Modelling of a large-format lithium-iron-phosphate-based lithium-ion battery cell with neural ordinary differential equations

Jennifer Brucker*, Wolfgang G. Bessler, and Rainer Gasper

Institute of Sustainable Energy Systems – Offenburg University of Applied Sciences
jennifer.brucker@hs-offenburg.de

Abstract. Lithium-ion batteries show strongly nonlinear behaviour regarding the battery current and state of charge. Therefore, the modelling of lithium-ion batteries is complex. Combining physical and data-driven models in a grey-box model can simplify the modelling. Our focus is on using neural networks, especially neural ordinary differential equations, for grey-box modelling of lithium-ion batteries. A simple equivalent circuit model serves as a basis for the grey-box model. Unknown parameters and dependencies are then replaced by learnable parameters and neural networks. We use experimental full-cycle data and data from pulse tests of a lithium iron phosphate cell to train the model. Finally, we test the model against two dynamic load profiles: one consisting of half cycles and one dynamic load profile representing a home-storage system. The dynamic response of the battery is well captured by the model.

Keywords: neural ordinary differential equations; grey-box model; equivalent circuit model; lithium-ion battery.

1 Introduction

Lithium-ion batteries play an important role in our everyday life: They supply portable devices such as smartphones with electrical energy, are a key technology for electromobility, and are used in stationary applications such as home-storage systems. Battery models are used to predict the dynamic voltage and current behaviour and to monitor internal states. There are many different types of models with different accuracy and complexity [1,2]. We summarize a grey-box (GB) modelling approach that uses an equivalent circuit model (ECM) as a basis and was first introduced in Ref. [3].

GB modelling is the combination of white-box (WB) and black-box (BB) modelling techniques. BB models learn relations between inputs and outputs of systems based on data. Neural networks belong to the BB modelling techniques. WB models use mathematical equations derived from prior physical, chemical or engineering knowledge to describe the behaviour of the underlying system [4,5,6,7].

Many approaches in current research use neural networks to model lithium-ion batteries. The authors of Ref. [8] and Ref. [9] use neural networks to predict the state of charge (SOC) of a battery whereas the authors of Ref. [10] focus on the state of health (SOH) prediction. In Refs. [11] and [12] neural ordinary differential equations (NODEs), a special form of neural network, are used for GB modelling of lithium-ion batteries. The authors of Ref. [11] model aging effects such as solid electrolyte interface formation, lithium plating, and active material isolation as well as the increase in the internal resistance. The final deviation of the physical model and measurement results is approximated with NODEs. In contrast to that, we built a GB model of a lithium-ion battery based on a simple ECM in our previous work [12]. Herein, the voltage drop across the included resistor–capacitor (RC) circuit is represented by NODEs.

We continued our previous work from Ref. [12] by considering the battery dynamics. The methods and results can be found in Ref. [3] in detail and are summarized here. We used additional data from charging and discharging with pulsed currents during the training of the GB model to identify the time constant of the fast battery dynamics. Neither temperature dependencies nor aging effects have been considered.

We applied our GB modelling approach to a large-format $180\,\mathrm{Ah}$ prismatic commercial lithium-ion cell with lithium iron phosphate (LFP)/graphite chemistry. The experimental properties of the cell have been investigated in great detail in Ref. [13]. The state diagnosis of LFP cells is challenging because of their flat, plateau-like open circuit voltage curve and charge-discharge voltage hysteresis [14]. One of our goals is to investigate the applicability of our GB modelling approach to this type of cell.

2 Methodology

In this section, we give a short introduction to NODEs. Furthermore, we present a simple ECM and show how to derive a GB model from it. The section is complemented with initialization, normalization, and training techniques. Finally, we give insights into the experimental basis.

2.1 Background: Neural Ordinary Differential Equations

The interested reader can find a detailed overview of neural networks in Ref. [15].

In Ref. [16] residual neural networks (ResNets) are introduced. They overcome problems with the degradation of the training loss with an increasing number of hidden layers in deep neural networks by adding additional short-cut connections to feedforward networks. These short-cut connections allow the direct addition of the input of a neuron to its output. ResNets can be used for time series prediction.

The state transformation from layer t to layer $t+1$ in a ResNet follows the recursive formula [16]

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \boldsymbol{f}\left(\boldsymbol{z}_t, \boldsymbol{\theta}_t\right), \quad t = 0, ..., T-1 \tag{1}$$

where, $\boldsymbol{z}_t \in \mathbb{R}^d$ is the vector of the hidden states at layer t, $\boldsymbol{\theta}_t$ the learned parameters of layer t and $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$ a learnable function. Herein, the vector $\boldsymbol{\theta}_t$ of learned parameters summarizes the learned weights and biases. The explicit Euler discretization of the initial value problem [17,18,19,20,21,22,23]

$$\frac{\mathrm{d}\boldsymbol{z}\left(t\right)}{\mathrm{d}t} = \boldsymbol{f}\left(\boldsymbol{z}\left(t\right), t, \boldsymbol{\theta}\right), \quad \boldsymbol{z}\left(0\right) = \boldsymbol{z}_0 \tag{2}$$

can be derived through parameter sharing across the layers ($\boldsymbol{\theta}_t = \boldsymbol{\theta}$ for $t = 0, ..., T-1$). The right-hand side of the differential equation according to Equation (2) is represented by the neural network $\boldsymbol{f}$. Therefore, it is called NODE. Starting from the initial state $\boldsymbol{z}\left(0\right)$ a differential equation solver delivers the output state $\boldsymbol{z}\left(T\right)$ [17,20,21,23].

The differential equation according to Equation (2) is generalized to consider external variables $\boldsymbol{u}\left(t\right)$ [12]:

$$\frac{\mathrm{d}\boldsymbol{z}\left(t\right)}{\mathrm{d}t} = f\left(\boldsymbol{z}\left(t\right), \boldsymbol{u}\left(t\right), t, \boldsymbol{\theta}\right), \quad \boldsymbol{z}\left(0\right) = \boldsymbol{z}_0. \tag{3}$$

As stated in Refs. [12] and [24] one can combine NODEs with differential equations derived from prior physical knowledge in one equation system to build a GB model.

2.2 Equivalent circuit modelling

Besides physical modelling, it is a common approach to use ECMs to describe the dynamics of lithium-ion batteries. Due to their simplicity, ECMs are often used for SOC and SOH predictions [25,26,27].

As in Ref. [12], we used a simple ECM as a basis for GB modelling of the battery. The chosen ECM is shown in Figure 1. It is composed of an SOC-dependent voltage source, a hysteresis voltage drop, a serial resistor, and one RC circuit. We include parameter dependencies on battery current and SOC. The ECM can be described by the following equation system:

$$\frac{\mathrm{dSOC}}{\mathrm{d}t} = -\frac{1}{C_{\mathrm{bat}}} i_{\mathrm{bat}} \tag{4a}$$

$$\frac{\mathrm{d}v_{\mathrm{RC1}}}{\mathrm{d}t} = \frac{1}{C_1} \cdot \left(i_{\mathrm{bat}} - \frac{1}{R_1\left(\mathrm{SOC}, i_{\mathrm{bat}}\right)} \cdot v_{\mathrm{RC1}} \right) \tag{4b}$$

$$v_{\mathrm{bat}} = v_{\mathrm{OC}}\left(\mathrm{SOC}\right) - v_{\mathrm{hys}} \cdot \mathrm{sgn}\left(i_{\mathrm{bat}}\right) - R_{\mathrm{S}} \cdot i_{\mathrm{bat}} - v_{\mathrm{RC1}}, \tag{4c}$$

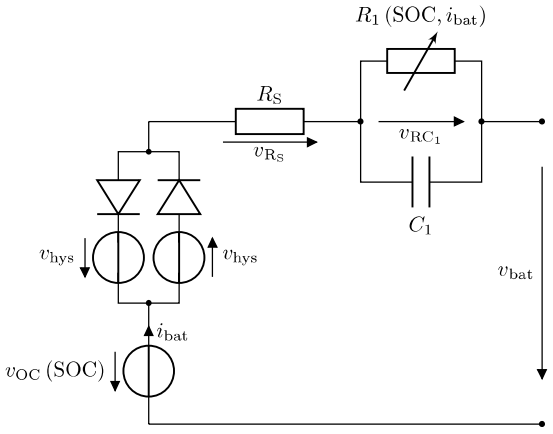


Fig. 1: ECM of a battery consisting of an SOC-dependent voltage source, a hysteresis voltage drop, a serial resistor, and an RC circuit. [3]. CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

with the battery capacity C_{bat}, the hysteresis voltage drop v_{hys}, the serial resistance R_{S}, the charge-transfer resistance $R_1\left(\mathrm{SOC}, i_{\mathrm{bat}}\right)$ in the RC circuit depending on SOC and battery current, the double-layer capacitance C_1 and the SOC-dependent open-circuit voltage (OCV) $v_{\mathrm{OC}}\left(\mathrm{SOC}\right)$. The battery voltage v_{bat} is the output of the dynamic system and the battery current i_{bat} serves as the external input. The current for battery discharge is defined as positive, and the current for battery charge is defined as negative.

2.3 Grey-box modelling

The ECM according to equation system (4) served as a basis for GB modelling. We replaced unknown parameters and dependencies with learnable parameters and neural networks. As they are unknown or only approximately known, we considered the battery capacity C_{bat} in Equation (4a), the double-layer capacitance C_1 in Equation (4b), the hysteresis voltage drop v_{hys}, and the serial ohmic resistance R_{S} in Equation (4c) as learnable parameters. The charge-transfer resistance and its dependency on battery current and SOC are also unknown. As observed experimentally [13], it may have different characteristics during charging and discharging. Due to this fact, we represented the charge-transfer resistance R_1 through two learnable functions or rather two neural networks. Depending on the sign of the battery current, one of these learnable functions is chosen; at zero current ($i_{\mathrm{bat}} = 0\,\mathrm{A}$) the mean is taken.

In the output equation 4c we had to establish a link between OCV and SOC. Therefore, we derived $v_{\mathrm{OC}}\left(\mathrm{SOC}\right)$ from dedicated quasi-OCV measurements (cf. [13]). Overall, we derived the following GB model:

$$\frac{\mathrm{dSOC}}{\mathrm{d}t} = -\frac{1}{\omega_0} i_{\mathrm{bat}} \tag{5a}$$

$$\frac{\mathrm{d}v_{\mathrm{RC1}}}{\mathrm{d}t} = \frac{1}{\omega_1} \cdot \left(i - \frac{1}{R_1\left(\mathrm{SOC}, i_{\mathrm{bat}}\right)} \cdot v_{\mathrm{RC1}} \right) \tag{5b}$$

$$R_1\left(\mathrm{SOC}, i_{\mathrm{bat}}\right) = \begin{cases} f\left(\mathrm{SOC}, i_{\mathrm{bat}}, \boldsymbol{\theta}_f\right) & \forall\, i_{\mathrm{bat}} < 0 \\ g\left(\mathrm{SOC}, i_{\mathrm{bat}}, \boldsymbol{\theta}_g\right) & \forall\, i_{\mathrm{bat}} > 0 \\ \frac{1}{2}\left(f\left(\mathrm{SOC}, i_{\mathrm{bat}}, \boldsymbol{\theta}_f\right) + g\left(\mathrm{SOC}, i_{\mathrm{bat}}, \boldsymbol{\theta}_g\right)\right) & \text{else} \end{cases} \tag{5c}$$

$$v_{\mathrm{bat}} = v_{\mathrm{OC}}\left(\mathrm{SOC}\right) - \omega_2 \cdot \mathrm{sgn}\left(i_{\mathrm{bat}}\right) - \omega_3 \cdot i_{\mathrm{bat}} - v_{\mathrm{RC1}} \tag{5d}$$

with the learnable parameters ω_0, ω_1, ω_2, and ω_3. The functions f and g represent feedforward networks with their respective learnable parameters $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_g$. We chose neural networks with one hidden layer and rectified linear unit (ReLU) activation for f and g. We varied the number of neurons in the hidden layer between 10 and 300. The SOC and the battery current serve as inputs to the neural networks and the ohmic resistance R_1 is the output.

The GB model combines physics-based ordinary differential equations (ODEs) and machine-learning-based NODEs in one equation system. This equation system is solved numerically within a single framework.

2.4 Experiments

We applied the proposed GB modelling approach to a single lithium-ion battery cell of the Chinese manufacturer CALB, model CA180FI. The cell is shown in Figure 2. The large-format prismatic cell with a nominal capacity of $180\,\mathrm{Ah}$ uses LFP at the positive electrode and graphite at the negative electrode. Experimental measurements were performed under a controlled laboratory environment (climate chamber CTS $40/200$ Li) using a battery cycler (Biologic VMP3). Details can be found in Refs. [3] and [13].

We used the measurement data from Ref. [3]. In detail, we used the constant current constant voltage (CCCV) charge and discharge curves with different C-rates of $0.1\,\mathrm{C}$, $0.28\,\mathrm{C}$, and $1\,\mathrm{C}$ (corresponding to $18\,\mathrm{A}$, $50\,\mathrm{A}$, and $180\,\mathrm{A}$, respectively) during the CC phase. The upper and lower cutoff voltages were $3.65\,\mathrm{V}$ and $2.5\,\mathrm{V}$, respectively, and a cut-off current of the CV phase of $\mathrm{C}/20$ was used. Additionally, one charge and one discharge curve with a pulsed current were recorded: During $50\,\mathrm{A}$ CC operation, every two SOC-percent the current was reduced to $25\,\mathrm{A}$ for $30\,\mathrm{s}$. Furthermore, two independent measurements that we used for model testing were carried out. Firstly, the cell was cycled with $50\,\mathrm{A}$ between $25\,\%$ and $75\,\%$ SOC, in the following referred to as half cycles. Secondly, measurements with a dynamic load profile over $48\,\mathrm{h}$ representing a home storage battery in a single-family house were performed. This synthetic load profile was taken from Ref. [28] and downscaled to the energy of the present cell. During all measurements the ambient temperature was kept at $T = 25\,^{\circ}\mathrm{C}$. To reduce the number of measurement points per data series, measurement values were deleted if the current only changed by $|\Delta i_{\mathrm{bat}}| \leq 0.5\,\mathrm{A}$ and the measured voltage changed by $|\Delta u_{\mathrm{bat}}| \leq 0.5\,\mathrm{mV}$. The measurement data were recorded and used as voltage versus time and current versus time series. The experiments are described in more detail in the original contribution [3].

Fig. 2: Photograph of the CALB cell

2.5 Normalization and initialization

It is recommended to scale the inputs of neural networks to simplify the training process. The input variables should be transformed in a way that their average over the complete training data set is close to zero. Additionally, the input variables should have similar value ranges [29].

We scaled all inputs to values between -1 and 1 to achieve a similar value range as for the SOC which is in the range of 0 to 1. The output values of the neural networks were normalized to the same value range. As we did not use different learning rates for different parameters, we also scaled the learnable parameters. For this reason, we had to estimate their order of magnitude. Additionally, the learnable parameters have to be initialized at the beginning. Therefore, we took a closer look at the training data to estimate the unknown parameters.

To find for example a good initial value for the battery capacity or rather the learnable parameter ω_0, one can calculate the charge throughput for a whole charging or discharging process by integrating the measured current over time. The other parameters can be estimated by analyzing the voltage response of the battery following a current step or by analyzing the overpotential. The interested reader is referred to Ref. [3] for more details.

2.6 Simulation and Optimization Methodology

We implemented the GB model in Python (version 3.7.6). Tensor computing and automatic differentiation were performed with the open-source machine learning framework PyTorch (version 1.9.0) [30]. We used the torchdiffeq library (version 0.2.1) [31] which builds on PyTorch to solve the ODEs and to backpropagate through the solutions. In detail, we used the Dopri8 method with an absolute tolerance of 10^{-5} and a relative tolerance of 10^{-3} to solve the differential equations and the standard odeint method from torchdiffeq for backpropagation. Finally, the loss minimization was carried out with an Adam optimizer.

2.7 Training

Only a small data base was available for training the GB model. Therefore, we decided to split the training into two consecutive steps: First, we used the CCCV data to train a simplified, static version of the GB model. Afterward, the battery dynamics were taken into account by considering the data from charging and discharging with a pulsed current.

In detail, in the first step we converted the differential equation (5b) into an algebraic equation by neglecting the double-layer capacitance:

$$v_{\mathrm{RC1}} = R_1\left(\mathrm{SOC}, i_{\mathrm{bat}}\right) \cdot i_{\mathrm{bat}}. \tag{6}$$

The resulting simplified GB model was trained with the time series data from CCCV charging and discharging with different C-rates. We initialized and scaled the learnable parameters ω_0, ω_2, and ω_3 and the neural networks f and g of the simplified model as described above and in Ref. [3] in detail. To solve the differential equation (5a) we had to provide an initial value for the SOC. We assumed that the battery was in equilibrium at the beginning of each time series and therefore the battery voltage was equal to the OCV. We used the inverted OCV(SOC) curve to obtain the respective SOC value. The learning rate of the Adam optimizer was decaying between 10^{-2} and 10^{-3}. The loss function was defined as the root mean squared error (RMSE) between the simulated battery voltage and the measured battery voltage. Additionally, approximated SOC values lower than 0 or higher than 1 were penalized. The total number of training epochs was varied as a hyperparameter of the training process. The optimization steps were carried out with stochastic gradient descent. We stored the parameters when the total loss of all considered training data sets decreased.

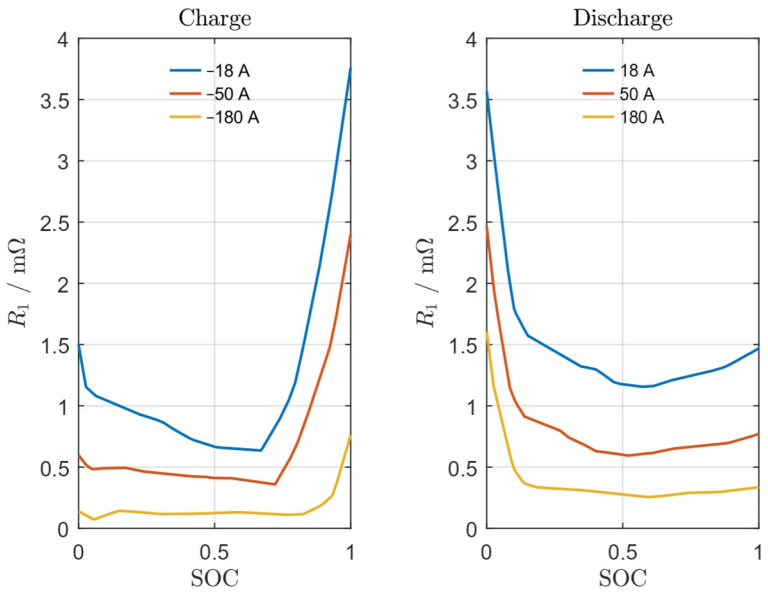


Fig. 3: Simulation results: R_1 as a function of SOC for different battery currents; (**left**): charging, (**right**): discharging. [3]. CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

In the second step, we completed the training with the whole GB model according to Equations (5a) to (5d). Therefore, we initialized ω_1 as stated previously. The other parameters were taken from the pre-trained model and the initial SOC was determined as before. The initial value of the voltage drop v_{RC1} across the RC circuit was needed as well. We assumed $v_{\mathrm{RC1}}\,(t=0) = 0\,\mathrm{V}$. The loss function was implemented as before. However, a constant learning rate of 10^{-3} was chosen. During the first ten training epochs of the second part, we only considered the data from charging and discharging processes with a pulsed battery current. Afterward, we also took the other training data sets into account. Additionally, we only considered ω_1 as a changeable parameter during the first 20 of 30 training epochs in this second part of training. The other parameters were frozen. Here, we chose batch gradient descent for parameter optimization.

As mentioned before, the number of hidden neurons in f and g was varied between 10 and 300. The number of training epochs in the first training step served as a second hyperparameter. It was varied between 100 and 1000 while training part two was not changed. The detailed evaluation of the results in Ref. [3] finally made us choose the trained model with 100 hidden neurons in f and g and 300 training epochs in training step one as the final GB model.

2.8 Test

We tested the final GB model against the data sets covering the half cycles and the synthetic load profile. For the half cycles, the integration of the differential equation system resulted in a step size underflow. Therefore, we had to increase the absolute tolerance of the solver to 10^{-3} for the half cycles. Otherwise, we proceeded the same way as for training.

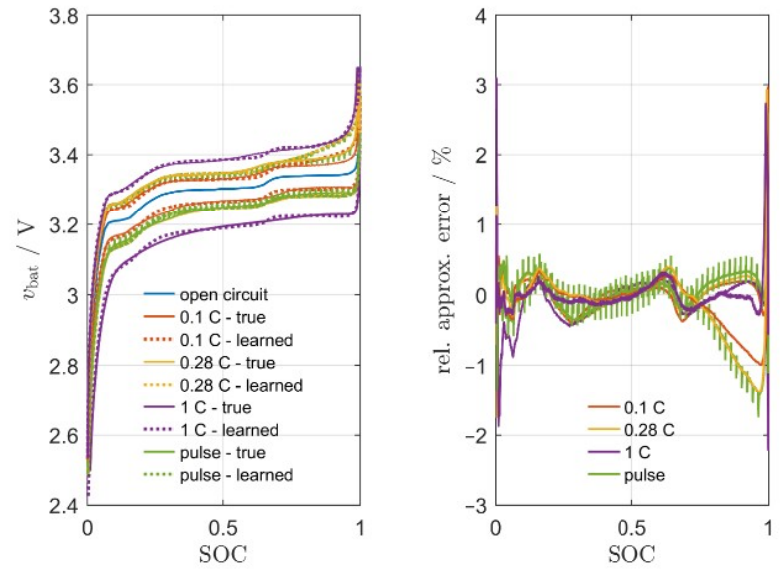
3 Results and Discussion

The training and test results can be found in detail in Ref. [3]. Here, we discuss the most important findings.

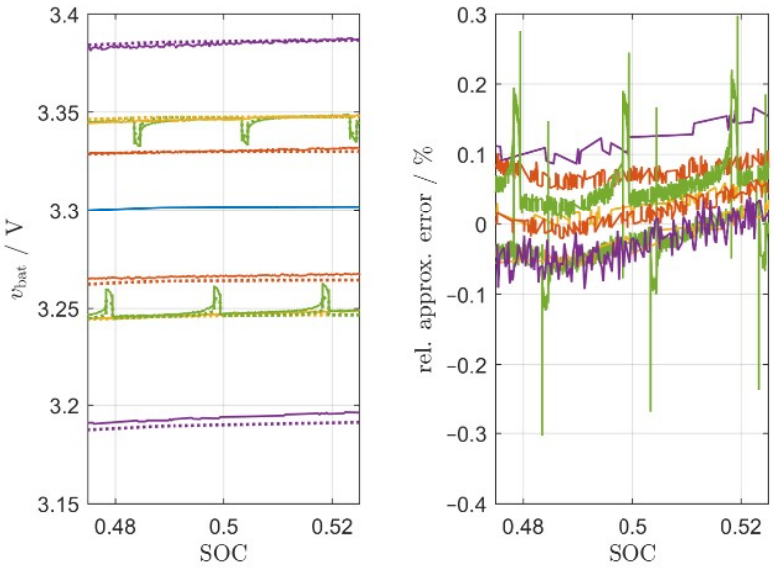
3.1 Training

After finishing the training, we took a closer look at the learned parameters and neural networks. The evaluation of the learned parameters with their respective scaling factors resulted in the following values:

$$\begin{aligned} C_{\mathrm{bat}} &= 191.5\,\mathrm{Ah} \\ C_1 &= 50.69\,\mathrm{kF} \\ v_{\mathrm{hys}} &= 11.25\,\mathrm{mV} \\ R_{\mathrm{S}} &= 281.4\,\mu\Omega. \end{aligned}$$



(a)



(b)

Fig. 4: Simulation results using NODEs for grey-box modelling of a lithium-ion battery in comparison to experimental data; left: charge and discharge curves for different C-rates at $T = 25\,^{\circ}\mathrm{C}$. The lower branches represent discharge (time progresses from right to left), while the upper branches represent charge (time progresses from left to right); right: relative approximation error; (**a**) the whole SOC range (**b**) focus on medium SOC. [3]. CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

The charge-transfer resistance is represented by the two neural networks f and g. Figure 3 illustrates the final results for R_1. The left panel shows the charge-transfer resistance for charging while the right panel shows the results for discharging as a function of SOC for different current values. The charge-transfer resistance is in the range of up to several milliohms.

When the absolute battery current is increased, the charge-transfer resistance decreases for both charging and discharging. At a medium SOC, the resistance is lower than at a low or a high SOC. During charge, the highest values occur when the cell is (nearly) full, during discharge, the highest values occur when the battery is (nearly) empty. This is typical for lithium-ion batteries with LFP cathode [13]. However, one has to keep in mind that the model is based on a simple equivalent circuit. It is therefore difficult to derive detailed electrochemical properties of the battery from the results.

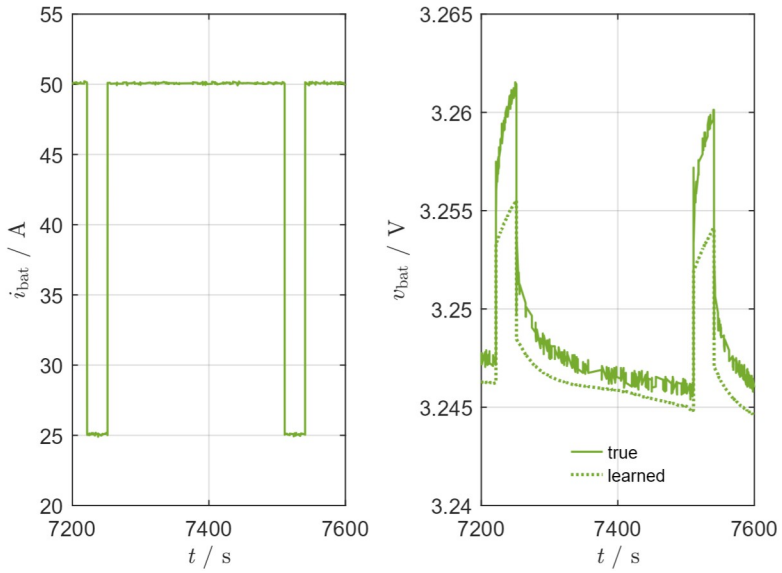


Fig. 5: Simulation results using NODEs for grey-box modelling of a lithium-ion battery in comparison to experimental data at $T = 25\,^{\circ}\mathrm{C}$. The focus is on discharging with a pulsed current at a medium SOC; (**left**): battery current versus time; (**right**): battery voltage versus time

3.2 Comparison of Model against Training Data

Figure 4 shows the training results in comparison to the measured battery voltage. Here we chose a representation in the form of voltage versus SOC. The measurement data was given as voltage versus time series. However, the graphical display versus SOC allows a better comparison for different C-rates. The measured and the learned battery voltage are shown in the left panel. The right panel shows the relative approximation error concerning the measured battery voltage. Figure 4a shows the complete SOC range while Figure 4b focuses on a medium SOC. Overall, the simulation results are well in accord with the experiments for all investigated C-rates. The absolute value of the relative approximation error is smaller than 1% for a wide range of SOC. Only for (nearly) full and (nearly) empty batteries, the approximation error reaches an absolute value of up to around $3\,\%$. As the OCV(SOC) curve (shown in blue in Figure 4a,b) is very steep for high and low values of SOC, higher approximation errors are expected. Especially, for this reason, the results are acceptable.

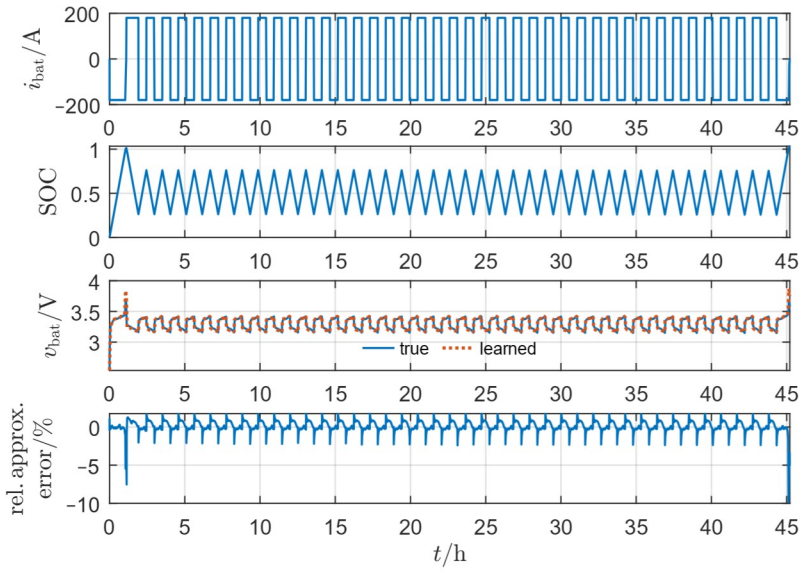
Figure 5 shows the training results for a pulsed current discharge in comparison to the measured battery voltage. In the original contribution [3] we have shown an excerpt of the charge branch for comparison. In contrast to Figure 4 we chose a temporal representation here. Figure 5 shows the voltage curve for a medium SOC. We achieved good results for the voltage response of the battery following a current step. However, the absolute voltage drop is underestimated by the model. The simulation shows an exponential behaviour resulting from the first-order dynamics of the RC element (Equation (5b)) which differs slightly from the experiment. Here, one has to keep in mind that we have chosen a rather simple ECM as a basis for GB modelling. The results are similar for other SOC values and the charge branch.

In conclusion, the training results are in good agreement with the experiment.

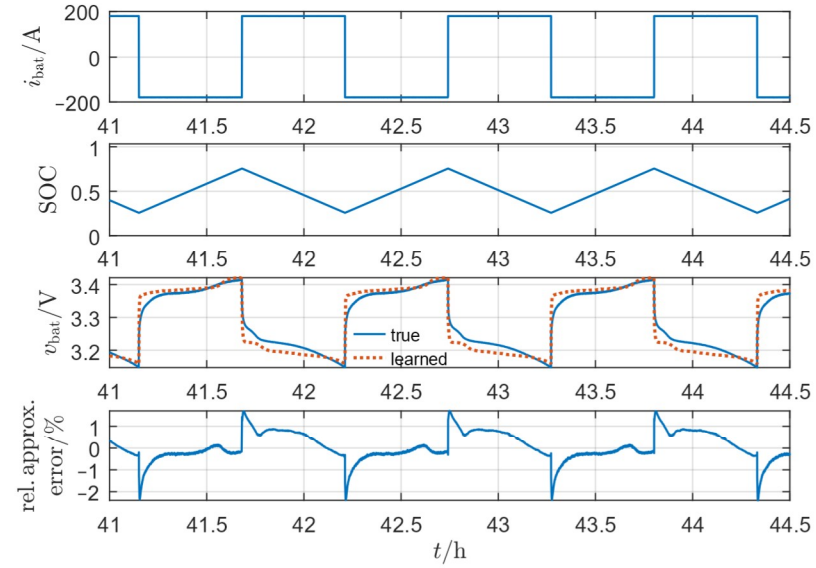
3.3 Comparison of Model against Test Data

We used data not included in the training data set to test the final GB model. The measurement data from several consecutive half cycles served as the first test data set. The approximation results are shown in Figure 6 in comparison to the experimental data. Figure 6a shows the complete time series. Overall, the results are in good agreement with the measured battery voltage. Figure 6b focuses on the last three half cycles of the time series. Particularly at the beginning of each half cycle, some deviations occur between simulation and experiment. However, overall the results are good.

The synthetic load profile of a home-storage battery was used as a second test data set. The results are shown in Figure 7. While Figure 7a covers the complete time series Figure 7b focuses on a section in the middle with fast dynamics. The test results are in good agreement with experimental data for the complete load profile. The highest relative approximation errors occur in the area of high SOC values. Note that the training loss was also high at high



(a)



(b)

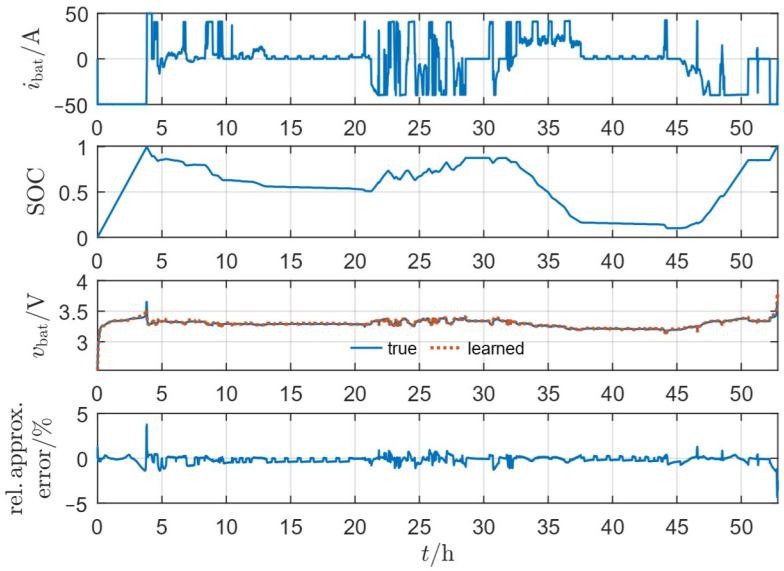
Fig. 6: Test results in comparison to experimental data at $T = 25\,^{\circ}\mathrm{C}$ for half cycles; (**a**) the complete time series; (**b**) focus on the last three half cycles. [3]. CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

values of SOC. The synthetic load profile covers the longest measuring time which is around 4.5 times as long as the longest training time series. Nevertheless, the test results are good for the whole load profile.

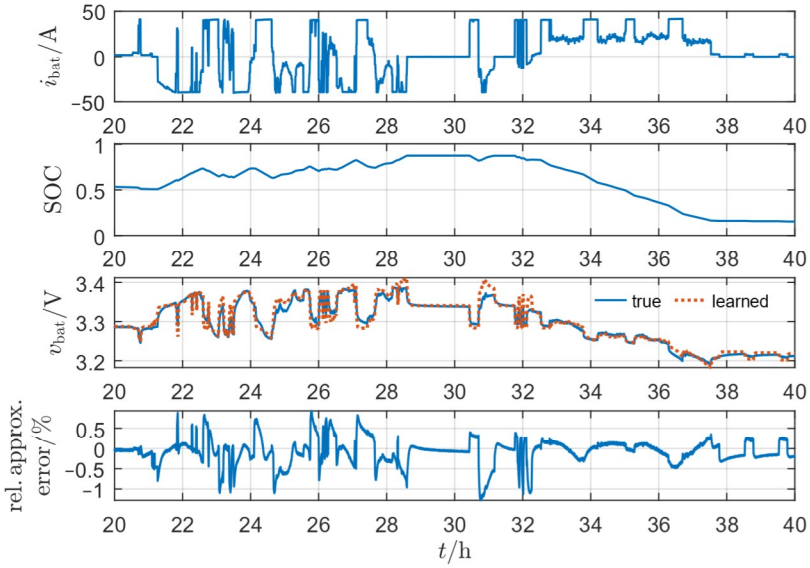
4 Summary and Conclusions

We have shown how to couple physics-based ODEs and NODEs for GB modelling of lithium-ion batteries. The derived model was trained and tested with experimental data of an LFP battery cell that is used in home-storage applications.

As there was only little training data available, we split the training into two steps: first, a simplified static model with neglected double-layer capacitance was trained with CCCV training data. In the second step, the fast dynamics of the battery were taken into account.



(a)



(b)

Fig. 7: Test results in comparison to experimental data at $T = 25\,^{\circ}\mathrm{C}$ for a synthetic load profile; (**a**) the complete time series (**b**) focus on the segment in the middle. [3]. CC BY 4.0 https://creativecommons.org/licenses/by/4.0/

The final GB model can reproduce the complete set of training data with good accuracy. For the test data, the simulations also show good agreement with the experiments. The highest approximation errors occur where the OCV curve is very steep.

It would be beneficial to have more training data. Especially, the usage of training data from pulse tests with different current steps would be of interest. A bigger data set would also allow a better model validation. A k-fold cross validation could be used to evaluate the robustness of the model against the chosen training data.

Summing up, we have shown how to combine NODEs and physics-based ODEs for GB modelling of lithium-ion batteries.

References

1. Franco, A.A., Doublet, M.L., Bessler, W.G., eds.: Physical Multiscale Modeling and Numerical Simulation of Electrochemical Devices for Energy Conversion and Storage: From Theory to Engineering to Practice. 1 edn. Green Energy and

Technology. Springer, London, London (2016)
2. Seaman, A., Dao, T.S., McPhee, J.: A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation. Journal of Power Sources **256** (2014) 410–423
3. Brucker, J., Behmann, R., Bessler, W.G., Gasper, R.: Neural ordinary differential equations for grey-box modelling of lithium-ion batteries on the basis of an equivalent circuit model. Energies **15**(7) (2022)
4. Estrada-Flores, S., Merts, I., de Ketelaere, B., Lammertyn, J.: Development and validation of "grey-box" models for refrigeration applications: A review of key concepts. International Journal of Refrigeration **29**(6) (2006) 931–946
5. Oussar, Y., Dreyfus, G.: How to be a gray box: dynamic semi-physical modeling. Neural Networks **14**(9) (2001) 1161–1172
6. Duarte, B., Saraiva, P.M., Pantelides, C.C.: Combined mechanistic and empirical modelling. International Journal of Chemical Reactor Engineering **2**(1) (2004)
7. Hamilton, F., Lloyd, A.L., Flores, K.B.: Hybrid modeling and prediction of dynamical systems. PLoS computational biology **13**(7) (2017) e1005655
8. Almeida, G.C.S., de Souza, A.C.Z., Ribeiro, P.F.: A neural network application for a lithium-ion battery pack state-of-charge estimator with enhanced accuracy. Proceedings **58**(1) (2020)
9. Jiménez-Bermejo, D., Fraile-Ardanuy, J., Castaño-Solis, S., Merino, J., Álvaro-Hermana, R.: Using dynamic neural networks for battery state of charge estimation in electric vehicles. Procedia Computer Science **130** (2018) 533–540
10. Yang, D., Wang, Y., Pan, R., Chen, R., Chen, Z.: A neural network based state-of-health estimation of lithium-ion battery in electric vehicles. Energy Procedia **105** (2017) 2059–2064
11. Bills, A., Sripad, S., Fredericks, W.L., Guttenberg, M., Charles, D., Frank, E., Viswanathan, V.: Universal battery performance and degradation model for electric aircraft. ChemRxiv (2020)
12. Brucker, J., Bessler, W.G., Gasper, R.: Grey-box modelling of lithium-ion batteries using neural ordinary differential equations. Energy Informatics **4**(S3) (2021) 1–13
13. Yagci, M.C., Behmann, R., Daubert, V., Braun, J.A., Velten, D., Bessler, W.G.: Electrical and structural characterization of large–format lithium iron phosphate cells used in home–storage systems. Energy Technology **9**(6) (2021)
14. Dreyer, W., Jamnik, J., Guhlke, C., Huth, R., Moskon, J., Gaberscek, M.: The thermodynamic origin of hysteresis in insertion batteries. Nature materials **9**(5) (2010) 448–453
15. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. Adaptive computation and machine learning. MIT Press, Cambridge, Massachusetts (2016)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 29th IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ, IEEE (2016) 770–778
17. Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.: Neural ordinary differential equations. CoRR **abs/1806.07366** (2018)
18. Haber, E., Ruthotto, L.: Stable architectures for deep neural networks. Inverse Problems **34**(1) (2017)
19. Ruthotto, L., Haber, E.: Deep neural networks motivated by partial differential equations. Journal of Mathematical Imaging and Vision **62** (2020) 352–364
20. Dupont, E., Doucet, A., Teh, Y.W.: Augmented neural odes. In: Advances in Neural Information Processing Systems 32, Red Hook, NY, USA, Curran Associates, Inc. (2019) 3140–3150
21. Zhang, T., Yao, Z., Gholami, A., Keutzer, K., Gonzalez, J., Biros, G., Mahoney, M.W.: Anodev2: A coupled neural ode evolution framework. CoRR **abs/1906.04596** (2019)
22. Haber, E., Ruthotto, L., Holtham, E., Jun, S.H.: Learning across scales - multiscale methods for convolution neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 32., Palo Alto, California USA, AAAI Press (2018)
23. Gholami, A., Keutzer, K., Biros, G., Gholaminejad, A.: Anode: Unconditionally accurate memory-efficient gradients for neural odes. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. (2019) 730–736
24. Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., Edelman, A.: Universal differential equations for scientific machine learning. CoRR **abs/2001.04385** (2020)
25. He, H., Xiong, R., Fan, J.: Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. Energies **4**(4) (2011) 582–598
26. Wang, Y., Fang, H., Zhou, L., Wada, T.: Revisiting the state-of-charge estimation for lithium-ion batteries: A methodical investigation of the extended kalman filter approach. IEEE Control Systems **37**(4) (2017) 73–96
27. Braun, J.A., Behmann, R., Schmider, D., Bessler, W.G.: State of charge and state of health diagnosis of batteries with voltage-controlled models. Journal of Power Sources **544** (2022)
28. Weißhar, B., Bessler, W.G.: Model-based lifetime prediction of an lfp/graphite lithium-ion battery in a stationary photovoltaic battery system. Journal of Energy Storage **14** (2017) 179–191
29. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In Orr, G.B., Müller, K.R., eds.: Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
30. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, Red Hook, NY, USA, Curran Associates, Inc. (2019) 8024–8035
31. Chen, R.T.Q.: torchdiffeq (version 0.2.1) (2021) `https://github.com/rtqichen/torchdiffeq`