Cooperative Adaptive Cruise Control: A Gated Recurrent Unit Approach

Original
Cooperative Adaptive Cruise Control: A Gated Recurrent Unit Approach / Musa, Alessia; Anselma, PIER GIUSEPPE; Spano, Matteo; Misul, DANIELA ANNA; Belingardi, Giovanni. - STAMPA. - (2022), pp. 208-213. (2022 IEEE Transportation Electrification Conference & Expo (ITEC) Anaheim, CA, USA 15-17 June 2022) [10.1109/ITEC53557.2022.9813990].

Availability:
This version is available at: 11583/2970029 since: 2022-07-12T09:44:01Z

Publisher:
IEEE

Published
DOI:10.1109/ITEC53557.2022.9813990

(Article begins on next page)

Cooperative Adaptive Cruise Control: A Gated Recurrent Unit Approach

Alessia Musa[1,3], Pier Giuseppe Anselma[2,3], Matteo Spano[2,3], Daniela Anna Misul[1,3], Giovanni Belingardi[2,3]
[1]Department of Energy (DENERG), Politecnico di Torino, Torino, IT
[2]Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, Torino, IT
[3]Center for Automotive Research and Sustainable mobility (CARS), Politecnico di Torino, Torino, IT
alessia.musa@polito.it

***Abstract*- Embedded artificial intelligence solutions are promising controllers for future sustainable and automated road vehicles. This study presents a deep learning-based approach combined with vehicle communication technology for the design of a real-time cooperative adaptive cruise control (CACC). A particular type of recurrent neural network has been selected, namely a gated recurrent unit (GRU). GRU exhibits improved learning performance in control problems such as the CACC since it avoids the vanishing gradient problems that characterize long time series. A GRU has been trained using ad-hoc CACC datasets build-up according to an optimal control policy, i.e. dynamic programming (DP), for a battery electric vehicle. In particular, DP optimizes the longitudinal speed trajectory of the Ego (Following) vehicle in CACC so to achieve energy savings and passenger comfort improvement. Results demonstrate that the Ego vehicle controlled by the trained GRU can achieve an eco-friendly driving in CACC without compromising passenger comfort and safety requirements. Unlike DP, GRU holds strong real-time potential. The performance of the proposed GRU approach for CACC is verified by benchmarking with the optimal performance obtained off-line using DP in several driving missions.**

I. INTRODUCTION

To achieve the so-called sustainable mobility, governments established a series of objectives to be achieved in the short term, with environmental implications as well as improvements in terms of transport efficiency. One example is the introduction at a regulatory level in newly homologated vehicles of advanced driving assistance systems (ADASs), specifically designed to support the driver by ensuring safer, more efficient, and more ecological driving. Adaptive cruise control (ACC), lane keeping (LK), and emergency brake assist are some basic examples. Although some of these are by now consolidated features, the growing possibilities dictated by vehicle-to-everything connectivity have led to redefine their characteristics, paving the way for an interconnected mobility. The potential benefits deriving from ADAS technologies are manifold [1]. Improvements in energy consumption, in passenger safety and comfort along with reduced travel times are among the mainly demonstrated in the literature [2], [3]. The cooperative adaptive cruise control (CACC) often connected to the concept of eco-driving, allows for adapting the driving trajectory using the information received from other vehicles or from the infrastructure. To this end, several technologies can be used. Vehicle communication technologies such as vehicle-to-everything (V2X) or on-board sensors (RADAR, LiDAR etc) are some examples. They supply time series data information that can be used to consequently adjust the vehicle longitudinal trajectory. However, managing the powertrain according to the condition of the vehicle as well as the activity of the external environment is a complex task. Recent studies have shown the increasing real-time potential of artificial intelligence techniques [4], [5]. Therefore, the present study aims at using a deep learning-based approach to control vehicle acceleration based on information from the vehicle immediately ahead in the same carriageway. Specifically, we propose the use of a gated recurrent unit (GRU), i.e. a variant of recurrent neural network designed to avoid vanishing gradient problem linked to long time series. GRU can extract the main information of the time series and it is capable to find the nonlinear interconnection between the input features and the output ones. In this study the output information refers to the longitudinal control of the battery electric vehicle considered based on the information relative to the vehicle in front.

II. BACKGROUND

In the literature there are several works related to the cooperative adaptive cruise control application. Targets achievable through this system include energy saving, comfort and safety enhancement along with improvements in traffic throughput. A variety of algorithms can be considered in the definition of the vehicle longitudinal trajectory. Controllers based on model predictive control (MPC) are widely used [6]–[9]. As an example, a learning-based stochastic MPC is developed and validated in [8] for several driving scenarios, including cut-in manoeuvres handling. In [9], a robust MPC approach is considered to evaluate CACC system performance in case of packet loss information. The choice of an MPC-based control derives mainly from its retroactive nature; however, the accuracy of the system and the related calculation times are depending on the selected type of MPC (i.e. linear, adaptive, non-linear, robust). Another not trivial point related to MPC-based models lies in the definition of the state equations, especially when trade-off solutions between accuracy and computational times are considered because in general, they require linearization of the main systems considered (not only in dynamics equations but also in components map). To overcome these problems, machine learning techniques were considered as a possible alternative. In 2011, Desjardins et al [10] presented a reinforcement learning-approach for vehicle control. More specifically, they used a policy gradient algorithm and a backpropagation neural network to achieve the desired control trajectory. Speaking about recurrent neural networks, Tian et al [11] proposed a long-short term memory neural network to predict the lead vehicle profiles related to longitudinal dynamics control aiming to compensate for communication delay of communication technology. Main findings refer to improvement in the string stability when the communication delay exceeds a certain threshold. Compared to the techniques currently present in the literature, the main contribution of the proposed control approach is the application of a gated recurrent unit to solve the CACC control problem instead of using it to compensate for any communication problems since the latter could be solved by introducing synthetic samples into the considered datasets.

III. METHODOLOGY

The vehicle considered in the present work is a battery electric vehicle (BEV) and its characteristics data are listed in TABLE *I*.

TABLE I Vehicle characteristics data

Curb weight	1474 kg
Battery capacity	42 kWh
EM peak power	83 kW

Main powertrain components such as electric machine and energy storage system were modelled using a map-based approach. The vehicle modelling has been described in the authors' previous works, for more details please refer to [12], [13]. The problem under analysis refers to a string of two vehicles on the same carriageway, equipped with V2V technology and on-board sensors such as LiDAR and Radar. They allow for the exchange of information in terms of the distance between the two vehicles as well as the speed of the vehicle in front. The considered driving scenario will refer to the vehicle that follows as 'ego vehicle', whereas to the vehicle at the front as 'leading vehicle'. The considered flow topology is predecessor following [14] meaning that the ego vehicle only receives the leading vehicle information in terms of velocity and position. For simplicity, a homogeneous string of vehicles has been considered i.e., all vehicles are of the same type. The control problem referred to has as its objective the definition of the optimal ego vehicle longitudinal trajectory by minimizing the energy consumption of the battery. The control problem under analysis was solved by considering a machine learning technique based on recurrent neural networks (RNNs). The reasons behind this choice are related to the characteristics of the RNNs, mainly designed for time series handling. This type of network is characterized by a memory cell which manages the information coming from the previous and current inputs to generate the current output(s). Referring to the proposed application, the present work will consider standard driving missions characterized by a time duration greater than 1000s. Therefore, to avoid vanishing gradient problems featuring the long time series, a particular type of RNN is selected, namely Gated Recurrent Unit (GRU). The latter is characterized by memory units with two gates, the update and the reset ones, mainly designed to let information pass selectively: their function is to establish which information must be preserved and which ones are to be discarded. Since for the considered application this approach has not been applied extensively, it might be worth clarifying the functioning of the gates and their structure. For a certain input vector x_t, the characteristic equations of the GRUs can be summarized in Eqs *(1)*-*(4)*

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{1}$$

$$r_t = \sigma(W_r\, x_t + U_r\, h_{t-1}) \tag{2}$$

$$\hat{h}_t = tanh(W x_t + r_t \odot h_{t-1}) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{4}$$

where $z_t, r_t\ and\ \hat{h}_t$ respectively refer to the update gate, reset gate and candidate activation vector. The sigmoid and hyperbolic activation functions are respectively indicated with σ and tanh whereas ⨀ represents element wise multiplication. W_z, U_z, W_r, U_r, W, U represent the weight matrices. The reset gate and the update gate have a similar formal structure but differ in their purpose. The reset gate r_t decides which information from the previous hidden state (h_{t-1}) must be substituted with the current inputs (Eq 3). The update gate z_t directly affects the current hidden state h_t deciding which information of the previous hidden state should be updated with the current candidate activation vector $\hat{h}_t$ (Eq 4). For more details regarding the GRU, please refer to [15]–[18].The GRU network has been trained using the results obtained through dynamic programming (DP) . Energy saving and passenger comfort enhancement result from the formulation of the DP optimization target performed by Anselma and Belingardi in [12] and Spano et al in [19]. The lead vehicle speed and position signals, ego vehicle velocity and position at the previous time instant and inter-vehicle distance (IVD) between the two vehicles were selected as network input features. As an output signal, the network processes the velocity signal, and consequently calculate the acceleration signal, of the ego vehicle. Each feature has been normalized to speed up the training and avoid prioritization phenomena between the different features (mainly dictated by differences in the range of variation of the selected features), as follows:

$$\widehat{x_{ij}^k} = \frac{x_{ij}^k - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{5}$$

where i is the generic i-th feature, j is the j-th sampling point, k represents the driving cycle considered and $\hat{x}$ the normalized feature. The GRU-based system was evaluated considering different batch sizes and GRU units. The batch size refers to the number of samples used during the training phase to update the neural network weights whereas the GRU units refers to the number of units in GRU 1st layer (and consequently in the other GRU layers since their value is derived from the 1st layer so to reduce the number of tunable parameters). The optimizer and the number of samples in the past were instead kept constant. The Adam optimizer has been selected owing to its capability to combine the advantages of RMSProp and AdaGrad [20], [21] optimizers. The number of samples in the past, on the other hand, was selected through a trial-and-error approach by selecting the minimum possible value so as not to have any memory problems in view of a possible hardware application. The metrics monitored during the training phase were the mean squared error (MSE) and the mean absolute error (MAE). MSE is defined as the average squared difference between the estimated value and the actual value whereas MAE refers to the average of the absolute error, as shown respectively in Eqs (6)-(7).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{6}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{7}$$

To improve the network performance, a regularization technique has been employed by reducing the learning rate for the optimizer if the validation loss has not improved since a pre-defined number of epochs.

IV. Results discussion

In this work, CACC control problem was addressed by a regression approach on the speed of the ego vehicle. The resulting acceleration profiles as well as the IVD were derived from the speed prediction profile. A set of conventional driving cycles (Artemis driving cycles, Worldwide Harmonized Light-duty vehicles Test Cycle - WLTP, EPA Highway Fuel Economy Test Cycle - HWFET, Supplemental Federal Test Procedure – US06) has been considered. TABLE 2 sums up their main characteristics in terms of cycle duration, distance, mean and maximum speed. Particularly, Artemis driving cycles have been considered in the GRU training phase. The training dataset has been defined considering diverse driving scenarios represented by different acceleration ranges, mean and maximum velocity.

The simulations were performed for three different sets of validation and test cycles. Specifically:

1. Validation cycle: WLTP; test cycles: US06 and HWFET
2. Validation cycle: US06; test cycles: WLTP and HWFET
3. Validation cycle: 20% of Artemis driving cycle (a portion not used for training); test cycles: WLTP, US06 and HWFET

The simulation settings of the GRU-system were derived by applying a grid search approach for the model fine-tuning; the goal was finding the optimal combination of the model's hyperparameters that results in more accurate predictions. For the specific case study under analysis, the hyperparameters considered are batch size and GRU number of units. The hyperparameters considered are only some of the possible choices but they were considered sufficient to demonstrate the potential of the proposed approach based on the experience of the authors and of the main works in the literature. TABLE 3 summarizes the main characteristics of the GRU architecture, its main settings, and the range of variations of the hyperparameters to be tuned in the grid search. The grid search results were analysed considering the GRU's prediction performance in terms of root mean square error (RMSE) of the predicted ego vehicle velocity with respect to DP profiles, thus providing an indication of the standard deviation of the prediction errors. In the following, the results related to the set of simulations using the WLTP cycle as the validation dataset will be analysed. This choice is motivated by the desire to use a driving cycle representative of different driving scenarios (urban, suburban and highway) as a validation cycle, thus guaranteeing network generalization performance. For the sake of conciseness, Fig. *1* shows only the results of two test cases evaluated for the entire batch size set. Specifically, Fig. *1* shows the RMSE values for all the cycles considered when the number of units of the first layer of the GRU is equal to 32 (upper part) and 256 (bottom part) respectively. In the case of 32 units, better performance is obtained for a batch size equal to 32; instead, considering 256 units the best results are obtained for batch size equal to 64. However, the improvement obtained, in our opinion, does not justify the increase in required computational times that result from the increase in the learnable parameters of the network itself (directly linked with the number of GRU units). For this reason, the selected combination of hyperparameters, as the best compromise between accuracy and computational times, is the one with 32 units in the first layer and batch size 32. For the selected combination, the training (blue) and validation (purple) loss trends are shown in *Fig. 2*. It is worth recalling that the metric selected for the loss was the MSE and that normalized features are used in the training phase, so the MSE refers to a normalized speed. As it can be seen from the loss trend figure, slightly lower values are obtained in validation than in training. This deviation can be attributed to the use of the dropout in the training phase but not in the validation phase; moreover, in validation the loss is evaluated at the end of the epoch rather than during the epoch itself. The resulting predicted velocity profiles are shown in black in *Fig. 3* for the test and validation datasets (left side of the figure), namely US06, HWFET and WLTP; the same figure shows the profiles obtained with the DP for the ego vehicle (red dash-dot line) and the speed profile of the lead vehicle (grey). On the right, a zoom of the first 100s of the driving cycles is shown to highlight the main differences between the control approaches presented. These profiles were analysed in post-processing, obtaining the relative results in terms of energy consumption and passenger comfort reported in TABLE 4.

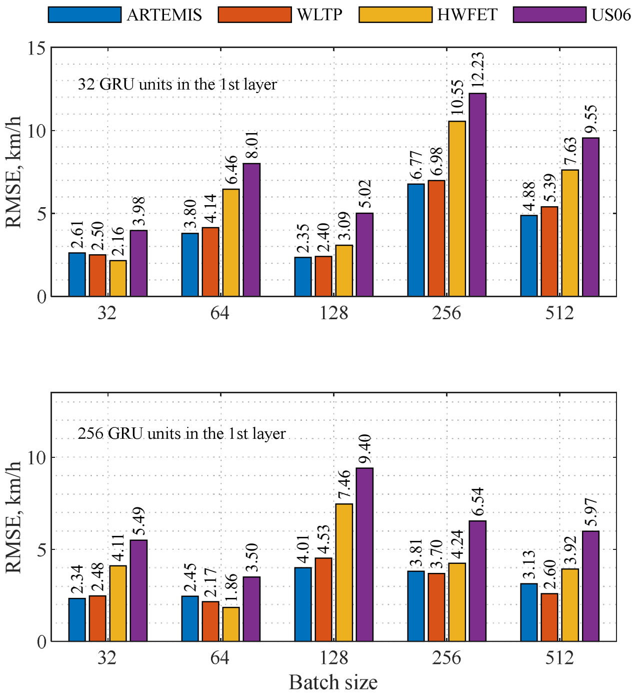

Fig. 1 RMSE results with respect to DP profiles for different batch sizes and driving cycles considering different GRU units in the 1st layer i.e 32 GRU units (upper part) and 256 (bottom part).

TABLE 2
MAIN CHARACTERISTICS OF TRAINING AND TESTING DATASETS

	Duration (s)	Distance (km)	Average Speed (km/h)	Max speed (km/h)
ARTEMIS URBAN	993	4.87	17.7	57.3
ARTEMIS RURAL	1082	17.3	57.5	111.1
ARTEMIS MOTORWAY	1068	29.6	99.6	150
US06	596	12.8	77.9	129
WLTP	1800	23	46.5	131.3
HWFET	765	16.45	77.7	97

TABLE 3
GRU ARCHITECTURE AND SETTINGS

Layer (Type)	**Units**	**Settings**	
GRU	m	Dropout rate	0.2
Dropout	-	GRU's Unit values range	32-512
GRU	m	Batch size range	32-512
Dropout	-	# of samples (in the past)	5
GRU	m/2	Optimizer	Adam
Dropout	-	Loss	MSE
Dense	1	Metrics	MSE, MAE

TABLE 4
RESULTS COMPARISON BETWEEN THE PROPOSED GRU APPROACH AND DP

		Lead Vehicle	Ego Vehicle (GRU)	Ego Vehicle (DP)
WLTP	Energy consumption (kWh/100km)	17.46	16.55	16.55
	RMS of vehicle acceleration (m/s^2)	0.52	0.37	0.34
HWFET	Energy consumption (kWh/100km)	17.05	16.86	16.75
	RMS of vehicle acceleration (m/s^2)	0.3	0.28	0.27
US06	Energy consumption (kWh/100km)	21.31	20.71	20.28
	RMS of vehicle acceleration (m/s^2)	0.99	0.64	0.61

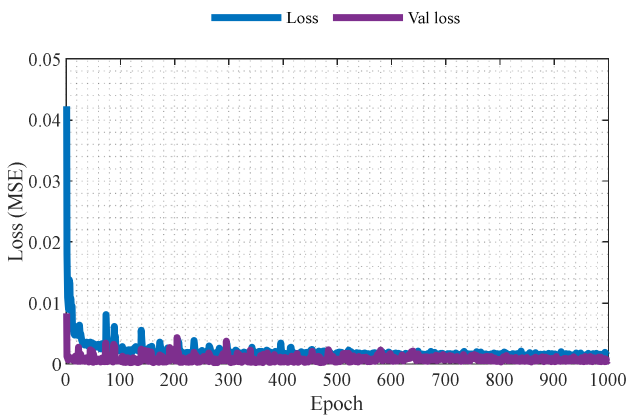

Fig. 2 Train and Validation loss trends. Simulation settings: ARTEMIS cycle for training, WLTP for validation, 32 GRU units in the 1st layer, 1000 # of epochs

The term *RMS of vehicle acceleration* refers to the ride quality evaluated based on the root mean square of acceleration signal [22], [23]. The results shown underline the potential in terms of generalization capacity of the neural network on the test datasets (HWFET and US06) as well as on those used in the training phase. However, several challenges and limits remain open and will need to be analysed in future works. Among the main ones, the proposed network should be implemented within a medium-fidelity simulation environment to be able to check the IVD signal between the two vehicles and re-evaluate the speed prediction in the event of any deviations from the established constraints. This consideration is especially true for long driving cycles that show a deterioration in performance towards the end of the cycle, as can be seen in the lower part of Fig. 3 for the WLTP cycle after 1200s.

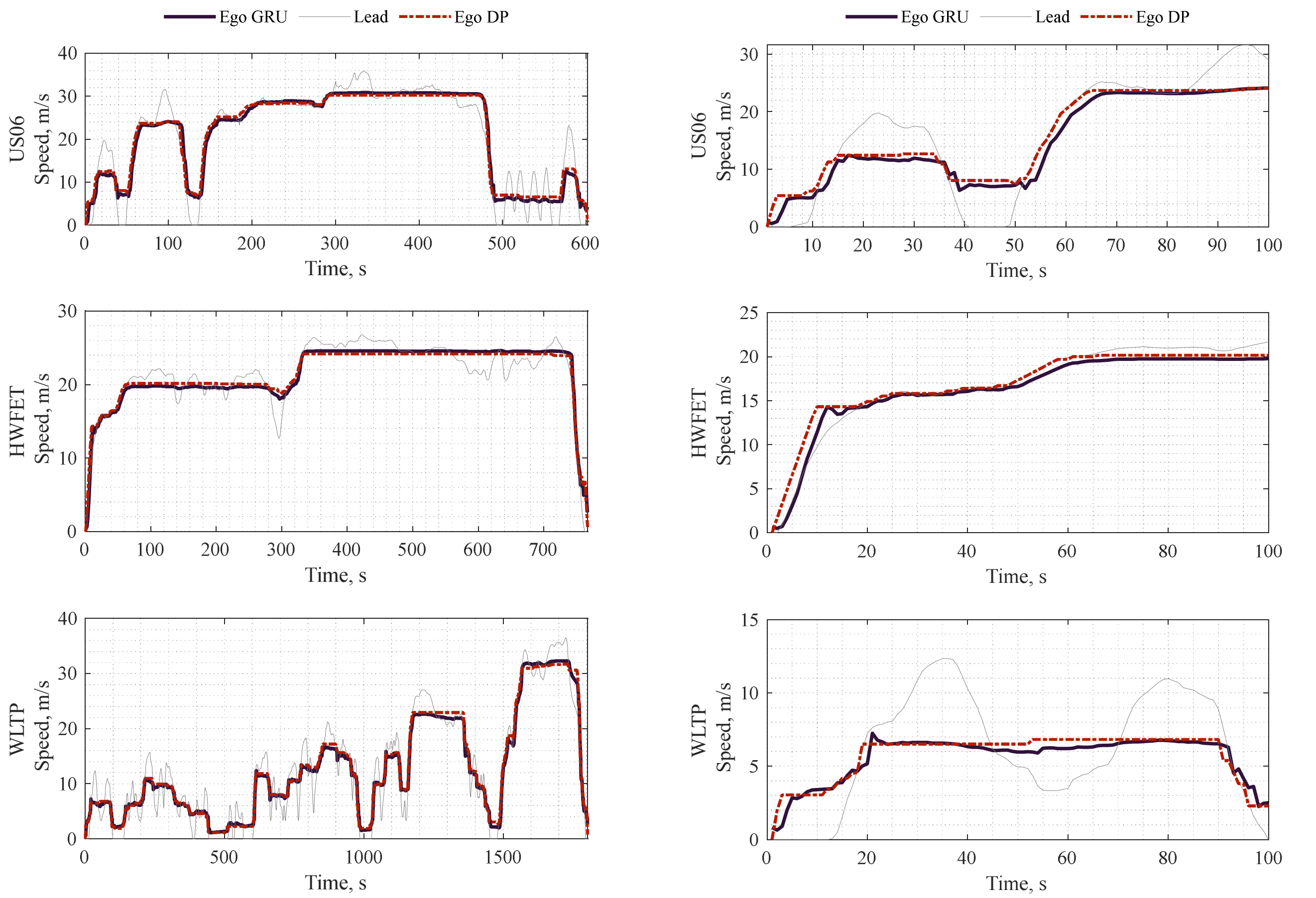

Fig. 3 Results comparison between the proposed GRU approach and DP in terms of speed for the test and validation datasets (left side of the figure), i.e. US06 (upper part), HWFET (middle) and WLTP (bottom part). On the right, a zoom of the first 100s. Simulation settings: ARTEMIS cycle for training, WLTP for validation, batch size 32, GRU units in the 1st layer 32, # of epochs 1000

This deterioration may be explained by the lack of an index indicating the remaining time to travel. However, introducing a feature to represent this information would lead to other considerations related to the user's need to always select a route on the GPS, opening further challenges beyond this work. In addition, the performance obtained in terms of ride comfort does not seem optimal. This lack is attributable to a choice of non-optimal input features to the model as well as a depth of the architecture that may require further investigation. It is worth recalling that depth of architecture refers to the number of layers in the neural network.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, a gated recurrent unit approach for the design of a real-time cooperative adaptive cruise control has been presented. The GRU has been trained using ad-hoc datasets build-up according to an optimal control policy, namely dynamic programming. The latter has been formulated to account for passenger comfort as well as energy saving enhancement. The GRU control was applied to a two-vehicle string and was found to efficiently mimic optimal trajectory determined through DP simulation. The problem was addressed using a regression approach with the aim of predicting the speed of the ego vehicle. A similar approach to vehicle ego acceleration would have been unsatisfactory given the nervous nature of the acceleration signal. The results shown highlight the potential of such an approach for solving the CACC problem achieving an eco-friendly driving without compromising passenger comfort and safety requirements. Unlike DP, GRU holds strong real-time potential: once trained, it can make predictions in real time (on the order of microseconds in a low fidelity simulation environment). However, there are still several limitations that will need to be addressed in future work. Among these, the need to improve performance in terms of comfort and accuracy in the final part of the mission without introducing any temporal features are the most challenging. Future works might include real driving missions to achieve a human-aware cooperative adaptive cruise control by training the GRU on user driving cycles so as to achieve a customized CACC.

ACKNOWLEDGMENT

This research work was developed in the framework of the activities of the Interdepartmental Center for Automotive Research and Sustainable mobility (CARS) at Politecnico di Torino.

References

[1] R. Jolly, «Advanced Driver Assistance Systems (ADAS): Cooperative Adaptive Cruise Control (CACC)», pag. 4.

[2] C. Massera Filho, M. H. Terra, e D. F. Wolf, «Safe Optimization of Highway Traffic With Robust Model Predictive Control-Based Cooperative Adaptive Cruise Control», *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, n. 11, Art. n. 11, nov. 2017, doi: 10.1109/TITS.2017.2679098.

[3] D. R. Lopes e S. A. Evangelou, «Energy savings from an Eco-Cooperative Adaptive Cruise Control: a BEV platoon investigation», in *2019 18th European Control Conference (ECC)*, giu. 2019, pagg. 4160–4167. doi: 10.23919/ECC.2019.8796226.

[4] A. Moujahid *et al.*, «Machine Learning Techniques in ADAS: A Review», in *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, giu. 2018, pagg. 235–242. doi: 10.1109/ICACCE.2018.8441758.

[5] D. Lee e H. Yeo, «Real-Time Rear-End Collision-Warning System Using a Multilayer Perceptron Neural Network», *IEEE Trans. Intell. Transport. Syst.*, vol. 17, n. 11, pagg. 3087–3097, nov. 2016, doi: 10.1109/TITS.2016.2537878.

[6] N. Wikström, A. F. Parrilla, S. J. Jones, e A. Grauers, «Energy-Efficient Cooperative Adaptive Cruise Control with Receding Horizon of Traffic, Route Topology, and Traffic Light Information», *SAE Intl. J CAV*, vol. 2, n. 2, Art. n. 2, mag. 2019, doi: 10.4271/12-02-02-0006.

[7] D. He, T. Qiu, e R. Luo, «Fuel efficiency-oriented platooning control of connected nonlinear vehicles: A distributed economic MPC approach», *Asian Journal of Control*, vol. 22, n. 4, pagg. 1628–1638, lug. 2020, doi: 10.1002/asjc.2049.

[8] H. Kazemi, H. N. Mahjoub, A. Tahmasbi-Sarvestani, e Y. P. Fallah, «A Learning-Based Stochastic MPC Design for Cooperative Adaptive Cruise Control to Handle Interfering Vehicles», *IEEE Trans. Intell. Veh.*, vol. 3, n. 3, pagg. 266–275, set. 2018, doi: 10.1109/TIV.2018.2843135.

[9] E. van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, e N. van de Wouw, «Robust model predictive cooperative adaptive cruise control subject to V2V impairments», in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, ott. 2017, pagg. 1–8. doi: 10.1109/ITSC.2017.8317758.

[10] C. Desjardins e B. Chaib-draa, «Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach», *IEEE Trans. Intell. Transport. Syst.*, vol. 12, n. 4, pagg. 1248–1260, dic. 2011, doi: 10.1109/TITS.2011.2157145.

[11] B. Tian, G. Wang, Z. Xu, Y. Zhang, e X. Zhao, «Communication delay compensation for string stability of CACC system using LSTM prediction», *Vehicular Communications*, vol. 29, pag. 100333, giu. 2021, doi: 10.1016/j.vehcom.2021.100333.

[12] P. G. Anselma e G. Belingardi, «Enhancing Energy Saving Opportunities through Rightsizing of a Battery Electric Vehicle Powertrain for Optimal Cooperative Driving», *SAE Intl. J CAV*, vol. 3, n. 2, pagg. 12-03-02–0007, giu. 2020, doi: 10.4271/12-03-02-0007.

[13] M. Spano, P. G. Anselma, A. Musa, D. A. Misul, e G. Belingardi, «Optimal Real-Time Velocity Planner Of A Battery Electric Vehicle In V2V Driving», presentato al 2021 IEEE Transportation Electrification Conference (ITEC 2021), giu. 2021.

[14] Y. Zheng, S. Eben Li, J. Wang, D. Cao, e K. Li, «Stability and Scalability of Homogeneous Vehicular Platoon: Study on the Influence of Information Flow Topologies», *IEEE Trans. Intell. Transport. Syst.*, vol. 17, n. 1, pagg. 14–26, gen. 2016, doi: 10.1109/TITS.2015.2402153.

[15] K. Cho, B. van Merrienboer, D. Bahdanau, e Y. Bengio, «On the Properties of Neural Machine Translation: Encoder–Decoder Approaches», in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, 2014, pagg. 103–111. doi: 10.3115/v1/W14-4012.

[16] K. Cho *et al.*, «Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation», in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pagg. 1724–1734. doi: 10.3115/v1/D14-1179.

[17] J. Chung, C. Gulcehre, K. Cho, e Y. Bengio, «Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling», *arXiv:1412.3555 [cs]*, dic. 2014, Consultato: 15 dicembre 2021. [Online]. Disponibile su: http://arxiv.org/abs/1412.3555

[18] W. Li, H. Wu, N. Zhu, Y. Jiang, J. Tan, e Y. Guo, «Prediction of dissolved oxygen in a fishery pond based on gated recurrent unit (GRU)», *Information Processing in Agriculture*, vol. 8, n. 1, pagg. 185–193, mar. 2021, doi: 10.1016/j.inpa.2020.02.002.

[19] M. Spano, A. Musa, P. G. Anselma, D. A. Misul, e G. Belingardi, «Battery Electric Vehicles Platooning: Assessing Capability of Energy Saving and Passenger Comfort Improvement», in *2021 AEIT International Conference on Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, Torino, Italy, nov. 2021, pagg. 1–6. doi: 10.23919/AEITAUTOMOTIVE52815.2021.9662788.

[20] D. P. Kingma e J. Ba, «Adam: A Method for Stochastic Optimization», *arXiv:1412.6980 [cs]*, gen. 2017, Consultato: 15 aprile 2022. [Online]. Disponibile su: http://arxiv.org/abs/1412.6980

[21] J. Brownlee, «Gentle Introduction to the Adam Optimization Algorithm for Deep Learning», *Machine Learning Mastery*, 2 luglio 2017. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ (consultato 15 aprile 2022).

[22] Department of Railway Vehicles, Faculty of Transport, University Politehnica of Bucharest, 313 Splaiul Independenţei, 060042, Bucharest, Romania e M. Dumitriu, «Analysis of the dynamic response in the railway vehicles to the track vertical irregularities. Part II: The numerical analysis», *JESTR*, vol. 8, n. 4, pagg. 32–39, ago. 2015, doi: 10.25103/jestr.084.05.

[23] C. C. Smith, D. Y. McGehee, e A. J. Healey, «The Prediction of Passenger Riding Comfort From Acceleration Data», *Journal of Dynamic Systems, Measurement, and Control*, vol. 100, n. 1, pagg. 34–41, mar. 1978, doi: 10.1115/1.3426338.