Brucker *et al. Energy Informatics* 2021, **4**(Suppl 3):15
https://doi.org/10.1186/s42162-021-00170-8

Energy Informatics

RESEARCH Open Access

Grey-box modelling of lithium-ion batteries using neural ordinary differential equations

Jennifer Brucker*, Wolfgang G. Bessler and Rainer Gasper

From The 10th DACH+ Conference on Energy Informatics
Virtual. 13–17 September 2021

*Correspondence:
jennifer.brucker@hs-offenburg.de
Institute of Energy Systems Technology, Offenburg University of Applied Sciences, Badstraße 24, Offenburg, Germany

Abstract

Grey-box modelling combines physical and data-driven models to benefit from their respective advantages. Neural ordinary differential equations (NODEs) offer new possibilities for grey-box modelling, as differential equations given by physical laws and neural networks can be combined in a single modelling framework. This simplifies the simulation and optimization and allows to consider irregularly-sampled data during training and evaluation of the model. We demonstrate this approach using two levels of model complexity; first, a simple parallel resistor-capacitor circuit; and second, an equivalent circuit model of a lithium-ion battery cell, where the change of the voltage drop over the resistor-capacitor circuit including its dependence on current and State-of-Charge is implemented as NODE. After training, both models show good agreement with analytical solutions respectively with experimental data.

Keywords: Neural ordinary differential equations, Dynamic systems, External variables, Grey-box model, Equivalent circuit model, Lithium-ion batteries

Introduction

Lithium-ion batteries have become an integral part of our everyday lives: They supply smartphones and laptops with electrical energy, they are used as mobile power source in electric vehicles, and they help to secure the energy supply as stationary storage plants. To operate lithium-ion batteries safely and efficiently, we need a comprehensive understanding of how the battery works and which internal processes take place. Battery modelling is complex and parameterization is a demanding task. We show a novel way of equivalent circuit modelling of lithium-ion batteries using neural ordinary differential equations (NODEs).

With increasing digitization and the associated larger amount of available data, artificial intelligence and especially neural networks gain importance. Neural networks belong to the class of black-box (BB) models. They use measured data to learn relations between inputs and outputs of systems (Döbel et al. 2018; Estrada-Flores et al. 2006; Oussar and Dreyfus 2001; Duarte et al. 2004; Hamilton et al. 2017). As Ljung (1999) has already stated in 1999, known aspects should never be estimated. Therefore, it is reasonable to consider other modelling techniques. Physical modelling based on the usage of prior knowledge is a contrary approach. The resulting white-box (WB) models describe the underlying system dynamics in form of mathematical equations. Grey-box (GB) models combine WB and BB modelling techniques and thus the respective advantages (Döbel et al. 2018; Estrada-Flores et al. 2006; Oussar and Dreyfus 2001; Duarte et al. 2004; Hamilton et al. 2017).

Missing values and irregularly-sampled time-domain data are still demanding when using neural networks. NODEs are a promising approach to deal with these problems (Chen et al. 2018). In the context of modelling dynamic systems, NODEs are used to solve homogeneous differential equations (Chen et al. 2018). However, external variables that often are important to describe the behaviour of a dynamic system have not yet been taken into account.

The focus of the present study is on GB modelling of lithium-ion batteries using NODEs. In the 'Neural ordinary differential equations' section we introduce NODEs and show how they can solve inhomogeneous differential equations. 'Grey-box modelling' section deals with GB modelling in general and GB modelling using NODEs in particular. 'Modelling of lithium-ion batteries' section applies the main results to equivalent circuit modelling of lithium-ion batteries. In the 'Discussion and conclusion' section we discuss the results and draw conclusions.

Neural ordinary differential equations

In this section, we give a general intro to NODEs. We extend NODEs for BB modelling of dynamic systems including external variables. A resistor-capacitor (RC) circuit serves as application example.

Background neural ordinary differential equations

Modelling dynamic systems using neural networks has been addressed in many publications. For instance, Che et al. (2018) used recurrent neural networks (RNNs) to model multivariate time series with missing values and Bailer-Jones et al. (1998) modelled dynamic systems including external variables through RNNs. Liao and Poggio (2016) interpreted a residual neural network (ResNet) with shared weights as RNN.

A ResNet transforms the hidden states from layer t to $t + 1$ according to the recursive equation

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

where $\boldsymbol{z}_t \in \mathbb{R}$ is the vector of the hidden states at layer t, $\boldsymbol{\theta}_t$ represents the learned parameters of layer t and $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a function preserving the dimension of the hidden states (He et al. 2016). Sharing the parameters across the layers ($\boldsymbol{\theta}_t = \boldsymbol{\theta}$ for $t = 0, ..., T$) leads to the explicit Euler discretization of the initial value problem (Chen et al. 2018; Haber and Ruthotto 2017; Ruthotto and Haber 2020; Dupont et al. 2019; Zhang et al. 2019; Haber et al. 2018; Gholami et al. 2019)

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

NODEs specify the continuous change of the hidden states according to Eq. 2 using a neural network. Starting from the input layer $\boldsymbol{z}(0)$, a differential equation solver calculates the output layer $\boldsymbol{z}(T)$ (Chen et al. 2018; Dupont et al. 2019; Zhang et al. 2019; Gholami et al. 2019).

The implementation of NODEs is challenging due to storage requirements for backpropagation during training. To deal with this storage problem Chen et al. (2018) proposed an adjoint sensitivity method for backpropagation. As discussed by Gholami et al. (2019), this method may lead to numerical instability and inaccurate gradients. Therefore, Gholami et al. (2019) introduced an Adjoint based Neural ordinary differential equation (ODE) framework using a checkpointing method. Chen et al. (2018) used NODEs for supervised learning tasks. They applied NODEs to normalizing flows even considering time-dependent dynamics and they modelled time series including irregularly-sampled data with NODEs. One can find many extensions to this approach in current research. For example, Dupont et al. (2019) augmented NODEs to model complex functions with simple flows. Zhang et al. (2019) developed a coupled ODE based framework allowing the network parameters to evolve with time. Gusak et al. (2020) proposed normalization techniques to improve the quality of the results achieved with NODEs. Zhang et al. (2020) discussed approximation capabilities of NODEs.

Chen et al. (2018) and Gholami et al. (2019) released PyTorch (Paszke et al. 2017) implementations of their ODE solvers.

Black-box modelling of dynamic systems including external variables with NODEs

Usually ODEs are used to describe dynamic systems. When external variables influence the behaviour of the system the corresponding differential equations become inhomogeneous. Typical dynamic systems follow the equations

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} &= \boldsymbol{f}\left(\boldsymbol{z}(t), \boldsymbol{u}(t)\right) \\ \boldsymbol{y}(t) &= \boldsymbol{g}\left(\boldsymbol{z}(t), \boldsymbol{u}(t)\right) \end{aligned} \tag{3}$$

where $\boldsymbol{z}$ are the state variables, $\boldsymbol{u}$ are the external variables, $\boldsymbol{y}$ are the outputs of the system, and $\boldsymbol{f}$ and $\boldsymbol{g}$ are (non-linear) continuous functions. The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ could also depend on the time t explicitly. Then the dynamic system would be time-variant. Bailer-Jones et al. (1998) proposed a special form of RNN to model time-invariant systems. This RNN encourages the usage of NODEs for modelling dynamic systems including external variables. We generalize the initial value problem according to Eq. 2:

$$\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = f\left(\boldsymbol{z}(t), \boldsymbol{u}(t), t, \boldsymbol{\theta}\right); \ \boldsymbol{z}(0) = \boldsymbol{z}_0. \tag{4}$$

As the neural network f in Eq. 4 depends on t explicitly, the considered systems may be time-variant. We can use one of the frameworks proposed by Chen et al. (2018) or Gholami et al. (2019) to implement the NODE. However, we have to consider the external variables as inputs to the neural network. Therefore, we provide a function describing the course of the external variables with time. The interpolation of measured data is a possible method. It is also conceivable to consider time dependencies (cf. time-dependent dynamics in Chen et al. (2018)). If it is known that the external variables follow specific functions of time, for example, sine functions, we provide the corresponding function

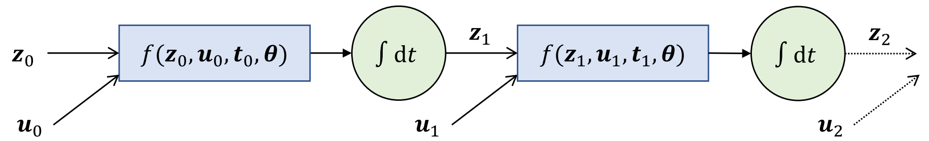

Fig. 1 Including external variables in NODEs; z_t represents the state variables at layer t, and u_t represents the respective external variables

parameters as inputs to the neural network. We then calculate the values at the considered time points during the forward pass. Figure 1 illustrates the suggested approach to include external variables when using NODEs schematically. A differential equation solver, for example, Euler, solves the NODE.

Minimization of the defined loss function optimizes the learnable parameters of the neural network. In the case that we cannot measure one or more of the state variables directly, a correct approximation of the course of these state variables with time cannot be guaranteed. If the corresponding trajectories are important, we have to provide additional information during training.

Application of black-box modelling with NODEs to RC circuit modelling

A parallel RC circuit fed by a current source that is part of a standard equivalent circuit model (ECM) serves as application example. The RC circuit is shown in Fig. 2. The output voltage v_a is the voltage drop across the parallel connection of the resistor and the capacitor.

It follows the differential equation

$$\frac{dv_a(t)}{dt} = -\frac{1}{RC} v_a(t) + \frac{1}{C} i(t) \tag{5}$$

where $R = 100\ \Omega$ denotes the ohmic resistance and $C = 10\,\text{mF}$ denotes the capacitance. The voltage v_a is the state variable and the output of the system. The current i is the external variable. The initial output voltage is set to $v_a(0) = 1\,\text{V}$. We defined a NODE to approximate the derivative of the output voltage with time:

$$\frac{dv_a(t)}{dt} = f\left(v_a(t), i(t), \theta\right), \tag{6}$$

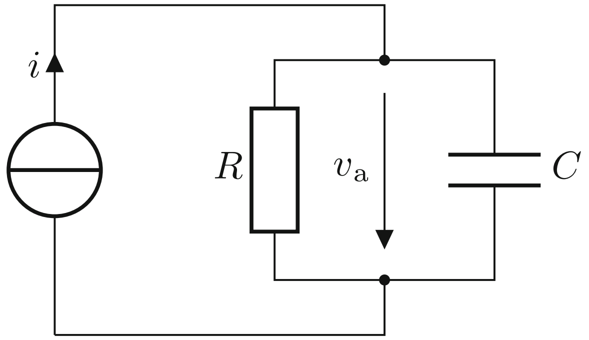

Fig. 2 RC circuit

where f is a linear feedforward network with two hidden layers with ten neurons each. We did not include biases. We initialized the weights from the uniform distribution $\mathcal{U}\left(-\sqrt{k}, \sqrt{k}\right)$, where $k = \frac{1}{l}$ with $l \in \mathbb{N}$ the number of inputs to the respective layer.

We used two sinusoidal current signals with different amplitudes, frequencies and phase shifts for training. The validation signal was also a sinusoidal signal. Its amplitude is time-dependent. In order to check the generalization ability of the investigated network, the product of a sine and a cosine function with different frequencies and phase shifts served as test signal. The different current signals were chosen as follows:

$$i_{\text{train1}}(t) = 1\,\text{A} \cdot \sin\left(2\pi \cdot 50\,\text{Hz} \cdot t + 5\right) \tag{7a}$$

$$i_{\text{train2}}(t) = 2\,\text{A} \cdot \sin\left(2\pi \cdot 30\,\text{Hz} \cdot t + 1\right) \tag{7b}$$

$$i_{\text{val}}(t) = 5\,\text{A} \cdot \sin\left(2\pi \cdot 25\,\text{Hz} \cdot t + 0\right) \cdot 15t \tag{7c}$$

$$i_{\text{test}}(t) = 3\,\text{A} \cdot \sin\left(2\pi \cdot 80\,\text{Hz} \cdot t + 2\right) \cdot \cos\left(2\pi \cdot 50\,\text{Hz} \cdot t\right). \tag{7d}$$

We implemented our example in Python (version 3.7.6). The regarded time interval of 0.05 s duration was spaced into 200 time steps of random size between 9.9998×10^{-10} s and 0.0013 s. The true output voltages were calculated through integration of Eq. 5 using the Python library SciPy (version 1.4.1) (Virtanen et al. 2020). No noise was included.

The standard odeint solver from torchdiffeq (version 0.1.1) (Chen et al. 2018) integrated the NODE using Euler's method. In each of the 3000 optimization steps, we chose one training sample randomly for gradient descent. In other words we performed 3000 optimization steps with stochastic gradient descent. Therefore, we provided the initial value of the voltage v_{a} and the parameters describing the course of the currents. The current values were calculated during the forward pass.

An Adam (Kingma and Ba 2014) optimizer with learning rate $l = 0.01$ minimized the MSE loss of the learned and the calculated output voltages. The chain rule was used for backpropagation. We do not address memory limits in this paper. Therefore, we did not use any of the adjoint methods proposed by Chen et al. (2018) or Gholami et al. (2019) for backpropagation. However, the principal approach would be the same. The validation set was used to avoid overfitting. We used the model parameters which led to the minimal validation loss for the final test. The training procedure was repeated three times.

The resulting loss values are summarized in Table 1. The differences between optimization run 1 and 3 are small. Run 2 led to worse results regarding the training losses. However, the validation and test losses are better than in run 1 and 3. In total, the test losses are around one order of magnitude higher than validation and training losses. Figure 3 shows the results after training for run 2 in comparison to the results for GB modelling (next section). On the left, the courses of the true and learned output voltages

Table 1 Comparison of the MSE losses using NODEs for black-box and grey-box modelling of the RC circuit for the given train, validation and test sets

MSE loss	**black-box modelling**			**grey-box modelling**		
	run 1	**run 2**	**run 3**	**run 1**	**run 2**	**run 3**
Training signal 1	0.000299	0.000366	0.000310	0.001073	0.000896	0.000891
Training signal 2	0.000554	0.000918	0.000712	0.001364	0.001678	0.001687
Validation signal	0.000318	0.000247	0.000362	0.001529	0.001512	0.001512
Test signal	0.006380	0.005587	0.006208	0.002169	0.002325	0.002329

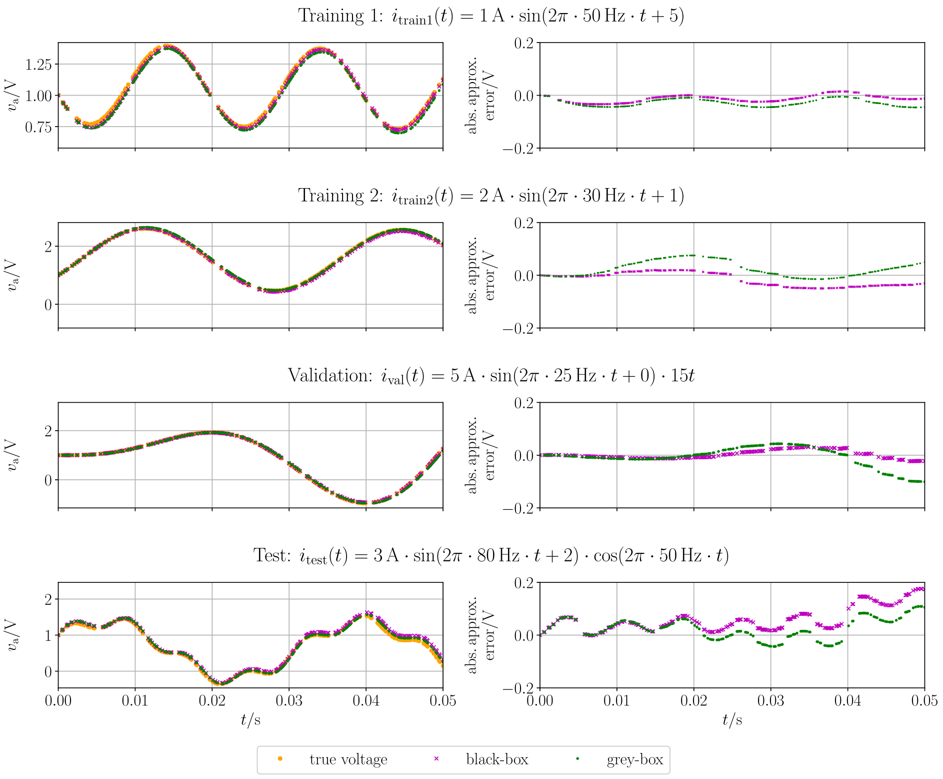

Fig. 3 Comparison of the simulation results using NODEs for black-box and grey-box modelling of the RC circuit; left: true and learned output voltages, right: absolute approximation error

are shown. On the right, the absolute approximation error displays the difference between learned and true output voltage. As receiving the optimum results is not in the focus, we neither performed hyperparameter tuning nor regularization.

Grey-box modelling

This section deals with GB modelling. We introduce a framework for GB modelling using NODEs and apply it to RC circuit modelling.

Background grey-box modelling

GB modelling combines the advantages of both WB and BB modelling. Prior knowledge is included in the modelling process. Therefore, reliable parameter estimation requires less data in comparison to BB modelling (Döbel et al. 2018; Estrada-Flores et al. 2006; Oussar and Dreyfus 2001; Duarte et al. 2004; Hamilton et al. 2017). Sohlberg (2003) differentiated two GB modelling procedures. One is to constrain the model parameters or variables of a BB model using prior knowledge. The resulting dark GB models use specific neuro-fuzzy network structures (Lindskog and Ljung 2000). The second procedure takes a WB model as basis for GB modelling. Sohlberg (2003) developed a GB model of a heating process based on a WB model. Hamilton et al. (2017) used Takens' method to build a GB model. Oussar and Dreyfus (2001) discretized a WB model and estimated unknown parameters.

Grey-box modelling with NODEs

Similar to the approaches of Oussar and Dreyfus (2001), Hamilton et al. (2017), and Sohlberg (2003), in which a WB model consisting of a system of differential equations forms the basis for GB modelling, we develop a GB modelling technique using NODEs. The GB model forms the forward pass of a neural network module. Therefore, we describe the derivatives of the state variables resulting from WB modelling inside the module. Single dependencies or entire equations in this WB model are then replaced with parametric parts. Additional assumptions going beyond the WB model can be added. The resulting combination of WB and BB parts forms the GB model. As described in section 'Black-box modelling of dynamic systems including external variables with NODEs', we can include external variables. A differential equation solver evaluates the evolution of the state variables. Just as for BB modelling, we choose a loss function and an optimizer depending on the modelling task.

Application of grey-box modelling with NODEs to RC circuit modelling

The RC circuit from 'Application of black-box modelling with NODEs to RC circuit modelling' section serves as application example for GB modelling as well. The change of the output voltage with time depends on the current and the voltage itself. For this example, we have assumed that the proportionality factor $1/C$ of the current is known, but that we are unsure about the proportionality factor $-1/(RC)$ of the voltage in Eq. 5. We used this prior knowledge of the dynamic system to derive a GB model using NODEs. The following relationship applies to the calculation of the derivative of the output voltage:

$$\frac{\mathrm{d}v_a(t)}{\mathrm{d}t} = \omega v_a(t) + \frac{1}{C} i(t), \tag{8}$$

where ω is the only learnable parameter. Equation 8 was implemented inside the forward pass of a neural network module. The general setting was the same as before during BB modelling. However, only 1000 optimization steps were carried out because the loss converged more quickly. To achieve lower loss values, we would have had to tune the hyperparameters. Again, we repeated the training three times.

The resulting MSE losses are shown in Table 1. The differences between the three runs are insignificant. The results of the first run outperform the others marginally regarding the test losses. Figure 3 shows the results in comparison to the results for BB modelling. The results will be discussed in 'Discussion and conclusion' section.

Modelling of lithium-ion batteries

In this section, we describe how to model a lithium-ion battery in form of an ECM. We give an overview of the usage of neural networks in the field of battery modelling. Finally, we apply GB modelling using NODEs to an equivalent circuit of a battery.

Background modelling lithium-ion batteries

Equivalent circuit modelling is a common approach for battery modelling. ECMs consist of electrical elements that describe the dynamic behaviour of batteries in a simple way and with a few parameters and states. Therefore, they often are used for SOC and State-of-Health prediction (He et al. 2011; Wang et al. 2017). Standard ECMs consist of an SOC-dependent voltage source, a series resistor and one or more RC circuits (He et al. 2011; Fleischer et al. 2014; Chen and Rincon-Mora 2006; Haifeng et al. 2009; Hu et al.

2009; Tong et al. 2015; Krewer et al. 2018). We can extend the standard ECM by taking into account that the circuit parameters depend on SOC, temperature, the applied current or the cycle number (Chen and Rincon-Mora 2006; Krewer et al. 2018).

Neural networks are used to model lithium-ion batteries more often. For example, Zhang et al. (2019), Jiménez-Bermejo et al. (2018), and Charkhgard and Farrokhi (2010), and Almeida et al. (2020) estimated the SOC of batteries with neural networks. Yang et al. (2017) used a neural network to estimate the State-of-Health of a battery with the parameters of an ECM as inputs. Krewer et al. (2018) summarized BB and GB modelling approaches to estimate the SOC and State-of-Health of lithium-ion batteries. In contrast to that, Wu et al. (2018) simplified battery design through neural networks. Turetskyy et al. (2019) combined a physical battery model and a feedforward network for end-of-line battery cell characterization.

Equivalent circuit model of a lithium-ion battery

We used a simple ECM consisting of an SOC-dependent voltage source, a series resistor and one RC circuit for modelling the dynamic behaviour of a lithium-ion battery. The equation system describing the chosen ECM can be found in He et al. (2011), and Tong et al. (2015). We directly included parameter dependencies on current and SOC:

$$\frac{\mathrm{dSOC}}{\mathrm{d}t} = -\frac{1}{C_\mathrm{N}} i_\mathrm{bat} \tag{9a}$$

$$\frac{\mathrm{d}v_\mathrm{RC1}}{\mathrm{d}t} = \frac{1}{C_1}\left(-\frac{1}{R_1\left(\mathrm{SOC}, i_\mathrm{bat}\right)} v_\mathrm{RC1} + i_\mathrm{bat}\right) \tag{9b}$$

$$v_\mathrm{bat} = v_\mathrm{OC}\left(\mathrm{SOC}\right) - R_\mathrm{S}\left(\mathrm{SOC}, i_\mathrm{bat}\right) i_\mathrm{bat} - v_\mathrm{RC1}, \tag{9c}$$

where C_N is the nominal battery capacity, $R_\mathrm{S}\left(\mathrm{SOC}, i_\mathrm{bat}\right)$ and $R_1\left(\mathrm{SOC}, i_\mathrm{bat}\right)$ are the ohmic resistances depending on SOC and battery current, C_1 is the capacitance, and $v_\mathrm{OC}\left(\mathrm{SOC}\right)$ is the SOC-dependent open-circuit voltage (OCV). The battery voltage v_bat is the output of the dynamic system, and the current i_bat is the external variable. Figure 4 shows the corresponding ECM.

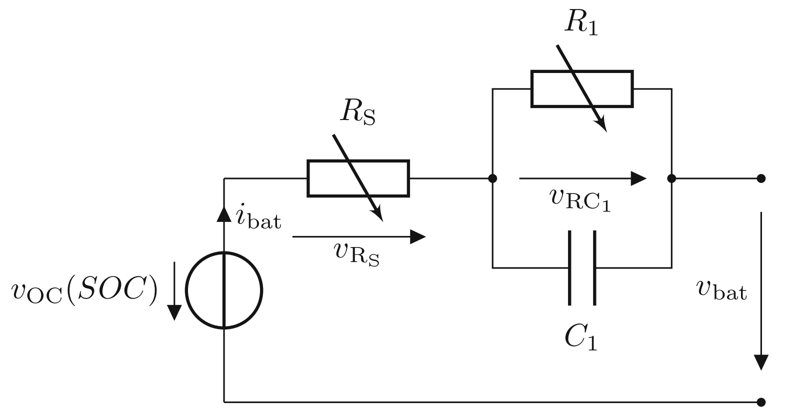

Fig. 4 ECM of a battery consisting of an SOC-dependent voltage source, a series resistor and an RC circuit

Grey-box modelling of a lithium-ion battery

As demonstration of the methodology, we described the charging and discharging characteristics of a lithium-ion battery used for stationary energy storage with a GB model using NODEs. The considered battery has been characterized experimentally in detail before by Yagci et al. (2021). It is a prismatic single cell of the Chinese manufacturer CALB with a rated (data-sheet) capacity of 180 Ah and a real (measured) capacity of 202 Ah. The chemistry is lithium iron phosphate (LFP) at the positive electrode and graphite at the negative electrode. This type of cell is typically applied in home storage systems. The electrical and thermal behaviour of the cell was investigated under controlled environment (CTS climate chamber) using a battery cycler (Biologic). For the present investigations, only a part of the available experimental data was used, in particular charge and discharge curves at $T = 20$ °C obtained with a constant current, constant voltage (CCCV) discharging/charging protocol at different C-rates (CC phase) of 0.02 C, 0.1 C, 0.28 C, and 1 C, and a cut-off current (CV phase) of C/20. The data was available and used as voltage-versus-time and current-versus-time series, for which we reduced the number of data points to 100 for each discharge and charge process. For the model, the experimental current was used as input. The presentation of the results below is in the form of voltage versus SOC, which allows a better comparison of the different C-rates. Details of the experimental approach can be found in Yagci et al. (2021).

We used the ECM according to Eq. 9 as basis for GB modelling. The state Eqs. 9a and 9b were implemented inside a neural network module. Linear interpolation of the measured current values led to a function describing the current's temporal progress. We provided this function as external input to the GB module. The derivatives of the states were the outputs of the module. As the nominal capacity C_N of the battery is known, we were able to implement Eq. 9a directly. According to the results in Yagci et al. (2021) we chose $C_\mathrm{N} = 202$ Ah. The ohmic resistances R_S and R_1 and the capacitance C_1 are not known. In addition, we wanted to include parameter dependencies on current and SOC. Therefore, we used neural networks to approximate Eq. 9b. In detail, one feedforward network approximated the change of the voltage drop across the RC circuit dependent on SOC and the voltage v_RC1 itself and a second neural network included the current dependency. Overall, we obtained the following equation system:

$$\frac{\mathrm{dSOC}}{\mathrm{d}t} = -\frac{1}{C_\mathrm{N}} i_\mathrm{bat} \tag{10a}$$

$$\frac{\mathrm{d}v_\mathrm{RC1}}{\mathrm{d}t} = f\left(g\left(\mathrm{SOC}, v_\mathrm{RC1}, \boldsymbol{\theta}_g\right), i_\mathrm{bat}, \boldsymbol{\theta}_f\right) \tag{10b}$$

where f and g are representing feedforward networks. We chose a linear network with six hidden layers with ten neurons each for f and a network with eight hidden layers with 50 neurons each, sigmoid activation, and one output neuron for g. We did not include biases and we initialized the weights as in 'Application of black-box modelling with NODEs to RC circuit modelling'. Again, we used the standard odeint solver from torchdiffeq (Chen et al. 2018) to integrate the equation system using Euler's method. The solutions of 10 were used to calculate the battery output voltage (cf. 9c). We implemented the non-linear v_OC (SOC) curve according to Yagci et al. (2021), and Mayur et al. (2019). We then used the calculated SOC to obtain the OCV v_OC. To approximate the voltage drop over the serial resistor R_S, we used an additional feedforward network. With this, the battery output voltage was calculated according to following equation:

$$v_{\mathrm{bat}} = v_{\mathrm{OC}}\left(\mathrm{SOC}\right) - h\left(\mathrm{SOC}, i_{\mathrm{bat}}, \boldsymbol{\theta}_h\right) - v_{\mathrm{RC1}} \tag{11}$$

with the linear feedforward network h consisting of five hidden layers with 50 neurons each. An Adam (Kingma and Ba 2014) optimizer minimized the L1 loss of the approximated and the measured battery voltage. Here we chose a decaying learning rate between 1×10^{-1} and 1×10^{-5}. The 0.02 C charge curve, the 0.1 C discharge curve, and the 1 C discharge and charge curves were used for training. In total, 2000 optimization steps were carried out with stochastic gradient descent. The 0.02 C discharge curve, the 0.1 C charge curve, and the 0.28 C discharge and charge curves were used for testing. To plot the measured voltages against the SOC, we calculated the SOC according to $\mathrm{SOC} = {}^{Q}\!/_{C_{\mathrm{N}}}$ with Q the measured charge throughput.

The results are shown in Fig. 5. The left panel shows a comparison of experimental and simulated CCCV charge and discharge curves for different C-rates at $T = 20$ °C. The absolute approximation error is shown in the right panel.

Discussion and conclusion

In the 'Application of black-box modelling with NODEs to RC circuit modelling' section we used a BB model with NODEs to approximate the output voltage of an RC circuit. We used different current signals as external variables. The results show that the BB model was able to fit the data and to include the external variable.

We approximated the output voltage of the same RC circuit using a GB model with NODEs in the 'Application of grey-box modelling with NODEs to RC circuit modelling' section. Using the provided current signal, the model was able to fit the course of the output voltage with time. Although the training and validation losses are higher than for BB

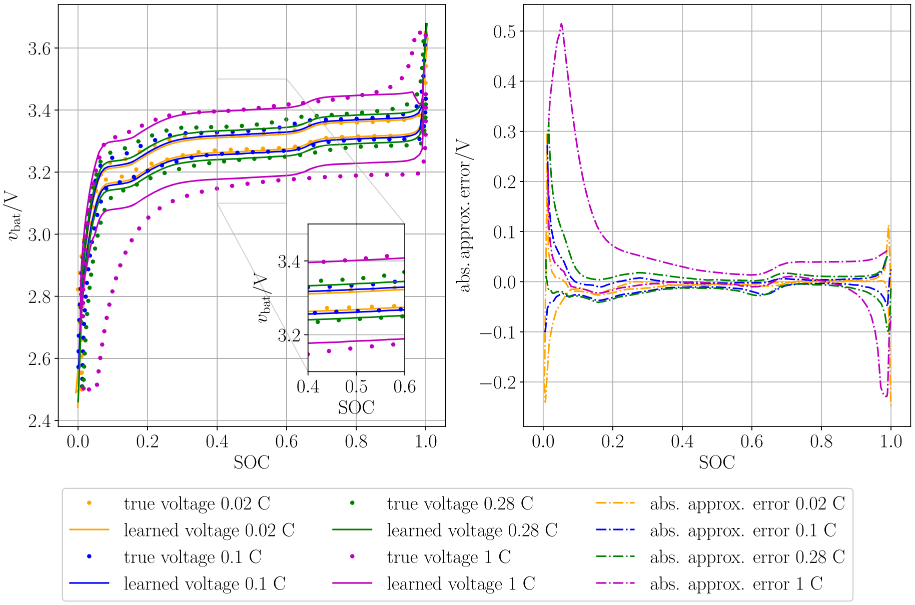

Fig. 5 Simulation results using NODEs for grey-box modelling of a lithium-ion battery in comparison to experimental data; left: CCCV charge and discharge curves for different C-rates at $T = 20$ °C. The lower branches represent discharge (time progresses from right to left), while the upper branches represent charge (time progresses from left to right); right: absolute approximation error

modelling, the test losses are smaller by a factor of around 2.5. Therefore, we can conclude that the generalization ability of the GB model is better. The usage of prior knowledge leads to simpler structures and dependencies to be learned by the BB part of the model.

Finally, we applied the proposed GB modelling framework to an equivalent circuit of a lithium-ion battery. In the 'Grey-box modelling of a lithium-ion battery' section we showed that NODEs can be used for modelling highly nonlinear functions including external variables. We demonstrated how to combine these with ODEs. The simulations show a reasonable agreement with experimental data for low C-rates (0.02 C, 0.1 C and 0.28 C). It is worthwhile noting that the OCV hysteresis typical for LFP cells (Dreyer et al. 2010) can be reproduced with the NODE without requiring an additional physical model equation. This demonstrates the flexibility of the methodology with respect to complex cell behaviour.

For 1 C the end of charging and discharging including the CV phase cannot be approximated in a proper way. Here, the training was difficult because at the beginning and the end of CCCV charging and discharging the voltage curves are very steep and the difference between OCV and output voltage of the battery is small. Some of the measurement curves even crossed the curve of the OCV in these areas. Possible reasons could be measurement inaccuracies, the impact of the capacity-rate effect or temporal changes in battery performance. As outlook, the use of more training data could improve the results. In particular, it would be interesting to use additional data from pulse tests for training and to simulate realistic load profiles. Additionally, more sophisticated ECMs could improve the simulation results. For example, we didn't include a Warburg impedance in our ECM.

We have shown that using NODEs can be a powerful strategy for modelling dynamic systems including external variables. NODEs allow the usage of irregularly-sampled data for training and evaluation. Furthermore, NODEs can be used for GB modelling. We have introduced a framework to combine ODEs and NODEs. This offers new possibilities in GB modelling of dynamic systems.

The approximation capabilities of the GB model using NODEs could be improved further applying regularization and hyperparameter tuning. Beyond this, the direct comparison of simulation results gained from a WB and a GB model using NODEs of a complex dynamic system would be interesting.

Acknowledgements
The authors gratefully acknowledge the financial support of the Carl-Zeiss-Stiftung. The authors thank Jonas Braun and Mehmet Yagci (Offenburg University of Applied Sciences) for their support.

About this supplement
This article has been published as part of Energy Informatics Volume 4 Supplement 3, 2021: Proceedings of the 10th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-4-supplement-3.

Authors' contributions
All authors contributed to the concept and the writing of the paper and they have approved the final manuscript.

Funding
Publication funding was provided by the German Federal Ministry for Economic Affairs and Energy.

Availability of data and materials
Not applicable here.

Declarations

Competing interests

The authors declare that they have no competing interests.

Published: 13 September 2021

References

Almeida GCS, de Souza ACZ, Ribeiro PF (2020) A neural network application for a lithium-ion battery pack state-of-charge estimator with enhanced accuracy. Proceedings 58(1):33

Bailer-Jones CAL, MacKay DJC, Withers PJ (1998) A recurrent neural network for modelling dynamical systems. Netw (Bristol Engl) 9(4):531–547

Charkhgard M, Farrokhi M (2010) State-of-charge estimation for lithium-ion batteries using neural networks and ekf. IEEE Trans Ind Electron 57(12):4178–4187

Che Z, Purushotham S, Cho K, Sontag D, Liu Y (2018) Recurrent neural networks for multivariate time series with missing values. Sci Rep 8(1):6085

Chen M, Rincon-Mora GA (2006) Accurate electrical battery model capable of predicting runtime and i–v performance. IEEE Trans Energy Convers 21(2):504–511

Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud DK (2018) Neural ordinary differential equations. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds). Advances in Neural Information Processing Systems, vol 31. Curran Associates, Inc., Red Hook. pp 6571–6583

Döbel I, Leis M, Vogelsang MM, Neustroev D, Henning P, Riemer A, Rüping S, Voss A, Wegele M, Welz J (2018) Maschinelles Lernen: Eine Analyse zu Kompetenzen, Forschung und Anwendung. Fraunhofer-Gesellschaft, München

Dreyer W, Jamnik J, Guhlke C, Huth R, Moskon J, Gaberscek M (2010) The thermodynamic origin of hysteresis in insertion batteries. Nat Mater 9(5):448–453

Duarte B, Saraiva PM, Pantelides CC (2004) Combined mechanistic and empirical modelling. Int J Chem React Eng 2(1):A3

Dupont E, Doucet A, Teh YW (2019) Augmented neural odes. In: Advances in Neural Information Processing Systems 32 (NIPS 2019), vol 32. Curran Associates, Inc., Red Hook

Estrada-Flores S, Merts I, de Ketelaere B, Lammertyn J (2006) Development and validation of "grey-box" models for refrigeration applications: A review of key concepts. Int J Refrig 29(6):931–946

Fleischer C, Waag W, Heyn H-M, Sauer DU (2014) On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models: Part 1. requirements, critical review of methods and modeling. J Power Sources 260:276–291

Gholami A, Keutzer K, Biros G, Gholaminejad A (2019) Anode: Unconditionally accurate memory-efficient gradients for neural odes. In: International Joint Conference on Artificial Intelligence, IJCAI. International Joint Conferences on Artificial Intelligence, California. pp 730–736

Gusak J, Markeeva L, Daulbaev T, Katrutsa A, Cichocki A, Oseledets I (2020) Towards understanding normalization in neural odes

Haber E, Ruthotto L (2017) Stable architectures for deep neural networks. Inverse Probl 34(1):014004

Haber E, Ruthotto L, Holtham E, Jun S-H (2018) Learning across scales - multiscale methods for convolution neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 32(1). Association for the Advancement of Artificial Intelligence (AAAI), Palo Alto

Haifeng D, Xuezhe W, Zechang S (2009) A new soh prediction concept for the power lithium-ion battery used on hevs. In: 2009 IEEE Vehicle Power and Propulsion Conference. pp 1649–1653

Hamilton F, Lloyd AL, Flores KB (2017) Hybrid modeling and prediction of dynamical systems. PLoS Comput Biol 13(7):1005655

He H, Xiong R, Fan J (2011) Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. Energies 4(4):582–598

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE. pp 770–778. https://doi.org/10.1109/cvpr.2016.90

Hu Y, Yurkovich S, Guezennec Y, Yurkovich BJ (2009) A technique for dynamic battery model identification in automotive applications using linear parameter varying structures. Control Eng Pract 17(10):1190–1201

Jiménez-Bermejo D, Fraile-Ardanuy J, Castaño-Solis S, Merino J, Álvaro-Hermana R (2018) Using dynamic neural networks for battery state of charge estimation in electric vehicles. Proc Comput Scis 130:533–540

Kingma DP, Ba JL (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Krewer U, Röder F, Harinath E, Braatz RD, Bedürftig B, Findeisen R (2018) Review—dynamic models of li-ion batteries for diagnosis and operation: A review and perspective. J Electrochem Soc 165(16):3656–3673

Liao Q, Poggio T (2016) Bridging the gaps between residual learning, recurrent neural networks and visual cortex. arXiv preprint arXiv:1604.03640

Lindskog P, Ljung L (2000) Ensuring monotonic gain characteristics in estimated models by fuzzy model structures. Automatica 36(2):311–317

Ljung L (1999) System identification. In: Webster JG (ed). Wiley Encyclopedia of Electrical and Electronics Engineering. Wiley, New York

Mayur M, Yagci MC, Carelli S, Margulies P, Velten D, Bessler WG (2019) Identification of stoichiometric and microstructural parameters of a lithium-ion cell with blend electrode. Phys Chem Chem Phys PCCP 21(42):23672–23684

Oussar Y, Dreyfus G (2001) How to be a gray box: dynamic semi-physical modeling. Neural Netw 14(9):1161–1172

Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in PyTorch. NIPS 2017 Workshop on Autodiff, Long Beach

Ruthotto L, Haber E (2020) Deep neural networks motivated by partial differential equations. J Math Imaging Vis 62(3):352–364. https://doi.org/10.1007/s10851-019-00903-1

Sohlberg B (2003) Grey box modelling for model predictive control of a heating process. J Process Control 13(3):225–238

Tong S, Klein MP, Park JW (2015) On-line optimization of battery open circuit voltage for improved state-of-charge and state-of-health estimation. J Power Sources 293:416–428

Turetskyy A, Laue V, Lamprecht R, Thiede S, Krewer U, Herrmann C (2019) Artificial neural network enabled p2d model deployment for end-of-line battery cell characterization. In: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN). IEEE, Piscataway. pp 53–58

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ,., Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P (2020) Scipy 1.0: fundamental algorithms for scientific computing in python. Nat Methods 17(3):261–272

Wang Y, Fang H, Zhou L, Wada T (2017) Revisiting the state-of-charge estimation for lithium-ion batteries: A methodical investigation of the extended kalman filter approach. IEEE Control Syst 37(4):73–96

Wu B, Han S, Shin KG, Lu W (2018) Application of artificial neural networks in design of lithium-ion batteries. J Power Sources 395:128–136

Yagci MC, Behmann R, Daubert V, Braun JA, Velten D, Bessler WG (2021) Electrical and structural characterization of large–format lithium iron phosphate cells used in home–storage systems. Energy Technol 9:2001122

Yang D, Wang Y, Pan R, Chen R, Chen Z (2017) A neural network based state-of-health estimation of lithium-ion battery in electric vehicles. Energy Procedia 105:2059–2064

Zhang H, Gao X, Unterman J, Arodz T (2020) Approximation capabilities of neural odes and invertible residual networks. In: International Conference on Machine Learning. Association for Computing Machinery, New York. pp 11086–11095

Zhang T, Yao Z, Gholami A, Keutzer K, Gonzalez J, Biros G, Mahoney MW (2019) Anodev2: A coupled neural ode evolution framework. arXiv preprint arXiv:1906.04596

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.