Article | Received 5 July 2025; Revised 26 January 2026; Accepted 27 January 2026; Published 27 February 2026
https://doi.org/10.55092/aias20260002

Real-time data-driven methods for estimating vehicle longitudinal and lateral velocities

Amin Habibnejad Korayem[1,2,*], Amir Khajepour[2], Ehsan Hashemi[3], Qingrong Zhao[1] and Alireza Kasaiezadeh[1]

[1] General Motors Company, Warren, USA

[2] Mechanical and Mechatronics Engineering Department, University of Waterloo, Waterloo, Canada

[3] Mechanical Engineering Department of the University of Alberta, Alberta, Canada

* Correspondence author; E-mail: amin.korayem@uwaterloo.ca.

Highlights:

- Novel feature formulation combining sensitivity analysis and physics-based normalization for cross-vehicle scalability.
- Two lightweight FNN estimators for real-time longitudinal and lateral velocity estimation.
- Validated in simulation and experiments, with strong generalization to unseen vehicles.

Abstract: Estimating vehicle velocity in the full-range of vehicle operation across various driving scenarios is a challenge for most model-based state observers, especially for vehicles operating at the limits of handling due to highly nonlinear vehicle dynamics, particularly on slippery roads. To address this challenge, this paper proposes a machine learning (ML) based method for vehicle velocity estimation. Two novel vehicle velocity estimators based on multi-layer feedforward neural network (FNN) are designed to estimate vehicle longitudinal and lateral velocities, respectively. The inputs of the proposed FNN estimators are selected based on a sensitivity analysis of vehicle dynamics models and are normalized by vehicle parameters to ensure that the trained FNN estimators have generic formulas transferable to vehicles of different specifications. Experimental and simulation results—including tests on vehicles not seen during training—demonstrate that the proposed velocity estimators provide accurate estimations in various challenging scenarios and are robust to differences in vehicle specifications. Moreover, the computational cost is low enough for real-time implementation.

Keywords: longitudinal velocity estimation; lateral velocity estimation; machine learning; deep neural network; vehicle dynamics; sensitivity analysis

1. Introduction

Vehicle velocity estimation is crucial for the design of various vehicle control systems such as stability control, motion control, active safety, and autonomous driving systems. The expanded operation domains

of modern vehicles demand even higher levels of accuracy, robustness, and adaptability for vehicle velocity estimation. Conventional velocity estimation methods mainly rely on kinematic models and vehicle models. For example, vehicle kinematic equations were utilized to estimate vehicle longitudinal and lateral velocities by using Kalman-based estimation algorithms [1–4], or non-linear observers [5,6]. However, due to the highly nonlinear behavior of a vehicle—particularly under high tire slip or saturation conditions—the linearized vehicle model Jacobian can locally lose rank, reducing observability for conventional model-based estimators. This weak observability arises not from a lack of excitation but from nonlinear coupling among tire forces, slip dynamics, and parameter uncertainties. Moreover, a known limitation of Kalman-based and observer-based estimators is the challenge of lateral velocity observability during straight-line driving, where lateral excitation is minimal. Although observability improves under dynamic maneuvers, accurate estimation remains sensitive to vehicle and tire parameterization [7,8]. Under such conditions, the conventional observers with access only to the standard vehicle dynamics sensors cannot cover the full range of operation, or may require a drastic change in calibrations. As a result, both limitations can lead to state estimation failure or inaccuracies.

To address the limitations of conventional observers, researchers have explored machine learning based approaches, such as Neural Networks (NN), to estimate vehicle states and parameters [9–17]. The high degree of nonlinearity of NN make it possible to accurately model complex system dynamic behaviors and lead to improved estimation performance. For instance, in reference [18], a NN structure comprising a sparse autoencoder and SoftMax regression was used to classify road excitation for semi-active suspension systems. Similarly, in references [19–24], a general regression NN was developed to observe the vehicle sideslip angle. The neural network method was also used in references [25–28] to estimate the vehicle roll angle. The findings in reference [28] showed that NN based estimation results are comparable to those obtained using model-based techniques and in certain scenarios, such as on slippery roads, the neural network approach outperformed the model-based estimation. These studies demonstrated the potential of NNs to in achieving excellent estimation performance, while generalizing across different surfaces, tires, and driving situations.

In recent years, a few studies have investigated the use of NNs for vehicle velocity estimation. A recurrent neural network (RNN) based estimation method was proposed by reference [29] to address velocity estimation in extreme scenarios, which showed that vehicle lateral velocity can be estimated with greater accuracy than model-based approaches. In reference [30], multiple RNN cells were constructed and trained individually to estimate lateral velocity and several other vehicle states, and simulated data were used to validate the work in several test scenarios. Additionally, references [31–33] proposed a long short-term memory (LSTM) network, a variant of RNN, to estimate vehicle lateral velocity, which demonstrated good performance at low and medium speeds, but not in the high-speed range. Although these studies shed lights on the NN based velocity estimator design, two important areas of research for estimation performance and real-world applications have received limited attention. The first is how to objectively select the most effective input features relevant to estimation of vehicle velocities, rather than relying solely on experience or intuition. The second is algorithm generality and transferability, as the current achievements are limited to a single vehicle model with the same specifications. If the algorithm is to be extended to a vehicle with different specifications, new data must be collected and the NNs must

also be re-trained from scratch, which is impractical given the amount of work, time, and cost involved.

This paper aims to provide insights into the two areas mentioned above by proposing two novel Feedforward Neural Network (FNN) based vehicle velocity estimators to estimate vehicle longitudinal and lateral velocities, respectively. The input features of the proposed FNN estimators are formulated in two steps. Firstly, the most relevant signals are selected through a sensitivity analysis of available sensor measurements in the vehicle. Then, the selected signals are further normalized by vehicle parameters based on vehicle dynamics analysis. This unique input feature formulation makes the FNN estimators generic to vehicles of different specifications and allows data of from different vehicles to be added during the training phase, thereby enriching the data over various driving scenarios. As a result, the trained FNN estimators can be widely applied to a variety of vehicles of different specifications without requiring re-training the NNs for each application. The proposed NN velocity estimators are tested and validated using both simulation data and real vehicle data, demonstrating their ability to provide accurate estimations in various challenging scenarios and their robustness to changes in vehicle specifications. Furthermore, the algorithm is lightweight enough for real-time implementation. In summary, the contributions of the paper are as follows:

- A novel method for formulating input features is proposed, which utilizes sensitivity analysis to select the most relevant input signals and can accommodate differences in vehicle specifications by incorporating vehicle parameters.
- Two FNN based estimators are designed to estimate longitudinal and lateral velocities, respectively, which are generic and applicable to vehicles of different specifications.
- Both simulated and experimental testing demonstrate that the proposed FNN algorithms provide accurate estimation performance and are robust to changes in vehicle specifications.
- The input feature formulation method in Equation (1) and the generic design of FNN estimators in Equation (2) can be readily generalized to the estimation of other state and parameters.

The rest of this paper is organized as follows: Section 2 introduces the overall structure of the proposed ML-based vehicle velocity estimation algorithms, including the design and formulation of the input feature vector to the network. Section 3 presents collected dataset, which contains both simulation and experimental test data, in details. The vehicle longitudinal and lateral velocity estimation results are illustrated and discussed in Sections 4. Finally, Section 5 draws the conclusions.

2. Machine learning based velocity estimation

2.1. Input feature selection

To achieve accurate longitudinal and lateral velocity estimation, the input features for each of the two FNN estimation models need to be formulated appropriately to well represent information required for estimating either longitudinal velocity or lateral velocity. A sensitivity analysis can be conducted over all the measurable inputs at the current time and their historical time steps to rank their importance with respect to the vehicle longitudinal and lateral velocities.

The Gradient Boosting Regression Tree (GBRT) method is used to evaluate signal importance.

The GBRT model used 300 estimators, a learning rate of 0.05, and a maximum tree depth of 5. The

top 10% of features ranked by normalized importance ($0.1 \leq$ importance) were retained for input selection. The sensitivity analysis was conducted on a dataset of 203 maneuvers covering DLC, SS, sine, and slalom tests across three vehicle types. GBRT begins by training a decision network where each observation is assigned with an equal weight [34]. GBRT method is used to rank input signal importance. GBRT builds an ensemble of decision trees, each correcting the errors of its predecessor, allowing it to capture nonlinear interactions between features. Compared to traditional methods like cross-correlation—which assume linear relationships—GBRT provides a more robust feature ranking in the presence of nonlinear vehicle dynamics. By evaluating the first layer, the weights of those observations that are difficult to classify or low are increased. By evaluating the weights of neurons, the importance of the signals and their history data with respect to the output (e.g. either lateral velocity or longitudinal velocity) is generated, as shown in Table 1 and Figure 1.

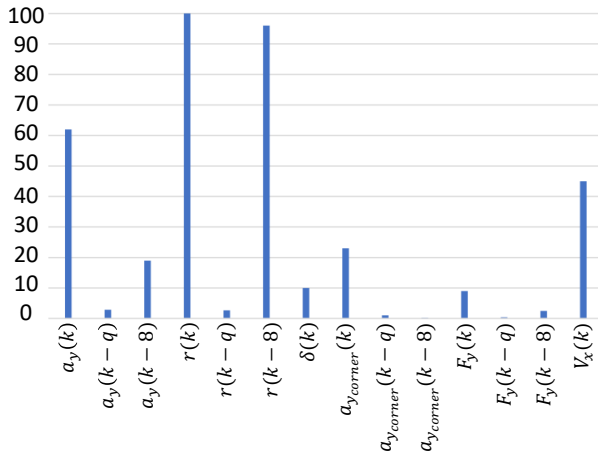

Figure 1. The input feature importance of available measurements for lateral velocity estimation.

The inputs for these sensitivity analyses tests are selected based on the model-based approaches [35] since model-based approaches are designed based on the vehicle longitudinal and lateral dynamics in which the relation between the measurements and output signal is specified.

Figure 1 illustrates the importance of the inputs and their temporal signals with respect to the output where $q \in [2,4,6]$ and $ay_{corner}(k) = a_y(k) - r(k)v_x(k)$ is an augmented input signal based on the model-based approaches. The term $ay_{corner}(k)$ represents the time derivative of lateral velocity, excluding roll effects and it is commonly referred to as cornering acceleration, as it captures the net lateral acceleration experienced during turning. As can be seen in Figure 1, the history of the augmented lateral acceleration, and lateral tire force with its history are less important and have less effect on the network for estimating the vehicle lateral velocity. Moreover, the vehicle yaw rate and lateral acceleration have the highest importance rate comparing with other available measurement signals. Notably, the steering wheel angle exhibits relatively low sensitivity since its effect on instantaneous vehicle velocity is primarily indirect—manifested through its influence on yaw rate, which already dominates the feature importance ranking. While the sensitivity analysis guided the initial feature selection, certain features included in Ax and Ay were additionally incorporated based on domain knowledge and empirical validation. This hybrid approach balances data-driven insights with physical interpretability, ensuring robustness across varying driving conditions.

Table 1. The input feature importance of available measurements for longitudinal velocity.

Input feature	ω(**k**)	**ax**(**k**)	**Yawrate**(**k**)	**SWA**(**k**)	**ay**(**k**)
Importance percentage	100	20	18.5	10.6	3.1

Table 1 shows the importance of the inputs and their temporal signals with respect to the output which is the vehicle longitudinal velocity. As can be seen in Table 1, the lateral acceleration, and steering wheel angle are less important and have less effect on the network for estimating the vehicle longitudinal velocity. Hence, variables can be considered in the input vector based on their importance level, and there is no need to consider all the available measurement signals as their effectiveness vary. Similarly, the sensitivity analysis results for vehicle lateral velocity estimation are shown in Figure 1.

Given the sensitivity analysis results, the initial list of volunteer input features to estimate vehicle longitudinal and lateral velocities are selected as follows:

$$\mathbf{A}_x(k) = \begin{bmatrix} a_x(k) & r(k) & \omega(k) \end{bmatrix}^T \tag{1}$$

$$\mathbf{A}_y(k) = \begin{bmatrix} a_x(k) \\ a_y(k) \\ r(k) \\ r(k-q) \\ \delta(k) \\ \omega(k) \\ ay_{corner}(k) \end{bmatrix} \tag{2}$$

where $a_x(k)$, $a_y(k)$, $r(k)$, $\delta(k)$, and $\omega(k)$ are representing the longitudinal and lateral accelerations, yaw rate, steering wheel angle, and wheel speeds, respectively.

One of the main goals envisioned for this paper is to have a scalable solution by which a reasonable performance is expected even for an unseen data set from a vehicle with a different configuration. To achieve this goal, vehicle parameters which are key to differentiate a vehicle from another are incorporated in the algorithm to normalize the input features based on vehicle dynamics formulas.

First For longitudinal velocity estimation, given the fact that the longitudinal vehicle dynamics is a first-order differential equation and the wheel acceleration ($\dot{\omega}(k)$) exists in the longitudinal vehicle dynamic, wheel speed difference ($\Delta\omega(k) = \omega(k) - \omega(k-1)$) is incorporated in the input feature. Moreover, four corner wheel acceleration difference which contain vehicle track width are added to the input feature by following the vehicle and wheel dynamics. Hence, the final input feature vector to estimate longitudinal velocity using FNN vector is updated based on Equation (1) as follows:

$$\mathbf{A}_x(k) = \begin{bmatrix} a_x(k) \\ a_x(k-q) \\ r(k) \\ R_{eff}\omega(k) \\ a_x(k) - T_w r(k) - R_{eff}\dot{\omega}(k) \\ \omega(k) - \omega(k-1) \end{bmatrix} \tag{3}$$

where T_w, and R_{eff} are the vehicle track width and tire effective radius, respectively.

Remark 1: The slip ratio effect is considered using the wheel rotational speed difference to make the algorithm robust against highly slippery conditions aiming to address the different system bandwidth in wheel stability.

For lateral velocity estimation, the selected initial input features in Equation (2) are further formulated based on lateral vehicle dynamics and normalized using influential vehicle parameters such as nominal mass, yaw moment of inertia, steering ratio, and effective wheel radius to ensure scalability and generalizability. These parameters are typically available in production vehicles or can be estimated via onboard diagnostics, making the approach practical for real-world deployment. As a result, the final input feature vector for the FNN-based lateral velocity estimator is defined as follows:

$$\mathbf{A}_y = \begin{bmatrix} ma_x(k) \\ ma_y(k) \\ I_z r(k) \\ I_z r(k-q) \\ \delta(k)/S_{ratio} \\ R_{eff}\omega(k) \\ a_y(k) - r(k)v_x(k) \end{bmatrix} \tag{4}$$

where m, I_z, and S_{ratio}, are the vehicle mass, yaw momentum of inertia, and steering ratio, respectively.

This unique input feature selection process allows data of different vehicle specifications to be added during the training phase, and to heal and enrich data over different driving scenarios, which can significantly leverage the data collection efforts by utilizing available data from different vehicles. As a result, the trained FNN estimators are generic to be widely applied to vehicles of different specifications without much extra efforts of re-training the networks.

2.2. FNN configuration

The NN architecture plays an important role on the estimation performance. Prior studies [29–31] proposed RNN and LSTM networks for vehicle velocity estimation. RNN and LSTM are designed to cope with the correlation within time series in both short and long term, which is helpful to address the dynamic behavior of the vehicle. However, due to the sophisticated structures, RNN and LSTM generally consume more computing resources than FNN (fully connected neural networks), which can be a roadblock for real world implementation in view of the capacity constraints of the production hardware, particularly for main-stream vehicles. It is worth exploring the use of FNN for vehicle velocity estimation. If the history of the input signals are appropriately incorporated into the FNN learning, it is possible that FNN can also address the dynamic behavior of the system, as LSTM and RNN do.

With a comprehensive investigation, a FNN with 10 hidden layers and 10 neurons at each layer is selected to estimate longitudinal velocity and lateral velocity, respectively, as shown in Figure 2.

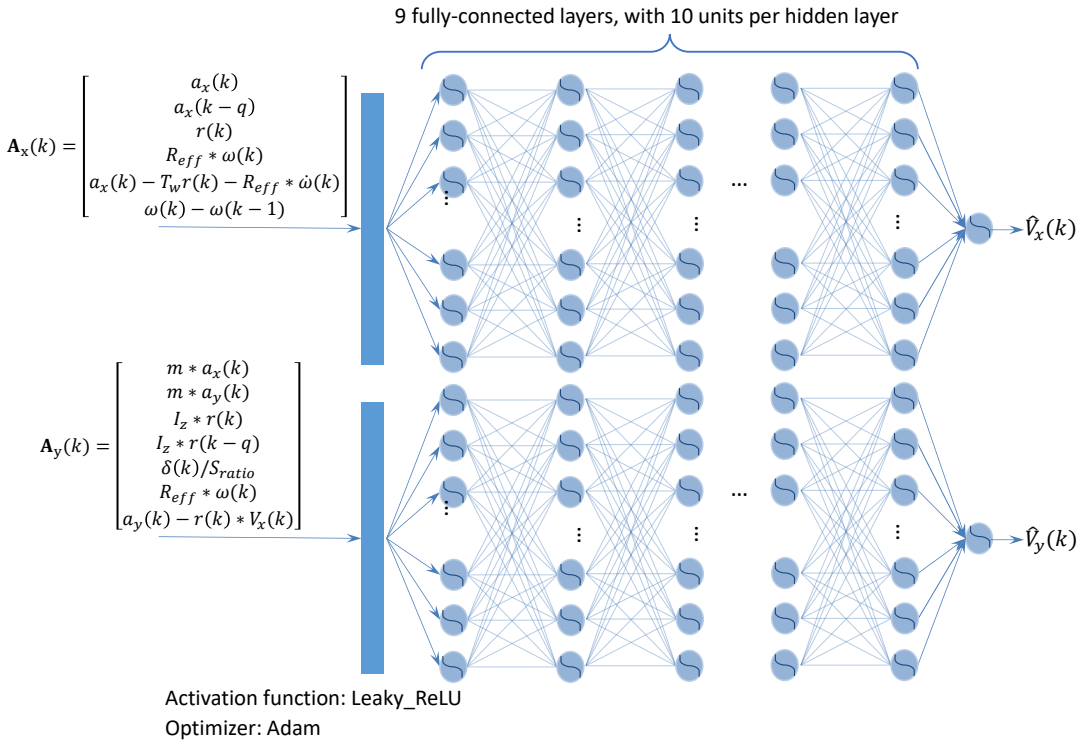

Figure 2. The FNN structure for vehicle longitudinal/lateral velocity estimation.

For training the network, an activation function for each neuron needs to be selected. In view that the inputs in this application can take both positive and negative values, a Leaky rectified linear unit function is considered for the neuron activation function, capable of carrying over both positive and negative inputs, and performing better than the rectified linear unit. The activation function is described as [36]:

$$f(x) = \begin{cases} x, & x > 0 \\ ax, & x \leq 0 \end{cases} \tag{5}$$

where a is the slope for negative values that has been designed by trial and error to increase the performance. The Leaky rectified linear function has a small slope for negative values. By setting/specifying the activation function, inputs, and number of layers and neurons, the training process starts to obtain biases/weights of the neurons. To deal with over-fitting issue and create less complex (parsimonious) model, L2 regularization (Ridge regression), where it drops all features with no significant impact by adding squared magnitude of coefficient as penalty term to the loss function, has been used.

By using the generated weights and biases, the estimated vehicle longitudinal and lateral velocities are calculated as

$$\begin{aligned} \hat{v}_x &= f(\mathbf{b}_{10,1} + \mathbf{w}_{10,1} \sum_{h=1}^{H_9} \mathbf{L}_{9,h}) \\ \hat{v}_y &= f(\mathbf{b}_{10,1} + \mathbf{w}_{10,1} \sum_{h=1}^{H_9} \mathbf{L}_{9,h}) \\ \mathbf{L}_{i,j} &= f(\mathbf{b}_{i,j} + \mathbf{w}_{i,j} \sum_{j=1}^{H_i} \mathbf{L}_{i-1,j}) \end{aligned} \tag{6}$$

where $i \in [1, 2, ..., 9]$, and $\mathbf{H}_{1\times 9} = [10, 10, ..., 10]$.

2.3. *Overall ML estimation scheme*

The overall structure of the proposed FNN vehicle longitudinal and lateral velocity estimation models are shown in Figure 3. As illustrated, vehicle states include the longitudinal acceleration, wheel speeds, and yaw rate measure with the sensors with five previous time steps of the longitudinal acceleration record as a memory signal are fed to the network to estimate the vehicle longitudinal velocity. The longitudinal acceleration temporal signal is added to the input feature for vehicle longitudinal velocity estimation based on the sensitivity analysis results obtained in Section 2. The calculated weights/biases over the training process are used to estimate the vehicle longitudinal velocity.

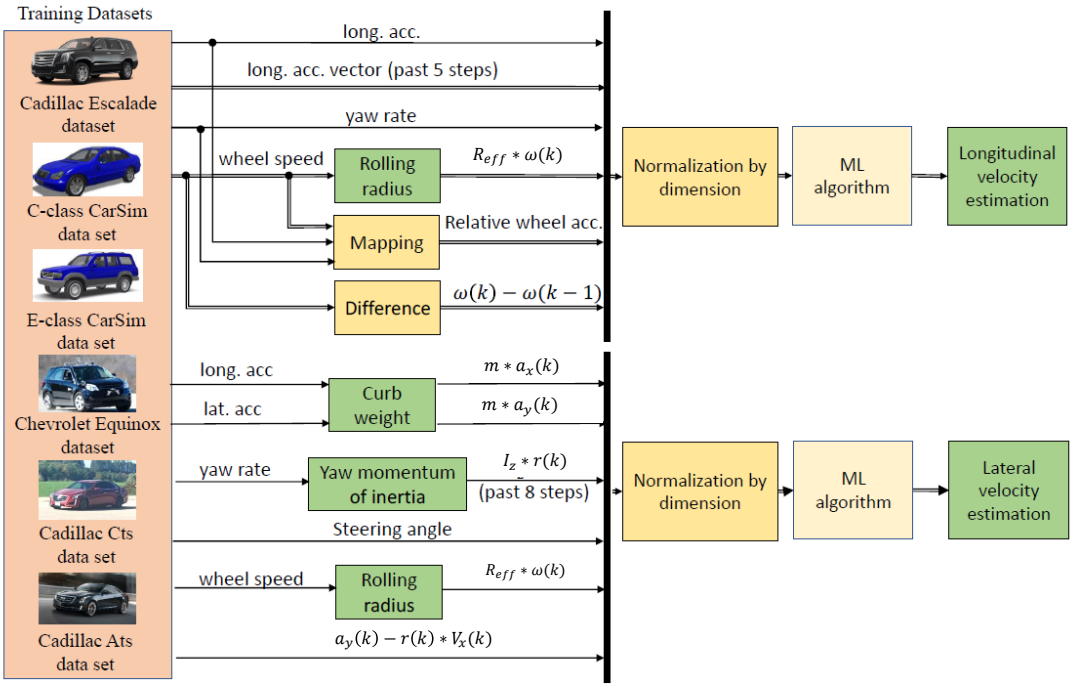

Figure 3. Overall vehicle longitudinal/lateral velocity estimation.

Similarly, using the calculated weights and biases during the training process for lateral velocity estimation, and the selected input vector, the lateral velocity is estimated. For lateral velocity estimation, the longitudinal velocity input is derived from the dedicated FNN trained for v_x prediction. This ensures consistency across the estimation pipeline and avoids reliance on wheel-speed measurements, which may be affected by slip or sensor noise. As Figure 3 illustrates, the input vector contains vehicle states include the longitudinal and lateral accelerations, wheel speeds, steering wheel angle, yaw rate, and longitudinal velocity with eight previous time steps of the yaw rate and lateral accelerations. The feature subset q = 8 includes higher-order dynamics and cross-coupling terms that significantly enhance lateral velocity estimation accuracy. Although subsets q = [2, 4, 6] represent incremental feature groups, q = 8 was highlighted due to its superior performance in both training and validation phases.

To enhance NN training performance and stability the input vector should be normalized to make sure they are on the same scale. To do so, each input of the network is normalized by its maximum value over the entire training dataset to make sure that all the inputs are scaled from -1 to 1. For instance, assume that the tire radius of the vehicle used in the training dataset is R_0, and the tire radius of the target vehicle, is R_1. Then

inputs corresponding to the wheel speed to the well-trained model should be re-calculated as:

$$input_{wheelspeed} = \frac{\omega_{wheel}^{target} R_1}{MAX_{wheel}^{training}}, \tag{7}$$

where the maximum wheel speed (m/s) in training dataset is $MAX_{v_{wheel}}^{training} = max\left(\omega_{wheel}^{training} R_0\right)$. The other input features are also processed by a similar normalization approach. The normalization process scales each feature relative to its physical limits and vehicle-dependent parameters (mass, inertia, tire radius, and steering ratio). This physically grounded scaling ensures that signals across vehicles with different specifications fall within the same numerical range, enabling model reuse without retraining. Furthermore, training data were filtered using a 2nd-order Butterworth low-pass filter (10 Hz cutoff) for synchronization with Global-Positioning-System (GPS) data. Training with unfiltered (raw) data was also tested and resulted in only a 1.2% higher NRMS error but greater sensitivity to noise. The neuron tuning parameters (weights and biases) are calculated by training the FNN with batch size of 203 and a maximum iteration of 1000 where it takes two hours to train (processor specification: Intel Core i7 machine, 32 GB of RAM, and a Nvidia GeForce 670 GPU).

The GPS system used is a dual-antenna RTK-enabled unit capable of providing high-fidelity velocity measurements. While preprocessing steps such as filtering, resampling, and segmenting were applied to ensure data quality for training, we acknowledge that these steps introduce latency. Future work will explore real-time implementations with lightweight preprocessing pipelines suitable for embedded deployment.

3. Dataset utilization

3.1. Main dataset

To train and test the network, both simulation data and experimental data are used. For the simulated data, about 203 maneuvers were generated in CarSim simulation environment from three different vehicle models including E-class SUV, C-class sedan, and Chevy Escalade. Although the dataset consists of 203 distinct maneuvers, each sequence contains hundreds of time steps, yielding over 210,000 data samples across three vehicles. This sample size is sufficient for neural network training and provides coverage across varying vehicle configurations. The simulated data attempt to cover a broad range of driving scenarios under concern. For instance, 15 Single Lane Change (SLC) maneuvers are considered including 5 tests with E-class SUV and 5 tests with C-class sedan, and 5 individual tests with Chevy Escalade. Moreover, at each test, the vehicle initial velocity and operation speed have been changed to collect sufficient data. The total number of tests with details are listed in Table 2 for both simulation and experimental tests.

The experimental data were mainly collected from three test vehicles, including an electric Chevrolet Equinox AWD, a Cadillac CTS and a Cadillac ATS. The specifications of the vehicles are listed in Table 3. The test vehicles were equipped with the required sensors such as IMU, GPS, and wheel speed sensors. The longitudinal/lateral accelerations and yaw rate of the vehicle are measured with a 6-axis IMU (and GPS) system RT2000 on Chevy Equinox and Cadillac CTS. The wheel speeds are measured using the regular ABS wheel speed sensors. Measured signals are communicated using a CAN-bus, and the sampling frequency for the experiment is fixed to 200 Hz.

Table 2. Test scenarios.

Maneuvers	Number of tests	Driven speeds [kph]
Low speed DLC dry road	15	10,20,30,40,45
Low speed DLC icy road	17	10,20,30,40,45,48,52,57
High speed DLC dry road	15	50,55,59,60,70
High speed DLC wet road	11	120,130
Tight DLC dry road	18	20,25,30,35,37,40,42,45,47
Tight DLC wet road	10	85
Tight DLC icy road	13	15,20,25,30,33
Low speed step steer (SS)	25	10,20,30,40,45
High speed step steer	16	60,80,90,100,100,110,120
Low speed sine wave (SW)	14	20,30,40,50
High speed sine wave	16	60,70,80,90,100,110,120
Parking steer	4	-
Roundabout	1	-
Highway merge	5	90,100,110,115,120
Loop road road on flat road	6	10,05,30,40,50
Low speed DLC on road with 17% grade angle	6	10,20,30,40,50,60
High speed DLC on road with 17% grade angle	7	70,80,90,100,110,120,130
Loop road with 17% grade angle	4	15,25,35,45,55

Table 3. Test vehicle specifications.

Parameters	E-class sedan	E-class SUV	Equinox
Vehicle's mass [kg]	1800	2249	2270
Vehicle yaw moment of inertia [kg.m^2]	2587	5657	4605
CG to front axle [m]	1.18	1.4	1.42
CG to rear axle [m]	1.35	1.7	1.44
Front axle length [m]	1.56	1.7	1.62
Effective tire radius [m]	0.34	0.38	0.34

As mentioned, to train/test the network, the input vector needs to be first normalized to have the same magnitude. Moreover, experimental signals are pre-processed (filtered, resampled, and segmented) to make sure that the data are clean enough and have the same sampling rate when being fed into feed the network.

The dataset was sampled at a frequency of 100 Hz, corresponding to a time step of 0.01 seconds. This resolution ensures sufficient temporal granularity for capturing transient vehicle dynamics during both nominal and aggressive maneuvers. As shown in Table 2, different maneuvers have been covered to train/test the

network including SLC, Double Lane Change (DLC), sine shape steer, random steer, Step Steer (SS), and full turn maneuvers. 90% of the collected data are used to train the network (training dataset) while 10% of them are utilized to test the network (testing dataset). The simulation and experimental results of the vehicle longitudinal and lateral velocity estimators are illustrated in next section.

4. Results and discussion

The presented vehicle longitudinal and lateral velocity estimation algorithms are tested and validated by both simulation and experimental studies. Some sample driving scenarios which generally pose challenges for velocity estimation with conventional model-based approaches are particularly selected for the studies.

4.1. Simulation studies

4.1.1. Test results for longitudinal vehicle velocity estimation

To examine the performance of the proposed longitudinal velocity estimator, several maneuvers including sine wave, slalom, and double lane change maneuvers have been considered. The results are illustrated in Figures 4 and 5.

Figure 4 shows a test scenario where the test vehicle E-class SUV was operated with multiple sine wave and slalom driving maneuvers, it can be observed that the estimated longitudinal velocity with the proposed FNN based algorithm matches well with the GPS ground true value. The estimation error in terms of NRMS is only 1.74 % out of the ground truth.

Figure 5 illustrates another challenging scenario where wheel flare occurs from t = 2.9 s to t = 5 s when a harsh and tight sine wave maneuver is conducted. Model based observers generally fail or generate poor estimates under this kind of scenario However, the proposed FNN based algorithm still handles this test scenario very well. The estimation error is only around 1.82% in terms of NRMS.

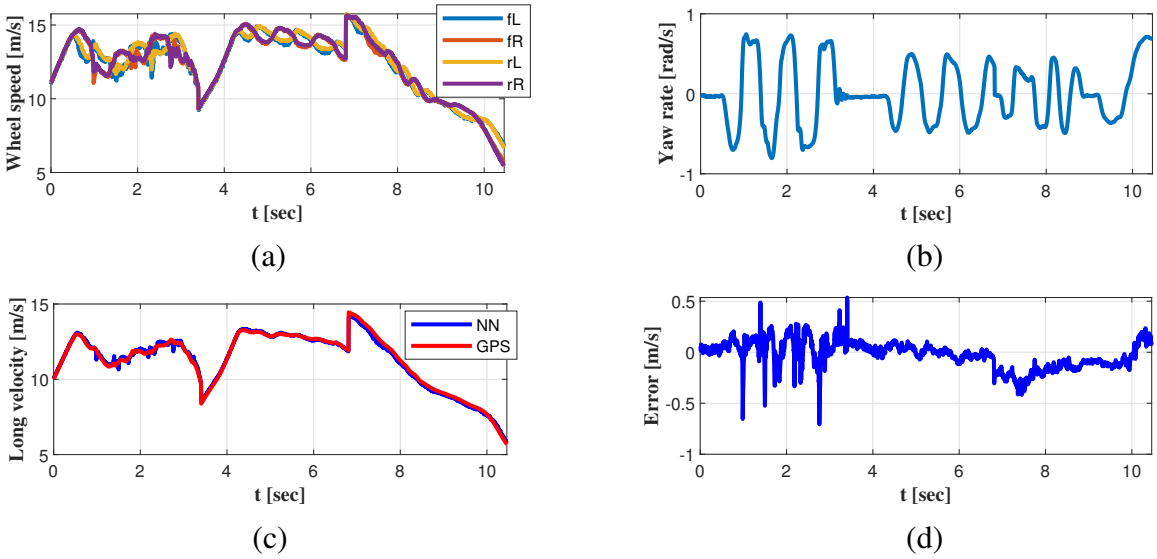

Figure 4. FNN based vehicle longitudinal velocity estimation test results for multiple sine wave and slalom maneuvers: **(a)** Vehicle wheel speeds; **(b)** Yaw rate; **(c)** Longitudinal velocity estimation; **(d)** Estimation error.

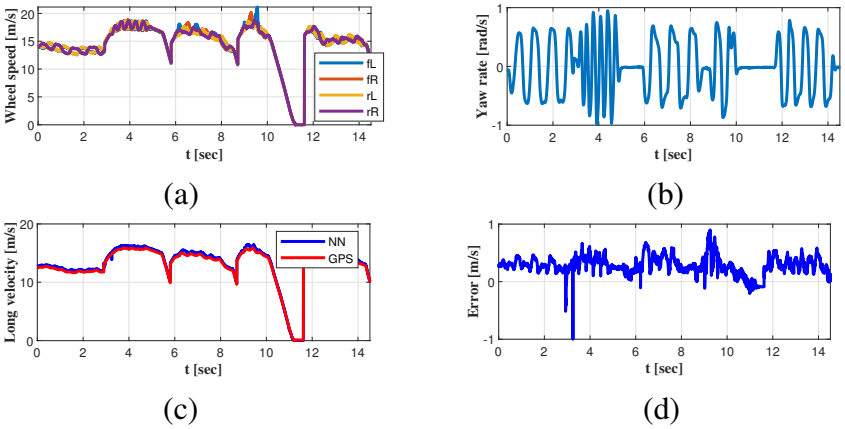

Figure 5. FNN based vehicle longitudinal velocity estimation test results for a double lane change and harsh sine wave maneuvers: **(a)** Vehicle wheel speeds; **(b)** Yaw rate; **(c)** Longitudinal velocity estimation; **(d)** Estimation error.

4.1.2. Test results for lateral vehicle velocity estimation

Given lateral velocity estimation is challenging for combined slip maneuvers for model-based observers, a harsh DLC maneuver with acceleration in turn tests, as shown in Figure 6, is considered to examine the performance of the designed FNN lateral velocity estimator. It can be seen from the FNN results that the accuracy of the design network is acceptable.

It is worth mentioning that in the test results shown in Figure 6, the wheels exhibit differential slip ratios, which presents a challenging estimation scenario. While such conditions are not uncommon, they remain difficult for many model-based approaches due to sensitivity to tire force modeling and parameter uncertainties [37,38]. The test results shown in Figure 6 contain wheels with combined slip conditions (Figure 6a from t = 9 s to t = 11.9 s). As can be seen in the Figure 6d, the NRMS of the lateral velocity estimation error is 5.09 percent out of the actual lateral velocity. Therefore, non-model-based approaches such as ML-based approaches have high potential for vehicle lateral estimation in harsh maneuvers.

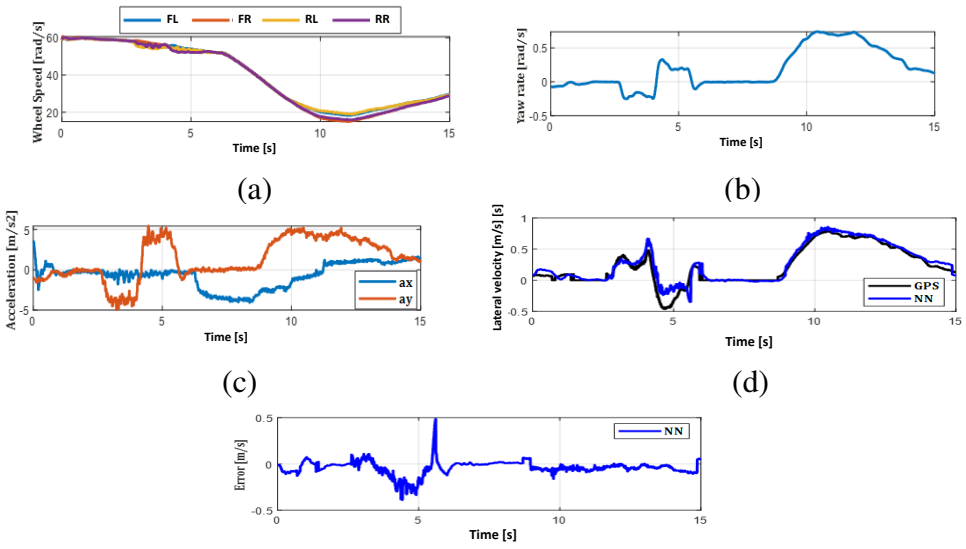

Figure 6. FNN based vehicle lateral velocity estimation test results for a DLC and acceleration in turn maneuvers: **(a)** Vehicle wheel speeds; **(b)** Yaw rate; **(c)** Long/Lateral accelerations; **(d)** Vehicle lateral velocity estimation; **(e)** Estimation error.

4.1.3. Generalization to unseen vehicle

To evaluate the generalizability of the proposed FNN-based estimator, tests were conducted on three unseen vehicles—a mid-size sedan, a compact SUV, and a pickup truck—none of which were included in training. Each vehicle performed a double lane-change maneuver at approximately 60 km/h, inducing significant longitudinal and lateral dynamics. The mean normalized RMSE across the three vehicles was $4.9\% \pm 0.7\%$ for longitudinal velocity and $7.6\% \pm 1.0\%$ for lateral velocity, consistent with results on the training set. Slight steady-state offsets observed in lateral velocity estimates are attributed to small sensor biases and lack of per-vehicle calibration, to be addressed in future work via bias compensation.

4.2. Experimental validation

One of the key challenges of machine learning based algorithms is that they are computationally expensive and sometimes causes issues for real-time implementation. To check the real time implementation capability of the proposed method, the longitudinal and lateral velocity estimation algorithms were implemented on a test vehicle. The test vehicle is an electrified Buick Enclave sport utility vehicle (SUV) fully loaded with tire force sensors, ABS sensor, GPS, Lidar, and Cameras as shown in Figure 7. As the vehicle is electrified, the driving and braking torque on each wheel as well as the wheel speeds at each corner are measurable. Each motor has up to $\pm$ 2100Nm and an ABS module is available on this vehicle. The longitudinal acceleration (Ax), lateral acceleration (Ay), and yaw rate (r) are measured with a 6-axis IMU (and GPS) system RT2000. The overall real-time implementation procedure (how to create the input vectors) are illustrated in Figure 7.

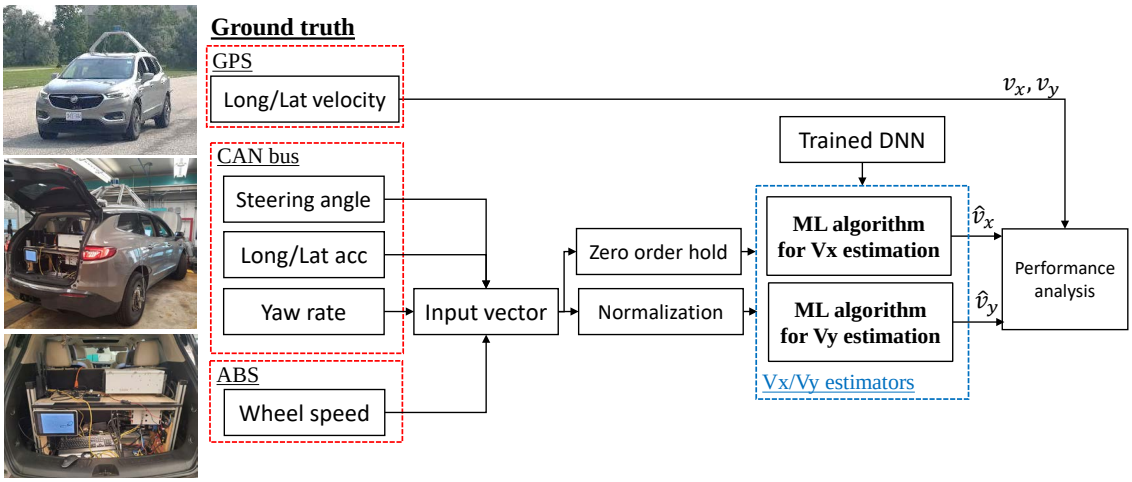

Figure 7. Real time implementation for Vx and Vy estimations.

4.2.1. Test results for longitudinal velocity estimation

A long-looped road test (University of Waterloo Ring Road) was conducted to evaluate the accuracy of the designed FNN. The driver was instructed to drive the vehicle at the maximum allowable speed on the ring road, but due to the presence of five stop signs, the driver had to reduce speed and stop the vehicle several times. The vehicle path, states, and estimated longitudinal velocity for the long-looped road test are shown in Figure 8.

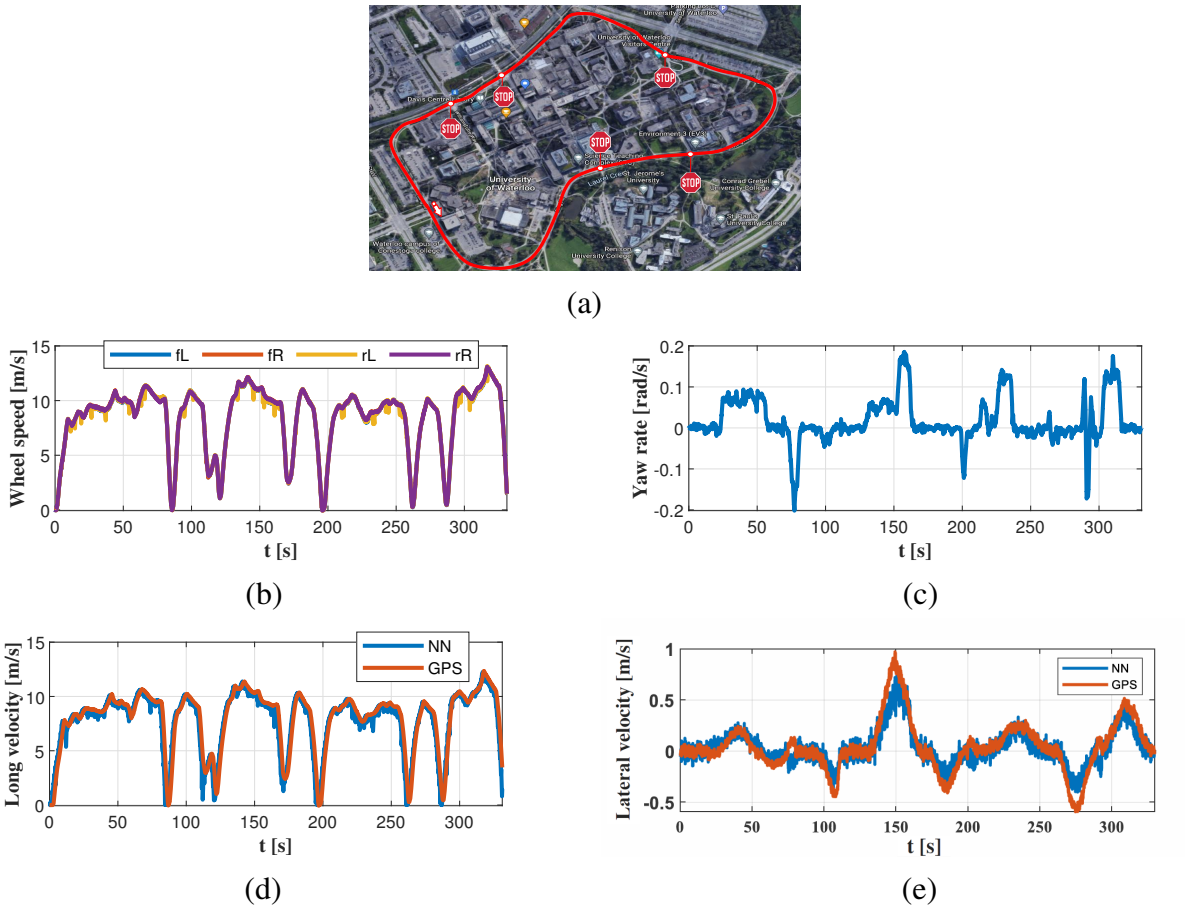

Figure 8. FNN based vehicle longitudinal velocity estimation test results for a long looped road test: **(a)** Vehicle path **(b)** Vehicle wheel speeds; **(c)** Yaw rate; **(d)** Vehicle longitudinal velocity estimation; **(e)** Vehicle lateral velocity estimation-experimental test.

As can be seen in the results shown in Figure 8d, the proposed FNN based longitudinal velocity estimator can accurately estimate the vehicle longitudinal velocity in both longitudinal acceleration and deceleration conditions. Moreover, since the history of the signals are incorporated into the designed algorithm, the sudden sparks in the wheel speed signals (Figure 8c @ t = 14 and t = 35.8) do not cause large error in the estimator.

4.2.2. Test results for lateral vehicle velocity estimation

A rectangle loop maneuver (four step steer) at our test field is considered. The test (shown in Figure 9) was intended to excite the lateral motion of the vehicle as much as possible to push the vehicle into the challenging non-linear zone. In Figure 9, the lateral velocity estimation error between 15 s and 20 s corresponds to a sharp maneuver with high slip ratios. Such conditions are known to challenge both model-based and data-driven estimators [22,23]. During these high-slip scenarios, increased sensor noise and tire saturation contribute to transient estimation spikes. Similar error trends are reported in [20,21], confirming that such discrepancies are characteristic of all data-driven estimators under slip ratios exceeding 15%. Although the sideslip angle during the 15–20 s interval remains below 2.5°, the rapid transient dynamics and tire slip transitions challenge the estimator's precision. Similar behavior has been reported in [22,23], where even advanced neural estimators exhibit localized error spikes under comparable conditions.

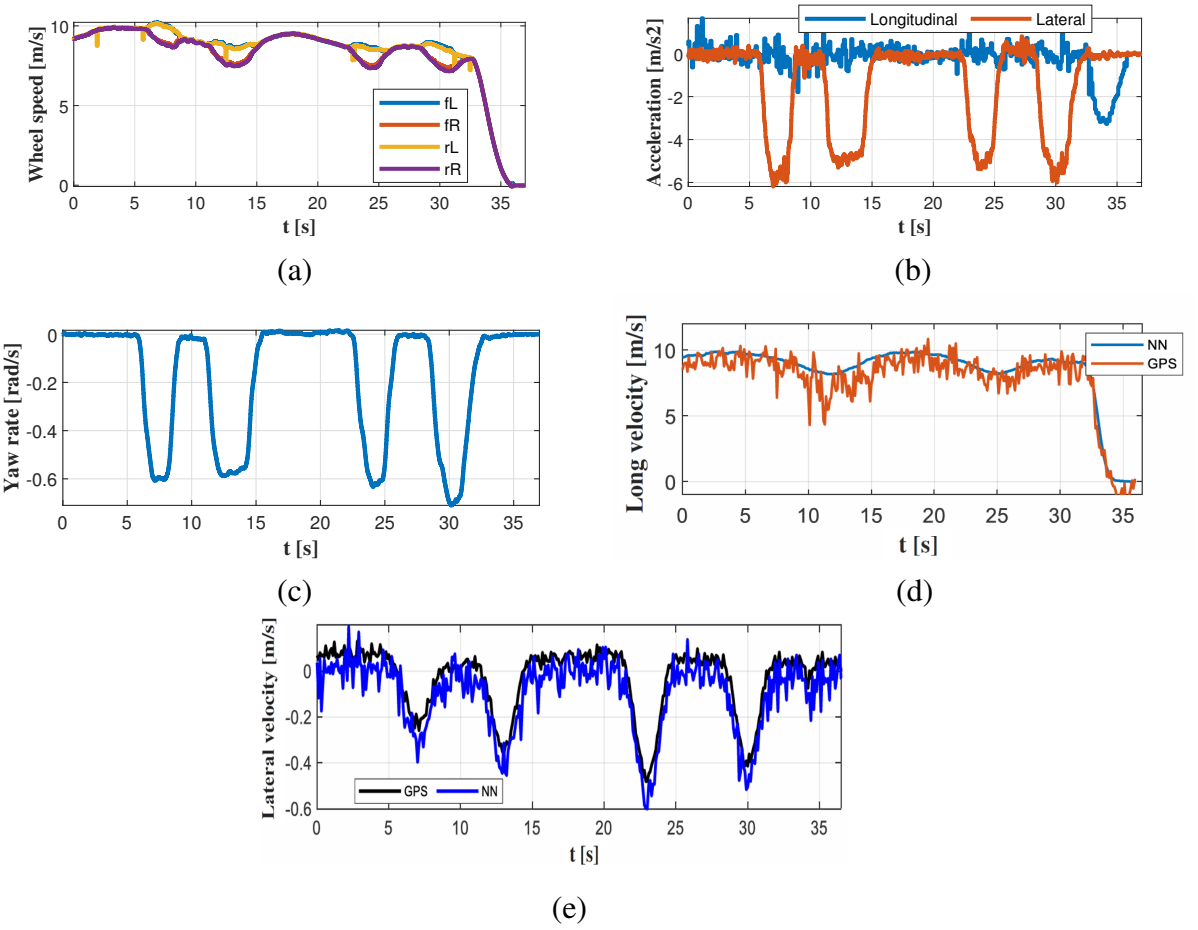

Figure 9. FNN based vehicle lateral velocity estimation test results for a multiple step steer maneuvers: **(a)** Vehicle wheel speeds; **(b)** Long/Lateral acceleration; **(c)** Yaw rate; **(d,e)** Vehicle Long/Lat velocity estimation-experimental test.

As can be seen in Figure 9, the proposed algorithm is repeatable and there are no wind-up errors (error accumulation) in the result, which is usually a concern when historical signals are used in the input vector of the seen neural network. It should be noted that a temporal offset exists between the measured yaw rate and the measured lateral velocity signals on raw data before synchronization. This offset originates from sensor-level timing differences between the IMU-based yaw rate and the GPS-based lateral velocity measurements used as reference, rather than from the proposed estimation method. The estimated lateral velocity remains synchronized with its corresponding measured signal, and the observed offset does not affect training or evaluation consistency.

Remark 2: Given the experimental test results, the FNN approach can run efficiently in real-time on the targeted ECU and has the acceptable accuracy even for the cases that large slip ratio occurs to the driven wheel.

4.3. Comparison with other approaches

In this study, the proposed FNN-based vehicle longitudinal/lateral velocity estimators are also compared with both model-based estimators and RNN based estimators for further verification. For model-based estimation approaches, we implemented the one presented in [35], which demonstrated overall better performance than many other model-based velocity estimators. For RNN based estimators, a ten hidden

layer RNN, with the similar architecture as the proposed FNN approach in Figure 2, is selected, trained, and tested for estimating the vehicle longitudinal and lateral velocities by considering the same input vector as presented in Equations (3) and (4).

To compare the performance of the proposed FNN based with the model-based observer (OBS) and RNN based approaches, an experimental test, involving a DLC and a SS maneuver, was considered. The estimation errors in terms of NRMS [23] for each type of approaches (FNN, OBS, and RNN) are shown in Table 4.

Based on the test results, it can be observed that the FNN method has a much faster error convergence rate compared to the model-based approach. In fact, there is almost no settling time for the FNN method since its output is calculated by multiplying the weights by the inputs. Additionally, the estimation error for the FNN method is almost 1.5% lower than that of the model based estimator. The model-based approach estimation error tends to increase for harsh maneuvers as the linearity assumptions are violated. However, the estimation error for the FNN based estimator will not increase for harsh maneuvers if the training dataset is rich enough and contains adequate data for such scenarios. Comparing the dynamic system model-based with the FNN approach, it can be concluded that the FNN approach improves the accuracy especially in cases where a large slip ratio occurs to the driven wheel, where the model-based approach reports less accuracy due to the utilization of wheel dynamics. It is worth noting that the accuracy is achieved without any additional cost, given the on-board sensor setting considered in the study.

Moreover, to compare the method with other existing ML architectures, As illustrated in Table 4, the RNN and FNN approaches have almost the same scale of estimation errors. Both approaches also have the similar algorithm inference time efficient enough for real time implementation. Similarly, CNN and GRU models performance have been studied given the generated dataset explained in Section II. CNN and GRU models achieve comparable NRMS accuracy but require roughly 35%–50% higher inference latency and parameter count. The proposed FNN provides nearly equivalent estimation precision with simpler architecture and lower computational load, confirming its suitability for real-time ECU deployment.

However, the FNN is much easier to implement in terms of architecture and has relatively less computational cost than RNN. On average, the RNN takes 2.5 hours to train while the FNN takes 2 hours to train. Therefore, the proposed FNN method is recommended for estimating the vehicle longitudinal and lateral velocities.

The vehicle longitudinal and lateral estimation results show that the FNN algorithm is generic, and if the vehicle specifications change, the algorithm can still estimate the vehicle longitudinal and lateral velocities since there were a few test data of the test vehicle (Buick Enclave) include in the training dataset (1.97%—4 out of 203).

To summarize, while both the model-based and ML-based approaches are capable of estimating vehicle longitudinal and lateral velocities accurately, the FNN approach has shown to have a lower error percentage compared to the dynamic system model-based approach. Furthermore, the FNN approach has the potential to further improve estimation accuracy with a more comprehensive dataset that contains additional vehicle configuration information. Therefore, the FNN approach appears to be a more promising solution for estimating vehicle longitudinal and lateral velocities than the model-based approach.

Table 4. NRMS of the errors for the dynamic system model-based and the FNN vehicle longitudinal and lateral velocity estimators.

Estimation Technique	DLC v_x Est [%]	DLC v_y Est [%]	SS v_x Est [%]	SS v_y Est [%]
OBS	2.41	6.54	2.58	7.41
FNN	2.54	5.44	2.43	6.24
RNN	2.91	5.87	2.1	6.01

4.4. Limitations and future work

An ablation study was performed to examine network depth and neuron count. Networks with 4–12 layers and 5–20 neurons per layer were tested; beyond 10×10 configuration, NRMS improvement was marginal ($< 0.3\%$), while smaller networks increased error by 6%–9%. Therefore, the chosen architecture provides an optimal trade-off between accuracy and computational efficiency. While the proposed multi-layer FNN estimators demonstrate strong performance across a range of driving scenarios and vehicle configurations, several limitations remain that warrant further investigation:

- Overfitting risks are currently mitigated using L2 regularization and dropout, yet the complexity of the architecture suggests that further ablation studies are needed. These would help isolate the most critical components of the model and identify opportunities for simplification without sacrificing accurac y. Alternative regularization strategies, such as early stopping or ensemble methods, may also be explored to enhance generalization.
- Sensor noise remains a challenge, particularly in real-world deployments where input signals such as accelerations, yaw rate, and wheel speeds are subject to drift and interference. Although the model performs well under nominal conditions, future work will focus on noise-robust training techniques. This includes data augmentation with synthetic noise, sensor fusion approaches, and the integration of filtering mechanisms like Kalman or particle filters to improve resilience and reliability.
- Hyperparameter choices—including the use of 10 hidden layers, Leaky ReLU activations, and fixed neuron counts—were selected empirically. While effective in initial trials, this approach may not yield optimal configurations across all vehicle types and driving conditions. A more systematic optimization strategy using grid search, Bayesian methods, or evolutionary algorithms will be pursued to fine-tune the architecture for both performance and efficiency.
- The current FNN architecture does not explicitly model temporal dependencies, relying instead on embedded historical inputs to capture short-term dynamics. This may limit its ability to learn long-range temporal patterns, especially in maneuvers involving delayed responses or cumulative effects. Hybrid models that combine FNNs with recurrent structures such as RNNs or LSTMs will be explored to better capture sequential behavior while maintaining computational feasibility.
- Although the normalization strategy enables generalization across multiple vehicle platforms, the training dataset may not fully represent edge cases such as off-road conditions, extreme weather, or highly customized vehicle configurations. Expanding the dataset to include such scenarios and validating the model's robustness under these conditions will be essential for broader applicability.

- Real-time deployment on production-grade embedded systems introduces constraints related to processing power and memory. While the current estimators have been validated on high-performance platforms, further work is needed to compress and optimize the models for real-time inference. Techniques such as pruning, quantization, and knowledge distillation will be considered to ensure compatibility with automotive-grade hardware.

5. Conclusion

This paper presented two generic FNN based vehicle velocity estimators to estimate longitudinal and lateral velocities, respectively. The generic character of the estimators is achieved by ingeniously incorporating vehicle parameters into the NN input feature formulation, which makes it possible for the proposed velocity estimators to be widely applied to vehicles of different specifications without the need of re-training the NNs. Simulation and experimental results demonstrate that the proposed FNN algorithms can provide accurate estimation in various challenging scenarios and are robust to the changes of vehicle specifications. In the future work, the database will be further enriched with more driving scenarios. With a comprehensive dataset, the proposed algorithms should be able to achieve higher estimation accuracy.

Data availability statement

No supplementary or additional data were generated in this study.

Declaration of generative AI and AI-assisted technologies

The authors did not use generative AI or AI-assisted technologies in the writing of this manuscript.

Acknowledgments

The authors would like to acknowledge the financial support of Ontario Research Fund (Funder ID: 10.13039/100012171), Natural Sciences and Engineering Research Council (NSERC) of Canada (Funder ID: 10.13039/501100000038), and technical and financial support of General Motors.

Authors' contribution

Conceptualization, Amin Habibnejad Korayem, Amir Khajepour, Ehsan Hashemi and Qingrong Zhao; methodology, Amin Habibnejad Korayem; software, Amin Habibnejad Korayem; validation, Amin Habibnejad Korayem, Amir Khajepour and Ehsan Hashemi; resources, Amir Khajepour and Alireza Kasaiezadeh; data curation, Amin Habibnejad Korayem; writing—original draft preparation, Amin Habibnejad Korayem; writing—review and editing, Amir Khajepour, Ehsan Hashemi, Qingrong Zhao and Alireza Kasaiezadeh; supervision, Amir Khajepour and Qingrong Zhao; funding acquisition, Amir Khajepour and Alireza Kasaiezadeh. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Wielitzka M, Busch A, Dagen M, Ortmaier T, Serra G. Unscented Kalman filter for state and parameter estimation in vehicle dynamics. In *Kalman Filters-Theory for Advanced Applications*, 1st ed. London: InTech, 2018. pp. 56–75.

[2] Korayem AH, Pazooki A, Durali L, Khajepour A, Fidan B, *et al.* Hitch angle estimation of a towing vehicle with arbitrary configuration. *IEEE Trans. Intell. Transp. Syst.* 2021, 23(7):7535–7546.

[3] Wenzel TA, Burnham K, Blundell M, Williams R. Dual extended Kalman filter for vehicle state and parameter estimation. *Veh. Syst. Dyn.* 2006, 44(2):153–171.

[4] Park G, Choi SB, Hyun D, Lee J. Integrated observer approach using in-vehicle sensors and GPS for vehicle state estimation. *Mechatronics* 2018, 50:134–147.

[5] Imsland L, Johansen TA, Fossen TI, Grip HF, Kalkkuhl JC, *et al.* Vehicle velocity estimation using nonlinear observers. *Automatica* 2006, 42(12):2091–2103.

[6] Zhang X, Xu Y, Pan M, Ren F. A vehicle ABS adaptive sliding-mode control algorithm based on the vehicle velocity estimation and tyre/road friction coefficient estimations. *Veh. Syst. Dyn.* 2014, 52(4):475–503.

[7] Selmanaj D, Corno M, Panzani G, Savaresi SM. Vehicle sideslip estimation: a kinematic based approach. *Control Eng. Pract.* 2017, 67:1–12.

[8] Righetti G, Lenzo B. A Combined dynamic—kinematic extended Kalman filter for estimating vehicle sideslip angle. *Appl. Sci.* 2025, 15(3):1365.

[9] Ngwangwa HM, Heyns PS, Labuschagne F, Kululanga GK. Reconstruction of road defects and road roughness classification using vehicle responses with artificial neural networks simulation. *J. Terramech.* 2010, 47(2):97–111.

[10] Bridgelall R. Connected vehicle approach for pavement roughness evaluation. *J. Infrastruct. Syst.* 2014, 20(1):04013001.

[11] Nitsche P, Stütz R, Kammer M, Maurer P. Comparison of machine learning methods for evaluating pavement roughness based on vehicle response. *J. Comput. Civ. Eng.* 2014, 28(4):04014015.

[12] Alghooneh M, Moore JK, Mettrick CJ, Kakert GP, Khajepour A, *et al.* Systems and methods for estimating tire wear. U.S. Patent No. 11,945,265 B2. 26 Mar. 2024.

[13] Korayem AH, Khajepour A, Fidan B. Road angle estimation for a vehicle-trailer with machine learning and system model-based approaches. *Veh. Syst. Dyn.* 2022, 60(10):3583–3604.

[14] Chu Z, Zhu D, Yang SX. Observer-based adaptive neural network trajectory tracking control for remotely operated vehicle. *IEEE Trans. Neural Netw. Learn. Syst.* 2016, 28(7):1633–1645.

[15] Taghavifar H. Neural network autoregressive with exogenous input assisted multi-constraint nonlinear predictive control of autonomous vehicles. *IEEE Trans. Veh. Technol.* 2019, 68(7):6293–6304.

[16] Guo J, Kim S, Wymeersch H, Saad W, Chen W. Guest editorial: introduction to the special section on machine learning-based internet of vehicles: theory, methodology, and applications. *IEEE Trans. Veh. Technol.* 2019, 68(5):4105–4109.

[17] Schoen A, Byerly A, Hendrix B, Bagwe RM, dos Santos EC, *et al.* A machine learning model for average fuel consumption in heavy vehicles. *IEEE Trans. Veh. Technol.* 2019, 68(7):6343–6351.

[18] Qin Y, Xiang C, Wang Z, Dong M. Road excitation classification for semi-active suspension system based on system response. *J. Vib. Control* 2018, 24(13):2732–2748.

[19] Wang W, Bei S, Zhang L, Zhu K, Wang Y, *et al.* Vehicle sideslip angle estimation based on general regression neural network. *Math. Probl. Eng.* 2016, 2016(1):3107910.

[20] Gräber T, Lupberger S, Unterreiner M, Schramm D. A hybrid approach to side-slip angle estimation with recurrent neural networks and kinematic vehicle models. *IEEE Trans. Intell. Veh.* 2018, 4(1):39–47.

[21] Melzi S, Sabbioni E. On the vehicle sideslip angle estimation through neural networks: numerical and experimental results. *Mech. Syst. Signal Process.* 2011, 25(6):2005–2019.

[22] Giuliacci TA, Ballesio S, Fainello M, Mair U, King J. Recurrent neural network model for on-board estimation of the side-slip angle in a four-wheel drive and steering vehicle. *SAE Int. J. Passeng. Veh. Syst.* 2023, 17(15-17-01-0003):37–48.

[23] Marotta R, Strano S, Terzo M, Tordela C. On the prediction of the sideslip angle using dynamic neural networks. *IEEE Open J. Intell. Transp. Syst.* 2024, 5:281–295.

[24] Chindamo D, Lenzo B, Gadola M. On the vehicle sideslip angle estimation: a literature review of methods, models, and innovations. *Appl. Sci.* 2018, 8(3):355.

[25] Garcia Guzman J, Prieto Gonzalez L, Pajares Redondo J, Montalvo Martinez MM, López Boada MJ. Real-time vehicle roll angle estimation based on neural networks in IoT low-cost devices. *Sensors* 2018, 18(7):2188.

[26] Chen X, Lu Y, Khajepour A, Cao D, Sun C. Bridging domain gaps in CNNs: a comprehensive approach with adaptation and randomization strategies. *Artif. Intell. Auton. Syst.* 2024, 1(2).

[27] Zhang Y, Zhang Y, Ai Z, Feng Y, Zhang J, *et al.* Estimation of electric mining haul trucks' mass and road slope using dual level reinforcement estimator. *IEEE Trans. Veh. Technol.* 2019, 68(11):10627–10638.

[28] Lv C, Xing Y, Lu C, Liu Y, Guo H, *et al.* Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle. *IEEE Trans. Veh. Technol.* 2018, 67(7):5718–5729.

[29] Srinivasan S, Sa I, Zyner A, Reijgwart V, Valls MI, *et al.* End-to-end velocity estimation for autonomous racing. *IEEE Robot. Autom. Lett.* 2020, 5(4):6869–6875.

[30] Dong G, Che G, Tian M, Zhao H, Gao B. Vehicle state estimation based on recurrent neural network. In *2021 40th Chinese Control Conference (CCC)*, Shanghai, China, July 26–28, 2021, pp. 2669–2673.

[31] Kong D, Wen W, Zhao R, Lv Z, Liu K, *et al.* Vehicle lateral velocity estimation based on long short-term memory network. *World Electr. Veh. J.* 2022, 13(1):1.

[32] Korayem AH, Khajepour A, Fidan B. A review on vehicle-trailer state and parameter estimation. *IEEE Trans. Intell. Transp. Syst.* 2021, 23(7):5993–6010.

[33] Khajepour A, Korayem AH, Hashemi E, Zhao Q, Mahabadi SK, *et al.* Universal machine learning based system for estimating a vehicle state. U.S. Patent No. 12,084,064 B2. 10 Sep. 2024.

[34] Korayem AH, Khajepour A, Fidan B. Trailer mass estimation using system model-based and machine learning approaches. *IEEE Trans. Veh. Technol.* 2020, 69(11):12536–12546.

[35] Hashemi E, Kasaiezadeh A, Khosravani S, Khajepour A, Moshchuk N, *et al.* Estimation of longitudinal speed robust to road conditions for ground vehicles. *Veh. Syst. Dyn.* 2016, 54(8):1120–1146.

[36] Bhardwaj A, Di W, Wei J. *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Birmingham: Packt Publishing Ltd., 2018.

[37] Rajamani R. *Vehicle Dynamics and Control*, 2nd ed. New York: Springer, 2012.

[38] Goh J, Wang D, Lee T, Tan H. Robust estimation of tire forces and slip ratios. *IEEE Trans. Veh. Technol.* 2019, 68(2):1152–1164.