INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.154

Machine Learning Based Simulation for Design Space Exploration

O. Bleisinger [1,✉], C. Malek [1] and S. Holbach [2]

[1] Fraunhofer IESE, Germany, [2] BorgWarner Turbo Systems Engineering GmbH, Germany

✉ oliver.bleisinger@iese.fraunhofer.de

Abstract

Design of software in the automotive domain often involves simulation to allow early software parametrization. Modeling complex systems or components impacted by the software in an analytical way can be time-consuming, require domain knowledge and executing the analytical models can result in high computational effort. In specific applications, these challenges can be overcome by applying machine learning based simulation. This contribution presents results of a case study in which powertrain components are modeled data-driven with artificial neural networks to support design space exploration

Keywords: artificial intelligence (AI), simulation-based design, data-driven design

1. Introduction

Since late detection and correction of design errors within the product development cycle leads to high costs, early testing is common practice (Jain et al., 2016). Especially for embedded software providing dedicated functions within mechatronic systems, testing has high demands on the consideration of the behaviour of its operational context. This is even more applicable if highly automated functions of modern system classes are involved. To cut costs for early testing and provide a virtual testbed for design and parametrization of embedded software, analytical simulation models can be used to analyse the impact on the electrical and mechanical system components and optimize the overall mechatronic system. Especially when considering different design alternatives with respect to specific design goals, analytical modelling and simulation for Design Space Exploration (DSE) is well established in the field of virtual product development.

The main challenges in utilizing analytical simulation models for electrical and mechanical system components are the need of specific domain knowledge, time-consuming modelling and in certain applications also the high demand on computational resources (Miralaya et al., 2016). Therefore, software developers willing to practice DSE-approaches might not possess necessary analytical simulation models to perform early testing of the overall mechatronic system their software will be embedded. Still, since critical features and functions are increasingly implemented by software, tracking physical effect chains by simulation are gaining importance.

To tackle the challenge of creating testbeds for virtual validation purposes, a data based approach can be beneficial in certain circumstances. This is e.g. the case for development projects, where measurement data from the operational phase of a previous system generation is available to apply machine learning (ML) based behaviour modelling (see Figure 1).

A concrete use case in the development of automotive embedded software this applies to is the implementation of energy management systems. Especially, if considering the impact of the software parametrization on the energy consumption of hybrid or battery electric vehicles, ML based modelling and simulation can provide a virtual testbed. Since the mathematical models describing relevant physical effect chains are non-linear (Dinnawi et al., 2014), optimizing the parametrization of the energy management software in an analytical way is not a trivial task. Therefore, in industry mostly iterative simulations with given load cycles are performed to tune the parameters (see Figure 2, left side). By variation of the parametrization, different design alternatives are investigated and a DSE is performed with respect to the energy consumption of the resulting hybrid or battery electric vehicle as seen in Figure 2, right side (Sabzewari et al., 2016). This leads to high computational effort for the iterative execution of the analytical simulation models, if the models are even available to software developers.

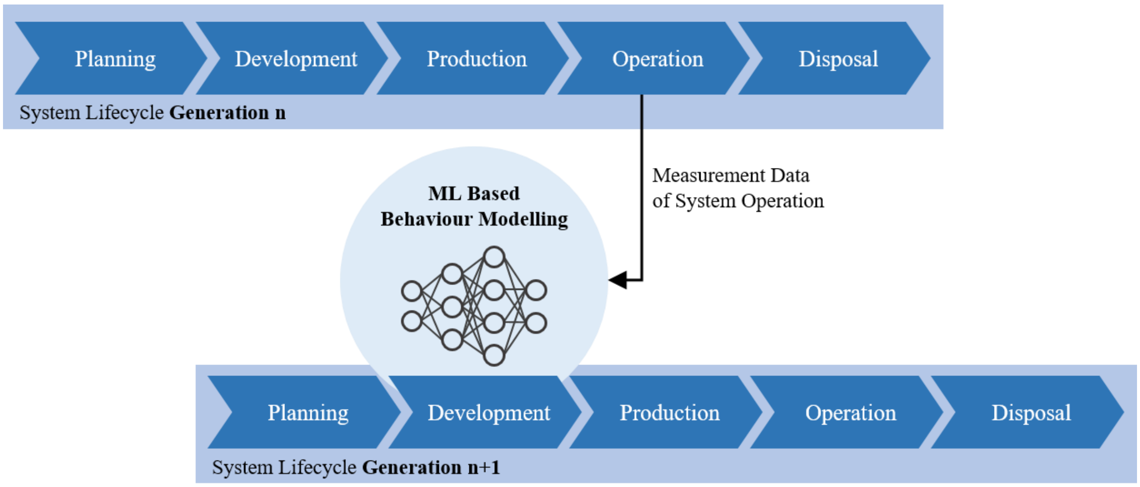

Figure 1. Linking operational and development phase of different system generations within the system lifecycle for ML based behaviour modelling

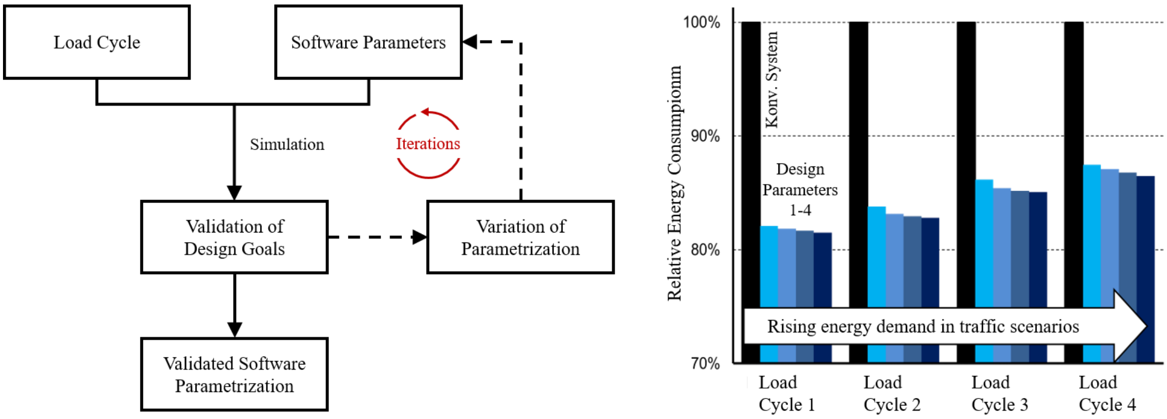

Figure 2. Optimization of software parametrization of an energy management system of a hybrid vehicle by iterative simulation and evaluation of energy consumption as design goals

Besides the advantage that the software developer does not need specific domain knowledge for model generation, another advantage of using ML based models is the fast execution as formerly already demonstrated for DRAM simulations (Feldmann et al., 2020). The work in our contribution describes how various ML algorithms are used to implement data-driven modelling and simulation of various electrical components used in the powertrain of hybrid and battery electric vehicles, namely the battery storage, the power inverter and the electric machine. One main challenge for the creation of suitable ML algorithms lies in the fact, that the considered electrical components have a dynamic system behaviour and therefore run and predict time series data. For this purpose, feedforward networks (FNNs) and recurrent neural networks (RNNs) in the form of gated recurrent units were investigated in particular in a case study to evaluate their capability of modelling and simulating electrical powertrain components.
In this publication the relevant state of the art of behaviour modelling for simulation is briefly presented (section 2) before similar work with respect to the use case of this publication is discussed in section 3. This is followed by the introduction of the case study using the aforementioned electrical

https://doi.org/10.1017/pds.2022.154 Published online by Cambridge University Press

powertrain components, which are modelled by artificial neural networks (section 4). After technical results are presented and evaluated (section 5), a short summary and outlook will be given in section 6.

2. State of the Art in Modelling for Simulation

Of course, there is already a variety of well-established methods, techniques and procedures for behaviour modelling for the purpose of simulating the properties of dynamic systems used in vehicle development. In the following two generalized concepts for modelling are presented in order to distinguish them from the machine learning based approach. The most widespread modelling approach is the manual analytical behaviour modelling by mathematical-physical equations (theoretical modelling) in the sense of a white-box model for components and involved processes. The modelling effort is dependent on the desired model detailing, but usually can be considered very high for complex systems and components, even when using pre-configured libraries. The higher the desired accuracy of the model, the higher are the demands on the previous knowledge of the modeler, who must know the physical effect chains of the system or component in order to identify relevant influences on the system properties to be simulated (domain knowledge). Therefore, the required domain knowledge for the theoretical modelling approach is normally quite high (Mughal 2016).

One concept for behaviour modelling of dynamic systems and components, where hardly any domain knowledge is required, is the so-called experimental modelling approach. In this case the behaviour model is defined by measuring input and output values of a system (e.g. on a test bench) and assuming general mathematical relationships (e.g. polynomial equations). For relatively simple systems or components this is achieved with sufficient accuracy, whereas for complex systems and without domain knowledge, the model quality is usually very low. Advanced techniques of system identification from the engineering branch of systems theory provide a remedy, but depending on the application, significant additional effort is required. Another advantage of experimental modelling is that a behaviour model is obtained which can be executed very quickly. Moreover, the black box approach protects the knowledge about internal structure, physical interactions and, if applicable, existing system or component functions without additional measures (Layer and Tomczyk, 2010).

In contrast, the ML based modelling approach only requires limited domain knowledge, enables fast execution of the resulting models (Mirilaya et al., 2016) and therefore offers potential for the optimization of activities in virtual validation of embedded software when considering its operational context. One prerequisite is sufficient data of the operational phase of the systems or components modelled, e.g. from a previous similar system generation. Following, some of the most relevant related scientific work for the case study as well as for the ML based modelling approach is presented.

3. Related Scientific Work

Taking the specific use case of this publication, namely the estimation of the state of charge (SOC) of hybrid vehicles and battery electric vehicles (section 4) for virtual validation into account, one of the most relevant related work is focused on the estimation of the SOC with RNNs (Bockrath et al., 2019). Using LSTMs (long-short-term memory) as a special type of RNNs, the mentioned publication shows the strong potential of RNNs for the SOC-prediction of lithium ion batteries for energy storage applications. Using LSTMs it was shown, that the ML based simulation approach outperformed classical approaches for SOC-prediction via analytical models, namely an analytical equivalent circuit model combined with an Extended Kalman Filter, with respect to the achieved accuracy. The same statement holds for another investigation regarding the use case of SOC prediction with RNNs (Chemali et al., 2018), but specializing in the use and prediction of measurement data of real driving cycles instead of test bench data under well-defined conditions in laboratory as used in Bockrath et al., 2019.

However, in the related work LSTMs were utilized whereas in this contribution gated recurrent units are investigated more closely. Furthermore, the mentioned publications only investigated the ML based simulation approaches for lithium ion batteries models and did not focus on modelling the whole physical effect chains from wheel to battery of hybrid or battery electric vehicles, which is necessary for the optimization and validation of energy management software (see section 4).

https://doi.org/10.1017/pds.2022.154 Published online by Cambridge University Press

In addition, the aspect of execution time was not investigated in the mentioned contributions. A recent contribution showing the potential of simulation speed-up by ML based simulation in contrast to analytical simulation was presented on the DATE conference 2020 (Feldmann et al., 2020). In the special use case of this publication a neural network based simulation approach for DRAM was investigated. Dependent on the load data/cycle provided to the models, the simulation speed-up goes up to factor 10 while the accuracy is considered around 95%. With respect to DSE support via ML, there are recent investigations as presented in Roy et al., 2018. These are rather targeted on the utilizing extrapolation capabilities of neural network to speed up DSE. These examples show the potential of ML based simulation for the acceleration of simulations in comparison to the usage of analytical models. Still, the presented use cases are very different from the case study discussed in the following section.

4. Modelling Approach

As described in section 1, the starting point of the investigation of the potential of ML based simulation is the need for virtual testbeds for embedded software development. Especially when considering the briefly depicted workflow in Figure 2 (left side), iterative simulation is necessary to optimize parameters of the energy management software of battery electric vehicles (BEVs) and hybrid vehicles. Exemplary software parameters in hybrid vehicles can for example represent a SOC-threshold to start the internal combustion engine if the battery capacity is low, engine torque- and SOC-thresholds to decouple the internal combustion engine while coasting or SOC-adaptive thresholds related to recuperation in different modes. These are just brief examples showing the need for SOC-estimation to test the parametrization of embedded energy management software, whereas it is obvious that for the estimation of the achievable range of BEVs the SOC prediction through simulation is crucial too.

Mostly, the software design is validated against design goals (e.g. the achievable range of a BEV) with respect to given load cycles. The load cycles in the automotive context can often be considered speed profiles, e.g. according to the “Worldwide Harmonized Light Vehicles Test Cycle Class 3” (WLTC Class 3), which applies to most passenger cars. Based on these speed profiles, it is possible to derive load cycles represented by rotational speed and torque necessary to follow the speed profiles. An example for measurement data of a speed profile and derivation of the engines rotational speed for a passenger car with automatic transmission is shown in Figure 3. For our specific case study, we are considering a BEV, which is why the ML-based simulation of the main electrical powertrain components, as shown in Figure 4 (right side), is investigated.

Since a BEV is analysed, the electrical powertrain consists mainly of the electric machine (E-machine), inverter and battery (Faraz et al., 2021). Taking into account, that for a given vehicle design the load cycle to follow a given speed profile can be represented as time series data of the rotational speed n_{EM} and the released torque M_{EM} of the E-machine on the wheel side, these are chosen as inputs of the model. The E-machine type that is used in this case study is the three-phase surface mounted permanent magnet synchronous machine. This permanent magnet synchronous machine has advantages over other machine types like induction machines or reluctance machines. Permanent magnet synchronous machines have a high power density, high efficiency, low noise and the inverter does not have to provide reactive power for magnetization (An and Binder 2016). The outputs of the E-machine model will be the peak voltage $\hat{u}_{EM}$, peak current $\hat{\imath}_{EM}$ as well as the phase shift φ_{EM} (see Figure 4, right side).

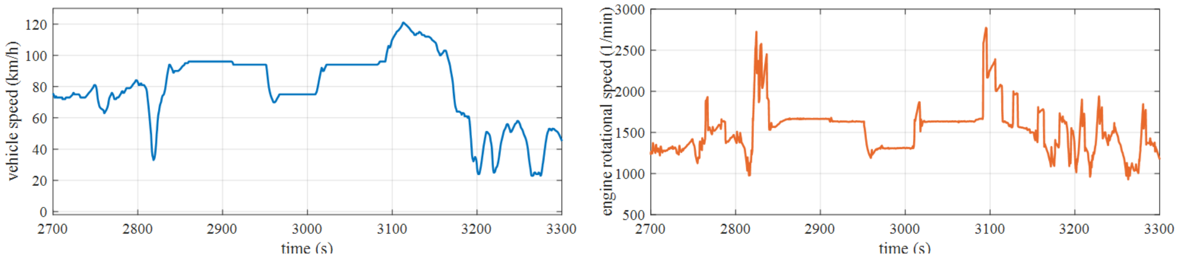

Figure 3. Exemplary measurement data of a speed profile as load cycle and derived engine rotational speed for a given automotive vehicle design

https://doi.org/10.1017/pds.2022.154 Published online by Cambridge University Press

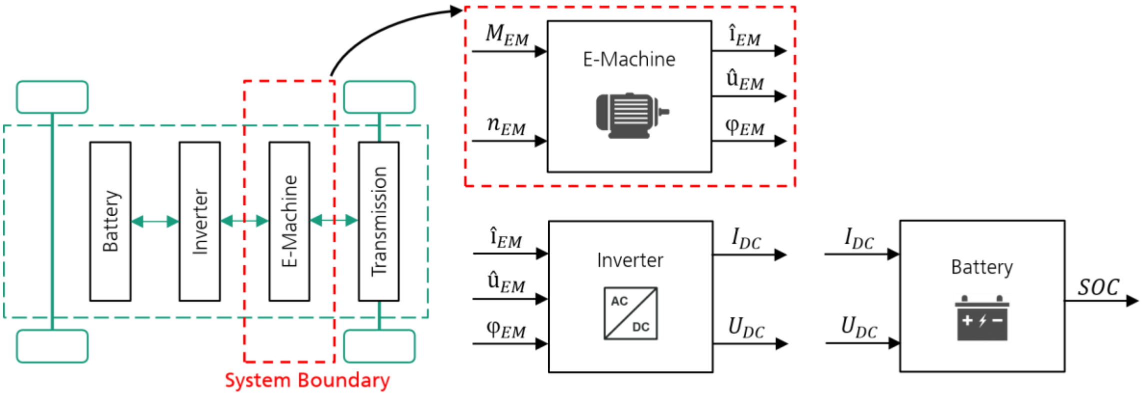

Figure 4. Typical architecture of a battery electric vehicle with respect to the main electric components investigated in the presented case study

In the BEV's topology the inverter is used to transform DC to AC power during motor operation and also AC to DC power during recuperation phases. Furthermore, the inverter is necessary to control the E-machine as the frequency and the amplitude of the applied three phase sinusoidal voltage can be manipulated by using the inverter. The output of the E-machine model is used as input of the inverter model. As the battery requires a DC voltage, the output of the inverter model is a DC voltage U_{DC} and a DC current I_{DC}, which are consequently the inputs of the battery model.
In general, there are different types of lithium ion batteries for usage in BEVs, which differentiate mostly in the material used for the electrodes – e.g. LiFePO4/graphite (LFP) and LiNiMnCoO2/graphite (NMC). The reason for the usage of lithium ion batteries in hybrid vehicles and BEVs is the high cell voltage as well as the high power density (Masias 2018). In our case study the battery AMP20M1HD A from the producer A123 is used, which is based on the LFP technology. This said, the output of our battery model is the SOC, whereas it is obvious that a coulomb approach will be represented by the artificial neural network model, since no further information, e.g. of the battery's temperature, is used as input of the battery model. Besides the described single component models (Figure 4, right side), which represent dedicated physical components, two other models are investigated. These two models represent compositions of different physical components, as depicted in Figure 5.

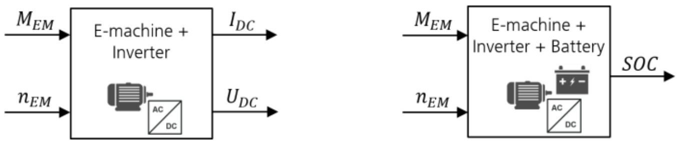

Figure 5. Depiction of the inputs and outputs of the composed electrical component models for the machine learning based simulation approach

5. Results and evaluation

Based on the defined input and output parameters shown in Figure 4 and Figure 5, FNN and RNN models are implemented, trained and optimized based on given data. The implementation is performed in Keras, a high level deep-learning API, which provides the possibility to access three open source deep learning libraries: Tensorflow, Theano or MXNET. For the further implementation Tensorflow is used as backend. The training data for the neural networks consist of different runs of the whole powertrain, while the necessary input and output data of all three components have been logged. The test data is kept separately from the training data and is not used for the training of the FNNs and RNNs. Regarding the different runs of the whole powertrain it is important to note, that these are performed in driving mode (discharge of the battery) as well as recuperation mode (charging of the battery). Therefore, the training and test data sets are synthetically sequentially stitched together from data of different runs, which can result in harsh transitions between data logged from different runs and combined to one data set. For the training of the ML-based simulation models, around 540.000 data point for each different in- and output parameter are used. The data has a sampling rate of around 10 kHz. Further 6.000 data points are used as

https://doi.org/10.1017/pds.2022.154 Published online by Cambridge University Press

test data for the evaluation of the model accuracy. For the evaluation, it is important to keep in mind that each training of neural networks can lead to slightly different accuracies of the models, e.g. due to the random initialization of the weights at the beginning of the training. Finding optimized configurations of hyperparameters for neural networks is a complex task, so only a short summary of the investigated values is provided in Table 1. However, it should be noted, that for optimization the stochastic gradient descent algorithm (SGD) and adaptive moment estimation (ADAM) were applied as well as an extensive grid search to find suitable configurations for each model.

Table 1. Investigated hyperparameters for the architectural design of the neural network models

Hyperparameter	Varied Values	Architecture
Number of hidden layers	2, 3, 4	RNN, FNN
Number of cells per layer	10, 20, 30	RNN, FNN
Activation function	SELU, ELU	FNN
Optimizer	SGD, Adam	RNN, FNN
Learning rate	0.001, 0.0001	RNN, FNN
Initialization	lecun normal, he normal	FNN
Length of sequence	5, 10, 20	RNN

In addition, to evaluate the neural network models with respect to their accuracy to predict output values suitable error measures must be taken into account. According to Géron 2019 the Root Mean Square Error (RMSE) should be the first choice for regression problems because of the fact that RMSE performs well for errors following a rather Gaussian distribution (Chai and Draxler 2014). Although in most regression problems this error distribution is assumed, the RMSE weights large errors higher than the mean average error (MAE), due to the square calculation. If a large number of outliners occurs, this higher weighting could distort the RMSE. In this case and in case of rather uniformly distributed errors, the MAE is more suitable for the evaluation of the accuracy of the neural network models. To be able to describe the deviations of the models regardless their error distribution and for better comparability independent of the physical units of the model outputs, both metrics are applied in their percentage representations (RMSPE, MAPE) according to (1) and (2).

$$\text{RMSPE}\ (\hat{y}) = 100\% \cdot \sqrt{\frac{\sum_{n=1}^{N}(y(n) - \hat{y}(n))^2}{\sum_{n=1}^{N} y(n)^2}} \tag{1}$$

$$\text{MAPE}\ (\hat{y}) = 100\% \cdot \frac{1}{N}\sum_{n=1}^{N}\left|\frac{y(n) - \hat{y}(n)}{y(n)}\right| \tag{2}$$

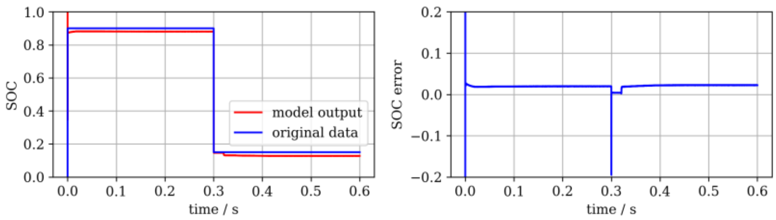

Figure 6. Comparison of original data and model output for the trained FNN battery model

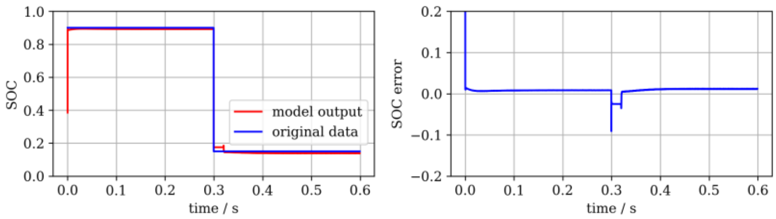

Figure 7. Comparison of original data and model output for the trained RNN of the battery

https://doi.org/10.1017/pds.2022.154 Published online by Cambridge University Press

As seen by Figure 8 in comparison to Figure 9, the E-machine simulation with the FNN shows a better performance for the peak current, whereas for the peak voltage and phase shift the RNN is more accurate. For the inverter the DC current, that is simulated by the FNN during recuperation (negative current), shows a big deviation from the original data, whereas the DC voltage fits well (see Figure 10). The RNN model in Figure 11, however, is capable of simulating the DC current of the inverter more precisely. Due to the FNN's inability to appropriately reproduce the DC current during recuperation, the FNN model that has been trained in this case study fails in this application. Still it should be mentioned one again, that each training run/experiment might lead to slightly different conclusion due to a randomized initialization of the weights of the neural nets at the beginning of the training.

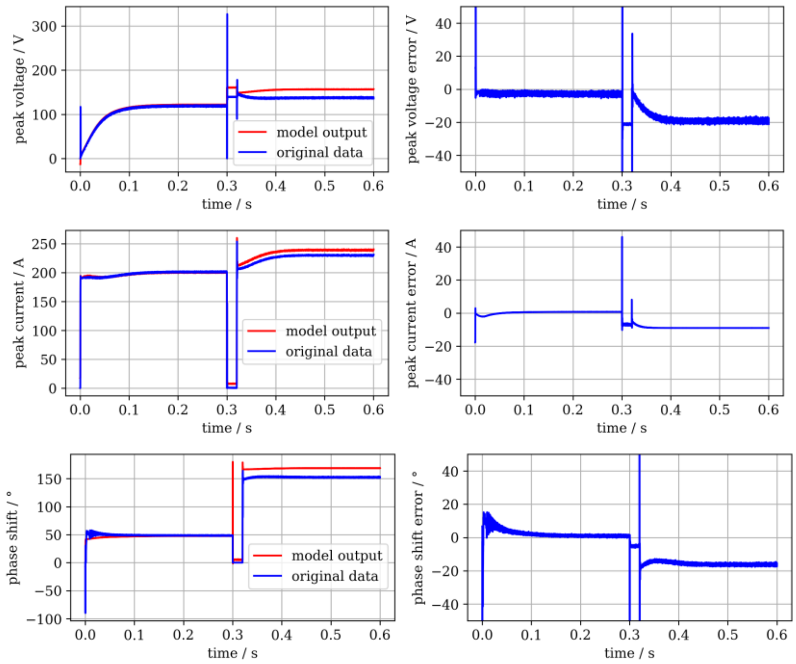

Figure 8. Results for the trained feedforward neural network model of the electric machine

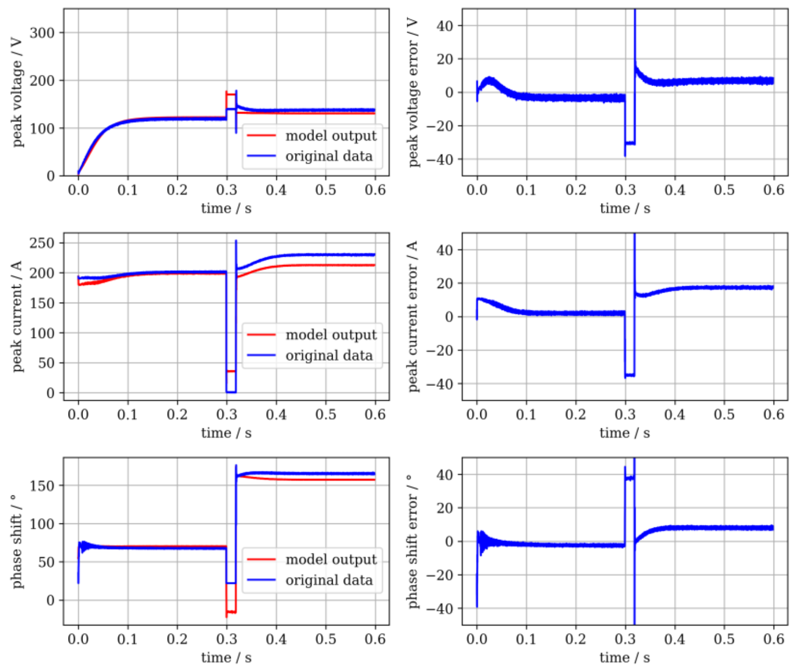

Figure 9. Results for the trained recurrent neural network model of the electric machine

https://doi.org/10.1017/pds.2022.154 Published online by Cambridge University Press

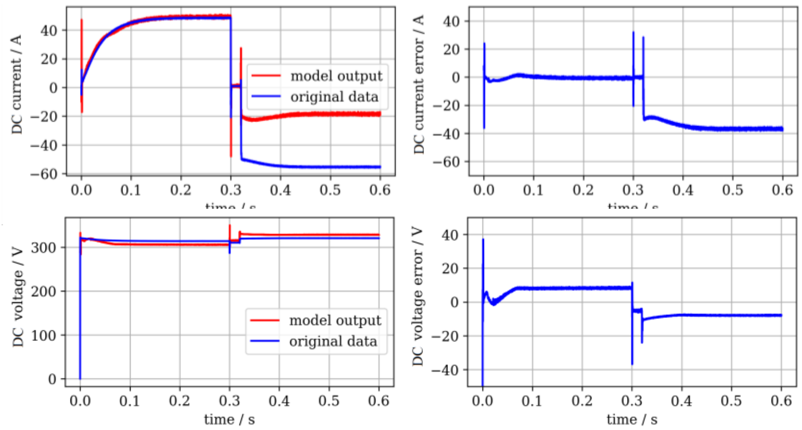

Figure 10. Results of the trained feedforward neural network model of the inverter

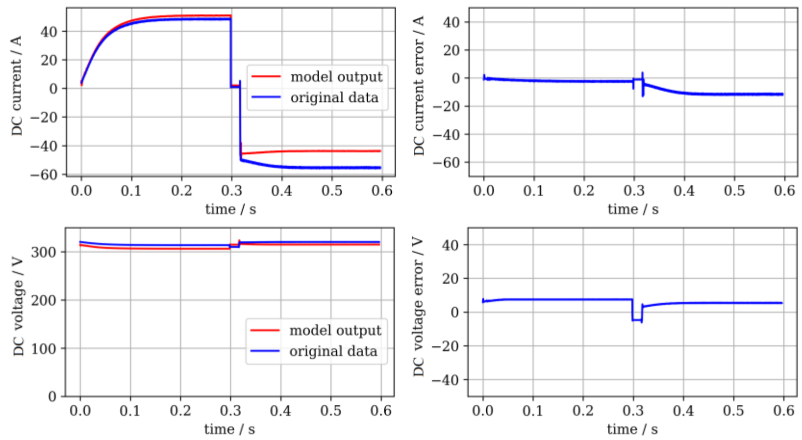

Figure 11. Results of the trained recurrent neural network model of the inverter

Regarding the battery model (Figure 6 and Figure 7), both types of neural networks, FNN as well as RNN, simulate the battery's SOC in a very precise manner. Table 2 summarizes calculated error measures for one of the best training runs for the different single component models, whereby it should not be forgotten, that the test data includes some harsh transitions between data of the driving mode (battery discharge) and recuperation mode (battery charge). Although the harsh transitions in the test data sets only generate synthetic errors for a very short duration, these can influence the calculation of the error measures as well as the randomized initialization of the weights of the neural networks.

Table 2. Output error of single component models for the best training run

	NN Type	Output	RMSPE	MAPE
E-machine	FNN	$\hat{\imath}_{EM}$	3.06 %	0.00318 %
		$\hat{u}_{EM}$	7.57 %	0.00925 %
		φ_{EM}	4.76 %	0.00371 %
	RNN	$\hat{\imath}_{EM}$	4.40 %	4.25 %
		$\hat{u}_{EM}$	10.60 %	0.01668 %
		φ_{EM}	4.16 %	0.00181 %
Inverter	FNN	I_{DC}	65.19 %	11.00 %
		U_{DC}	2.83 %	0.00387 %
	RNN	I_{DC}	6.18 %	6.50 %
		U_{DC}	2.05 %	0.00341 %
Battery	FNN	*SOC*	3.78 %	4.06 %
	RNN	*SOC*	1.92 %	4.03 %

https://doi.org/10.1017/pds.2022.154 Published online by Cambridge University Press

As seen in Table 3, it is obvious that using FNNs as well as RNNs for the composition models, especially for the model representing the E-machine, inverter and battery all in one neural network, lacks accuracy significantly. The problem at hand for the composition models is the missing information in the input values needed for correct reproduction of the output values. An optimization of the chosen ML algorithms and their hyperparameters will not provide better results as the lack of information is the data's property. For the combined E-machine and inverter model, this means that according to physical laws the mechanical and electrical power on the input and output of the model must be the same, when neglecting the power losses. Nevertheless, the relation of DC current to DC voltage cannot be mapped solely from the mechanical input values properly. The DC values and also their exact relation are not sensitive to the mechanical input values. Due to this lacking information, the DC values cannot be reproduced accurately with the chosen input values. The same is true for the composition of all three electrical components (E-machine, inverter and battery) as the battery's SOC is also not directly sensitive to mechanical values.
In summary, regarding the single component models the RNN architecture performs more accurately than the FNN architecture when predicting dynamic system behaviour. The reason for the higher performance of the RNN architecture might be due to the input sequences. The RNN models do not only use input values at one point in time, but additionally the information of former points in time, which is according to Wang et al., 2019 better suited for dynamic system behaviour.

Table 3. Output error of composition component models for the best training run

	NN Type	Output	RMSPE	MAPE
E-machine, inverter composition	FNN	I_{DC}	10.40 %	12.39 %
		U_{DC}	5.39 %	0.00672 %
	RNN	I_{DC}	13.03 %	8.40 %
		U_{DC}	1.68 %	0.00275 %
E-machine, inverter, battery composition	FNN	*SOC*	48.15 %	88.18 %
	RNN	*SOC*	54.62 %	125.00 %

6. Summary and Outlook

In this contribution the application of a machine learning based simulation approach for electrical powertrain components of a battery electric vehicles was presented. For the electric machine, inverter and battery, which were modelled by feedforward and recurrent artificial neural networks, the achieved accuracy is between 93 to 99% according to the mean average percentage error in the best training runs. Taking the accuracy into account, it is possible to estimate the state of charge precisely to validate embedded software design, specifically the parametrization of energy management software for battery electric vehicles. The presented case study shows that the machine learning based simulation approach is suitable to create virtual testbeds of physical systems for Design Space Exploration.
Not yet investigated, was the actual speed-up of machine learning based simulation in comparison to execution time of similar analytical models. In addition, the need for domain knowledge when applying the machine learning based modelling approach in comparison to an analytical modelling approach was not investigated. The definition of measures for comparison of required domain knowledge is not trivial and therefore not applied in this publication. Furthermore, in the presented case study only a coulomb based estimation of the state of charge of the battery via neural networks was considered. Taking further information as inputs of the neural networks into account, e.g. the cell temperature of the battery, might lead to higher accuracy of the models. Due to insufficient data while performing the case study, both mentioned aspects will be investigated in upcoming research.
In addition, it was seen, that by combining the different electrical components into one composition model, the accuracy of the neural network's prediction was reduced dramatically. Therefore, investigating the opposite, namely breaking down the single component models into smaller single composed neural network models would be interesting. However, since this would require special knowledge of the electrical engineering domains, the machine learning based simulation approach would become unpractical for typical software developers and so the further breakdown of the component models is not further investigated. Still, the results of the composed electrical models show, that system decomposition must be done carefully if applying machine learning based modelling.

One special application with respect to embedded systems design, which was not yet considered, is the use of the machine learning based approach for model predictive control. Since it can be assumed that the execution time of neural network simulations is lower than the execution time of complex non-linear analytical models (Mirilaya et al., 2016), the use of neural network models as state prediction in model predictive control will be investigated. Another benefit will be the online adaptability of the models as discussed in recent publications about learning based model predictive control (Kabzan et al., 2019).

References

S. Jain and H. Joshi, "Impact of early testing on cost, reliability and release time," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2016, pp. 318-322, https://dx.doi.org/10.1109/ICRITO.2016.7784973.

S. S. Miriyala, P. Devi Pantula, S. Majumdar and K. Mitra, "Enabling online optimization and control of complex models through smart surrogates based on ANNs," 2016 Indian Control Conference (ICC), 2016, pp. 214-221, https://dx.doi.org/10.1109/INDIANCC.2016.7441131.

R. Dinnawi, D. Fares, R. Chedid, S. Karaki and R. A. Jabr, "Optimized energy management system for fuel cell hybrid vehicles," MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference, 2014, pp. 97-102, https://dx.doi.org/10.1109/MELCON.2014.6820514.

K. Sabzewari, O. Bleisinger, “Decoupling of Auxiliary Consumers from the Powertrain as Possibility for Energy Savings”, in Proceedings of the 4th Commercial Vehicle Technology Symposium (CVT 2016), Shaker, Düren, 2016, pp. 208-218, ISBN: 978-3-8440-4229-0.

J. Feldmann, K. Kraft, L. Steiner, N. Wehn and M. Jung, "Fast and Accurate DRAM Simulation: Can we Further Accelerate it?," 2020 Design, Automation & Test in Europe Conference & Exhibition, 2020, pp. 364-369, https://dx.doi.org/10.23919/DATE48585.2020.9116275.

K. Roy, H. T. Mert and M. Swaminathan, "Preliminary Application of Deep Learning to Design Space Exploration," 2018 *IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, 2018, pp. 1-3, https://dx.doi.org/10.1109/EDAPS.2018.8680888.

A. M. Mughal, “Analytical Formulation by Bond Graph Modeling.,” in Real Time Modeling, Simulation and Control of Dynamical Systems, Springer, Cham., 2016, pp. 63-84, https://dx.doi.org/10.1007/978-3-319-33906-1_5.

E. Layer, K. Tomczyk, “Model Development.,” in Measurements, Modelling and Simulation of Dynamic Systems, Springer, Berlin, Heidelberg, 2010, pp. 83-126, https://dx.doi.org/10.1007/978-3-642-04588-2_4.

S. Bockrath, A. Rosskopf, S. Koffel, S. Waldhör, K. Srivastava and V. R. H. Lorentz, "State of Charge Estimation using Recurrent Neural Networks with Long Short-Term Memory for Lithium-Ion Batteries," IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, 2019, pp. 2507-2511, https://dx.doi.org/10.1109/IECON.2019.8926815.

E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed and A. Emadi, "Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-ion Batteries," in IEEE Transactions on Industrial Electronics, vol. 65, no. 8, pp. 6730-6739, Aug. 2018, https://dx.doi.org/10.1109/TIE.2017.2787586.

A. Faraz, A. Ambikapathy, S. Thangavel, K. Logavani, G. Arun Prasad, “Battery Electric Vehicles (BEVs).,” in N. Patel, A. K. Bhoi, S. Padmanaban, J. B. Holm-Nielsen (eds), Electric Vehicles, Green Energy and Technology, Springer, Singapore, 2021, pp. 137-160, https://dx.doi.org/10.1007/978-981-15-9251-5_8.

J. An, A. Binder, “Permanent magnet synchronous machine design for hybrid electric cars with double e-motor and range extender,” in Elektrotech. Inftech. 133, 2016, pp. 65–72, https://dx.doi.org/10.1007/s00502-016-0397-7.

A. Masias, “Lithium-Ion Battery Design for Transportation.,” in G. Pistoia, B. Liaw (eds), Behaviour of Lithium-Ion Batteries in Electric Vehicles, Green Energy and Technology, Springer, Cham., 2018, pp. 1-33, https://dx.doi.org/10.1007/978-3-319-69950-9_1.

A. Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.).,” O'Reilly, Heidelberg, 2019, ISBN: 978-1-4920-3264-9.

T. Chai, R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,” Geoscientific Model Development 7(3), 2014, pp. 1247 1250, https://dx.doi.org/10.5194/gmd-7-1247-2014, 2014.

Chi-Hsu Wang, Pin-Cheng Chen, Ping-Zong Lin and Tsu-Tian Lee, "A dynamic neural network model for nonlinear system identification," 2009 IEEE International Conference on Information Reuse & Integration, 2009, pp. 440-441, https://dx.doi.org/10.1109/IRI.2009.5211647.

J. Kabzan, L. Hewing, A. Liniger and M. N. Zeilinger, "Learning-Based Model Predictive Control for Autonomous Racing," in IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3363-3370, Oct. 2019, https://dx.doi.org/10.1109/LRA.2019.2926677.