A Prediction of Time Series Driving Motion Scenarios Using LSTM and ESN

Mohammad Reza Chalak Qazani
Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
m.r.chalakqazani@gmial.com

Farzin Tabarsinezhad
The Faculty of New Sciences and Technologies
University of Tehran
Tehran, Iran
farzin.tabarsi@ut.ac.ir

Houshyar Asadi
Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
houshyar.asadi@deakin.edu

Chee Peng Lim
Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
chee.lim@deakin.edu.au

Adetokunbo Arogbonlo
Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
a.arogbonlo@deakin.edu.au

Shehab Alsanwy
Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
sabdulraqeb@gmail.com

Shadi Mohamed
Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
m.r.chalakqazani@gmial.com

Mehrdad Rostami
Centre of Machine Vision and Signal Processing, Faculty of Information Technology
University of Oulu
Oulu, Finland
mehrdad.Rostami@oulu.fi

Saeid Nahavandi
Institute for Intelligent Systems Research and Innovation
Deakin University
Geelong, VIC 3125, Australia
&
Harvard Paulson School of Engineering and Applied Sciences, Harvard University, Allston, MA 02134 USA
saeid.nahavandi@deakin.edu.au

***Abstract*—The motion signals are generated for a simulator user based on the visual understanding of the environment using virtual reality. In this respect, a motion cueing algorithm (MCA) is employed to reproduce the motion signals based on the real driving motion scenarios. Advanced MCAs are required to predict precise driving motion scenarios. Nonetheless, investigations on effective methods for predicting the driving motion scenarios accurately are limited. Current state-of-the-art studies mainly focus on the averaged motion signals from several simulator users pertaining to a specific map or from feedforward neural network and non-linear autoregressive. The existing methods are unable to yield precise predictions of the driving scenarios. In this research, the echo state network and long short-term memory models are employed for the first time in MCA to forecast the driving motion signals. Our evaluation proves the efficiency of our proposed methods in comparison with existing methods.**

Keywords— modelling and prediction, computing methodologies, manipulators, signal processing.

I. Introduction

A motion simulator is a practical tool to simulate the dynamics of vehicles in combination with a virtual reality environment [1-5]. It is a safe and cost-effective tool for improving road safety, investigating intelligent transportation systems, and advancing vehicle manufacturing technologies [6-10]. Despite its usefulness, no simulator can follow the unfiltered driving motion signals because of the constrained workspace [11-20]. As a result, the motion cueing algorithm (MCA) [21, 22] is devised to re-generate the motion sensations for the simulator user to experience the same sensation as if in a vehicle, within the limited working area. MCA is divided to classical [23], adaptive [24-29], optimal [30-33] and model predictive [34-41] methods. In this regard, false motion cues lead to motion sickness in simulator users, which is the main disadvantage of the MCA. Motion sickness [42] occurs where there is a discrepancy between the motion sensation and actual motion as received by the human vestibular and visual systems. The vestibular system inside the inner ear is responsible for detecting the head motion, stabilising the visual axis, and maintaining the body and head postures [43]. The vestibular system consists of the otolith organs for sensing the translational motions and the semicircular canals for sensing the rotational motions. The otolith organs are not able to distinguish the difference between sustainable accelerations from tilt motions. As such, the sustainable acceleration sensation can be re-generated using somatogravic illusion inside the tilt-coordination channel [44].

The state-of-the-art MCAs, such as model predictive control (MPC), have been introduced recently to re-generate the high accurate motion signals [6]. The efficiency of MPC-based MCA with time-varying reference signals depends on the prediction of the driving motion scenarios. In addition, the pre-positioning techniques are useful for varying the end-effector centre to virtually enlarge the linear limitations of the simulator. Knowing the driving motion scenarios in advance along the prediction horizon is able to increase the efficiency of the current pre-positioning techniques through better understanding the motion signals. Mohammadi et al. [45] used the feedforward neural network (NN) in the MCA domain for the first time to predict the driving scenarios using a hidden layer with a size of 36 nodes. However, the method was not accurate enough to accurately anticipate the driving

motion signals in a long prediction horizon. Moreover, other researchers used the averaged motion signals to generate a pool of drivers as the predicted driving scenarios [6].

Both statistical and artificial intelligence methods have been employed to predict the time series data in recent years. Han et al. [46] used recurrent neural network (RNN) in the prediction of the complicated, chaotic time-series data. Their proposed RNN was trained via a self-adaptive backpropagation through time to increase the efficiency for multistep prediction applications. A long short-term memory (LSTM) model is used by Altche and de La Fortelle [47] to estimate the autonomous vehicles' driving motion signal in highways along with longitudinal and lateral channels. The results of their proposed method satisfied both accuracy and safety terms in state-of-the-art applications. A conventional neural network (CNN) model is employed by Lee et al. [48] to predict the lane change for a model predictive controller [49] unit. Xin et al. [50] proposed the ensemble LSTM models to increase the prediction horizon of driving motion signal with indication of driver intention. Deo and Trivedi [51] increased the accuracy of Xin et al. [50] model by substituting the multi-layer LSTM with single-layer LSTM. The employed semantic-based intention and IntentNet is used by Hu et al. [52] and Casas et al. [53] to forecast the motion signals with higher accuracy, respectively. The first CNN-LSTM model in prediction of driving motion signal is proposed by Zhao et al. [54] with concentration on driver action and road status. The prediction of driving motion signal using different scenarios are studied by Mozaffari et al. [55] based on input, output and prediction signals. Qazani et al. [56] improved the efficiency of the Mohammadi's work [45] with consideration of different parameters such as car position in the road. Vural et al. [57] introduced the LSTM-based adaptive learning using second-order Extended Kalman filter (EKF) to reach the online solution without dependency to the future information. Their proposed method was able to increase the accuracy 10%-45% compared to widely-used adaptive methods. Also, It was able to reduce the computational load 10%-15% compared to EKF. Kiran et al. [58] surveyed motion signal prediction methods using reinforcement learning techniques in autonomous vehicles. Qazani et al. [59] introduced the first nonlinear autoregressive (NAR) to predict the driving scenarios using a hidden layer with a size of 45 nodes and 100 time-delays. While it is the simplest and fastest method in forecasting, it yields low accuracy because of using single layer compared with other complex NN models.

The previously proposed methods [45, 59] for the prediction of the driving motion signals in MCAs are not accurate enough because they focused on the lower computational load rather than the precision. Among various deep learning methods, long short-term memory (LSTM) and echo state network (ESN) achieve better results as compared with those traditional prediction methods. The LSTM model was proposed in [60] to address drawbacks of RNN such as strictly dependency on time lag and inability in capturing long-term dependencies with better capability of handling long data sequences using memory blocks. On the other hand, motivated by the development in neurophysiological studies, the ESN [61] has also been successfully used in many areas, including speech recognition, control, robotic, and noise modelling. To the best of the authors' knowledge, ESN and LSTM have not been employed in the motion simulator domain for predictions of driving motion scenarios. Motivated by this research gap, we aim to design and develop a high-efficiency method using ESN and LSTM to anticipate the motion signals.

In section II, the studies related to this research are discussed. The structure of the proposed ESN and LSTM methods are explained in section III. In section IV, the formulated ESN and LSTM models are trained and evaluated in simulated environments using the MATLAB software, with the outcomes discussed and analysed. Concluding remarks of this research are presented in section V.

II. Related Studies

A. Traditional Method

Accurate prediction of time-series driving motion scenarios for use with the MCA is very critical to increasing the efficiency of the re-generated motion cues. The pre-positioning technique is introduced in MCA to vary the centre of the end-effector in order to virtually increase the linear limitation of the simulator. The prediction of driving motion signals allows the pre-positioning algorithm to extract the best off-centre position with respect to the end-effector. Besides that, state-of-art MCA models, such as MPC-based MCA [6], needs the reference signal to obtain the best input signal while respecting the workspace limitations of the simulator platform. As a result, many studies use the time-invariant reference signal in MPC-based MCA because there is very difficult, if not impossible, to precisely predict the reference signals [6].

As there is no systematic procedure to predict the driving motion scenarios, the most straightforward method is to record the motion signals from a pool of drivers and use an average of the generated motion signals as a representation of the predicted motion signals [6]. As different drivers have different driving behaviours, the use of an average of the mean generated motion signals is not reliable in advanced MCA models.

B. Feedforward NN

Mohammadi et al. [45] employed the feedforward NN (with a total of 49,262 inputs and 36 hidden neurons) to train and predict the motion signals for 2.5 seconds of motion. Also, Mohammadi et al. [45] divided the inputs and outputs of the training motion signals into five groups based on the i^{th} sampling time, as follows:

$$x_{\text{in},i} = [x_{i-5} \quad x_{i-4} \quad \cdots \quad x_i] \tag{1}$$

$$x_{\text{out},i} = [x_{i+1} \quad x_{i+2} \quad \cdots \quad x_{i+5}] \tag{2}$$

where x_{in} and x_{out} are the input and output of the network, which is the past and future of the motion signal at the current i^{th} sampling time. The total numbers of input and output motion signals are:

$$U_{\text{in},m} = \sum_{n=1}^{m} x_{\text{in},n}\,, \forall m \in \{-5,-4,\cdots,0\} \tag{3}$$

$$U_{\text{out},m} = \sum_{n=1}^{m} x_{\text{out},n}\,, \forall m \in \{1,2,\cdots,5\} \tag{4}$$

where $U_{\text{in},m}$ and $U_{\text{out},m}$ are the total numbers of input (past) and output (future) motion signals at m^{th} group. Then, the input and output of the network in each group can be calculated as:

$$X_{\text{in},m}(i) = \langle u_n | i + U_{\text{in},m-1} < n < i + U_{\text{in},m} \rangle \quad (5)$$

$$X_{\text{out},m}(i) = \langle u_n | i + U_{\text{out},m-1} < n < i + U_{\text{out},m} \rangle \quad (6)$$

As the feedforward NN is not suitable for a large amount of data, the average data sample pertaining to each group is selected for further analysis to reduce the computational load. As such, the input and output vectors of the overall feedforward NN consist of five elements as follows:

$$NN_{\text{in},i} = [\bar{x}_{i-5}, \bar{x}_{i-4}, \cdots, \bar{x}_i] \quad (7)$$

$$NN_{\text{out},i} = [\bar{x}_{i+1}, \bar{x}_{i+2}, \cdots, \bar{x}_{i+5}] \quad (8)$$

In this study, in order to increase the correlation coefficient (CC) of the prediction, a little amendment is applied in sequencing the inputs and outputs in comparison with Mohammadi et al. [45], which are presented in Eqs. (1-8). The input of all the investigated the algorithm is the history of the motion signal via choosing the following delays:

$$\langle 1{:}\,1{:}\,25\rangle + \langle 26{:}\,2{:}\,50\rangle + \langle 52{:}\,4{:}\,100\rangle + \langle 104{:}\,8{:}\,200\rangle + \langle 208{:}\,16{:}\,500\rangle \quad (9)$$

wherein $\langle a{:}\,b{:}\,c\rangle$, a is the start-point, b is the step, and c is the endpoint. Also, it should be noted that the output sequence is the input data for the next time-step. In addition, the output of the proposed model is the next time-step motion signal.

C. NAR

Qazani et al. [59] employed the non-linear autoregressive (NAR) (with a total of 300,00 inputs and 45 hidden neurons to train and predict the motion signals. It can reach better results compared with feedforward NN because feedforward NN is not able to estimate the non-mapping nonlinear behaviour of the system. It is due to the absence of feedback signals inside the feedforward NN. Then, the NAR model can easily predict the motion signals in a one-dimensional time series. While NAR is a simple model, the drawback is its inaccuracy in predictions. Because of the highly nonlinear behaviour pertaining to the driving motion signal, a nonlinear method is able to produce better prediction results in comparison with those from linear methods. The discretised NAR model for tackling a time series can be formulated as:

$$y(t) = f\big(y(t-1), y(t-2), \cdots, y(t-p)\big) + \epsilon(t) \quad (10)$$

where p is the past values of the series used to predict the upcoming value; f is an unknown function (which can be derived from optimisation [62] of the network weights and neuron bias); and $\epsilon(t)$ is the network error at time t. As NAR is a time-series NN model, the presented sequencing technique in Eq. (10) is defined as feedback delay in the proposed NAR. It should be noted that NAR chooses the next time-step motion signal as the output of the system without specifying to the network as it is a time-series, compatible model.

The scaled conjugate gradient (SCG) is employed to train the model. It is a fast algorithm, as it computes the second-order derivative without calculating the Hessian matrix. Among other training methods SCG has a rapid convergence speed which is based on supervised learning. Also, the SCG training method is supported by GPU to decrease the computational time of the training process. The training procedure of SCG is based on the second-order information of the model. This training method does not have any dependency to the other parameters with usage of the memory. The parameters are updated at each sampling time faster than other training methods.

III. Methodology

While the feedforward NN and NAR are the fast-training models in the prediction of the driving motion signal, they are not accurate methods to predict the motion signal accurately. Two deep learning methods, including ESN, and LSTM are examined in this section to anticipate the driving motion scenarios, as follows:

A. ESN

The ESN is a variant of RNNs proposed in [61], as motivated by the recent neurophysiological studies. Fig. 1 presents a structure of the ESN model, which consists of the input, internal and output units. As a type of RNN, the ESN has a non-trainable recurrent module and a linear readout. The main part of the ESN is the internal unit, which consists of a huge number of neurons with random inter-and self-connections. The internal unit is fixed, while the output connection is changed during the training process using online recursive least square or offline linear regression methods. The internal state $\boldsymbol{\varepsilon}(t)$ at the t^{th} time step within the internal units of ESN is updated, as follows:

$$\varepsilon(t) = \varphi\left(w^{in}x(t) + w^{DR}\varepsilon(t-1) + w^{back}y(t-1)\right) \quad (11)$$

where φ is the activation function (e.g. *tanh* function); $\varepsilon(t) = [\varepsilon_1(t), \cdots, \varepsilon_M(t)]^{\mathrm{T}}$ and $x(t) = [x_1(t), \cdots, x_N(t)]^{\mathrm{T}}$ are the input activation and internal state, respectively. In addition, $y(t-1)$ is the action based on the output neuron from the previous time step; while w^{in}, w^{DR}, and w^{back} are the weight matrices of the input, internal, and feedback connections, respectively. Then, the ESN output, $y(t)$, at the t^{th} time step is :

$$y(t) = \varphi^{out}\left(w^{out}.\begin{bmatrix}\varepsilon(t)\\ y(t-1)\end{bmatrix}\right) \quad (12)$$

where φ^{out} is a linear or sigmoidal function based on the model complexity; w^{out} is a weight matrix of the output connections (which can be regulated via an online or offline training process). The dimension of weight matrices with an M-neuron input, N-neuron reservoirs, and single-neuron output is $w^{in} \in R^{N\times M}$, $w^{DR} \in R^{N\times N}$, $w^{back} \in R^{N\times 1}$ and $w^{out} \in R^{1\times N}$, respectively.

There are 600 neurons with sigmoidal activation functions inside the internal unit. The sparsity of the internal weight matrix is 30%, i.e., 30% of the internal weighting matrix are

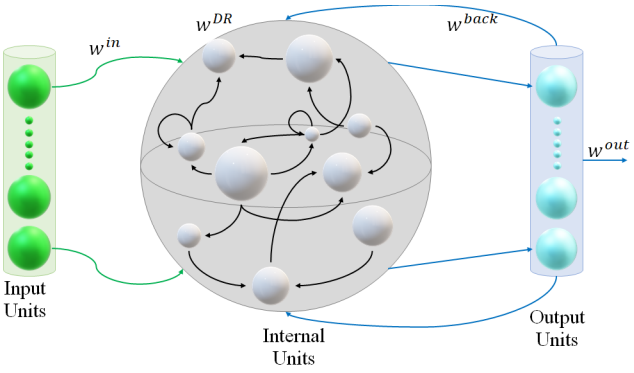

Fig. 1. The schematic structure of the ESN model.

non-zero. These non-zeros elements should be between -0.5 and 0.5. The input weight matrix is randomly selected formed between -0.25 and 0.25, with a prediction order equal to 15. There is no feedback connection between the output and internal neurons because of the predictive nature of the model. The spectral radius of the interval unit is 0.3, and the activation function of the output neuron is sigmoidal, owing to the highly nonlinear behaviour of the model. In order to extract the inputs and output of the ESN during the training process, the motion signals using Eq. (8) time-step delay at each time-step is saved in cell data as an input. Also, the next time-step of the motion signal along the future is saved in another cell to be used as output training data. Then, the trained network is able to anticipate the next time-step ahead of the current time at the current time. It means that the network is able to anticipate the next time-step motion signals by looking at 5 seconds of the motion signal history. Table I shows the hyperparameters of the proposed ESN.

B. LSTM

The RNN is a useful method to predict the driving motion scenarios in view of the dynamical nature of the motion signals. Its internal memory blocks are able to process an arbitrary sequence of inputs using the internal memory in order to facilitate learning of the temporal sequence. Unfortunately, the RNN has some drawbacks, including strict dependence on time lag and the inability to capture the long-term dependencies. Finding the optimal time lag is based on a time-consuming, trial-and-error method. Pre-positioning of the motion simulator is to vary the centre of the end-effector

TABLE I
THE SIMULATION SETUP PARAMETERS FOR THE LSTM NN METHOD

Index (unit)		Value
LSTM	Input Signal	$[u_{i-500} \quad u_{i-499} \quad \cdots \quad u_i]$
	Output Signal	$[u_{i+1} \quad u_{i+2} \quad \cdots \quad u_{i+200}]$
	Time Step	0.01 Second
	Maximum Number of Training Epoch	500
	Minimum Batch Size	120
	Count of Hidden Layers in LSTM Unit	400
ESN	Nr	50
	Rate of leak	0.3
	spectral radius	0.5
	regularization	1×10^{-8}
	washout	100

in order to virtually enlarge the linear displacement limitations of the end-effector. In addition, to exploit the reference time-varying MPC method, the quality of the predicted motion signals is crucial. While it is difficult to train the RNN using 5–10 time lags because of the diminishing gradient problem, Hochreiter and Schmidhuber [60] had demonstrated the effectiveness of the LSTM to address these disadvantages of the RNN. Owing to the unpredictable behaviours of the simulator users, the re-generated motion signals are highly arbitrary. At the same time, the RNN is able to handle the arbitrary sequential data samples because of the feedback connections. The gradient information eliminates or destroys if an extremely long sequence is employed for learning. As an improved version of the RNN, the LSTM model is capable of handling a long sequence of data samples using its memory blocks. Fig. 2.a-b shows a schematic structure of the LTSM model and its layers for the prediction of time-series driving motion scenarios. Based on Fig. 2.a, the proposed LSTM model for predicting the motion scenario is composed of an input generator unit, normalisation unit, sequence input layer, followed by a hidden layer with 400 neurons, regression output, and predicted motion scenarios. The output is generated by a fully connected layer and a regression layer. It should be noted that during the training process of the model, the inputs and outputs are defined to the model based on the history of the motion signal (duration of 5 seconds) and actual future of the motion signal (next time-step) respectively. In addition, m_i and c_i are the hidden state and cell state at time step *t*, as shown in Fig. 2.b, respectively.

The forgotten historical information and updated memory units facilitate learning in the LSTM model. It consists of three gates, namely the input, output, and forget gates. Each gate is composed of the dot product and sigmoid function to protect the gradient information against distortion or elimination, as well as to control the information flow. Fig. 3 presents a memory block depicting f multiplicative gating units to determine the information flow and memory cells to connect and memorise the temporal state.

The model input and output sequences are denoted by $x = (x_1, x_2, \cdots, x_T)$ and $m = (m_1, m_2, \cdots, m_T)$, respectively where T is the prediction period. The memory cell of the j^{th} neuron at time t is denoted by c_t^j, The output of the j^{th} neuron, m_t^j, is :

$$m_t^j = o_t^j \tanh(c_t^j). \tag{13}$$

where o_t^j is the output gate that decides the information to be propagated. The output gate is expressed as:

$$o_t^j = \sigma(W_o x_t + U_o m_{t-1} + V_o c_t)^j. \tag{14}$$

where m_{t-1} and c_t are the vector representations of m_{t-1}^j and c_t^j, respectively; while W_o, U_o, and V_o are the diagonal weight matrices that require online tuning with respect to the minimisation of a loss function. In addition, σ is a standard logistic sigmoid function defined as:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{15}$$

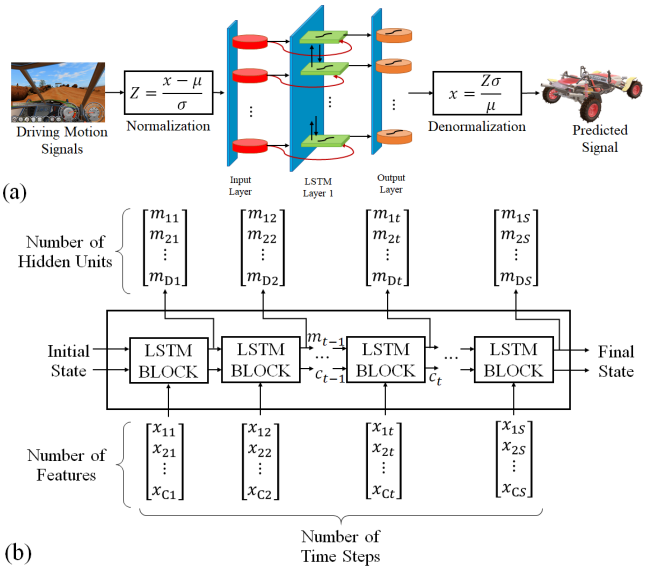

Fig. 2. (a): This diagram of the proposed LSTM NN; (b): The Architecture of the LSTM layer.

The focus of the memory cell is a recurrent constant error carousel (CEC) unit, which is activated to generate the cell state. The CEC helps to eliminate the error by opening and closing the multiplicative gates in the LSTM model. The memory cell, c_t^j, should be updated at each time step by elimination of the current memory cell and addition of the new memory value, $\tilde{c}_t^j$, as follows:

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j. \quad (16)$$

where the new memory value is :

$$\tilde{c}_t^j = \tanh(W_c x_t + U_c m_{t-1})^j. \quad (17)$$

A forget gate is used to prohibit the internal cell values from increasing without limit while continuing the time series mechanism (instead of segmenting). Then, the outdated information flow resets, and the CEC weight is substituted with the multiplicative forget gate activation. After updating the memory cell based on the new memory value, the forget gate, f_t^j, is computed as:

$$f_t^j = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right)^j. \quad (18)$$

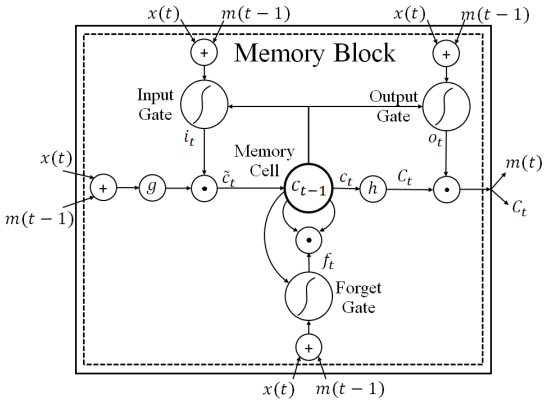

Fig. 3. A memory block depicting the data flow at time step *t*.

where W_f, U_f, and V_f are the diagonal weight matrices. The same methodology is adopted in the input gate, which determines the reserved new features as follows:

$$i_t^j = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1})^j. \quad (19)$$

where W_i, U_i, and V_i are the diagonal weight matrices. It should be noted that the value of the three gates is between 0 and 1. The LSTM output is formulated as:

$$y = g(W_d h_t + b_d). \quad (20)$$

where g is a centred logistic sigmoid function within the range $[-2,2]$, i.e.,

$$g(x) = \frac{4}{1+e^{-x}} - 2. \quad (21)$$

Training of the LSTM is based on a modified real-time recurrent learning (RTRL) and a truncated backpropagation through time (BPTT) along with the gradient descent optimisation method. The loss function is defined as the sum of square errors. The memory cell shortens the errors by exploiting the linear CEC of the memory cell. Inside the CEC, the error recedes and is discharged from the cell in a degraded exponential manner. This is the main capability of the LSTM in dealing with a long prediction horizon, as compared with the RNN. The training process of the LSTM continuously updates the weight matrices in each layer, $W_t = \langle W_o, U_o, V_o, W_c, U_c, W_f, U_f, W_i, U_i, W_d, b_d \rangle$, with respect to minimising the loss function. Table I shows the LSTM parameters used in this study, including the epoch size, batch size, numbers of layers and neurons. The inputs and outputs of the network during the training process have been shown in

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

Item	Feedforward NN		NAR		ESN		LSTM	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MSE	5.6479×10^{-4}	1.4×10^{-3}	1.1×10^{-3}	1.1×10^{-3}	**3.4397×10^{-4}**	**1.4593×10^{-6}**	3.5590×10^{-4}	3.2774×10^{-4}
RMSE	2.38×10^{-2}	3.8×10^{-2}	3.26×10^{-2}	3.26×10^{-2}	**5.8649×10^{-3}**	**1.2080×10^{-3}**	1.8865×10^{-2}	1.8104×10^{-2}
NMSE	2.858×10^{-1}	1.216×10^{-1}	2.704×10^{-1}	1.704×10^{-2}	**1.1327×10^{-1}**	**10521×10^{-3}**	2.2666×10^{-1}	5.5772×10^{-2}
Mean	**-8.2732×10^{-5}**	-1.19×10^{-2}	1.7941×10^{-4}	1.46×10^{-2}	-2.0109×10^{-3}	**1.09167×10^{-4}**	-2.8729×10^{-3}	-4.1207×10^{-3}
STD	2.38×10^{-2}	3.61×10^{-2}	3.26×10^{-2}	2.91×10^{-2}	**5.8651×10^{-3}**	**1.2040×10^{-3}**	1.8646×10^{-2}	1.7641×10^{-2}
R^2	**1**	9.999×10^{-1}	9.999×10^{-1}	9.997×10^{-1}	**1**	**1**	**1**	**1**

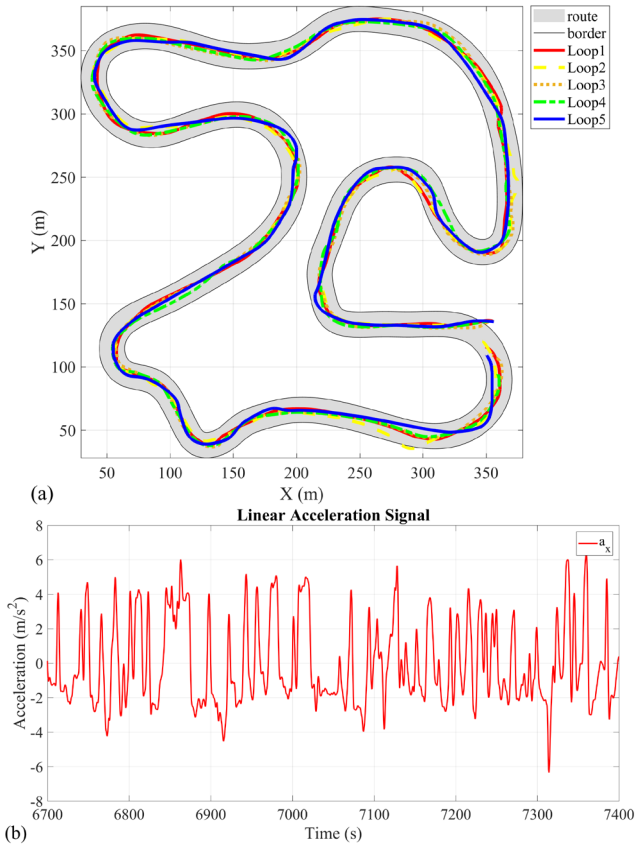

Fig. 4. (a) The training data for duration of five loops along the road map; (b) translational testing data of driving motion scenario along *x*-axis.

the first and second rows of Table I. It is shown that the input and output of the LSTM during the training process are as same as the feedforward NN and ESN, which is explained before. The trained LASTM is able to predict the next motion signal by looking at 5 seconds of the motion.

IV. Results and Discussions

The ESN and LSTM for the prediction of time-series driving motion scenarios are modelled using MATLAB. Our previous method using the feedforward NN [63] and NAR [59] are used for performance comparison. A total of 754,923 data samples are recorded via a six-lap drive-in road track using Rigs of Rods (RoR) simulation software (version 0.39.5) with its soft-body physics engine. The motion scenarios are composed of different manoeuvres such as parking, braking, sudden motion and sharp turning. Fig. 4.a shows the road track and the position of the vehicle in 5 laps. Also, Fig. 4.b presents the linear acceleration signal along the *x*-axis for the last lap of the driving, which is a candidate as a test data of the investigated methods. The four investigated models are used to estimate the driving motion signals along the *x*-axis. The same method can be employed to forecast the motion signals along both the *y*-axis and *z*-axis. It should be noted that the training, validation, and test processes contain 90%, 5%, and 5% of the data samples, respectively.

The error is the difference between the target and the obtained linear acceleration signal of the motion simulator platform. The mean of the error during the training and testing process of the LSTM is -2.8729×10^{-3} and -4.1207×10^{-3} m/s^2, respectively. Also, the standard deviation of the error during the training and testing process of the LSTM is 1.8646×10^{-2} and 1.7641×10^{-2} m/s^2, respectively. It should be noted that the shape of the error profiles of the feedforward NN, NAR, and LSTM are symmetric, while the shape of the error profiles of the ESN is a bit skewed right, respectively. The R-square between the actual and predicted linear acceleration signal of the autonomous vehicle [64] during the testing process are 1, 1, 0.9997 and 0.9999 using LSTM, ESN, NAR, and feedforward NN, respectively. The higher R-square of the proposed LSTM proves the efficiency of the model handling the prediction of the linear acceleration signal compared with other investigated methods.

Table II represents the root means square error (RMSE), mean square error (MSE), normalised root mean square error (NRMSE), rank correlation (RC), mean error (ME), and standard deviation (STD) of four investigated methods during the training and testing process.

The ESN method yields better results in the estimation of the driving motion signals with low RMSE and high R-square scores. While training of the ESN method is not time-consuming, it has an excellent performance in achieving a perfect R-square metric in the prediction of the motion signals.

V. Conclusion

In utilising the simulator for vehicle dynamics evaluation, it is not possible to generate the motion signals of a real vehicle because of the physical and dynamical limitations of the platform. As a result, an MCA is used to re-generate the motion signals and keep the motion sensation of the simulator as close to that of a real vehicle as possible while satisfying the workspace boundary constraints. The effectiveness of the MCA, either filter-based (traditional) or predictive-based (advanced), can be reinforced using accurate estimation of the driving motion signals. This can be achieved through the pre-positioning technique for filter-based MCA models or time-varying reference for predictive-based MCA models. In the literature, researchers mainly use the averages of re-generated motion signals from various operators as the estimate motion signals. While a symmetric way to forecast the motion signals using the feedforward NN and NAR have shown successes, the methods are not scalable to using large data sets. In this study, the ESN, and LSTM deep learning models, have been developed to estimate the motion signals, yielding higher accuracy rates in comparison with the feedforward NN and NAR. It is due to the structure of the ESN and LSTM, as they consist of many layers to handle the complexity of the data. Evaluated with comprehensive simulation studies in MATLAB, the ESN has produced the best results, with a perfect R-square score. Nevertheless, a low computational load is required for ESN training. As a future study, highly advanced machine learning methods [65-71] with consideration of physiological data [22, 72-75] will consider for prediction of the motion signal.

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] 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.

[3] N. Mohajer, S. Nahavandi, H. Abdi, and Z. Najdovski, "Enhancing passenger comfort in autonomous vehicles through vehicle handling analysis and optimization," *IEEE Intelligent Transportation Systems Magazine,* 2020.

[4] N. Mohajer, H. Asadi, S. Nahavandi, and C. P. Lim, "Evaluation of the path tracking performance of autonomous vehicles using the universal motion simulator," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018: IEEE, pp. 2115-2121.

[5] S. Pedrammehr, H. Asadi, and S. Nahavandi, "A study on vibrations of hexarot-based high-G centrifugal simulators," *Robotica,* vol. 38, no. 2, pp. 299-316, 2020.

[6] 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.

[7] M. R. C. Qazani *et al.*, "Kinematic analysis and workspace determination of hexarot-a novel 6-DOF parallel manipulator with a rotation-symmetric arm system," *Robotica,* vol. 33, no. 8, pp. 1686-1703, 2015.

[8] M. R. C. Qazani, S. Pedrammehr, and M. J. Nategh, "An investigation on the motion error of machine tools' hexapod table," *International Journal of Precision Engineering and Manufacturing,* vol. 19, no. 4, pp. 463-471, 2018.

[9] M. R. C. Qazani, V. Mohammadi, H. Asadi, S. Mohamed, and S. Nahavandi, "Development of Gantry-Tau-3R Mechanism Using a Neuro PID Controller," in *ACRA 2019: Proceedings of the Australasian Conference on Robotics and Automation*, 2019: [Australian Robotics & Automation Association], pp. 1-8.

[10] M. R. C. Qazani, H. Asadi, S. Mohamed, S. Nahavandi, J. Winter, and K. Rosario, "A Real-Time Motion Control Tracking Mechanism for Satellite Tracking Antenna Using Serial Robot," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021: IEEE, pp. 1049-1055.

[11] S. Pedrammehr, M. R. C. Qazani, H. Asadi, and S. Nahavandi, "Control system development of a hexarot-based high-G centrifugal simulator," in *2019 IEEE International Conference on Industrial Technology (ICIT)*, 2019: IEEE, pp. 78-83.

[12] S. Pedrammehr, M. R. C. Qazani, H. Asadi, and S. Nahavandi, "Kinematic manipulability analysis of hexarot simulators," in *2019 IEEE International Conference on Industrial Technology (ICIT)*, 2019: IEEE, pp. 133-138.

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

[14] S. Pedrammehr, H. Asadi, and S. Nahavandi, "The forced vibration analysis of hexarot parallel mechanisms," in *2019 IEEE International Conference on Industrial Technology (ICIT)*, 2019: IEEE, pp. 199-204.

[15] S. Pedrammehr, M. R. C. Qazani, H. Asadi, and S. Nahavandi, "Manipulability Analysis of Gantry-Tau parallel manipulator," in *ACRA 2019: Proceedings of the Australasian Conference on Robotics and Automation*, 2019: [Australian Robotics & Automation Association], pp. 1-7.

[16] M. R. C. Qazani, H. Asadi, S. Pedrammehr, and S. Nahavandi, "Performance analysis and dexterity monitoring of hexapod-based simulator," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018: IEEE, pp. 226-231.

[17] M. R. C. Qazani, S. Pedrammehr, and M. J. Nategh, "A study on motion of machine tools' hexapod table on freeform surfaces with circular interpolation," *The International Journal of Advanced Manufacturing Technology,* vol. 75, no. 9-12, pp. 1763-1771, 2014.

[18] M. R. Chalak Qazani, S. Pedrammehr, A. Rahmani, M. Shahryari, A. Khani Sheykh Rajab, and M. M. Ettefagh, "An experimental study on motion error of hexarot parallel manipulator," *The International Journal of Advanced Manufacturing Technology,* vol. 72, no. 9, pp. 1361-1376, 2014.

[19] S. Pedrammehr, M. R. C. Qazani, and S. Nahavandi, "A novel axis symmetric parallel mechanism with coaxial actuated arms," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018: IEEE, pp. 476-480.

[20] M. R. Chalak Qazani, S. Pedrammehr, H. Abdi, and S. Nahavandi, "Performance evaluation and calibration of gantry-tau parallel mechanism," *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering,* vol. 44, no. 4, pp. 1013-1027, 2020.

[21] M. Chalak Qazani, "Modelling and simulation of a motion cueing algorithm using prediction and computational intelligence techniques," Deakin University, 2020.

[22] H. Asadi, S. Mohamed, K. Nelson, S. Nahavandi, and D. R. Zadeh, "Human perception-based washout filtering using genetic algorithm," in *International Conference on Neural Information Processing*, 2015: Springer, pp. 401-411.

[23] H. Asadi *et al.*, "A particle swarm optimization-based washout filter for improving simulator motion fidelity," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016: IEEE, pp. 001963-001968.

[24] 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.

[25] 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.

[26] M. R. C. Qazani, H. Asadi, T. Bellmann, S. Perdrammehr, S. Mohamed, and S. Nahavandi, "A new fuzzy logic based adaptive motion cueing algorithm using parallel simulation-based motion platform," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2020: IEEE, pp. 1-8.

[27] H. Asadi, C. P. Lim, S. Mohamed, D. Nahavandi, and S. Nahavandi, "Increasing motion fidelity in driving simulators using a fuzzy-based washout filter," *IEEE Transactions on Intelligent Vehicles,* vol. 4, no. 2, pp. 298-308, 2019.

[28] H. Asadi, T. Bellmann, S. Mohamed, C. P. Lim, A. Khosravi, and S. Nahavandi, "Adaptive Motion Cueing Algorithm using Optimized Fuzzy Control System for Motion Simulators," *IEEE Transactions on Intelligent Vehicles,* 2022.

[29] H. Asadi, S. Mohamed, and S. Nahavandi, "Incorporating human perception with the motion washout filter using fuzzy logic control," *IEEE/ASME Transactions on Mechatronics,* vol. 20, no. 6, pp. 3276-3284, 2015.

[30] H. Asadi, S. Mohamed, K. Nelson, S. Nahavandi, and M. Oladazimi, "An optimal washout filter based on genetic algorithm compensators for improving simulator driver perception," in *DSC 2015: Proceedings of the Driving Simulation Conference & Exhibition*, 2015: Max Planck Institute for the Advancement of Science, pp. 1-10.

[31] 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.

[32] 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.

[33] H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "A linear quadratic optimal motion cueing algorithm based on human perception," in *ACRA 2014: Proceedings of Australasian Conference on Robotics and Automation*, 2014: Australian Robotics and Automation Association, pp. 1-9.

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

[35] H. Asadi *et al.*, "A model predictive control-based motion cueing algorithm using an optimized nonlinear scaling for driving simulators," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019: IEEE, pp. 1245-1250.

[36] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A model predictive control-based motion cueing algorithm with consideration of joints' limitations for hexapod motion platform," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019: IEEE, pp. 708-713.

[37] 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.

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

[39] 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.

[40] M. R. C. Qazani *et al.*, "An MPC-based Motion Cueing Algorithm Using Washout Speed and Grey Wolf Optimizer," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021: IEEE, pp. 1627-1633.

[41] M. R. C. Qazani *et al.*, "Whale Optimization Algorithm for Weight Tuning of a Model Predictive Control-Based Motion Cueing Algorithm," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021: IEEE, pp. 1042-1048.

[42] A. Koohestani *et al.*, "A knowledge discovery in motion sickness: a comprehensive literature review," *IEEE access,* vol. 7, pp. 85755-85770, 2019.

[43] D. E. Angelaki and K. E. Cullen, "Vestibular system: the many facets of a multimodal sense," *Annu. Rev. Neurosci.,* vol. 31, pp. 125-150, 2008.

[44] 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.

[45] A. Mohammadi, S. Asadi, K. Nelson, and S. Nahavandi, "Future reference prediction in model predictive control based driving simulators," in *Australasian conference on robotics and automation (ACRA2016)*, 2016.

[46] M. Han, J. Xi, S. Xu, and F.-L. Yin, "Prediction of chaotic time series based on the recurrent predictor neural network," *IEEE transactions on signal processing,* vol. 52, no. 12, pp. 3409-3416, 2004.

[47] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017: IEEE, pp. 353-359.

[48] D. Lee, Y. P. Kwon, S. McMains, and J. K. Hedrick, "Convolution neural network-based lane change intention prediction of surrounding vehicles for acc," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017: IEEE, pp. 1-6.

[49] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "High-fidelity hexarot simulation-based motion platform using fuzzy incremental controller and model predictive control-based motion cueing algorithm," *IEEE Systems Journal,* vol. 14, no. 4, pp. 5073-5083, 2019.

[50] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018: IEEE, pp. 1441-1446.

[51] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018: IEEE, pp. 1179-1184.

[52] Y. Hu, W. Zhan, and M. Tomizuka, "Probabilistic prediction of vehicle semantic intention and motion," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018: IEEE, pp. 307-313.

[53] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Conference on Robot Learning*, 2018: PMLR, pp. 947-956.

[54] T. Zhao *et al.*, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12126-12134.

[55] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems,* 2020.

[56] M. R. C. Qazani, H. Asadi, S. Mohamed, and S. Nahavandi, "Prepositioning of a land vehicle simulation-based motion platform using fuzzy logic and neural network," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 10, pp. 10446-10456, 2020.

[57] N. M. Vural, S. Ergut, and S. S. Kozat, "An efficient and effective second-order training algorithm for lstm-based adaptive learning," *IEEE Transactions on Signal Processing,* 2021.

[58] B. R. Kiran *et al.*, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems,* 2021.

[59] M. R. C. Qazani, H. Asadi, S. Mohamed, C. P. Lim, and S. Nahavandi, "Prediction of Motion Simulator Signals Using Time-Series Neural Networks," *IEEE Transactions on Aerospace and Electronic Systems,* 2021, doi: Accepted Recently.

[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[61] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report,* vol. 148, no. 34, p. 13, 2001.

[62] M. R. C. Qazani, S. M. J. Jalali, H. Asadi, and S. Nahavandi, "Optimising Control and Prediction Horizons of a Model Predictive Control-Based Motion Cueing Algorithm Using Butterfly Optimization Algorithm," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020: IEEE, pp. 1-8.

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

[64] T. Tettamanti, A. Mohammadi, H. Asadi, and I. Varga, "A two-level urban traffic control for autonomous vehicles to improve network-wide performance," *Transportation research procedia,* vol. 27, pp. 913-920, 2017.

[65] M. R. C. Qazani, H. Asadi, M. Al-Ashmori, S. Mohamed, C. P. Lim, and S. Nahavandi, "Time Series Prediction of Driving Motion Scenarios Using Fuzzy Neural Networks:* Motion Signal Prediction Using FNNs," in *2021 IEEE International Conference on Mechatronics (ICM)*, 2021: IEEE, pp. 1-6.

[66] 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.

[67] M. R. C. Qazani, V. Pourmostaghimi, M. Moayyedian, and S. Pedrammehr, "Estimation of tool–chip contact length using optimized machine learning in orthogonal cutting," *Engineering Applications of Artificial Intelligence,* vol. 114, p. 105118, 2022.

[68] M. R. C. Qazani, H. Parvaz, and S. Pedrammehr, "Optimization of Fixture Locating Layout Design Using Comprehensive Optimised Machine Learning," 2022.

[69] F. Nazari, N. Mohajer, D. Nahavandi, A. Khosravi, and S. Nahavandi, "Comparison Study of Inertial Sensor Signal Combination for Human Activity Recognition based on Convolutional Neural Networks," *arXiv preprint arXiv:2206.04480,* 2022.

[70] F. Nazari, Nahavandi, D., Mohajer, N., & Khosravi, A, "Comparison of Deep Learning Techniques on Human Activity Recognition using Ankle Inertial Signals," presented at the International Conference on Systems, Man, and Cybernetics (SMC), 2022.

[71] F. Nazari, Mohajer, N., Nahavandi, D., & Khosravi, A, "Comparison of gait phase detection using traditional machine learning and deep learning techniques," 2022.

[72] M. Oladazimi, F. Molaei-Vaneghi, M. Safari, H. Asadi, and S. Aghay Kaboli, "A review for feature extraction of EMG signal processing," in *4th International Conference on Computer and Automation Engineering (ICCAE 2012)*, 2012: ASME Press, pp. 85-94.

[73] M. R. C. Qazani *et al.*, "A Fast and Reliable Approach for Driving Style Customization in Autonomous Vehicles," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021: IEEE, pp. 1869-1875.

[74] F. Nazari, N. Mohajer, D. Nahavandi, A. Khosravi, and S. Nahavandi, "Critical Review of Exoskeleton Technology: State of the art and development of physical and cognitive human-robot interface," *arXiv preprint arXiv:2111.12860,* 2021.

[75] F. Nazari, D. Nahavandi, N. Mohajer, and A. Khosravi, "Human Activity Recognition from Knee Angle Using Machine Learning Techniques," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021: IEEE, pp. 295-300.