

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

Chalak Qazani, Mohamad Reza, Asadi, Houshyar, Zakarya, Muhammad, Lim, Chee Peng, Liew, Alan Wee-Chung, Karkoub, Mansour, and Nahavandi, Saeid (2023) *A Neural Network-Based Motion Cueing Algorithm Using the Classical Washout Filter for Comprehensive Driving Scenarios*. IEEE Transactions on Intelligent Transportation Systems, 25 (6) pp. 5112-5121.

Access to this file is available from:

<https://researchonline.jcu.edu.au/86720/>

© 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.2023.3340290>

A Neural Network-based Motion Cueing Algorithm Using the Classical Washout Filter for Comprehensive Driving Scenarios

Mohammad Reza Chalak Qazani, Houshyar Asadi, *Member, IEEE*, Muhammad Zakarya, Chee Peng Lim, Alan Wee-Chung Liew, Mansour Karkoub, Saeid Nahavandi, *Fellow, IEEE*

Abstract—The motion cueing algorithm (MCA) enables lifelike motion in simulators resembling real driving. Regenerated motions must adhere to workspace constraints. Vehicle motion signals (linear acceleration, angular velocity) are generated in a simulated vehicle environment utilised in MCA for motion cues. These signals are categorised into levels (slow, medium, fast) based on frequency and amplitude. The commonly used MCA, the classical washout filter, is typically fine-tuned using worst-case (fast-driving) scenarios to meet the simulator's requirements across various situations. However, this approach reduces the MCA's effectiveness in handling slower driving scenarios, resulting in conservatism in platform workspace usage for slow and medium driving. Consequently, a noticeable motion sensation error arises between real vehicle drivers and motion simulator users. To rectify this issue, a novel neural network-based MCA is developed in this study. Three distinct classical washout filters are meticulously tuned to cater to slow, medium, and fast driving scenarios. These filters generate precise motion cues for simulator users at corresponding levels of driving scenarios. The neural network-based MCA is constructed using the synthesised signals from these classical washout filters. This proposed method is thoroughly validated through the utilisation of MATLAB software. In direct comparison with the standard classical washout filter, the proposed MCA significantly reduces the motion sensation error, enriches motion fidelity, and optimises the utilisation of the simulator's workspace.

Index Terms— motion cueing algorithm, classical washout filter, neural network, multilayer perceptron, optimisation methods, parallel processing.

I. INTRODUCTION

MOTION simulators have a broad spectrum of applications, extending beyond automobiles to aviation, research facilities, and driver/pilot training across various vehicles [1-3]. The precision of motion signals is integral to replicating an authentic, real-world experience. However, a persistent

challenge in this domain is the prevalence of motion sickness, which significantly impacts user engagement [4]. This discomfort stems from the incongruence between the vestibular and visual motion perceptions. The vestibular system within the inner ear detects motion [5, 6], whereas our visual system processes visual cues [7]. Simulators employ manipulators with numerous degrees of freedom to generate intricate motion signals. To alleviate the effects of motion sickness, the motion cueing algorithm (MCA) works towards harmonising these sensory inputs, thus enriching the realism of the experience. Despite these advancements, the combination of the limited workspace inherent to simulators and an imperfect MCA design contributes to the generation of inaccurate cues, thereby heightening the risk of motion sickness. Efforts to meticulously refine the modelling and precise implementation of the MCA hold promise in tackling these challenges. This approach has the potential to significantly enhance the overall simulator experience, ensuring a more seamless and nausea-free user encounter.

Conrad and Schmidt [8] introduced the first MCA design, known as the classical washout filter, and its first phase was designed for a land vehicle simulator. Reid and Nahon [9] modified the proposed MCA model [8] to be applied to an aircraft. The classical washout filter is the most widely-used form in the industry to formulate the MCA, owing to its applicability, affordability, low computational load, ease of tuning, stability, and simplicity [10, 11]. On the other hand, adaptive and optimal washout filters are introduced to solve the disadvantages of classical washout filters, such as non-consideration of the human vestibular system, fixed-term washout filter parameters, and insufficiency of abrupt motion signals [12-15]. Recently, model predictive control (MPC) has been used in MCA design to extract the optimal motion signals of the simulator at each time-step, considering the workspace limitations [16]. Qazani *et al.* [17] used the concept of terminal

(Corresponding author: Mohammad Reza Chalak Qazani)

M. R. C. Qazani and M. Zakarya are with the Faculty of Computing and Information Technology, Sohar University, Sohar, Sohar 311, Oman (e-mail: m.r.chalakqazani@gmail.com; mzakarya@su.edu.om).

H. Asadi, and C. P. Lim are with the Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, VIC 3125, Australia (e-mail: houshyar.asadi@deakin.edu.au; chee.lim@deakin.edu.au; saeid.nahavandi@deakin.edu.au).

C. Liew is with the Institute for Integrated and Intelligent Systems, Griffith University, Queensland, Australia (e-mail: a.liew@griffith.edu.au).

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

S. Nahavandi is an Associate Deputy Vice-Chancellor Research and Chief of Defence Innovation, Swinburne University of Technology, John St, Hawthorn VIC 3122, Australia, and he is with Harvard Paulson School of Engineering and Applied Sciences, Harvard University, Allston, MA 02134 USA (e-mail: saeid.nahavandi@ieee.org).

condition to increase the stability of the MPC-based MCA facing an abrupt urban motion signal. The following concept has been followed in developing a time-varying MPC-based MCA to bypass the problem of a linear fixed-term plant model [18]. Qazani *et al.* [19] used nonlinear MPC to pre-position the end-effector of a simulator to address its workspace limitations. Asadi *et al.* [20] proposed a decoupled MPC to increase the efficiency of the MCA to allow better usage of the tilt coordination channel.

While neural network (NN) models in MCA designs are new, they have a high potential to enhance efficiency without increasing system complexity as in classical methods. Rengifo *et al.* [21] employed a recurrent NN model to optimise MPC-based MCA with less complexity than traditional methods. The developed model focused on one channel for simplicity. Extending it to all channels remains an open research gap, requiring the handling of multi-input-multi-output (MIMO) systems. Koyuncu *et al.* [22] used a nonlinear autoregressive network to create a fully NN-based solution from a linear MPC-based MCA model. However, they didn't fully explain online optimisation without algorithm training. The proposed method's training on one MPC-based MCA faces challenges due to worst-case scenario tuning.

The primary drawback in MCA applications is using the worst-case scenario to tune all MCA designs, including classical and MPC, to meet the motion simulator's workspace constraints [8-22]. This approach stems from ensuring that the end-effector remains within workspace boundaries across all scenarios (fast, medium, and slow). Nonetheless, this strategy diminishes the MCA's effectiveness in handling medium- and slow-driving motion signals. We propose developing three MPC-based MCA models corresponding to slow, medium, and fast driving scenarios to address this issue. This approach aims to optimise efficiency for each set of motion signals. The next step involves recording the optimal motion signals from these models within the MATLAB workspace environment. This recording forms the basis for creating a comprehensive MCA model capable of generating highly efficient motion signals tailored to various driving scenarios. It should be noted that the development of purely NN-based MCA is new and has a high potential to enhance efficiency without increasing system complexity, such as classical methods. In this study, a multilayer perceptron (MLP) NN model is trained offline and employed within the classical washout filter instead of using either high- or low-pass (HP/LP) filters. The key advantage of using NNs instead of HP/LP filters is eliminating the worst-case scenario method to reproduce highly efficient motion cues for different categories of motion signals.

In the next section, the concept of a washout filter is described. In Section III, the methodology of the proposed method is explained in detail, including the data regeneration and data pre-processing methods, the development of the NN-based MCA model, and the Bayesian optimisation method. The proposed NN-based MCA model using a classical washout filter is validated using MATLAB software, as presented in Section IV. The concluding remarks of this research are summarised in Section V.

II. WASHOUT FILTER

This section describes the human vestibular model and the classical washout filter. Fig. 1 shows a schematic diagram of the longitudinal channel of a classical washout filter. The following sub-sections explain the human vestibular model and classical washout filter.

A. Human Vestibular System

The human perceptual component primarily comprises auditory, visual, and kinaesthetic inputs and is responsible for hearing, seeing, and feeling. The human motion perception system infers the direction and speed of the body using visual, proprioceptive, and vestibular inputs. Extraction of the mathematical model representing the human vestibular system in the MCA model is necessary to validate the proposed model's efficiency. The vestibular system is located in the inner ear, known as the bony labyrinth. It consists of the otolith organs and semicircular canals. It mainly contributes to motion sensation [5, 6].

The otolith organs are composed of a layered otolithic membrane connected to the sensory cells from the base. It comprises upper, lower, otoconial, and gelatinous layers [5, 23]. The combination of the linear acceleration (LA) and gravity motion signals sensed by the otolith organs, known as the sensed specific force, $\mathbf{S}_F$, can be formulated as [24]:

$$\mathbf{S}_F = \mathbf{a} - \mathbf{R}(\phi, \theta, \psi)\mathbf{g} \triangleq \begin{bmatrix} a_x + g\theta \\ a_y - g\phi \\ a_z - g \end{bmatrix} \quad (1)$$

where $\mathbf{a}$ and $\mathbf{g}$ are the LA and earth gravity vectors, respectively. In addition, $\mathbf{R}(\phi, \theta, \psi)$ is a rotation matrix concerning the driver's head position within the simulator.

Different mathematical models of the human otolith organs have been comprehensively reviewed in [5]. In this study, the Telban and Cardullo otolith organs model [23] is employed to compute the sensed specific force $\hat{f}$ using the applied LA signal f as follows:

$$\frac{\hat{f}(s)}{f(s)} = K_{OTO} \left(\frac{(\tau_a s + 1)}{(\tau_L s + 1)(\tau_s s + 1)} \right) \quad (2)$$

where τ_a , τ_s , and τ_L are the neural processing term lead operator, short-time constant, and long-time constant of the otolith organs model, respectively. In addition, $K_{OTO} = 0.4$ specifies the static sensitivity regarding the afferent firing rate per unit of acceleration (ips/g) in the x -, y - and z -axis, in which $\tau_L = 5.3(s)$ in the x -axis, $\tau_L = 5.33(s)$ in the y - and z -axis, $\tau_s = 0.016(s)$ and $\tau_a = 13.2(s)$ in the x -, y - and z -axis.

The semicircular canals are filled with endolymph. They consist of two sets of three elliptical canals: the horizontal, anterior and posterior canals [6]. In this study, the mathematical semicircular canal model of Telban and Cardullo [23] is employed to compute the sensed angular velocity (AV) $\hat{\omega}$ using the applied AV ω as follows:

$$\frac{\hat{\omega}}{\omega} = \frac{G_{SCC}\tau_1\tau_a s^2(1+\tau_L s)}{(1+\tau_a s)(1+\tau_1 s)(1+\tau_2 s)} \quad (3)$$

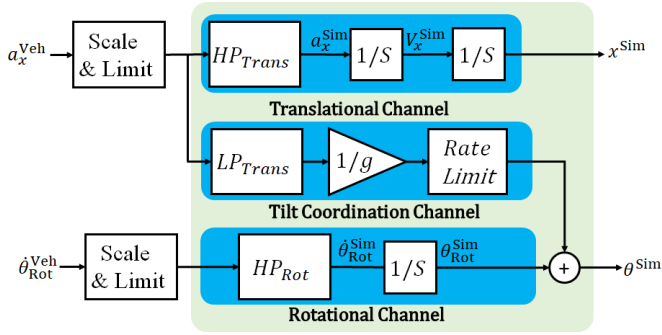


Fig. 1. The schematic structure of the classical washout filter along longitudinal channel consists of three sub-channels with addition of scale and limit units.

where τ_1 , τ_2 , τ_a and τ_L are the semicircular time constants and $\tau_1 = 5.73(s)$, $\tau_2 = 0.005(s)$, $\tau_a = 80(s)$, and $\tau_L = 0.06(s)$ in the x -, y - and z -axis. In addition, G_{SCC} is the AV threshold that scales the response of the semicircular model with 28.6479 in the x - and y -axis and 35.8099 in the z -axis.

B. Classical Washout Filter

The classical washout filter is a mixture of HP and LP filters to generate the appropriate motion signals for the motion simulator to satisfy the workspace limitations and to regenerate the motion sensations of a real vehicle for the simulator user. In this respect, a driving motion can be analysed with four different channels, i.e., longitudinal (x -axis and pitch-angle), lateral (y -axis and roll-angle), heave (z -axis), and yaw (yaw-angle) [25]. The longitudinal channel is composed of LA along the x -axis and AV along the pitch-angle of a real vehicle and the simulator motion signals as the inputs and outputs, respectively. The lateral channel comprises LA along the y -axis, AV along the roll-angle of a real vehicle, and the simulator motion signals as the inputs and outputs, respectively. The longitudinal and lateral channels comprise three sub-channels: translational, tilt coordination and rotational. The details are described as follows.

The translational sub-channel combines a second-order HP filter and two integrators. This second-order HP filter is used to erase the low-frequency LA signals. As such, the linear displacement of a real vehicle can be reproduced in a bounded linear workspace of the motion simulator. The cut-off frequency of the second-order HP filter has to be tuned based on the linear workspace of the simulator. The damping ratio also affects the response time of the HP filter. The tilt coordination sub-channel combines a second-order LP filter, gain, and rate limiter. This sub-channel is used to tilt the driver's seat below the human threshold unit of the sensed AV to use the LA of gravity as low-frequency LA signals. This is also known as somatogravic illusion [26, 27]. The second-order LP filter is responsible for sensing low-frequency LA signals. Then, the low-frequency LA signals are divided into gravity motion (9.81 m/s^2) using a gain unit. The rate limiter is employed to keep AV under the human threshold unit to not mix the sensation of linear and angular motions. The cut-off frequency is tuned with respect to the angular workspace of the simulator. The rotational sub-channel consists of a first-order HP filter and an integrator unit. The first-order HP filter is

responsible for filtering low-frequency AV signals of a real vehicle to compute the AV of the motion simulator. Meanwhile, the integrator unit is used to calculate the angular position of the simulator concerning its AV. The input motion signals should be scaled and limited before importing them inside the sub-channels based on the workspace limitation of the motion simulator. The simulator's angular position is extracted via rotational and tilt coordination sub-channels.

The next section explains the current research methodology to develop a comprehensive NN-based MCA model using the captured data by the classical washout filter.

III. METHODOLOGY

This section describes the research methodology to investigate the proposed NN-based MCA model. Fig. 2 shows a schematic structure of the proposed two-stage framework, which consists of (a) data generation and pre-processing and (b) NN-based MCA with optimisation developed using the Simulink/MATLAB software packages, respectively. In addition, the Bayesian optimisation technique is employed to identify the optimal parameters of the MLP network. The developed solution can produce accurate motion cues for the simulator user with a low effect of motion sickness. The following sub-sections explain data generation and pre-processing, MLP structure, and the Bayesian optimisation method.

A. Data Generation and Pre-Processing

Fig. 2a shows the first stage of the proposed framework, i.e., data generation and pre-processing. It consists of three main steps: regeneration of three motion scenarios (slow-, medium-, and fast-driving), tuning the classical washout filter, execution of the classical washout filter and recording the data for NN training. The details are as follows:

Step 1: The Rigs of Rod (RoR) software regenerates the vehicle motion signals. RoR is an open-source vehicle simulation environment for gaming and recording facilities [28]. The game consists of interconnected nodes to form the wheels and chassis in real-time to simulate deformable objects using a soft-body physics engine. It is available in a virtual 9km^2 sandbox environment.

Step 2A: The driver of a virtual vehicle performs slow-, medium-, and fast-driving motions. The same road map is used for the regeneration of these three groups of motion scenarios. It should be noted that the test scenario is generated based on the driver's behaviour preference for evaluation purposes of the developed NN-based MCA. After regenerating the motion signals, the classical washout filter is designed and modelled in Simulink/MATLAB. The associated HP and LP filters are tuned based on three different generated motion signals from RoR. Table I shows the filter parameters along the longitudinal channel, which satisfy the slow-, medium-, and fast-driving strategies. These fixed-term parameters are employed inside each corresponding Simulink module of the classical washout filter to yield the best solution for each driving strategy. The longitudinal workspace boundary of the motion simulator is $\pm 0.12 \text{ (m)}$ and $\pm 0.33 \text{ (rad)}$, pertaining to the linear and angular motions of the simulator, respectively.

Step 2B: Three tuned classical washout filters from Step 2A extract the motion signals from the simulator based on their respective motion scenarios. For example, the slow-driving motion signals are employed inside the classical washout filter, which is tuned under the slow-driving motion scenario.

Step 3: After running three classical washout filters, the signals generated in MATLAB are recorded, which are used for training and testing purposes of the proposed NN-based MCA model, where the filter input comprises one-dimensional data. In this study, the motion signals from the RoR driver, including LA and AV, form the filter inputs. Increasing the number of features in NN modelling can increase network accuracy. As such, the integrator of filter input is chosen as the second feature for NN training and testing. In addition, the difference between these two parameters is leveraged as the third feature of the NN model, as it represents a better regeneration of high-fidelity motion signals under different motion scenarios. The input signals of the NN-based HP filter consist of LA, linear velocity, and the difference between LA and linear velocity. The same input signals are used for the NN-based LP filter in the tilt coordination channel. Meanwhile, the AV, angular position and the difference between AV and angular position form the input signals of the NN-based rotational HP filter. The delay function is used to increase the NN input size based on the specified delays, as follows:

$$\text{Input} = \begin{bmatrix} (a_x)_{1:100:1}, (a_x)_{102:200:2}, (a_x)_{204:300:4}, \\ (a_x)_{414:500:16}, (a_x)_{526:600:32} \end{bmatrix} \quad (4)$$

where the first and second arguments are the start and end points, while the third argument is the time step. They are depicted in Fig. 2b as a time-series data block, which converts the one-dimensional signal into a 30-dimensional counterpart. In addition, the model output is defined based on the next linear position along the x -axis and angular position along the $pitch$ -angle of the simulator.

This study has 5 participants and 175 seconds of recorded driving for each scenario (slow-driving, medium-driving, and fast-driving). The driving mode is solo driving on a long-closed map using one dynamic car model. Using a time step of 0.01 seconds, the training data set consists of 87500 samples. For the test, the RoR vehicle is driven for 20 seconds on the same road map, producing a test data set with 2000 samples.

B. Multilayer Perceptron Neural Network

The MLP [29] is a feedforward NN that consists of an input layer, one or more hidden layers, and an output layer. The

TABLE I
THE CLASSICAL WASHOUT FILTER PARAMETERS FOR FAST, MEDIUM, AND SLOW DRIVING SCENARIOS ALONG THE LONGITUDINAL CHANNEL.

Item	Translational	Tilt Coordination	Rotational
Fast-Driving	$\omega_n = 3$ $\xi = 2$	$\omega_n = 4.25$ $\xi = 1$	$\omega_n = 0.2$
Medium-Driving	$\omega_n = 2.3$ $\xi = 2$	$\omega_n = 5.25$ $\xi = 1$	$\omega_n = 0.15$
Slow-Driving	$\omega_n = 1.8$ $\xi = 2$	$\omega_n = 6.25$ $\xi = 1$	$\omega_n = 0.1$

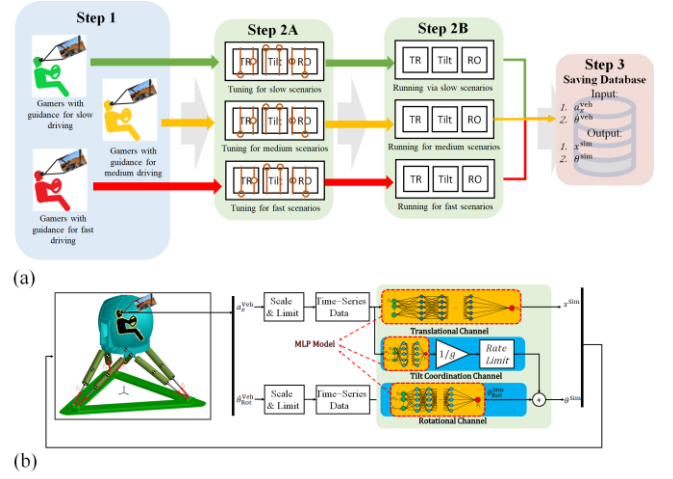


Fig. 2. A graphical illustration of the proposed two-stage framework. (a): data generation and pre-processing; (b): NN-based MCA development and optimisation.

hidden and output nodes have nonlinear activation functions, while those associated with the input nodes are linear.

The MLP is widely used for solving classification and regression problems using supervised learning methods. During supervised learning, the inputs and outputs are used by the backpropagation algorithm to adjust the network parameters that produce the minimum error between the target and predicted values, as measured with several performance indicators, e.g. mean square error (MSE) or root mean square error (RMSE). The backpropagation algorithm uses a stochastic gradient descent technique to compute and back-propagate the weights and biases through the MLP. Given an n^{th} data sample, the error from the k^{th} output node during training is computed as follows:

$$e_k(n) = t_k(n) - y_k(n) \quad (5)$$

where t and y are the target and predicted output values.

The entire network error is computed for weight adjustment, i.e.,

$$\varepsilon(n) = \frac{1}{2} \sum_k e_k^2(n) \quad (6)$$

The gradient descent method is used to yield.

$$\Delta w_{ij}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial z_i(n)} y_i(n) \quad (7)$$

where z_i and y_i are the previous neuron's induced local field and outputs, respectively; η is the learning rate, which affects the convergence of the MLP. The weight derivative with respect to the induced local field is calculated as follows:

$$-\frac{\partial \varepsilon(n)}{\partial z_i(n)} = e_k(n) \phi'(z_k(n)) \quad (8)$$

where ϕ' , i.e., derivative of the activation function, is a constant concerning the induced local field and weight. Then, the weight derivative can be computed as:

$$-\frac{\partial \varepsilon(n)}{\partial z_k(n)} = \phi'(z_k(n)) \sum_j -\frac{\partial \varepsilon(n)}{\partial z_j(n)} w_{jk}(n) \quad (9)$$

From Eq. (8), the variation of the j^{th} node weight in the output layer affects the variation of hidden node weights. Similarly, variation of the hidden node weights affects the input node weights based on a derivative of the activation function. This is the gist of backpropagation in weight adjustment [30].

The HP filter output is integrated to calculate the linear and angular configurations of the simulator in the translational and rotational sub-channels, respectively. The linear and angular displacements are chosen as the target outputs of the HP filter, which need to be trained offline.

C. Bayesian Optimisation Algorithm

Bayes' theorem is useful for extracting the probability of an incident occurrence using system knowledge [31]. Bayesian optimisation can be implemented in a bounded solver to calculate the system states to minimise a cost function [32]. The cost function can be stochastic or deterministic, while the state can be an integer, real, or categorical value. This study uses the Bayesian optimisation method to tune the MLP parameters. It works through an iterative mechanism of global optimisation concerning the cost function. The optimisation starts with a likelihood and prior functions. A posterior distribution is calculated based on the previous evaluation of the objective function in each iteration. The key elements in Bayesian optimisation comprise a Gaussian model of the cost function, Bayesian updating procedure, and acquisition function. An acquisition function assesses the location of the state using the posterior distribution function. The acquisition function modifies its anticipated state location using the method proposed in [33].

Fig. 3 depicts the structure of Bayesian optimisation in this

study. It is used to search for the optimal parameters that affect the network learning rate, as well as the optimal numbers of layers, neurons and validations of the NN-based MCA, subject to the following settings:

1. The number of layers is between 1 and 20.
2. The number of neurons is between 1 and 400.
3. The number of validations is between 1 and 20.
4. The mutation rate is between 0.0001 and 0.01.
5. The sigma rate is between 1×10^{-6} and 1×10^{-4} .

An appropriate cost function is essential for producing good results. A modular strategy is used with Bayesian optimisation for the NN network. The model is trained and evaluated using the predefined values via the optimisation algorithm, and the cost function is defined as the goal. Initially, the motion sensation error is utilised to decrease the difference in experience between the RoR vehicle driver and the motion simulator user. Then, the RMSE rate between these two targets and the obtained outputs are computed, which is the first factor of the cost function. On the other hand, motion fidelity is important in regenerating accurate motion cues for the motion simulator user. Asadi *et al.* [1] devised the correlation coefficient (CC) between the motion sensation signal of the RoR vehicle driver and motion simulator user as a motion fidelity indicator. The CC score between the target and obtained outputs forms the second target of the cost function. As such, the overall cost function J_{Total} is expressed as follows:

$$J_{\text{Total}}(X) = J_{\text{RMSE}}(X) + J_{\text{CC}}(X) \quad (11)$$

where J_{RMSE} , and J_{CC} are the sub-functions based on the RMSE and CC scores between the target and obtained outputs with respect to the actual filters and MLP models. The parallel computation technique in MATLAB (i.e. parfor) accelerates the

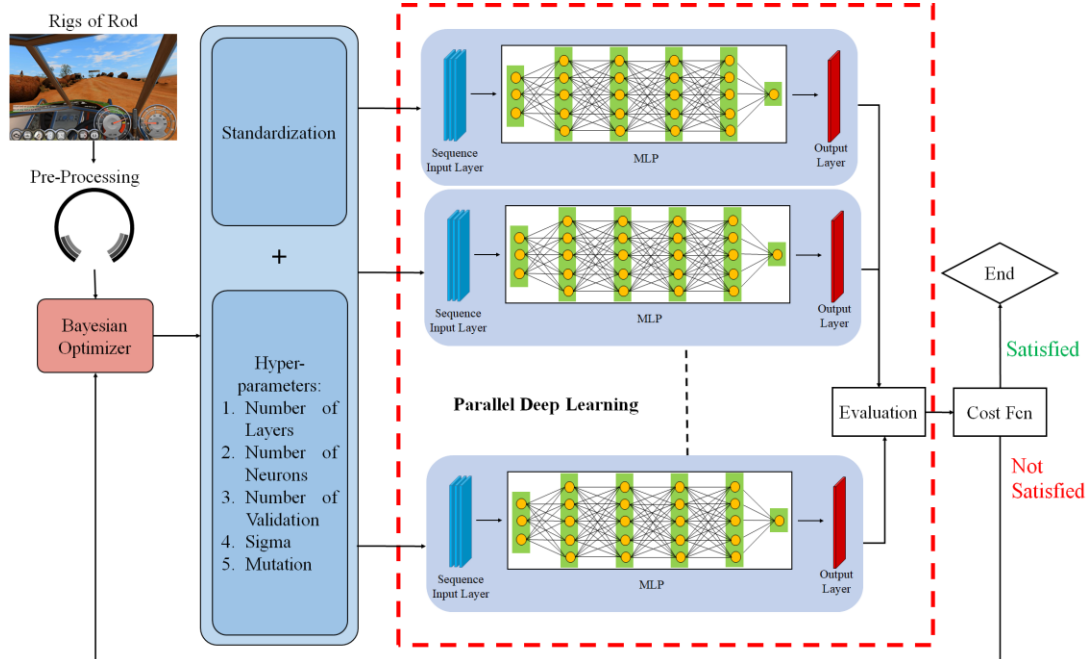


Fig. 3. The Bayesian optimization method to extract the optimal parameters of the MLP including number of layers, neurons and validation, as well as rate of mutation and sigma for classical washout filter along longitudinal channel.

tuning process using Bayesian optimisation. A total of 10 networks are simultaneously simulated instead of one in each trial. This parallel method can reduce the Bayesian optimisation's computational time and make it feasible for real-time applications.

The results of the proposed NN-based MCA with Bayesian optimisation are explained in detail in the upcoming section. Its usefulness, as compared to that of the classical method, is evaluated and presented in the next section.

IV. RESULTS AND DISCUSSION

In this section, the results of the proposed NN-based MCA model are analysed, which cover pre-processing, optimisation, training, and testing. The proposed model is employed within the classical washout filter to evaluate its efficiency against a classical washout filter. In addition, a recently developed MPC-based MCA is implemented for performance comparison, demonstrating the efficiency of the proposed model against an advanced method [16].

A. Neural Network

Fig. 4 shows the raw LA and linear velocity motion signals of the RoR vehicle along the longitudinal channel used in the pre-processing stage. The linear velocity and the difference between LA and linear velocity are employed as the second and third inputs of the MLP (along the translational sub-channels of the longitudinal channel) to enhance its accuracy. The pre-processed data set is used to evaluate the proposed NN-based MCA. The standardised LA and linear velocity of the RoR vehicle along the translational sub-channels of the longitudinal channel is shown in Fig. 4. The proposed technique is validated against the classical washout filter, which is the most common industrial form of the MCA developed by Conrad and Schmidt [8, 34].

Both classical and NN-based MCA models with Bayesian optimisation are developed using the Simulink/MATLAB software. The best MLP model with supervised training is obtained, aiming to overcome the disadvantages of classical washout filters, primarily fixed parameters and insufficient workspace usage. The developed model supports two data pre-processing techniques to ensure flexibility: normalisation and standardisation. If necessary, both techniques can be disabled so that raw data can be used for network training. After several trials, the standardisation technique yields better results than those from normalisation and raw data. As such, this study employs the standardisation technique for data pre-processing.

In the first step, the effect of different numbers of layers and neurons pertaining to NN accuracy is evaluated. The error is calculated using the RMSE between the target and obtained outputs. The error extraction using all possible configurations of the neurons and layers is extremely time-consuming. It deteriorates with an increasing number of samples. It is important to check the influence of the parameters (number of validations, number of neurons, number of layers, rate of mutation, and rate of sigma) on the errors of a configuration. This study employs the Bayesian optimisation method to search for the optimal MLP parameters. Fig. 5 shows the fitness

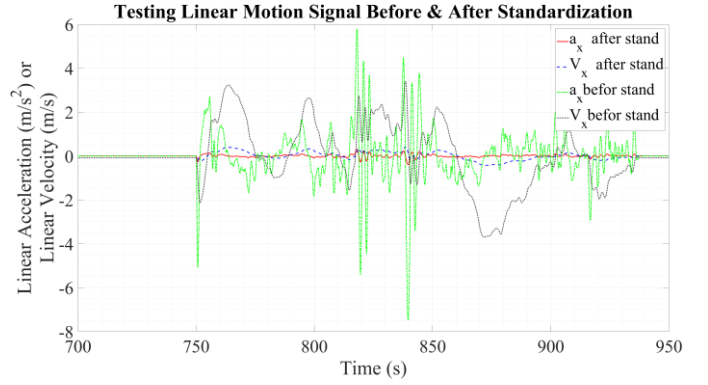


Fig. 4. The test signals of the proposed NN-based classical washout filter including LA and linear velocity before/after standardization along the longitudinal channel.

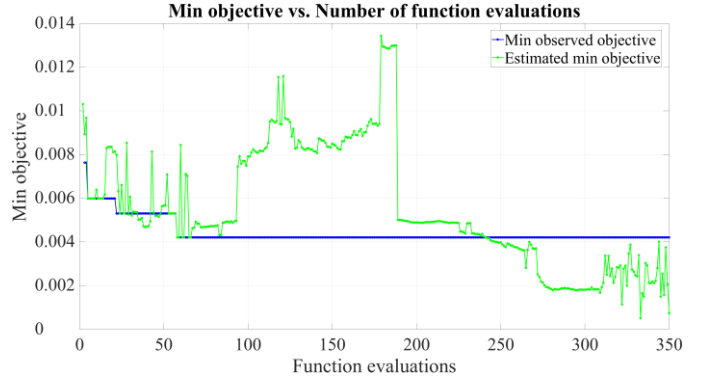


Fig. 5. The MLP fitness function for the translational HP filter using Bayesian optimization.

function of the translational HP filter along the longitudinal channel with the following setting:

1. The optimisation time is 14×60×60 seconds.
2. The maximum number of iterations is 500.

The computational time of Bayesian optimisation is 5886 seconds, which is extremely fast because of the use of parallel computation with GPU-based processing. The Bayesian optimisation algorithm yields 0.11414 and 0.066557 for the estimated and observed objective functions. The same method is followed along the other two sub-channels of the longitudinal channel, namely, tilt coordination and rotational, to identify the optimal MLP models for these sub-channels. Table II shows the setting for search and optimisation of the MLP network structures for the translational, tilt coordination, rotational and sub-channels concerning the longitudinal channel.

TABLE II
THE EXTRACTED OPTIMAL MLP PARAMETERS USING BAYESIAN OPTIMIZATION FOR THREE SUB-CHANNELS.

Item	Translational	Tilt Coordination	Rotational
Number of Layers	4	4	1
Number of Neurons	28	86	10
Maximum Validation Number	22	81	9
Mutation Rate	9.8×10^{-3}	1.0×10^{-4}	1.6×10^{-4}
Sigma	7.6×10^{-5}	1.1×10^{-6}	5.3×10^{-5}

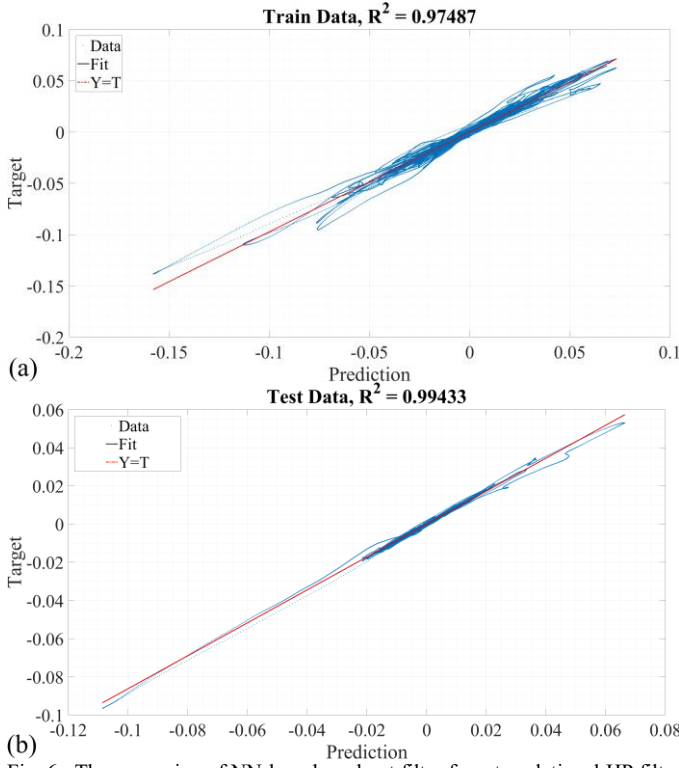


Fig. 6. The regression of NN-based washout filter for a translational HP filter using Bayesian optimization algorithm during (a): training process; (b): testing process.

Fig. 6a-b depict the proposed NN-based MCA results using the parameters identified from Bayesian optimisation concerning a translational HP filter. The horizontal and vertical axes in Fig. 6 indicate the target and obtained outputs. The CC scores between the target and obtained outputs during training and testing are 0.99742 and 0.82079, respectively. The higher CC result of the proposed method proves its filtering efficacy, which is as good as that of the classical washout filter. The R-square scores during the training and testing of MLP are 0.99288 and 0.97125, respectively. In addition, the MSE rates during training and testing of MLP are 0.00021082 and -0.00090221 m/s^2 , with their corresponding standard deviation of error rates of 0.0090 and 0.0085 m/s^2 , respectively.

Fig. 7 illustrates the target and obtained outputs of the proposed NN-based MCA during the test. The target output is generated based on the optimally-tuned classical washout filter using the RoR vehicle motion signals. The motion signals comprise a combination of slow-, medium- and fast-driving scenarios. It is not possible to use the optimally-tuned classical washout filter parameters owing to the violation of the workspace limitations subject to motion signals with abrupt changes. This is the main reason for the worst-case tuning requirement of the classical washout filter. However, Fig. 7 shows higher accuracy of the regenerated linear displacement of the motion simulator between the tuned classical washout filter and the NN-based model.

The proposed method is also implemented along other sub-channels of the longitudinal channel, namely tilt coordination and rotational. The results are summarised in Table III based on

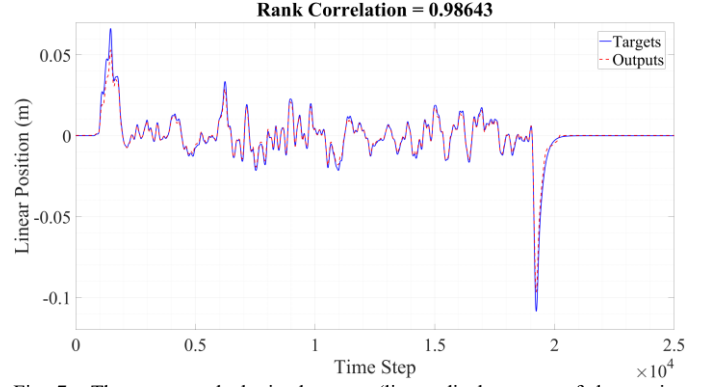


Fig. 7. The target and obtained output (linear displacement of the motion simulator platform) using the ideal classical washout filter and NN-based MCA during the testing process of the model.

the RMSE, R-square, MSE, and CC indicators for all three sub-channels of the longitudinal channel.

Model training using the parameters identified by Bayesian optimisation for the translational HP filter of longitudinal channel takes 42 seconds using parallel computation with GPU processing. The PC specifications used for simulation studies are: Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz 3.60 GHz and NVIDIA GeForce RTX 2060. The training process was terminated after 1888 epoch.

B. Motion Cueing Algorithm

In the following sub-section, the developed NN-based MCA model is employed within the classical washout filter to compute the motion signals of the simulator based on those from the RoR vehicle (Fig. 2b). The human vestibular model is derived in Eqs. (2-3) are employed to compute the motion sensation signals of the RoR vehicle driver and simulator user. The difference between them constitutes the motion sensation error. To have an objective performance comparison, the same scaling factors are employed for the three investigated methods: NN-based MCA, classical washout filter, and MPC-based MCA. They are 1/5 for LA and 1/2 for AV of the RoR vehicle signals along the longitudinal channel.

Fig. 8a-b depict the motion sensation signals of the RoR vehicle driver and the simulator user using NN-based, classical, and MPC-based MCA models along the longitudinal channel. Specifically, Fig. 8a presents the sensed LA signals of the RoR vehicle driver and simulator user from three compared models.

TABLE III
THE RESULTS OF BAYESIAN OPTIMIZATION FOR THREE SUB-CHANNELS OF THE LONGITUDINAL CHANNEL USING ALL DATA.

Item	Translational	Tilt Coordination	Rotational
MSE	4.11×10^{-6}	1.50×10^{-5}	4.45×10^{-4}
RMSE	2.05×10^{-3}	3.87×10^{-3}	2.11×10^{-2}
NMSE	-4.91	-9.36×10^{-1}	$-1.14 \times 10^{+2}$
Mean	1.53×10^{-6}	-3.32×10^{-5}	2.18×10^{-4}
STD	2.03×10^{-3}	3.86×10^{-3}	2.11×10^{-2}
CC	0.99	0.99	0.94

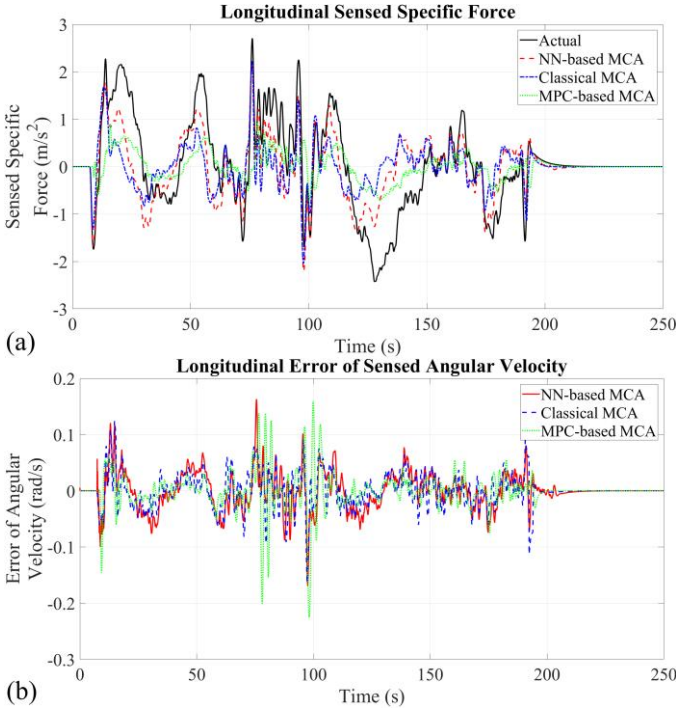


Fig. 8. Sensed specific force for the real vehicle driver and the user of the motion simulator platform using classical washout filter and NN-based MCA along longitudinal channel.

The RMSE score of sensed specific force along the x-axis yielded by the NN-based MCA is 22.40% lower than that from the classical MCA. Also, the RMSE of sensed specific force along the x-axis from the NN-based MCA is 14.69% lower than that from the MPC-based MCA. The CC scores of the sensed specific force between the RoR vehicle driver and simulator user from NN-based, classical, and MPC-based MCA models are 0.7390, 0.4901 and 0.7119, respectively. Fig. 8b presents the sensed AV error signals of the RoR vehicle driver and simulator user from three compared models. The results ascertain a better regeneration of motion cues from the NN-based MCA model for the simulator user, owing to its capability to overcome the worst-case tuning technique used in the classical washout filter. Table IV shows the results in Fig. 8, including CC and RMSE of the motion sensation for the compared methods.

TABLE IV
THE RESULTS OF TWO INVESTIGATED METHODS INCLUDING CLASSICAL WASHOUT FILTER AND NN-BASED MCA.

Index	NN-based MCA	Classical Washout Filter	MPC-based MCA
SSF x-axis	CC = 0.7390	CC = 0.4901	CC = 0.7119
	RMSE = 0.6285 (m ² /s)	RMSE = 0.8099 (m ² /s)	RMSE = 0.7367 (m ² /s)
SAV pitch-angle	RMSE = 0.0331 (rad/s)	RMSE = 0.0304 (rad/s)	RMSE = 0.0336 (rad/s)
Max(LD)	0.1056 (m)	0.0634 (rad)	0.0509 (rad)
Max(AD)	0.0310 (m)	0.2911 (rad)	0.1352 (rad)

SSF: Sensed Specific Force; SAV: Sensed Angular Velocity; CC: Correlation Coefficient; RMSE: Root Means Square Error; LD: Linear Displacement; AD: Angular Displacement; m: meter; rad: radium.

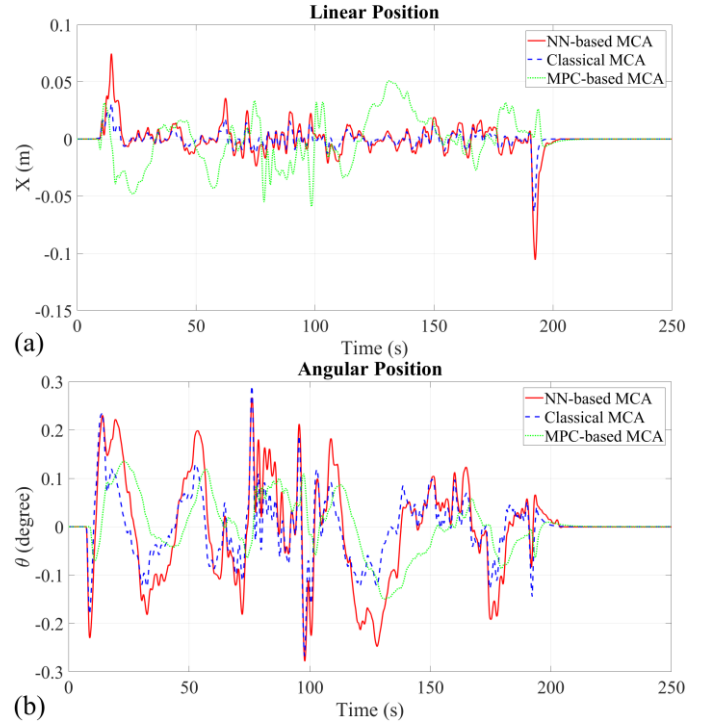


Fig. 9. (a): Linear displacement of the end-effector using NN-based MCA and classical washout filter along the x-axis; (b): Angular displacement of the end-effector using NN-based MCA and classical washout filter along the pitch-angle.

Fig. 9a-b present the linear and angular displacements of the simulator during the test process of the proposed NN-based and standard classical MCA models, respectively. The higher efficiency in usage of the workspace by the NN-based MCA model is depicted in comparison to that of the classical washout filter. In addition, the NN-based MCA model is able to increase the workspace usage efficiency of the simulator (similar to the MPC-based MCA model) without using complicated optimisation methods. The maximum linear displacements of the NN-based, classical, and MPC-based MCA models are 0.1056, 0.0634, and 0.0509 (m), respectively. The maximum angular displacements of the NN-based, classical, and MPC-based MCA models are 15.29, 16.68, and 7.75 (degrees), respectively. As observed, the classical filter tuned with the worst-case scenario exploits the tilt coordination angle more than that from the translational channel. This is another advantage of the NN-based MCA model compared to the classical washout filter. On the other hand, the MPC-based MCA model cannot use the tilt coordination channel efficiently because of the absence of a rate limiter. The NN-based MCA model leverages the linear displacement capability of the simulator instead of the angular displacement capability to increase motion fidelity for the simulator user.

The developed NN-based, classical and MPC-based MCA models are validated and compared using the test signals. The results positively ascertain a better regeneration of the motion sensation signals for the simulator user using the NN-based MCA as compared with those from the standard classical washout filter and MPC-based mode in terms of higher CC and lower RMSE between the RoR vehicle driver and simulator

user. The NN-based MCA model can better exploit the workspace limitations owing to its advantage in avoiding the worst-case tuning technique adopted in the classical washout filter.

V. CONCLUSION

The classical washout filter, the first generation of the washout filter, has been used widely to ensure a motion simulator operates within its workspace constraints while regenerating motion cues for the simulator user to feel the same motion sensation as experienced by a real vehicle driver. As the most popular and commercial form of the MCA, the classical washout filter has several advantages over adaptive, optimal and MPC-based MCA models, which include low computational load, cost efficiency, ease of tuning, and ease of use. However, excluding a human perception model and fixed filter settings (cut-off frequency and damping ratio) are disadvantages of the classical washout filter. These compromise its efficiency as compared to other MCA models. The inefficient use of the workspace of the classical washout filter is due to the worst-case tuning technique for determining the filter parameters. The fast-driving motion signals are widely used for worst-case scenario tuning of the washout filters to ensure the simulator platform does not violate its physical workspace limitations. Therefore, the usage of the platform workspace will be conservative for slow- and medium-driving scenarios, which can cause a large motion sensation error between the driver of a real vehicle and the user of a motion simulator. While advanced control techniques can overcome the drawbacks of the classical washout filter, the associated filter complexity and computational load increase. NN models have recently been used for MCA design, as they do not have high complexity compared to those from advanced control techniques. In this study, the MLP model has been employed to imitate the dynamics of a classical washout filter. Three different filters about the behaviours of motion signals from a vehicle, i.e., slow-, medium- and fast-driving scenarios, have been investigated. The trained MLP model can regenerate highly efficient motion signals for the simulator user without relying solely on the worst-case tuning technique, one of the main disadvantages of classical and other MCA designs. Data standardisation has been implemented to increase the accuracy of the developed MLP model, as it can reduce data complexity. The MLP model has been employed to compute the motion signals of the motion simulator based on that of a real vehicle. The Bayesian optimisation technique has also been leveraged to obtain the optimal MLP parameters, including the number of layers, neurons, and validations and the rates of sigma and mutation. The developed solution has been simulated and evaluated in MATLAB. The results have demonstrated the efficacy of the MLP model and other MCA components. The developed model can regenerate highly accurate motion signals and high-fidelity motion cues with lower error rates for the simulator user than those from a vehicle driver. It should be noted that the proposed NN-based MCA is not restricted to individuals as it has been developed using data samples gathered from different participants. Nonetheless, one

limitation is the need to re-train the model using different road maps in RoR, as the current model has been developed using only data sets generated from one road map. For further work, other deep learning models, such as long-short term memory or convolutional NN models, can be investigated for designing MCAs, along with the MPC technique [35, 36].

REFERENCES

- [1] 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.
- [2] T. Bellmann, J. Heindl, M. Hellerer, R. Kuchar, K. Sharma, and G. Hirzinger, "The dlr robot motion simulator part i: Design and setup," in *2011 IEEE International Conference on Robotics and Automation*, 2011: IEEE, pp. 4694-4701.
- [3] 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.
- [4] J. Iskander *et al.*, "From car sickness to autonomous car sickness: A review," *Transportation research part F: traffic psychology and behaviour*, vol. 62, pp. 716-726, 2019.
- [5] 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.
- [6] 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.
- [7] G. Zacharias, "Motion sensation dependence on visual and vestibular cues," Massachusetts Institute of Technology, 1977.
- [8] B. Conrad and S. Schmidt, "The calculation of motion drive signals for piloted flight simulators," 1969.
- [9] L. Reid and M. Nahon, "Flight simulation motion-base drive algorithms: part I. developing and testing equations," *UTIAS Report, No. 296*, 1985.
- [10] F. Colombet, M. Dagdelen, G. Reymond, C. Pere, F. Merienne, and A. Kemeny, "Motion cueing: What is the impact on the driver's behavior," in *Proceedings of the Driving Simulation Conference*, 2008, pp. 171-181.
- [11] A. H. J. Jamson, *Motion cueing in driving simulators for research applications*. University of Leeds, 2010.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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*, vol. 15, no. 1, pp. 445-453, 2020.
- [18] 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. 51, no. 10, pp. 6096-6110, 2019.

- [19] M. R. C. Qazani *et al.*, "A New Pre-positioning Technique of a Motion Simulator Platform Using Nonlinear Model Predictive Control and Recurrent Neural Network," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [20] 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.
- [21] C. Rengifo, J.-R. Chardonnet, D. Paillot, H. Mohellebi, and A. Kemeny, "Solving the constrained problem in model predictive control based motion cueing algorithm with a neural network approach," 2018.
- [22] A. B. Koyuncu, E. Erçelik, E. Comulada-Simpson, J. Venrooij, M. Kaboli, and A. Knoll, "A Novel Approach to Neural Network-based Motion Cueing Algorithm for a Driving Simulator," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020: IEEE, pp. 2118-2125.
- [23] R. J. Telban and F. M. Cardullo, "Motion cueing algorithm development: Human-centered linear and nonlinear approaches," *NASA TechReport CR-2005-213747*, 2005.
- [24] H. Asadi, S. Mohamed, D. Rahim Zadeh, and S. Nahavandi, "Optimisation of nonlinear motion cueing algorithm based on genetic algorithm," *Vehicle System Dynamics*, vol. 53, no. 4, pp. 526-545, 2015.
- [25] J. M. Van Verth and L. M. Bishop, *Essential mathematics for games and interactive applications*. CRC Press, 2015.
- [26] E. Groen and W. Bles, "How to use body tilt for the simulation of linear self motion," *Journal of Vestibular Research*, vol. 14, no. 5, pp. 375-385, 2004.
- [27] G. Reymond and A. Kemeny, "Motion cueing in the Renault driving simulator," *Vehicle System Dynamics*, vol. 34, no. 4, pp. 249-259, 2000.
- [28] "Rigs of Rod," <https://www.rigsofrods.org/> (accessed 18/03/2021).
- [29] F. Rosenblatt, *The perceptron, a perceiving and recognising automaton Project Para*. Cornell Aeronautical Laboratory, 1957.
- [30] S. Haykin and N. Network, "A comprehensive foundation," *Neural networks*, vol. 2, no. 2004, p. 41, 2004.
- [31] R. N. Longenecker, *Paul, Apostle of Liberty*. Wm. B. Eerdmans Publishing, 2015.
- [32] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimisation algorithm," in *Proceedings of the genetic and evolutionary computation conference GECCO-99*, 1999, vol. 1: Citeseer, pp. 525-532.
- [33] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimisation with unknown constraints," *arXiv preprint arXiv:1403.5607*, 2014.
- [34] B. Conrad and S. F. Schmidt, "A study of techniques for calculating motion drive signals for flight simulators," 1971.
- [35] K. Kumar *et al.*, "SpinalXNet: Transfer Learning with Modified Fully Connected Layer for X-Ray Image Classification," in *2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE)*, 2021: IEEE, pp. 1-7.
- [36] M. R. C. Qazani, B. S. Bidabadi, H. Asadi, S. Nahavandi, and F. S. Bidabadi, "Multiobjective Optimization of Roll-Forming Procedure Using NSGA-II and Type-2 Fuzzy Neural Network," *IEEE Transactions on Automation Science and Engineering*, 2023.