); \quad \Delta f_{i,sh}^j = k_i f_{ij}^v - f_{ij}^s, \quad i \in \{x, y, z\} \tag{28}
\]

The partial criteria for specific force components

\[
\lambda_{1f,sc} = \frac{1}{N + 1} \cdot \sqrt{\sum_{j=0}^{N} \sum_{i \in \{x, y, z\}} \left( \Delta f_{i,sc}^j \right)^2} \tag{29}
\]

\[
\lambda_{1f,sh} = \frac{1}{N + 1} \cdot \sqrt{\sum_{j=0}^{N} \sum_{i \in \{x, y, z\}} \left( \Delta f_{i,sh}^j \right)^2} \tag{30}
\]

From this, the “good criteria” for simulator cues and their time derivative are obtained as:

\[
\lambda_1^* = (100/f_{nr}) \cdot \lambda_{1f} + (100/\omega_{nr}) \cdot \lambda_{1\omega} \tag{31}
\]

\[
\lambda_2^* = (100/f_{nr}) \cdot \lambda_{2f} + (100/\dot{\omega}_{nr}) \cdot \lambda_{2\omega} \tag{32}
\]

With

\[
\lambda_{pq} = (\lambda_{pq,sc} + \lambda_{pq,sh}), \quad p \in \{1, 2\}; \quad q \in \{f, \omega\} \tag{33}
\]

\[
f_{nr} = \max(|f_{ij}^v|)(i \in \{x, y, z\})
\]

\[
\omega_{nr} = \max(\omega_{vx}, \omega_{sx}, |\omega_{vy} + \omega_{sy}|, |\omega_{sz}|)\]

\[
\dot{f}_{nr} = \max(|f_{ij}^v|)(i \in \{x, y, z\})
\]

\[
\dot{\omega}_{nr} = \max(\dot{\omega}_{vx}, \dot{\omega}_{sx}, |\dot{\omega}_{vy} + \dot{\omega}_{vy}|, |\dot{\omega}_{sz}|)
\]

The comparison the two objective measures are shown in Figure 11(a), and the results of the partial “good” components in Figure 11(b), both for case 2, as case 1 produced too high false angular velocity cues for almost all rides.

One can see that both objective measures yield a similar distribution for the different MCAs, and that the path-tuned optimal control ZyRo* algorithm displays the second-best perception quality. The algorithm has better both index of the good criterion than explicit model-control exMPC algorithm, which take computation times of approximately from 30 min to 3 h on a PC desktop with CPU type—quadcore 2.66 GHz, as compared to 1.5 min for the ZyRo* algorithm. The poor performance of the exMPC algorithm may be explained by the required handling of the ride simulator limits, which forces the simulator to decelerate even when the tilting angle is not yet large enough to compensate for the lack of specific force,
leading to unrealistic departures from the target acceleration. Overall the here newly proposed adapted MPC* and ZyRo* algorithms are shown to yield the best results for both objective measures. However, when the computational complexity of the MPC* and the ZyRo* are compared, the MPC* consumes roughly thirty times the ZyRo*. The length of the prediction horizon and the complexity of the control model used in the MPC* appear to be the reasons behind the large time consuming. Moreover, the parameters of the ZyRo algorithm are transparent to the simulated quantities, making parameter adjustment easier than that of classical, adaptive, and optimum washout filters, as well as MPC*. However, because the washout effect is not used in the simulated angles, the steady-state must be adjusted to bring the final simulated angle to zero.

It’s worth noting that both MPC* and ZyRo* effectively use the motion platform’s workspace to discover the best combination of translational and rotational motion to reproduce the two biggest scaled specific forces. However, because the MPC* closely matches the target specific force in Figure 10(a), it provides angular velocity false cues.

| Grade Meaning  | 1 Very good | 1.3 Good | 1.7 Satisfactory | 2 Sufficient | 2.3 Not sufficient |
|---------------|-------------|---------|-----------------|-------------|-----------------|

| Table 12. Average subjective scores. |
|--------------------------------------|
| CL | AD | Tel-YoMe | Tel | OpS | OpRN | MPC* | exMPC | ZyRo | ZyRo* |
| SS | 2.24 | 2.35 | 2.24 | 2.12 | 2.06 | 2.06 | 2.24 | 2.82 | 2.47 | 2.18 |
| SF | 0.5 | 0.5 | 0.42 | 0.42 | 0.42 | 0.42 | 1 | 0.58 | 0.4 | 0.58 |

| Preliminarily subjective validation |
|-------------------------------------|
| The objective criteria normally provide the first estimation of the simulated motion fidelity regarding the predicted false cues. Considering the human perception, the second phase test of simulated motion with the real passengers was implemented in the next-step on the motion platform. The preliminary subjective tests are important to validate the correlation between objective and subjective measures. Similar tests were performed in Berthoz et al.\textsuperscript{23} to investigate the effect of the washout-filters. The test was implemented with a total of 17 participants including 12 males and 5 females who have the age in range 20–30 years. The participants were asked to rate the 10 rides of case 2 on five satisfying levels in the range \{very good, good, satisfactory, sufficient, not sufficient\} corresponding to the grade scale \{1, 2, 3, 4, 5\} with ±0.3 as shown in the Table 11. |
This scaling based on German school grading marks was used due to well-perceived by all test subjects and no more guide was needed. A participant successively tried 10 rides in arbitrary order and asked to give his/her evaluation after each ride. The interclass correlation for rates of 17 judges is $\text{ICC}(3,k) = 0.4937$ with $p = 0.0462$—probability of false-rejection—($k = 17$ according to Berthoz et al.\textsuperscript{23}). The ICC values express the fair reliability of the rates.

The individual subjective ratings (SS) and the scale factor (SF) of simulated specific forces are shown in Table 12. For the graphical representation, the “bubble plot” (Figure 12) uses the circle size to describe the number of rates for individual subjective scores which are fairly variable in the full scale 1–5, in general. However, there are also some abnormal rates for such worst ride (exMPC algorithm) that one participant gives the best score as “very good.” The abnormal scores are controversial and could be happened because the participants have the particular threshold of rotational motion cue.

In order to compare the objective perception ratings with the subjective ratings, the partial components of the objective ratings were fitted by linear regression with the average subjective measures for the linear regression

$$F = w_0 + w_{\lambda_1^*}\lambda_1^* + w_{\lambda_2^*}\lambda_2^* + w_{J_\omega}J_\omega$$

![Figure 12. Individual subjective and objective score.](image-url)
with the regress function in MATLAB. In order to have an idea of the impact of each term, we switched on and off each term for all possible combinations. The results are shown in the Table 13, which displays the root mean square error (RMSE) and Pearson correlation (CORR) for each combination of terms. One can see that both $\lambda_1$ and $\lambda_2$ have a bad correlation with the average subjective measure and that even the best fit of $\lambda_1$ nor $\lambda_2$ together yield only a fair correlation. However, by adding the penalty term $J_\omega$, this term alone gives a better correlation with the average subjective measure than the best of $\lambda_1$ and $\lambda_2$, which is improved only slightly if the aforementioned two are added to the linear regression. This shows that false angular velocity cues have a significant impact on motion perception quality.

In a second round, we split the linear regression model even finer by taking into account all partial components of the “good criterion” as

$$ F = \sum_{q \in \{1, 2\}} \sum_{p \in \{f, \omega\}} \sum_{n \in \{se, sh\}} w_{qp,n} \cdot \lambda_{qp,n} + w_{J_\omega} J_\omega $$  \hspace{1cm} (36)

The coefficients $w_{qp,n}$ are listed in the Table 14. One can see that applying the suitable fitting coefficients to the “Good criterion” components leads very good correlations with the average subjective score achieved. This is even further optimized when including the penalty term $J_\omega$, yielding Pearson correlation 95.42% and a

Table 13. Coefficient of numerical expected functions \{F_1, \cdots, F_7\}.

|   | $w_0$ | $w_{\lambda_1}$ | $w_{\lambda_2}$ | $w_{J_\omega}$ | RMSE  | CORR  |
|---|-------|-----------------|-----------------|----------------|--------|--------|
| $F_1$ | 2.2716 | 0.0075 | 0 | 0 | 0.2167 | 0.0056 |
| $F_2$ | 2.4314 | 0 | -0.2199 | 0 | 0.2091 | 0.2624 |
| $F_3$ | 1.8270 | 1.2780 | -0.9127 | 0 | 0.1825 | 0.5396 |
| $F_4$ | 2.2076 | 0 | 0 | 0.6145 | 0.1184 | 0.8378 |
| $F_5$ | 2.1985 | 0.0107 | 0 | 0.6145 | 0.1183 | 0.8378 |
| $F_6$ | 2.2100 | 0 | -0.0032 | 0.6136 | 0.1183 | 0.8378 |
| $F_7$ | 2.1722 | 0.0910 | -0.0577 | 0.5990 | 0.1182 | 0.8382 |

Figure 13. MCA ranking by average subjective and numerical score.
root mean square error of 0.0660. By plotting the thus optimized numerical scores in the individual/average subjective rating plot of Figure 12 (red crosses). It could be noted that the numerical scores track much better the average subjective score than the individual ones. This is seen also in Figure 13, which shows the ranking of the 10 MCA both in terms of average subjective and objective measures.

The preliminary results thus show that (1) the individual objective parameters $\lambda_1$ and $\lambda_2$ seem not to correlate well with average subjective ratings; (2) the violation of the false angular velocity cue threshold influences significantly the subjective ride perception; (3) after down-scaling target accelerations there are little perception differences for most MCAs; and (4) by proper linear regression of the partial components of the “good criterion” together with the penalty terms for false cue angular velocity a very good correlation between average subjective and objective measures can be achieved. However, further research is necessary to better understand the factors impacting on subjective ratings of a simulator task because the individual scores are not consistent and have fair reliability.

Conclusions

The present paper first reviews the families of offline MCAs from the literature including the classical washout, adaptive washout, optimal washout, optimal control, and model-predictive control MCAs. Then, the MCAs were implemented in MATLAB for virtual rides over a planar S-shaped curve with lateral acceleration only. Next, the comparison of their behavior and the performance evaluation are carried out using both “performance indicator” and “good criterion” in which the deviation of the simulated and target motion cues (specific forces and angular velocities) are estimated. The comparison of different simulated responses in the case 1 shows that the optimal tracking control approach (ZyRo algorithm) with the tuning parameters yields the best results in tracking the target signals. The ZyRo and MPC algorithm exploit better workspace of the simulator. In the comparison of case 2 and the preliminary subjective evaluation, because of the clear disagreement of current numerical criteria and subjective evaluation, the proposed numerical criteria are built as the linearized function of performance indexes or sub-indexes. The penalty function $J_{w_r}$, inserted in some functions to increase the correlation of numerical score w.r.t subjective score, shows the significant effect of false cues of angular velocity to the motion fidelity. The function $F_9$ has the best correlation and could be used as the numerical measure for the numerical prediction of

### Table 14. Coefficient of numerical expected functions $\{F_8, F_9\}$.

|        | $w_{1f,sc}$ | $w_{2f,sc}$ | $w_{1f,sh}$ | $w_{2f,sh}$ | $w_{1w,sh}$ | $w_{2w,sh}$ | $w_{j_v}$ | RMSE | CORR |
|--------|-------------|-------------|-------------|-------------|-------------|-------------|-----------|------|------|
| $F_8$  | 14.6490     | -13.7195    | -0.9510     | 2.6956      | 0.1424      | -0.2323     | 0         | 0.0788| 0.9352|
| $F_9$  | 9.1670      | -7.9579     | -0.2280     | 1.8073      | 1.9510      | -2.1366     | 1.4126    | 0.0660| 0.9542|

|
sensations. In this paper, only the numerical measures of the simulated cues, that are generated by motion cueing algorithms, and the preliminary results with 17 test-subjects have been addressed. The proposed numerical criteria can also be utilized to evaluate the MCA response’s performance quality throughout the parameter tuning process, reducing the amount of time spent on trial and error.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2020-SAHEP-013

ORCID iD

Duc-Toan Nguyen https://orcid.org/0000-0001-9619-4476

References

1. Giordano PR, Masone C, Tesch J, et al. A novel framework for closed-loop robotic motion simulation. Part II: Motion cueing design and experimental validation. In: 2010 IEEE international conference on robotics and automation (ICRA), Anchorage, AK, 3–7 May 2010, pp.3896–3903. New York, NY: IEEE.
2. Conrad B and Schmidt SF. Motion drive signals for piloted flight simulators. Washington, DC: National Aeronautics and Space Administration, 1970.
3. Parrish RV, Dieudonne JE and Martin DJ Jr. Coordinated adaptive washout for motion simulators. J Aircr 1975; 12: 44–50.
4. Reid LD and Nahon MA. (1985). Flight simulation motion-base drive algorithms: Part 1. Developing and testing the equations (Rep. No. 296). University of Toronto: UTIAS. https://repository.tudelft.nl/islandora/object/uuid:45b071c0-0568-4e8f-948f-dfa52d350665
5. Schweig S and Kammers H. Bewegungssteuerung eines RoboCoaster Kuka Roboters zur Ride Simulation mit Hilfe von Washout Filtern. Project Report. Duisburg, Germany: University of Duisburg Essen, 2011.
6. Nahon MA and Reid LD. Simulator motion-drive algorithms — A designer’s perspective. J Guid Control Dyn 1990; 13: 356–362.
7. Sivan R, Ish-Shalom J and Huang JK. An optimal control approach to the design of moving flight simulators. IEEE Trans Syst Man Cybern 1982; 12: 818–827.
8. Telban RJ and Cardullo FM. Motion cueing algorithm development: human-centered linear and nonlinear approaches. NASA technical report no. CR-2005-213747, May 2005, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050180246.pdf
9. Zywol HJ Jr and Romano R. Motion drive algorithms and simulator design to study motion effects on infantry soldiers. In: DSC North America 2003, Dearborn, MI, 2003. https://apps.dtic.mil/sti/citations/ADA572240

Pham and Nguyen
10. Augusto BDC and Loureiro RJL. Motion cueing in the Chalmers driving simulator: a model predictive control approach. MSc Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2009. https://odr.chalmers.se/bitstream/20.500.12380/98871/1/98871.pdf

11. Baseggio M, Beghi A, Bruschetta M, et al. An MPC approach to the design of motion cueing algorithms for driving simulators. In: 2011 14th international IEEE conference on intelligent transportation systems (ITSC), Washington, DC, 5–7 October 2011, pp.692–697. New York, NY: IEEE.

12. Fang Z and Kemeny A. Motion cueing algorithms for a real-time automobile driving simulator. In: Proceedings of the driving simulation conference, Paris, France, 2012, pp.159–174. Paris: Arts et Metiers Paristech.

13. Khusro YR, Zheng Y, Grottoli M, et al. MPC-based motion-cueing algorithm for a 6-DOF driving simulator with actuator constraints. Vehicles 2020; 2: 625–647.

14. Katliar M, Fischer J, Frison G, et al. Nonlinear model predictive control of a cable-robot-based motion simulator. IFAC-papersonline 2017; 50: 9833–9839.

15. Katliar M, Olvari M, Drop FM, et al. Offline motion simulation framework: optimizing motion simulator trajectories and parameters. Transp Res Part F Traffic Psychol Behav 2019; 66: 29–46.

16. Katliar M, Drop FM, Teuffel H, et al. Real-time nonlinear model predictive control of a motion simulator based on a 8-DOF serial robot. In: 2018 European control conference (ECC), Limassol, Cyprus, 12–15 June 2018, pp.1529–1535. New York, NY: IEEE.

17. Mohammad A, Asadi H, Mohamed S, et al. MPC-based motion cueing algorithm with short prediction horizon using exponential weighting. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC), Budapest, Hungary, 9–12 October 2016, pp.000521–000526. New York, NY: IEEE.

18. Asadi H, Mohamed S, and Nelson K. An optimal washout filter based on genetic algorithm compensators for improving simulator driver perception. In: Proceedings of the driving simulation conference & exhibition DSC 2015, Tübingen, 16–18 September 2015, pp.173–182. Munich: Max Planck Institute for the Advancement of Science.

19. Pouliot NA, Gosselin CM and Nahon MA. Motion simulation capabilities of three-degree-of-freedom flight simulators. J Aircr 1998; 35: 9–17.

20. Fischer M. Motion-Cueing-Algorithmen für eine realitätsnahe Bewegungssimulation. PhD Thesis, Deutsches Zentrum für Luft und Raumfahrt in der Helmholtz Gemeinschaft, DLR, 2009.

21. Pham DA, Röttgermann S, Flores FG, et al. Optimal motion cueing algorithm selection and parameter tuning for sickness-free robocoaster ride simulations. In: Corves B, Lovasz EC and Hüsing M (eds) Mechanisms, Transmissions and Applications. Vol. 31. Berlin: Springer, 2015, pp.127–135.

22. Borah J, Young LR and Curry RE. Optimal estimator model for human spatial orientation. Ann N Y Acad Sci 1988; 545: 51–73.

23. Berthoz A, Bles W, Bülthoff HH, et al. Motion scaling for high-performance driving simulators. IEEE Trans Hum Mach Syst 2013; 43: 265–276.

24. Advani SK. The kinematic design of flight simulator motion bases. PhD Dissertation. Delft University of Technology, Delft, The Netherlands, 1998.

25. Shrout PE and Fleiss JL. Intraclass correlations: uses in assessing rater reliability. Psychol Bull 1979; 86: 420–428.

26. Qazani MRC, Jalali SMJ, Asadi H, et al. Optimising control and prediction horizons of a model predictive control-based motion cueing algorithm using butterfly optimization
algorithm. In: 2020 IEEE congress on evolutionary computation (CEC), Glasgow, UK, 19–24 July 2020, pp.1–8. New York, NY: IEEE.

27. Mohammadi, A, Asadi, SH, Nelson, K. Future reference prediction in model predictive control based driving simulators. In: Proceedings of the Australasian conference on robotics and automation (ACRA2016), Brisbane, QLD, Australia, 5 -6 December 2016. Brisbane, QLD, Australia: ACRA. https://www.araa.asn.au/acra/acra2016/papers/pap109s1.pdf

28. Grottoli M, Cleij D, Pretto P, et al. Objective evaluation of prediction strategies for optimization-based motion cueing. Simulation 2019; 95: 707–724.

29. Asadi H, Mohamed S, Lim CP, et al. Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm. IEEE Trans Syst Man Cybern Syst 2016; 47: 238–254.

30. Asadi H, Mohamed S and Nahavandi S. Incorporating human perception with the motion washout filter using fuzzy logic control. IEEE ASME Trans Mechatron 2015; 20: 3276–3284.

Author biographies

Duc-An Pham, Doctor in the School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam. He got a Bachelor degree in the Talented Engineering Programme of Mechatronics and a Master degree in Mechanics Engineering at Hanoi University of Science and Technology, Vietnam. Then he graduated PhD degree in Mechatronics and Robotics at Duisburg-Essen University, Germany. His research interests are Machining Process, CAD/CAM/CAE, Robotics, Motion cueing algorithm, Driving simulators and Auto-tuning method.

Duc-Toan Nguyen, Professor in the School of Mechanical Engineering, Hanoi University of Science and Technology (HUST), Vietnam. He was educated at the Hanoi University of Technology for his Engineer and Master’s degree in the area of Mechanical Engineering and graduated Ph. D. at the Kyungpook National University. Then, he pursued his professional career at the leading university of Vietnam (HUST), where he undertook research related to the Plasticity of Material, Machining Processes, CAD/CAM/CAE, and Robotics. Currently, he serves as a Deputy Head of Metal Machining and Industrial instrument. He is also serving as the Vice-president of the Vietnam Association of Science Editing (VASE).