Taylor & Francis
Taylor & Francis Group

Vehicle System Dynamics
International Journal of Vehicle Mechanics and Mobility

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/nvsd20

Maneuver-based deep learning parameter identification of vehicle suspensions subjected to performance degradation

Yongjun Pan, Yu Sun, Chuan Min, Zhixiong Li & Paolo Gardoni

To cite this article: Yongjun Pan, Yu Sun, Chuan Min, Zhixiong Li & Paolo Gardoni (2022): Maneuver-based deep learning parameter identification of vehicle suspensions subjected to performance degradation, Vehicle System Dynamics, DOI: 10.1080/00423114.2022.2084424

To link to this article: https://doi.org/10.1080/00423114.2022.2084424

VEHICLE SYSTEM DYNAMICS
https://doi.org/10.1080/00423114.2022.2084424

Maneuver-based deep learning parameter identification of vehicle suspensions subjected to performance degradation

Yongjun Pan[a], Yu Sun[a], Chuan Min[a], Zhixiong Li[b] and Paolo Gardoni[c]

[a]College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, People's Republic of China; [b]Faculty of Mechanical Engineering, Opole University of Technology, Opole, Poland; [c]Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, USA

ABSTRACT

A novel parameter identification method was proposed for vehicle suspensions subjected to performance degradation. The proposed method does not require the measurement of the stiffness and damping coefficients of suspensions. Instead, it uses vehicle states to calculate the stiffness and damping coefficients based on an efficient multibody model and inverse dynamics. First, a full-vehicle system was modelled using a semirecursive multibody formulation, and the dynamic properties of suspensions, chassis frame, and tires were considered. Second, dynamic simulations on a bumpy road were performed, and vehicle state data were collected. A deep neural network (DNN) model, whose inputs and outputs were vehicle states and suspension parameters, was developed. The DNN model can estimate the stiffness and damping coefficients based on vehicle states measured by sensor networks. The parameter identification was achieved by deep learning of the relationship between vehicle states and suspension parameters in a given maneuver. Finally, the model accuracy was investigated in terms of different DNN inputs, data samples, and hidden layers. The results showed that the DNN model predicts accurate stiffness and damping coefficients in real time. This maneuver-based parameter identification method can be used for the condition-based monitoring or fault diagnosis of vehicle suspensions subjected to performance degradation.

ARTICLE HISTORY
Received 7 December 2021
Revised 5 April 2022
Accepted 16 April 2022

KEYWORDS
Vehicle suspension; parameter identification; dynamic simulation; deep neural networks; multibody model

1. Introduction

Vehicle suspensions play a crucial role in vehicle systems in terms of ride comfort and vehicle handling. The performance of vehicle suspensions is determined by key parameters, such as spring stiffness and damping coefficients. Accurate information about suspension performance is critical to achieving advanced chassis control [1,2]. Due to worsening road environments, the suspension performance degrades over time. Therefore, estimation of suspension parameters has attracted interest using both signal- and model-based methods [3–6].

Signal-based estimation methods were developed to directly analyse the data obtained from sensor networks. They do not require complex dynamic modelling of suspensions

CONTACT Yongjun Pan yongjun.pan@cqu.edu.cn

or a full vehicle and the knowledge of the excitation inputs of the road. These methods make relative comparisons between measured signals and hence lessen the effects of parameter uncertainties that influence outputs. Thus, simple and effective algorithms must be designed to analyse the signal data for accurate estimation of suspension parameters [7–9]. However, because of high costs and other associated impracticalities, the suspension parameters of a running vehicle are difficult or sometimes impossible to measure directly within the desired accuracy level [10].

To overcome these issues, model-based estimation methods were developed based on the use of mathematical models to analyse the relationship between vehicle states and suspension parameters. Sensor-collected signals were used as inputs for the models to estimate suspension parameters. The estimated outputs were compared with the reference outputs for accuracy analysis and design optimisation [11,12]. Widely used model-based methods include the inverse modelling method and the Kalman filter-based method (e.g. extended Kalman filter, adaptive Kalman filter, unscented Kalman filter, and Rao-Blackwellised particle filter) [13–16]. Because the relationship between vehicle states and suspension parameters is complex and highly nonlinear, it is difficult to formulate dynamic equations to accurately describe this nonlinear relationship. To accurately estimate suspension parameters, complicated models, algorithms, or techniques are required, which can be time-consuming in integration and tuning. The examples in the literature rely on high computational power or low sampling rates, especially for online applications [17].

To estimate suspension parameters in real time, we proposed a maneuver-based deep learning parameter identification method. This method uses an efficient semirecursive multibody model, inverse modelling approach, and deep neural networks (DNNs). The multibody model enables accurate reconstruction of the time-domain signals of suspensions, providing a promising alternative to signal-based methods [18,19]. In this work, an efficient vehicle multibody model was developed based on a sophisticated semirecursive multibody formulation. Unlike other vehicle models (e.g. approximate model, surrogate model, and nonlinear dynamics model), the multibody model considers the dynamic properties of each component and uses Pacejka's tire forces. This enables accurate dynamic modelling of the entire system [20–22]. Subsequently, the multibody dynamic simulations were performed on a bumpy road to collect the data of vehicle states for DNN modelling.

DNN modelling is used for the identification of vehicle suspension parameters [23–25]. It not only reduces the effects of noise or uncertainties but also avoids the numerical instability problem. Another key feature of DNN methods is that they are capable of massive parallel computations [26,27]. Vehicle states such as the position of the vehicle centroid, the vertical velocity and acceleration, and the pitch angle are used as inputs for DNN training. The stiffness and damping coefficients of the front and rear suspensions are used as DNN outputs. The identification of suspension parameters is achieved by deep learning of the relationship between vehicle states and suspension parameters. The DNN model is developed offline and used online for the estimation of suspension parameters. Given the vehicle states measured by sensor networks, the stiffness and damping coefficients of vehicle suspensions can be accurately estimated. As a result, the suspension performance degradation due to road roughness and mechanical fatigue can be evaluated for enhanced ride performance and vehicle handling. The highlights of this work are summarised as follows:

- We presented a full-vehicle model, including the dynamic properties of suspensions, based on a semirecursive multibody formulation for dynamic simulations.
- We developed a DNN modelling method for estimating the stiffness and damping coefficients of vehicle suspensions via vehicle states.
- The DNN model was investigated and verified with different DNN inputs, data samples, and hidden layers.

The rest of this work is organised as follows. In Section 2, an efficient semirecursive multibody formulation is introduced for the dynamic modelling of a full vehicle. In Section 3, the vehicle multibody simulations are performed, and the dynamics data are acquired for DNN modelling. In Section 4, the DNN modelling method is proposed to estimate suspension parameters using vehicle states. In Section 5, the DNN results are discussed in detail, and the model accuracy is analysed in terms of error functions. In Section 6, the capacity of DNNs to estimate suspension parameters is studied with different DNN inputs, data samples, and hidden layers. In Section 7, the conclusions of the work are provided.

2. Vehicle dynamics model

We used a semirecursive multibody method to model vehicle system dynamics. The semirecursive multibody approach involves two-step velocity transformation and rod-removal technique [28,29]. First, the dynamic equations of the vehicle multibody system were formulated via Cartesian coordinates. Thus, the first velocity transformation was used to express the Cartesian coordinates in terms of relative coordinates. The equations of motion of an open-loop vehicle system can be formulated by applying the method of virtual power and using the first velocity transformation as [30,31]

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\mathbf{T}\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\left(\bar{\mathbf{Q}} - \bar{\mathbf{P}}\right) \tag{1}$$

or in compact notation as

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left(\mathbf{Q}^{\Sigma} - \mathbf{P}^{\Sigma}\right) \tag{2}$$

where, $\ddot{\mathbf{z}}$ represents the relative accelerations of the vehicle system. The matrices $\mathbf{R}_{\mathrm{d}}$ and $\mathbf{T}$, respectively, represent the first velocity transformation matrix and path matrix. The path matrix describes system connectivity and tree-topology. The matrices $\bar{\mathbf{M}}$, $\bar{\mathbf{P}}$ and $\bar{\mathbf{Q}}$, respectively, represent the global mass matrix, velocity-dependent inertia forces, and external forces. Matrices $\mathbf{M}^{\Sigma}$, $\mathbf{P}^{\Sigma}$ and $\mathbf{Q}^{\Sigma}$, respectively, represent the corresponding accumulated terms. By using the system tree-topology, tire and suspension forces were computed and accumulated from the leafs to the ground. This process is very efficient and can be achieved by programming techniques easily.

Let us consider a closed-loop vehicle system, and kinematic joints or slender rods that need to be cut or eliminated to make full use of the system-tree topology. To model the dynamics of the closed-loop vehicle system, the loop-closure constraint equations need to be taken into account based on Equation (2). If slender rods are eliminated to open the closed-loops, the rod-related inertia and external forces, e.g. tire forces, spring and damping forces, must be calculated and added to the neighbouring bodies. When the forces are transmitted to the tree structure, the compliance and related errors of the cut-joints and

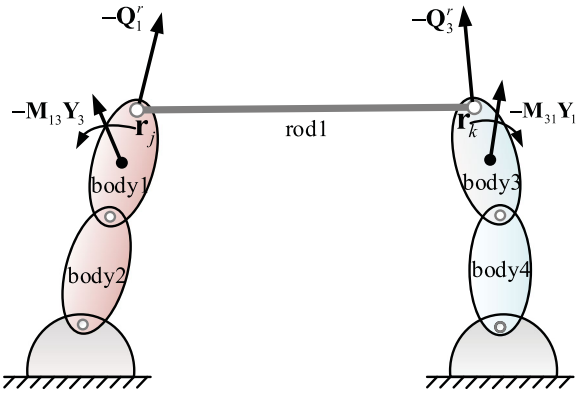


Figure 1. Rod-removal technique.

removed-rods are not considered in this work. This rod-removal technique is described using Figure 1. It contributes to the high efficiency of the semirecursive multibody method.

Subsequently, the second velocity transformation was introduced based on the constraint equations to express the relative coordinates in terms of independent relative coordinates. By considering the loop-closure constraint equations and introducing the second velocity transformation matrix, the equations of motion of the closed-loop multibody system can be derived as follows [32]:

$$\mathbf{R}_{\mathbf{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathbf{z}}\ddot{\mathbf{z}}^{\mathrm{i}} = \mathbf{R}_{\mathbf{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left[\mathbf{Q}^{\Sigma} - \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\frac{\mathrm{d}\left(\mathbf{T}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathbf{z}}\right)}{\mathrm{d}t}\dot{\mathbf{z}}^{\mathrm{i}}\right] \tag{3}$$

where, $\dot{\mathbf{z}}^{\mathrm{i}}$ and $\ddot{\mathbf{z}}^{\mathrm{i}}$, respectively, represent the independent relative velocities and accelerations of the vehicle system. Matrix $\mathbf{R}_{\mathbf{z}}$ represents the second velocity transformation matrix. It is a basis of the constraints Jacobian null space.

The expression in Equation (3) is more complicated than the multibody formulations used in commercial software packages, e.g. ADAMS or RecurDyn. It uses a small set of independent relative accelerations $\ddot{\mathbf{z}}^{\mathrm{i}}$, leading to high computational efficiency. In addition, the ordinary differential form of Equation (3) enables the stable simulation of vehicle systems with high-order numerical integration schemes. The Adams-Bashforth-Moulton and 4th-order Runge-Kutta methods were widely used in real-time simulations with longer simulation times [33,34]. This multibody modelling method considers the dynamic properties of each component, e.g. suspensions, chassis, and tires. It is valid for a range of driving conditions and road environments.

The investigated vehicle system consists of MacPherson strut suspensions in the front axles and multilink suspensions in the rear axles. Additional information about this vehicle system is shown in Table 1. The vehicle system was modelled based on the semirecursive multibody method, where the joint-cut and rod-removal techniques were applied, and independent relative coordinates were used. To build the system tree-topology, the universal, spherical, and free joints were broken down into several auxiliary massless bodies and a number of one-degree-of-freedom revolute and prismatic joints. For example, the free joint between the ground and the chassis frame was replaced using five auxiliary massless bodies with three prismatic joints and three revolute joints. To open the closed-loops, two

Table 1. Parameters of the vehicle system.

Parameters	Value	Parameters	Value
Vehicle mass	1155 kg	Tire rolling radius	0.4673 m
Distance from centroid to front axle	0.7209 m	Centroid height	0.5373 m
Distance from centroid to rear axle	2.0791 m	Wheelbase	2.8 m
Stiffness of front suspensions	40,000 N/m	Damping of front suspensions	1980 Ns/m
Stiffness of rear suspensions	35,000 N/m	Damping of rear suspensions	1980 Ns/m

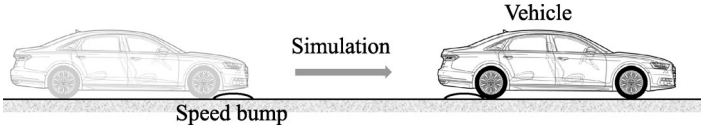


Figure 2. Bumpy road for dynamic simulations.

spherical joints and twelve slender rods were cut and removed, respectively. As a result, 18 loop-closure constraint equations were generated. Finally, the semirecursive multibody model of the vehicle was provided.

3. Vehicle simulation and data processing

The dynamic simulations on a bumpy road were performed, and the data of vehicle states (e.g. the position of the vehicle centroid, the vertical velocity and acceleration, and the pitch angle) were collected. The simulation time was 3 s, the initial velocity was 20 m/s, and a 500 Nm torque was imposed on the front wheels. The bumpy road environment for dynamic simulations is described in Figure 2. We used speed bumps to imitate the bumpy road. The speed bumps were usually 30–60 mm in height and 300–600 mm in width. The arc-shaped bumps with a height of 50 or 60 mm have better speed-control effects [35]. The shape of the speed bump is described in Figure 3.

The multibody model described in this work was developed based on a real vehicle. Thus, the suspension parameters, namely stiffness and damping coefficients in the front and rear, were obtained from the measured data of the vehicle. We decreased the suspension parameters to imitate performance degradation to perform dynamic simulations for acquiring vehicle states. The stiffness of front suspensions was varied from its original value (40,000 N/m) to 35,500 N/m. The stiffness of rear suspensions was varied from its original value (35,000 N/m) to 30,500 N/m. The damping of front and rear suspensions was varied from their original value (1980 Ns/m) to 1800 Ns/m. The stiffness and damping coefficients decreased at intervals of −500 N/m and −20 Ns/m, respectively. Note that the variation range of suspension parameters can be adjusted according to real constraints to reasonably imitate performance degradation. The DNN modelling procedure for parameter estimation is still useful and effective.

To acquire vehicle states, we combined different parameters of the front and rear suspensions and performed a large number of dynamic simulations on the bumpy road. The vehicle states at the time when the rear wheels passed through the speed bump were captured. This means that a set of suspension parameters corresponds to a set of vehicle states. The vehicle states at this specific time are obvious, which is beneficial to DNN modelling. In this way, it is unnecessary to measure the vehicle states over a continuous period for

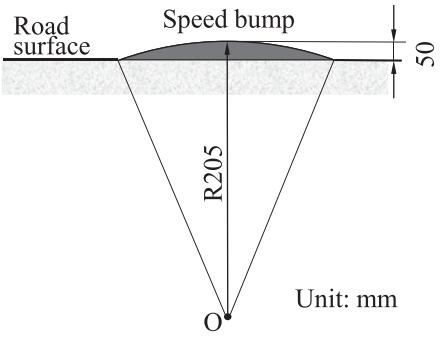


Figure 3. Speed bump shape.

DNN modelling. The data were divided into 10 layers, and the hierarchical fixed proportion method was used to determine the number of samples in each layer. In this work, 2000, 1500, and 1000 sets of samples were selected for DNN modelling. We used 70%, 15% and 15% of samples for training, verification, and testing, respectively, which were randomly selected from all samples. All sample data were normalised during data processing. Therefore, the original data were transformed into nondimensional index values. This prevents higher values from weakening the effects of lower values. It is critical to balance the contribution of each feature. In addition, normalisation speeds up the gradient descent to find optimal solutions. In this section, min-max normalisation was used for data processing as follows:

$$x_i^* = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{4}$$

where, x_i^* denotes the normalised value, which is used as the DNN input, x_i denotes the original value of the ith initial array, and min() and max() denote the minimum and maximum values in the sample array, respectively. Min-max normalisation is a linear transformation of the original data, which restricts the result to a value between 0 and 1.

4. Parameter identification via DNNs

In this section, we presented a DNN modelling method with multiple hidden layers. The input and output layers corresponding to vehicle states and suspension parameters both have four neurons. The four vehicle states are the position of the vehicle centroid (V_p), the vertical velocity (V_v) and acceleration (V_a), and the pitch angle (V_θ). The four suspension parameters include the stiffness and damping coefficients of front and rear suspensions (F_k, F_c, R_k, R_c). The DNN structures with 3, 4, and 5 hidden layers were introduced. We also used deeper DNN experiments. The deeper DNNs not only require longer training time but also reduce the accuracy of suspension parameter prediction and even expand the prediction error because the DNN with multiple hidden layers has a stronger ability to reflect the nonlinear relationship between inputs and outputs in theory; however, too many hidden layers bring overfitting problems, affecting the prediction accuracy [36]. The number of neurons in the hidden layers was determined using empirical formulas and a large number of experiments [37]. When the training samples are determined by adjusting the

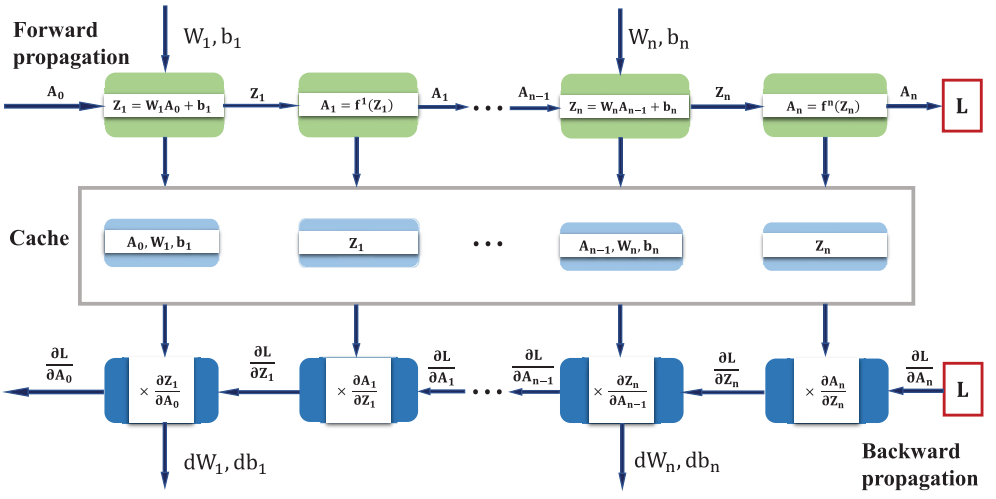


Figure 4. Forward and backward propagation of DNNs.

neurons of each hidden layer via random experiments, the optimal number of hidden layer neurons can be selected according to the network training error. The learning process of DNNs includes the forward propagation of the data from the input layer and the backward propagation of the errors from the output layer. This learning process can be described in Figure 4.

The input, weight, and bias matrices need to be defined in forward propagation. They can be described as

$$\mathbf{Z}_0 = (\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3, \ldots \mathbf{i}_m) \tag{5}$$

$$\mathbf{W}_n = \left(\mathbf{w}_{n_1}, \mathbf{w}_{n_2}, \mathbf{w}_{n_3}, \ldots \mathbf{w}_{n_m}\right) \tag{6}$$

$$\mathbf{B}_n = \left(\mathbf{b}_{n_1}, \mathbf{b}_{n_2}, \mathbf{b}_{n_3}, \ldots \mathbf{b}_{n_m}\right) \tag{7}$$

where, m denotes the number of data samples, n denotes the total number of hidden layers and the output layer, and $\mathbf{Z}_0$ denotes the input matrix, which contains vehicle states. The matrices $\mathbf{W}_n$ and $\mathbf{B}_n$ denote the weight and bias matrices of the nth layer, respectively.

Progressively, the forward propagation can be described as

$$\mathbf{A}_0 = \mathbf{Z}_0 \tag{8}$$

$$\mathbf{Z}_i = \mathbf{W}_i^{\mathrm{T}} \mathbf{A}_{i-1} + \mathbf{B}_i, \quad i = 1 \ldots n \tag{9}$$

$$\mathbf{A}_i = \mathbf{f}^i\left(\mathbf{Z}_i\right), \quad i = 1 \ldots n \tag{10}$$

where, $\mathbf{Z}_i$ denotes the input of the ith layer, $\mathbf{A}_i$ denotes the output of the ith layer, and $\mathbf{f}^i$ denotes the ith activation function. The Tansig function was selected to use in the hidden layers, and the Purelin function was used between the hidden layer and the output layer [38].

For backward propagation, an appropriate loss function must be defined to calculate the errors between the DNN results and reference data. The reference data are the stiffness

and damping coefficients of the vehicle suspensions obtained from the multibody dynamics model. The mean square error was used as the loss function, which can be expressed as

$$\mathbf{L} = \frac{\|\mathbf{Y} - \mathbf{A}_n\|_2^2}{2m} \tag{11}$$

where, $\mathbf{L}$ denotes the loss of the DNN model, $\mathbf{Y}$ denotes the reference data, and $\mathbf{A}_n$ denotes the results of the DNN model. During backward propagation, the weight and bias matrices of the ith layer were updated based on the following equations

$$\mathbf{W}_i = \mathbf{W}_i - \alpha \frac{\partial \mathbf{L}}{\partial \mathbf{W}} \tag{12}$$

$$\mathbf{b}_i = \mathbf{b}_i - \alpha \frac{\partial \mathbf{L}}{\partial \mathbf{b}} \tag{13}$$

where, α denotes the learning rate. By continuously adjusting the weight and bias matrices, the loss function keeps decreasing. In general, the DNN model accuracy increases with the decrease in the loss function.

It is critical to select a suitable optimisation algorithm in the backpropagation. In this section, the Levenberg-Marquardt algorithm (LMA) was used, which is a hybrid technique that uses both Gauss-Newton and steepest descent approaches for convergence to an optimal solution. The LMA uses aspects of the steepest descent approach to traverse the design space and find a potential solution area, and then find the optimum. This technique is particularly effective in solving systems of nonlinear equations. Furthermore, the training errors of multiple DNN models with different inputs and training samples are similar and close to the same order of magnitude, proving that the DNN models capture the global minimum. By implementing the forward and backward propagations, we finally built a DNN model to describe the relationship between vehicle states and suspension parameters. The DNN modelling procedure is described in Figure 5. It is noteworthy that the response surface method and the neural network method are widely-used methods to describe this relationship. However, compared with the DNN modelling method, they have low prediction accuracy. The DNN method, which is based on a large amount of experimental or simulated data, has higher accuracy and efficiency in parameter prediction.

5. Results

5.1. DNN results

In this section, 2000 sets of data samples were used for DNN prediction. The number and size of the hidden layers affect the training error and prediction accuracy. The use of deeper layers and too many neurons may lead to overfitting, which increases the difficulty of training and makes the model difficult to converge. In addition, using too few neurons in the hidden layers leads to underfitting [36,37]. Thus, a DNN model with 3 hidden layers corresponding to 27, 19, and 11 neurons in each layer was developed. Figure 6 describes the relationship between DNN inputs (vehicle states) and outputs (suspension parameters). The estimated suspension parameters were also compared with the reference data (multibody model results) for accuracy analysis. The abscissa and ordinate in Figure 6 represent

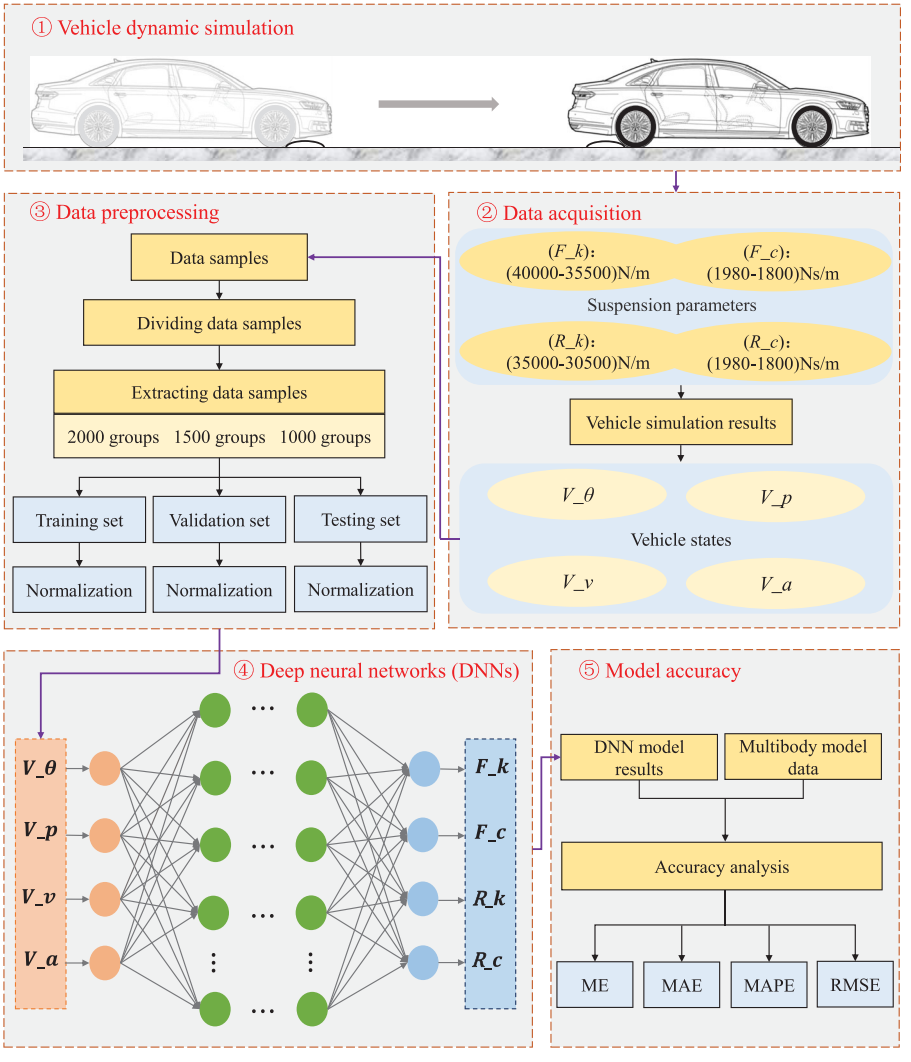
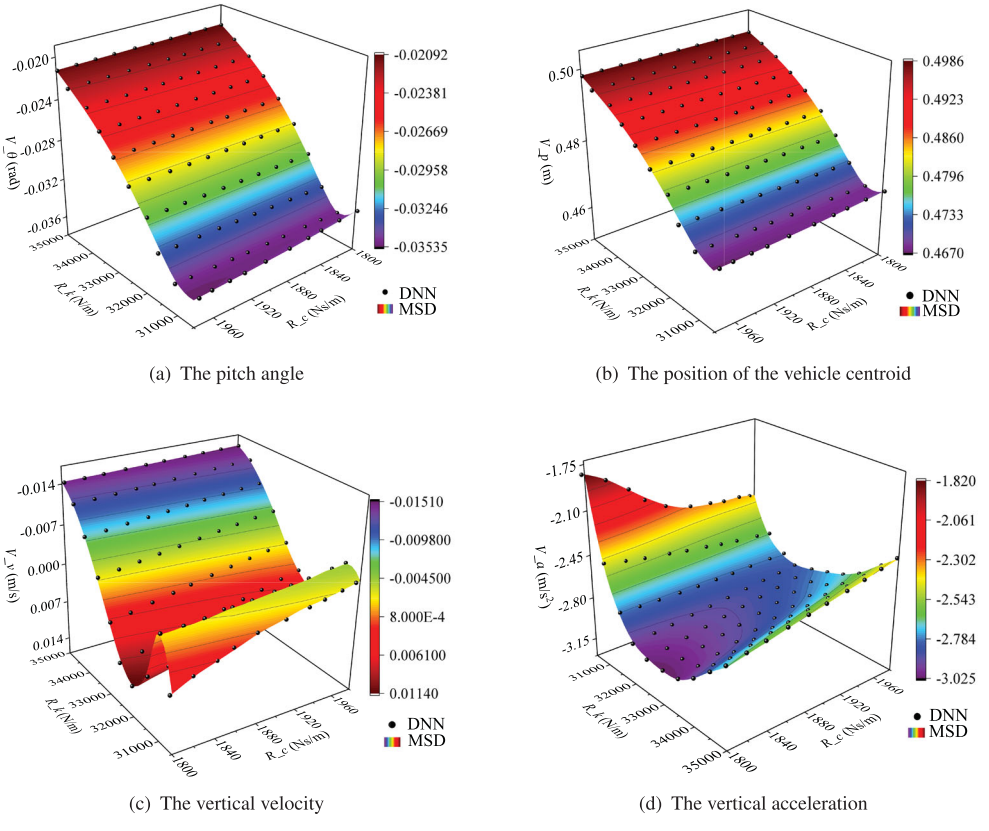


Figure 5. DNN modelling procedure for parameter identification of vehicle suspensions.

the stiffness and damping coefficients of rear suspensions, respectively. The vertical coordinate represents the four vehicle states. For better visualisation, DNN results are partially shown in Figure 6.

We used box plots, shown in Figure 7, to describe the absolute percentage errors between the DNN results and the reference data. The elements in the box plots include the median, mean, and lower and upper quartiles. The median is the middle value of the DNN results when arranged either in descending or ascending order. The mean is the average of the DNN results. Quartiles are the partitioned values, which divide the whole series into four equal parts. The difference between the upper quartile and the lower quartile is known as the interquartile range (IQR). They clearly depict the data dispersion and the bias of the data and can be used to describe the discrete distribution of data. Figure 7 shows the centre position and spreading range of the data distribution. The height of the box reflects



(a) The pitch angle

(b) The position of the vehicle centroid

(c) The vertical velocity

(d) The vertical acceleration

Figure 6. DNN and multibody model (reference) results. (a) The pitch angle. (b) The position of the vehicle centroid. (c) The vertical velocity and (d) The vertical acceleration.

the fluctuation of the data to a certain extent. The percentage absolute errors of more than 90% of the DNN results are very small.

Based on Figures 6 and 7, the following observations can be summarised.

- The outputs of the DNN model are close to the reference data. The maximum absolute percentage errors of suspension stiffness and damping are less than 0.2% and 0.5%, respectively.
- The analysi of the DNN results show that the average absolute percentage errors of suspension stiffness and damping are less than 0.05% and 0.2%, respectively.

The DNN model developed based on a large amount of data can effectively estimate suspension parameters. The DNN results fully meet the requied solution accuracy.

5.2. DNN model accuracy

In this section, the error functions, including the mean absolute error (MAE), the mean absolute percentage error (MAPE), the maximum absolute error, the root mean square

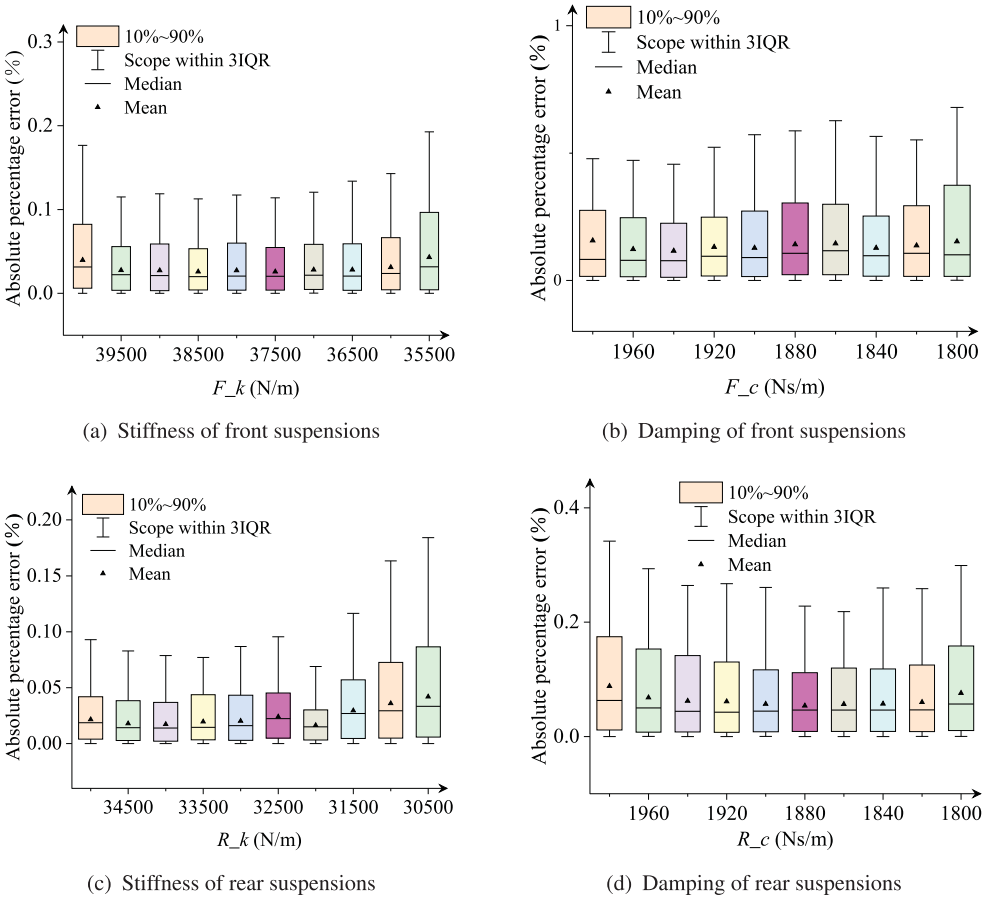


Figure 7. Box plots of percentage absolute error: suspension parameters. (a) Stiffness of front suspensions. (b) Damping of front suspensions. (c) Stiffness of rear suspensions and (d) Damping of rear suspensions.

error (RMSE), and R^2, were used to investigate the model accuracy. R^2 denotes the coefficient of determination and is an important statistical indicator to illustrate the reliability of the variable change of the regression model. These error functions are expressed as

$$\mathrm{ME} = \max_{1\le i\le l}\left\{\left|y_i - \hat{y}_i\right|\right\} \tag{14}$$

$$\mathrm{MAE} = \frac{1}{l}\sum_{i=1}^{l}\left|y_i - \hat{y}_i\right| \tag{15}$$

$$\mathrm{MAPE} = \frac{1}{l}\sum_{i=1}^{l}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{16}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{l}\sum_{i=1}^{l}\left(y_i - \hat{y}_i\right)^2} \tag{17}$$

Table 2. Accuracy analysis of DNN models (2000 data sets).

DNN model	Parameters	MAE	MAPE (%)	RMSE	R^2
DNN3_2000	*F_k*	11.4	0.03	15.9	0.999
	F_c	2.6	0.13	4.2	0.995
	R_k	7.9	0.02	10.6	0.999
	R_c	1.2	0.06	1.7	0.999
DNN4_2000	*F_k*	11.5	0.03	17.6	0.999
	F_c	2.5	0.13	5.3	0.992
	R_k	7.0	0.02	9.5	0.999
	R_c	1.2	0.06	1.8	0.999
DNN5_2000	*F_k*	11.6	0.03	17.4	0.999
	F_c	2.5	0.13	4.4	0.994
	R_k	7.6	0.02	10.7	0.999
	R_c	1.2	0.06	1.7	0.999

Table 3. Accuracy analysis of DNN models (1500 data sets).

DNN model	Parameters	MAE	MAPE (%)	RMSE	R^2
DNN3_1500	*F_k*	12.2	0.03	17.4	0.999
	F_c	2.9	0.15	5.0	0.992
	R_k	8.8	0.02	12.0	0.999
	R_c	1.3	0.07	1.9	0.999
DNN4_1500	*F_k*	13.0	0.03	18.9	0.999
	F_c	2.8	0.14	5.1	0.992
	R_k	8.8	0.02	12.0	0.999
	R_c	1.3	0.06	2.0	0.998
DNN5_1500	*F_k*	9.8	0.02	15.3	0.999
	F_c	2.7	0.14	5.1	0.992
	R_k	6.5	0.02	9.6	0.999
	R_c	1.1	0.06	1.9	0.999

$$R^2 = 1 - \frac{\sum_{i=1}^{l} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{l} (y_i - \bar{y}_i)^2} \tag{18}$$

where, y_i denotes the ith value of the reference data, $\hat{y}_i$ denotes the ith value of the DNN results, $\bar{y}_i$ denotes the ith mean value of the DNN results, and l denotes the number of groups. The five error functions were used as evaluation metrics to quantify the model accuracy. The smaller the first four metrics are, the more accurate is the DNN model. However, if R^2 is closer to 1, it indicates that the DNN model fits well.

Tables 2, 3 and 4 depict the evaluation metrics of the outputs of DNN models when 2000, 1500, and 1000 data sets and 3, 4, and 5 hidden layers were used, respectively. In these tables, the abbreviations were used for convenience (e.g. DNN3_2000 represents the DNN model with 3 hidden layers and 2000 data sets).

These tables show that the MAE, MAPE, and RMSE are relatively small, and R^2 is close to 1. The results show that these DNN models can accurately estimate the suspension parameters via the four vehicle states. The DNN models based on 2000 data sets are the most accurate. This coincides with the theory of DNNs. However, overfitting may accur in the case of a large amount of data sets. For the DNN models based on 2000, 1500 and 1000 data sets, with the increase in the number of of hidden layers, the accuracy improvement is not obvious, but the training time increases. To better visualise the DNN results and their errors, we took MAE as an example to compare the DNN models, as shown in Figure 8.

Table 4. Accuracy analysis of DNN models (1000 data sets).

DNN model	Parameters	MAE	MAPE	RMSE (%)	R^2
DNN3_1000	*F_k*	12.3	0.03	18.0	0.999
	F_c	3.8	0.20	7.1	0.985
	R_k	10.7	0.03	14.6	0.999
	R_c	1.4	0.07	2.3	0.998
DNN4_1000	*F_k*	11.1	0.03	16.8	0.999
	F_c	3.0	0.15	5.6	0.991
	R_k	6.6	0.02	7.4	0.999
	R_c	1.2	0.06	1.8	0.999
DNN5_1000	*F_k*	13.8	0.03	22.4	0.999
	F_c	3.2	0.17	5.5	0.991
	R_k	8.5	0.02	11.8	0.999
	R_c	1.3	0.07	2.1	0.999

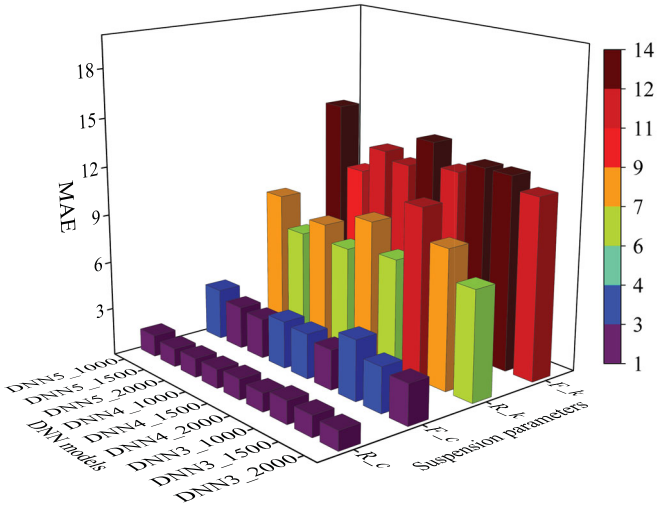


Figure 8. Comparison results of different DNN models in terms of MAE.

6. Discussion

The increase in the number of hidden layers does not lead to more accurate DNN results. To further improve the estimation accuracy of the stiffness and damping coefficients of the vehicle suspensions, we added pitch rate data of the vehicle to the inputs to retrain the DNN model. The inputs of the neural networks were the position of the vehicle centroid, the vertical velocity and acceleration, and the pitch angle and rate. The DNN results obtained using the 5 inputs were compared with the previous DNN results. To quantify the differences between the DNN models with 4 and 5 inputs, the MAE, MAPE, and RMSE of each suspension parameter are depicted in Table 5.

The DNN4 → 4 represents the DNN model with 4 inputs (vehicle states) and 4 outputs (suspension parameters). The DNN5 → 4 represents the DNN model with 5 inputs and 4 outputs. Table 5 shows that the accuracy of the DNN model with 5 inputs is significantly higher. The comparison results are described in Figure 9. The DNN results obtained considering the vehicle pitch rate are much more accurate, regardless of how the number of hidden layers and the size of data sets were varied. It indicates that the effective inputs can

Table 5. Accuracy analysis of DNN models with 4 and 5 inputs.

Suspension parameters	Evaluation metrics	DNN4 → 4	DNN5 → 4
Front stiffness (*F_k*)	MAE	11.4	6.8
	MAPE	0.03	0.01
	RMSE	15.9	9.2
Front damping (*F_c*)	MAE	2.6	1.6
	MAPE	0.13	0.08
	RMSE	10.6	5.5
Rear stiffness (*R_k*)	MAE	7.9	4.2
	MAPE	0.02	0.01
	RMSE	10.6	5.5
Rear damping (*R_c*)	MAE	1.2	0.89
	MAPE	0.06	0.04
	RMSE	1.7	1.2

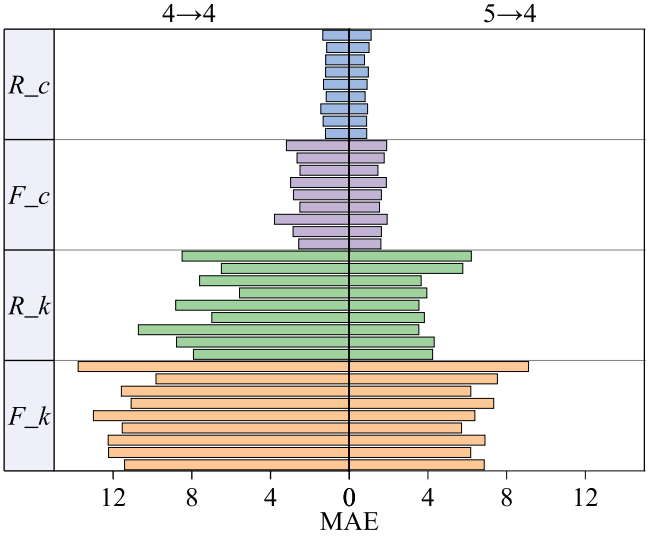


Figure 9. Comparison results with different DNN inputs in terms of MAE.

improve the accuracy of the DNN model significantly. However, the pitch rate of the vehicle is difficult or even impossible to measure by sensor networks. This issue will be addressed by other advanced instruments, techniques, or algorithms in the future.

7. Conclusions

In this work, a deep learning-based parameter identification method was developed to estimate the stiffness and damping coefficients of vehicle suspensions subjected to performance degradation. Dynamic simulations on a bumpy road were performed based on a vehicle multibody model to capture vehicle states. The multibody model considers the dynamic properties of each component (e.g. the chassis frame, suspensions, and tires), leading to accurate vehicle state determination. The data of vehicle states and suspension parameters were defined as DNN inputs and outputs, respectively, to train the DNN model. Consequently, the DNN model was used to estimate the stiffness and damping coefficients of the suspensions using real vehicle states obtained by sensor networks.

The accuracy of the DNN model was investigated in terms of different DNN inputs, data samples, and hidden layers. The error functions were used to quantify the errors of the DNN results. The comparison results showed that the average absolute percentage errors were less than 0.05% and 0.2% for the stiffness and damping coefficients, respectively. Furthermore, the pitch rate of the vehicle was added to DNN inputs to improve the prediction accuracy. The results indicate that the accuracy of stiffness and damping coefficient prediction can be improved using the DNN model with more effective inputs. Overall, a maneuver-based parameter identification method was proposed using a vehicle multibody model and DNNs. This method can be used for the fault diagnosis of vehicle suspensions subjected to performance degradation.

Disclosure statement

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Funding

This work was supported by the National Natural Science Foundation of China (Project Nos. 12072050 and 12211530029), the Research Project of State Key Laboratory of Mechanical System and Vibration (Project No. MSV202216).

References

[1] Savaresi S, Poussot-Vassal C, Spelta C, et al. Semi-active suspension control design for vehicles. Oxford: Elsevier. 2010.
[2] Wang X, Bi F, Du H. Reduction of low frequency vibration of truck driver and seating system through system parameter identification, sensitivity analysis and active control. Mech Syst Signal Process. 2018;105:16–35.
[3] Liu X, Alfi S, Bruni S. An efficient recursive least square-based condition monitoring approach for a rail vehicle suspension system. Vehicle Syst Dyn. 2016;54(6):814–830.
[4] Wang H, Jing X. Vibration signal–based fault diagnosis in complex structures: A beam-like structure approach. Struct Health Monit. 2018;17(3):472–493.
[5] Na J, Huang Y, Wu X, et al. Active adaptive estimation and control for vehicle suspensions with prescribed performance. IEEE Trans Control Syst Technol. 2018;26(6):2063–2077.
[6] Wang C, Wang Z, Zhang L, et al. A vehicle rollover evaluation system based on enabling state and parameter estimation. IEEE Trans Ind Inform. 2020;17(6):4003–4013.
[7] Rahim AAA, Abdullah S, Singh SSK, et al. Relationship between time domain and frequency domain strain signal–application to real data. J Mech Eng. 2018;6:178–191.
[8] Dumitriu M. Fault detection of damper in railway vehicle suspension based on the cross-correlation analysis of bogie accelerations. Mech Ind. 2019;20(1):102.
[9] Liu F, Zhang H, He X, et al. Correlation signal subset-based stochastic subspace identification for an online identification of railway vehicle suspension systems. Vehicle Syst Dyn. 2020;58(4):569–589.
[10] Singh KB, Arat MA, Taheri S. Literature review and fundamental approaches for vehicle and tire state estimation. Vehicle Syst Dyn. 2019;57(11):1643–1665.
[11] Reina G, Paiano M, Blanco-Claraco JL. Vehicle parameter estimation using a model-based estimator. Mech Syst Signal Process. 2017;87:227–241.
[12] Li C, Luo S, Cole C, et al. A signal-based fault detection and classification method for heavy haul wagons. Vehicle Syst Dyn. 2017;55(12):1807–1822.

[13] Wang Z, Dong M, Qin Y, et al. Suspension system state estimation using adaptive Kalman filtering based on road classification. Vehicle Syst Dyn. 2017;55(3):371–398.
[14] Stambaugh C, Abbott P, Vlajic N. Parameter identification in a magnetic suspension force transduction system. J Dyn Syst Meas Control. 2019;141(12):Article ID 124503.
[15] Lee DH, Kim GW. Estimation of tire stiffness variation based on adaptive extended Kalman filter of suspension systems and its application to indirect tire pressure monitoring system. In: Huang H, editor. Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020; Vol. 11379. SPIE; 2020. p. 355–360.
[16] Zhang Z, Xu N, Chen H, et al. State observers for suspension systems with interacting multiple model unscented Kalman filter subject to Markovian switching. Int J Autom Technol. 2021;22(6):1459–1473.
[17] Bogdanski K, Best MC. Kalman and particle filtering methods for full vehicle and tyre identification. Vehicle Syst Dyn. 2018;56(5):769–790.
[18] Choi BC, Cho S, Kim CW. Kriging model based optimization of macpherson strut suspension for minimizing side load using flexible multi-body dynamics. Int J Prec Eng Manuf. 2018;19(6):873–879.
[19] Risaliti E, Tamarozzi T, Vermaut M, et al. Multibody model based estimation of multiple loads and strain field on a vehicle suspension system. Mech Syst Signal Process. 2019;123:1–25.
[20] Pan Y, Xiang S, He Y, et al. The validation of a semi-recursive vehicle dynamics model for a real-time simulation. Mech Mach Theory. 2020;151:Article ID 103907.
[21] Pan Y, Dai W, Huang L, et al. Iterative refinement algorithm for efficient velocities and accelerations solutions in closed-loop multibody dynamics. Mech Syst Signal Process. 2021;152:Article ID 107463.
[22] Rodríguez AJ, Sanjurjo E, Pastorino R, et al. State, parameter and input observers based on multibody models and kalman filters for vehicle dynamics. Mech Syst Signal Process. 2021;155:Article ID 107544.
[23] Luo H, Huang M, Zhou Z. Integration of multi-gaussian fitting and LSTM neural networks for health monitoring of an automotive suspension component. J Sound Vib. 2018;428:87–103.
[24] Kong Y, Abdullah S, Schramm D, et al. Optimization of spring fatigue life prediction model for vehicle ride using hybrid multi-layer perceptron artificial neural networks. Mech Syst Signal Process. 2019;122:597–621.
[25] Zhang M, Jing X, Wang G. Bioinspired nonlinear dynamics-based adaptive neural network control for vehicle suspension systems with uncertain/unknown dynamics and input delay. IEEE Trans Ind Electron. 2021;68(12):12646–12656.
[26] Sze V, Chen YH, Yang TJ, et al. Efficient processing of deep neural networks: A tutorial and survey. Proc IEEE. 2017;105(12):2295–2329.
[27] Hossain MS, Ong ZC, Ismail Z, et al. Artificial neural networks for vibration based inverse parametric identifications: A review. Appl Soft Comput. 2017;52:203–219.
[28] Rodríguez JI, Jiménez JM, Funes FJ, et al. Recursive and residual algorithms for the efficient numerical integration of multi-body systems. Multibody Syst Dyn. 2004;11(4):295–320.
[29] Funes FJ, García de Jalón J. An efficient dynamic formulation for solving rigid and flexible multibody systems based on semirecursive method and implicit integration. J Comput Nonlinear Dynam. 2016;11(5):Article ID 051001.
[30] Jerkovsky W. The structure of multibody dynamic equations. J Guid Control Dynam. 1978;1(3):173–182.
[31] von Schwerin R. Multibody system simulation, numerical methods, algorithms and software. Heidelberg: Springer; 1999.
[32] García de Jalón J, Álvarez E, de Ribera F, et al. A fast and simple semi-recursive formulation for multi-rigid-body systems. In: Ambrósio J, editor. Advances in computational multibody systems. Springer Netherlands; 2005. Chapter 1. p. 1–23. (Computational methods in applied sciences; vol. 2).
[33] Pan Y, García de Jalón J. Iterative refinement of accelerations in real-time vehicle dynamics. J Comput Nonlinear Dyn. 2017;13(1):Article ID 011009.

[34] Pan Y, He Y, Mikkola A. Accurate real-time truck simulation via semirecursive formulation and adams-bashforth-moulton algorithm. Acta Mech Sinica. 2019 Feb;35:641–652.
[35] Antić B, Pešić D, Vujanić M, et al. The influence of speed bumps heights to the decrease of the vehicle speed–Belgrade experience. Saf Sci. 2013;57:303–312.
[36] Qiao L, Li X, Umer Q, et al. Deep learning based software defect prediction. Neurocomputing. 2020;385:100–110.
[37] Tian H, Wang P, Tansey K, et al. An IPSO-BP neural network for estimating wheat yield using two remotely sensed variables in the Guanzhong Plain, PR China. Comput Electron Agric. 2020;169:Article ID 105180.
[38] Prasad R, Pandey A, Singh K, et al. Retrieval of spinach crop parameters by microwave remote sensing with back propagation artificial neural networks: a comparison of different transfer functions. Adv Space Res. 2012;50(3):363–370.