

This is the author-created version of the following work:

Chalak Qazani, Mohamad Reza, Asadi, Houshyar, Chen, Yutao, Abdar, Moloud, Karkoub, Mansour, Mohamed, Shady, Lim, Chee Peng, and Nahavandi, Saeid
(2023) *An optimal nonlinear model predictive control-based motion cueing algorithm using cascade optimization and human interaction*. IEEE Transactions on Intelligent Transportation Systems, 24 (9) pp. 9191-9202.

Access to this file is available from:

<https://researchonline.jcu.edu.au/86728/>

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

Please refer to the original source for the final version of this work:

<https://doi.org/10.1109/TITS.2023.3271361>

An Optimal Nonlinear Model Predictive Control-based Motion Cueing Algorithm Using Cascade Optimisation and Human Interaction

Mohammad Reza Chalak Qazani, Houshyar Asadi, *Member, IEEE*, Yutao Chen, Moloud Abdar, Mansour Karkoub, *Senior Member, IEEE*, Shady Mohamed, Chee Peng Lim, Saeid Nahavandi, *Fellow, IEEE*

Abstract—Nonlinear model predictive control has been used in motion cueing algorithms recently to consider the nonlinear dynamics model of the system. The entire motion cueing algorithm indexes, including the physical and dynamical constraints of the actuators and physical constraints of passive joints, can be controlled with precision using nonlinear model predictive control. However, several weighting parameters in the nonlinear model predictive control-based motion cueing algorithm (including driving sensation, motion description of the actuators, and passive joints) require proper and laborious tuning to attain an optimal design structure. In this work, the optimal weighting parameters of a nonlinear predictive control-based motion cueing algorithm model are calculated using cascade optimisation and human interaction. A cascade optimisation method consisting of a particle swarm optimisation and genetic algorithm is designed to identify the best weighting parameters compared to those from one optimiser. In addition, the human decision-making units are added to the two-level cascade optimiser to determine the best solution from a Pareto front. The proposed cascade optimiser decreases the run-time with better extraction of the optimal weighting parameters to increase the motion fidelity compared to a single optimiser. It should be noted that the proposed methodology is applied along longitudinal channel. While the same methodology can be applied along lateral, heave and yaw channels for further evaluation of the proposed method. The proposed model is simulated utilising the MATLAB software and the results prove the efficiency of the newly proposed model compared to those from the previous single optimiser in reproducing more accurate motion signals with better usage of the driving motion platform workspace.

Index Terms—cascade optimisation, human interaction, motion cueing algorithm, nonlinear model predictive control, weight tuning.

I. INTRODUCTION

A motion simulator platform is useful for the control and simulation of air or land vehicles using parallel or serial

robot manipulators [1, 2]. The combination of the motion simulator platform, with 6-degree-of-freedom (6-DoF), and virtual reality headsets provides a realistic virtual environment for simulation-based training [3, 4]. The workspace of the motion simulator platforms is a finite area based on the working envelope of the actuators and passive joints [5, 6]. A motion cueing algorithm (MCA) aims to regenerate the driving signal and provide a similar motion sensation to a real vehicle's driver for the simulator user while respecting the workspace limitations [7]. Motion sickness, which is a main drawback of poor motion cues, occurs when the simulator user perceives discrepancies between the visual cues and motion cues using eyes and vestibular systems, respectively [8].

The classical washout filter is the first form of MCA introduced by Conrad and Schmidt [9]. It is composed of second-order low-pass and high-pass filters, aiming to eliminate the low-frequency component of the translational and rotational driving signals. The filtered driving signals can be followed by a simulator platform movement with a restricted workspace around the neutral position. The optimal structure of a classical washout filter using the linear quadratic regulator has been employed to produce an optimal washout filter [10, 11]. Unfortunately, the classical and optimal washout filters do not consider the workspace limitation of the simulator, which can be taken into consideration when model predictive control (MPC) is used [12]. In addition, adaptive washout filter keeps the platform within limits at the cost of unrealistic motion when approaching the limits [13]. However, all washout filter-based MCAs are not able to consider the driving scenarios and thus require retuning for driving conditions with differing motion characteristics [14].

Initially, the linear time-invariant MPC model was used by Dagdelen *et al.* [14] in designing an MPC-based MCA for a driving simulator. The linear time-invariant MPC model can consider the simulator constraints in the end-effector coordinate system. This is achieved by calculating the optimal input signal

(Corresponding author: Mohammad Reza Chalak Qazani)

M. R. C. Qazani is with the Faculty of Computing and Information Technology, Sohar University, Sohar, Sohar 311, Oman (e-mail: m.r.chalakqazani@gmail.com)

M. R. C. Qazani, H. Asadi, M. Abdar, Chee Peng Lim and S. Nahavandi are with the Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, VIC 3216, Australia (e-mail: m.r.chalakqazani@gmail.com; houshyar.asadi@deakin.edu.au; m.abdar1987@gmail.com;

shady.mohamed@deakin.edu.au;

saeid.nahavandi@deakin.edu.au).

Y. Chen is with the Fuzhou University-Qishan Campus, College of Electric Engineering and Automation, Xueyuan Rd. 2, Fuzhou, 350108, China (e-mail: ytchen_fzu@163.com).

M. Karkoub is with the Mechanical Engineering Department, Lamar University, Beaumont, TX, USA (e-mail: mkarkoub@lamar.edu)

chee.lim@deakin.edu.au,

that minimizes the output error along the prediction horizon concerning the dynamics model of the system. It is not possible to consider the dynamic model of the simulator, including physical and dynamical constraints of the actuators using linear MPC. In Qazani *et al.* [15], a linear time-varying MPC model that integrates the displacement boundaries of the actuators instead of the end-effector constraints was employed. Later, Qazani *et al.* [16] introduced the time-varying weight MPC in order to eliminate the disadvantages of the fixed weights with higher motion sensation error.

Nevertheless, it is impossible to control the entire MCA indices, such as the boundaries of the actuators and passive joints, dexterity, and shape-following capability, without implementing a full nonlinear dynamics system using nonlinear MPC (NMPC). Bruschetta *et al.* [17] proposed the first NMPC for a 9-DoF simulator platform system incorporating the actuators' displacement constraints using the ACADO nonlinear optimisation toolkit. Katliar *et al.* [18] employed the real-time Iteration scheme (RTI) to develop an NMPC-based MCA model for an 8-DoF serial simulator platform system based on the discretised inverse kinematic solution of the manipulator. In addition, ACADO toolkit is used by Khusro *et al.* [19] in the development of the NMPC-based MCA for a 6-DoF hexapod simulator with ability of smoothing the end-effector's motion via an adaptive weight tuning technique. It should be noted that the existing NMPC models [17, 18, 20] have not been compared with linear MPC in the MCA domain. Only Khusro *et al.* [19] compared their NMPC design with linear MPC and the classical washout filter. The NMPC efficiency, however, has not been compared with that of linear MPC owing to the difficulty of the NMPC tuning process. Recently, Qazani *et al.* [21] used NMPC for prepositioning purpose of the end-effector in order to increase the virtual workspace limitation of the simulator.

The tuning process of the MCA indices for increasing the model efficiency is challenging. Casas *et al.* [22] investigated a genetic algorithm-based (GA-based) tuning technique to identify the optimal classical washout filter parameters, such as translational and rotational scale factors, cut-off frequency, damping factor, and gains. Mohammadi *et al.* [23] utilized a single-objective GA to calculate the MPC-based MCA's best MPC horizons to release the mechanism's computational burden and reduce the motion feeling error between the real vehicle's and simulator's users. Later, Mohammadi *et al.* [24] employed a Multiobjective GA and the interaction of human decision to satisfy the user demands. Unfortunately, the NMPC tuning process is more critical and time-consuming than in the linear MPC case because of the model's complexity, which involves the consideration of nonlinear dynamics equations with many weights tuning parameters. It is impossible to extract the optimal tuning parameters without a massive number of populations. However, the large population size exponentially increases the computational time of the process.

Cascade techniques hinge on a combination of several methods to tackle a complex problem. In general, a cascade optimisation technique employs several optimisers working one-by-one in tandem in a particular sequence to eliminate the

disadvantage of a single optimiser using different variables. Zhang *et al.* [25] applied a cascade ensemble classifier to recognise handwritten digits. The achieved results demonstrate the significant impact of their strategy. Neoh *et al.* [26] proposed a combination of GA and PSO optimisers in an "external optimiser–internal optimiser" framework. Neoh *et al.* [26] proved that the combination of GA-PSO outperforms the GA, with 50% saving in the computational time.

The main contributions of this study can be divided into two parts. First, an NMPC-based MCA model is designed and developed using MATMPC [27] to incorporate a full nonlinear dynamics MCA structure for a hexapod simulator system. Inclusion of the full nonlinear dynamics of the MCA leads to better regeneration of motion cues compared to linear time-varying and time-invariant MPC-based MCAs. Also, usage of the MATMPC lightens the computational load of the proposed method in comparison to existing MPC-based MCAs [17-20]. Second, a cascade optimisation technique (which comprises a Multiobjective GA in the first phase and a Multiobjective PSO algorithm in the second phase) is developed to identify the optimal weighting parameters of the NMPC, along with an interaction of a human decision. The proposed method targets the human motion sensation threshold unit, motion sensation error reduction, motion fidelity increase, end-effector constraints, physical and dynamical constraints of the joints, and computation load of the mechanism. The human decision unit selects the best-extracted outputs from the PSO optimiser in the initial phase multiple times and the best-extracted outputs from the GA optimiser in the final phase to yield the final optimal results. The goal is to increase the tuning process's efficiency without compromising the optimisation process's computational load. The proposed technique can be used as a fast adjustable NMPC-based MCA based on the datasets from every simulator's user.

The rest of the paper is organized as follows: the human perception and inverse kinematic models of a hexapod simulator system for use with an NMPC-based MCA model are described in section II. Section III introduces the proposed cascade optimisation technique with a human decision unit to extract the optimal weighing matrices. In section IV, the simulation results are presented and discussed. Section V presents the conclusions of this research.

II. NMPC-BASED MCA

This section investigates a hexapod simulator system's human vestibular and kinematic model to formulate the nonlinear MCA model. Then, an NMPC is developed based on the extracted nonlinear MCA model. The MPC technique can be used easily in multi-input and multi-output systems

A. Human Vestibular Model

The human vestibular system comprises both the semicircular canals and otolith organs as the significant sensors of mammals to detect the orientation and motion in space, respectively [10]. The model of the otolith organs based on the study by Telban and Cardullo [10] and Asadi *et al.* [28] can be represented as:

$$\frac{\hat{f}(s)}{f(s)} = K_{OTO} \left(\frac{(\tau_a s + 1)}{(\tau_L s + 1)(\tau_s s + 1)} \right). \quad (1)$$

where K_{OTO} , τ_a , τ_L and τ_s are static sensitivity, long time constant, short time constant and adaptation operator constant of the otolith organs, respectively. In addition, f and $\hat{f}$ are the applied force and sensed specific force by the simulator user and the real vehicle driver, respectively. Their values are shown in Table I.

The mathematical model of Telban and Cardullo [10] and Asadi *et al.* [29], pertaining to the human semicircular canals, can be represented as:

$$\frac{\hat{\omega}}{\omega} = \left(\frac{\tau_1 \tau_2 s^2}{(1 + \tau_1 s)(1 + \tau_2 s)} \right). \quad (2)$$

where τ_1 and τ_2 are long time constant and short time constant of the semicircular canals, respectively. In addition, ω and $\hat{\omega}$ are the applied angular velocity and sensed angular velocity by the simulator user and the real vehicle driver, respectively. Their values are represented in Table I. Note that the longitudinal channel in Table I covers a combination of the x-axis linear motion and pitch-angle rotational motion. The lateral channel combines the y-axis linear motion and roll-angle rotational motion. In addition, the vertical channel is separated into two heave and yaw sub-channels, which cover the z-axis linear and yaw-angle rotational motions, respectively.

The combination of Eq. (1) and Eq. (2) results in a state space model of the human vestibular system, as follows:

$$\dot{\xi}_V = A_V \xi_V + B_V u. \quad (3.a)$$

$$\hat{y}_V = C_V \xi_V + D_V u. \quad (3.b)$$

where A_V , B_V , C_V and D_V are the state space matrices of the human vestibular model. Also, $u = [\dot{\theta} \ a_x]^T$ and $\hat{y}_V = [\hat{\theta} \ \hat{f}_x]^T$ are the input and output vectors of the human vestibular model, respectively.

To ease the computational burden of NMPC, the human vestibular model is divided into different categories, i.e., the longitudinal, lateral, heave, and yaw channels. The control inputs of the longitudinal channel are the linear acceleration signal along the x-axis (a_x) and pitch angular velocity ($\dot{\theta}$), while the outputs are the sensed specific force along the x-axis ($\hat{f}_x$) and sensed angular velocity ($\hat{\theta}$). The control inputs of the lateral channel are the linear acceleration along the y-axis (a_y) and roll angular velocity ($\dot{\phi}$). It should be noted that there is no tilt-coordination in the heave channel as it is in the same direction

TABLE I
THE VESTIBULAR MODEL PARAMETERS OF THE HUMAN BODY

Index	Longitudinal Channel	Lateral Channel	Vertical Channel
K_{OTO}	0.4	0.4	0.4
τ_a (s)	10	10	10
τ_L (s)	5	5	5
τ_s (s)	0.016	0.016	0.016
τ_1 (s)	5.73	5.73	5.73
τ_2 (s)	80	80	80

as gravity; as such, its control input is only the linear acceleration along the z-axis (a_z), and the output is the sensed specific force along the z-axis ($\hat{f}_z$). The same methodology is followed for the yaw channel using the yaw angular velocity ($\dot{\psi}$) as the model input. In this study, we only investigate the proposed method along the longitudinal channel. However, the same methodology can be followed for the other three channels.

B. Dynamical Model of Hexapod Simulator

The hexapod manipulator is a well-known parallel structure, as shown in Fig. 1. It comprises six prismatic joints connected to a fixed plate using universal passive joints, which are then connected to an end-effector (cockpit) using spherical passive joints [30, 31]. The cockpit can be configured based on the length of the prismatic actuators. As a result, the translational and rotational motions of the cockpit can be generated by displacing the prismatic actuators. Inverse position kinematics are employed to identify the prismatic joints' length using the cockpit's position. The vectorial position of the actuators ($\mathbf{L}_i$) can be calculated for the hexapod's kinematic chain shown in Fig. 1 as:

$$\mathbf{L}_i + \mathbf{b}_i = \mathbf{c} + \mathbf{a}_i \quad i = 1 - 6 \quad (4)$$

where $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c} = [x \ y \ z]^T$ are the vectorial position of the i^{th} spherical joints, universal joints, and centre of the cockpit regarding the $\mathbf{W}$ coordinate system, respectively. The vectorial position of the i^{th} spherical joints can be extracted as:

$$\mathbf{a}_i = \mathbf{R}^P \mathbf{a}_i \quad (5)$$

where $\mathbf{P} \mathbf{a}_i$ is the vectorial position of the i^{th} spherical joint in the $\mathbf{P}$ coordinate system, while $\mathbf{R}$ refers to the rotation matrix of the cockpit around three directions. The length of the actuators in Eq. (4) can be re-written:

$$l_i \mathbf{n}_i = \mathbf{c} + \mathbf{R}^P \mathbf{a}_i - \mathbf{b}_i \quad (6)$$

where l_i and $\mathbf{n}_i$ are the length and unit vector of the i^{th} prismatic joint.

The inverse velocity kinematics solution is the calculation of the i^{th} prismatic joints velocity ($\dot{l}_i$) using the velocity of the cockpit, which can be found by taking the derivative of the above equation and performing a dot product of the outcomes with $\mathbf{n}_i$, i.e.,

$$\dot{l}_i = \dot{\mathbf{c}} \cdot \mathbf{n}_i + \boldsymbol{\omega} \cdot (\mathbf{R}^P \mathbf{a}_i \times \mathbf{n}_i) \quad (7)$$

where $\dot{\mathbf{c}}$ and $\boldsymbol{\omega}$ are the linear and angular velocities of the cockpit; $\dot{\mathbf{c}} = [V_x \ V_y \ V_z]^T$ and $\boldsymbol{\omega} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$, respectively. The inverse velocity kinematics problem in Eq. (7) can be re-written in a matrix form, i.e.,

$$[\dot{l}]_{6 \times 1} = \mathbf{J}^{-1} [\dot{\mathbf{c}}] \quad (8)$$

where $\mathbf{J}^{-1}$ is the inverse Jacobian matrix of the velocity, i.e.,

$$\mathbf{J}^{-1} = \begin{bmatrix} \mathbf{n}_1^T & \mathbf{a}_1 \times \mathbf{n}_1 \\ \vdots & \vdots \\ \mathbf{n}_6^T & \mathbf{a}_6 \times \mathbf{n}_6 \end{bmatrix} \quad (9)$$

The acceleration of the i^{th} pod $\ddot{l}_i$ can be calculated by taking the derivative of Eq. (7) as:

$$\ddot{l}_i = \ddot{\mathbf{c}} \cdot \mathbf{n}_i + \boldsymbol{\alpha} \times \mathbf{R}^P \mathbf{a}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{R}^P \mathbf{a}_i) \quad (10)$$

where $\ddot{\mathbf{c}}$ and $\boldsymbol{\alpha}$ are the linear and angular accelerations of the cockpit centre, respectively. It can be rewritten in a matrix for as follows:

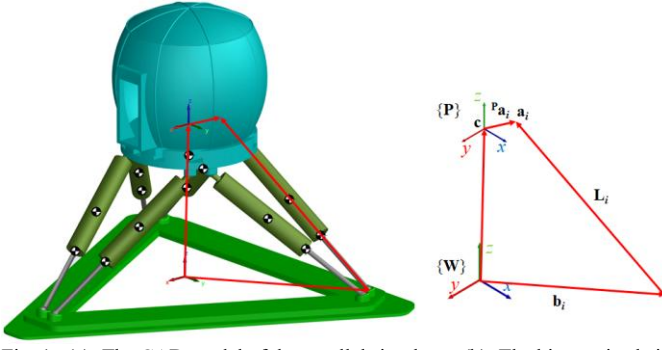


Fig. 1. (a): The CAD model of the parallel simulator; (b): The kinematic chain of the parallel simulator.

$$\ddot{\mathbf{I}}_{6 \times 1} = \mathbf{J}_{6 \times 6}^{-1} \left[\ddot{\mathbf{c}}_{\alpha} \right] + \frac{d\mathbf{J}^{-1}}{dt} \left[\dot{\mathbf{c}}_{\omega} \right] \quad (11)$$

where $\frac{d\mathbf{J}^{-1}}{dt}$ is the derivative of the inverse velocity Jacobian matrix and can be written as:

$$\frac{d\mathbf{J}^{-1}}{dt} = \begin{bmatrix} (\boldsymbol{\omega}_1 \times \mathbf{n}_1)^T & ((\boldsymbol{\omega}_1 \times \mathbf{a}_1) \times \mathbf{n}_1 + \mathbf{a}_1 \times (\boldsymbol{\omega}_1 \times \mathbf{n}_1))^T \\ \vdots & \vdots \\ (\boldsymbol{\omega}_6 \times \mathbf{n}_6)^T & ((\boldsymbol{\omega}_6 \times \mathbf{a}_6) \times \mathbf{n}_6 + \mathbf{a}_6 \times (\boldsymbol{\omega}_6 \times \mathbf{n}_6))^T \end{bmatrix} \quad (12)$$

where $\boldsymbol{\omega}_i$ for $i = 1, \dots, 6$ refers to the angular velocity of the i^{th} prismatic joint.

The workspace constraints of the hexapod simulator system can be chosen based on the size of the manipulator. The workspace constraints are divided into the actuators' physical and dynamical constraints and the passive joints' physical constraints. These constraints on the joints lead to the constraints on the cockpit in the fixed plate Cartesian coordinate system, $\mathbf{W}$, as presented in Table II.

The combination of the human vestibular model and inverse acceleration kinematics, presented in Eq. (3) and Eq. (10), respectively, results in the NMPC model. As NMPC can be used via a continuous mode of the system, the state of the dynamics model along the longitudinal channel can be

TABLE II

THE LIMITATIONS OF THE HEXAPOD SIMULATOR ALONG X-AXIS

Index	Description
l_i	$l_i = 1.0 - 1.2 \text{ (m)}; \dot{l}_i = \pm 0.35 \text{ (m/s)}; \ddot{l}_i = \pm 2 \text{ (m/s}^2\text{)}$
UPPER	$\begin{bmatrix} 0.6428 & 0.6428 & 0.342 & -0.9848 & -0.9848 & 0.342 \end{bmatrix}$
PASSIVE	$\begin{bmatrix} -0.766 & 0.766 & 0.9397 & 0.1736 & -0.1736 & -0.9397 \end{bmatrix}$
JOINT	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
Lower	$\begin{bmatrix} 2.9971 & 2.9971 & -1.3852 & -1.6119 & -1.6119 & -1.3852 \end{bmatrix}$
passive	$\begin{bmatrix} -0.1309 & 0.1309 & 2.6610 & 2.5302 & -2.5302 & -2.6610 \end{bmatrix}$
joint	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
Mass	$m_{\text{COCKPIT}} = 1216.9 \text{ (Kg)}; m_{\text{UPPERLEG}} = 143.9 \text{ (Kg)};$ $m_{\text{LOWERLEG}} = 92.1 \text{ (Kg)}$
Inertia	$\mathbf{I}_{\text{cockpit}} = \begin{bmatrix} 304.5 & 0 & 0 \\ 0 & 304.5 & 0 \\ 0 & 0 & 304.5 \end{bmatrix};$
Matrix	$\mathbf{I}_{\text{upperleg}} = \begin{bmatrix} 67.2 & 0 & 0 \\ 0 & 67.2 & 0 \\ 0 & 0 & 67.2 \end{bmatrix};$
	$\mathbf{I}_{\text{lowerleg}} = \begin{bmatrix} 43 & 0 & 0 \\ 0 & 43 & 0 \\ 0 & 0 & 43 \end{bmatrix}$
x	$x = \pm 0.20 \text{ (m)}; \dot{x} = \pm 0.55 \text{ (m/s)}; \ddot{x} = \pm 2.5 \text{ (m/s}^2\text{)}$
θ	$\theta = \pm 0.5 \text{ (rad)}; \dot{\theta} = \pm 0.1 \text{ (rad/s)}; \ddot{\theta} = \pm 0.25 \text{ (rad/s}^2\text{)}$

m: meter; rad: radian; s: second;

represented as:

$$\boldsymbol{\xi} = [\boldsymbol{\xi}_V \ V_x \ x \ \varphi \ \ddot{l}_{i=1-6} \ \dot{l}_{i=1-6} \ l_{i=1-6}]^T \quad (13)$$

Also, the input and output vectors of the dynamics model can be defined as:

$$\mathbf{u} = [\dot{\theta} \ a_x]^T \quad (14.a)$$

$$\mathbf{y} = [\hat{\theta} \ \hat{f}_x \ V_x \ x \ \varphi \ \ddot{l}_{i=1-6} \ \dot{l}_{i=1-6} \ l_{i=1-6}]^T \quad (14.b)$$

It should be noted that the represented dynamics model of the mechanism is decoupled along four channels to alleviate the computational load of the mechanism. Only the inputs along the longitudinal channel are used in the model.

C. Nonlinear Model Predictive Control

An NMPC model in the discretised mode using direct multiple shooting and sampling time $T_s = 0.01 \text{ (s)}$ along the prediction horizon (t_0 to t_f) can be formulated as:

$$\min_{\xi_k, u_k} \sum_{k=0}^{N-1} \left\{ \frac{1}{2} \|h_k(\xi_k, u_k)\|_W^2 + \frac{1}{2} \|h_N(\xi_N)\|_{W_N}^2 \right\} \quad (15.a)$$

$$\text{s.t.} \quad 0 = \xi_0 - \hat{\xi}_0 \quad (15.b)$$

$$0 = \xi_{k+1} - \Phi_k(\xi_k, u_k), k = 0, 1, \dots, N-1 \quad (15.c)$$

$$\underline{r}_k \leq r_k(\xi_k, u_k) \leq \bar{r}_k, k = 0, 1, \dots, N-1 \quad (15.d)$$

$$\underline{r}_N \leq r_k(\xi_N) \leq \bar{r}_N \quad (15.e)$$

where $N = (t_f - t_0)/T_s$ is the shooting intervals; $\hat{\xi}_0$ is the state at the initial time of the system; h is the objective function of the optimal control problem, which is used to minimise the driving sensation error and displacement of the cockpit and joints; Φ is the nonlinear dynamics model of the plant, which is a combination of Eq. (3) and Eq. (10); ξ_k and u_k are the states and control input at every sampling time; r is the linear and nonlinear constraints of the model with the upper limits $\bar{r}$ and lower limits $\underline{r}$; respectively. Eq. (15.e) defines the terminal constraints of the state as the end of the prediction horizon to increase the model stability. The first item of the extracted control action is implemented in the system, which is known as the receding horizon technique.

Using sequential quadratic programming (SQP) for the discretised state and input, Eq. (15) can be formulated as a quadratic programming (QP) problem, i.e.,

$$\min_{\Delta \xi, \Delta u} \sum_{k=0}^{N-1} \left\{ \left(\frac{1}{2} \begin{bmatrix} \Delta \xi_k \\ \Delta u_k \end{bmatrix}^T \mathbf{H}_k^i \begin{bmatrix} \Delta \xi_k \\ \Delta u_k \end{bmatrix} + \mathbf{g}_k^i \begin{bmatrix} \Delta \xi_k \\ \Delta u_k \end{bmatrix} \right) + \frac{1}{2} \Delta \xi_N^T \mathbf{H}_N^i \Delta \xi_N + \mathbf{g}_N^i \Delta \xi_N \right\} \quad (16.a)$$

$$\text{s.t.} \quad \Delta \xi_0 = \hat{\xi}_0 - \xi_0 \quad (16.b)$$

$$\Delta \xi_{k+1} = \mathbf{A}_k^i \Delta \xi_k + \mathbf{B}_k^i \Delta u_k + \mathbf{d}_k^i \quad (16.c)$$

$$\underline{c}_k^i \leq \mathbf{C}_k^i \Delta \xi_k + \mathbf{D}_k^i \Delta u_k \leq \bar{c}_k^i \quad (16.d)$$

$$\underline{c}_N^i \leq \mathbf{C}_N^i \Delta \xi_N \leq \bar{c}_N^i \quad (16.e)$$

where $\boldsymbol{\xi} = [\xi_0^T \ \dots \ \xi_N^T]^T$, $\mathbf{u} = [u_0^T \ \dots \ u_N^T]^T$, $\Delta \boldsymbol{\xi} = \boldsymbol{\xi} - \boldsymbol{\xi}^i$ and $\Delta \mathbf{u} = \mathbf{u} - \mathbf{u}^i$, $\mathbf{H}_k^i$ and $\mathbf{g}_k^i$ are the Hessian matrices; $\mathbf{A}_k^i$ and $\mathbf{B}_k^i$ are matrices from the linearisation of the system with respect to the initial states and control over the prediction horizon, respectively. The MPC horizons (including prediction and control horizons) should be tuned based on the computational load limits of the system, aiming to enable the system to be run for real-time applications. A right balance among the computational load, motion sensation error, and efficient usage

of the workspace limits should be considered in the tuning process of the MPC horizon. In this study, the prediction and control horizons of the MPC are selected to be 50 time-step and 4 time-step, respectively. These have been selected based on trial and error aiming to reduce the computational load of the proposed method while yielding reasonable results. It should be noted that time-step is equal to 0.01 (s). C_k^i is the Jacobian constraint matrix; while d_k^i refers to the existing gap between the previous shooting and the following starting points. H_k^i and g_k^i are the k^{th} block and sub-vector of the Hessian matrix and the gradient of the objective function, respectively. The Hessian matrix is given by:

$$H_k^i = \frac{\partial h_k^i}{\partial(\xi_k, u_k)}^T \frac{\partial h_k^i}{\partial(\xi_k, u_k)} \quad (17)$$

which is always positive and semi-definite. Eq. (16) can be solved using sparse or condensed techniques. The condensing technique [32] removes the variation of the states and re-shapes Eq. (16) as follows:

$$\min_{\Delta u} \left\{ \frac{1}{2} \Delta u^T H_c \Delta u + g_c^T \Delta u \right\} \quad (18.a)$$

$$s. t. \quad \underline{c}_c \leq C_c \Delta u \leq \bar{c}_c \quad (18.b)$$

Eq. (18) is the standard shape of the QP problem, which can be solved.

III. CASCADE OPTIMISATION TECHNIQUE WITH HUMAN INTERACTION

A single optimisation algorithm may not be able to solve all optimisation problems at the same time. Therefore, a cascade optimisation using different optimisation techniques could be more efficient. In this method, the combination of GA and PSO is used to extract the cascade optimisation method to reach a better tuning method for a nonlinear MPC. GAs is based on random selection of the individual with strength power in finding the best solution to the optimisation problem. However, extracting the optimal value using a complex system will take a long time. Therefore, it would be better to extract the weighting parameters of the sensed specific force and angular velocity using Multiobjective GAs since these two parameters have a critical role in decreasing the motion sensation error. Also, PSO is faster than GA, but it cannot find the optimal solution using a very complex system. There is a chance of sticking to the local optimal that will affect the results of the PSO method. Therefore, the rest of the weighting parameters, including the acceleration, velocity and position of the active joints, acceleration, velocity and position of the end-effector and the motion input signals, should be selected in the second phase of the cascade algorithm using the PSO method.

A. GA Operator

An GA operator is an optimisation scheme based on the evolutionary algorithms' natural selection. GAs, which are bio-inspired operators, use mutation, crossover, and selection, to solve a wide range of optimization problems. GAs often start with generating a random population of chromosomes which are subsequently "assessed" using an objective function. The ones that best approximate the "ideal solution" have a higher probability of reproducing [33].

GA parameters must be carefully chosen to maximize the accuracy and speed of the final result's convergence. The crossover and mutation parameters must be adjusted correctly to ensure the success of the GA and the acquired output. The high mutation rate raises the danger of missing a solution close to the present state. Setting the mutation parameter to a small value raises the likelihood of being trapped in the local optimum. Additionally, crossover prevents the offspring's from becoming identical replicas of the parents from the prior generations. The mutation parameter had to be set to a fair value and the crossover parameter to a significant value, as suggested by Mitchell [33].

B. PSO Operator

Kennedy *et al.* [34] developed PSO as a swarm-based algorithm category of the meta-heuristic algorithms. PSO is a stochastic population-based optimisation method via the collaborative behaviour of a different colony of species known as populations. It is similar to evolutionary computation methods such as GA [33]. It starts by defining the initial particles to evaluate the objective function at every particle's location. It then decides the minor objective function as the best position. It calculates the new velocities according to the best position of the particles and their neighbours and the current velocity. The mentioned procedure continues iteratively until the algorithm fulfils the stopping criteria. PSO is capable of improving the candidate solution of an optimisation problem iteratively. The search method is based on the four parameters, including the current position p_i , the current velocity v_i , the global best position G_{best} , and the best position P_{best} . The best position vector as a minimization problem can be calculated as:

$$P_{best,i}^{t+1} = \begin{cases} P_{best,i}^t & \text{if } f(p_i^{t+1}) \geq P_{best,i}^t \\ p_i^{t+1} & \text{if } f(p_i^{t+1}) \leq P_{best,i}^t \end{cases} \quad (19)$$

where f and t are the fitness function and iteration number. The global best position can be calculated as:

$$G_{best} = \min_{i \in \{1, \dots, n\} \wedge n > 1} (P_{best,i}^t) \quad (20)$$

where G_{best} is the best-fit particle in the neighbours. The particle should be updated in every iteration by the velocity vector to find the particle's new position.

C. Hybridization of GA, PSO using Human Interaction

Because of the usefulness of cascade optimisation in past studies, it is employed here to identify the most appropriate parameter setting of nonlinear MPCs with the interaction of the human decision unit. To do so, we apply two well-established evolutionary algorithms in two stages: a GA in the first stage and a PSO algorithm in the second stage. In addition, we incorporate human interactions to check the obtained weights in each step of the GA and PSO optimisers. Human-computer interaction (HCI) uses computer technologies to establish connections between individuals (users) and computers. The proposed HCI comprises two stages to interact with the user, which aim to select both groups of weights based on the system requirements.

According to Fig. 2a, the weighting parameters, $W_k = \{W_1, \dots, W_{10}\}$, are divided into the two main subsets: $W_{s1} =$

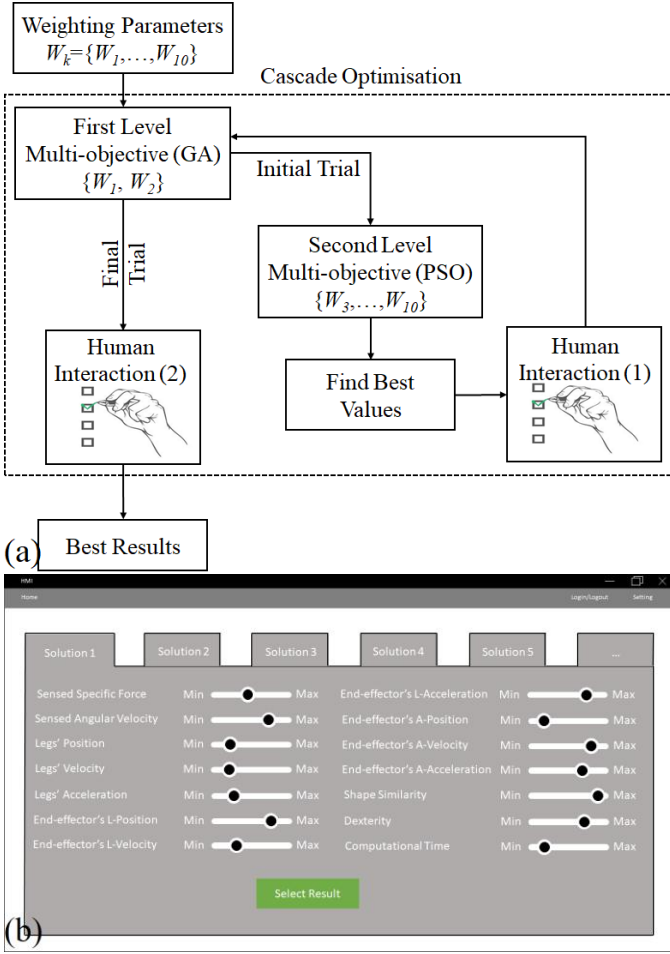


Fig. 2. (a): Schematic of cascade optimisation technique; (b): HCI visual environment.

$[W_{\hat{f}_x}; W_{\hat{\omega}_\theta}]$ and $W_{s_2} = [W_x; W_{V_x}; W_{a_x}; W_\theta; W_{\omega_\theta}; W_l; W_i; W_{a_x}; W_{\alpha_\theta}]$. First, PSO is applied to find the best values of eight parameters in W_{s_2} . Subsequently, a human check the obtained values and tunes the value of each parameter based on his/her experience (human interaction 1). The goal of human interaction 1 is to decide the best possible solution between the obtained optimal solution based on the specification of the simulator. For instance, the regeneration of the $0.2 \text{ (m/s}^2\text{)}$ acceleration is possible using two different solutions while using different linear displacements of the simulator. Then, choosing the solution using less linear displacement is wise to save more space for the following motion signal. In addition, Fig. 2b shows a graphical representation of the proposed HCI for selecting the best solution by the operator after each generation by the evolutionary algorithm (GA or PSO) algorithm. The main goal is to reduce the system complexity by increasing the interaction of experts in selecting the optimal solution.

Subsequently, the best values of the two remaining parameters in W_{s_1} are produced by the GA. Once again, the obtained best values of eight parameters in W_{s_2} are checked by a human (human interaction 2). Table III shows the parameters used in this method. The cascade optimisation technique is

trained and tested on four different signal data sets, including white noises, vehicle sharp turning, acceleration and deceleration for 30 (s) in the simulation environment to extract the optimal weighting parameters for the nonlinear MPC suitable for common motion scenarios. This leads to consistent results in dealing with motion scenarios in each data set.

Fig. 3 presents the pseudocode of the proposed cascade optimisation algorithm to find the optimal weighting indices of the NMPC-based MCA. The cost functions of the Multiobjective GA and PSO are calculated as follows:

$$J_{GA1,PSO1} = w_{\hat{\omega}_\theta} \int ((\hat{\omega}_\theta)_{veh} - (\hat{\omega}_\theta)_{SBMP}) dt + w_{\hat{f}_x} \int ((\hat{f}_x)_{veh} - (\hat{f}_x)_{SBMP}) dt \quad (21.a)$$

$$J_{GA2,PSO2} = w_x \int (\max(x))^2 dt + w_{V_x} \int (\max(V_x))^2 dt + w_{a_x} \int (\max(a_x))^2 dt + w_\theta \int (\max(\theta))^2 dt + w_l \int (\max(l_i))^2 dt + w_i \int (\max(i_i))^2 dt + w_i \int (\max(\ddot{l}_i))^2 dt + w_{CC} \int (CC)^2 dt + w_d \int (d)^2 dt + w_{\omega_\theta} \int (\max(\omega_\theta))^2 dt + w_{\alpha_\theta} \int (\max(\alpha_\theta))^2 dt + w_T \int (\max(T))^2 dt \quad (21.b)$$

where $w_{\hat{\omega}_\theta}$, $w_{\hat{f}_x}$, w_x , w_{V_x} , w_{a_x} , w_θ , w_l , w_i , w_l , w_{CC} , w_d , w_{ω_θ} , w_{α_θ} and w_T are the influence factors of the sensed angular velocity, sensed specific force, linear displacement, linear velocity, linear acceleration, and angular displacement of the end-effector, leg's position, velocity, and acceleration, shape similarity, dexterity, angular velocity and angular acceleration of the end-effector and running time of the process.

Fig. 4 depicts the selected solution based on two fitness functions formulated in Eq. (21a-b). It should be noted that the Pareto front distributions of the best solutions are regenerated after finishing each generation of GA. Then, the operator can select the best solution among them based on the system requirements, including the motion sensation errors, control action, motion parameters, dexterity and computational load.

It should be noted that the tuning time of the cascade and pure GA optimisation techniques are 36 and 118 hours, respectively, using the system with a processor: Intel® Core™ i7-9700K CPU 3.60 GHz and Installed memory 32.0 GB.

TABLE III
CASCADE ADJUSTMENT PARAMETERS

Parameters	GA	PSO
Variables	$[W_{\hat{f}_x}; W_{\hat{\omega}_\theta}]$	$[W_x; W_{V_x}; W_\theta; W_{\omega_\theta}; W_l; W_i; W_{a_x}; W_{\alpha_\theta}]$
Population Size & Swarm Size	35	45
Maximum Generation & Maximum Stall	15	10
Upper Limit	$[2.4 \times 10^6; 2.4 \times 10^4]$	$[1.2 \times 10^4; 2 \times 10^{-1}; 2 \times 10^3; 2 \times 10^{-1}; 2 \times 10^{-1}; 1 \times 10^3; 2 \times 10^{-4}; 2 \times 10^{-4}]$
Lower Limit	$[0; 0]$	$[0; 0; 0; 0; 0; 0; 0; 0]$
Crossover Rate	0.8	NA
Mutation Function	Adaptive Feasible	NA
Function Tolerance	10-4	10-4

Algorithm 1: Hybridization of GA, PSO using Human Interaction	
Input: weighting parameters $W_k = \{W_1, \dots, W_{10}\}$,	
Output: Best Values of $W_k = \{W_1, \dots, W_{10}\}$,	
Begin	
1. Separation of Weights:	
// Divide the weighting parameters into two separate sub-sets.	
$W_{s_1} = [W_{f_x}; W_{\omega_\theta}]$,	
$W_{s_2} = [W_x; W_{V_x}; W_{a_x}; W_\theta; W_{\omega_\theta}; W_l; W_i; W_{a_x}; W_{a_\theta}]$,	
2. Set Cost Functions of GA and PSO Using Eqs. 20a and 20b,	
3. Apply Cascading Optimisation // Using GA and PSO in two steps.	
3.1. Apply PSO to W_{s_2} :	
3.2. Initialize Particle Swarm	
3.3. Evaluate Fitness of Particle Swarm	
For n = 1 to the Number of Particles	
3.3.1. Find pbest	
3.3.2. Find gbest	
3.4. Return the Best Values of	
$W_x; W_{V_x}; W_{a_x}; W_\theta; W_{\omega_\theta}; W_l; W_i; W_{a_x}; W_{a_\theta}$	
to the First Human Interaction Center,	
3.4.1. Choose the Best Values of	
$W_x; W_{V_x}; W_{a_x}; W_\theta; W_{\omega_\theta}; W_l; W_i; W_{a_x}; W_{a_\theta}$,	
End for	
3.5. Apply GA to W_{s_1} :	
3.6. Initialize Population with P Chromosomes and the Other Parameters,	
For each Chromosome in the Population	
3.6.1. Return the Best Values of	
$W_{f_x}; W_{\omega_\theta}$ to the Second Human Interaction Center,	
3.6.2. Choose the Best Values of $W_{f_x}; W_{\omega_\theta}$,	
4. Return the best values of W_k ,	
End for	
End.	

Fig. 3. The pseudocode of the proposed cascade optimisation algorithm in order to find the optimal weighting parameters of the NMPC-based MCA.

IV. RESULTS AND DISCUSSION

The objective and subjective studies of the MCA is a standard approach to evaluating the effectiveness of these methods. The initial objective evaluation of the MCA was conducted by Casas *et al.* [35], followed by other researchers [14, 17]. On the other hand, a subjective study was conducted by Arjona *et al.* [36] on a flight simulator platform. As the subjective study faces many uncertainty issues, such as different understanding of the motion by different human subjects, effectiveness of a practical implementation of the MCA, and accuracy of the used simulator platform, the objective study is the most reliable method for evaluation of different MCAs. This section contains two subsections including verification and validation. In the first part of this section, the steps for developing the proposed model using MATLAB software is explained in detail. Then, the following subsection is used to show the improvement of the proposed model in comparison with those of previous models.

A. Verification

The proposed NMPC-based MCA model in section II is tuned using the cascade optimisation technique in section III, known as NMPC-CAS. The proposed NMPC-based MCA

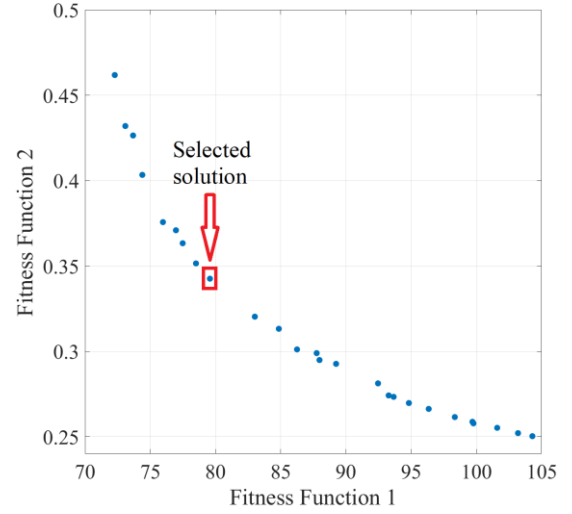


Fig. 4. (a): Pareto-front of cascade optimisation technique.

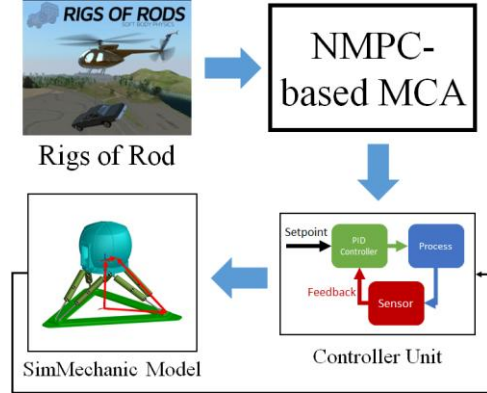


Fig. 5. The schematic representation of the whole process.

model is tuned using a Multiobjective GA (known as NMPC-GA) for comparison purposes. The optimally designed linear time-invariant MPC-based MCA model based on the work of Mohammadi *et al.* [24] using a Multiobjective GA (known as LMPC-GA) is developed for comparison with NMPC-based MCA for the first time in this contribution. In addition, the linear time-varying MPC-based MCA model in Qazani *et al.* [15] (known as LTV MPC-GA) is developed for comparison with the proposed NMPC-based MCA model. In short, four methods, namely LMPC-GA, LTV MPC-GA, NMPC-GA and NMPC-CAS, are compared in this study. Fig. 5 shows the procedure for the proposed method using the signal generation unit, MCA, inverse kinematic unit, PID controller and SimMechanic model of the hexapod simulator. The proposed model comprises two main parts under MATLAB/Simulink environment. The first is the development of NMPC-based MCA using MATMPC. The second is the development of the Hexapod simulator system using a PID controller and the SimMechanic Toolbox of MATLAB. The SimMechanic model of the Hexapod system is the closest duplication of this mechanism in a simulation environment with the definition of dynamical, mechanical, and geometrical parameters of the simulator platform. It should be noted that the driving scenario is recorded in the rigs-of-rod application about vehicle

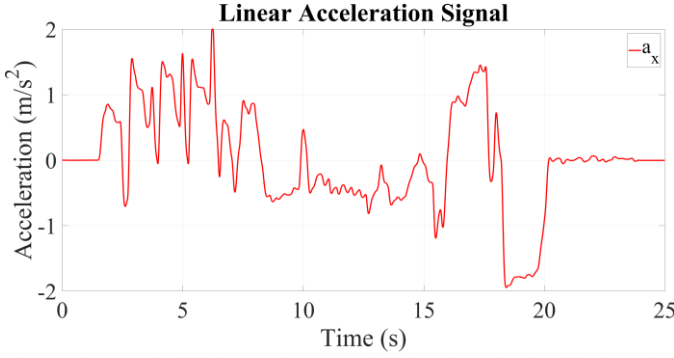


Fig. 6. The recorded driving motion scenario using rigs-of-rod soft body engine (version 0.39.5 the vehicle simulation environment), translational motion signal.

simulation with 30 (s) of motion from the starting point, harsh turning on the edge of the road, and severe braking to stop the vehicle. The MCA regenerated the motion cues for the simulator using the real driving scenario. The tasks of inverse kinematic and PID units are the calculation of the joints' position and control activities of the joints. Then, the SimMechanic model of the Hexapod simulator generated the 6-DoF motions. The models mentioned above are verified using the MATLAB/Simulink software and simulated using ODE-3 (Bogacki-Shampine) with a time step of 0.01 (s). It should be noted that the MATMPC package supports both MATLAB coding and the Simulink environment [27]. All four main blocks, shown in Fig. 5, are modelled in MATLAB to validate the proposed model's results as compared to the existing ones.

Fig. 6 shows the translation motion signal of the vehicle along the longitudinal channel for 25 (s) of motion. The models are validated using three parameters, i.e., the root means square error (RMSE) of the motion sensation between the real vehicle driver and simulator user, CC between the reference motion signal perceived via the real vehicle driver and the simulator user, and the phase lag between the signals. The idea of using CC to evaluate the regenerated motion signals via the MCA model was introduced by Asadi *et al.* [37]. It can depict the shape similarity between two signals since the human vestibular model cannot understand the motion signal quantitatively. Grottooli *et al.* [38] introduced a phase lag between two signals to evaluate the quality of the regenerated motion signal via the MCA model. The phase lag is determined based on the time delay between two signals to reach the maximum shape similarity. Table IV presents the RMSE, CC, phase lag of the translational motion sensation signal and maximum linear and angular displacement of the end-effector using LMPC-GA, LTV MPC-GA, NMPC-GA, and NMPC-CAS. RMSE and CC between two signals can be calculated as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (T_i - \hat{T}_i)^2}{n}} \quad (22.a)$$

$$\text{CC}(T, \hat{T}) = \frac{\sum_{i=1}^n (x_i - \bar{x})(T_i - \bar{T})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (T_i - \bar{T})^2}} \quad (22.b)$$

where n , T_i , and $\hat{T}_i$ are the number of samples, the i^{th} output, and the i^{th} predicted value, respectively. Also, x_i , $\bar{x}$, and $\bar{T}$ are the i^{th} input, mean of inputs, and mean of outputs, respectively.

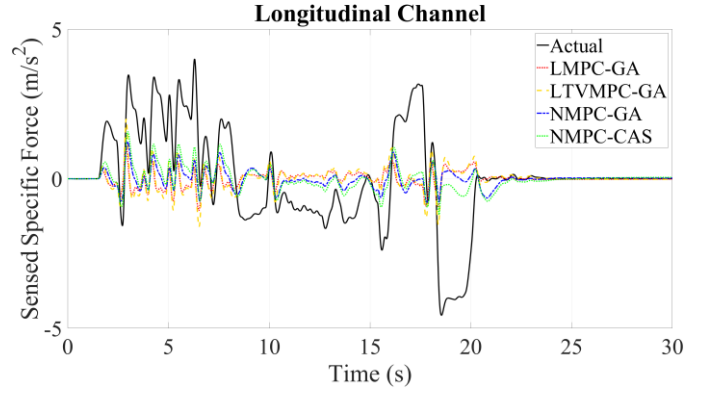


Fig. 7. The sensed specific force for linear time-invariant tuning via GA, time-varying tuning via GA, nonlinear tuning via GA and nonlinear tuning via cascade MPC-based MCAs along x -axis.

B. Validation

Fig. 7 depicts the sensation of linear motion signals for the real vehicle driver and the simulator user using LMPC-GA, LTV MPC-GA, NMPC-GA, and NMPC-CAS. The CC scores pertaining to the sensed linear acceleration signal via the simulator user as compared with that of a real vehicle driver using LMPC-GA, LTV MPC-GA, NMPC-GA and NMPC-CAS are 0.1292, 0.2868, 0.6936 and 0.8091, respectively. The shape-following factor using NMPC-CAS is 6.2624 times, 2.8211 times and 1.1665 times more than those of LMPC-GA, LTV MPC-GA, and NMPC-GA. It initially proves the efficiency of the linear time-varying MPC in regenerating high-fidelity motion signals for simulator users compared to linear time-variant MPC. However, this improvement is negligible compared with that of the NMPC which is the first main contribution of this paper. Also, the higher shape similarity between the motion sensation signal of the real vehicle's driver and simulator's user using the NMPC-CAS as compared with that of NMPC-GA proved the efficiency of the proposed cascade optimisation tuning technique.

Fig. 8 presents the sensation error of the linear motion signal between the real vehicle driver and the simulator user using LMPC-GA, LTV MPC-GA, NMPC-GA and NMPC-CAS. The RMSE rates of the sensed linear acceleration signals via the simulator user compared to those of the real vehicle driver are 1.6341, 1.6323, 1.4893, and 1.3311 (m/s^2) using LMPC-GA, LTV MPC-GA, NMPC-GA and NMPC-CAS, respectively. Based on these results, the dramatic improvement of both NMPC models tuned using GA and the cascade technique is

TABLE IV
THE RESULTS FOR THE LMPC-GA, LTV MPC-GA, NMPC-GA, AND NMPC-CAS ALONG LONGITUDINAL CHANNEL

Parameters	LMPC-GA	LTV MPC-GA	NMPC-GA	NMPC-CAS
RMSE (m/s^2)	1.6341	1.6323	1.4893	1.3311
CC	0.1292	0.2868	0.6936	0.8091
Phase Lag (Time-step)	13	12	6	6
Maximum Linear Displacement (m)	0.0705	0.1327	0.1445	0.1850
Maximum Angular Displacement (rad)	0.307	0.330	0.298	0.399

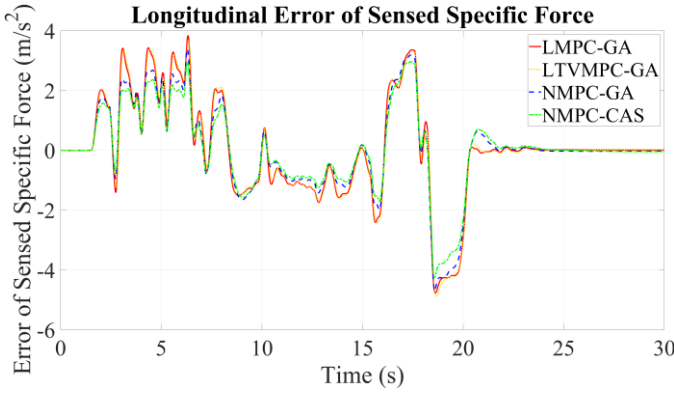


Fig. 8. The error of sensed specific force for linear time-invariant tuning via GA, time-varying tuning via GA, nonlinear tuning via GA and nonlinear tuning via cascade MPC-based MCAs along x -axis.

noticeable, compared to those of the linear MPC. Second, the results indicate that time-varying MPC, even when including the nonlinear dynamics model of the system, cannot generate more accurate motion signals than those of the nonlinear MPC. Since it is linear time-varying, the MPC linearises the model and the error increases during the linearisation process. As such, its RMSE rate increases gradually when calculating the motion signal. Third, the cascade optimisation technique can identify good weighting indices of the NMPC to reduce the RMSE rate of the sensed specific force compared with those of the GA-based tuning process.

Fig. 9.a-b represent the linear and angular displacements of the cockpit along the x -axis and $pitch$ -angle, respectively. The results prove that the linear time-varying MPC uses linear and angular motion better than the traditional linear time-invariant MPC. Because the linear time-varying MPC can include the actuators' constraints rather than those of the cockpit, it is not necessary to implement additional safety assumptions for the cockpit in the linear time-varying MPC. Comparatively, the NMPC can include the actuators' constraints better than the linear time-varying MPC because it is not necessary to linearise the nonlinear dynamical model of the plant in NMPC. The better usage of the cockpit constraints is obvious using NMPC-CAS compared to the other three previously mentioned methods.

The pod lengths are depicted in Fig. 10.a-c using LTVMPC-GA, NMPC-GA and NMPC-CAS, respectively. The pod lengths of LMPC-GA are not shown because there is an extra safety assumption to respect the actuator constraints. Based on Fig. 10.b-c, it is evident that cascade tuning can produce optimal weighting parameters for the NMPC and exploit the actuator constraints more efficiently than the GA-tuning method.

Also, the NMPC-CAS has a low online computational load compared with NMPC-GA. Fig. 11 shows the computational time of the MATMPC using GA and cascade optimisation algorithms. Based on Fig. 11, the RMSE of the NMPC-CAS and -GA are 1.7470 and 1.8665 (ms), respectively. This proves the lower computational load of the NMPC-CAS compared to NMPC-GA. The lower computational load of the mechanism is necessary for increasing the system's applicability.

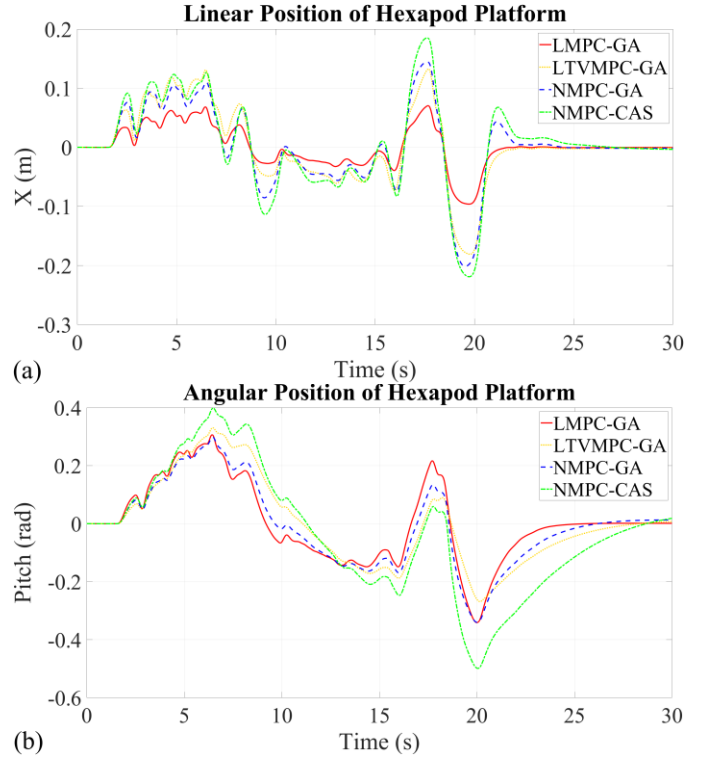


Fig. 9. Linear and angular displacement of the simulator using linear time-invariant tuning via GA, time-varying tuning via GA, nonlinear tuning via GA and nonlinear tuning via cascade MPC-based MCAs along (a) x -axis; (b) $pitch$ -angle.

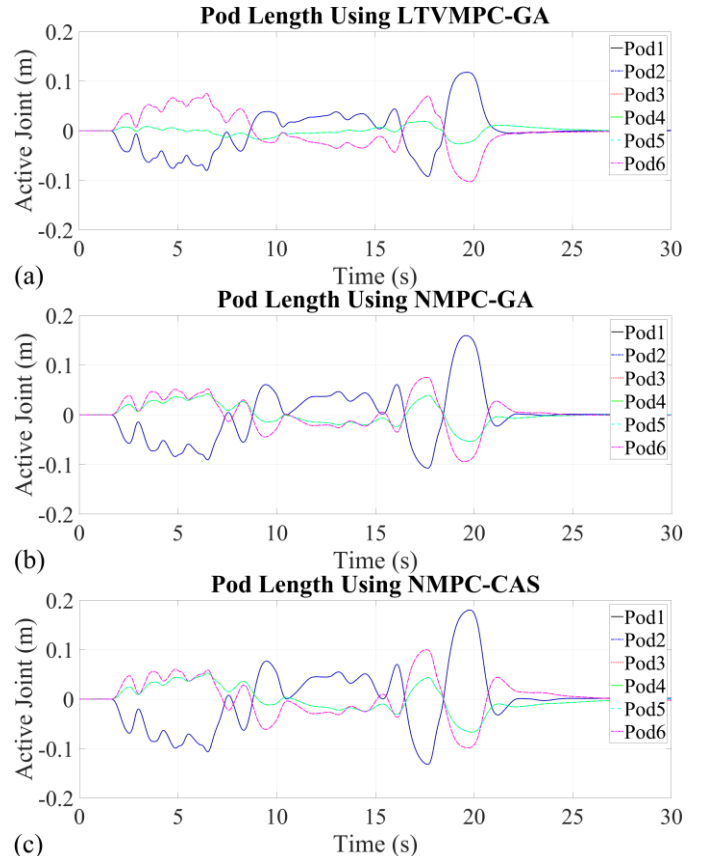


Fig. 10. Active joints' positions: (a) LTV MPC-based MCA tuning via GA; (b) NMPC-based MCA tuning via GA; (c) NMPC-based MCA tuning via cascade optimisation.

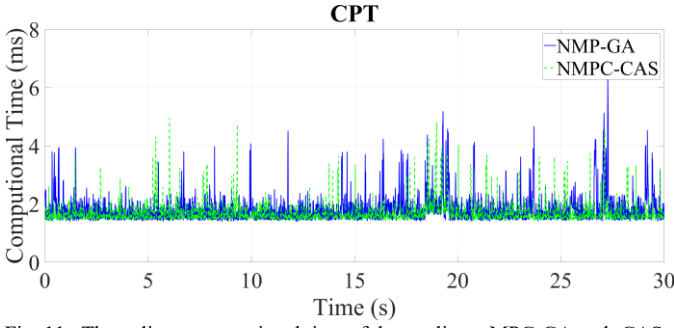


Fig. 11. The online computational time of the nonlinear MPC-GA and -CAS.

At the end, three different critical motion scenarios, including acceleration/deacceleration, sharp turning, and slalom are applied to the investigated methods of this study for further evaluation and to show the improvement of the newly proposed NMPC-CAS. The three investigated motion scenarios are shown in Fig. 12. It should be noted that acceleration/deacceleration and sharp turning lasts for 30 seconds of motion, while slalom was for 80 seconds. The results are shown in Table V, including RMSE, CC maximum linear and angular displacement of the end-effector using LMPC-GA and NMPC-CAS. Based on the represented results in Table V, NMPC-CAS is able to improve the CC between the sensed specific forces between the user of simulator and real vehicle driver compared with LMPC-GA using acceleration/deacceleration, sharp turning, and slalom motion scenarios with 33.49%, 33.25%, and 32.97%, respectively. In addition, RMSE of sensed specific force between the user of simulator and real vehicle driver with implementation of acceleration/deacceleration, sharp turning, and slalom motion scenarios using NMPC-CAS are decreased 44.86%, 45.86%, and 55.58% compared with LMPC-GA. NMPC-CAS is able to use more linear and angular displacement of the end-effector using all three different motion scenarios compared with LMPC-GA because of the consideration of actuators and passive joints limitations based on the represented results in fifth and sixth row of Table V.

The aforementioned results prove the efficiency of the NMPC-based MCA tuned by a two-step cascade optimisation technique, including GA and PSO. In addition, the NMPC has been compared with linear MPCs for the first time in this

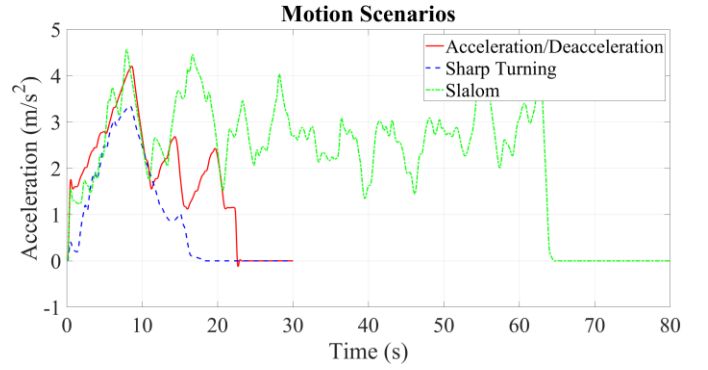


Fig. 12. Three critical motion scenarios including acceleration/deacceleration, sharp turning, and slalom.

research, and improvement of the NMPC is shown with a reduction of the RMSE rate and an increase of the CC score between the sensed linear acceleration signals via the real vehicle driver and the simulator user.

V. CONCLUSION

The generated driving signals from a real vehicle, including the angular velocity and linear acceleration, cannot be implemented in a driving motion platform system because of the position, velocity and acceleration constraints of the joints. A motion cueing algorithm is used to regenerate feasible and realistic driving motion signals to produce realistic driving experiences for the motion platform user while satisfying the motion platform constraints. Recently, linear model predictive control has been used in most motion cueing algorithm models because of its ability to keep the cockpit inside the driving motion platform boundaries based on the end-effector coordinate system. However, the joints of the manipulators are the main reason for the cockpit constraints, and it is not possible to exploit the entire workspace without considering the actuator constraints. The linear time-varying and nonlinear model predictive control models have been introduced to overcome this disadvantage of linear model predictive control. However, the efficiency of linear time-varying model predictive control is not as good as that of nonlinear model predictive control. As the first main contribution of this paper, a nonlinear model predictive control-based motion cueing algorithm using the MATMPC package is designed and developed. The algorithm is developed to increase the efficiency of the fidelity of the motion cues for the driving motion platform's user compared to real vehicle's driver with a low computational load. However, the tuning process of the nonlinear model predictive control is not as easy as its linear counterpart. The introduction of a cascade optimisation technique reduces the duration of the tuning process. It also offers better results as compared to those of a one-layer Multiobjective genetic algorithm. Since many weighting parameters need to be appropriately tuned to increase the nonlinear model predictive control's efficiency, one-layer optimisation methods, such as genetic algorithms, may not be able to identify the optimal weighting indices. The proposed cascade method has been designed and developed using Simulink/MATLAB and compared to the linear model predictive control, linear time-varying model predictive control

TABLE V
THE RESULTS FOR THE LMPC-GA AND NMPC-CAS ALONG LONGITUDINAL CHANNEL USING ACCELERATION/DEACCELERATION, SHARP TURNING AND SLALOM MOTION SCENARIOS

Parameters	LMPC-GA			NMPC-CAS		
	Ac/De	Tur	Sla	Ac/De	Tur	Sla
RMSE (m/s ²)	2.1945	1.5706	2.6796	1.2101	0.8503	1.1902
CC	0.6124	0.6536	0.5642	0.8175	0.8709	0.7502
Maximum Linear Displacement (m)	0.1031	0.0824	0.1115	0.1830	0.1514	0.2283
Maximum Angular Displacement (rad)	0.3502	0.4100	0.3604	0.4120	0.4946	0.4389

Ac/De: Acceleration/Deacceleration; Tur: Sharp Turning; Sla: Slalom.

and nonlinear model predictive control methods. All are tuned using genetic algorithms. The results of this research have validated the efficiency of the proposed method, which includes the nonlinear dynamics model of motion cueing and a proper tuning method. The improvement of the CC score and reduction of the RMSE rate pertaining to the motion sensation along the longitudinal channel indicate the viability of the proposed method. It should be noted that since the proposed method can decrease the computational time of the tuning process, it can be employed as a fast adjustable method for implementing the highly-efficient nonlinear model predictive control-based motion cueing algorithm for each user of the driving motion platform. Even though the proposed model has several advantages, it has some challenges to be addressed in future studies. First, the proposed method will be investigated using large data sets to validate its performance under a wide range of conditions. Second, the GA and PSO are separately applied in two different steps, which is akin to a single decision process of choosing the optimal values. To remedy this problem, group decision-making (also called collaborative decision-making) will be investigated to improve the optimal weighting parameters. Third, only numerical studies have been conducted here; however, future work will partially focus on the design and development of an experimental setup to validate the theoretical findings. In addition, the implementation of validation for all four channels, including longitudinal, lateral, heave, and yaw will be investigated. Last, parallel and incremental machine learning algorithms [39] and fuzzy logic [40] can be used to better and faster find the optimal weighting parameters. The results of these suggested studies will be published in the future and compared to the current state-of-the-art.

REFERENCES

- [1] M. R. C. Qazani, H. Asadi, S. Mohamed, and S. Nahavandi, "Prepositioning of a land vehicle simulation-based motion platform using fuzzy logic and neural network," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10446-10456, 2020.
- [2] H. Asadi, T. Bellmann, M. C. Qazani, S. Mohamed, C. P. Lim, and S. Nahavandi, "A Novel Decoupled Model Predictive Control-based Motion Cueing Algorithm for Driving Simulators," 2023.
- [3] H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 2, pp. 238-254, 2016.
- [4] M. Chalak Qazani, "Modelling and simulation of a motion cueing algorithm using prediction and computational intelligence techniques," Deakin University, 2020.
- [5] M. R. C. Qazani, S. Pedrammehr, and M. J. Nategh, "A study on motion of machine tools' hexapod table on freeform surfaces with circular interpolation," *The International Journal of Advanced Manufacturing Technology*, vol. 75, no. 9-12, pp. 1763-1771, 2014.
- [6] M. R. C. Qazani *et al.*, "Kinematic analysis and workspace determination of hexarot-a novel 6-DOF parallel manipulator with a rotation-symmetric arm system," *Robotica*, vol. 33, no. 8, pp. 1686-1703, 2015.
- [7] M. R. C. Qazani, H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "An optimal washout filter for motion platform using neural network and fuzzy logic," *Engineering Applications of Artificial Intelligence*, vol. 108, p. 104564, 2022.
- [8] K. Money, "Motion sickness," *Physiological reviews*, vol. 50, no. 1, pp. 1-39, 1970.
- [9] B. Conrad and S. F. Schmidt, "A study of techniques for calculating motion drive signals for flight simulators," 1971.
- [10] R. J. Telban and F. M. Cardullo, "Motion cueing algorithm development: Human-centered linear and nonlinear approaches," *NASA TechReport CR-2005-213747*, 2005.
- [11] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "An Optimal Motion Cueing Algorithm Using the Inverse Kinematic solution of the Hexapod Simulation Platform," *IEEE Transactions on Intelligent Vehicles*, 2021.
- [12] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.
- [13] R. V. Parrish, J. E. Dieudonne, R. L. Bowles, and D. J. Martin Jr, "Coordinated adaptive washout for motion simulators," *Journal of aircraft*, vol. 12, no. 1, pp. 44-50, 1975.
- [14] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, "Model-based predictive motion cueing strategy for vehicle driving simulators," *Control Engineering Practice*, vol. 17, no. 9, pp. 995-1003, 2009.
- [15] M. R. C. Qazani, H. Asadi, S. Khoo, and S. Nahavandi, "A linear time-varying model predictive control-based motion cueing algorithm for hexapod simulation-based motion platform," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. Early Access, pp. 1-15, 2019.
- [16] M. R. C. Qazani, H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "A Time-Varying Weight MPC-Based Motion Cueing Algorithm for Motion Simulation Platform," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [17] M. Bruschetta, F. Maran, and A. Beghi, "A nonlinear, mpc-based motion cueing algorithm for a high-performance, nine-dof dynamic simulator platform," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 686-694, 2016.
- [18] M. Katliar, F. M. Drop, H. Teuffel, M. Diehl, and H. H. Bühlhoff, "Real-Time Nonlinear Model Predictive Control of a Motion Simulator Based on a 8-DOF Serial Robot," in *2018 European Control Conference (ECC)*, 2018: IEEE, pp. 1529-1535.
- [19] Y. R. Khuro, Y. Zheng, M. Grottoli, and B. Shyrokau, "MPC-Based Motion-Cueing Algorithm for a 6-DOF Driving Simulator with Actuator Constraints," *Vehicles*, vol. 2, no. 4, pp. 625-647, 2020.
- [20] A. Lamprecht, J. Haecker, and K. Graichen, "Constrained motion cueing for driving simulators using a real-time nonlinear MPC scheme," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018: IEEE, pp. 7466-7471.
- [21] M. R. C. Qazani *et al.*, "A New Prepositioning Technique of a Motion Simulator Platform Using Nonlinear Model Predictive Control and Recurrent Neural Network," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [22] S. Casas, I. Coma, C. Portalés, and M. Fernández, "Towards a simulation-based tuning of motion cueing algorithms," *Simulation Modelling Practice and Theory*, vol. 67, pp. 137-154, 2016.
- [23] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Optimizing model predictive control horizons using genetic algorithm for motion cueing algorithm," *Expert Systems with Applications*, vol. 92, pp. 73-81, 2018.
- [24] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Multiobjective and interactive genetic algorithms for weight tuning of a model predictive control-based motion cueing algorithm," *IEEE transactions on cybernetics*, vol. 49, no. 9, pp. 3471-3481, 2018.
- [25] P. Zhang, T. D. Bui, and C. Y. Suen, "A novel cascade ensemble classifier system with a high recognition performance on handwritten digits," *Pattern Recognition*, vol. 40, no. 12, pp. 3415-3429, 2007.
- [26] S.-C. Neoh, N. Morad, C.-P. Lim, and Z. A. Aziz, "A Layered-Encoding Cascade Optimization Approach to Product-Mix Planning in High-Mix-Low-Volume Manufacturing," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 133-146, 2009.
- [27] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "Matmpc-a matlab based toolbox for real-time nonlinear model predictive control," in *2019 18th European Control Conference (ECC)*, 2019: IEEE, pp. 3365-3370.
- [28] H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "A review on otolith models in human perception," *Behavioural brain research*, vol. 309, pp. 67-76, 2016.

- [29] H. Asadi, S. Mohamed, C. P. Lim, S. Nahavandi, and E. Nalivaiko, "Semicircular canal modeling in human perception," *Reviews in the Neurosciences*, vol. 28, no. 5, pp. 537-549, 2017.
- [30] M. R. C. Qazani, S. Pedrammehr, and M. J. Nategh, "An investigation on the motion error of machine tools' hexapod table," *International Journal of Precision Engineering and Manufacturing*, vol. 19, no. 4, pp. 463-471, 2018.
- [31] M. R. C. Qazani, H. Asadi, S. Pedrammehr, and S. Nahavandi, "Performance analysis and dexterity monitoring of hexapod-based simulator," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018: IEEE, pp. 226-231.
- [32] J. Andersson, "A general-purpose software framework for dynamic optimization," PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical ..., 2013.
- [33] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.
- [34] J. Kennedy and R. Eberhart, "Particle swarm optimization (PSO)," in *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, 1995, pp. 1942-1948.
- [35] S. Casas-Yrurzum, C. Portalés-Ricart, P. Morillo-Tena, and C. Cruz-Neira, "On the Objective Evaluation of Motion Cueing in Vehicle Simulations," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [36] D. Arjoni *et al.*, "Experimental Evaluation of the Human Performance on a RoboticFlight Simulator based on FOQA Parameters," in *Aerospace Technology Conference 2016, Swedish Aerospace Technology in a Globalised World, October 11-12, Solna, Stockholm, Sweden*, 2016, vol. 1, pp. 1-11.
- [37] H. Asadi, S. Mohamed, and S. Nahavandi, "Incorporating human perception with the motion washout filter using fuzzy logic control," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 3276-3284, 2015.
- [38] M. Grottolli, D. Cleij, P. Pretto, Y. Lemmens, R. Happee, and H. H. Bühlhoff, "Objective evaluation of prediction strategies for optimization-based motion cueing," *Simulation*, vol. 95, no. 8, pp. 707-724, 2019.
- [39] M. R. C. Qazani, H. Asadi, C. P. Lim, S. Mohamed, and S. Nahavandi, "Prediction of motion simulator signals using time-series neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 5, pp. 3383-3392, 2021.
- [40] M. R. C. Qazani, H. Asadi, T. Bellmann, S. Mohamed, C. P. Lim, and S. Nahavandi, "Adaptive Washout Filter Based on Fuzzy Logic for a Motion Simulation Platform With Consideration of Joints' Limitations," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12547-12558, 2020.



Mohammad Reza Chalak Qazani received the Bachelor of Engineering in manufacturing and production from University of Tabriz, Tabriz, Iran in 2010 and the master's degree in robotic and mechanical engineering from the Tarbiat Modares University, Tehran, Iran, in 2013. He received the Ph.D. degree in modelling and simulation of a motion cueing algorithm using prediction and computational intelligence techniques from the Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Australia, in 2021. He was an Alfred Deakin Postdoctoral Research Fellow in the IISRI for two years working in the areas of model predictive control, motion cueing algorithm and soft computing controllers.

He is currently an Assistant Professor in the Faculty of Computing and Information Technology (FoCIT), Sohar University, Sohar, Sohar 311, Oman. His teaching areas are data structure and algorithm, enterprise resource planning modelling and implementation, modelling and visualization, computer architecture, introduction to artificial intelligence, and advance machine learning.



Houshyar Asadi received the Bachelor of Engineering degree (First Class Hons.) in electrical control systems in 2008, and Master degree in Industrial Electronic and Control Engineering from University of Malaya in 2012. He received the Ph.D. degree in human perception-based washout filtering using Artificial Intelligence (AI)

from the Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Australia, in 2015.

He is currently an associate professor at IISRI and leading the research area of motion simulator technologies. His current research interests include AI, motion simulator technologies, motion cueing algorithms, control, and human factors in virtual environments.



Yutao Chen received his Ph.D. in 2018 at the Department of Information Engineering, University of Padova, Italy. From 2019–2020, he was a post-doc researcher at the Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands.

He is now an Associate Professor at the College of Electrical Engineering and Automation, Fuzhou University, China. His research interests include model predictive control algorithms, intelligent transportation systems and human-in-the-loop control techniques.



Moloud Abdar holds a Ph.D. degree in Engineering (Computer, Machine Learning) from the Institute for Intelligent Systems Research and Innovation (IISRI) at Deakin University, Australia, which he obtained in 2022. Prior to that, he completed his Master's degree in Computer Science from the University of Aizu, Aizu, Japan, in 2018, and earned his Bachelor's degree in Computer Engineering from Damghan University, Iran, in 2015.

He currently serves as an Alfred Deakin Postdoctoral Research Fellow with the IISRI at Deakin University, Australia. His research expertise encompasses a wide range of areas, including machine learning, deep learning, computer vision, medical image analysis, sentiment analysis, explain ability, and uncertainty quantification.



Mansour Karkoub received B.S., M.S., and PhD degrees in Mechanical Engineering from the University of Minnesota, Twin Cities. In 2003, he received an H.D.R. from the University of Versailles, France. He started his academic career as a lecturer at the University of Minnesota and researcher at the Minnesota Super Computer Institute. In 2002, he joined the French "National Institute for Research in

Computer Science and Controls (INRIA)” in Rocquencourt, France as a professor and researcher in smart transportation systems. In 2009, Dr. Karkoub joined Texas A&M University/Texas A&M University at Qatar as professor of mechanical engineering.

He is currently chair of the Mechanical Engineering Department and Michael E. and Patricia P. Aldridge Endowed Chair in Engineering at Lamar University in Beaumont, Texas. Dr. Karkoub is a fellow of ASME, fellow of the Institute of Engineering Technology (IET), and a senior member of IEEE. His interests include robust control, autonomous ground and underwater robotic systems, self-driving vehicles, mechatronics, artificial intelligence, as well as engineering education.

Australia (FIEAust), the Institution of Engineering and Technology (FIET). Saeid is a Fellow of the Australian Academy of Technology and Engineering (ATSE).



Shady Mohamed received the B.Sc. and M.Sc. degrees in information technology from Cairo University, Giza, Egypt, in 2000 and 2003, respectively, and the Ph.D. degree in control theory from Deakin University, Geelong, VIC, Australia, in 2009. He is currently an associated professor with the Institute for Intelligent

Systems Research and Innovation (IISRI), Deakin University, Australia.

He is interested in interdisciplinary research involving signal processing, control theory, human biodynamics, haptics and medical imaging.



Chee Peng Lim received his PhD degree from the University of Sheffield, UK, in 1997. His research interests include computational intelligence-based systems for pattern recognition, data mining, fault detection, condition monitoring, decision support, and process optimization. He has published over 450 technical papers in these areas.

He is currently a professor at Institute for Intelligent Systems Research and Innovation, Deakin University, Australia.



Saeid Nahavandi received a Ph.D. from Durham University, U.K. in 1991. He is an Alfred Deakin Professor, Pro Vice-Chancellor, Chair of Engineering, and the Founding Director of the Institute for Intelligent Systems Research and Innovation at Deakin University. His research interests include modelling of complex systems, robotics and haptics. He has published over 900 scientific papers in various international journals and conferences.

Professor Nahavandi is Editor-In-Chief: IEEE SMC Magazine, the Senior Associate Editor: IEEE Systems Journal, Associate Editor of IEEE Transactions on Systems, Man and Cybernetics: Systems, and IEEE Press Editorial Board member. Professor Nahavandi is a Fellow of IEEE (FIEEE), Engineers