

RESEARCH ARTICLE

Investigation of vibration's effect on driver in optimal motion cueing algorithm

Hazoor Ahmad¹, Muhammad Tariq², Awais Yasin³, Sohail Razaq⁴, Muhammad Ahmad Chaudhry⁵, Inam Ul Hasan Shaikh⁶, Ahsan Ali⁵, Saeed Mian Qaisar^{6,7}, Jamshed Iqbal^{8*}

1 Department of Electrical Engineering, Information Technology University (ITU), Lahore, Pakistan, **2** Alternate Energy Research & Innovation Lab (AERIL), Al-Khwarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan, **3** Computer Engineering Department, National University of Technology (NUTech), Islamabad, Pakistan, **4** Faculty of Information Technology, Majan University College, Muscat, Sultanate of Oman, **5** Department of Electrical Engineering, University of Engineering and Technology (UET), Taxila, Pakistan, **6** Electrical and Computer Engineering Department, Effat University, Jeddah, Saudi Arabia, **7** CESI LINEACT, Lyon, France, **8** School of Computer Science, Faculty of Science and Engineering, University of Hull, Hull, United Kingdom

* j.iqbal@hull.ac.uk



OPEN ACCESS

Citation: Ahmad H, Tariq M, Yasin A, Razaq S, Chaudhry MA, Shaikh IUH, et al. (2023) Investigation of vibration's effect on driver in optimal motion cueing algorithm. PLoS ONE 18(11): e0290705. <https://doi.org/10.1371/journal.pone.0290705>

Editor: Mohd Nadhir Ab Wahab, Universiti Sains Malaysia, MALAYSIA

Received: April 18, 2022

Accepted: August 15, 2023

Published: November 30, 2023

Copyright: © 2023 Ahmad et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abbreviations: α_{lat} , Lateral acceleration; α_{long} , Longitudinal acceleration; δ , Orientation of moving platform in screw axis notation; θ , Angle of screw axis in screw axis notation; $\hat{f}_{x(s)}$, Sensation signal in

Abstract

The increased sensation error between the surroundings and the driver is a major problem in driving simulators, resulting in unrealistic motion cues. Intelligent control schemes have to be developed to provide realistic motion cues to the driver. The driver's body model incorporates the effects of vibrations on the driver's health, comfort, perception, and motion sickness, and most of the current research on motion cueing has not considered these factors. This article proposes a novel optimal motion cueing algorithm that utilizes the driver's body model in conjunction with the driver's perception model to minimize the sensation error. Moreover, this article employs H_{∞} control in place of the linear quadratic regulator to optimize the quadratic cost function of sensation error. As compared to state of the art, we achieve decreased sensation error in terms of small root-mean-square difference (70%, 61%, and 84% decrease in case of longitudinal acceleration, lateral acceleration, and yaw velocity, respectively) and improved coefficient of cross-correlation (3% and 1% increase in case of longitudinal and lateral acceleration, respectively).

Introduction

Evaluation of Virtual Reality in driving and on-road field studies have shown encouraging resemblances [1, 2]. However, when compared to natural world driving, questions are raised on the validity of driving simulators. It will not be easy to perceive a driver's behavior in such an environment. Driving simulators are considered very important to improve road safety through proper indoor training and improved vehicle design [3]. Their use is crucial when the number of vehicles is considerably large, and road structure is also not very appropriate. Research has shown that a virtually trained driver saves money, time, life, and material. Nowadays, driving simulators are used not only as training simulators but also as entertainment

case of linear motion; ${}^A P$, Position of moving platform; ${}^A a_i$, Position of A_i with respect to O_A ; ${}^A b_i$, Position of B_i with respect to O_A ; ${}^B b_i$, Position of B_i with respect to O_B ; A_i , i spherical joint at the base platform of Stewart platform; a_x , Linear acceleration along the surge; B_i , i spherical joint at moving platform of Stewart platform; d_k , Search direction identifier in Gauss–Newton method; dof , Degrees of freedom of robot; e , Error between sensation signal and vehicle signal; $f_x(s)$, Force per unit mass along surge; G_s , Transfer function for acceleration transition; G_T , Transfer function for velocity transition; G_Ω , Ratio of vestibular organ to rotational velocity; G_p , Ratio of otolith organ to the specific force; G_{BL} , Band-limiting filter; G_{HP} , High-pass filter; G_{LP} , Low-pass filter; J , Quadratic cost function; j , No. of joints of a robot; L , Vector of given limb lengths; l_i , i limb length of i kinematic chain of Stewart platform; n_L , No. of links of a robot; O_A , Coordinate frame attached to base platform; O_B , Coordinate frame attached to moving platform; P_i , i prismatic joint between fixed and moving platform; p_x , x -coordinate of position of moving platform; p_y , y -coordinate of position of moving platform; p_z , z -coordinate of position of moving platform; Q , Weighting matrix for error e ; R , Weighting matrix for input signal u_s ; t , Vector of position and orientation in screw axis notation; $u_s(s)$, Input to driver's body model; $u_v(s)$, Input to simulator; v_{yaw} , Yaw velocity; $W(s)$, Transfer function for optimal motion cueing; λ_k , Search direction identifier in Levenberg–Marquardt Method; Ω_p , Angular velocity or derivative of pitch; $\hat{\Omega}_p(s)$, Sensation signal in case of angular motion; dof_i , Degrees of freedom of i joint; dof_p , Degrees of freedom of passive mechanism; dof_s , Degrees of freedom of space; ${}^A R$, Orientation of moving platform in rotation matrix notation.

simulators. Researchers have evaluated drivers' behavior, performance, and attention. In the automobile industry, they are used for optimal and risk-free vehicle design, resulting in reduced vehicle cost and increased safety. We can have efficient hardware implementations of driving simulators using energy efficiency techniques, such as SuperSlash [4], which is used for energy efficiency of artificial intelligence (AI) based models.

The motion cueing algorithm (MCA) converts linear accelerations and angular velocities of a virtual vehicle into movements of a driving simulator. One of the first methods to transform these accelerations and velocities was the classical MCA proposed by Conrad et al. [5]. Classical MCA is a linear cueing algorithm that is simple and easily tunable and has been widely used in different types of simulators [1, 6]. Classical motion cueing has a short processing time and stable performance [7]. However, classical MCA has many drawbacks. For example, it needs a regulation procedure that emphasizes the worst-case situation resulting in conventional motion with deprived workspace usage, hence not being flexible in all situations. Moreover, it completely ignores the human vestibular system [8] and may generate false cues, distort the signal, and lead to wrong motion signals due to filter characteristics [9]. Also, a low-pass filter leads to phase delay [10]. The coordinated adaptive washout algorithm has been applied to flight simulators in [11]. The authors replaced the tilt coordination block of the classical washout filter with an adaptive filter to avoid residual motion base displacements. In [12], adaptive motion cueing using fuzzy-based tilt coordination has been proposed. Due to many other possibilities of tilt coordination [13], the study of this algorithm is just at its initial stages.

An optimal MCA was proposed ignoring the vestibular system [14]. Optimal MCA has some drawbacks, but it generally reduces the sensation error between the virtual surrounding and real drivers. However, the results are not satisfactory to a real human perception due to physical limitations and an optimal washout filter needs to be applied to a larger workspace [15]. Getting realistic motion cues on a six-degree-of-freedom Stewart platform compared to the actual vehicle is still a problem. In some cases, the motion system has to be effectively turned off to avoid improper motion cues. A robust optimal MCA based on the linear quadratic regulator (LQR) method and the genetic algorithm (GA) has been studied in [16]. The driver feedback has been used in [17] to characterize desirable features of race cars; then the numerical optimal control (NOC) was used to decide platform motions. NOC is a useful open-loop method for analyzing performance constraints of the driver in loop strategies [17]. Other methods, including LQR-based optimal control and GA-based optimal control [16], exhibit better performance in improving the sensation output, producing realistic motion cues but cannot enhance workspace usage due to several constraints. The LQR-based method improves the workspace, but it has more sensation error and the GA-based method decreases error but with poor workspace usage [16]. Recently, many methods have introduced neural networks and computational intelligence-based motion cueing [18, 19]; however, these techniques are computationally complex.

This research utilizes H_∞ -based optimal MCA, which shows superior performance compared to the state-of-the-art methods. This article has the following main contributions:

1. A novel optimal MCA is proposed that utilizes the driver's body model together with the driver's perception model.
2. The LQR is replaced with an H_∞ -based controller to optimize the quadratic cost function of sensation error.

A significant advantage of using H_∞ -based MCA is that it can be used for any driving simulator without parameter tuning. We need to adjust the physical constraints of the new system and do not need to care about "trial and error" for parameter tuning.

The article is organized as follows: the Background section comprises the math behind the Stewart platform and its kinematics; the Proposed Methodology section includes the application of proposed optimal MCA on the Stewart platform; the Results and Discussion section compares different motion cueing techniques. Finally, the Conclusion section concludes the article by discussing the improvements made by the proposed methodology.

Background

In this section, we will discuss the mathematics behind the Stewart platform along with its kinematics. Fig 1 refers to the schematics of a generic Stewart platform. There are two parts to every limb length (l_i) connected by a prismatic joint (P_i). This limb's upper and lower sides are connected to the upper and lower platforms with spherical (or universal) joints. In this robot, six identical hydraulic actuators can be actuated to obtain six-degree-of-freedom motions of the moving platform. Six universal joints at the base platform of the Stewart platform are referred to as A_i , where $i = 1, 2, \dots 6$. Similarly, six spherical joints at the moving platform of the Stewart platform are referred to as B_i , where $i = 1, 2, \dots 6$. The prismatic joints between fixed and moving platforms are referred to as P_i , where $i = 1, 2, \dots 6$. The limb length of every kinematic chain of Stewart platform is referred to as l_i , where $i = 1, 2, \dots 6$. Two frames O_A and O_B are attached to the fixed and moving platforms, respectively.

Inverse kinematics

Inverse kinematics deals with the determination of limb lengths when the position and orientation of the moving platform are known [20–22]. In the case of a parallel robot, like a Stewart platform, inverse kinematics is easy [23]. The position AP of the moving platform and orientation \hat{s} are given, and limb lengths l_i s are to be determined. The position is given as follows:

$${}^A P = [p_x \ p_y \ p_z]^T. \tag{1}$$

In the case of screw axis representation, the orientation of the platform is given as follows:

$$\{\hat{s} = [s_x \ s_y \ s_z], \theta\}. \tag{2}$$

In the case of rotation matrix representation, the orientation of the platform is given as follows:

$${}^A R = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix}. \tag{3}$$

Using the loop closure equation for each limb on the geometry of Fig 1,

$$l_i {}^A \hat{s} = {}^A P + {}^A b_i - {}^A a_i, \tag{4}$$

where ${}^A b_i = {}^A R {}^B b_i$.

Eliminating \hat{s} from Eq (4),

$$l_i = \left[{}^A P^T A P + {}^B b_i^T B b_i + {}^A a_i^T A a_i - 2 {}^A P^T A a_i + 2 {}^A P^T [{}^A R {}^B b_i] - 2 [{}^A R {}^B b_i]^T A a_i \right]^{1/2}. \tag{5}$$

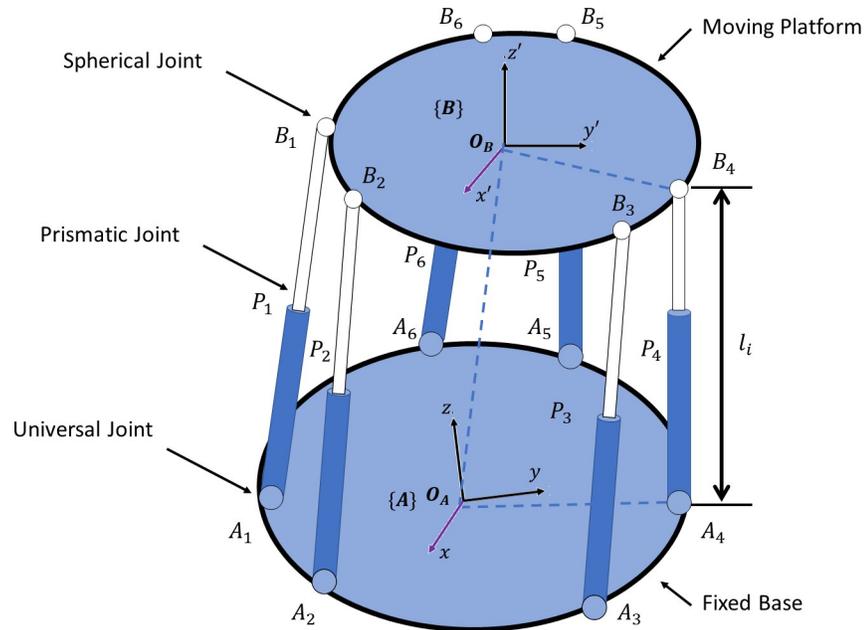


Fig 1. A generic Stewart platform with basic frames and parameters.

<https://doi.org/10.1371/journal.pone.0290705.g001>

The procedure in [17] will be followed to find actuator velocity. Taking the derivative of Eq (5) and simplifying for angular velocity, we get

$${}^A_B \dot{P} = \dot{l}_i {}^A \hat{s}_i - {}^A \dot{b}_i + {}^A \dot{a}_i \tag{6}$$

Forward kinematics

Forward kinematics is the method of determining the position and orientation of moving platforms when limb lengths of the Stewart platform are known. As shown in Fig 1, if vector L of limb lengths l_i , where $i = 1, 2, \dots, 6$, is known, then the problem is to determine position ${}^A P$ and orientation ${}^A_B R$.

The optimal solution method having many iterations needs high-speed computation due to the heavy computational burden [24]. A good initial guess has to be made to establish a rapid and converging solution. Forward kinematics can be obtained using nonlinear least-squares algorithms based on Levenberg–Marquardt or trust-region-reflective methods [23]. The trust-region-reflective method is based on the interior-reflective Newton method proposed in [25]. The Levenberg–Marquardt method has been proposed in [26, 27]. To apply one of these algorithms, the first step will be to merge Eqs (1) and (2):

$$t = [p_x \ p_y \ p_z \ s_x \ s_y \ s_z \ \theta] \tag{7}$$

or, equivalently,

$$t = [t_1 \ t_2 \ t_3 \ t_4 \ t_5 \ t_6 \ t_7]. \tag{8}$$

Position ${}^A P$ and orientation ${}^A_B R$ will be determined as follows:

$$P(t) = [t_1 \ t_2 \ t_3] \tag{9}$$

and

$$R(t) = \begin{bmatrix} t_4^2 vt_7 + ct_7 & t_4 t_5 vt_7 - t_6 st_7 & t_4 t_6 vt_7 + t_5 st_7 \\ t_5 t_4 vt_7 + t_6 st_7 & t_5^2 vt_7 + ct_7 & t_5 t_6 vt_7 - t_4 st_7 \\ t_6 t_4 vt_7 - t_5 st_7 & t_6 t_5 vt_7 + t_4 st_7 & t_6^2 vt_7 + ct_7 \end{bmatrix}, \tag{10}$$

where $vt_7 = (1 - \cos(t_7))$, $st_7 = \sin(t_7)$ and $ct_7 = \cos(t_7)$.

$$s(t) = [t_4 \ t_5 \ t_6 \ t_7]. \tag{11}$$

From Fig 1, the moving platform has points B_i 's and the base platform has points A_i 's and their vectors with respect to the base are b_i 's and a_i 's, respectively. The following set of nonlinear equations can be solved for the forward kinematics:

$$F_i(t) = -l_i^2 + [P(t) + R(t)b_i - a_i]^T [P(t) + R(t)b_i - a_i] \tag{12}$$

and

$$F_7(t) = \sum_{k=4}^6 t_k^2 - 1. \tag{13}$$

Optimization methods can be employed here to solve the above seven nonlinear equations (Eqs (12) and (13)). In case of least-squares optimization, minimize $f(t) = \frac{1}{2} \sum_i F_i(t)^2$, where $t \in \mathcal{R}$.

The solution of forward kinematics is demonstrated as a flowchart in Fig 2. In the first flowchart process, Eqs 9 and 10 are used for the calculation of p and R , respectively. Then, block F_i is evaluated using Eq (12). Now, $f(t)$ is determined as demonstrated above. The conditional block checks if this value is less than required accuracy ϵ or not. In case $f(t)$ is less than ϵ , the function iteratively calculates the new value of t and sends it back to calculate p and R , to recalculate the new values of p and R corresponding to the new value of t .

The input trajectories of position and orientation are shown in Fig 3. The calculated limb lengths of the Stewart platform are obtained using inverse kinematics, as shown in Fig 4. The method of forward kinematics using the Levenberg–Marquardt method is applied to get values of position and orientation back from values of limb lengths of Fig 4. These values are compared with the previous position and orientation with negligible error.

Motion cueing algorithm

Motion cueing refers to the approaches through which the environment of a driving simulator is made to feel like a real environment. These approaches are known as cues and are classified as acoustic, visual, and motion cues. A driving simulator of any class must contain an efficient visual system. Moreover, most visual systems are accompanied by acoustic systems. Motion simulators are required to provide motion cues sensed by vestibular, body, and perception system models. In the real environment, motion system parameter values exceed the limits of the driving simulator motion system. Therefore, it is essential to convert those parameters. The methods applied to perform this conversion are known as MCAs, motion drive algorithms, or washout filters. These algorithms aim to reproduce the motion cues in a realistic way, ensuring the usage of maximum workspace. The first was classical motion cueing built in [5] for flight simulators. Naturally, the requirements for a driving simulator are somewhat different; nonetheless, the classical algorithm can be adapted with some modifications.

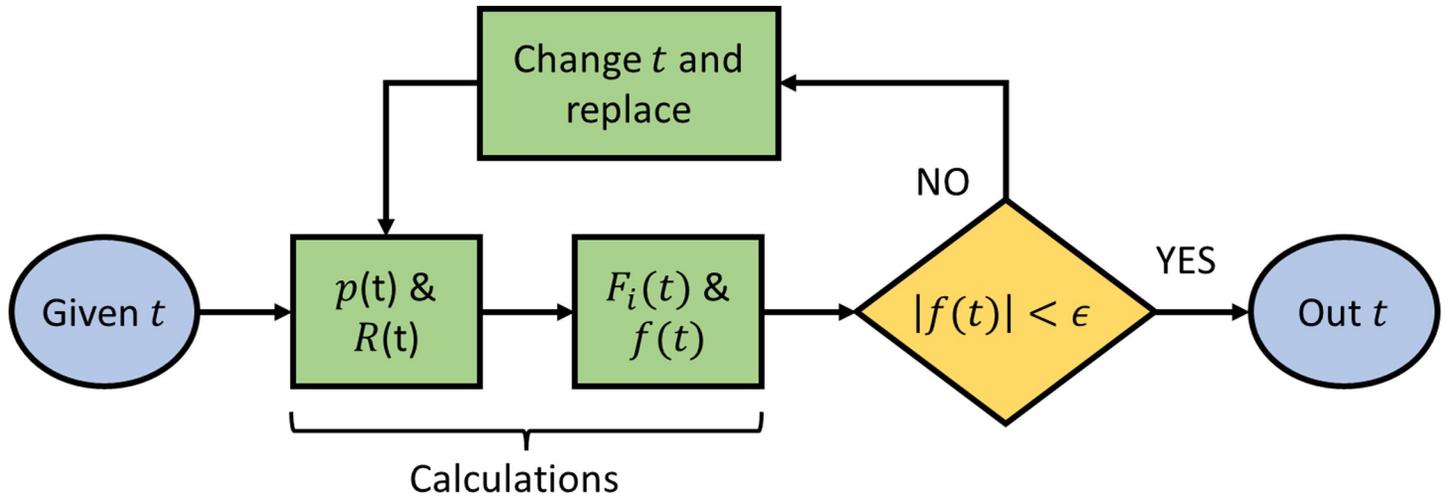


Fig 2. Iterative solution of forward kinematics.

<https://doi.org/10.1371/journal.pone.0290705.g002>

Proposed methodology

Optimal MCA for vehicle driving simulators that incorporate driver’s body and perception models is proposed so as to include human body vibrations [15, 28–30]. Human vibrations are of three types: the mass-spring-damper model [31], the model from the finite element method [32], and the transfer function model [14]. Methods for measuring periodic, random, and transient whole-body vibrations have been discussed in [28].

Fig 5 shows the basicentric coordinate system of a human body commonly used to illustrate human body vibrations. The driver’s body model is given by Eqs (14–19) as a transfer function that contains a band-limiting filter GBL in Eq (18). This filter contains high-pass filter G_{HP}

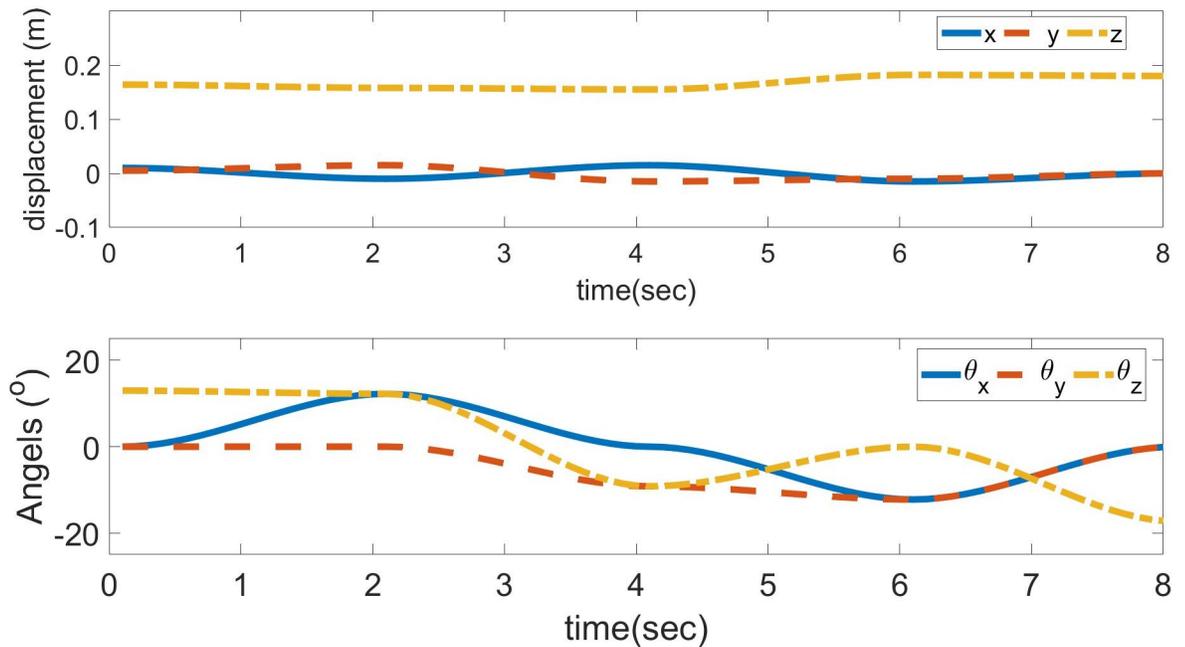


Fig 3. Supposed trajectories of position and orientation.

<https://doi.org/10.1371/journal.pone.0290705.g003>

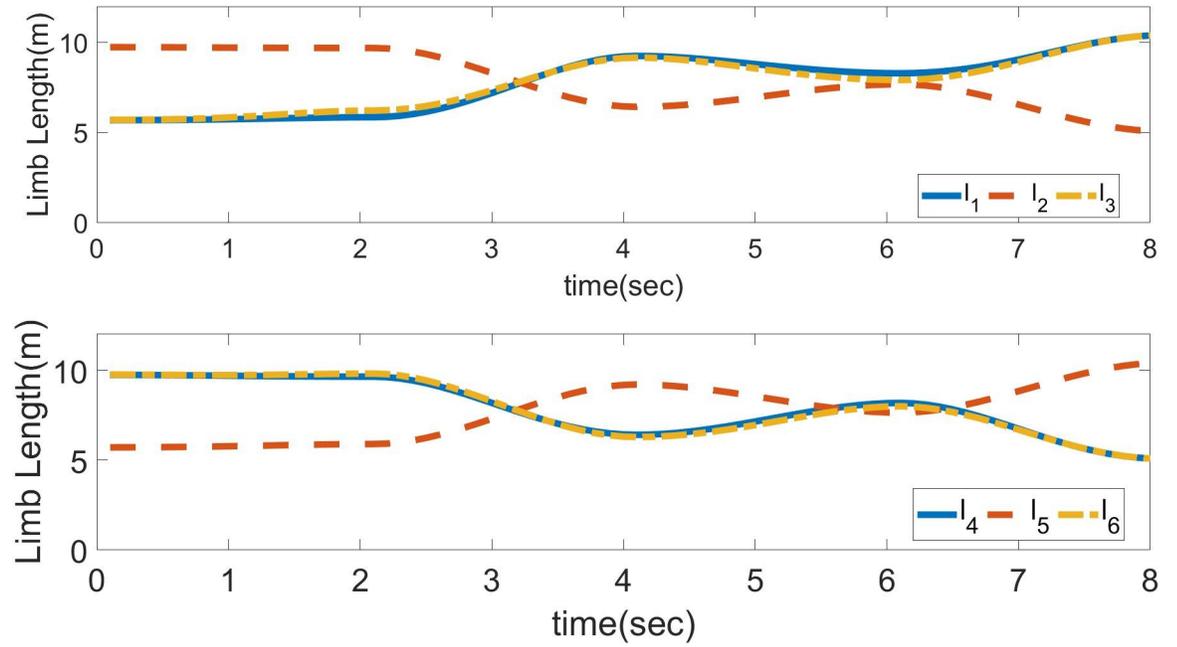


Fig 4. Calculated values of limb lengths.

<https://doi.org/10.1371/journal.pone.0290705.g004>

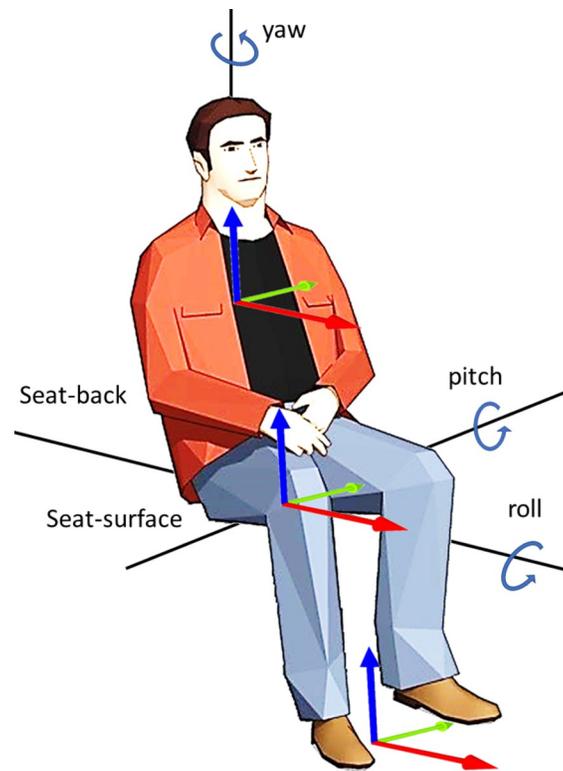


Fig 5. Basicentric coordinate system of the driver body (seated position).

<https://doi.org/10.1371/journal.pone.0290705.g005>

and low-pass filter G_{LP} :

$$G_{HP} = \frac{s^2}{s^2 + \alpha_1\sqrt{2}s + \alpha_1^2}, \tag{14}$$

$$G_{LP} = \frac{1}{s^2 + \alpha_2\sqrt{2}s + \alpha_2^2}, \tag{15}$$

$$G_{BL} = G_{HP} \times G_{LP}, \tag{16}$$

$$G_T = \frac{1 + \frac{s}{z_3}}{1 + \frac{s}{H_4 z_4} + \left(\frac{s}{z_4}\right)^2}, \tag{17}$$

$$G_s = \frac{1 + \frac{s}{H_5 z_5} + \left(\frac{s}{z_5}\right)^2}{1 + \frac{s}{H_6 z_6} + \left(\frac{s}{z_6}\right)^2} \times \left(\frac{\alpha_5}{\alpha_6}\right)^2, \tag{18}$$

$$G = G_{BL} \times G_T \times G_s, \tag{19}$$

where $a_i, i = 1, 2, 3, \dots, 6$ are the frequency weighted acceleration values and $H_i, i = 1, 2, 3, \dots, 6$ are the resonant quality factors. These factors evaluate the overall frequency weighting of the transfer function. The values of these variables are obtained from [28].

The driver’s perception model comprises vestibular and otolith organs. These organs are responsible for the perception of rotational velocity and specific force. Specific force is the difference between linear acceleration and gravity. Two transfer functions are shown in Eqs (20) and (21). The first transfer function is the ratio of the vestibular organ to that of rotational velocity:

$$G_{\Omega} = \frac{L_L L_A s^2}{(1 + L_L s)(1 + L_S s)(1 + L_A s)}. \tag{20}$$

The second transfer function is the ratio of the otolith organ to the specific force given as follows:

$$G_p = \frac{k(1 + \tau_A s)}{(1 + \tau_s s)(1 + \tau_L s)}, \tag{21}$$

where $L_L, L_A,$ and L_S are the weighting factors from vestibular organ and $\tau_L, \tau_A,$ and τ_s are weighting factors from the otolith organ.

Problem statement

The block diagram of a driving simulator, including an optimal motion cueing block $W(s)$, is shown in Fig 6.

The input to this block is $u_v(s)$ and its output is $u_s(s)$, which serves as input to the driver’s body model. The six motions of the moving platform are grouped into four modes as follows:

1. Longitudinal mode
2. Lateral mode

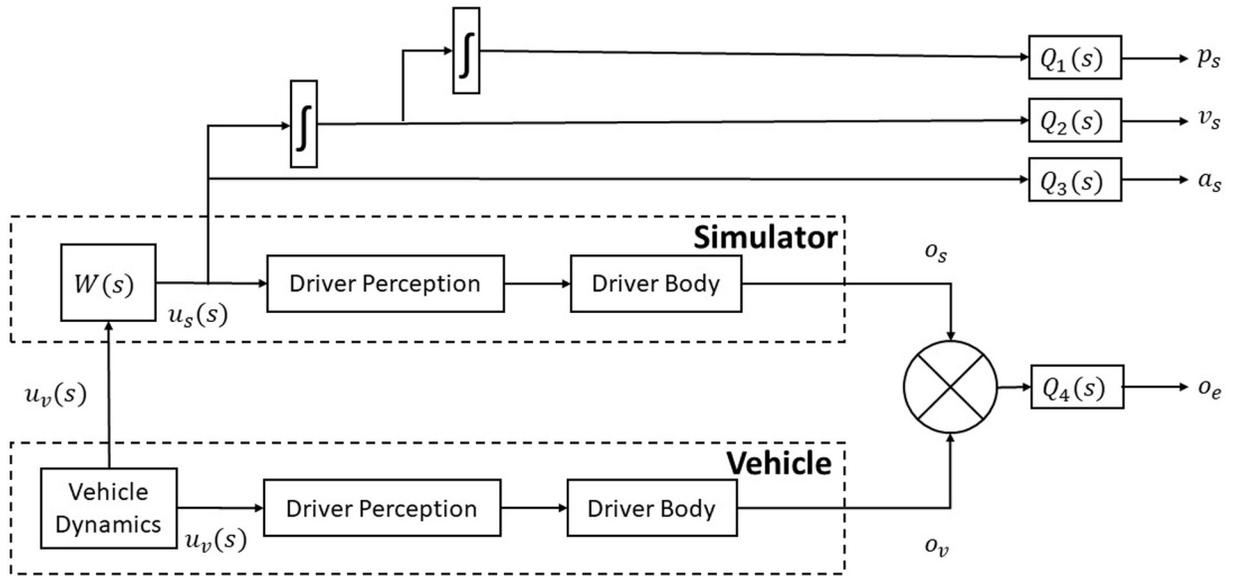


Fig 6. Proposed optimal cueing framework.

<https://doi.org/10.1371/journal.pone.0290705.g006>

- 3. Heave mode
- 4. Yaw mode

The longitudinal mode contains surge and pitch as linear and angular motion components, respectively. Sway has linear motion and corresponding roll as an angular motion component in the case of the lateral mode. Heave mode only contains heave as a linear motion component and yaw is the only component in yaw mode. In each mode, the state-space equation can be obtained from the transfer function of the driver’s body models. The following text includes the description of the algorithm of longitudinal mode, where Ω_p is the angular velocity or derivative of pitch and a_x is linear acceleration along with surge.

The structure of the proposed MCA is given in Fig 6. Eq (22) shows longitudinal components of input $u_v(s)$ to simulator block.

$$u_v(s) = \begin{bmatrix} \Omega_p \\ a_x \end{bmatrix} = \begin{bmatrix} u_{v1} \\ u_{v2} \end{bmatrix}. \tag{22}$$

The specific force $f_x(s)$ is defined as force per unit mass along surge and is given as follows:

$$f_x(s) = \begin{bmatrix} g & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{v1}/s \\ u_{v2} \end{bmatrix}. \tag{23}$$

The output of this cueing model is $u_s(s)$:

$$u_s(s) = \begin{bmatrix} \hat{\Omega}_p(s) \\ \hat{f}_x(s) \end{bmatrix} = W_s(s) \begin{bmatrix} u_{v1} \\ u_{v2} \end{bmatrix} = \begin{bmatrix} u_{s1} \\ u_{s2} \end{bmatrix}. \tag{24}$$

The problem is to determine a transfer function $W_s(s)$ that will be used to compute the output sensational signals for simulator moving platform position and orientation.

Determination of sensation signal. In optimal control theory, a cost function is to be optimized considering the motion system constraints [32–34]. In this case, the sensation error

between virtual surroundings and real drivers is minimized. Sensation error can be determined by the observable canonical form of state space. In the case of angular motion channel, Eqs (25) and (26) are used, and in the case of translational motion channel, Eqs (27) and (28) are used.

$$\dot{X}_p = A_p X_p + B_p u_v, \tag{25}$$

$$\hat{\Omega}_p = C_p X_p. \tag{26}$$

Similarly, using Eq (23),

$$\dot{X}_x = A_x X_x + B_x u_v, \tag{27}$$

$$\hat{f}_x = C_x X_x. \tag{28}$$

Combining Eqs (25)–(28) with Eqs (29) and (30),

$$\dot{X}_v = \begin{bmatrix} \dot{X}_p \\ \dot{X}_x \end{bmatrix} = \begin{bmatrix} A_p & 0_{9 \times 16} \\ 0_{9 \times 16} & A_x \end{bmatrix} \begin{bmatrix} X_p \\ X_x \end{bmatrix} + \begin{bmatrix} B_p \\ B_x \end{bmatrix} u_v = A_v X_v + B_v u_v, \tag{29}$$

$$\begin{bmatrix} \hat{\Omega}_p \\ \hat{f}_x \end{bmatrix} = \begin{bmatrix} C_p & 0 \\ 0 & C_x \end{bmatrix} \begin{bmatrix} X_p \\ X_x \end{bmatrix}. \tag{30}$$

If the input to driver’s body model is u_s , then similar equations are obtained:

$$\dot{X}_v = A_v X_v + B_v u_s, \tag{31}$$

$$\begin{bmatrix} \hat{\Omega}_p \\ \hat{f}_x \end{bmatrix} = C_v X_v. \tag{32}$$

Error dynamic equation can be obtained from Eqs (29)–(32) and is given as follows:

$$\dot{X}_e = A_v X_v + B_v u_s - B_v u_v, \tag{33}$$

$$e_v = C_v X_e. \tag{34}$$

It is seen in practical situations that the vehicle signal is accompanied by a filtered white noise signal, so it must be considered in the error signal. Eqs (35) and (36) show the state space of vehicle signal for noise considering w as filtered white noise input:

$$\dot{X}_n = A_n X_n + B_n w, \tag{35}$$

$$u_v = X_n. \tag{36}$$

Adding the effect of white noise to the error dynamic equation for the determination of cumulative error e ,

$$\dot{X} = \begin{bmatrix} \dot{X}_e \\ \dot{X}_n \end{bmatrix} = AX + Bu + Hw, \tag{37}$$

$$e = CX, \tag{38}$$

where

$$A_n = \begin{bmatrix} -1 & 0 \\ 0 & -4\pi \end{bmatrix}, B_n = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

$$A = \begin{bmatrix} A_v & -B_v \\ 0_{2 \times 25} & A_n \end{bmatrix}, B = \begin{bmatrix} B_v \\ 0_{2 \times 1} \end{bmatrix}, H = \begin{bmatrix} 0_{25 \times 1} \\ B_n \end{bmatrix}, C = [C_v \ 0].$$

The main objective of this article is to produce realistic motion cues, i.e., motion cues similar to the real vehicle. To achieve this objective, the error between the vehicle signal and simulator motion signal, given by Eq (38), must be minimized. In the last section error, the dynamic equation is modeled; in this section, this error is minimized using the appropriate choice of the optimization method. As explained in the motion cueing section, there are workspace limitations of the motion simulator. These limitations must be included in the optimal control problem to generate realistic motion cues. Objective and constraints of motion system can be combined into a quadratic cost function and the problem is formulated as follows:

Find value of u_s that minimizes J :

$$J = \int_{t_1}^{t_2} (e^T Q e + u_s^T R u_s) dt. \tag{39}$$

Subject to the following constraints,

$$l_{min} \leq l_i \leq l_{max}, \tag{40}$$

$$v_{min} \leq v_i \leq v_{max}, \tag{41}$$

$$a_{min} \leq a_i \leq a_{max}, \tag{42}$$

$$X_v = \begin{bmatrix} \hat{\Omega}_p \\ \hat{f}_x \end{bmatrix} = f(X_p, u_v), \tag{43}$$

$$u_s = g(u_v, w), \tag{44}$$

where J is the quadratic cost function to be minimized, Q and R are weighting matrices, and $l_i, v_i, a_i, i = 1, 2, 3, \dots, 6$ are limb length, velocity, and acceleration of every leg of the motion system, respectively.

The enhanced sensitivity to the sensation error. The problem is to decide the inputs u_s such that our quadratic cost function is minimized within the limits of constraints. Fig 6 shows the input vehicle dynamics u_v as a vector of longitudinal acceleration, lateral acceleration, and yaw velocity. A similar vector u_s is generated as simulator input. Both of these inputs are used

for the calculation of states of the system and outputs as discussed in Eq (30)–(38). The displacement, velocity and acceleration of the actuators are limited by the constraints of the cost function in Eq (40)–(42). The output error (o_e) is the function of these states:

$$y(t) = \begin{bmatrix} a_{REF_{long}}(t) - a_{s_{long}}(t) \\ a_{REF_{lat}}(t) - a_{s_{lat}}(t) \\ v_{REF_{yaw}}(t) - v_{s_{yaw}}(t) \end{bmatrix} \tag{45}$$

In short, there is a need of selecting such parameters which minimize the sensed longitudinal acceleration, lateral acceleration, and yaw velocity. The accelerations are translational whereas the velocity is a rotational parameter. However, this is a lengthy iterative process and requires repeated driver feedback. This work includes driver’s body model, to minimize the workload of drivers as shown in Eq (46) below:

$$y(t) = \begin{bmatrix} a_{REF_{long}}(t) - a_{s_{long}}(t) - a_{v_{long}}(t) \\ a_{REF_{lat}}(t) - a_{s_{lat}}(t) - a_{v_{lat}}(t) \\ v_{REF_{yaw}}(t) - v_{s_{yaw}}(t) - v_{v_{yaw}}(t) \end{bmatrix} \tag{46}$$

Analysis of cost function. We require to cue the brake system and the accelerate system while in longitudinal mode. For both situations we use numerical control techniques for error free acceleration profiling. The basicentric coordinate system of a human body in Fig 5 shows the direction of the longitudinal acceleration, lateral acceleration, and yaw velocity.

The reference (blue solid line) and the simulated (red dashed line) linear acceleration in Fig 7. The simulated wave is obtained by minimizing the time invariant objective function from Eq (44) with only minimizing the square of the sensation error. The platform must start

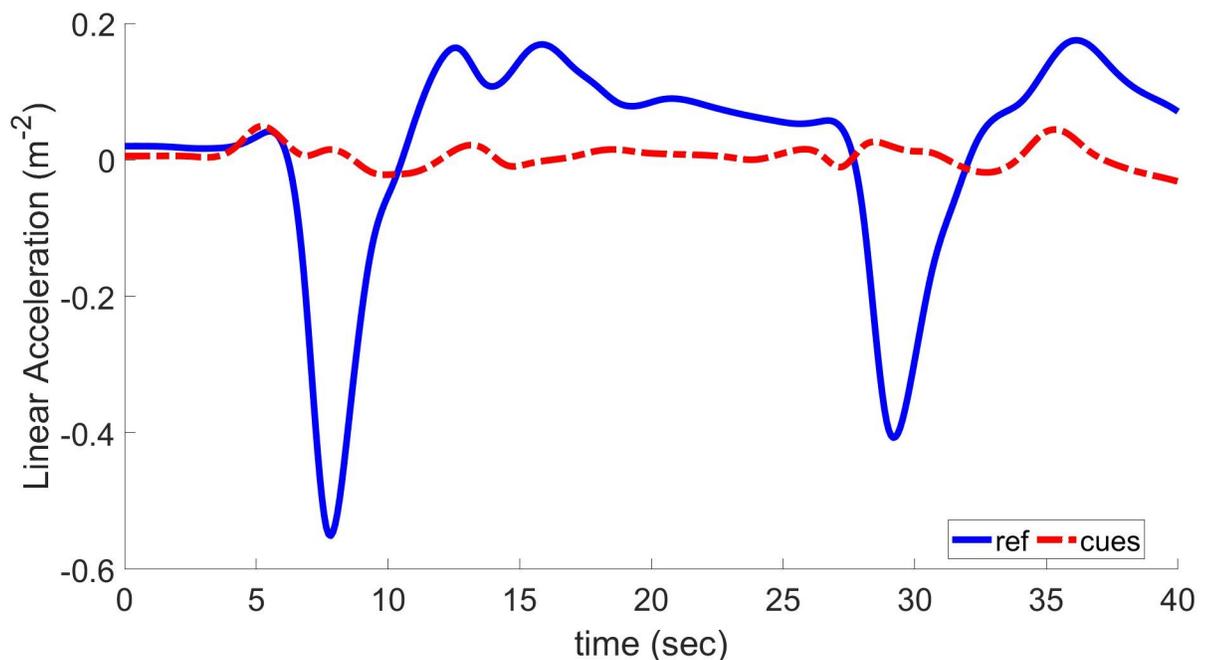


Fig 7. Longitudinal acceleration and resulting cues using linear quadratic regulator based on optimal MCA.

<https://doi.org/10.1371/journal.pone.0290705.g007>

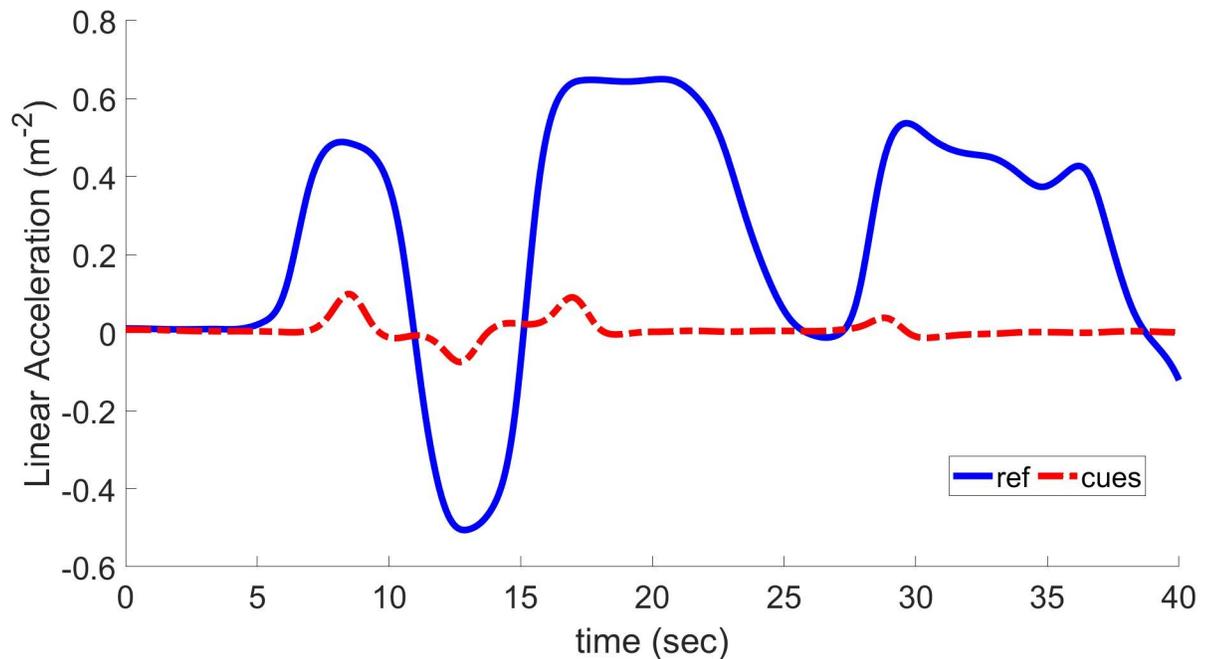


Fig 8. Lateral acceleration and resulting cues using linear quadratic regulator based on optimal MCA.

<https://doi.org/10.1371/journal.pone.0290705.g008>

the maneuver at the front of the workspace and must conclude it there for the optimum reaction to operate. As a result, the platform was set up for braking, and the maneuver concluded in an acceleration-ready posture. This cueing signal has two characteristics that need discussion. The first is a delayed reaction that causes the platform's peak deceleration to coincide with the cars. The platform comes to a stop near the end of the braking action due to a mistake made during workspace management, which is the motion's second distinguishing feature. This answer falls short of what the racing car drivers we contacted demand in terms of a strong early onset cue, hence it is not acceptable. Discussions led to the conclusion that drivers need a strong first onset cue followed by a mistake of the smallest feasible size. Since the cueing acceleration produced lacks these traits, it needed to be modified.

In a drawn-out high-speed turn, lateral acceleration signals can be of great amplitude and extended duration and have a substantially different form from the longitudinal case. Simulators with a constrained workspace cannot simulate this kind of acceleration. Again, the acceleration at the beginning is cued but not maintained. Since the yaw acceleration and the lateral cue have a similar shape, an objective function of the type Eq (44) is also utilized in this situation.

Results and discussion

A good motion cueing system should possess the following properties:

1. Realistic motion cues, i.e., the wave shape of output acceleration cues, should be more similar to that of input linear acceleration. Similarly, the wave shape of output angular velocity cues should be more similar to that of input angular velocity.
2. Usage of more workspace, i.e., the limits of theoretical workspace, should be achieved by optimum position and orientation of motion base.
3. It does not generate false cues.

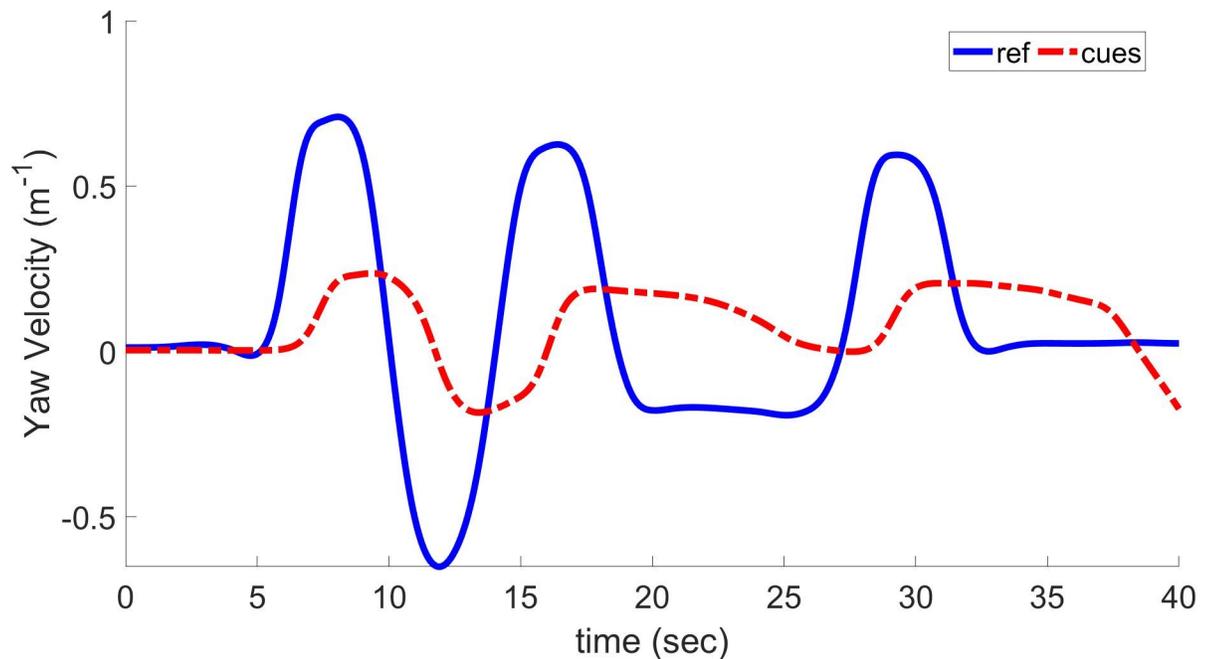


Fig 9. Yaw velocity and resulting cues using linear quadratic regulator based on optimal MCA.

<https://doi.org/10.1371/journal.pone.0290705.g009>

In this article, all the above three qualities have been practically achieved. Results are compared based on the root-mean-square difference (RMSD) and coefficient of cross-correlation (COCC) that show similarity between two waveforms to produce realistic motion cues. The following section includes the application of control strategies step by step for motion cueing and compares the results.

State-of-the-art motion cueing algorithms

Optimal MCA based on the linear quadratic regulator (LQR) is modified by changing the axis of rotation to the driver's body axis instead of the motion system's center of rotation [16, 22]. The main aim behind this change is to achieve realistic motion cues and remove false cues. Fig 5 shows the basicentric coordinate system of the driver body (seated position). The cross-coupling between the simulator and driver depends on the distance between their respective coordinate systems. As discussed in the motion cueing section, MCA receives linear acceleration and angular velocity as input to produce a proper cueing signal for the simulator motion base. Furthermore, a tilt rate limiter is introduced after cueing has been achieved for better tilt coordination. This addition investigates the effects on both motion cue fidelity and displacement.

The translational input has time values of longitudinal, lateral, and heave-specific forces. These inputs vary depending on acceleration, deceleration, and retardation, as shown in Figs 7 and 8. In these Figs., the blue plot shows values of input linear acceleration and illustrates that these values are never uniform; i.e., they keep on changing with time. According to Fig 7, longitudinal specific force values are applied to linear quadratic regulator-based optimal MCA, and resultant cues are plotted in a red dashed line. By thorough investigation of this plot, it can be observed that the coherence between waveforms is less guaranteed when specific force is changed quickly. Therefore, the LQR-based MCA can better track and follow the input signals during uniform conditions. The NOC-based technique [17] has a poor COCC of 0.8 and more RMSD of 0.455 than the LQR-based method, which has an improved COCC of 0.88 and

minimized RMSD to 0.218. This shows that the LQR method provides much better results in the form of generating realistic motion cues. RMSD should be decreased to zero to achieve realistic motion cues.

Fig 8. shows the comparison between normalized lateral specific force inputs and generated motion cues using the LQR-based optimal MCA. Like the previous case of longitudinal specific force, it is observed that the coherence between waveforms is less guaranteed when the specific force is changed quickly. Therefore, LQR-based MCA can better track and follow the input signals during uniform conditions. The NOC-based technique [17] has a lower COCC of 0.83 and RMSD of 0.196 than the LQR-based method, which has an improved COCC of 0.86 and increased RMSD to 0.221. This shows that the LQR method provides much better results in the form of generating realistic motion cues. Greater RMSD shows the doubtful performance of the algorithm during sudden changes of input like braking. To achieve realistic motion, a more precise MCA must be developed that minimizes RMSD to zero.

Fig 9 shows the comparison between yaw velocity and generated velocity cues based on the LQR-based optimal MCA. In contrast to the numerical optimal MCA, this method has an improved COCC of 0.89 and an angle of 3.22. The RMSD was minimized to 0.219. In the case of NOC, these values are 0.81, 2.23, and 0.417, respectively. This shows that LQR-based MCA can decrease sensation error between virtual surroundings and real drivers. The improvement is significant compared with the previous MCA, but it still needs to be improved.

We now compare two popular motion cueing methods based on LQR and GA. The LQR method discussed earlier in this section shows some improvements in RMSD and COCC when compared with the NOC strategy, but it still has undesirable RMSD. All methods are based on the mathematics derived in the motion cueing section, which employs human perception and the human body model to achieve desired results.

These methodologies improve sensation signals by minimizing RMSD and improving the COCC. Using more simulator workspace results in increased sensation error, but an optimal solution may result in the optimized workspace with RMSD as minimum as possible, which needs to have more constraints to be considered with relative simplicity in each constraint.

For GA implementation, some parameters have to be chosen, for example, population size, crossover rate, and mutation rate. By choosing these parameters, the selection is made between the degree of freedom of solution and speed of convergence of the GA. Thus, proper selection of crossover and mutation rate should be ensured to obtain the best results [16]. Depending on different applications, these parameters may vary as they are chosen by trial and error. In the case of a large mutation rate, the individual chromosomes may jump to the closer solution, whereas in the case of a small mutation rate, they may stick in local minima.

The recommended settings of these parameters are included in [22]. GA consists of the settings of crossover rates and mutation rates, which are satisfactory and meet the demands of the experimental study. Evolutionary operators are assigned to individual chromosomes that constitute the population. It should be noted that the search capacity of the GA is limited because of the small number of chromosomes in the population. If the number of chromosomes increases, the computational time also increases, not suitable for the proposed application. Therefore, an optimal value for population size should be selected to avoid the slow performance of the algorithm. From [7], the population size of fifty chromosomes is appropriate for acceptable results. If the number of chromosomes increases, it will not improve the solution considerably; however, it will significantly decrease computational speed and increase the time of convergence of the solution.

Alternatively, chromosomes are initialized based on the values of the LQR method, as discussed in the previous section. According to the literature [7], the parameter settings for GA

parameters are as follows: the crossover rate is 0.8, the mutation rate is 0.2, and half of the size of the chromosomes for individual regeneration.

As the population size tends to increase, the results become more satisfactory in terms of sensation error, but the computational speed tends to decrease accordingly. According to a convergence test on 20, 50, and 100 chromosomes in [16], the best fitness function value, human sensation error, and displacement are obtained using a population size of 50 chromosomes instead of 20. No fruitful results are achieved by increasing the population size from 50 chromosomes. Finally, GA has been successful in decreasing human sensation error and displacement and improving the COCC.

It must be kept in mind that the speed of computation for a population size of 100 chromosomes decreases significantly, resulting in an increased computational burden. When the population size is selected greater than 50, the speed of computation becomes too slow without significant improvement in fitness function value.

Computational time per iteration in the case of 20 chromosomes is 13.90s, whereas in the case of 50 chromosomes is 27.72 s and in the case of 100 chromosomes, it is 52.06 s [16]. GA-based optimal MCA stops searching at 455th iteration in case of 50 chromosomes. The density of genetic problems is based on several constraints, generations, and iterations per second, tends to increase with the increase in population size, and is less significantly affected using a population size of 50 compared to a population size of 100.

Fig 10 shows normalized values of reference longitudinal linear accelerations for comparing MPC-, LQR- and GA-based optimal MCAs with the proposed one. A range of longitudinal acceleration, deceleration, and retardation is applied to the LQR-based optimal MCA and GA-

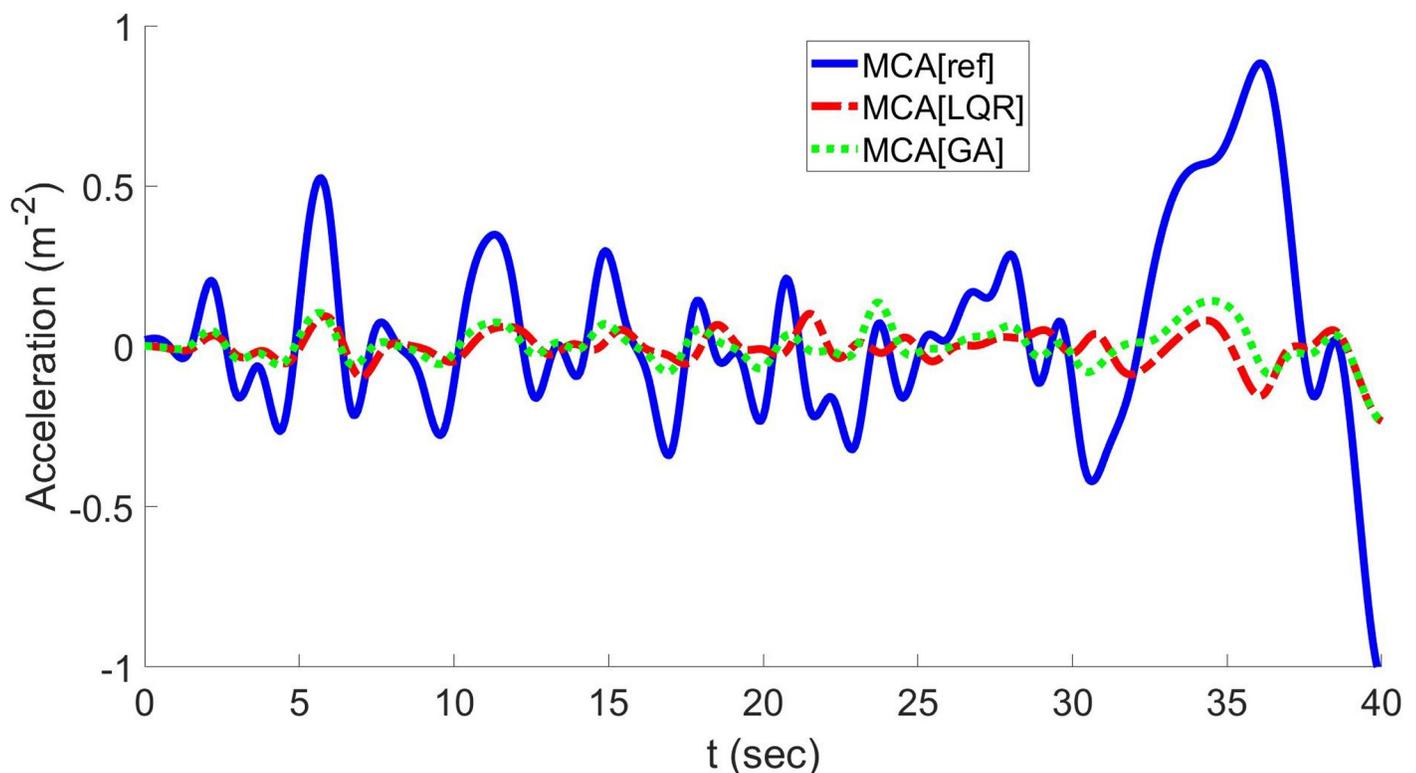


Fig 10. Plots of longitudinal acceleration and resulting cues show a comparison between the MCAs based on MPC, LQR MCA, GA, and the proposed one.

<https://doi.org/10.1371/journal.pone.0290705.g010>

Table 1. Comparison of RMSD.

| | A_{long} | a_{lat} | v_{yaw} |
|-----------------|--------------|--------------|--------------|
| NOC [17] | 0.455 | 0.196 | 0.417 |
| LQR [16] | 0.218 | 0.221 | 0.219 |
| GA [33] | 0.072 | 0.092 | 0.082 |
| OPT [35] | 0.335 | 0.271 | - |
| MPC [36] | 0.194 | 0.310 | - |
| Proposed | 0.022 | 0.036 | 0.013 |

<https://doi.org/10.1371/journal.pone.0290705.t001>

based optimal MCA. The resulting longitudinal acceleration cues are generated and plotted in Fig 10.

According to Fig 10, the GA-based optimal MCA shows better performance than the LQR method because the GA can track input-specific force with high accuracy. In addition, false cues are removed using the GA.

A comparison of RMSD and COCC between various algorithms is shown in Tables 1 and 2, respectively. The RMSD has been compared with NOC [17], LQR [16], GA [33], OPT [35], and MPC [36]. From the Table 1 it can be seen that the NOC behaves imperfectly in case of longitudinal acceleration and yaw velocity, whereas lateral acceleration is inefficient in case of MPC. This is because the NOC is more related to conventional techniques and MPC cannot better predict the model while in lateral direction. The proposed technique is well suited in all cases as it employs the extended sensation error. Similarly, while we see the coefficient of cross correlation in Table 2 for the sake of comparison between the above techniques, our results are comparable to the GA based approach in all cases of longitudinal & lateral acceleration and also in yaw velocity. This is because our methodology is more closely related to the OPT and GA, benefitting from both. Later we also plot these values of COCC in Fig 10 for the sake of visual comparison with MPC, LQR, GA based techniques.

According to Fig 10, the sensation error between virtual surroundings and the real driver has decreased in the case of the GA method compared to the LQR method. This error can be further decreased and taken below the human threshold using other techniques such as one utilized in the next subsection using H_{∞} control.

A comparison of maximum displacement and maximum angle between various algorithms is shown in Tables 3 and 4, respectively. As shown in Table 3, it is clear that displacement of the simulator in surge direction using GA-based optimal MCA significantly drops to 0.37, which was 0.44 in the case of LQR-based optimal MCA. However, the OPT based approach results least values of both surge and Heave. This can be due to the constraints applied by their methodology. Moreover, the yaw angle from Table 4 in the case of GA-based optimal MCA decreases to 2.04, which was 3.22 in the case of LQR-based optimal MCA. Hence, the GA-based optimal MCA reduces workspace usage by decreasing RMSD and improving COCC.

Table 2. Comparison of COCC.

| | A_{long} | a_{lat} | v_{yaw} |
|-----------------|-------------|-------------|-------------|
| NOC [17] | 0.80 | 0.83 | 0.81 |
| LQR [16] | 0.88 | 0.86 | 0.89 |
| GA [33] | 0.95 | 0.96 | 0.98 |
| OPT [35] | 0.90 | 0.92 | - |
| MPC [36] | 0.68 | - | - |
| Proposed | 0.98 | 0.97 | 0.98 |

<https://doi.org/10.1371/journal.pone.0290705.t002>

Table 3. Comparison of maximum displacement.

| | Surge (m) | Sway (m) | Heave (m) |
|-----------------|-------------|-------------|-------------|
| NOC [17] | 0.33 | 0.37 | 0.20 |
| LQR [16] | 0.44 | 0.45 | 0.21 |
| GA [33] | 0.37 | 0.39 | 0.18 |
| OPT [35] | 0.27 | - | 0.18 |
| Proposed | 0.50 | 0.49 | 0.22 |

<https://doi.org/10.1371/journal.pone.0290705.t003>

H_{∞} control-based optimal MCA

The human body model and human perception model have been integrated into the classical washout filter to observe the sensation error, the error between virtual surroundings and real driver. Driver's health comfort and motion sickness are significantly apparent if this error is more than a specified threshold. To minimize this error, hence, to produce realistic motion cues, optimal motion cueing is applied using different optimization methods, like LQR, GA, PSO, and H_{∞} .

All previous methods result in some acceptable improvement in error with less workspace usage. H_{∞} control strategies are helpful for their ready applicability to systems involving multiple variables with cross-coupling between channels. However, for an H_{∞} controller, it is difficult to optimize robust performance and stabilization simultaneously. However, many methods, known as extensions of this strategy, exist, such as H_{∞} loop-shaping, which permits the designer to apply classical loop-shaping concepts to the frequency response of the multi-variable system and achieve robust performance.

Then favorable robust stabilization is achieved by optimizing the system's response near bandwidth frequency. Here, the Riccati equation is reformulated using optimization and linear matrix inequalities with fewer constraints.

Figs 11–13 show the results of H_{∞} control-based optimal MCA in the case of lateral accelerations, longitudinal acceleration, and yaw velocities, respectively, compared to reference signals along with their cues.

The sensation error between the simulator and driver and the time derivative of the sensation reference signal is input to the H_{∞} control method. Then, the H_{∞} controller generates a supplementary reimbursing signal "Fi," which removes the sensation error between the simulator and driver. The human body has limited ability to enumerate motion applied to it, and the human vestibular system cannot differentiate between small, medium, or large motions in various directions. This motion is divided into logical perceptible ranges. Therefore, in this proposed H_{∞} control method, sensation error values have been divided into five groups: heavily positive, positive, zero, negative, and heavily negative. In this way workspace of the simulator is improved.

In the translational channel of the H_{∞} controller, both inputs are classified as the otolith organ and vestibular organ. The result of the high-pass filter is input to this block and then

Table 4. Comparison of maximum angle.

| | Roll (°) | Pitch (°) | Yaw (°) |
|-----------------|-------------|-------------|-------------|
| NOC [17] | 2.50 | 2.79 | 2.23 |
| LQR [16] | 3.50 | 2.98 | 3.22 |
| GA [33] | 0.37 | 0.39 | 0.18 |
| Proposed | 3.33 | 4.34 | 3.58 |

<https://doi.org/10.1371/journal.pone.0290705.t004>

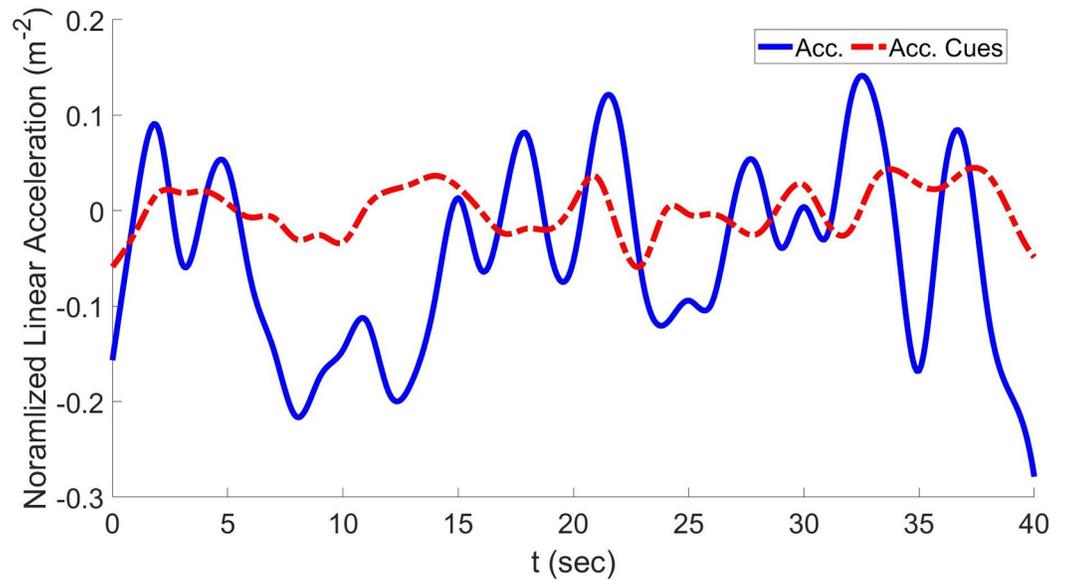


Fig 11. Lateral acceleration and resulting cues using the proposed approach.

<https://doi.org/10.1371/journal.pone.0290705.g011>

compared with vehicle input. Then, this error is compared as an input to H_{∞} controller that eliminates error in translational sensational signal.

In the rotational channel of the H_{∞} controller, inputs are provided through the otolith organ and vestibular organ. The result of the low-pass filter is input to this block for tilt coordination and then compared with linear vehicle input to determine the error. Then, this error is then provided as an input to the H_{∞} controller.

As seen in Table 1, in the case of GA-based MCA, the RMSD in the longitudinal direction is 0.072 and the COCC is 0.95. In the case of H_{∞} -based optimal MCA, RMSD is significantly reduced to 0.022 and the COCC is significantly improved to 0.98. Moreover, in the case of

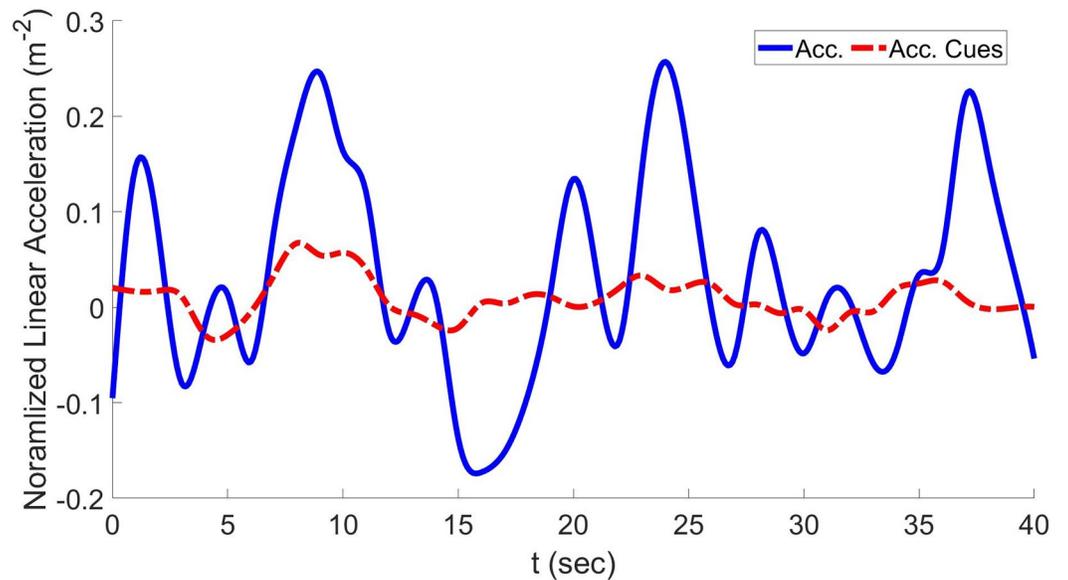


Fig 12. Longitudinal acceleration and resulting acceleration cues using the proposed approach.

<https://doi.org/10.1371/journal.pone.0290705.g012>

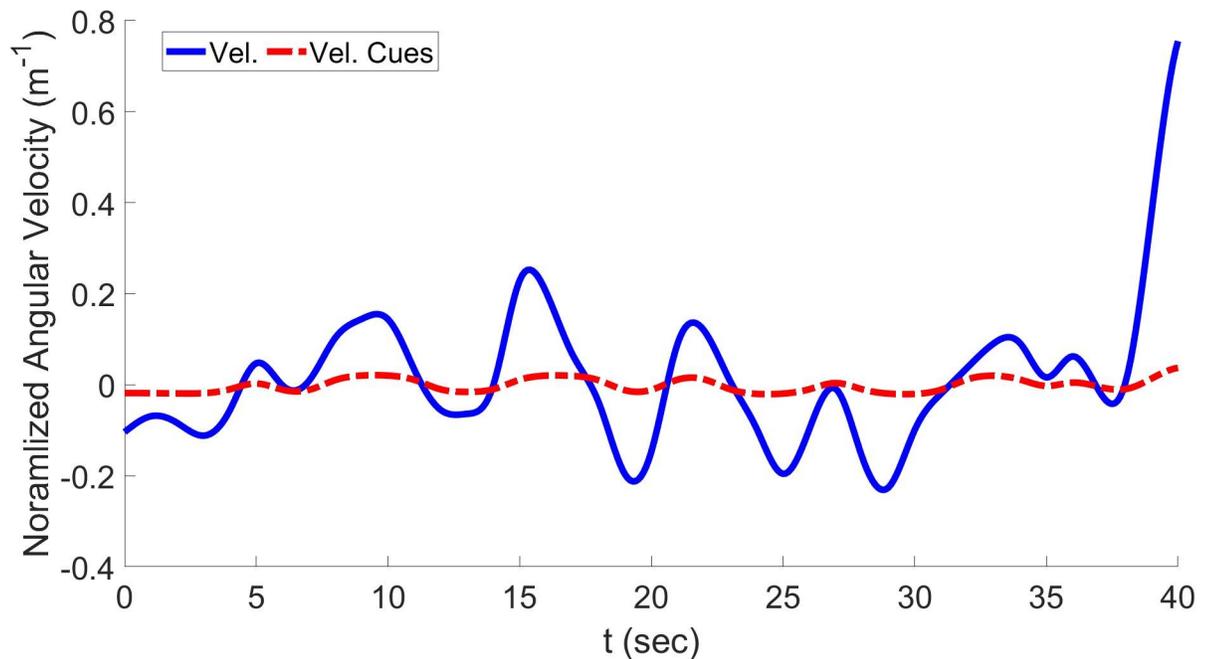


Fig 13. Yaw velocity and resulting cues using the proposed approach.

<https://doi.org/10.1371/journal.pone.0290705.g013>

GA-based optimal MCA, the RMSD in the lateral direction is 0.092 and the COCC in the same direction is 0.96. In the case of the H_∞ -based optimal MCA, RMSD is significantly reduced to 0.036 and the COCC is significantly improved to 0.97. The RMSD in the yaw velocity direction in the case of GA-based optimal MCA is 0.082 and the COCC in the same direction is 0.98. In the case of the H_∞ -based optimal MCA, RMSD is significantly reduced to 0.013 and the COCC is significantly improved to 0.98.

The RMSD, COCC, maximum displacement, and maximum angle of NOC, LQR, GA, and H_∞ control-based MCAs are given in Tables 1–4, respectively.

From Table 3, H_∞ -based optimal MCA shows a significant increase in displacement of the simulator to 0.50 in the surge direction. In the case of LQR-based optimal MCA, it is 0.44; in the case of GA-based optimal MCA, it is 0.37. Also, the yaw angle is increased to 4.34 in the case of H-based optimal MCA, 3.22 in the case of LQR-based optimal MCA, and 2.04 in the case of GA-based MCA. Hence, the H_∞ -based optimal MCA improves workspace usage by decreasing RMSD and improving the COCC. This leads to effective implementation of the H_∞ -based optimal routine for motion cueing of driving simulators, considering fewer constraints, and resulting in more workspace usage.

The proposed MCA confirms the production of realistic motion cues by minimizing RMSD. Workspace usage is maximized in two ways:

1. Optimally solving forward kinematics
2. Using H_∞ controller

This leads to the effective implementation of optimal routine for motion cueing of driving simulators, considering fewer constraints and resulting in more workspace usage.

In short, all the current optimal MCAs do not utilize a sufficient simulator workspace. Linear data fitting techniques for motion cueing problems are limited by second-order performance indices, such as acceleration values. In contrast, model predictive control involves different

systems with different physical constraints. NOC overcomes these limitations by integrating time-varying cost functions adjusted by driver's feedback. Although NOC is an open-loop method, it is still helpful in analyzing performance constraints in loop strategies. Other methods, such as LQR-based optimal control and GA-based optimal control, show better performance in improving the sensation signal. However, they cannot enhance workspace usage due to more constraints involved. The LQR-based method improves the workspace but has more sensation error, and the GA-based method decreases error but with poor workspace usage.

Conclusion

This research proposes a novel optimal MCA for vehicle driving simulators that decreases the sensation error between virtual surroundings and real drivers. The quadratic cost function of the error between vehicle signal and simulator output is minimized using the conventional H_∞ -based optimal control. In addition, usage of the simulator workspace is improved by implementing driver's body model with driver's perception model in optimal MCA. The procedure to design optimal MCA $W(s)$ and the performance of this algorithm through computer simulations have been presented. The simulation results have shown significant improvement both quantitatively and qualitatively.

For future work, other optimization techniques like PSO can be applied for assessment. Moreover, fuzzy logic controller-based optimal MCA can be employed. Like the H_∞ controller, a fuzzy logic controller can be adopted to enhance motion fidelity.

Author Contributions

Conceptualization: Muhammad Ahmad Chaudhry, Inam Ul Hasan Shaikh, Ahsan Ali.

Data curation: Awais Yasin, Saeed Mian Qaisar.

Formal analysis: Awais Yasin.

Investigation: Hazoor Ahmad, Muhammad Tariq, Awais Yasin, Inam Ul Hasan Shaikh, Saeed Mian Qaisar.

Methodology: Hazoor Ahmad, Muhammad Tariq, Sohail Razzaq.

Project administration: Muhammad Ahmad Chaudhry.

Resources: Sohail Razzaq, Jamshed Iqbal.

Software: Hazoor Ahmad, Muhammad Tariq.

Supervision: Inam Ul Hasan Shaikh, Ahsan Ali.

Validation: Jamshed Iqbal.

Visualization: Ahsan Ali, Saeed Mian Qaisar.

Writing – original draft: Hazoor Ahmad, Muhammad Tariq.

Writing – review & editing: Jamshed Iqbal.

References

1. Lee WS, Kim JH, Cho JH. A driving simulator as a virtual reality tool. In: Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146). vol. 1. IEEE; 1998. p. 71–76.
2. Ahmed E, Iqbal J, Arshad M, Liang W, Youn I. Predictive control using active aerodynamic surfaces to improve ride quality of a vehicle", *Electronics*, 2020, 9(9):1463.
3. Casas-Yrurzum S, Portal'es-Ricart C, Morillo-Tena P, Cruz-Neira C. On the Objective Evaluation of Motion Cueing in Vehicle Simulations. *IEEE Transactions on Intelligent Transportation Systems*. 2020; 22(5):3001–3013.

4. Ahmad H, Arif T, Hanif MA, Hafiz R, Shafique M. SuperSlash: A unified design space exploration and model compression methodology for design of deep learning accelerators with reduced off-chip memory access volume. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2020; 39(11):4191–4204.
5. Conrad B, Schmidt S, Douvillier J. Washout circuit design for multi-degrees-of-freedom moving base simulators. In: *Visual and Motion Simulation Conference*; 1973. p. 929.
6. Nehaoua L, Arioui H, Mohellebi H, Espi'e S. Restitution movement for a low cost driving simulator. In: *2006 American Control Conference*. IEEE; 2006. p. 6–pp.
7. Khusro Y R, Zheng Y, Grotoli M, Shyrokau B. MPC-based motion-cueing Algorithm for a 6-DOF driving simulator with actuator constraints. *Vehicles*. 2020; 2:625–647.
8. Asadi H, Mohamed S, Nahavandi S. Incorporating human perception with the motion washout filter using fuzzy logic control. *IEEE/ASME Transactions on Mechatronics*. 2015; 20(6):3276–3284.
9. Asadi H, Mohamed S, Nelson K, Nahavandi S, Zadeh DR. Human perception based washout filtering using genetic algorithm. In: *International Conference on Neural Information Processing*. Springer; 2015. p. 401–411.
10. Kim MS, Moon Y, Kim G, Lee M. Partial range scaling method based washout algorithm for a vehicle driving simulator and its evaluation. *International journal of automotive technology*. 2010; 11(2):269–275.
11. Pellerin E, Pigeon L, Delisle S. Self-adaptive parameters in genetic algorithms. In: *Data Mining and Knowledge Discovery: Theory, Tools, and Technology VI*. vol. 5433. International Society for Optics and Photonics; 2004. p. 53–64.
12. Asadi H, Mohammadi A, Mohamed S, Nahavandi S. Adaptive translational cueing motion algorithm using fuzzy based tilt coordination. In: *International Conference on Neural Information Processing*. Springer; 2014. p. 474–482.
13. Qazani MRC, Asadi H, Bellmann T, Perdrammehr S, Mohamed S, Nahavandi S. A new fuzzy logic based adaptive motion cueing algorithm using parallel simulation-based motion platform. In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE; 2020. p. 1–8.
14. Han MC, Lee HS, Lee S, Lee MH. Optimal motion cueing algorithm using the human body model. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*. 2002; 45(2):487–491.
15. Mahmoodi A, Kazemi I. Optimal motion cueing algorithm for accelerating phase of manned spacecraft in human centrifuge. *Chinese Journal of Aeronautics*. 2020; 33(7):1991–2001.
16. Asadi H, Mohamed S, Lim CP, Nahavandi S. Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2016; 47(2):238–254.
17. Salisbury IG, Limebeer DJ. Optimal motion cueing for race cars. *IEEE Transactions on Control Systems Technology*. 2015; 24(1):200–215.
18. Hassanat A, Almohammadi K, Alkafaween E, Abunawas E, Hammouri A, Prasath V. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information*. 2019; 10(12):390.
19. Koyuncu AB, Ercelik E, Comulada-Simpson E, Venrooij J, Kaboli M, Knoll A. A Novel Approach to Neural Network-based Motion Cueing Algorithm for a Driving Simulator. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE; 2020. p. 2118–2125.
20. Iqbal J, Khan AA, Ullah M I, Irfan M. Towards sophisticated control of robotic manipulators: Experimental study on a pseudo-industrial arm. *Strojniški vestnik—Journal of Mechanical Engineering*. 2015; 61(7–8): 465–470.
21. Khan O, Pervaiz M, Ahmad E, Iqbal J, On the derivation of novel model and sophisticated control of flexible joint manipulator, *Revue Roumaine des Sciences Techniques-Serie Electrotechnique et Energétique*, 2017, 62(1): 103–108.
22. Ullah M I, Ajwad S A, Irfan M, Iqbal J. Non-linear control law for articulated serial manipulators: Simulation augmented with hardware implementation. *Elektronika ir Elektrotechnika*. 2016; 22(1): 3–7.
23. Taghirad HD. *Parallel robots: mechanics and control*. CRC press; 2013.
24. Anjum M, Khan Q, Ullah S, Hafeez G, Fida A, Iqbal J, et al. R. Maximum power extraction from a stand-alone photo voltaic system via neuro-adaptive arbitrary order sliding mode control strategy with high gain differentiation, *Applied Sciences*, 2022, 12(6): 2773.
25. Coleman TF, Li Y. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*. 1996; 6(2):418–445.
26. Fachinotti V, Anca A, Cardona A. A method for the solution of certain problems in least squares. *Int J Numer Method Biomed Eng*. 2011; 27(4):595–607.

27. Marquardt DW. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*. 1963; 11(2):431–441.
28. AN N. *Mechanical Vibration and Shock—Evaluation of Human Exposure to Whole Body Vibration—Part 4: Guidelines for the Evaluation of the Effects of Vibration and Rotational Motion on Passenger and Crew Comfort in Fixed-Guideway Transport Systems*. ISO. 2007;.
29. Duc-An P, Duc-Toan N. A novel motion cueing algorithm integrated multi-sensory system—Vestibular and proprioceptive system. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*. 2020; 234(2):256–271.
30. Ellensohn F, Spannagl M, Agabekov S, Venrooij J, Schwiembacher M, Rixen D. A hybrid motion cueing algorithm. *Control Engineering Practice*. 2020; 97:104342.
31. Jarrah M, Wasseem W, Othman M, Gdeisat M. Human body model response to mechanical impulse. *Medical engineering & physics*. 1997; 19(4):308–316. [https://doi.org/10.1016/s1350-4533\(96\)00081-1](https://doi.org/10.1016/s1350-4533(96)00081-1) PMID: 9302670
32. Kitazaki S, Griffin MJ. A modal analysis of whole-body vertical vibration, using a finite element model of the human body. *Journal of Sound and Vibration*. 1997; 200(1):83–103.
33. Mohammadi A, Asadi H, Mohamed S, Nelson K, Nahavandi S. Optimizing model predictive control horizons using genetic algorithm for motion cueing algorithm. *Expert Systems with Applications*. 2018; 92:73–81.
34. Qazani MRC, Asadi H, Nahavandi S. A decoupled linear model predictive controlbased motion cueing algorithm for simulation-based motion platform with limited workspace. In: 2019 IEEE International Conference on Industrial Technology (ICIT). IEEE; 2019. p. 35–41.
35. Qazani M. R. C., Asadi H., & Nahavandi S. (2021). An optimal motion cueing algorithm using the inverse kinematic solution of the hexapod simulation platform. *IEEE Transactions on Intelligent Vehicles*, 7(1), 73–82.
36. Qazani M. R. C., Asadi H., Mohamed S., Alqumsan A. A., Rahimzadeh G., Lim C. P., et al. (2021, October). An MPC-based Motion Cueing Algorithm Using Washout Speed and Grey Wolf Optimizer. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1627–1633). IEEE.