A New Prepositioning Technique of a Motion Simulator Platform Using Nonlinear Model Predictive Control and Recurrent Neural Network

Mohammad Reza Chalak Qazani[1], Houshyar Asadi[1], *Member, IEEE*, Li Zhang[2], Farzin Tabarsinezhad[3], Shady Mohamed[1], *Member, IEEE*, Chee Peng Lim[1], Saeid Nahavandi[1,4], *Fellow, IEEE*

***Abstract*—The motion cueing algorithm (MCA) is the main algorithm in motion simulators in charge of generating vehicle motions within the platform's constraints. The classical washout filter is one of the popular types of MCA, which is used in air and land vehicle motion simulators. The fixed home position of the simulator platform is always cogitated in the MCA to washout the motion simulator after generating each motion. Unfortunately, considering the fixed home position reduces the efficient consumption of the workspace in the linear directions. The linear motion of the motion simulator is due to the production of the high-pass frequency part of the motion scenarios. Prepositioning is used to tackle this assumption by varying the home position rather than the fixed position. The linear motion limitations of the motion simulator can virtually be enlarged using the prepositioning method. The efficient regeneration of the high-pass motion cues using a new propositioning technique is the main goal of this study to increase the motion realism of the simulator and remove any false motion cues due to the platform limitations. The proposed model utilised the recurrent neural network (RNN) to estimate the motion scenario along the prediction horizon. The nonlinear model predictive control (MPC) uses the estimated motion signals to extract the best optimal off-centre position of the motion simulator platform. The newly developed prepositioning technique is developed in the simulation environment of MATLAB to validate the proposed technique in terms of efficiency and applicability. The outcomes prove the capability of the proposed technique against the recently developed prepositioning technique using fuzzy logic and RNN.**

***Index Terms*—prepositioning, motion cueing algorithm, nonlinear model predictive control, neural network, motion simulator platform.**

I. Introduction

THE motion cueing algorithm (MCA) is used to reproduce the motion cues for the motion simulator platform's user based on the driving signals produced by the vehicle user in the virtual environment [1-3]. In general, implementing the virtual vehicle motion signals to the motion simulator is not possible because of the workspace boundaries. The MCA aims to regenerate the high-fidelity motion signals for the motion simulator user, identical to what a real driver experience in a real vehicle. The low-fidelity motion signals may cause motion sickness. The experienced motion sickness caused by false motion cues can force the user to stop using the simulator.

The human vestibular organ oversees sensing the driving motion signal and stabilising the body's motion. It consists of semicircular systems and otolith organs which is situated inside the inner ears [4]. The otolith organs perceive linear motion, while the semicircular systems perceive the angular motions. It is very important to respect the threshold unit of the semicircular systems and otolith organs to avoid motion sickness for the motion simulator user.

Conrad and Schmitt [5] invented the first MCA known as a classical washout filter, and Reid and Nahon [6] improved the classical washout filter used in the aeroplane simulator platform. The classical washout filter is one of the common shapes of the MCA, which is very applicable, easy to tune, low computational load and extremely safe. Unfortunately, the classical washout filter has poor usage of the workspace. The MCA is developed by different researchers since the introduction using different techniques that can be categorized in classical, adaptive [7-9], optimal [10-12] and model predictive control (MPC) [13-15] sectors. The variation of the home position of the motion simulator extends the linear displacement of the motion simulator known as prepositioning technique.

The prepositioning technique is achieved using the regeneration of the motion compensation signal, which works in parallel with a classical washout filter. This method can be categorised as an adaptive washout filter. The initial prepositioning technique is introduced by Weiß [16] based on

(*Corresponding author: Mohammad Reza Chalak Qazani*)
The authors of 1 are with the Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, VIC 3125, Australia (e-mail: m.r.chalakqazani@gmial.com; houshyar.asadi@deakin.edu.au; shadym@deakin.edu.au; chee.lim@deakin.edu.au; saeid.nahavandi@deakin.edu.au).
The author of 2 is with the Department of Computer Science, Royal Holloway, University of London, Surrey, UK (li.zhang@rhul.ac.uk).
The author of 3 is with the Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran (farzin.tabarsi@ut.ac.ir).
The author of 4 is with the Harvard Paulson School of Engineering and Applied Sciences, Harvard University, Allston, MA 02134 USA (e-mail: nahavandi@seas.harvard.edu).

the motion indexes of the end-effector without forecasting the motion signals via the virtual vehicle user. But this method [16] is relatively weak, facing unpredictable motion signals or sudden changes such as sharp turns or changing the lane. The reason for reproducing false motion cues is due to the non-prediction of the motion signals of the virtual vehicle along the prediction horizon. Chapron and Colinot [17] used the information of the road to anticipate the next action of the driver. However, analytical analysis of the road is not possible for every motion scenario. Hansson et al. [18] used the information of the road and a vehicle to estimate the motion signal along with longitudinal and lateral modes. Unfortunately, they did not consider the current position of the end-effector, which could dramatically decrease the efficiency of their proposed technique. Recently, Qazani et al. [19] proposed the new propositioning technique to tackle all disadvantages of the previous methods [16-18], such as not considering the current end-effector positions or neglecting the estimation of the motion signals. Qazani et al. [19] utilised the feedforward neural network (NN) to estimate the driving scenarios along the prediction horizon. The appropriate prepositioning of the end-effector is calculated based on the high-pass filter of the estimated motion signals and the current position of the end-effector. However, their proposed method [19] uses the fuzzy logic controller to calculate the new home position of the motion simulator. However, they did not use the optimal off-centre position of the motion simulator since the work did not take into account any optimisation methods such as optimal or model predictive methods. In addition, the high-pass filtering of the predicted motion signals using feedforward NN is very time-consuming, and it will affect the real-time implementation of the method. The linear model predictive control (MPC) was used in MCA by Dagdalan et al. [20] for the first time to consider the workspace limitation of the simulator in regeneration of the motion cues for the user simulator based on the real vehicle motion signals. Qazani et al. [21] used the terminal condition in linear MPC-based MCA to increase the efficiency of the system facing abrupt motion signals. Also, the nonlinear MPC was recently employed in the MCA domain. As an example, Katliar et al. [22] employed the RTI method [23] to develop an NMPC-based MCA model for an 8-DoF serial DMP system with cogitation of the discretised inverse kinematic solution of the manipulator. Lamprecht et al. [24] used GRAMPC to design an NMPC-based MCA model for a 7-DoF hexapod DMP system. Khusro et al. [25] developed the NMPC-based MCA for a 6-DoF hexapod DMP using the ACADO toolkit [26] with adaptive weight tuning to smooth out the motion of the end-effector.

The main motivation of this study can be divided into two parts, i.e. the lower computational load and higher accuracy, in comparison with those of the recently introduced prepositioning algorithm [19] due to using recurrent neural network (RNN) and nonlinear MPC. Also, the main contribution of this study is to increase the perceived motion fidelity for the user of the simulator via better generation of linear motion of the end-effector based on the current position of the end-effector and the prediction of driving motion signals. The MPC is substituted with a high-pass filter and fuzzy logic system used in [4] to recalculate the off-centre position to achieve a lower computational load. Also, an RNN is used to estimate the driving motion signal with the attempt to achieve higher accuracy rates in comparison with those of the traditional shallow feedforward NN [19], as the second motivation of this work. It should be noted that RNN is able to reach better results as compared with feedforward NN for tackling time-series driving motion signals [27]. In addition, the nonlinear MPC method calculates the motion simulator's optimal off-centre position with higher accuracy rates than those of the fuzzy logic system adopted in [4]. The RNN model is used to predict the driving motion signal, while the nonlinear MPC method calculates the optimal off-centre position based on the current position of the end-effector as well as the predicted motion signal via the RNN. It should be noted that the nonlinear MPC method is designed and developed using MAT-MPC [28] to decrease the computational cost of the system for facilitating real-time applications. Using the proposed method, the linear acceleration signal is regenerated as the motion compensation signal in order to minimise the motion perception error for the user pertaining to the motion simulator platform via the highly efficient usage of the linear workspace limitations.

The methodology of the newly developed prepositioning technique is described in Section II. It includes the description of the motion simulator platform, a human vestibular system, the classical washout filter, as well as the proposed prepositioning algorithm. The newly developed method is verified and validated in Section III utilising MATLAB/Simulink. The conclusions of the study are remarked in Section IV.

II. Methodology

This Section describes the employed motion simulator, a mathematical model of the human vestibular model, classical washout filter, and proposed prepositioning technique using RNN and nonlinear MPC.

A. Motion Simulator Platform

The MATLAB/SimMechanic model of the parallel Stewart-Gough platform is used to reproduce the motion cues in the simulation study of the proposed method in this work [29, 30]. It is a parallel-based motion simulator with the ability to generation 6-DoF motions, accurate kinematic solution and high acceleration. Wang et al. [31] employed a parallel-based motion simulator platform to investigate the efficiency of their industrial developed collision warning algorithm. Nillson et al. [32] employed the motion simulator platform to comprehensively compare the lane change for long vehicle manual and autonomous vehicles. Casas et al. [33] objectively compared the different motion cueing algorithms using parallel- and serial-based motion simulator platforms. Qazani et al. [11] consider the limitations of the parallel-based motion simulator platform in the development of the optimal MCA. The Simulink model uses a multi-body system technique to solve the parallel robot's inverse kinematic and dynamic problem with high resolution. The parallel-based motion simulator platform can be

used in various sectors such as training the air/land vehicle users, virtual prototyping, study of motion sickness, etc. The dynamical models of the intended vehicles can be imported into the motion simulator platform to test the vehicle's driving experience before manufacturing those models. It should be noted that the workspace boundaries of the investigated Stewart-Gough platform along x-axis are ±0.6 metres and pitch-angle ±25 degrees. Also, the velocity limitations of the investigated platform along the *x*-axis and pitch-angle are 1.3 m/s and 61 degrees/s, respectively. Lastly, the acceleration limitations of the investigated platform along the *x*-axis and pitch-angle are 3.3 m/s^2 and 600 degrees/s^2, respectively.

B. Human Vestibular Model

The driving motion signals of the real vehicle refer to the regenerated motion signals of the vehicle recorded via the driver using the steering wheel, gas pedal and brake pedal in a virtual vehicle dynamic environment. It should be noted that these motions are what the simulator driver is supposed to feel using the simulator; that is why we called them the driving motion signal of the real vehicle or motion reference. These signals need to be tracked and regenerated by a motion simulator to provide the most realistic motion feeling to the simulator user, similar to what they can experience while driving a real vehicle.

The motion perception model comprises three systems, including the auditory, vestibular and visual systems. The vestibular system senses the rotational and translational motions [34-36], shown in Fig. 1.

Telban and Cardullo [34] and Asadi et al. [35] reviewed various mathematical models of the human otolith organs, and they found the best model to link the sensed specific force of the motion simulator's user $\hat{f}$ with the stimulus-specific force f as follows:

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

where K_{OTO} is the static sensitivity of the otolith organs. Also, τ_a, τ_L and τ_s are the otolith time constants.

Also, Telban and Cardullo [34] and Asadi et al. [36] extracted the best transfer function filter of the semicircular canals as follows:

$$\frac{\hat{\omega}}{\omega} = \frac{G_{SCC}\tau_1\tau_a s^2(1+\tau_L s)}{(1+\tau_a s)(1+\tau_1 s)(1+\tau_2 s)} \quad (2)$$

where $\hat{\omega}$ and ω are the sensed angular velocity and applied angular velocity. Also, G_{SCC} is the angular velocity threshold, and τ_1, τ_2, τ_a and τ_L are the semicircular time constants.

Then, using Eq. (1) and Eq. (2), the sensed specific force and sensed angular velocity based on the applied linear acceleration and angular velocity signals can be calculated for the user of the motion simulator and a real vehicle. It should be noted that the stimulus-specific force along three directions regarding the angular displacement of the motion simulator head along the horizons can be found as below:

$$\begin{cases} f_x \triangleq a_x + g\theta \\ f_y \triangleq a_y - g\phi \\ f_z \triangleq a_z - g \end{cases} \quad (3)$$

where a_x, a_y and a_z are the linear acceleration signal along the *x*-, *y*- and *z*-axis. ϕ and θ are *roll*- and *pitch*-angle, and $g = 9.81$ m/s^2 is the gravity. The angular displacement of the end-effector should be less than 30 degrees using Eq. (3) to use the simplified version of sine and cosine functions.

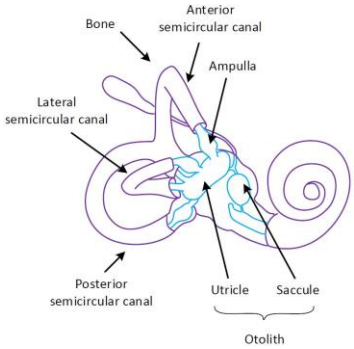


Fig. 1. The schematic structure of the human vestibular system.

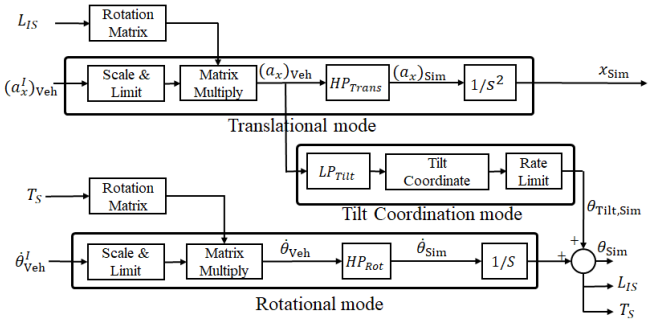


Fig. 2. The longitudinal mode of classical washout filter.

C. Classical Washout Filter

The structure of the classical washout filter is shown in Fig. 2, along with longitudinal mode. The longitudinal mode consists of three sub-modes: translational, tilt coordination, and rotational modes. Based on Fig. 2, the linear acceleration signal of the virtual vehicle in the world frame $(a_x^I)_{\text{Veh}}$ is the input signal of the translational mode. Also, the angular velocity signal of the virtual vehicle in the world frame $\dot{\theta}^I_{\text{Veh}}$ is the input signal of the rotational mode. There is a scale and limit block in a translational and rotational mode which should be chosen based on the workspace limitations of the motion simulator platform. These motion signals should be transformed from the world frame to the vehicle seat frame via multiplying the rotation matrix, including L_{IS} and T_S for translational and rotational modes, respectively. The calculated linear acceleration signal $(a_x)_{\text{Veh}}$ in the seat frame of the virtual vehicle is employed as an input signal of the high-pass HP_{Trans} (second-order) and low-pass LP_{Tilt} (second-order) filters to calculate the linear acceleration $(a_x)_{\text{Sim}}$ and tilt angle $(\theta_{Tilt})_{\text{Sim}}$ of the end-effector. Also, the high-pass filter of the rotation mode is first-order. These high-pass and low-pass filters can be defined as:

$$HP_{\text{trans}} = \frac{s^2}{s^2+2\xi\omega_n s+{\omega_n}^2} \quad (4.a)$$

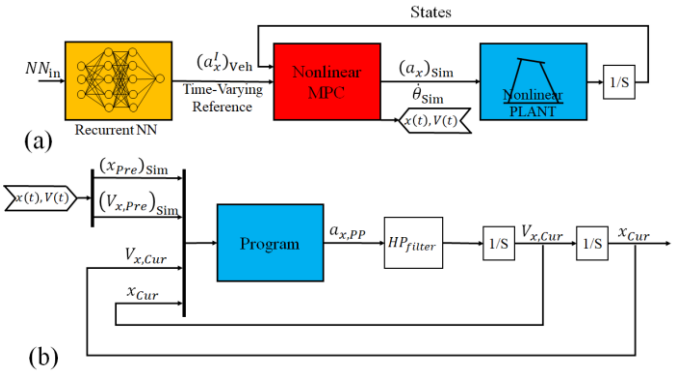


Fig. 3. The structure of the newly developed prepositioning classical washout filter using NN and nonlinear MPC; (a): The proposed nonlinear MPC using the output of RNN as a reference signal to calculate the linear displacements and velocities of the platform along prediction horizon; (b): The decision block program to calculate the prepositioning linear acceleration signal to increase the fidelity of the regeneration motion signals for the simulator's user.

$$LP_{\text{tilt}} = \frac{{\omega_n}^2}{s^2+2\xi\omega_n s+{\omega_n}^2} \quad (4.b)$$

$$HP_{\text{rot}} = \frac{s}{s+\omega_n} \quad (4.c)$$

where ξ and ω_n are constant parameters of the filters, including damping ratios and cut-off frequencies.

The proposed prepositioning algorithm using nonlinear MPC and RNN in the following subsection should be implemented in the translational mode of the investigated classical washout filter.

D. Prepositioning Using Nonlinear MPC and RNN

The schematic structure of the newly developed prepositioning method is shown in Fig. 3. It consists of the RNN, nonlinear MPC and a decision block to calculate the prepositioning along with the longitudinal mode. The idea of using state-of-art technology such as NN and MPC reaches the accurate results compared with traditional methods in lots of area such as pose estimation [37], image detection [38], electronic [39], emotion detection [40], and text detection [41].

Recurrent Neural Network: An RNN treats the input data in a sequential or time-series format. The network has been widely used in temporal or ordinal problems such as natural language processing, speech recognition, language translation, and image captioning. The practical applications which are developed using the RNN can be exampled as voice search, Google Translate and Siri. The RNN uses the training data to learn from the existing instance records like other phases of NN forms such as feedforward NN. The most distinguished part of the RNN compared with feedforward NN is a memory cell that can hold the information from initial inputs to affect the current input and output. The results of RNN depend on the initial inputs within the sequence, while the inputs and outputs are considered independently using feedforward NN. Another distinguishable part of the RNN compared with feedforward NN is the parameter sharing capacity across each network layer. It means that the RNN has the same weights across each network layer, while the feedforward NN has different weights

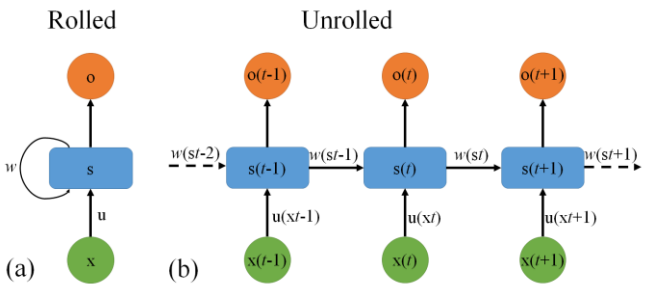


Fig. 4. The structures of the RNN (a) Rolled version; (b): Unrolled version.

across each node. These weights are determined through the gradient descent and processes of backpropagation to facilitate reinforcement learning. As a result, an RNN can reach better results in the presence of arbitrary input sequences than those of the feedforward NN.

Fig. 4. a-b shows the RNN's schematic rolled and unrolled structures. Each structure consists of input, hidden and output layers at the t^{th} time of the training, respectively. The memory block can be described as the hidden states of the network in the unrolled version of the RNN. Then, the memory of the network can be generated using the previous memory and the current input of the network as indicated below:

$$s(t) = f(u_{xt} + w_{st-1}) \quad (5)$$

where s_t, u_{xt} and w_{st} are the hidden layer, input and memorised states of the network, respectively.

The leverage backpropagation through time (BPTT) is used to calculate the gradients of the network [42, 43]. It should be noted that the gradients of the BPTT are a bit different from the traditional backpropagation because it is responsible for handling the sequential data. However, the principles of BPTT are similar to those in other NNs, as the model makes attempts to correct itself via calculation of the error from its output layer to its input layer. The difference from traditional backpropagation is the strategy of calculating the error. The error is the sum of the errors at each time step using BPTT since RNN shares the parameters across each layer.

In this study, the driving motion signal of the virtual vehicle (linear acceleration signal) is the input of the RNN. The network sequences the history of the data using 50 time-step delays as the input, while the next time-step linear acceleration signal is the output of the network. Then, the driving motion signal can be predicted using the prediction model after training the RNN. It happens via calculation of the linear acceleration of the next time step and substituting this value as the current linear acceleration signal of the virtual vehicle, in order to calculate the second future linear acceleration signal. This strategy is followed until the driving motion signal is predicted along the prediction horizon. Using the RNN, the driving motion signal of the virtual vehicle provides predictions along the prediction horizon. This signal is exploited inside the nonlinear MPC using a time-varying reference signal. In order to prevent the over fitting problem regarding the usage of RNN for prediction of the driving motion signals, the dropout layer is defined equal to 0.5 [44, 45]. Then, if the problem occurs, the

size of the network will be decreased as the second solution for preventing this phenomenon. In addition, the data regularization block is added to the algorithm to reduce the complexity of the model and decrease the chance overfitting [46, 47].

Nonlinear Model Predictive Control: The nonlinear plant model of the motion simulator is considered a nonlinear plant model of the MPC. Besides, the nonlinear MPC that operates the linear position and velocity of the end-effector along the prediction horizon can be calculated from the QP solution of the nonlinear MPC during each sampling time. The linear acceleration signal of the prepositioning is determined based on the linear position and velocity calculated via nonlinear MPC and linear position and velocity of the home position at the current sampling time. The nonlinear MPC and decision-making blocks are explained in this subsection.

A nonlinear MPC model in the discretised mode using direct multiple shooting [48] and sampling time $T_s = 0.01\ (s)$ along the prediction horizon (t_0 to t_f) can be formulated as:

$$\min_{x_k,u_k} \sum_{k=0}^{N-1} \left\{ \frac{1}{2} \|h_k(x_k,u_k)\|_W^2 + \frac{1}{2} \|h_N(x_N)\|_{W_N}^2 \right\} \quad (6.a)$$

$$s.t. \quad 0 = x_0 - \hat{x}_0 \quad (6.b)$$

$$0 = x_{k+1} - \emptyset_k(x_k,u_k), k = 0,1,\dots,N-1 \quad (6.c)$$

$$\underline{r}_k \le r_k(x_k,u_k) \le \overline{r}_k, k = 0,1,\dots,N-1 \quad (6.d)$$

$$\underline{r}_N \le r_k(x_N) \le \overline{r}_N \quad (6.e)$$

where $N = (t_f - t_0)/T_s$ is the shooting intervals; $\hat{x}_0$ is the state at the initial time of 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; $\emptyset$ is the nonlinear dynamical model of the plant; x_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 $\overline{r}$ and lower limits $\underline{r}$; respectively. Eq. (5.e) is used to define 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, known as the receding horizon technique.

Using sequential quadratic programming (SQP) concerning the discretised state and input, quadratic programming (QP) problem form of Eq. (5) is:

$$\min_{\Delta x,\Delta u} \sum_{k=0}^{N-1} \left\{ \left(\frac{1}{2} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix}^{\mathrm{T}} \mathrm{H}_k^i \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} + \mathrm{g}_k^{i\,\mathrm{T}} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} \right) + \frac{1}{2} \Delta x_N{}^{\mathrm{T}} \mathrm{H}_N^i \Delta x_N + \mathrm{g}_N^{i\,\mathrm{T}} \Delta x_N \right\} \quad (7.a)$$

$$s.t. \quad \Delta x_0 = \hat{x}_0 - x_0 \quad (7.b)$$

$$\Delta x_{k+1} = A_k^i \Delta x_k + B_k^i \Delta u_k + d_k^i \quad (7.c)$$

$$\underline{c}_k^i \le C_k^i \Delta x_k + D_k^i \Delta u_k \le \overline{c}_k^i \quad (7.d)$$

$$\underline{c}_N^i \le C_k^i \Delta x_k \le \overline{c}_N^i \quad (7.e)$$

where $\mathbf{x} = [x_0^{\mathrm{T}} \ \cdots \ x_N^{\mathrm{T}}]$, $\mathbf{u} = [u_0^{\mathrm{T}} \ \cdots \ u_N^{\mathrm{T}}]$, $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}^i$ and $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}^i$, H_k^i and g_k^i are the Hessian matrices; A_k^i and B_k^i are matrices from linearisation of sensitivity concerning the initial states and control over the prediction horizon, respectively; C_k^i is the Jacobian constraint matrix; while d_k^i refers to the existing gap between the previous shooting point and the next starting point; H_k^i and g_k^i are the kth block and sub-vector of the Hessian matrix and the gradient of the objective function.

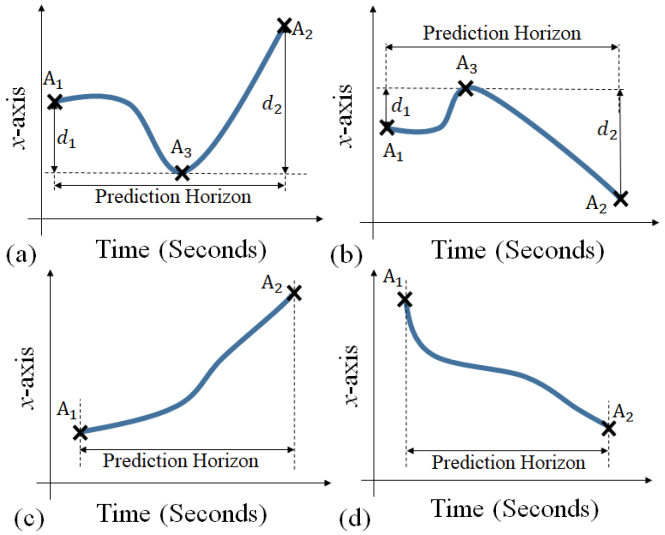


Fig. 5. The whole possible linear motion of the end-effector along prediction horizon: (a) prepositioning toward the opposite side of *x*-axis; (b) prepositioning toward the positive side of *x*-axis; (c) washout towards the initial home position; (d) washout towards the initial home position.

Decision Block: The decision block is employed to calculate the linear acceleration signal to preposition the end-effector for increasing the motion fidelity of the motion simulator user. Fig. 5.a-d shows the all-possible motions of the end-effector along the prediction horizon, which is estimated at every sampling time using the developed nonlinear MPC.

There is a change in the behaviour of the end-effector motion in Fig. 5.a-b. Based on Fig. 5.a, the motion simulator will initially move along the opposite side of the *x*-axis (A_1 to A_3), then it will move along the positive side of the *x*-axis (A_3 to A_2). Also, the motion simulator will initially move along the positive side of the *x*-axis (A_1 to A_3), then it will move along the opposite side of the *x*-axis (A_3 to A_2) in Fig. 5.b. It gives the privilege of using the prepositioning technique. The motion behaviour of the end-effector in Fig. 5.c-d is purely ascending and descending, respectively. The vertical line is the end-effector's position along the *x*-axis, and the horizontal line is the time. The motion scenario, which is shown in Fig. 5.a, presents that the end-effector moves towards the opposite side of the *x*-axis from A_1 position, but after a while, it changes its behaviour by moving towards the positive side of the *x*-axis from A_2 position to A_3 position. This motion behaviour of the end-effector can be used for prepositioning by moving the end-effector towards the opposite side of the *x*-axis before reaching A_2 position concerning the perception threshold of the sensed specific force. The same behaviour occurs for the end-effector as in Fig. 5.b by reversing the directional movement of the end-effector. In order to extract the linear acceleration of the prepositioning, the position of A_3 should be obtained using the platform's velocity along the prediction horizon using $\mathrm{find}(V_{x,Pre}==0)$. If d_1 is the linear motion of the platform from A_1 to A_3, and d_2 is the linear motion of the platform from A_2 to A_3, d_1 and d_2 can be found as follows:

Algorithm 1: Decision Block
Input: Predicted linear motion of the simulator along prediction horizon using nonlinear MPC A(x). Output: Best linear acceleration signal $a_{x,PP}$,

Begin
1. Calculation of A_1 and A_2 using the beginning and last position of the end-effector along prediction horizon.
// Solving the MATLAB function find(Vx,Pre==0) for calculation of A_3.
A_1 = Begining Position, A_2 = Last Position, A_3 = Optimal Position.
2. Calculation of $h = \left\|\frac{d_2}{d_1}\right\|$, while $d_1 = A_3 - A_1$, and $d_2 = A_3 - A_2$.
3. Deciding the possible motion scenario based on A_1, A_2, and A_3 (Fig. 5)
3.1. If $A_1 > A_3$ and $A_2 > A_3$, then $a_{x,PP} = -0.17h$
3.2. If $A_1 < A_3$ and $A_2 < A_3$, then $a_{x,PP} = +0.17h$
3.3. If 3.1 and 3.2 are not applied, then $a_{x,PP} = -\frac{V_{x,Cur}^2}{2x_{Cur}}$
Saturation of $a_{x,PP}$ between -0.17 m/s^2 and +0.17 m/s^2 to the implement the recalculated $a_{x,PP}$ to the classical washout filter.
End for t = t + 1
End.

Fig. 6. The pseudocode of the proposed decision block algorithm in order to evaluate the motion scenario and calculate the best correction linear acceleration signal for the motion simulator platform.

$$d_1 = A_3.\hat{x} - A_1.\hat{x} \quad (8.a)$$
$$d_2 = A_3.\hat{x} - A_2.\hat{x} \quad (8.b)$$

where $\hat{x}$ is the unit vector along the *x*-axis. The influence factor of the prepositioning h can be found as:

$$h = \left|\frac{d_2}{d_1}\right| \quad (9)$$

Definition 1: If $A_1.\hat{x}$ and $A_2.\hat{x}$ are greater than $A_3.\hat{x}$, then the end-effector follows the motion presented in Fig. 5.a. The linear acceleration signal of the prepositioning can be defined as:

$$a_{x,PP} = -0.17h \quad (10)$$

where ± 0.17 m/s^2 is the threshold unit of the otolith organs for sensing the linear acceleration signal via the otolith organ following the work in [35].

Definition 2:

If the $A_1.\hat{x}$ and $A_2.\hat{x}$ are smaller than $A_3.\hat{x}$, then the end-effector follows the shown motion in Fig. 5.b. Then, the linear acceleration signal of the prepositioning can be exhibited as:

$$a_{x,PP} = +0.17h \quad (11)$$

Definition 3: If none of both scenarios happens, then the end-effector should be washout towards the zero-home position under the motion perception threshold based on the study by Fang and Kemeny [49], in order to increase stability as well as efficiency, as follows:

$$a_{x,PP} = -\frac{V_{x,Cur}^2}{2x_{Cur}} \quad (12)$$

where $V_{x,Cur}$ and x_{Cur} are the current velocity and position of the platform using the prepositioning technique.

It should be noted that if the extracted $a_{x,PP}$ value is bigger than +0.17 m/s^2 or smaller than -0.17 m/s^2, the linear acceleration of the prepositioning should be saturated to +0.17 or -0.17 m/s^2, in order to respect the perception threshold of the sensed specific force. In the next step, the extracted $a_{x,PP}$ value should be filtered using a high-pass filter in order to washout the end-effector towards the home position after prepositioning. Fig. 6 presents the proposed algorithm's pseudocode to find the optimal off-centre position of the end-effector.

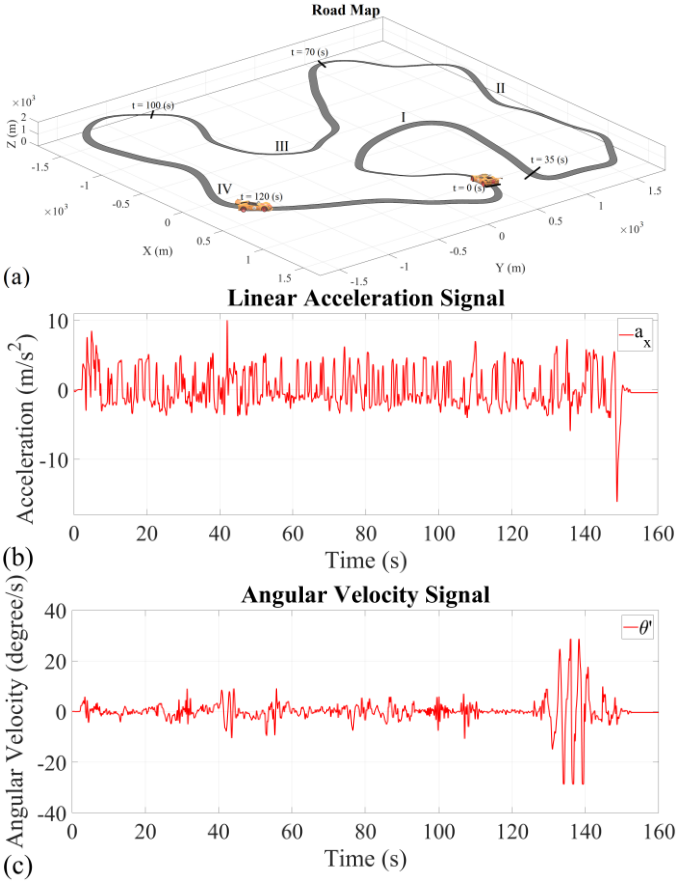


Fig. 7. The virtual vehicle driving scenario using Rigs of Rod: (a) road map; (b) linear acceleration signal along *x*-axis; (c) angular velocity signal along *pitch*-axis.

III. Result and Discussion

MATLAB software is used to verify the newly developed prepositioning technique to model and simulate the model presented in Section II. Using prepositioning, the extracted linear acceleration is added as a motion compensation signal to the classical washout filter [5]. Then, the proposed model is initially compared with the classical washout filter without using the prepositioning method. In addition, the newly developed prepositioning technique is compared with a previous prepositioning technique developed by Qazani et al. [19] as it is the most recent technique which can increase motion fidelity more than previous prepositioning techniques [16-18].

The driving scenario is recorded using the Rigs of Rod vehicle simulation environment for 150 seconds (version 0.39.5). The road map, linear acceleration signal along the *x*-axis and angular velocity along the *pitch*-angle are shown in Fig. 7.a-c, respectively. The driving scenario includes every possible motion such as sudden moving, massive accelerations and decelerations, and sharp turns.

The vehicle has been driven for ten loops in the road map as

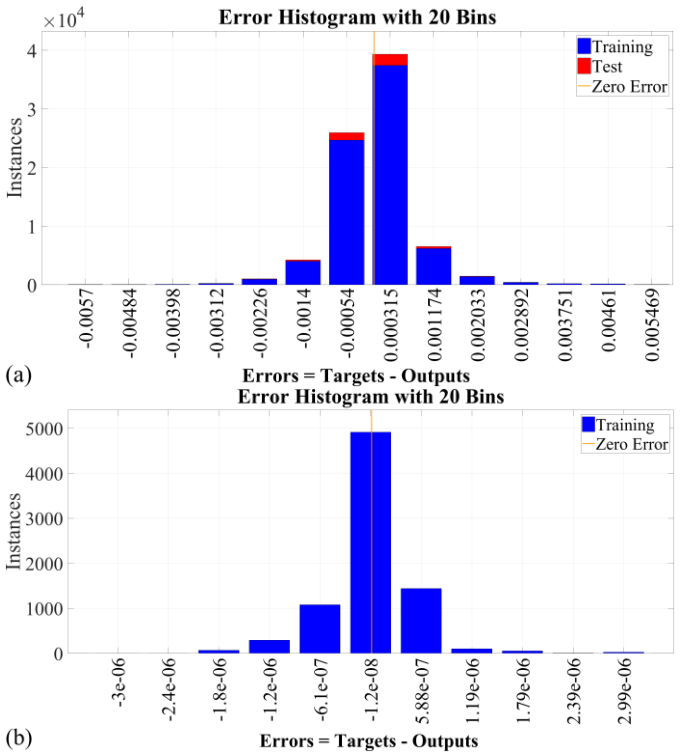


Fig. 8. Error histograms for estimation of the driving motion signals along *x*-axis using (a): feedforward NN; (b): RNN.

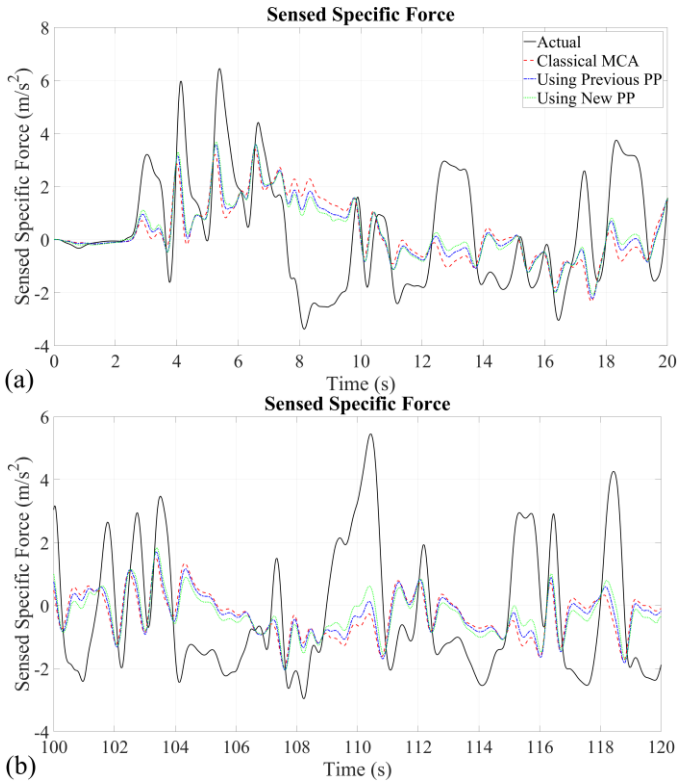


Fig. 9. The sensed specific force for the virtual vehicle and the motion simulator users using the classical washout filter without using prepositioning technique, classical washout filter using the previous prepositioning technique and the classical washout filter using the newly developed prepositioning technique for duration of: (a) 0-20 seconds of driving scenario; (b) 100-120 seconds of driving scenario.

TABLE I
THE RESULTS OF SENSED SPECIFIC FORCE ERROR BETWEEN THE VIRTUAL VEHICLE DRIVER AND THE SIMULATOR USER USING THREE INVESTIGATED METHODS

Index	RMSE			CC		
	CL-WO PP	CL-W PRE PP	CL-W NEW PP	CL-WO PP	CL-W PRE PP	CL-W NEW PP
Part I	2.1598	1.9039	1.7229	0.2300	0.7263	0.8441
Part II	1.9455	1.7067	1.5439	0.0270	0.7270	0.8618
Part III	2.1128	1.8511	1.6776	0.0388	0.7555	0.8771
Part IV	2.6440	2.3335	2.1121	0.0925	0.7922	0.9005
All Part	2.1721	1.9125	1.7312	0.1135	0.7564	0.8755

CC: Correlation Coefficient; CL-WO PP: Classical Washout Filter Without Prepositioning Technique; CL-W PRE PP: Classical Washout Filter Using Previous Prepositioning Technique; CL-W NEW PP: Classical Washout Filter Using Newly Prepositioning Technique; RMSE: Root Mean Square Error.

shown in Fig. 7.a to produce the driving motion signals (linear acceleration and angular velocity) for training the RNN, which is mentioned in the previous Section. The linear acceleration signal is used for training the RNN. As the RNN is a time-series forecasting network, there is no need for sequencing the data. The hyperparameters of the feedforward NN is employed based on the recommended settings in [19] with a single hidden layer with 36 neurons. Also, the same number of layers and neurons are chosen for the RNN to reach a fair comparison with feedforward NN. It should be noted that the data is divided into 80% and 20% for training and testing, respectively. The mean square errors (MSEs) using RNN and feedforward NN during the testing process of the network are 6.14×10^{-10} and 3.29×10^{-5}, respectively. These empirical results indicate the efficiency of RNN in comparison with feedforward NN. The error histograms of feedforward NN and RNN are shown in Fig. 8.a-b, respectively. Based on the represented data in Fig. 8.a-b, the mean error of RNN is -1.2×10^{-8}, which is lower than the mean error of 3.15×10^{-4} for the feedforward NN. It should be noted that RNN is trained based on the regenerated vehicle motion signals in the road map which is shown in Fig. 7.a. Therefore, the change of road map needs retraining of RNN in order not to affect the efficiency of the model.

Fig. 9.a-b shows the sensed specific force for the virtual vehicle and the motion simulator users using the three investigated MCA models, namely the classical washout filter without using the prepositioning technique, classical washout filter using the previous prepositioning technique [19], and classical washout filter using the newly developed prepositioning technique in two different periods. The first period is chosen between 0 and 20 seconds to show the start of the motion. Also, the second period (100-120 seconds) is chosen, as the most abrupt motion signals are generated in this period. The RMSE sensation values between the virtual vehicle and the motion simulator users indicate the first validation between different MCA models to reveal the associated efficiency. The RMSE scores of the sensed specific force between motion simulator and virtual vehicle users using classical washout filter without prepositioning technique,

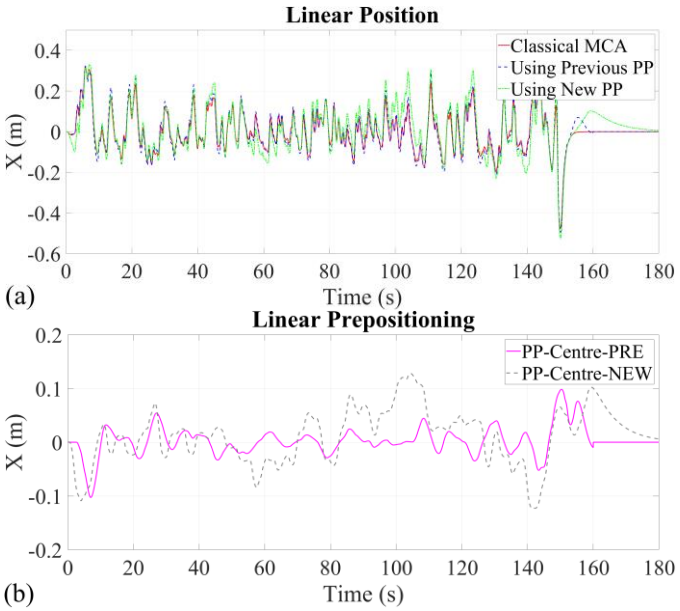


Fig. 10. (a) Linear position of the end-effector using the classical washout filter without using prepositioning technique, classical washout filter using the previous prepositioning technique and the classical washout filter using the newly developed prepositioning technique; (b) linear prepositioning of the end-effector via classical washout filter using the previous and new prepositioning techniques.

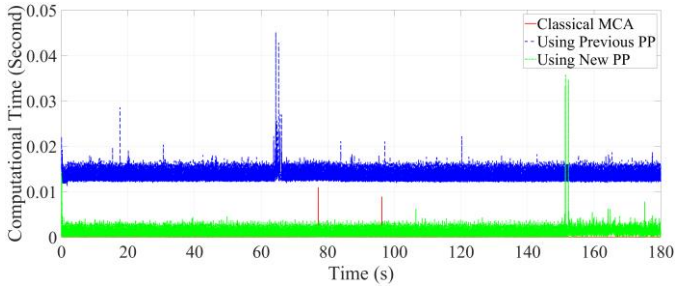


Fig. 11. The computational time of three investigated methods including effector using the classical washout filter without using prepositioning technique, classical washout filter using the previous prepositioning technique and the classical washout filter using the newly developed prepositioning technique.

classical washout filter using previous prepositioning technique and classical washout filter using newly developed prepositioning method during the whole 160 seconds of the recorded driving scenario are 2.1721, 1.9125 and 1.7312 m/s^2, respectively. The reduction of the RMSE sensed specific force between motion simulator and virtual vehicle users using the newly developed prepositioning method compared with the previous prepositioning method is due to the better reproduction of the high-pass frequency linear acceleration motion signal. The better reproduction of the high-pass frequency linear acceleration signal is important as it plays the main role in motion sickness and artefact. The second validation parameter is the correlation coefficient (CC) to reveal the shape similarity between two signals, which is used in MCA for the first time by Asadi et al. [50]. The CC scores between the virtual car and the motion simulator users using three techniques are 0.1135, 0.7564 and 0.8755, respectively. These performance scores indicate a better reproduction of the motion signals using the newly developed prepositioning method compared to those of the classical washout filter without using the existing prepositioning technique. Also, the classical washout filter using the newly developed prepositioning technique have a minor improvement over the classical washout filter using the existing prepositioning technique. It is critical not to sacrifice the sensed angular velocity to improve the sensed specific force. As the newly developed prepositioning technique tries to use the linear motion of the end-effector to improve the motion fidelity for the SNMP user, it does not affect the sensed angular velocity. Then, this improvement of the sensed specific force is important because it employs the high-pass frequency linear acceleration signal to generate better motion cues where the tilt coordination mode cannot be used. The RMSE and CC of the three investigated methods are reported in Table I for the whole and four separated zones of the motion scenario between the virtual vehicle driver and the simulator's user.

Fig. 10.a-b illustrates the end-effector's linear position and prepositioning using the three methods, including the two prepositioning techniques. As shown in Fig. 10.a, the newly developed prepositioning technique can use the workspace more productively than those of the classical washout filter with and without employing the previous prepositioning method. Based on Fig. 10.b, a better prepositioning of the end-effector using the proposed technique is shown due to the nonlinear MPC, which can estimate the linear displacement of the motion simulator along the prediction horizon. It should be mentioned that the low-pass filter of the three methods is the same, then the same angular displacement is reached using three methods.

Also, Fig. 11 presents the computational time of the processes using three investigated methods, including classical MCA without prepositioning, classical MCA using previous prepositioning, and classical MCA using proposed prepositioning methods. The results prove the lower computational load of the newly proposed method as compared with the other two methods.

The outcomes of this research prove the efficiency of the newly proposed algorithm compared with those of the previous methods, i.e., it can generate better high-frequency motion signals. It should be noted that generating inaccurate high-frequency motion signals is the main reason causing motion sickness of the motion simulator user.

IV. Conclusion

The prepositioning technique in MCA is introduced to allow highly efficient workspace usage with consideration of the same workspace limits. The previous prepositioning technique is designed using the NN to estimate the motion signal and fuzzy logic units to calculate the best off-centre position of the end-effector. No matter how it improves the workspace usage of the motion simulator, the fuzzy logic cannot extract the optimal prepositioning of the platform. As a result, a new prepositioning technique is developed to calculate the best optimal prepositioning of the end-effector. The nonlinear MPC

is employed to reach this goal because MPC can consider the workspace boundaries of the motion simulator and the human vestibular model. The investigated techniques are simulated in MATLAB. The outcomes prove a slightly better production of the high-pass frequency linear acceleration signal than those of the classical washout filter with and without using the previous prepositioning method. As a future study, an uncertainty qualification, convolutional NN and fault diagnosis system [51-54] can be used to increase the efficiency of the MCA in motion simulator platform.

REFERENCES

[1] H. Arioui, L. Nehaoua, S. Hima, N. Séguy, and S. Espié, "Mechatronics, design, and modeling of a motorcycle riding simulator," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 805-818, 2010.

[2] 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.

[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] S. J. Herdman, "Vestibular rehabilitation," *Current opinion in neurology*, vol. 26, no. 1, pp. 96-101, 2013.

[5] B. Conrad and S. F. Schmidt, "A study of techniques for calculating motion drive signals for flight simulators," 1971.

[6] L. Reid and M. Nahon, "Flight simulation motion-base drive algorithms: part 1. developing and testing equations," *UTIAS Report, No. 296*, 1985.

[7] 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*, 2020.

[8] M. R. C. Qazani, H. Asadi, M. Rostami, S. Mohamed, C. P. Lim, and S. Nahavandi, "Adaptive Motion Cueing Algorithm Based on Fuzzy Logic Using Online Dexterity and Direction Monitoring," *IEEE Systems Journal*, 2021.

[9] M. R. C. Qazani, H. Asadi, T. Bellmann, S. Perdrammehr, S. Mohamed, and S. Nahavandi, "A New Fuzzy Logic Based Adaptive Motion Cueing Algorithm Using Parallel Simulation-Based Motion Platform," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2020: IEEE, pp. 1-8.

[10] 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.

[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] H. Asadi, C. P. Lim, A. Mohammadi, S. Mohamed, S. Nahavandi, and L. Shanmugam, "A genetic algorithm–based nonlinear scaling method for optimal motion cueing algorithm in driving simulator," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 232, no. 8, pp. 1025-1038, 2018.

[13] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A decoupled linear model predictive control-based motion cueing algorithm for simulation-based motion platform with limitted workspace," in *2019 IEEE International Conference on Industrial Technology (ICIT)*, 2019: IEEE, pp. 35-41.

[14] M. R. C. Qazani, H. Asadi, S. Mohamed, and S. Nahavandi, "An Inverse Kinematic-based Model Predictive Motion Cueing Algorithm for a 6-DoF Gantry-Tau Mechanism," in *ACRA 2019: Proceedings of the Australasian Conference on Robotics and Automation*, 2019: [Australian Robotics & Automation Association], pp. 1-9.

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

[16] C. Weiß, "Control of a dynamic driving simulator: Time-variant motion cueing algorithms and prepositioning," Universität Stuttgart, DLR, 2006.

[17] T. Chapron and J.-P. Colinot, "The new psa peugeot-citroen advanced driving simulator overall design and motion cue algorithm," in *Proceedings of Driving Simulation Conference*, 2007, vol. 42, p. 173.

[18] P. Hansson, A. Stenbeck, A. Kusachov, F. Bruzelius, and B. Augusto, "Prepositioning of driving simulator motion systems," *International Journal of Vehicle Systems Modelling and Testing (IJVSMT)*, vol. 10, no. 3, pp. 288-304, 2015.

[19] 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.

[20] 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.

[21] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A Motion Cueing Algorithm Based on Model Predictive Control Using Terminal Conditions in Urban Driving Scenario," *IEEE Systems Journal*, 2020.

[22] M. Katliar, F. M. Drop, H. Teufell, M. Diehl, and H. H. Bülthoff, "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.

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

[24] 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.

[25] Y. R. Khusro, Y. Zheng, M. Grottoli, and B. Shyrokau, "MPC-Based Motion-Cueing Algorithm for a 6-DOF Driving Simulator with Actuator Constraints," *Vehicles*, vol. 2, no. 4, pp. 625-647, 2020.

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

[27] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[28] 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.

[29] 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.

[30] 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.

[31] X. Wang, M. Chen, M. Zhu, and P. Tremont, "Development of a kinematic-based forward collision warning algorithm using an advanced driving simulator," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2583-2591, 2016.

[32] P. Nilsson, L. Laine, and B. Jacobson, "A simulator study comparing characteristics of manual and automated driving during lane changes of long combination vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2514-2524, 2017.

[33] 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.

[34] R. J. Telban and F. M. Cardullo, "Motion cueing algorithm development: Human-centered linear and nonlinear approaches," *NASA TechReport CR-2005-213747*, 2005.

[35] 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.

[36] 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.

[37] A. P. Fard, H. Abdollahi, and M. Mahoor, "ASMNet: A Lightweight Deep Neural Network for Face Alignment and Pose Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1521-1530.

[38] M. H. Alkawaz, M. T. Veeran, and R. Bachok, "Digital image forgery detection based on expectation maximization algorithm," in *2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, 2020: IEEE, pp. 102-105.

[39] T.-C. Chen *et al.*, "Application of machine learning in rapid analysis of solder joint geometry and type on thermomechanical useful lifetime of electronic components," *Mechanics of Advanced Materials and Structures*, pp. 1-9, 2021.

[40] M. Z. Asghar *et al.*, "Performance evaluation of supervised machine learning techniques for efficient detection of emotions from online content," *arXiv preprint arXiv:1908.01587*, 2019.

[41] A. Mavridaki, E. Galiotou, and E. C. Papakitsos, "Developing a Software Application for the Study and Learning of Linear a Script," *Review of Computer Engineering Research*, vol. 8, no. 1, pp. 8-13, 2021.

[42] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design*. PWS Publishing Co., 1997.

[43] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training recurrent neural networks with the Levenberg–Marquardt algorithm for optimal control of a grid-connected converter," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 9, pp. 1900-1912, 2014.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[45] C. Ha, V.-D. Tran, L. N. Van, and K. Than, "Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout," *International Journal of Approximate Reasoning*, vol. 112, pp. 85-104, 2019.

[46] X. Ying, "An overview of overfitting and its solutions," in *Journal of Physics: Conference Series*, 2019, vol. 1168, no. 2: IOP Publishing, p. 022022.

[47] G. Mustafa, A. Ghaffar, and M. Aslam, "A subdivision-regularization framework for preventing over fitting of data by a model," *Applications and Applied Mathematics: An International Journal (AAM)*, vol. 8, no. 1, p. 11, 2013.

[48] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603-1608, 1984.

[49] Z. Fang and A. Kemeny, "An efficient Model Predictive Control-based motion cueing algorithm for the driving simulator," *Simulation*, vol. 92, no. 11, pp. 1025-1033, 2016.

[50] 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.

[51] D. Milanés-Hermosilla *et al.*, "Monte Carlo Dropout for Uncertainty Estimation and Motor Imagery Classification," *Sensors*, vol. 21, no. 21, p. 7241, 2021.

[52] D. M. Hermosilla *et al.*, "Shallow convolutional network excel for classifying motor imagery EEG in BCI applications," *IEEE Access*, vol. 9, pp. 98275-98286, 2021.

[53] J. C. Fernández, L. B. Corrales, I. F. Benítez, and J. R. Núñez, "Fault Diagnosis of Combustion Engines in MTU 16VS4000-G81 Generator Sets Using Fuzzy Logic: An Approach to Normalize Specific Fuel Consumption," Cham, 2022: Springer International Publishing, in Intelligent Computing Systems, pp. 17-29.

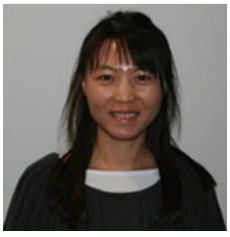
[54] M. Abdar *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243-297, 2021.

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 is currently an Alfred Deakin Postdoctoral Research Fellow in the Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University. His current research interests include model predictive control, motion cueing algorithm and soft computing controllers.

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 artificial intelligence, motion simulator technologies, motion cueing algorithms, control, and human factors in virtual environments.



LI ZHANG (Senior Member, IEEE) received the Ph.D. degree from the University of Birmingham. She gained postdoctoral experience from the University of Birmingham. She is currently a Reader with the Department of Computer Science, Royal Holloway, University of London, U.K. Her research interests include deep learning, machine learning, intelligent robotics, and evolutionary computation. She is an Associate Editor of Decision Support Systems.

Farzin Tabarsinezhad received the B.Sc. degree in computer software engineering (First Class Hons.) from Shahid Shamsi Pour Institute of Technology and the M.Sc. degree in information technology engineering from the University of Tehran, Iran, in 2020. He is currently a research assistant at University of Tehran.

His current research interests include Artificial Intelligence, Modeling and Simulation.

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