

Received September 14, 2020, accepted September 20, 2020, date of publication September 28, 2020, date of current version October 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027499

Predictability of Vibration Loads From Experimental Data by Means of Reduced Vehicle Models and Machine Learning

LEO DOSTAL¹, HELGE GROSSERT¹,
DANIEL A. DUECKER¹, (Graduate Student Member, IEEE),
MALTE GRUBE¹, DANIEL KREUTER², KAI SANDMANN²,
BENJAMIN ZILLMANN², AND ROBERT SEIFRIED¹

¹Institute of Mechanics and Ocean Engineering, Hamburg University of Technology, 21073 Hamburg, Germany

²Robert Bosch GmbH, 71272 Renningen, Germany

Corresponding author: Leo Dostal (dostal@tuhh.de)

This work was supported in part by the European Union's Horizon 2020 research and innovation program under Grant 769506, and in part by the Hamburg University of Technology (TUHH) in the funding programme Open Access Publishing for the Open Access fees.

ABSTRACT Nowadays electric cars are in the spotlight of automotive research. In this context we consider data based approaches as tools to improve and facilitate the car design process. Hereby, we address the challenge of vibration load prediction for electric cars using neural network based machine learning (ML), a data-based frequency response function approach, and a hybrid combined model. We extensively study the challenging case of vibration load prediction of car components, such as the traction battery of an electric car. We show using experimental data from Fiat 500e and VW eGolf cars that the proposed ML approach is able to outperform the classical model estimation by means of ARX and ARMAX models. Moreover, we evaluate the performance of a hybrid-ML concept for combination of ML and ARMAX. Our promising results motivate further research in the field of vibration load prediction using machine learning based approaches in order to facilitate design processes.

INDEX TERMS Automotive engineering, machine learning, time series analysis, vibration measurement.

I. INTRODUCTION

Traction batteries with high energy densities which power electric cars are a focus area in automotive research and development. This dynamic development motivates to rethink and improve the car design process. The battery mass usually exceeds several hundred kilograms. Hence, the battery replaces the traditional combustion engine as the heaviest single component. Recently, Ruiz et al. presented an extensive survey [1] on existing international and national testing standards and regulations for battery systems in electric and hybrid electric vehicles. The authors group mechanical testing into classes covering mechanical shocks, drops, penetration, immersion, crush/crash, roll-over, and vibrations. Interestingly, nearly all classes target event-based fail-safe

behavior e. g. after accidents, rather than long-term durability which is covered by vibration testing.

However, vibration loads on the traction battery caused by the vehicle can be significant when driving on rough road surfaces or during highly dynamic maneuvers. Thus, it must be ensured that the battery can sustain these vibration loads, as damaged battery cells can lead to hazardous fire scenarios caused by thermal runaways [2]. Vibration tests can proof whether a system, e. g. the traction battery, is reliable against a random vibration induced by rough road driving as well as internal vibration of the power train. The main failures to be identified by this test are component breakage and fracture resulting in the loss of electrical energy. Hence, vibration load prediction is of major importance in the design process. Vibration fatigue analysis is a part of the mechanical reliability evaluation to ensure safety and satisfy the required lifetime. The evaluation involves usually a lot

The associate editor coordinating the review of this manuscript and approving it for publication was Min Xia¹.

of measurement at vehicle level and component level as well. A detailed analysis starts often with measurements on vehicle level. Therefore, acceleration sensors are going to be mounted on global locations, e.g. mounting points of the battery system. The vehicle measurements can be performed as a driving scenario on conventional roads with different portions of city roads, rural roads and highway, depending on the use case of the vehicle. Another option is an accelerated test under more severe conditions with a shorter testing time. For this case a rough road track can be used with significantly higher excitation amplitudes. However, the measurement gives a feedback about the effective vibration energy on the component, which was transferred from the road surface over the wheels, suspension and frame to the battery. The measured signals need to be extrapolated to higher testing times to reach a damage level which is equal to the defined lifetime of the component. Here, the extrapolation factor strongly depends on the measured scenario. Finally, an end of life test can be performed by taking the vibration profile and the extrapolation factor into account. These tests are usually performed on shakers with input data from the vehicle measurement. The vibration reliability test is usually stopped in case of failure or if the defined lifetime limit was reached. This study is aimed to significantly reduce the amount of vehicle testing by describing the vibration transfer path from the road to the component. This is achieved by identifying a suitable simulation model from data. Existing vibration test standards show considerable variations of the vibration profiles over a wide range of frequencies and amplitudes. It is worth mentioning that vibration profiles in these standards are often derived from generic measurements on conventional vehicles at locations appropriate for mounting traction batteries in electric vehicles.

So far and in accordance with [1] only very few work has been published on vibration profiles designed specifically for electric and hybrid electric vehicles. This is supported by Hooper and Marco [3], [4] pointing out that many of the vibration profiles described in the ISO-standards represent only a short term abuse rather than a mechanical durability test to represent a battery life cycle. Moreover, existing studies [5], [6] mainly focus on the individual battery cell's resiliency and performance drops due to loss of electric energy.

Consider the load propagation pipeline starting from an external excitation towards the individual cell within the traction battery pack. A gap exists covering the vibration load prediction from the source of excitation (e.g. tire on a bumpy road) towards the traction battery pack. However, the load profiles available in standards and regulations such as in [7] show a considerable variation and might be confusing for responsible design engineers. This leads to an over-engineered battery pack with high weight and cost that is prohibitive for successful vehicle integration. From the reliability point of view it would be helpful to have a simplified method to approximate vibration profiles in an early design process. In this context the early design process means that

a vehicle class is defined and the size, position and rough design of the battery is known. A detailed design state of the (new) car is usually not available at this time, especially for suppliers. Due to this fact, a simplified prediction model could give a first impression of the expected load data on the battery system.

The prediction model has to be both, versatile and efficient, since the design process persists of multiple iteration cycles. Moreover, the integration of prediction models into simulation frameworks may allow automated optimization procedures to generate optimized results for various design parameters e.g. the battery's positioning and mounting within the car body.

Fatigue damage spectrum (FDS) is a widely used method to estimate fatigue processes and component damages from external excitation and can be determined in closed form from acceleration data [8]. Thus, the corresponding acceleration of the respective component, e.g. the traction battery, has to be either known or simulated to be used in a subsequent fatigue analysis.

A standard procedure is the synthesis of sophisticated mechanical models of the real-world system using gray-box identification techniques in order to estimate dynamic loads on individual components. In a first step, a mechanical modeling is carried out. The corresponding parameters are then identified in a second step. Thereby, the modeling step is subjected to simplifications and assumptions being made by the engineer in order to limit the model complexity. While complex models might capture a detailed system behavior, they cannot guarantee a satisfactory match with the systems real-world behavior. Note that, as described before, complex models often suffer from a lack of detailed information about the real-world system, such as detailed material parameters and CAD-models which are usually only available to the OEM-company.

The absence of detailed prior system data requires extensive system identification experiments in order to obtain information on the real-world system behavior. Extensive resources are required to obtain information-rich data for gray-box identification techniques. Therefore, recent advances in the field of data-driven prediction models make the transfer of machine learning methods to the problem of vibration load prediction attractive.

The concept of artificial neural networks (NN) as a universal function approximator can be traced back to the fifties and sixties [9]. However, the advances in computing power within the last decade paved the way for NNs. While the original boost came from the field of image classification various deviates of NNs have been developed to meet the needs of specific problem categories. Convolutional neural networks (CNN) form a subgroup within the large class of feed-forward NN which map input data directly on their output [9]. In contrast, recurrent neural networks (RNN) consider data sequences as an input and have been shown to be suitable for data series prediction. However, RNNs face the problem of vanishing gradients which renders their

training challenging. This was targeted by long-short-term memory (LSTM) RNNs which were originally introduced in [10] and are nowadays commonly used for a wide range of applications. Examples include speech recognition [11], time series prediction [12], and material fatigue fault prediction [13], [14]. Moreover, identification of transport flow from data is studied in [15] using neural networks, in [16] using LSTM, in [17], [18] using fuzzy neural networks as well as in [19] using support vector machine and data denoising schemes. Another approach focuses on using a periodic function in order to improve model prediction performance [20]. In addition, we refer to Sec. IV for a more detailed discussion of NN concepts for system identification.

The contribution of this work is three-fold. We use acceleration measurements from two battery-electric vehicles driven over a bumpy road with constant speed in order to learn the vehicle model. Therefore, we, first, study the suitability of various data-driven vibration load prediction concepts. The approaches include, ARX, ARMAX as well as LSTM neural networks. Second, we propose a novel hybrid approach based on LSTM neural networks and an ARMAX model. Third, we evaluate and critically discuss the algorithms' prediction performance based on a real-world data set recorded on a rough road track with two experimental platforms, namely, VW eGolf and Fiat 500e electric cars.

The remainder of this work is structured as follows. In Sec. II we give a brief overview on vibration measurements on electric cars. In Sec. III we present a data based frequency response function approach to system identification. Section IV covers the nonlinear system identification, using a pure neural network approach and a combined error estimation approach respectively. Then we evaluate the performance of our concepts in real-world experiments using a Fiat 500e and a VW eGolf as test platforms on rough bumpy road. Finally, we summarize our results and draw conclusions in Sec. VII

II. VIBRATION MEASUREMENTS ON AN ELECTRIC VEHICLE

Experimental data are required to parameterize and validate the vehicle model. For this study a Fiat 500e passenger car was used with a performance of 87 kW, as depicted in Fig. 1. The vehicle is equipped with tri-axial accelerometers on the traction battery system, chassis and wheel hubs. A total of 13 accelerometers have been installed on the four wheel hubs and chassis. The installation position of the accelerometer at the front left wheel carrier is shown in Fig. 2. Further measurements were conducted on the traction battery, as shown in Fig. 3.

At the wheel carriers, the acceleration sensors T356A02 from PCB were used, which are suitable for measuring at a frequency range of 1 – 5000 Hz. At the wheel hubs and the battery, the acceleration sensors 4524B from B&K with a frequency range of 0.25 – 3000 Hz were used.

The measurements were performed on a rough road track that consists of sections of regularly distributed humps.



FIGURE 1. Fiat 500e test car. Sensor positions A1, A3 (wheel carrier), A5 (dome bearing), A7 (top of shock absorber).

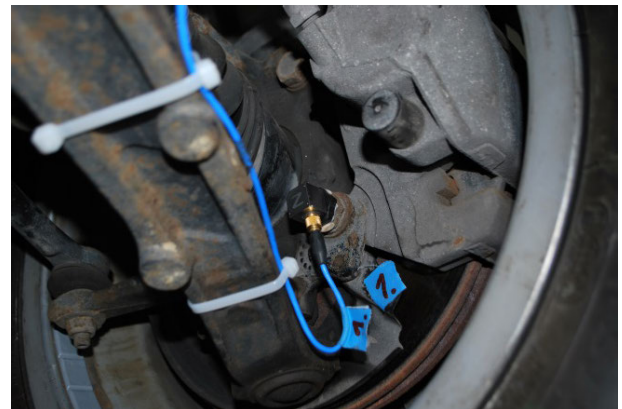


FIGURE 2. Accelerometer at front left wheel carrier.

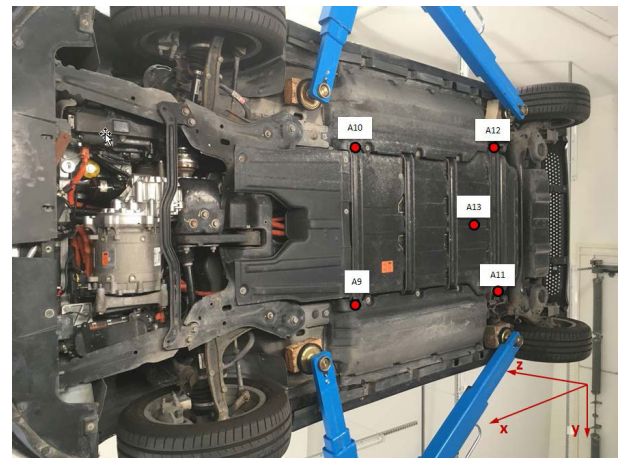


FIGURE 3. Accelerometer positions at the traction battery.

Those lead to section-wise periodic excitation when the car is driven with constant speed. The rough road track is depicted in Fig. 4. For data-based model synthesis, an “information-rich” data set is preferable. Such a data set might be created by taking measurements of the system under different conditions. Regarding the given situation, this was achieved by

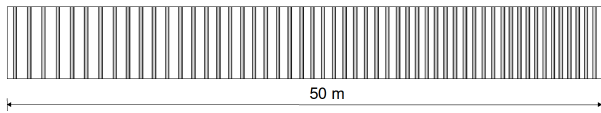


FIGURE 4. Depiction of the standardized rough road track.

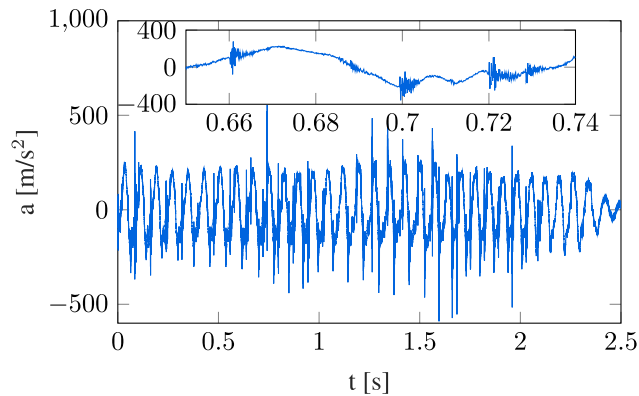


FIGURE 5. Recorded acceleration time series at left front wheel carrier for $v = 40$ km/h.

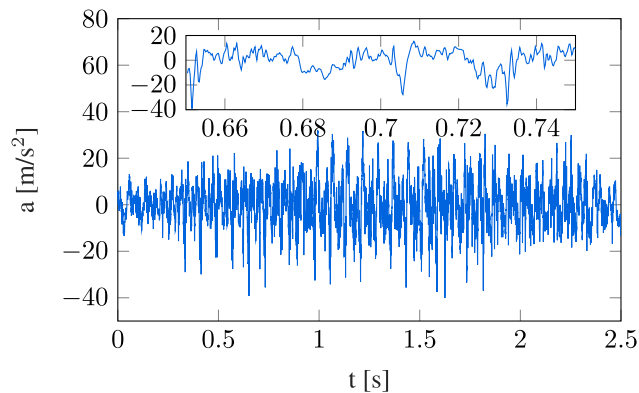


FIGURE 6. Recorded acceleration time series at front left of the battery for $v = 40$ km/h.

driving the car on the track with constant speeds of 20 km/h, 30 km/h, 40 km/h, and 50 km/h, respectively. An exemplary measurement extract from those experiments is shown in Figs. 5 and 6. They depict some part of the accelerometer time signals measured at 40 km/h vehicle velocity at the front left wheel carrier (Fig. 5) and the front left of the battery system (Fig. 6), as the test car drives on the rough road track. The data from the experiment were measured at a sampling frequency of 12000 Hz. This results in about 108000 sample points for $v = 20$ km/h, 72000 sample points for $v = 30$ km/h, 54000 sample points for $v = 40$ km/h and 43200 sample points for $v = 50$ km/h. However, these data were resampled to 3000 Hz, in order to match the frequency range of the acceleration sensors.

For the design of vehicle components, the occurring accelerations and the corresponding power spectra are of importance. Thereby, the power spectrum $S(f)$ is a function with respect to frequency f . The power spectra of the acceleration

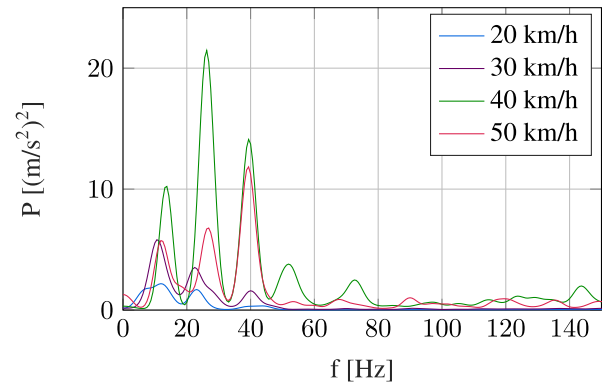


FIGURE 7. Power spectra of the acceleration at the front left of the battery for different velocities in detail.

at the front left of the battery are exemplary plotted in Fig. 7 for different velocities of the Fiat 500e. From the spectral densities other measures like variance var , mean upcrossing rate v_0^+ , or spectral moments m_i can be obtained. These are given by

$$m_i = \int_0^\infty f^i S(f) df, \quad var = m_0, \quad v_0^+ = \frac{1}{2\pi \sqrt{m_0/m_2}}. \quad (1)$$

For this reason, we present our results in terms of spectral densities and time series of the acceleration. Moreover, power spectra of relevant data and results are provided for download from the publisher. Often access to a sufficiently detailed vehicle model is not available for the determination of the occurring accelerations. Of particular interest are the accelerations occurring at the batteries of electric cars, since these are comparatively new and less well investigated components.

One approach is to create a model of the vehicle based on measured data in the form of transfer functions between relevant measuring points. For this purpose, linear transfer function models for linear system behavior and neural networks for nonlinear system behavior are used.

To reduce the effort for the collection of the measurement data necessary for system identification a good knowledge of the required measurement data is important. Thereby, both the position of the accelerometers, the number of required measurements, and the relevant frequency range are of interest for system identification.

The aim of this work is to determine which methods are suitable for the prediction of vibration loads on the basis of experimental data from vehicle tests.

III. DATA BASED STATE SPACE MODEL

A. TRANSFER FUNCTIONS

A widely used framework for linear system identification is the prediction error method [21]. The general model structure of the prediction error method with the input signal u , the output signal y and an unknown disturbance e is shown in Fig. 8. The most widely used models based on the prediction error method are the ARX [21, p. 81] and the

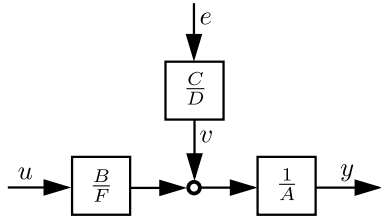


FIGURE 8. General model structure of the Prediction Error Method according to [21, p. 85].

ARMAX-model [21, p. 83]. These are discussed in detail as multi-input-single-output (MISO) models in the Appendix.

B. ARX AND ARMAX FITTING PROCEDURE

We have used a data set with one time series for each of the car velocities 20 km/h, 30 km/h, 40 km/h, and 50 km/h. One part of the car velocities is used for training and the left out part is used for validation. This additionally shows the interpolation capabilities of the identified models. Because the rough road our data set was recorded on consists of several barriers with decreasing distance in order to excite different frequencies, splitting each of the time series into a training and a validation part is not reasonable. For system identification, acceleration values measured at a vehicle speed of 20, 30 and 50 km/h are used in the training set. In the identification procedure for each of the data sets in the training set a separate transfer function is identified. Finally, these transfer functions are merged in order to obtain a single model. For merging, the transfer functions are weighted with their inverse covariance matrices as described in [21, p. 464 f.]. For validation acceleration values at a vehicle speed of 40 km/h are used. As output signal the battery acceleration front left (Bat_FL) is considered exemplary. As input signal different combinations of signals measured at the wheel carriers are examined. The data is filtered with a lowpass filter at a cutoff frequency of 1500 Hz and a sampling rate of 3000 Hz.

1) RESULTS

In Fig. 9 and Fig. 10 the mean squared error of the estimation

$$MSE = \frac{1}{N} \sum_{k=1}^N (y(k) - y_{pred.}(k))^2 \tag{2}$$

on the validation set for a vehicle speed of 40 km/h is displayed for different polynomial orders and exemplary input signal combinations. In these figures combinations of the acceleration signals of the wheel carriers at the front left (FL), front right (FR) and at the back left (BL) are considered as input signals. The best prediction result can be achieved using the signals from the front and the rear wheel carriers as input signals. If using only the front input signal, the estimation is much worse. Using more than two input signals does not result in a big improvement anymore.

For the input signal combinations (FL) and (FL, FR) with small and moderate polynomial orders the estimation error

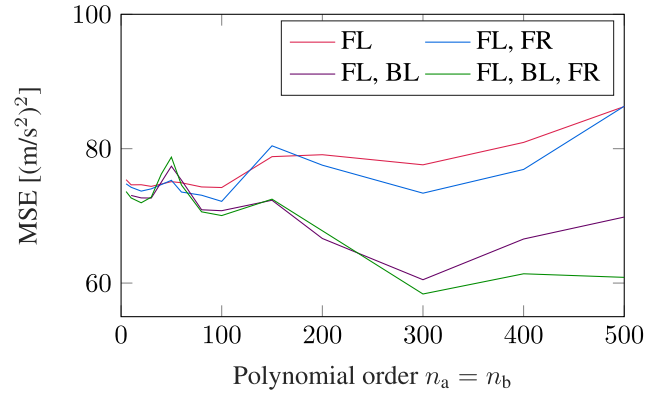


FIGURE 9. Estimation error: Influence of ARX model order for different input signals.

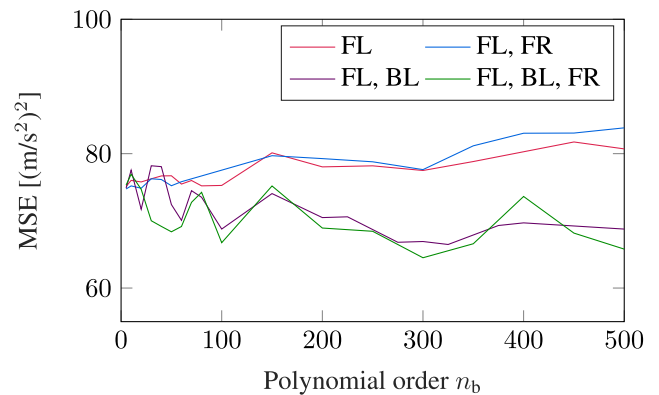


FIGURE 10. Estimation error: Influence of ARMAX model order for different input signals, n_a = n_c = 30.

decreases slightly for higher polynomial orders and increases for high polynomial orders. For the input signal combinations (FL, BL) and (FL, BL, FR) the simulation result of the ARX-model strongly improves for increasing polynomial orders until a polynomial order of 300 is reached. For higher polynomial orders the simulation result deteriorates. For the ARMAX-model only a slight improvement of the simulation result for higher polynomial orders can be observed.

The increasing estimation error on the validation set for increasing polynomial orders at high polynomial orders can be explained by overfitting for both the ARX and the ARMAX-model. Additionally, very high polynomial orders can lead to problems in the optimization which then converges to a not satisfactory local minimum. Especially the ARMAX-model can become unstable and therefore has to be stabilized during optimization.

For high polynomial orders of the ARMAX-model, especially for high polynomial orders of the C-polynomial, a stabilization is very difficult and often leads to comparatively bad local minima. Therefore, the polynomial orders n_a and n_c are limited to 30. For polynomial orders n_a and n_c as high as 60 no improvement of the simulation result could be achieved. Even higher polynomial orders do not lead to a stable result. For both models the order of the polynomial B has the biggest influence on the prediction error.

For lower polynomial orders better results can be achieved with the ARMAX-model. For high polynomial orders the results of the ARX-method are better because of the better convergence properties.

In the following polynomial orders of $n_a = n_b = 300$ for the ARX-model and polynomial orders of $n_a = n_c = 30, n_b = 100$ for the ARMAX-model are used.

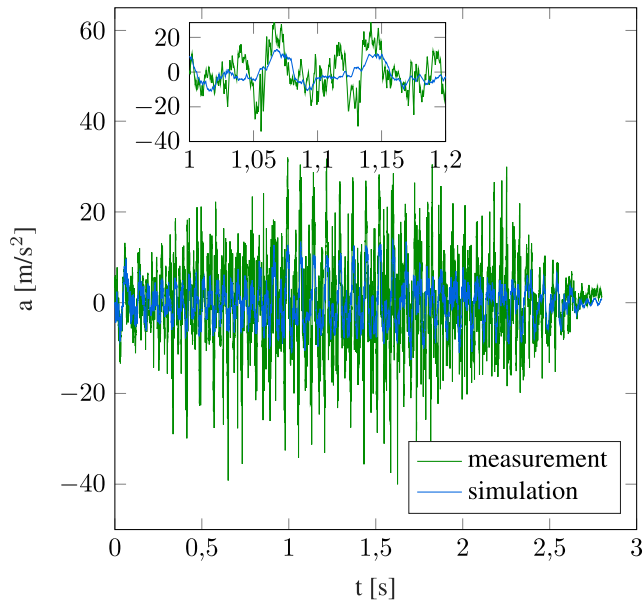


FIGURE 11. ARX prediction for 40 km/h.

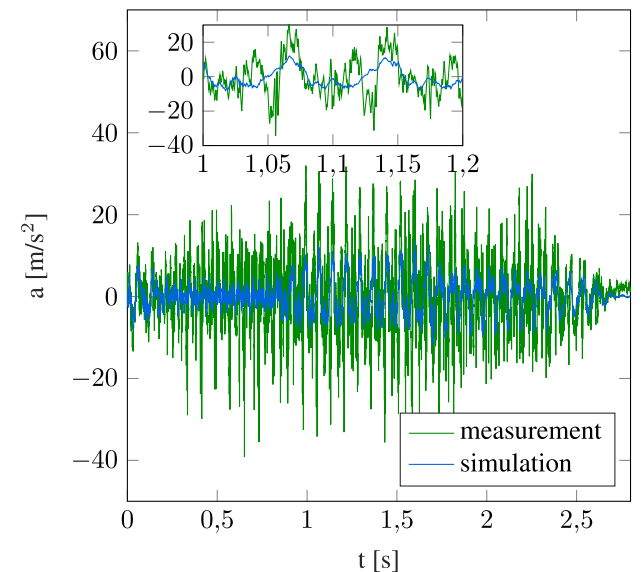


FIGURE 12. ARMAX prediction for 40 km/h.

The simulation results of the ARX and the ARMAX-method for the vertical acceleration of the battery at the front left, using the wheel carrier acceleration at the front left and the back left as input signals, are shown in Fig. 11-14 exemplary for a vehicle speed of 40 km/h. The time domain

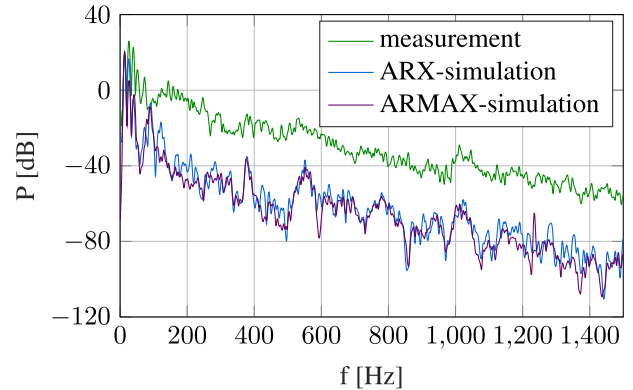


FIGURE 13. Power spectrum of prediction for 40 km/h.

comparison between measured and simulated acceleration is shown in Fig. 11 and Fig. 12. The power spectrum of the measured and predicted acceleration signal is displayed in Fig. 13 and Fig. 14. The acceleration spectral density is a measure for the energy distribution of an acceleration signal over the frequency. A direct numerical computation of the power of an acceleration signal is often unreliable because of drifts caused by the numerical integration of the acceleration signal.

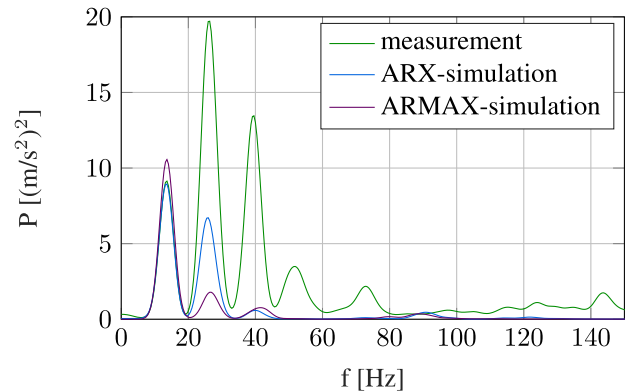


FIGURE 14. Power spectrum prediction for 40 km/h in detail.

It follows from Fig. 13 that the energy of the acceleration signal is mainly concentrated in the frequency range much lower than 200 Hz. Especially the frequency range from 10 – 60 Hz dominates the energy of the acceleration signal. For low frequencies up to 20 Hz both methods estimate the power of the acceleration signal very accurate. The power of the acceleration signal in a frequency range of 20 – 60 Hz is underestimated, which leads to an underestimation of the energy of the acceleration signal.

This can be seen in Fig. 11 and Fig. 12 as well. While the lower frequencies of the acceleration signal are simulated well the acceleration peaks, which are dominated by higher frequency signal components, are clearly underestimated. Thus the higher frequency components are important for simulating the maximum amplitude of the acceleration, but

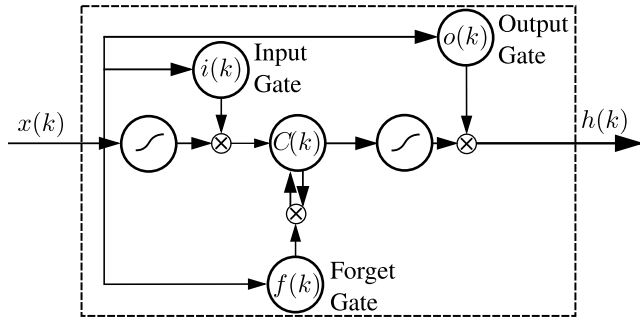


FIGURE 15. LSTM structure according to [24], [25].

only have a minor influence on the energy of the acceleration signal.

The higher discrepancy of the measured and simulated acceleration spectral density for very high frequencies above 200 – 400 Hz in Fig. 13 has only a minor influence on the simulation result because the power of the signal in that signal range is much lower than the power for lower frequencies.

IV. SYSTEM IDENTIFICATION WITH NEURAL NETWORKS

For the identification of nonlinear systems, classical feed-forward networks and recurrent neural networks (RNNs) are particularly suitable [21]. In contrast to feed-forward networks, recurrent networks allow a bidirectional information flow. In the context of time series prediction, this means that the output of such a network serves as part of the input to the same network in the next time step. This allows a good representation of time-dependent system dynamics. In this paper, we use a specific class of recurrent neural networks, the Long-Short-Term-Memory networks (LSTMs) [10], in order to predict the loads on the battery cell. We provide a brief overview in the following.

A. LSTMS

LSTM networks efficiently address the problem of RNNs regarding long-term dependencies [22]. Their capability of solving such problems is based on their special structure, that is depicted in Fig. 15. It consists of an input gate $i(k)$, an output gate $o(k)$, a forget gate $f(k)$ and the cell state $C(k)$. Thereby, $x(k)$ denotes the input vector and $h(k)$ the output vector. Every gate is a neural network itself and contains the weights and biases that are optimized during training.

The crucial part of the LSTM is its cell state, that stores information from previous inputs. The input gate controls, based on the current input, which information is added to a cell state from the input itself. In contrast to that, the forget gate controls, which information from the old cell state is conserved and transferred to the new state. The output vector $h(k)$ is generated based on the new cell state and the output of the output gate, that the input vector is fed into.

The neural networks used in this paper are implemented in Python 3 using TensorFlow [23].

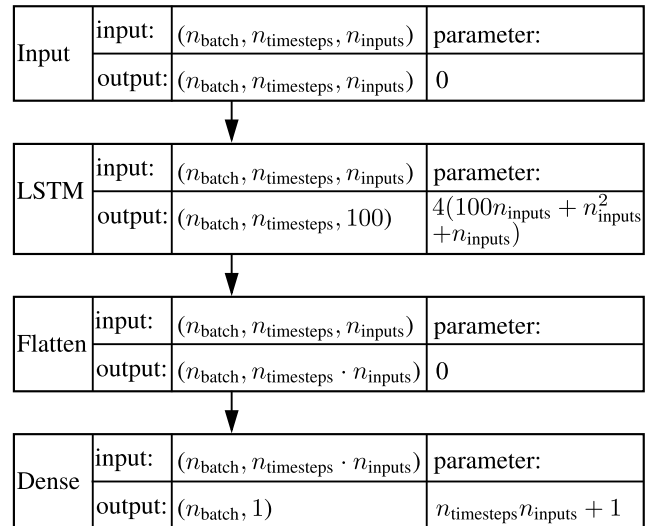


FIGURE 16. Structure of the considered neural network.

B. DIRECT ESTIMATE OF THE OUTPUT SIGNAL

The most common method of nonlinear system identification is the direct estimation of the output signal of the system from the input signals of the system. In this work, we use a neural network with a hidden LSTM layer and a dense layer as output layer. The model structure of the neural network is shown in Fig. 16. Compared to other tested model structures, this model structure achieved the best results in this study. In particular, better results were obtained with a hidden LSTM layer than with a hidden dense layer.

As the input signals the accelerations at the wheel carriers front left and rear left are used. As the output signal acceleration at the front left of the battery is considered. As described in section III-B for the ARX and the ARMAX model we have used acceleration values measured at 20, 30 and 50 km/h for training and values measured at 40 km/h for validation. In the following parameter study, the MSE error on the validation set is considered. The considered measurement series and the input and output signals are listed in Tab. 1.

TABLE 1. Measurement data used in the parameter study.

	training	validation
vehicle velocity [km/h]	20, 30, 50	40
input signals	wheel carrier: FL, RL	
output signal	battery: FL	

Before the network is trained, the batch size n_{batch} and the number of time steps $n_{\text{timesteps}}$, which the LSTM neurons can store, has to be set. Also the number n_{epoch} of epochs must be specified, in which the neural network is trained.

1) STORED TIME STEPS OF THE LSTM NEURONS

An important parameter for LSTM neurons is the number of stored time steps $n_{\text{timesteps}}$. With the sampling interval T of the training and validation data, this can be converted into the

more descriptive storage time

$$t_{\text{LSTM}} = T n_{\text{timesteps}} \quad (3)$$

In order to determine the optimal number of stored time steps, the neural network is trained in 200 epochs for a batch size of 500 and 1000, whereby the number of stored time steps is varied.

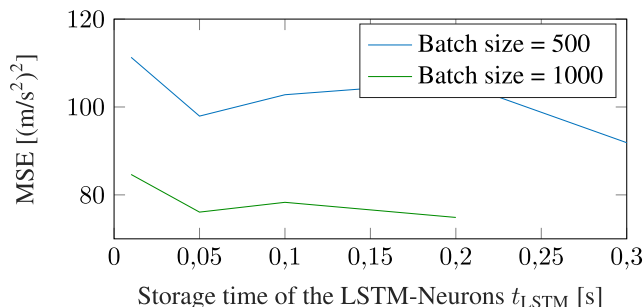


FIGURE 17. Influence of the storage time t_{LSTM} of the LSTM neurons on the prediction error.

A higher number of stored time steps leads to a lower prediction error. This is shown in Fig. 17, where the same trend for both chosen batch sizes can be seen. The number of trainable parameters of the neural network does not depend on the storage time. The required computing time and the required GPU memory increase linearly with the number of stored time steps.

2) BATCH SIZE

In order to determine an optimal batch size, the neural network is trained using $t_{\text{LSTM}} = 0.1$ s and $n_{\text{epoch}} = 200$ for different batch sizes. In Fig. 18 the influence of the batch size on the simulation error and the computing time is shown. For larger batch sizes a better result can be achieved. The maximum batch size is limited by the memory of the GPU used for training. Very small batch sizes lead to convergence against bad local minima which leads to very different results depending on the random initialization of the neural network.

3) NUMBER OF EPOCHS

Another important parameter for the training of neural networks is the number of epochs. If the number of epochs is too small, underfitting occurs. Thereby, large errors occur both on the training set and on the validation set. If the number of epochs is selected too large, the error on the training set is minimized, but the generalization and thus the error on the validation set becomes worse. This is called overfitting.

In order to determine the influence of the number of epochs for training the neural network, it is trained for $t_{\text{LSTM}} = 0.1$ s and different batch sizes, while the number of epochs is varied. In Fig. 19 the error on the validation set is plotted over the number of epochs for different batch sizes. For small number of epochs the error reduces with increasing number of epochs. If more than 50 epochs are used, the prediction error on the validation set increases due to overfitting. Using

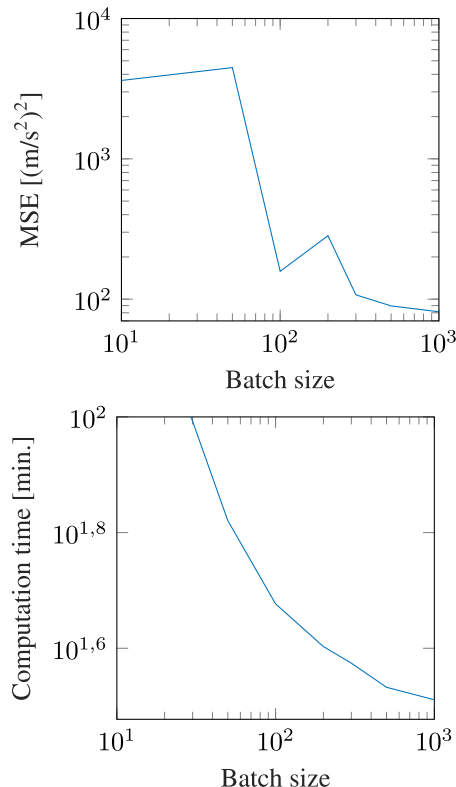


FIGURE 18. Influence of the batch size of the neural network.

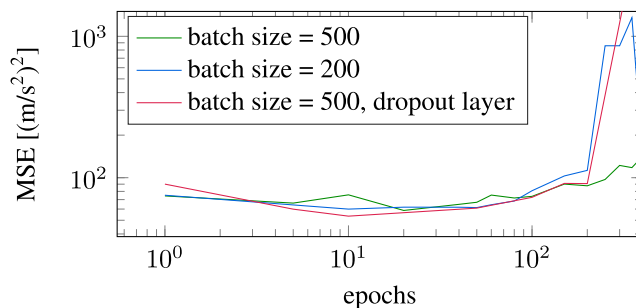


FIGURE 19. Influence of the number of epochs on the prediction error for $t_{\text{LSTM}} = 0.1$ s.

a dropout-layer between the LSTM-layer and the dense-layer does not improve the result. The results in Fig. 19 also show that up to about 100 epochs, the influence of the chosen batch sizes is negligible. The number of required epochs is closely related to the training set size. Using a larger training set size with unchanged batch size leads to more iterations per epoch. Consequently the number of epochs can be reduced to achieve a comparable training result.

C. SIMULATION RESULTS

The parameters of the neural network used in the following are listed in Tab. 2. The resulting simulation results are presented below.

The comparison between measured and simulated vertical acceleration is shown in Fig. 20 exemplary for a vehicle speed of 40 km/h.

TABLE 2. Neural network parameters for training on rough road track data.

parameter	value
training set length	21 s
storage time	0.3 s
batch size	500
epochs	200

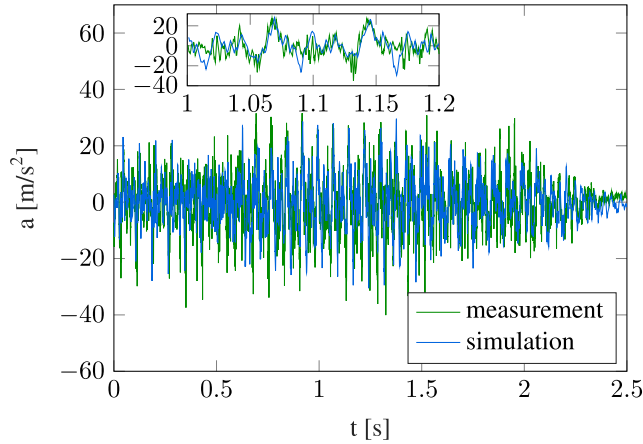


FIGURE 20. LSTM neural network prediction for 40 km/h.

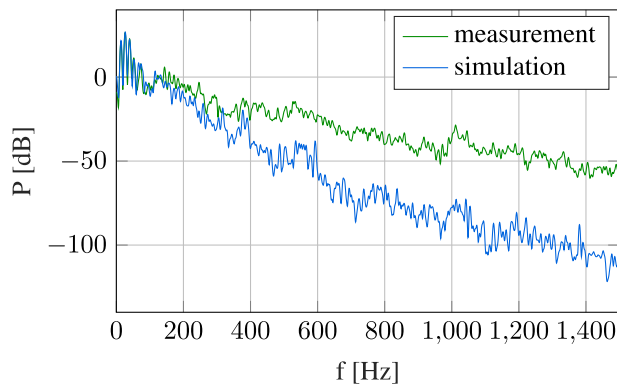


FIGURE 21. Power spectrum of LSTM neural network prediction for 40 km/h.

With the direct prediction of the system response with neural networks a much better result can be achieved than with the prediction with the ARX or ARMAX-model from Sec. III. Especially the power spectrum shown in Fig. 21 and Fig. 22 is simulated much better. It should be noted that nonlinear systems in the frequency domain are only considered in a linearized way. Here measurement and simulation agree very well. It follows that the energy of the acceleration signal is also simulated well. The higher discrepancy of the measured and simulated acceleration spectral density for high frequencies above 200 – 400 Hz in Fig. 21 has only a minor influence on the simulation result because the power of the signal in that signal range is much lower than the power for lower frequencies.

V. HYBRID COMBINATION OF NEURAL NETWORKS AND THE ARMAX-MODEL

A disadvantage of neural networks is the difficulty to validate them. Neural networks usually have to be considered

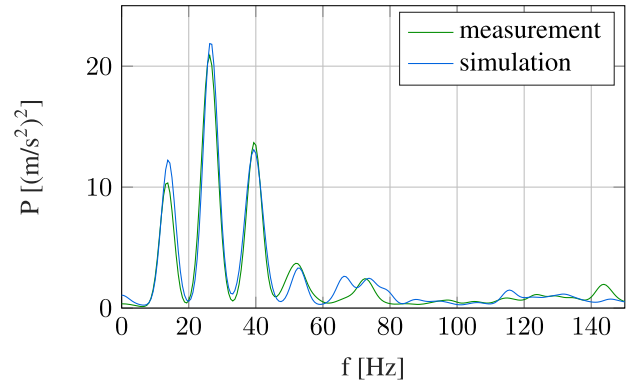


FIGURE 22. Power spectrum of LSTM neural network prediction for 40 km/h in detail.

as black box models. The generalization properties of neural networks can usually only be checked on the basis of validation data. A direct verification of the properties of the identified system based on the model structure and the identified model parameters is generally not possible. The validation of neural networks is currently the subject of research in various disciplines and limits the applicability of neural networks [26], [27].

An avoidance of this problem is possible by combining neural networks with linear transfer functions. Due to the good understanding of linear transfer functions, identified linear transfer functions can be validated comparatively easily and reliably. By combining neural networks with linear transfer functions, their higher accuracy can be used without having to give up the understanding of linear transfer functions completely. This approach is particularly suitable for weakly nonlinear systems, which can already be mapped comparatively well with linear transfer functions.

Here, two different approaches are possible. One approach is to identify both a linear transfer function and a nonlinear transfer function represented by neural networks. For the simulation the results of both transfer functions are compared. If the deviation between the simulation results is too large, the simulation result is rejected.

Another approach, which is used here, is to estimate the simulation error of an identified linear transfer function with neural networks. For this purpose, a linear transfer function is first identified with the training data, for example with the ARMAX model. Afterwards a prediction for all output signals in the training set is calculated using this transfer function. This data can be used to train a neural network in order to estimate the prediction error of the transfer function. The result is an improved estimation. This approach is referred here as the *hybrid-model* and is examined in more detail below.

1) PARAMETER

For the difference estimation the network structure and the parameters of the neural network for the direct estimation of the output signal are taken from section IV-B.

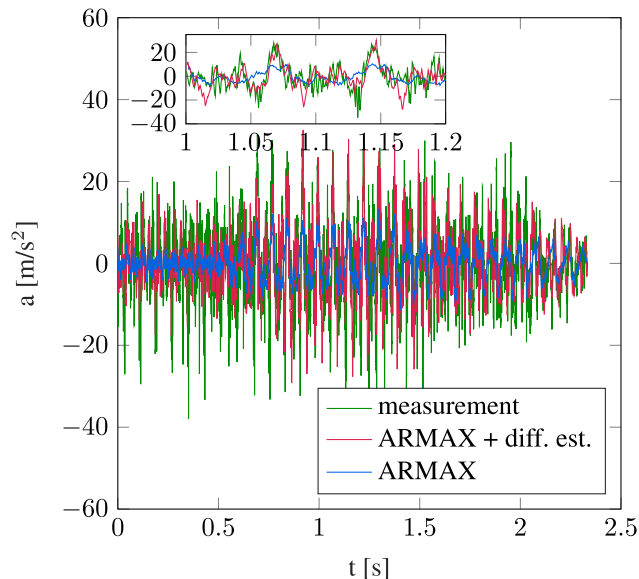
TABLE 3. Parameters and signals used for difference estimation.

	Parameters and signals
input signal ARMAX	wheel carrier: FL, BL
output signal ARMAX	battery: FL
parameter ARMAX	$n_a = 30, n_b = 100, n_c = 30$
input signal NN	wheel carrier: FL, BL
output signal NN	battery: FL – FL _{ARMAX}
parameter NN	$n_{batch} = 500, n_{epoch} = 200, t_{LSTM} = 0.3$ s

However, the output signal is not the battery acceleration, but the difference between simulated and measured battery acceleration. Exemplary, the output signal of the ARMAX model is used for the simulated battery acceleration. Tab. 3 lists the parameters and input and output signals used for the difference estimation.

2) SIMULATION RESULTS

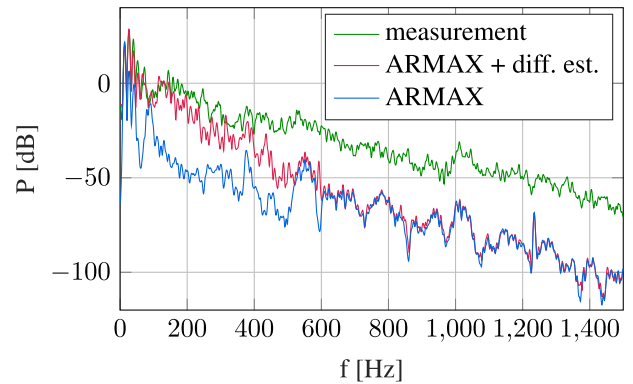
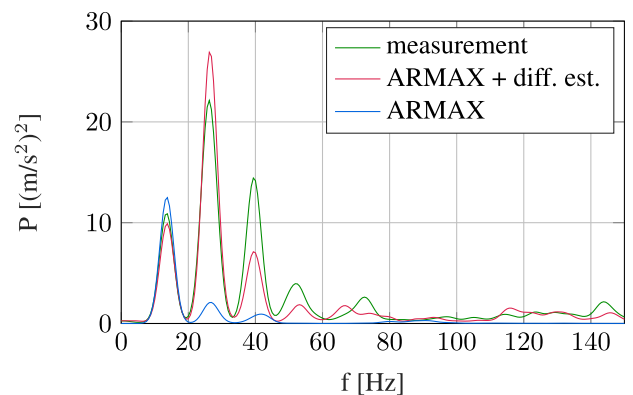
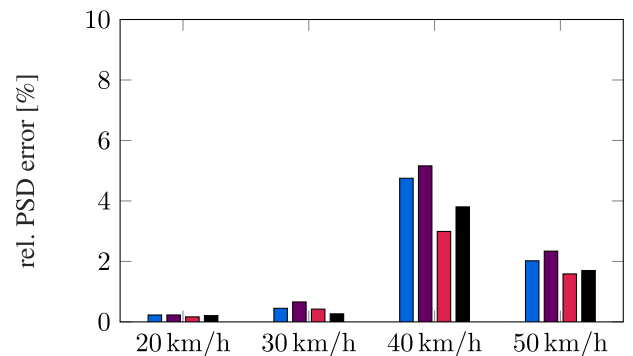
The results show a significantly better performance compared to the ARMAX results. However they are still worse than the neural network results. As can be seen by comparison of the power spectra from Figs. 22 and 25, as well as from the results in Figs. 20 and 23. The advantage of using a combination (of an established model and a machine learning approach) is that a part of the dynamical behavior is already predicted by a well-known model, which is here the ARMAX model. Then, the neural network model has to predict a smaller part of the considered dynamical system.

**FIGURE 23. Acceleration prediction with difference estimation for 40 km/h.**

3) COMPARISON OF SIMULATION RESULTS

In order to compare the performance of the different identified models the prediction error of the power spectrum of the simulation is introduced. It is defined by

$$\text{PSD error} = \int_{f=0}^{f_{\max}} |\text{PSD}(y_{\text{meas}}, f) - \text{PSD}(y_{\text{sim}}, f)| df \quad (4)$$

**FIGURE 24. Power spectrum of acceleration prediction with difference estimation for 40 km/h.****FIGURE 25. Power spectrum of acceleration prediction with difference estimation for 40 km/h in detail.****FIGURE 26. Comparison of the relative PSD errors of the ARX model (blue), the ARMAX model (purple), the neural networks (red) and the combination of the ARMAX model with the difference estimation (black) for different vehicle velocities.**

with the measurement data y_{meas} , the simulation y_{sim} and f_{\max} chosen as the sampling frequency of the simulated data. As we are not interested in an exact simulation of the time domain response, but in the frequency domain, this error allows a better representation of the performance of the identified model compared to classical error representations in time domain, like the mean square error. For different vehicle velocities the errors of the identification methods examined in this paper are shown in Fig. 26. As can be seen from the power

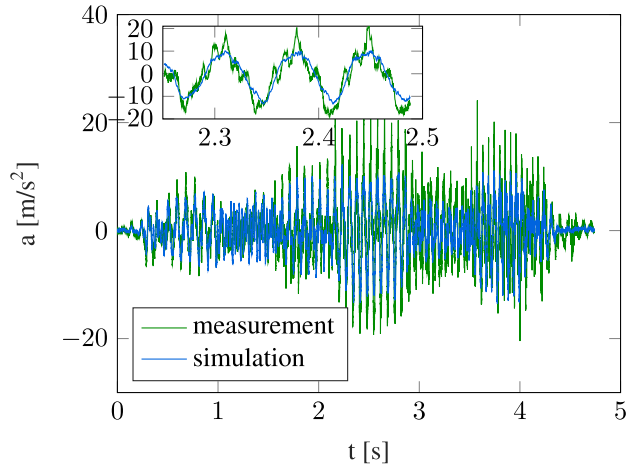


FIGURE 27. ARX prediction for 50 km/h on Golf data.

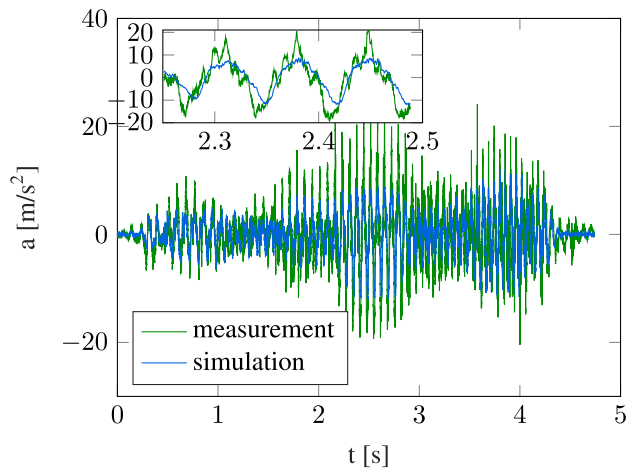


FIGURE 28. ARMAX prediction for 50 km/h on Golf data.

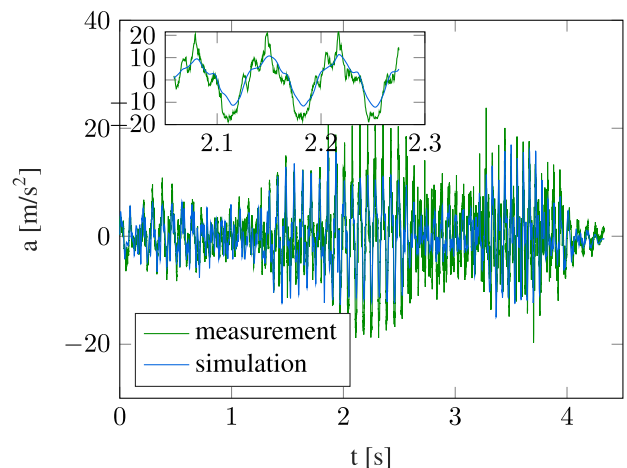


FIGURE 29. LSTM neural network prediction for 50 km/h on Golf data.

spectral density plots, too, the best result can be achieved using the LSTM-method. The difference estimation results in a slightly larger error while the ARX and the ARMAX model

TABLE 4. Parameters and signals used for estimation.

	Parameters and signals
training data	30, 40, 80, 100 km/h
validation data	50 km/h
input signal	wheel carrier: FL, FR, BL, BR
output signal	front battery front left
parameter ARX	$n_a = 300, n_b = 300$
parameter ARMAX	$n_a = 30, n_b = 100, n_c = 30$
parameter NN, diff. est.	$n_{batch} = 500, n_{epoch} = 200, t_{LSTM} = 0.3$ s

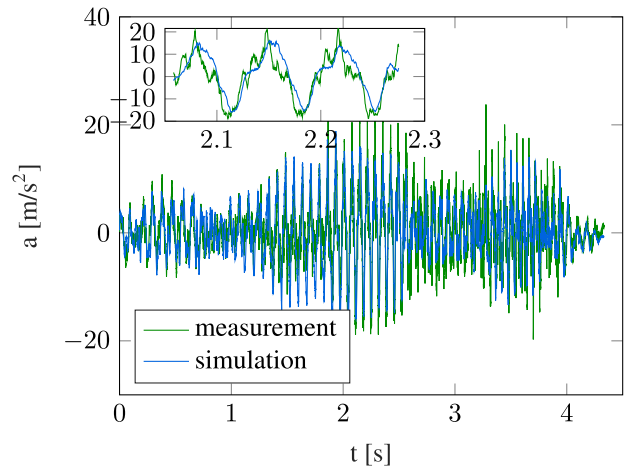


FIGURE 30. Hybrid-model prediction for 50 km/h on Golf data.

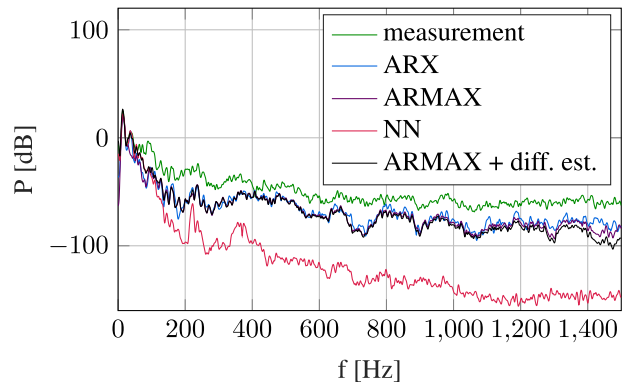


FIGURE 31. Power spectrum of different estimation methods for 50 km/h on Golf data.

lead to the largest errors. It can be clearly seen from Fig. 26 that the largest error occurs with a vehicle speed of 40 km/h in the validation set. In order to demonstrate conservative results for the Fiat 500e test car, we have chosen this speed in the figures from sections III and IV.

VI. RESULTS WITH ANOTHER VEHICLE

For validation we additionally applied the examined methods to acceleration data of a VW eGolf. The experimental setup to obtain the acceleration data was the same as for the Fiat 500e, but different vehicle velocities were used. The used training parameters are given in Tab. 4. They are identical to the parameters used with the Fiat 500e data. Only the vehicle velocities of the data sets are different.

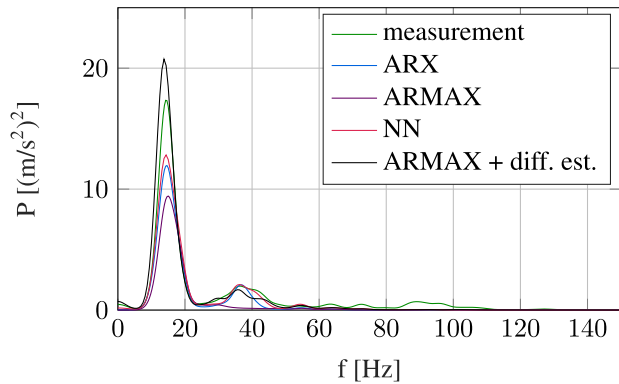


FIGURE 32. Power spectrum of different estimation methods for 50 km/h on Golf data in detail.

The simulation results in time domain for the different methods are shown in Figs. 27-30. The power spectra of the simulation results are compared in Fig. 31 and in Fig. 32 in detail. The simulation results on the eGolf data are comparable to the results of the Fiat 500e data. This shows that the examined methods as well as the chosen parameters can be used on different data sets corresponding to different vehicles. On new data sets the parameters used here are good initial values. However, a variation of parameters can be useful in order to improve the results.

VII. SUMMARY AND CONCLUSION

Different methods of predicting the load on the battery of electric vehicles were investigated on the basis of experimental vehicle measurements. For the system identification a neural network with a hidden LSTM layer and a dense layer as starting layer was used. Furthermore, a system identification with ARX and ARMAX models based on the Prediction Error Method was used. First, suitable model parameters for the transfer functions were determined. Thereby, the results of the ARMAX model were better compared to results of the ARX model. However, the parameters must be carefully determined if using the ARMAX model, since unfavorable parameter combinations can lead to unstable system behavior or convergence against comparatively poor local minima. The ARX model is much more robust and is always stable and does not converge against local minima.

Furthermore, a *hybrid-model* consisting of an LSTM neural network and the ARMAX model was studied, whereby the LSTM neural network was used to estimate the error between the ARMAX model results and the measurement data. It turned out that direct prediction of the system behavior by means of LSTM neural networks led to significantly better results than using linear system identification by means of ARX or ARMAX models. We have also obtained results for the *hybrid-model*, which has shown a slightly less well performance compared to the direct use of LSTM neural networks.

Every investigated method has some advantages and disadvantages that restrict its usage. An example for this is the

missing flexibility of LSTM networks regarding the sample rate of the measured signals. The network is trained on time series with a fixed step size. This cannot be changed afterwards, as well as the hyperparameters of the network as discussed in Sec. IV-A, which can prevent additional training of the network if available data is sampled with another rate. Another limitation of the shown methods is their black-box character. The models might predict the occurring accelerations at certain points of the car (where signals were measured) but not on the whole structure (which is possible with sophisticated approaches from the area of multibody dynamics). A solution to this problem might be interpolation between the predicted signals, which yet has to be investigated. In Tab. 5 the advantages and disadvantages of the examined methods are summarized. The ARX and the ARMAX model have a well understood structure and therefore can be easily and reliably validated. However, only linear system behavior can be modeled. Due to the better simulation results, direct estimation of the output signal by means of LSTM neural networks is recommended. However, the comparison of ARX and LSTM are useful for validation of LSTM neural network results.

TABLE 5. Advantages and disadvantages of the examined methods.

	Advantages	Disadvantages
ARX/ARMAX	Robust and well known	Lower accuracy than other two methods
LSTM	Best prediction results	worse in extrapolation, validation difficult
Hybrid	easier validation than for LSTM, preserves the better extrapolation behavior of linear model in combination with more accurate results, if difference between data and linear model is low	Lower performance than LSTM only

APPENDIX

1) ARX

One of the simplest methods of the prediction error method is the ARX model where *AR* refers to autoregressive and *X* for an exogenous input signal respectively. In the ARX model, the polynomials $C(z^{-1})$, $F(z^{-1})$ and $D(z^{-1})$ from Fig. 8 are assumed to be one.

The system dynamics in discrete form are given by

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + \frac{1}{A(z^{-1})}e(k) \quad (5)$$

with

$$A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_{n_a}z^{-n_a} \quad (6)$$

and the matrix polynomial

$$B(z^{-1}) = z^{-n_k} \left[b_{n_k} + b_{n_k+1}z^{-1} + \dots + b_{n_k+n_b-1}z^{-n_b+1} \right]. \quad (7)$$

The one-step-ahead prediction $\hat{y}(k)$ of the measured value $y(k)$ thus results in

$$\hat{y}(k|k-1, \theta) = \mathbf{B}(z^{-1})\mathbf{u}(k) + [1 - A(z^{-1})]y(k). \quad (8)$$

Since no poles occur here, the one-step-ahead prediction for any polynomial coefficients is stable.

With the parameter vector

$$\theta = [a_1 \dots a_{n_a} \quad \mathbf{b}_{n_k}^T \dots \mathbf{b}_{n_k+n_b-1}^T]^T \quad (9)$$

and the regression vector

$$\varphi(k, \theta) = [-y_{k-1} \dots -y_{k-n_a} \quad \mathbf{u}_{k-n_k} \dots \mathbf{u}_{k-n_k-n_b}]^T \quad (10)$$

the one-step ahead prediction of the system dynamics can be written as

$$\hat{y}(k|k-1, \theta) = \varphi(k)^T \theta \quad (11)$$

and the prediction error as

$$\varepsilon(k|k-1, \theta) = y(k) - \varphi(k)^T \theta. \quad (12)$$

From the mean squared error of the prediction the cost function

$$V_{\tilde{N}}(\theta, Z^{\tilde{N}}) = \frac{1}{2\tilde{N}} \sum_{k=p}^{\tilde{N}} \varepsilon(k|k-1, \theta)^2, \quad (13)$$

$$p = \max\{n_a, n_b\}, \quad (14)$$

$$N = \tilde{N} - p \quad (15)$$

is derived. For the ARX-model the minimization of this cost function can be simplified to the solution of the linear system of equations

$$\mathbf{R}\theta = \mathbf{f} \quad (16)$$

with

$$\mathbf{R} = \frac{1}{N} \sum_{k=p}^N \varphi(k)\varphi^T(k) \quad (17)$$

and

$$\mathbf{f} = \frac{1}{N} \sum_{k=p}^N \varphi(k)y(k) \quad (18)$$

[21, p. 203]. This system of equations is uniquely solvable if the input signal $u(k)$ is persistently exciting of sufficient order [21, p. 412].

2) ARMAX

An extension of the ARX model is the ARMAX-model after [28]. The polynomials $D(z^{-1})$ and $F(z^{-1})$ are chosen to be one, as with the ARX-model, but $C(z^{-1})$ is not. This allows a more complex model for the external disturbance v .

The system dynamics of the ARX-model from equation (5) are extended to

$$y(k) = \frac{\mathbf{B}(z^{-1})}{A(z^{-1})}\mathbf{u}(k) + \frac{C(z^{-1})}{A(z^{-1})}e(k) \quad (19)$$

with

$$C(z^{-1}) = 1 + c_1z^{-1} + \dots + c_{n_c}z^{-n_c}. \quad (20)$$

The one-step-ahead prediction thus results in

$$\hat{y}(k|k-1, \theta) = \mathbf{B}(z^{-1})\mathbf{u}(k) + [1 - A(z^{-1})]y(k) + [1 - C(z^{-1})]\varepsilon(k). \quad (21)$$

This can also be written as

$$C(z^{-k})\hat{y}(k, \theta) = \mathbf{B}(z^{-k})\mathbf{u}(k) + [C(z^{-k}) - A(z^{-k})]y(k). \quad (22)$$

The zeros of the nominator polynomial $C(z^{-1})$ are poles of the ARMAX-model. To ensure stability, they must lie outside the unit circle. With the parameter vector

$$\theta = [a_1 \dots a_{n_a} \quad b_{n_k} \dots b_{n_k+n_b-1} \quad c_1 \dots c_{n_c}]^T \quad (23)$$

and the regression vector

$$\varphi(k, \theta) = [-y_{k-1} \dots -y_{k-n_a} \quad \mathbf{u}_{k-n_k} \dots \mathbf{u}_{k-n_k-n_b} \quad \varepsilon_{k-1, \theta} \dots \varepsilon_{k-n_c, \theta}]^T \quad (24)$$

the one-step-ahead prediction and the prediction error can be written as

$$\hat{y}(k|k-1, \theta) = \varphi^T(k, \theta)\theta \quad (25)$$

and

$$\varepsilon(k|k-1, \theta) = y(k) - \varphi(k, \theta)^T \theta. \quad (26)$$

For an efficient solution of the optimization problem minimizing the cost function from equation (13)-(15) the calculation of the gradient $\psi(k, \theta)$ of $\varphi(k, \theta)$ is necessary.

According to [21, p. 329 ff.] the gradient results from equation (22) with

$$C(z^{-1})\frac{\partial}{\partial a_i}\hat{y}(k, \theta) = -z^{-i}y(k) \quad (27)$$

$$C(z^{-1})\frac{\partial}{\partial b_i}\hat{y}(k, \theta) = -z^{-i-n_k}\mathbf{u}(k) \quad (28)$$

$$z^{-i}\hat{y}(k, \theta) + C(z^{-1})\frac{\partial}{\partial c_i}\hat{y}(k, \theta) = -z^{-i}y(k) \quad (29)$$

in

$$\psi(k, \theta) = C^{-1}(z^{-1})\varphi(k, \theta). \quad (30)$$

With this gradient the optimization problem according to [21, p. 327] can be solved with the Levenberg-Marquardt algorithm [29, p. 111 ff.].

REFERENCES

- [1] V. Ruiz, A. Pfrang, A. Kriston, N. Omar, P. Van den Bossche, and L. Boon-Brett, "A review of international abuse testing standards and regulations for lithium ion batteries in electric and hybrid electric vehicles," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 1427–1452, Jan. 2018.
- [2] B. Liu, Y. Jia, C. Yuan, L. Wang, X. Gao, S. Yin, and J. Xu, "Safety issues and mechanisms of lithium-ion battery cell upon mechanical abusive loading: A review," *Energy Storage Mater.*, vol. 24, pp. 85–112, Jan. 2020.
- [3] J. M. Hooper and J. Marco, "Understanding vibration frequencies experienced by electric vehicle batteries," in *Proc. IET Hybrid Electr. Vehicles Conf. (HEVC)*. Piscataway, NJ, USA, 2013, pp. 9.1–9.1.
- [4] J. M. Hooper and J. Marco, "Defining a representative vibration durability test for electric vehicle (EV) rechargeable energy storage systems (RESS)," *World Electr. Vehicle J.*, vol. 8, no. 2, pp. 327–338, Jun. 2016.
- [5] D. Ren, X. Liu, X. Feng, L. Lu, M. Ouyang, J. Li, and X. He, "Model-based thermal runaway prediction of lithium-ion batteries from kinetics analysis of cell components," *Appl. Energy*, vol. 228, pp. 633–644, Oct. 2018.
- [6] L. Zhang, Z. Ning, H. Peng, Z. Mu, and C. Sun, "Effects of vibration on the electrical performance of lithium-ion cells based on mathematical statistics," *Appl. Sci.*, vol. 7, no. 8, p. 802, Aug. 2017.
- [7] *Electrically Propelled Road Vehicles—Test Specification for Lithium-Ion Traction Battery Packs and Systems—Part 1: High Power Applications*. Standard ISO 12405-1, International Organization for Standardization, 2011.
- [8] C. Lalanne, *Mechanical Vibration & Shock*. Hoboken, NJ, USA: Wiley, 2002.
- [9] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
- [10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012.
- [12] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in *Neural Nets WIRN Vietri-01*. London, U.K.: Springer, 2002, pp. 193–200.
- [13] H. Luo, M. Huang, and Z. Zhou, "Integration of multi-Gaussian fitting and LSTM neural networks for health monitoring of an automotive suspension component," *J. Sound Vib.*, vol. 428, pp. 87–103, Aug. 2018.
- [14] A. ElSaid, B. Wild, J. Higgins, and T. Desell, "Using LSTM recurrent neural networks to predict excess vibration events in aircraft engines," in *Proc. IEEE 12th Int. Conf. e-Sci. (e-Sci.)*, Oct. 2016, pp. 260–269.
- [15] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.
- [16] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.
- [17] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017.
- [18] J. Tang, L. Li, Z. Hu, and F. Liu, "Short-term traffic flow prediction considering spatio-temporal correlation: A hybrid model combining type-2 fuzzy C-means and artificial neural network," *IEEE Access*, vol. 7, pp. 101009–101018, 2019.
- [19] J. Tang, X. Chen, Z. Hu, F. Zong, C. Han, and L. Li, "Traffic flow prediction based on combination of support vector machine and data denoising schemes," *Phys. A, Stat. Mech. Appl.*, vol. 534, Nov. 2019, Art. no. 120642.
- [20] X. Yang, Y. Zou, J. Tang, J. Liang, and M. Ijaz, "Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models," *J. Adv. Transp.*, vol. 2020, pp. 1–16, Jan. 2020.
- [21] L. Ljung, *System Identification: Theory for the User* (Prentice-Hall Information and System Sciences Series), 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
- [22] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
- [23] M. Abadi. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: <https://www.tensorflow.org/>
- [24] Y. Wang, "A new concept using LSTM neural networks for dynamic system identification," in *Proc. Amer. Control Conf.*, Piscataway, NJ, USA, 2017, pp. 5324–5329.
- [25] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. IEEE Conf. Publication*, Sep. 1999, vol. 2, no. 470, pp. 850–855.
- [26] B. J. Taylor, *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*. New York, NY, USA: Springer, 2006.
- [27] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2018.
- [28] K.-J. Åström and B. Torsten, "Numerical identification of linear dynamic systems from normal operating records," *IFAC Proc. Volumes*, vol. 2, no. 2, pp. 96–111, Sep. 1965.
- [29] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, PA, USA: SIAM, 1996.



LEO DOSTAL received the Diploma degree in industrial mathematics from the University of Hamburg, in 2007, and the Dr. rer. nat. degree in applied mathematics from the Hamburg University of Technology, in 2015. Since 2015, he holds a postdoctoral position at the Institute of Mechanics and Ocean Engineering, Hamburg University of Technology. Since 2019, he is also a member of the Fields-CQAM Laboratory for Inference and Prediction, University of Waterloo. His research

interests include inference and prediction, machine learning methods, as well as random nonlinear dynamical systems and their control. His awards and honors include fellowships of the German Academic Exchange Service (DAAD).



HELGE GROSSERT received the bachelor's degree in general engineering science and the master's degree in theoretical mechanical engineering from the Hamburg University of Technology, Hamburg, Germany, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Institute of Mechanics and Ocean Engineering. His research interests include identification of mechanical systems, the modeling of such systems as rigid and flexible multibody systems and the combination of such models with techniques from machine learning to obtain universal differential equations.



DANIEL A. DUECKER (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering and the M.B.A. degree in technology management from the Hamburg University of Technology, in 2012, 2015, and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Institute of Mechanics and Ocean Engineering. During his studies, he spent two semester at the University of California, Berkeley, where besides course work, he conducted research at the Berkeley Center for Control and Identification under the supervision of Prof. Andy Packard. He is currently a Research Associate with the Institute of Mechanics and Ocean Engineering, Hamburg University of Technology. His research interests include dynamics, sensor fusion, and control of micro autonomous underwater robotic systems, as well as the application of machine learning in robotics. His awards and honors include fellowships of the German Academic Exchange Service (DAAD), the Lufthansa Technik AG scholarship program, and the Germany Academic Foundation.



MALTE GRUBE received the B.S. degree in mechanical engineering from the Hamburg University of Technology, in 2018, where he is currently pursuing the M.S. degree in theoretical mechanical engineering. His research interests include soft robotics and machine learning.



DANIEL KREUTER received the M.S. (Dipl.-Ing) and Ph.D. (Dr.-Ing) degrees in mechanical engineering from the Dresden University of Technology, in 2007 and 2015, respectively. He has been working with the Corporate Research, BOSCH Group, since 2016. His research interest includes vibration and vibration fatigue.



KAI SANDMANN received the B.Eng. degree in mechanical engineering from UAS Rhein Main, Rüsselsheim, Germany, the M.Sc. degree in mechanical engineering from UAS Karlsruhe, Germany, and the Ph.D. degree in mechanical engineering from TU Dresden, in 2017. Since 2017, he has been a Research Engineer with the Corporate Research, Robert Bosch GmbH. His main research topic is the analysis of field load distributions of vehicles as well as development of simulation methods for generating the distributions of mechanical loads.



BENJAMIN ZILLMANN received the Dipl.-Ing. and Ph.D. degrees in mechanical engineering from Technische Universität Chemnitz, Germany, in 2009 and 2017, respectively. From 2010 to 2015, he was a Research Assistant with Technische Universität Chemnitz, Institute of Materials Science and Engineering. Since 2015, he has been a Researcher with the Robert Bosch GmbH, Cooperative Sector Research and Advance Engineering, Renningen, Germany. In 2015, he was acknowledged for outstanding scientific effort from Zwick Roell Group.



ROBERT SEIFRIED received the Dipl.-Ing degree in mechanical engineering from University of Stuttgart, in 2001, and the Ph.D. degree in mechanical engineering from the University of Stuttgart, in 2005. From 2006 to 2007, he held a postdoctoral position with the Department of Mechanical Engineering, University of California, Berkeley, with the Prof. R. Horowitz. From 2008 to 2013, he was an Assistant Professor with the Cluster of Excellence Simulation Technology (SimTech), University of Stuttgart. From 2013 to 2014, he was a Full Professor and a Chair for dynamical systems in vehicle technology with the University of Siegen. Since 2014, he has been a Full Professor and the Head of the Institute of Mechanics and Ocean Engineering, Hamburg University of Technology.

• • •