ResearchOnline@JCU

JAMES COOK UNIVERSITY AUSTRALIA

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

Moayyedian, Mehdi, Chalak Qazani, Mohamadreza, and Pourmostaghimi, Vahid (2022) ***Optimized injection-molding process for thin-walled polypropylene part using genetic programming and interior point solver.*** **International Journal of Advanced Manufacturing Technology, 124 pp. 297-313.**

Access to this file is available from:

https://researchonline.jcu.edu.au/86734/

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

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

https://doi.org/10.1007/s00170%2D022%2D10551%2D2

A CNN-LSTM Based Model to Predict Trajectory of Human-Driven Vehicle

Shehab Alsanwy, Houshyar Asadi, Mohammad Reza Chalak Qazani, Shady Mohamed and Saeid Nahavandi

***Abstract*—Vehicle trajectory prediction is essential in ensuring the safe and efficient operation of advanced driver assistance systems (ADAS) and autonomous vehicles (AVs), as it enables highly efficient collision avoidance, path planning, and traffic control. However, existing models for vehicle trajectory prediction predominantly focus on limited driving scenarios, resulting in limited applicability. To address this limitation, we present a novel vehicle trajectory prediction approach that employs a Convolutional Long Short-Term Memory (CNN-LSTM) model, incorporating simulated environments and vehicle dynamic time series data, including longitudinal, vertical, and latitudinal position and acceleration. Our approach is distinguished by its ability to handle diverse urban driving scenarios, such as highways, roundabouts, intersections, and turns, which enhances its applicability and generalizability. We experimented and collected vehicle data from 17 drivers using a stationary driving simulator and the Euro Truck Simulator software. For the model implementation and validation, we utilized Python 3.9 and Google Colab, as well as the Scikit-learn library for Deep learning algorithms. The proposed CNN-LSTM model leverages a convolutional layer to learn local patterns and an LSTM layer to capture long-term temporal dependencies, improving performance in predicting vehicle trajectories. The experimental results demonstrate that the CNN-LSTM model provides more accurate predictions for longitudinal and lateral positions compared to traditional vehicle trajectory prediction methods that employ LSTM and Recurrent Neural Network (RNN). This research contributes to developing robust and reliable vehicle trajectory prediction systems vital for ADAS and AVs' safe and efficient operation. The proposed approach broadens the applicability of trajectory prediction models, enabling better-informed decision-making in various driving conditions and ultimately improving road safety and efficiency in the rapidly evolving field of autonomous transportation.**

I. Introduction

Vehicle trajectory prediction is critical in developing advanced driver assistance systems (ADAS) and autonomous vehicles (AVs), as it ensures safe and efficient transportation. Accurate prediction of human-driven vehicle trajectories is vital for various applications, such as collision avoidance, path planning, and traffic control [1]. The rapid evolution of sensing technologies, including GPS, LIDAR, and cameras, has led to a surge in the availability of high-quality data related to dynamic vehicle behaviour. Consequently, researchers have explored deep learning techniques to exploit this wealth of data for accurate vehicle trajectory prediction [2], [3].

In recent years, deep learning has demonstrated significant potential for vehicle trajectory prediction [4], [5]. Deep learning models, including Long Short-Term (LSTM) models, have effectively captured intricate patterns and trends in data, resulting in accurate predictions of a vehicle's future trajectory [6] and motion [7], [8]. A key advantage of deep learning-based models is their ability to handle time series data, which is essential for capturing the sequential nature of vehicle positions [9]. Several studies have reported the effectiveness of deep learning models for vehicle trajectory prediction. Benterki et al. [5] proposed a model employing LSTM and Gated Recurrent Network (GRU) to predict vehicle trajectories using historical GPS data and vehicle dynamics. The model exhibited strong performance on the publicly available Next-Generation Simulation (NGSIM) dataset, which comprises real-world traffic data collected from various US highways as part of the NGSIM program by the Federal Highway Administration (FHWA) [10]. Similarly, Yang et al. [11] developed a model using LSTM-RNNs for long-term trajectory prediction based on vehicle dynamics and historical GPS data, reporting high accuracy with the NGSIM dataset.

There has been some research on trajectory prediction that relied on the NGSIM dataset to predict vehicle Trajectory [6], [12]–[14]. However, these models' reliance on the NGSIM dataset restricts their predictive capabilities to only highway scenarios, excluding the complexity of roundabouts and intersections of urban environments. In addition, other studies have focused on specific driving scenarios using their own datasets. Huang et al. [15] proposed an approach for predicting driver trajectories in urban intersection scenarios. However, its applicability is primarily confined to intersections, excluding contexts such as highways and roundabouts. Considering these limitations, developing trajectory prediction models that can generalize across various driving conditions and scenarios, such as highways, roundabouts, intersections, and turns, is essential, thereby contributing to a more robust vehicle trajectory. To address these limitations, this study proposes an ego-vehicle trajectory prediction approach that utilizes simulated environments and time series data of vehicle positions. To do this, we conducted an experiment with 16 participants using a driving simulator. There are different types of driving simulators, equipped with advanced motion cueing algorithms, to create a realistic driving experience for

*Research was supported by the Australian Research Council (ARC) (Project ID: DE210101623), as well as by the Institute for Intelligent Systems Research and Innovation, Deakin University.

Shehab Alsanwy, Houshyar Asadi, and Shady Mohamed are with the Institute of Intelligent Systems Research and Innovation (IISRI) at Deakin University Geelong, VIC, 3126, Australia. (e-mail: sabdulraqeb, houshyar.asadi, shady.mohamed@deakin.edu.au).

MRC Qazani is with the Department of Computer Science and Information Systems (CSIS) at Murray State University, Murray, KY 42071, USA. (e-mail: m.r.chalakqazani@gmail.com).

Saeid Nahavandi is with Swinburne University of Technology, Hawthorn, VIC 3122, Australia, (e-mail: saeid.nahavandi@ieee.org).

the participants [16]–[20]. In this study, we collected vehicle data for diverse scenarios using a stationary driving simulator and the Euro Truck Simulator software. The proposed method employs a CNN-LSTM model, which has proven effective in various applications, including solar radiance predictions [21] and stock price forecasts [22]. In the CNN-LSTM model, the convolutional layers are designed to learn local patterns and structures in the time series data, such as short-term trends or fluctuations. Subsequently, the extracted features are passed to the LSTM layers, which capture the long-term temporal dependencies in the data, enabling the prediction of future positions as time series data. Furthermore, the dataset used in this study includes left turns, lane changes, and intersections, broadening the model's applicability beyond highway scenarios to encompass a broader range of driving conditions and situations. This approach enhances the generalizability and robustness of the model. It contributes to the ongoing efforts in the field to develop more comprehensive and reliable vehicle trajectory prediction systems.

The organization of this paper is as follows: Section II delves into the research methodology, encompassing data collection Procedure, data preprocessing, and the proposed model for predicting vehicle trajectory. The proposed model's ability to generate accurate predictions is validated in Section III. Finally, Section IV offers concluding observations.

II. Research Materials and Methods

This section presents the data collection methods, the proposed model description, data preprocessing and preparation, the study's performance metrics, and the proposed Convolutional Long Short-Term Memory (CNNLSTM) model implementation.

A. Data Collection

This research employed a Next Level Racing Motion Platform V3 to closely replicate realistic driving conditions, providing participants with a high-quality experience [23]. To ensure consistency across participants and avoid potential motion sickness caused by unpredicted motion, the motion in the platform was disabled during the study [24]. A Logitech G920 Driving Force Racing Wheel was also used to simulate authentic steering and braking [25]. The Eurotruck driving simulator, modified for car usage, was chosen as the driving simulator for this study [26]. Participants were offered a 180-degree view through a triple-screen monitor setup, which allowed for natural head movements and increased immersion within the virtual environment. Moreover, Future studies will consider incorporating a Universal Motion Simulator [27] and a Motion Cueing Algorithm [28]–[34] for more accurate and controlled simulation experiences. The experimental configuration can be observed in Fig 1.

The vehicle dynamic data, including longitudinal, vertical, and latitudinal position and acceleration, were collected through Eurotruck driving simulator software for all participants. Tobii Pro Glasses 2 [35] and an Equivital belt [36] were employed as physiological sensors to collect further data involving participants' physiological responses. However, the primary focus of this study did not center on the participants' physiological data, which will be reserved for future research. This study involved 25 healthy participants aged between 19 and 40, with an average age of 32.7 and a standard deviation



Figure. 1. Experimental Setup

of 8.96. All participants possessed prior driving experience and held full driver's licenses. Engaging participants with pre-existing driving experience was crucial, as it ensured their familiarity with fundamental driving mechanics. Before the experiment, participants filled out questionnaires regarding their driving experience and demographics. Additionally, they were assessed for motion sickness using the Simulator Sickness Questionnaire (SSQ) [37]–[39]. Participants prone to motion sickness or those with severe vision problems were excluded from the study. The research received approval from Deakin University's Human Ethics Advisory Groups (HEAGs). During the experiment, drivers were instructed to navigate from one point to another using a provided map. The chosen map featured various road conditions, including highways, roundabouts, intersections, and turns, as depicted in Fig 2.

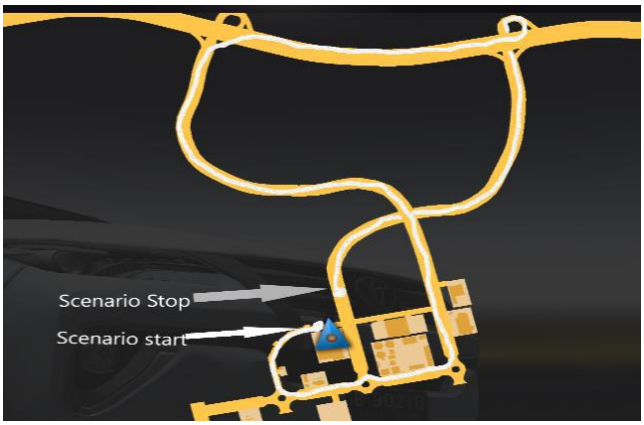


Figure. 2. Driving Map

B. Model Description and Hyperparameters

The proposed model is based on CNN-LSTM. The basic structure of a CNN-LSTM cell is similar to that of a traditional LSTM cell. However, it includes convolutional and pooling layers for feature extraction. The convolutional layer allows the CNN-LSTM to learn spatial dependencies in the input data. In contrast, the LSTM layer allows it to learn temporal dependencies. CNN-LSTMs have been shown to perform exceptionally well on various problems [21]. Each CNN-LSTM unit contains an input gate, an output gate, and a forget gate, similar to a regular LSTM. The input gate activation vector and the memory cell state are calculated as follows:

$$i_t = \sigma(W_i * x_t + U_i * h_(t-1) + b_i) \quad (1)$$

$$C\sim t = \sigma(W_c * x_t + U_c * h_(t-1) + b_c) \quad (2)$$

Here W_i, U_i, b_i, W_c, U_c, and b_c are the weights and biases for the input gate and memory cell state, respectively.

The forget gate activation f_t is calculated using:

$$f_t = \sigma(W_f * x_t + U_f * h_(t-1) + b_f) \quad (3)$$

which controls how much of the previous cell state is retained. The current memory cell state C_t is then updated as follows:

$$C_t = f_t * C_(t-1) + i_t * C\sim t \quad (4)$$

where $C_(t-1)$ is the previous memory cell state. Finally, the output gate O_t is calculated as follows:

$$O_t = \sigma(W_o * x_t + U_o * h_(t-1) + b_o) \quad (5)$$

The output gate determines how much of the current cell state is used to produce the output. The final output at each time step h_t is computed as:

$$h_t = O_t * \tanh(C_t) \quad (6)$$

Our approach will train the CNN-LSTM based on a historical data sequence and use a sliding window of 500 timesteps to predict future values. The manually optimized hyperparameters for the CNN-LSTM model are summarized in Table I. The model consists of 32 filters in the convolutional layer with a kernel size of 3 and a Tanh activation function. The LSTM layer has 64 hidden units and employs a Tanh activation function. Other settings include a sequence length of 500, a batch size of 32, a learning rate of 0.001, the Adam optimizer, Mean Absolute Error as the loss function, and 50 epochs for training.

C. Data Preprocessing and Preparation

In this study, a total number of 25 participants was initially recruited. However, upon further examination, eight individuals were identified to have visual impairments and reported experiencing motion sickness. Motion sickness frequently occurs when there is a discrepancy between visual input and the signals detected by the inner ear's vestibular system [40]–[45]. Consequently, these participants were excluded from the study, resulting in a final sample size of 17 subjects. The collected data from multiple subjects were preprocessed before being used in the model. The data preprocessing steps included data alignment, formatting, and standardization. During data Alignment, data from each subject were aligned in time to ensure that corresponding data points from different subjects could be compared and combined. This step is crucial to ensure that the data is comparable and useful for analysis. For this step, we used X, Y, and Z positions within the map to pinpoint an exact location within the map for all subjects. The dataset comprised 50,949 samples, partitioned into distinct subsets for various purposes. Specifically, 70% of the data (35,664 samples) was allocated for training, while 20% (10,190 samples) was reserved for validation. The remaining 10% (5,095 samples) were utilized for model evaluation and performance assessment.

TABLE I
MODEL HYPER-PARAMETERS

Model Hyperparameter	**Value**
Convolutional Layer Filter	32
Convolution Layer kernel size	3
Max pooling layers	3
Convolution Layer Activation Function	Tanh
LSTM Hidden Units	64
LSTM Activation Function	Tanh
Sequence length	500
Batch Size	32
Learning Rate	0.001
Optimizer	Adam
Loss Function	Mean Absolute Error
Epochs	50

1) Data Standardization

The Z Score, also known as the Standard Score, transforms all values of a given attribute into a standardized range. Standardizing variables is a common preprocessing step in statistical analysis and machine learning and deep learning models, as it helps reduce the impact of differences in scale and range, leading to better model accuracy and generalization. Formula 7 calculates the standardized value of datapoint (x), referred as Z in the formula. The formula takes the original value (x), subtracts the mean of the distribution (μ), and divides it by the standard deviation (σ).

$$Z = \frac{x-\mu}{\delta} \quad (7)$$

After the data was aligned, formatted, and standardized, it was combined by concatenating the data sets or each subject's data. This step enables the use of data from multiple subjects in the model.

2) Transforming data for supervised learning

The first crucial step after importing data is to ensure they are in the appropriate shape and format for the model to learn from them effectively. In this research, we aim to train a model that can use 500 (25s) of historical data to predict the future signal 5 seconds ahead accurately. We employed 500 historical samples, which form a sliding window, as the number of historical data of the model considers predicting a one-time step into the future. Once the model predicts the future timestep, it shifts the data by one to achieve auto-recursive prediction. It adds the predicted 500-time step to the input, and this process continues for the length of the input data. An example of the first input and output sequence is presented in equations 8 and 9, respectively:

$$\begin{bmatrix} P_{x0} & P_{y0} & P_{z0} & a_{x0} & a_{y0} & a_{z0} & r_{x0} & r_{y0} & r_{z0} & s_1 & b_1 \\ P_{x1} & P_{y1} & P_{z1} & a_{x1} & a_{y1} & a_{z1} & r_{x1} & r_{y1} & r_{z1} & s_2 & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{x499} & P_{y499} & P_{z499} & a_{x499} & a_{y499} & a_{z499} & r_{x499} & r_{y499} & r_{z499} & s_{500} & b_{500} \end{bmatrix} \quad (8)$$

$$[P_{x500} \quad P_{y500}] \quad (9)$$

Here P_x, P_y, P_z and a_x, a_y, a_z denote the position and acceleration signals in the longitudinal, lateral, and vertical directions. After training the model, the testing data are provided for evaluation. The model is then used to predict 100 timesteps in the future, corresponding to a 5-second prediction horizon. Finally, the output data are converted to their original form and compared with the actual data to assess the model's performance.

D. Performance Metrics

The Root Mean Square Error (RMSE) and Pearson's Correlation r quantify the model's performance. RMSE measures the standard deviation of prediction errors,

indicating how closely the data points are clustered around the best-fit line. At the same time, Pearson's r assesses the linear relationship between two variables. The RMSE can be calculated using the equation below:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{m}(y_j - \hat{y}_j)^2}{N}} \quad (10)$$

where j represents the index of the data point, while m represents the total number of data points. represents the actual/real observation at index j in the time series, and represents the predicted observation at index j in the time series. represents the total number of observations in the time series. By utilizing the RMSE metric, we can evaluate the accuracy of prediction models and determine the degree to which their predictions deviate from the actual values [8].

1) Pearson Correlation coefficient (r)

The Pearson Correlation Coefficient (r) reflects the linear relationship between two variables. A positive coefficient suggests that one variable tends to increase or decrease in tandem. Conversely, a negative coefficient implies that as one variable's value rises, the other's value declines [8]. The Pearson Correlation Coefficient (r) can be determined using the equation below:

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 (y_i - \overline{y})^2}} \quad (11)$$

where x_i donates the individual x-variable values within the sample and $\overline{x}$ represents the average of all the x-variable values. While y and $\overline{y}$ refer to individual y-variable values in the sample and the average of all the y-variable values.

III. RESULTS AND DISCUSSION

Before This section presents a comparative analysis of the proposed CNN-LSTM model for predicting vehicle trajectories, benchmarked against other well-established time series and vehicle trajectory prediction algorithms, including LSTM and RNN. Initially, the computational loads for the training of the proposed CNN-LSTM model and the other algorithms are compared., Subsequently, the efficacy of these models in accurately predicting vehicular trajectories is thoroughly assessed by RMSE and r as performance metrics.

A. computational load performance and analysis

The results presented in Table II showcase the training times of the different machine learning models applied to a dataset consisting of vehicle data. The table compares the training times of Long LSTM, RNN, and CNN-LSTM models. The model was built using Python 3.9 and created within Google Colab. This cloud-based computing platform allows users to run Python code in a Jupyter notebook environment, with access to a Tesla K80 GPU with 12 GB of memory.

Table II shows that the LSTM model exhibits a training time of 676.32 seconds, while the RNN model demonstrates a considerably higher training time of 940.44 seconds. The proposed CNN-LSTM model, on the other hand, was trained in 690.11 seconds, which is slightly higher than the training time of the LSTM model. The results indicate that both LSTM and CNN-LSTM models demonstrate significantly lower training times compared to the RNN model. This suggests that these models may be more efficient in capturing the relationships in the vehicle data, leading to faster convergence during the training process. Additionally, the proposed CNN-LSTM model, despite incorporating a convolutional layer, still maintains a training time comparable to that of the LSTM model. This highlights the potential of the proposed CNN-LSTM model for vehicle trajectory prediction while keeping computational costs relatively low.

TABLE II
MODELS TRAINING TIME IN SECONDS

Model	Training Time Vehicle Data (s)
LSTM	676.32
RNN	940.44
CNN-LSTM	690.11

B. Trajectory Prediction performance

The results presented in Table III compare the performance of various deep-learning models for predicting vehicle trajectories using vehicular data. The table compares the Root Mean Square Error (RMSE) and the correlation coefficient (r) for longitudinal and lateral position predictions using LSTM, RNN, and CNN-LSTM models. The values presented in the table are the average values for a 5-second prediction horizon.

Table III illustrates that the CNN-LSTM model outperforms both LSTM and RNN models in terms of RMSE and r for both longitudinal and lateral position predictions. Specifically, the CNN-LSTM model achieves an RMSE of 0.35049 and an r value of 0.89901 for longitudinal position prediction, resulting in an RMSE of 0.0999 and an r value of 0.99342 for lateral position prediction. The LSTM and RNN models demonstrate higher RMSE values and lower r values for both longitudinal and lateral position predictions. Furthermore, the proposed model significantly outperforms previous studies [5] , [46] in terms of prediction RMSE score for a 5-second prediction horizon. However, it should be noted that this model did not consider other vehicles and agents within the road as inputs to the model. Fig 3 and Fig 4 illustrate the prediction of lateral and longitudinal positions of the vehicle for five seconds in the future.

These results indicate that the proposed CNN-LSTM model better predicts vehicle trajectories using vehicular data, including longitudinal, vertical, and lateral position and acceleration signals. The incorporation of convolutional layers in the CNN-LSTM results in improved accuracy for both longitudinal and lateral position predictions. This highlights the potential of the CNN-LSTM model as a robust and effective method for vehicle trajectory prediction in various applications, such as advanced driver assistance systems (ADAS) and autonomous vehicles (AVs).

IV. CONCLUSIONS

This study presents a novel ego-vehicle trajectory prediction approach using a CNN-LSTM model that transcends the limitations of existing studies in predicting vehicle trajectory in highway-specific scenarios. The proposed method demonstrates versatility by encompassing a broader range of urban driving conditions, including driving on highways, roundabouts, intersections, and turns. We experimented and collected vehicle data from 17 drivers using a driving simulator and the Euro Truck Simulator software. After

TABLE III
PREDICTION PERFORMANCE

Category		Vehicle trajectory (positions) prediction using vehicular data			
		Longitudinal Position		Lateral Position	
		RMSE	r	RMSE	r
Deep Learning Models	LSTM	0.421012	0.82007	0.1889	0.99100
	RNN	0.593811	0.67095	0.1929	0.98901
	CNN-LSTM	0.35049	0.89901	0.0999	0.99342

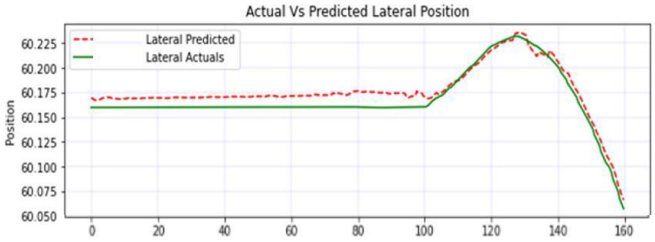


Figure. 3. A comparison between the actual and predicted Lateral position using CNN-LSTM.

implementing and training the proposed CNN-LSTM model

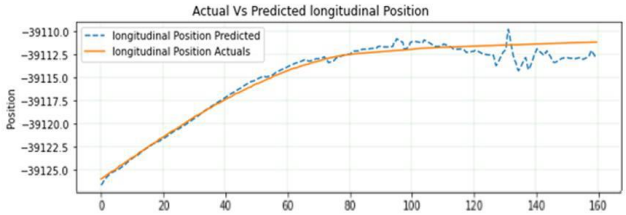


Figure. 4. A comparison between the actual and predicted longitudinal position using CNN-LSTM.

using the collected vehicle data, the experimental results indicate that the proposed model outperforms both LSTM and RNN models in terms of accuracy for longitudinal and lateral position predictions. It should be noted that the computational load of the proposed CNN-LSTM model for training is 690.11 seconds. By addressing the current research limitations, this study contributes to developing a comprehensive and reliable vehicle trajectory prediction model for advanced driver assistance systems and autonomous vehicles.

Future research endeavors should focus on exploring methods to generalize trajectory prediction models across a diverse array of driving conditions and scenarios. Additionally, integrating other data sources, including physiological and environmental factors, could further enhance the accuracy and robustness of the trajectory prediction model. This would ultimately lead to the developing of sophisticated and context-aware models that can adapt to a wide variety of driving environments and situations, thereby ensuring the safe and efficient operation of advanced driver assistance systems and autonomous vehicles.

Furthermore, the utilization of more advanced motion simulators, such as Universal Motion Simulator (UMS), with state-of-the-art motion cueing algorithms [47]–[64] be considered for future studies, as this approach could substantially improve the realism of simulated driving scenarios, thereby enhancing the testing and validation of these trajectory prediction models under diverse, real-worldlike driving conditions. In addition, other methods [7], [65]– [68] for trajectory prediction can be used to explore their effectiveness compared to the proposed method in this study.

V. Acknowledgment

The authors would like to acknowledge the support of Australian Research Council (Project ID: DE210101623) and IISRI, Deakin University for providing required equipment and Labs to successfully conduct this project.

References

[1]. S. Lefèvre, D. Vasquez, and C. Laugier, "Lefèvre2014_Article_ASurveyOnMotionPredictionAndRi.pdf," pp. 1–14, 2014.

[2]. A. Zyner, S. Worrall, and E. Nebot, "Naturalistic Driver Intention and Path Prediction Using Recurrent Neural Networks," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 4, pp. 1584–1594, 2020, doi: 10.1109/TITS.2019.2913166.

[3]. A. Díaz-Álvarez, M. Clavijo, F. Jiménez, and F. Serradilla, "Inferring the Driver's Lane Change Intention through LiDARBased Environment Analysis Using Convolutional Neural Networks," Sensors, vol. 21, no. 2, p. 475, Jan. 2021, doi: 10.3390/s21020475.

[4]. A. Quintanar, D. Fernández-Llorca, I. Parra, R. Izquierdo, and M. A. Sotelo, "Predicting Vehicles Trajectories in Urban Scenarios with Transformer Networks and Augmented Information," no. Iv, 2021, [Online]. Available: http://arxiv.org/abs/2106.00559.

[5]. A. Benterki, V. Judalet, M. Choubeila, and M. Boukhnifer, "Long-Term Prediction of Vehicle Trajectory Using Recurrent Neural Networks," IECON Proc. (Industrial Electron. Conf., vol. 2019-Octob, pp. 3817–3822, 2019, doi: 10.1109/IECON.2019.8927604.

[6]. F. Altche and A. De La Fortelle, "An LSTM network for highway trajectory prediction," IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC, vol. 2018-March, pp. 353–359, 2018, doi: 10.1109/ITSC.2017.8317913.

[7]. M. R. C. Qazani et al., "A Prediction of Time Series Driving Motion Scenarios Using LSTM and ESN," in 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2022, pp. 1592–1599, doi: 10.1109/SMC53654.2022.9945220.

[8]. S. Alsanwy et al., "Prediction of Vehicle Motion Signals for Motion Simulators Using Long Short-Term Memory Networks," Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., vol. 2022- Octob, pp. 34–39, 2022, doi: 10.1109/SMC53654.2022.9945209.

[9]. J. Gao, Y. L. Murphey, and H. Zhu, "Multivariate time series prediction of lane changing behavior using deep neural network," Appl. Intell., vol. 48, no. 10, pp. 3523–3537, 2018, doi: 10.1007/s10489-018-1163-9.

[10]. J. C. John Halkias, "No Title," Interstate 80 Freeway Dataset, 2006. https://www.fhwa.dot.gov/publications/research/operations/06137 /.

[11]. Z. Yang, Z. Gao, F. Gao, C. Shi, L. He, and S. Gu, "Intelligent Vehicle Moving Trajectory Prediction Based on Residual Attention Network," World Electr. Veh. J., vol. 13, no. 3, 2022, doi: 10.3390/wevj13030047.

[12]. M. Hasan, A. Solernou, E. Paschalidis, H. Wang, G. Markkula, and R. Romano, "Maneuver-Aware Pooling for Vehicle Trajectory Prediction," 2021, [Online]. Available: http://arxiv.org/abs/2104.14079.

[13]. C. Ju, Z. Wang, C. Long, X. Zhang, and D. E. Chang, "Interaction-aware Kalman Neural Networks for Trajectory Prediction," IEEE Intell. Veh. Symp. Proc., pp. 1793–1800, 2020, doi: 10.1109/IV47402.2020.9304764.

[14]. H. Xiao et al., "UB-LSTM: A Trajectory Prediction Method Combined with Vehicle Behavior Recognition," J. Adv. Transp., vol. 2020, 2020, doi: 10.1155/2020/8859689.

[15]. X. Huang, S. G. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," Proc. - IEEE Int. Conf. Robot. Autom., vol. 2019- May, pp. 9718–9724, 2019, doi: 10.1109/ICRA.2019.8794282.

[16]. 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, pp. 78–83, doi: 10.1109/ICIT.2019.8755117.

[17]. 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, pp. 133–138, doi: 10.1109/ICIT.2019.8755129.

[18]. H. Asadi, S. Mohamed, and S. Nahavandi, "Incorporating Human Perception With the Motion Washout Filter Using Fuzzy Logic Control," IEEE/ASME Trans. Mechatronics, vol. 20, no. 6, pp. 3273284, 2015, doi: 10.1109/TMECH.2015.2405934.

[19]. M. R. Chalak 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 Trans. Syst. Man, Cybern. Syst., vol. PP, pp. 1– 15, 2019, doi: 10.1109/TSMC.2019.2958062.

[20]. [20] 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 Trans. Intell. Veh., vol. 8, no. 1, pp. 390–403, 2023, doi: 10.1109/TIV.2022.3147862.

[21]. S. Mohammad, J. Jalali, and S. Ahmadian, "Automated Deep CNN-LSTM Architecture Design for Solar Irradiance Forecasting," vol. 52, no. 1, pp. 54–65, 2022.

[22]. W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," Complexity, vol. 2020, 2020, doi: 10.1155/2020/6622927.

[23]. Nextlevelracing.com, "No Title," 2021. https://nextlevelracing.com/products/next-level-racing-motionplatform- v3/.

[24]. O. X. Kuiper, J. E. Bos, E. A. Schmidt, C. Diels, and S. Wolter, "Knowing What's Coming: Unpredictable Motion Causes More Motion Sickness.," Hum. Factors, vol. 62, no. 8, pp. 1339–1348, Dec. 2020, doi: 10.1177/0018720819876139.

[25]. Https://www.logitechg.com/, "G920/G29," [Online]. Available: https://www.logitechg.com/en-au/products/driving/driving-forceracing- wheel.html.

[26]. Https://eurotrucksimulator2.com/, "No Title." 2012 - 2023 SCS Software s.r.o., 2022, [Online]. Available: https://eurotrucksimulator2.com/.

[27]. S. R. Alsanwy et al., "The Effects of Different Body Positions on Human Physiological Responses Using Universal Motion Simulator," 2021 IEEE Int. Conf. Syst. Man, Cybern., vol. 6, no. 1, pp. 1209–1214, 2021, doi: 10.1109/SMC52423.2021.9658601.

[28]. M. R. Chalak 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, pp. 1–8, doi: 10.1109/FUZZ48607.2020.9177599.

[29]. 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, pp. 1–8, doi: 10.1109/CEC48606.2020.9185842.

[30]. 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 Trans. Syst. Man, Cybern. Syst., vol. 47, no. 2, pp. 238–254, 2017, doi: 10.1109/TSMC.2016.2523906.

[31]. M. R. Chalak 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 Syst. J., vol. PP, pp. 1–11, 2019, doi: 10.1109/JSYST.2019.2940754.

[32]. 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 Trans. Veh. Technol., vol. 69, no. 10, pp. 10446–10456, 2020, doi: 10.1109/TVT.2020.3006319.

[33]. H. Asadi, S. Mohamed, D. R. Zadeh, and S. Nahavandi, "Optimisation of nonlinear motion cueing algorithm based on genetic algorithm," Veh. Syst. Dyn., vol. 53, no. 4, pp. 526–545, 2015, doi: 10.1080/00423114.2014.1003948.

[34]. M. Al-Ashmori, V. T. Huynh, M. R. C. Qazani, A. Arogbonlo, S. Al-Serri, and S. Alsanwy, "Comparison of Optimisation Solvers for Model Predictive Control-Based Motion Cueing Algorithm," Australas. Conf. Robot. Autom. ACRA, vol. 2022-Decem, no. February 2023, 2022.

[35]. Www.tobiipro.com/, "No Title." www.tobiipro.com/.

[36]. Https://www.equivital.com/, "No Title," [Online]. Available: https://www.equivital.com/.

[37]. R. S. Kennedy, N. E. Lane, S. Kevin, and M. G. Lilienthal, "The International Journal of Aviation Psychology Simulator Sickness Questionnaire : An Enhanced Method for Quantifying Simulator Sickness," Int. J. Aviat. Psychol., vol. 3, no. 3, pp. 203–220, 1993.

[38]. W. Al-Ashwal et al., "Cybersickness Measurement and Evaluation During Flying a Helicopter in Different Weather Conditions in Virtual Reality," in 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021, pp. 2152–2157, doi: 10.1109/SMC52423.2021.9659215.

[39]. J. Iskander et al., "From car sickness to autonomous car sickness: A review," Transp. Res. Part F Traffic Psychol. Behav., vol. 62, pp. 716–726, 2019, doi: 10.1016/j.trf.2019.02.020.

[40]. H. Asadi, S. Mohamed, C. Lim, and S. Nahavandi, "A Review on Otolith Models in Human Perception," Behav. Brain Res., vol. 309, Apr. 2016, doi: 10.1016/j.bbr.2016.03.043.

[41]. A. Koohestani et al., "A Knowledge Discovery in Motion Sickness: A Comprehensive Literature Review," IEEE Access, vol. 7, pp. 85755–85770, 2019, doi: 10.1109/ACCESS.2019.2922993.

[42]. H. Asadi, S. Mohamed, K. Nelson, S. Nahavandi, and D. Rah, "Human Perception-Based Washout Filtering Using Genetic Algorithm," 2015, doi: 10.1007/978-3-319-26535-3_46.

[43]. H. Asadi, T. Bellmann, M. C. Qazani, S. Mohamed, C. P. Lim, and S. Nahavandi, "A Novel Decoupled Model Predictive Control-Based Motion Cueing Algorithm for Driving Simulators," IEEE Trans. Veh. Technol., vol. 72, no. 6, pp. 7024– 7034, 2023, doi: 10.1109/TVT.2023.3237317.

[44]. H. Asadi, S. Mohamed, C. P. Lim, S. Nahavandi, and E. Nalivaiko, "Semicircular canal modeling in human perception:," Rev. Neurosci., vol. 28, no. 5, pp. 537–549, 2017, doi: doi:10.1515/revneuro-2016-0058.

[45]. H. Asadi, A. Mohammadi, C. Lim, S. Mohamed, D. Nahavandi, and S. Nahavandi, "Increasing Motion Fidelity in Driving Simulators Using a Fuzzy-Based Washout Filter," IEEE Trans. Intell. Veh., vol. PP, p. 1, Mar. 2019, doi: 10.1109/TIV.2019.2904388.

[46]. 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," IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC, vol. 2018-Novem, pp. 1441–1446, 2018, doi: 10.1109/ITSC.2018.8569595.

[47]. 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 Syst. J., pp. 1–9, 2021, doi: 10.1109/JSYST.2021.3059285.

[48]. M. R. Chalak Qazani, H. Asadi, and S. Nahavandi, "An Optimal Motion Cueing Algorithm Using the Inverse Kinematic solution of the Hexapod Simulation Platform," IEEE Trans. Intell. Veh., p. 1, 2021, doi: 10.1109/TIV.2021.3068286.

[49]. H. Asadi, A. Mohammadi, S. Mohamed, and S. Nahavandi, "Adaptive Translational Cueing Motion Algorithm Using Fuzzy Based Tilt Coordination," in Neural Information Processing, 2014, pp. 474–482.

[50]. 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, pp. 1963–1968, doi: 10.1109/SMC.2016.7844527.

[51]. 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, pp. 2115–2121, doi: 10.1109/SMC.2018.00364.

[52]. H. Asadi, C. P. Lim, A. Mohammadi, S. Mohamed, S. Nahavandi, and L. Shanmugam, "A genetic algorithm–based nonlinear scaling method for optimal motion cueing algorithm in driving simulator," Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng., vol. 232, no. 8, pp. 1025–1038, May 2018, doi: 10.1177/0959651818772940.

[53]. 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 Syst. J., vol. 15, no. 1, pp. 445–453, 2021, doi: 10.1109/JSYST.2020.2994154.

[54]. 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, pp. 226–231, doi: 10.1109/ICCAR.2018.8384675. 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," Eng. Appl. Artif. Intell., vol. 108, p. 104564, 2022, doi: https://doi.org/10.1016/j.engappai.2021.104564.

[55]. S. Al-Serri et al., "Implementation of the Grasshopper Optimisation Algorithm to Optimize Prediction and Control Horizons in Model Predictive Control-based Motion Cueing Algorithm," in 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2022, pp. 3317–3323, doi: 10.1109/SMC53654.2022.9945416.

[56]. 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, pp. 708–713, doi: 10.1109/SMC.2019.8914458.

[57]. 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), Oct. 2019, pp. 1245–1250, doi: 10.1109/SMC.2019.8914597.

[58]. M. R. Chalak 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, pp. 1–22, Oct. 2019, doi: 10.1017/S0263574719001516.

[59]. A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "MPC-based motion cueing algorithm with short prediction horizon using exponential weighting," in 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2016, pp. 521–526, doi: 10.1109/SMC.2016.7844292.

[60]. 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, pp. 35–41, doi: 10.1109/ICIT.2019.8755051.

[61]. 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. 2015.

[62]. M. R. Chalak 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 Trans. Veh. Technol., vol. 69, no. 11, pp. 12547–12558, 2020, doi: 10.1109/TVT.2020.3023478.

[63]. M. R. Chalak Qazani, H. Asadi, S. Mohamed, and S. Nahavandi, "An Inverse Kinematic-based Model Predictive Motion Cueing Algorithm for a 6- DoF Gantry-Tau Mechanism," 2019.

[64]. M. R. Chalak 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), Mar. 2021, pp. 1–6, doi: 10.1109/ICM46511.2021.9385693.

[65]. M. R. C. Qazani, H. Asadi, C. P. Lim, S. Mohamed, and S. Nahavandi, "Prediction of Motion Simulator Signals Using Time-Series Neural Networks," IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 5, pp. 3383–3392, 2021, doi: 10.1109/TAES.2021.3082662.

[66]. 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, pp. 1869–1875, doi: 10.1109/SMC52423.2021.9659191.

[67]. M. R. C. Qazani et al., "A Prediction of Time Series Driving Motion Scenarios Using LSTM and ESN," Conf. Proc. – IEEE Int. Conf. Syst. Man Cybern., vol. 2022-Octob, pp. 1592–1599, 2022, doi: 10.1109/SMC53654.2022.9945220.