Cognitive Robotics 4 (2024) 116–127

Contents lists available at ScienceDirect

Cognitive Robotics

journal homepage: http://www.keaipublishing.com/en/journals/cognitive-robotics/

High-fidelity learning-based motion cueing algorithm by bypassing worst-case scenario-based tuning technique

Mohammad Reza Chalak Qazani [a,*], Houshyar Asadi [b], Zoran Najdovski [b], Shehab Alsanwy [b], Muhammad Zakarya [a], Furqan Alam [a], Hassen M. Ouakad [c], Chee Peng Lim [b], Saeid Nahavandi [d,e]

[a] *Faculty of Computing and Information Technology (FoCIT), Sohar University, Sohar, Sohar 311, Oman*
[b] *Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, VIC 3216, Australia*
[c] *Renewable Energy Engineering Department, Mediterranean Institute of Technology, South Mediterranean University, Les Berges du Lac 2, 1053, Tunis, Tunisia*
[d] *Swinburne University of Technology, Hawthorn, VIC, Australia*
[e] *Harvard Paulson School of Engineering and Applied Sciences, Harvard University, Allston, MA, USA*

ARTICLE INFO

Keywords:
Motion-planning
Motion cueing algorithm
Simulator platform
Learning methods
Multilayer perceptron
Simulation
Worst-case tuning

ABSTRACT

The motion cueing algorithm (MCA) enhances the realism of simulator driving experiences by generating vehicle motions within platform limitations. Existing MCAs are typically tuned for worst-case scenarios, limiting their efficiency for medium or slow driving motions. This study proposes a comprehensive MCA unit using learning-based models to overcome this problem and efficiently utilise the simulator workspace for all driving scenarios. Data samples are regenerated to cover various motion signal levels, and three classical washout filters are tuned to extract optimal motion signals. A multilayer perceptron (MLP) is trained with these extracted datasets, forming an AI-based MCA that provides high-fidelity driving motions for any scenario while optimising the platform workspace. Simulink/MATLAB is used for modelling and evaluation. Results demonstrate the proposed model's superior performance, with lower motion sensation errors, a higher correlation between sensed motion signals, and more efficient platform workspace usage.

1. Introduction

The application of motion simulators is not only limited to the automobile sector but also they are widely being used in aviation centres, research laboratories and institutes for training of users (air/sea/land vehicles), rapid virtual prototyping and testing of new vehicles, and study of motion sickness [1,2]. In the human body, the vestibular system is the main motion sensory organ in charge of detecting motion [3]. On the other hand, the visual system in our body detects visual cues from any motion via the eyes [4]. A simulator generates motion signals using parallel, serial, or mixed structure manipulators in multiple degrees of freedom (DoF). Motion sickness happens if there is a difference between the perceived vestibular cues and visual [5]. The motion cueing algorithm (MCA) decreases the mismatch between these two sensing mechanisms in our body, enhancing the realism of the motion sensation experience for the simulator user. The combination of a highly restricted motion simulator platform and a poor MCA unit (which can be improved via precise modelling and implementation) is the main reason for the reproduction of inaccurate motion signals for the user of simulator, leading to a high chance of experiencing motion sickness.

* Corresponding author.
E-mail address: m.r.chalakqazani@gmail.com (M.R.C. Qazani).

https://doi.org/10.1016/j.cogr.2024.07.001
Received 15 August 2023; Received in revised form 10 January 2024; Accepted 22 July 2024
Available online 23 July 2024

Conrad and Schmidt [6,7] developed the first MCA, a classical washout filter. It was initially designed for a land vehicle simulator. Reid and Nahon [8,9] modified the MCA model developed in [6,7] for the flight application. The classical washout filters suffer from fixed parameters, including damping ratio and cut-off frequency. Also, it cannot consider the workspace limitation of the simulator and the human vestibular model. In addition, there are three other MCA types: adaptive, optimal and model predictive control. These are introduced in order to solve the shortcomings of the previous methods. However, the classical washout filter is used in industry-based applications owing to its applicability, simplicity, low computational load, ease of tuning, stability and availability [10,11].

Previous researchers [12-20] used the idea of evolutionary algorithms in order to identify the optimal settings of a classical washout filter to satisfy the workspace limitations with low motion sensation error between the motion simulator and actual vehicle users.

The existing washout filters [6-20], including non-optimal and optimal settings, are generally tuned using the worst-case driving scenario. This tuning technique decreases the MCA efficiency in regenerating other driving scenarios, such as medium or slow driving motion signals, due to inefficient use of the limitations of the simulator workspace. This study aims to develop learning-based models using comprehensive motion signals to increase MCA efficiency. This study records three different levels of motion signals, i.e., slow, medium, and fast driving scenarios, to find the optimal settings of classical washout filters for each motion signal. The multilayer perceptron (MLP) model is trained offline and then employed within the classical washout filter instead of using either high-pass or low-pass filters. The input and output of these filters are recorded for training the MLP network. The essential advantage of using learning-based methods instead of filters is eliminating the problem associated with worst-case scenario tuning techniques aiming to reproduce highly efficient motion cues for different levels of motion signals.

The following Section investigates the classical washout filters, focusing more on the previously proposed methods. The human vestibular model, data regeneration and the MLP are explained in Section 3 as the proposed method of this study. The proposed Learning-based MCA is validated using MATLAB software, and its functionality is compared with the classical washout filter in Section 4. The conclusions of this research are mentioned in Section 5.

2. Previous studies

Fig. 1 represents a schematic of a classical washout filter for longitudinal mode. The translational sub-channel combines a second-order high-pass filter and two integrators. The second-order high-pass filter is used to remove the low-frequency linear acceleration (LA) signal as follows:

$$\frac{s^2}{s^2 + 2\zeta_h\omega_h + {\omega_h}^2} \tag{1}$$

where ζ_h and ω_h are the damping ratio and cut-off frequency of the high-pass second-order filter, respectively. Then, an actual vehicle's linear displacement (LD) can be reproduced in a bounded linear workspace of the motion simulator platform. The cut-off frequency of the second-order high-pass filter must be tuned based on the linear constraints of the end-effector within the platform workspace. The damping ratio also affects the response time of the high-pass filter.

The tilt coordination sub-channel combines a second-order low-pass filter, gain, and rate limiter, as follows:

$$\frac{s^2}{s^2 + 2\zeta_l\omega_l + {\omega_l}^2} \tag{2}$$

where ζ_l and ω_l are the damping ratio and cut-off frequency of the low-pass second-order filter, respectively. This sub-channel is used to tilt the driver's seat under the human threshold unit to use the LA of gravity as a low-frequency part of the LA signal. This is

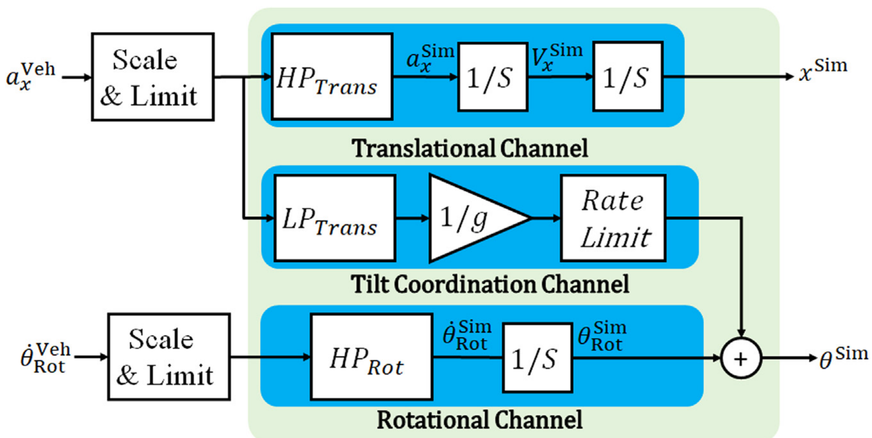

Fig. 1. The schematic of the classical washout filter along the longitudinal channel consists of three sub-channels.

also known as somatogravic illusion [21,22]. The second-order low-pass filter is responsible for sensing the low-frequency LA signal. Then, the LA signal (low-frequency portion) is divided into gravity (9.81 m/s^2) motion using a gain unit. The cut-off frequency is tuned while considering the angular workspace of the simulator platform.

The rotational sub-channel consists of the first-order high-pass filter and an integrator unit, as follows:

$$\frac{s}{s+\omega_{h\theta}} \tag{3}$$

where $\omega_{h\theta}$ is the cut-off frequency of the high-pass first-order filter, respectively. The first-order high-pass filter filters an actual vehicle's low-frequency angular velocity (AV) signal to compute the motion simulator's AV. Meanwhile, the integrator unit calculates the platform's angular displacement (AD) via the simulator's AV. The input motion signal should be scaled and limited before applying it to the sub-channels to not violate the workspace boundaries of the platform. The end-effector's AD is extracted via rotational and tilt coordination sub-channels.

Asadi et al. [12] genetic algorithm-based classical washout filter to improve the human sensation, maximize reference signal following, and enhance the platform's efficiency within motion constraints, outperforming classical washout filters in performance, sensation error, and workspace utilization.

Casas et al. [13] introduced a simulation-based approach for tuning classical washout filters in vehicle simulators. Their proposed method offers a faster and alternative solution to the traditionally subjective and time-consuming tuning process using objective metrics and a genetic algorithm on a motion simulator.

Thöndel [14] used the GA toolbox of MATLAB to select the optimal settings for classical washout filters to eliminate rippling and minimise jerk effects. The use of learning-based models in MCA design is new. They have a high potential to enhance the efficiencies of the MCAs, such as classical methods, without increasing system complexity. De Mooij et al. [15] adapted the classical washout filter to reduce latency and false cues. Their proposed method was based on the minimum usage of the filters based on the facing motion scenarios. Positive feedback from naive subjects and professional drivers suggests improved simulator fidelity and reduced motion sickness. Qazani et al. [16] used a butterfly optimization algorithm to extract the optimal horizons for model predictive MCA. The main contribution of their study was the cheap computational load with lowering the motion sensation error between the virtual vehicle driver and the simulator user. Pham [17] introduced an auto-tuning method (using mean-variance mapping optimization) to efficiently adjust the parameters of motion cueing algorithms for driving simulators, addressing the challenges of time-consuming manual tuning. Qazani et al. [18] used the grey wolf optimiser method to vary the weight of the model predictive MCA to increase the efficiency of motion cues for the user for urban driving scenarios. The grey wolf optimiser picks the most efficient weight for model predictive control facing the abrupt motion signals. Pham and Nguyen [19] recently introduced a novel motion platform using a KUKA Robocoaster industrial robot for driving simulators. It presents the development and application of MCAs in cylindrical coordinates, addressing rotational motion challenges. It proposes an auto-tuning method (using mean-variance mapping optimization) for efficient parameter adjustment, demonstrating improved performance and reduced false cues in the simulation. Recently, Qazani et al. [20] proposed the cascade meta-heuristic tuning technique for an extremely complicated nonlinear model predictive MCA to bypass the time-consuming tuning process. They reached the highest efficiency considering the computational load inside the cost function.

3. Methodology

This Section describes the methodology for investigating the newly proposed learning-based MCA and classical washout filter. Fig. 2 presents the schematic structure of the newly proposed method. It consists of motion signal regeneration, pre-processing, training and evaluation of the developed learning-based solution. In addition, the Bayesian regularisation is used as the training function of the proposed MLP. The data samples for training and evaluation are regenerated to cover all levels of motion signals, including

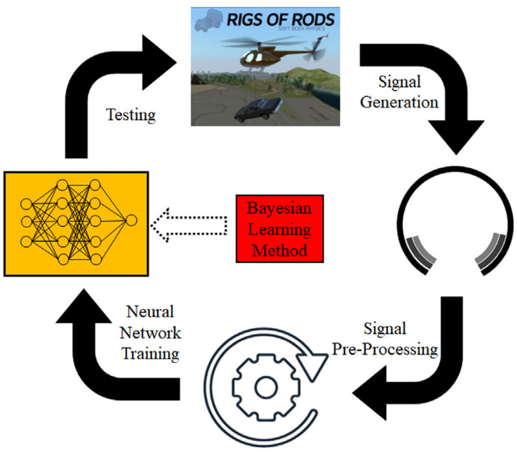

Fig. 2. The structure of the proposed method includes data regeneration, data pre-processing, training network and testing.

slow, medium, and fast driving scenarios and their combinations. These data samples are processed with the Simulink/MATLAB software package to develop the classical washout filters. The Simulink module of the classical washout filters aims to produce the best structure of the MCA for processing fast, medium and slow driving motion signals. The main module in MATLAB is composed of the MLP network to extract the optimal motion signals for the simulator based on the current driving scenario without facing the problem associated with the worst-case tuning technique that causes inefficient usage of the platform workspace. The following steps should be followed in order to duplicate the same procedure which is developed in this study:

Step 1: In the first step, the Rigs of Rod (RoR) vehicle simulation environment generates the motion signals, including linear acceleration and angular velocity in three different channels (longitudinal, lateral, and vertical). The moderator supervises the user of the RoR in driving the virtual vehicle with three different strategies, including slow, medium, and fast driving modes. The gathered information from the 10 participants is stored in the data storage area for the next step, which is the optimal tuning of the classical washout filter.

Step 2: The gathered motion signals are categorized into slow, medium, and fast. Each category is used for optimal tuning of the classical washout filter to regenerate the optimal motion signal for each assigned group. For instance, the optimal tunned classical washout filter for medium motion signals can regenerate the optimal motion cues for medium motion signals. At the end of this step, there are three different classical washout filters, each capable of regenerating the optimal motion cues for their assigned motion scenarios.

Step 3: The three optimally tuned classical washout filters extracted are used to generate the optimal motion cues for the simulator. As there are three different sub-channels inside each channel in a classical washout filter (Fig. 1), there is a need to develop three different neural network-based models to imitate the same behaviour as high- and low-pass filters. Based on Fig. 1, there is a high-pass filter for the translational channel, a low-pass filter for the tilt coordination channel, and a high-pass filter for the rotational channel. Then, the output of these three filters is recorded to be assigned as the system's output during the system training process in the next step.

Step 4: The input and output of the system, which is recorded in Step 3, are defined to supervised neural networks model to accomplish the same process as filters after training. The MLP is chosen to accomplish this stage because of the low computational load in the training process for a fast system implementation. After training these three MLP systems, they will be substituted with high- and low-pass filters inside the structure of the classical washout filter. The final stage is the practical evaluation of the system with the implementation of testing motion scenarios without any limitations to the magnitude of the motion signal.

The following sub-sections explain more details of each step, focusing on the human vestibular model, data generation and MLP structure.

3.1. Human vestibular model

The human perception system infers our body's motion direction and speed using visual, proprioceptive, and vestibular inputs. The human vestibular system is a part of the bony labyrinth, which is placed in the inner ear and consists of the semicircular canals and otolith organs. It mainly contributes towards motion sensation [3].

In this study, we employed the model of otolith organs formulated by Telban and Cardullo [3] to compute the sensed specific force (SSF), $\hat{f}$, using the applied LA signal, f, as follows:

$$\frac{\hat{f}(s)}{f(s)} = K_{\mathrm{OTO}} \frac{\left(\tau_a s + 1\right)}{\left(\tau_L s + 1\right)\left(\tau_s s + 1\right)} \tag{4}$$

where τ_a, τ_s, and τ_L are the otolith organs' time constants. In addition, K_{OTO} specifies the static sensitivity regarding the afferent firing rate per acceleration unit.

Also, the mathematical model of semicircular canals proposed by Telban and Cardullo [3] is employed to compute the sensed angular velocity (SAV), $\hat{\omega}$, based on the applied angular velocity, ω, as follows:

$$\frac{\hat{\omega}}{\omega} = \frac{G_{\mathrm{SCC}} \tau_1 \tau_b s^2 \left(\tau_l s + 1\right)}{\left(\tau_b s + 1\right)\left(\tau_1 s + 1\right)\left(\tau_2 s + 1\right)} \tag{5}$$

where τ_1, τ_2, τ_b, and τ_l are the time constants of the semicircular model. In addition, G_{SCC} is used to scale the response of the semicircular canals known as the AV threshold unit.

The combination of the gravity and LA motion signals sensed by the otolith organs (SSF), $\mathbf{S}_F$, can be formulated as:

$$\mathbf{S}_F = \mathbf{a} - \mathbf{R}(\phi, \theta, \psi)\mathbf{g} = \begin{bmatrix} a_x - g\theta \\ a_y - g\phi \\ a_y - g \end{bmatrix} \tag{6}$$

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

Table 1
The Results of the two investigated methods include classical washout filter and learning-based MCA.

Item	Translational	Tilt	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	$\omega_n = 1.8$ $\xi = 2$	$\omega_n = 6.25$ $\xi = 1$	$\omega_n = 0.1$

3.2. Data generation

RoR is an open-source vehicle simulation environment that can be used for virtual driving and recording vehicle motion signals [23]. In this research, RoR is used to generate four different motion signals. The virtual vehicle driver is guided to drive the car using four strategies. Initially, the driver drives fast because the goal is to reach the target position as fast as possible. Then, the driver is usually instructed to drive, similar to driving in an urban area. The third one is the slowest driving scenario, and the aim is to minimise the usage of gas and brake pedals. These signals are employed during the network training process. The filter input comprises one-dimensional data. In this work, the filters' inputs are the unfiltered driving motion signals of the game environment (AV and LA). Increasing the number of features in the learning-based model can increase network accuracy. As such, the integrators of filter inputs are chosen as the second feature for the training/testing of the MLP network. In addition, the difference between these two parameters (AV and LA) is selected as the third feature of the model. Then, the input signals of the translational high-pass filter learning-based model are LA, linear velocity (LV), and the difference between LV and LA. The same input is used for the low-pass filter learning-based model in the tilt coordination channel of the MCA.

Meanwhile, the AV, AD and the difference between AV and AD are the input signals of the rotational high-pass filter learning-based model. The delay function is used to increase the input size and learn the system about the signal's past in the learning-based model based on the specified delays. The delays chosen in this study are:

$$\mathbf{In} = \left[\left(a_x\right)_{1:100:1}, \left(a_x\right)_{102:200:2}, \left(a_x\right)_{204:300:4}, \left(a_x\right)_{414:500:32}\right] \tag{7}$$

The same angular velocity delay should be applied to extract the input dataset.

The first and second arguments are the start and endpoints. At the same time, the third is the step utilised to convert the one-dimensional signal into a 30-dimensional counterpart.

After regenerating the motion signals, the investigated classical washout filter in Section 2 is designed and modelled in Simulink/MATLAB. Classical washout filters are tuned based on three different generated motion signals in RoR. The fixed filter parameters for the longitudinal mode are shown in Table 1, which satisfy the driving strategies of fast, medium, and slow. The longitudinal workspace boundary of the motion simulator is ± 0.12 (m) for LD and ± 0.33 (rad) for AD. As presented in Table 1, the translational high-pass filter's cut-off frequency for fast driving is higher than that in medium and slow driving. This is because the fast-driving scenario has more abrupt motion signals than those of medium and slow-driving modes. The cut-off frequency of the low-pass filter in the tilt coordination channel for the fast-driving mode is lower than those of the medium and slow-driving modes. These fixed-term parameters are employed inside each corresponding Simulink file of the classical washout filter to yield the best solution for each driving strategy.

3.3. Multilayer perceptron

MLP is a fully connected feedforward neural network aiming to create a set of outputs from a set of inputs, which was invented by Rosenblatt [24]. MLP is composed of three layers, including output, hidden and input. The neurons of hidden and output nodes are nonlinear activation functions, while they are linear for input nodes.

MLP is widely used for solving regression problems. In contrast, it can solve classification problems using supervised learning methods. Supervised learning is constructed using labelled datasets to train the algorithms to classify/predict outcomes. The optimal weights and biases are typically extracted using the backpropagation algorithm using validation methods such as mean square error (MSE) or root mean square error (RMSE). It should be noted that the backpropagation method uses a stochastic gradient descent technique in order to backward the weights and biases through the MLP. The k^{th} output node in the n^{th} data point can be calculated during the training process as:

$$e_k(n) = t_k(n) - y_k(n) \tag{8}$$

where t and y are the target and actual outputs, respectively.

The minimisation of the error of the whole network output, $\varepsilon(n)$, is used to calculate the weights as follows:

$$\varepsilon(n) = \frac{1}{2}\sum_k e_k^2(n) \tag{9}$$

As the gradient descent method is used in the backpropagation algorithm, it can be extracted as:

$$\Delta w_{ij}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial z_i(n)} y_i(n) \tag{10}$$

where z_i and y_i are the previous neuron's induced local field and output, respectively. Also, η is the learning rate, which specifies the convergence speed of the MLP. The simplified form of the weight derivative concerning the induced local field can be calculated as follows:

$$-\frac{\partial \varepsilon(n)}{\partial z_i(n)} = e_k(n)\phi'\left(z_k(n)\right) \tag{11}$$

where ϕ', the activation function's derivative. Based on the above-represented relations, the weight derivative can be formalised as follows:

$$-\frac{\partial \varepsilon(n)}{\partial z_k(n)} = \phi'\left(z_k(n)\right) - \sum_j -\frac{\partial \varepsilon(n)}{\partial z_i(n)} w_{jk}(n) \tag{12}$$

Based on Eq. (12), the weight of the j^{th} node (output layer) affects the variation of the hidden node's weight. In addition, the variation of the hidden layer weights affects the output layer weights based on the activation function's derivative. This is how the activation function's backpropagation [25].

Based on trial and error, we have established that the learning-based model cannot imitate the high-pass filter's behaviour very accurately, as it is necessary to washout the end-effector movement to keep the platform within its workspace boundaries. The high-pass filter output is integrated to calculate the linear and angular configurations of the end-effector in translational and rotational sub-channels, respectively. The LD and AD are chosen as the target outputs of the high-pass filters, which need to be used for training.

4. Results and discussion

This Section analyses the outcomes of the proposed learning-based model, which covers pre-processing, optimisation, training, and testing. The LV and the difference between LA and LV are employed as the second and third inputs of the MLP (along the translational sub-channels of the longitudinal mode) to enhance the accuracy of the proposed method. The proposed technique is validated against the classical washout filter, which is tuned based on the worst-case driving scenario. Fig. 3 shows the user input motion signals, including LA (a_x) and LV (V_x), for three different motion scenarios (slow, medium and fast) and the normal motion scenario for the testing purpose of the network. The human vestibular model derived in Eqs. (1-2) are employed to compute the motion sensation signals of the actual vehicle driver and motion simulator user. The difference between them constitutes the error in motion sensation.

Both classical and learning-based MCAs are developed using the Simulink/MATLAB software. The proposed learning-based MCA with Bayesian regularisation training function and comprehensive motion signals are coded in MATLAB to formulate the best MLP model using offline supervised training. It aims to overcome the disadvantages of classical washout filters, primarily fixed parameters and insufficient workspace usage. The standardisation method is employed in this study to decrease the complexity of the input signals before using them for network training.

In the first step, the effect of different numbers of layers and neurons on MLP accuracy is evaluated. Fig. 4 shows the contour representation of the error between the learning-based MCA and classical washout filter filtered LA signal in the translational sub-channel of the longitudinal channel. It depicts the arrangements of neurons and layers that are suitable to reach the suitable network with lower error between the target and obtained outputs. It is configured with a lower resolution. Then, the resolution is increased in the area with lower error to identify the best possible solution. The error is calculated using the RMSE between the target and obtained outputs.

After capturing the optimal number of neurons and layers for three assigned classical washout filters (high-pass filter for translational channel, low-pass filter for translational channel, and high-pass filter for rotational channel), the optimal rate of mutation and sigma is captured using the same methodology. The optimal number of neurons, number of layers, mutation rate, and sigma rate are reported in Table 2 after doing a comprehensive case study as error and trial. These extractor hyperparameters in Table 2 are employed inside the model during the network's training stage to reach the system's highest possible accuracy.

Three learning-based models are trained based on the provided datasets in Fig. 3a-c for slow, medium, and fast motion scenarios. The results of three extracted learning-based washout filters are reported in Table 3. It shows the mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), correlation coefficient (CC), and R-square between the target (signals filtered by

Table 2
The extractor optimal hyperparameters of the three learning-based washout filters, including a high-pass filter for a translational channel, a low-pass filter for a translational channel, and a high-pass filter for a rotational channel.

Learning-based Washout Filter	Number of Layers	Number of Neurons	Mutation	Sigma
High-pass Filter for Translational Channel	4	28	9.82×10^{-3}	7.57×10^{-5}
Low-pass Filter for Translational Channel	4	86	1.01×10^{-4}	1.09×10^{-6}
High-pass Filter for Rotational Channel	1	10	1.59×10^{-4}	5.35×10^{-5}

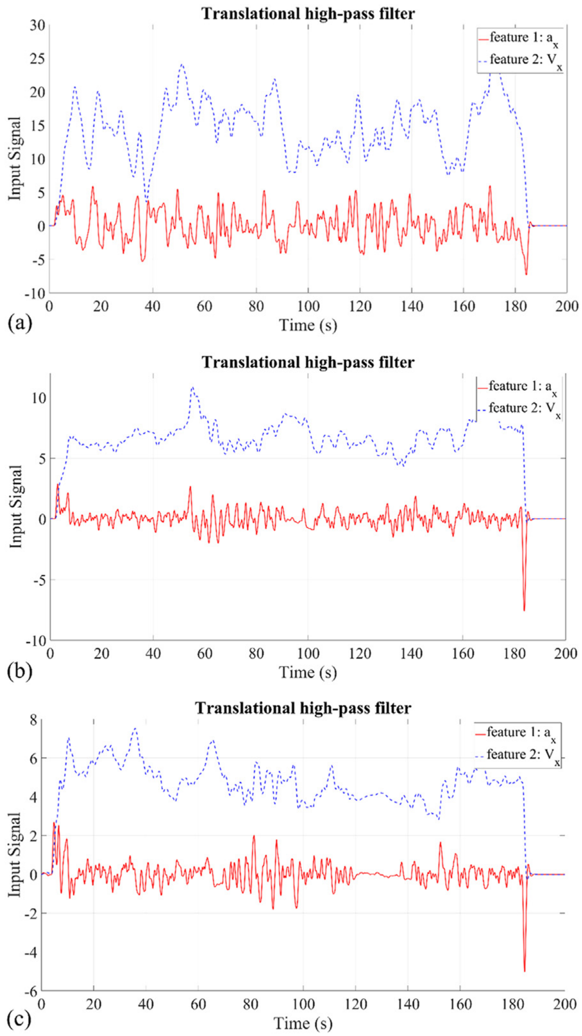

Fig. 3. The input motion signals during the training and testing process, including LA and AV (a): fast driving scenario; (b): medium driving scenario; (c): fast driving scenario.

Table 3
The results were extracted while training and testing learning-based models for three channels.

Learning-based Washout Filter		High-pass Filter for Translational Channel	Low-pass Filter for Translational Channel	High-pass Filter for Rotational Channel
MAE	Training	7.9×10^{-3}	3.73×10^{-2}	6.3×10^{-3}
	Testing	9.5×10^{-3}	2.91×10^{-2}	3.7×10^{-3}
	All	8.1×10^{-3}	3.63×10^{-2}	6.0×10^{-3}
MSE	Training	1.1333×10^{-4}	3.0×10^{-3}	8.18×10^{-5}
	Testing	3.6772×10^{-4}	2.5×10^{-3}	2.71×10^{-5}
	All	1.4309×10^{-4}	2.9×10^{-3}	7.54×10^{-5}
RMSE	Training	1.06×10^{-2}	5.47×10^{-2}	9.0×10^{-3}
	Testing	1.92×10^{-2}	5.04×10^{-2}	5.2×10^{-3}
	All	1.20×10^{-2}	5.42×10^{-2}	8.7×10^{-3}
CC	Training	0.9689	0.9765	0.9879
	Testing	0.9505	0.9603	0.9635
	All	0.9626	0.9755	0.9875
R-Square	Training	0.9387	0.9536	0.9759
	Testing	0.8541	0.9220	0.9231
	All	0.9258	0.9516	0.9752

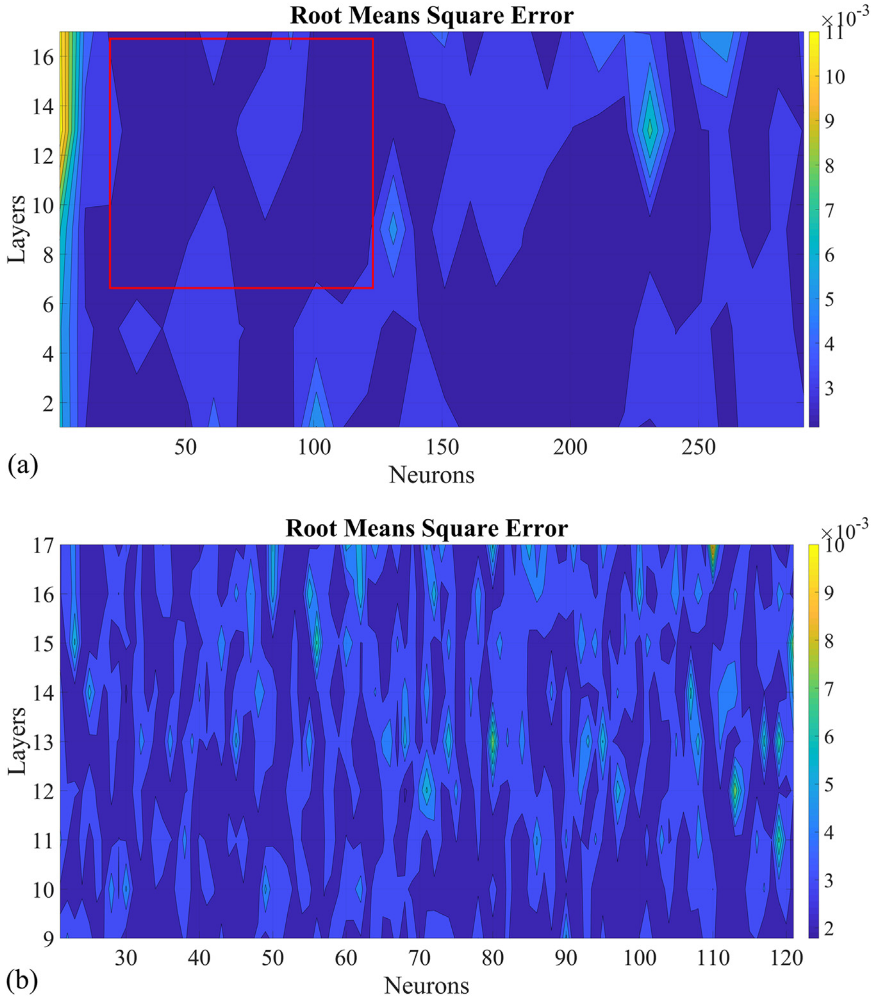

Fig. 4. The contour representation of the error between the target and obtained outputs of the NN-based MCA based on different settings of the translational HP filter (a): layers = 1 to 17 with a resolution of 4 layers, and neurons = 1 to 291 with a resolution of 50 neurons; (b): layers = 9–17 with a resolution of 1 layer and Neurons = 21–121 with a resolution of 1 neuron.

optimal filters) and the predicted (signals filters by learning-based model). While the lower MAE, MSE, and RMSE prove the higher efficiency of the systems, the higher CC and R-square are essential to reach the same motion cueing pattern for the simulator user. The three trained models have high CC and R-Square, which proves their effectiveness in handling the regeneration of high-fidelity motion cues. In addition, the MAE, MSE, and RMSE of the low-pass filter for the translational channel are higher than the other two channels, including high-pass filters for translational and rotational channels. However, it is due to handling the low-pass frequency part of the signal via the tilt coordination channel, as the signal exists for more time in this channel.

Fig. 5 shows the regression of the learning-based model for usage in the high-pass filter of the translational channel during training, validation, testing, and all dataset implementations. The regressions during training, validation, testing, and all dataset implementations are 0.99691, 99662, 0.9966, and 0.99688, respectively.

Fig. 6a-b shows the real vehicle drivers' and simulator users' motion sensation signals using normal and learning-based MCA for the longitudinal mode. Specifically, Fig. 6a presents the SSF signals of the real vehicle driver and motion simulator user from both investigated models. The CC scores (i.e., the shape similarity factor) of the SSF between the real vehicle driver and motion simulator user using learning-based and normal classical washout filters are 0.7390 and 0.4901, respectively. The high dexterity of the learning-based model is obvious in this figure because it bypasses the worst-casing tuning technique in the classical washout filter. In addition, Fig. 6b demonstrates the SAV signals of the real vehicle driver and motion simulator user from both investigated models. Based on the represented results in Fig. 6b, the CC of the SAV between the real vehicle driver and motion simulator user using learning-based and normal classical washout filters are 0.7701 and 0.8587, respectively. As is shown, the CC of classical washout filters is higher than that of the learning-based system. However, it should be mentioned that the human body cannot perceive the motion sensation error below the threshold unit of the human semicircular system. Then, this over-accuracy in the CC of SAV via the classical washout

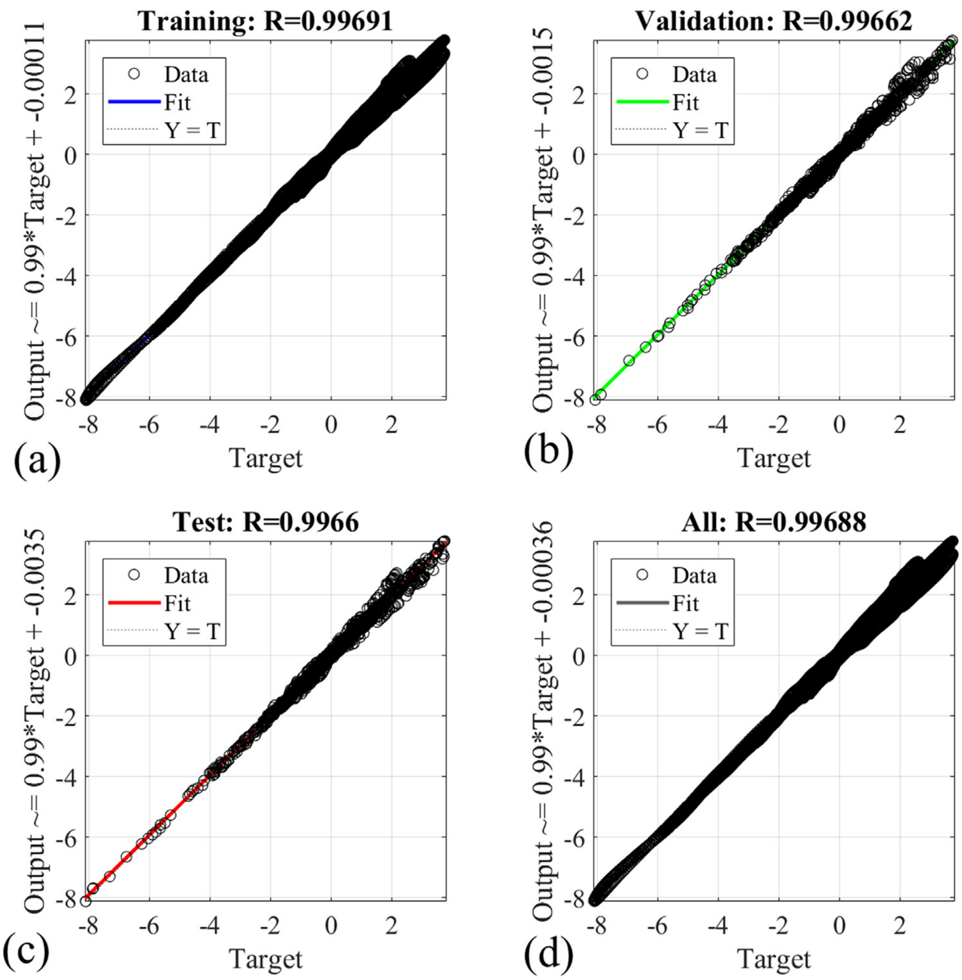

Fig. 5. The regression of the learning-based model for high-pass translational channels during (a): training; (b): validation; (c): testing; (d): all datasets.

filter compared with a learning-based system is useless. This is another advantage of a learning-based system over a classical washout filter for intelligent usage of the semicircular channel.

Fig. 7 demonstrates the difference in motion sensation signals between real vehicle drivers and simulator users using normal and learning-based MCA for the longitudinal mode. The RMSE of SSF along the x-axis for the learning-based MCA is 22.40 % lower than that for the worst-case scenario tunned classical washout filter (Fig. 7a). Fig. 7b presents the SAV error signals of the real vehicle driver and motion simulator user using both investigated MCAs. The RMSEs are 0.0331 and 0.0304 (rad/s) using the learning-based MCA and worst-case scenario tuned classical washout filter for the longitudinal mode. The results ascertain a better regeneration of motion cues from the learning-based MCA for simulator users, owing to overcoming the problem associated with the worst-case tuning technique for the classical washout filter.

Fig. 8a-b presents the LD and AD of the end-effector during the test process of the proposed learning-based and standard classical MCAs using the worst-case scenario tuning strategy, respectively. The higher efficiency in usage of the workspace by the learning-based MCA is depicted in comparison with that of the classical washout filter. The maximum LDs of the learning-based MCA and classical washout filter are 0.747(m) and 0.310 (m), respectively. The maximum AD of the learning-based MCA and classical washout filter are 15.29 (degrees) and 16.68 (degrees), respectively. As observed, the classical filter tuned using the worst-case scenario exploits the tilt coordination angle more than the translational channel. This is another advantage of learning-based MCA compared to the classical washout filter. Indeed, it is preferred to leverage the LD capability of the motion simulator instead of the AD capability to increase motion fidelity for the simulator user.

Table 4 shows the represented results in Figs. 6-8, including CC and RMSE of the motion sensation for both investigated methods. As the statistical results in Table 4 reveal that the dexterous usage of the workspace area using a learning-based model (linear displacement of 0.747 m and 0.310 rad) compared with classical washout filter (linear displacement of 0.2669 m and 0.2911 rad)

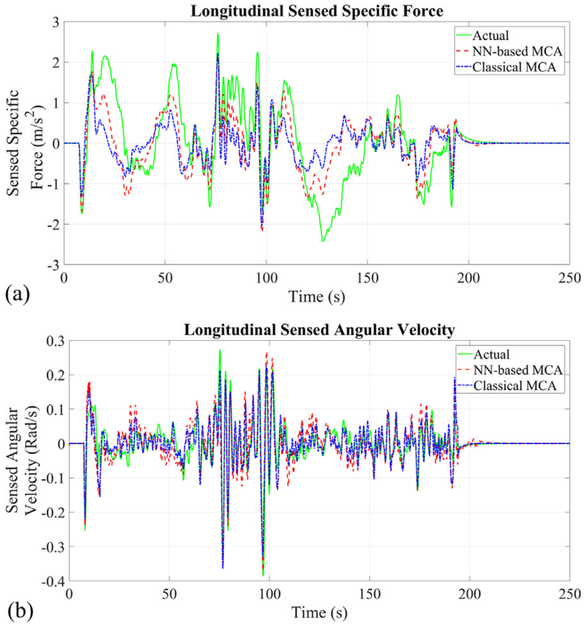

Fig. 6. The real vehicle and the motion simulator's users using classical washout filter and learning-based MCA for longitudinal mode, including (a): SSF; (b): SAV.

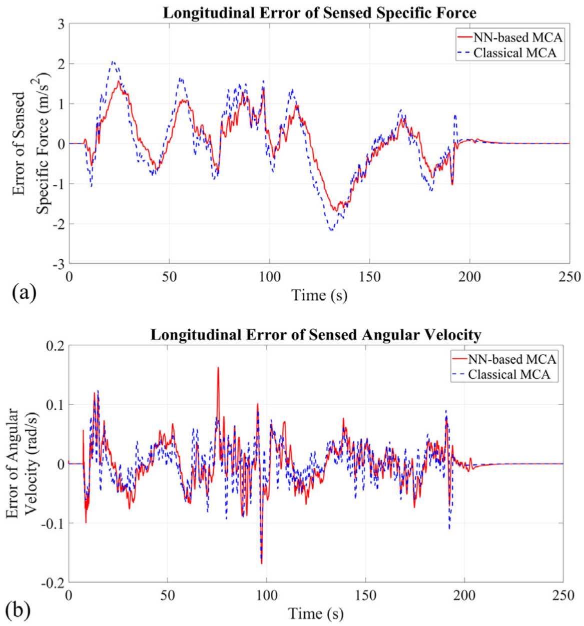

Fig. 7. The real vehicle and the motion simulator's users using classical washout filter and learning-based MCA for longitudinal mode, including (a): error of SSF; (b): error of SAV.

leads the better regeneration of SSF motion signal using the learning-based model with higher fidelity (CC of 0.7390 and RMSE of 0.6285 m/s^2) compared with classical washout filter (CC of 0.4901 and RMSE of 0.8099 m/s^2).

The Simulink models developed based on the learning-based model and classical washout filters are validated and compared using the test signals. The results positively ascertain a better extraction of the motion sensation signals for the simulator user using the learning-based MCA compared with those from the standard classical washout filter due to higher CC and lower RMSE between the vehicle driver and the simulator user. The learning-based MCA can better exploit the workspace limitations owing to its advantage in avoiding the worst-case tuning technique adopted in the classical washout filter.

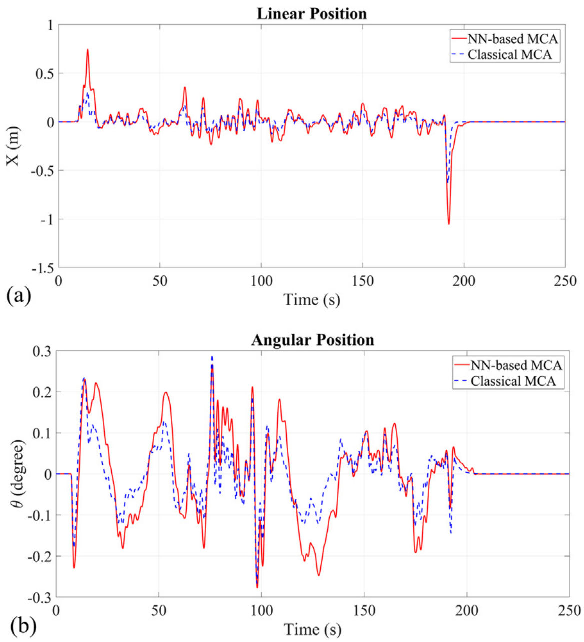

Fig. 8. (a): LD of the platform using learning-based MCA and classical washout filter along the x-axis; (b): AD of the platform using learning-based MCA and classical washout filter along the pitch-angle.

Table 4
The Results of the two investigated methods include classical washout filter and learning-based MCA.

Index	Learning-based MCA	Classical Washout Filter
SSF x-axis	CC = 0.7390	CC = 0.4901
	RMSE = 0.6285 m/s^2	RMSE = 0.8099 m/s^2
SAV pitch-angle	CC = 0.7701	CC = 0.8587
	RMSE = 0.0331 rad/s	RMSE = 0.0304 rad/s
Maximum Linear Displacement	0.747 (m)	0.2669 (m)
Maximum Angular Displacement	0.310 (rad)	0.2911 (rad)

5. Conclusion

The classical washout filter is used as a filtering technique to keep the motion simulator within the workspace constraints while reproducing motion cues for the motion simulator users to deliver them realistic driving motion experiences. The advantages of the classical washout filter are the low computational load, cost efficiency, ease of tuning, and simplicity compared with other advanced forms of MCAs, such as model predictive control, optimal, and adaptive. However, the fixed filter settings (damping ratios and cut-off frequencies), tuned based on the worst-case driving scenario, are the classical washout filter's main disadvantages, dramatically decreasing its efficiency and platform workspace usage. In contrast, advanced control techniques used in the MCA can overcome the drawbacks of classical washout filters. However, they increase the model complexity and computational load. The learning-based models have recently been used in the MCA as they are not highly complex compared to other advanced control techniques. In this study, the MLP model has been employed to imitate the dynamics of the classical washout filter. Three classical washout filters for different motion signals of a real vehicle, i.e., slow, medium or fast driving scenarios, have been initially developed. The MLP model has been utilised to compute the suitable motion signals of the motion simulator based on the motions of a real vehicle so that the simulator's workspace can be efficiently used for all sorts of driving scenarios. The proposed method has been modelled and evaluated using SIMULINK/MATLAB. In summary, learning-based MCA exhibits superior performance with a 22.40 % lower RMSE in SSF and a higher CC score (0.7390) compared to the classical washout filter (CC: 0.4901), ensuring enhanced motion perception for simulator users. It proves that the recently developed model performs better in the workspace.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

CRediT authorship contribution statement

All authors contributed equally to conceptualization, data curation, formal analysis, funding acquisation, investigation, methodology, project administration, resources, software, supervision, validation, writing-orginal draft, and writing-review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Asadi, et al., Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm, IEEE Trans. Syst., Man, Cybern. 47 (2) (2016) 238–254.
[2] T. Bellmann, et al., The dlr robot motion simulator part i: design and setup, 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011.
[3] R.J. Telban, F.M. Cardullo, NASA TechReport CR-2005-213747, 2005.
[4] G. Zacharias, Motion Sensation Dependence On Visual and Vestibular Cues, Massachusetts Institute of Technology, 1977.
[5] J. Iskander, et al., From car sickness to autonomous car sickness: a review, Transp. Res. Part F 62 (2019) 716–726.
[6] Conrad, B. and S.F. Schmidt, *A study of techniques for calculating motion drive signals for flight simulators*. 1971.
[7] Conrad, B. and S. Schmidt, *The calculation of motion drive signals for piloted flight simulators*. 1969.
[8] L. Reid, M. Nahon, UTIAS Report, No. 296, 1985.
[9] Reid, L. and M.A. Nahon, Flight Simulation Motion-Base Drive Algorithms.: part 2, Selecting The System Parameters. UTIAS report, 1986(N307).
[10] Nordmark, S., Driving simulators, trends and experiences. 1994.
[11] A.H.J. Jamson, Motion Cueing in Driving Simulators For Research Applications, University of Leeds, 2010.
[12] H. Asadi, et al., Human perception-based washout filtering using genetic algorithm, International Conference on Neural Information Processing, Springer, 2015.
[13] S. Casas, et al., Towards a simulation-based tuning of motion cueing algorithms, Simul. Model. Pract. Theory. 67 (2016) 137–154.
[14] E. Thöndel, Design and optimisation of a motion cueing algorithm for a truck simulator, 26th annual European Simulation and Modelling Conference, 2012.
[15] M. de Mooij, E. de Vries, J. van Doornik, Adaptation of the classic cueing algorithm for automotive applications, AIAA Scitech 2019 Forum, 2019.
[16] M.R.C. Qazani, et al., Optimising control and prediction horizons of a model predictive control-based motion cueing algorithm using butterfly optimization algorithm, 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020.
[17] D.-A. Pham, Mean-variance mapping optimization for auto-tuning parameters of classical motion cueing algorithm, in: Proceedings of the 2nd Annual International Conference on Material, Machines and Methods for Sustainable Development (MMMS2020), Springer, 2021.
[18] M.R.C. Qazani, et al., An MPC-based motion cueing algorithm using washout speed and grey wolf optimizer, 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2021.
[19] D.-A. Pham, D.-T. Nguyen, Auto-Tuning parameters of motion cueing algorithms for high performance driving simulator based on Kuka Robocoaster, Sci. Prog. 105 (2) (2022) 00368504221104333.
[20] M.R.C. Qazani, et al., An optimal nonlinear model predictive control-based motion cueing algorithm using cascade optimization and human interaction, IEEE Trans. Intell. Transp. Syst. (2023).
[21] E. Groen, W. Bles, How to use body tilt for the simulation of linear self motion, J. Vestibular Res. 14 (5) (2004) 375–385.
[22] G. Reymond, A. Kemeny, Motion cueing in the Renault driving simulator, Veh. Syst. Dyn. 34 (4) (2000) 249–259.
[23] *Rigs of Rod*. 18/03/2021]; Available from: https://www.rigsofrods.org/.
[24] F. Rosenblatt, The perceptron, a Perceiving and Recognizing Automaton Project Para, Cornell Aeronautical Laboratory, 1957.
[25] S. Haykin, N. Network, A comprehensive foundation, Neural Netw. 2 (2004) (2004) 41.