A Modelica Library for Simulation of Elecric Energy Storages

M. Einhorn[1] F. V. Conte[1] C. Kral[1] C. Niklas[1] H. Popp[1] J. Fleig[2]

[1]AIT Austrian Institute of Technology
Mobility Department, Electric Drive Technologies
Giefinggasse 2, 1210 Vienna, Austria
markus.einhorn@ait.ac.at

[2]Vienna University of Technology
Institute of Chemical Technologies and Analytics
Getreidemarkt 9/164ec 1060 Vienna, Austria

Abstract

This article gives an overview of the Electric Energy Storage (EES) library, which is proposed for inclusion in the Modelica Standard Library. The library contains models with different complexity for simulating of electric energy storages like batteries (single cells as well as stacks) interacting with loads, battery management systems and charging devices. It is shown how the models are defined and how they can be parametrized. Finally, two example simulations are presented.

Keywords: Energy storages, library, battery simulation

1 Introduction

Simulation is a commonly used technique to reduce costs during the design and development process. The energy storage system is a key issue, especially for electric vehicles. Basic models of electric energy storages (EES) are already included in the commercial `SmartElectricDrives` (`SED`) library [1]. The `EES` library, presented in this article, provides basic as well as more complex models for battery cells and for battery stacks. It includes models for battery monitoring and measurement, chargers, loads, sensors and battery management. This library can be used to simulate the behavior of electric energy storages in mobile devices, stationary applications and in transportation systems including hybrid as well as electric vehicles. The models of the `EES` library are designed as universally as possible so that even very specific scenarios can be simulated by varying the parametrization. In the future it is intended to include the `EES` library in the Modelica Standard Library (`MSL`).

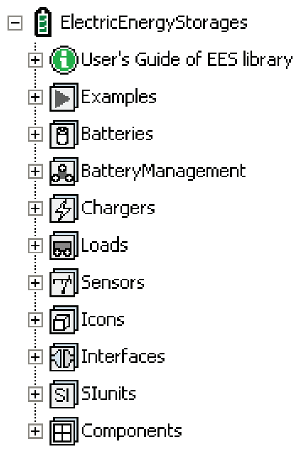


Figure 1: Electric Energy Stroage (EES) library structure

This article shows how the `EES` library is structured, how the fundamental models are defined and which parameters are needed. Finally, two example simulations are presented. For implementing the `EES` library, Dymola 7.4 and Modelica 3.2 are used [2–4].

2 Library structure

The EES library is structured as shown in Fig. 1. The fundamental packages and models are now explained in more detail.

2.1 Batteries

The `Batteries` package contains models for cells as well as for stacks with n_s serially connected cells and n_p cells in parallel. Its structure is shown in Fig. 2.

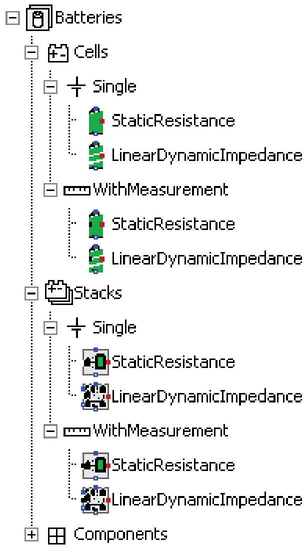


Figure 2: Structure of the `Batteries` package as a part of the EES library.

2.1.1 Cells

In the `Cells` package there are two different types of cell models: `Single` and `WithMeasurement`. While the `Single` cell models are models of the battery cell only, the `WithMeasurement` cell models extend the `Single` cell models with basic measurement. Each of these models can either be a simple cell model with just an ohmic impedance (`StaticResistance`) or a more complex cell model considering basic self discharge, a variable ohmic impedance and a variable number of variable RC elements (`LinearDynamicImpedance`).

Single

The `StaticResistance` single cell model as well as the `LinearDynamicImpedance` single model are shown in Fig. 3 and combined in the `Single` package.

Both single cell models have a positive (`pin_p`), a negative (`pin_n`) and an optional temperature connector (`heatPort`). If the `heatPort` is not used, it can be disabled and the model operates at a fixed temperature ($T_{operational}$). The common parameters of the `StaticResistance` and the `LinearDynamicImpedance` cell models are given in Table 1. For the `StaticResistance` single cell model additionally to Table 1, the parameter given in Table 2 is necessary. Instead, for the `LinearDynamicImpedance` single cell model, additionally to Table 1 the parameters given in Table 3 are necessary.

Table 1: Common input parameters of the `StaticResistance` and the `LinearDynamicImpedance` cell model.

name	unit	description
SOC_{ini}		initial state of charge
$OCVtable$	V	lookup table for the open circuit voltage OCV vs. the state of charge SOC
t_{total}	s	total cell life time
Q_{ini}	C	initial transferred charge
Q_{total}	C	total transferable charge
C_0	C	capacity at T_{ref} for $Q_{abs} = 0$ and $t = 0$
$k_{C\ t}$	C/s	linear t dependency of the capacity C
$k_{C\ Q_{abs}}$	C/C	linear Q_{abs} dependency of the capacity C
x_C		factor at which value of the capacity C $SOH_C = 0$
$useHeatPort$		boolean variable for using the heat port
$T_{operational}$	K	operational temperature if the heat port is not used
T_{ref}	K	reference temperature
$alphaRs$	K^{-1}	linear temperature coefficient of R_s
$alphaC$	K^{-1}	linear temperature coefficient of the capacity C

Table 2: Additional input parameter of the `StaticResistance` cell model.

name	unit	description
R_{sref}	Ω	ohmic resistance at reference temperature T_{ref}

The output variables both for the `StaticResistance` single cell model and the `LinearDynamicImpedance` are given in Table 4 and the calculation of them are presented in the following.

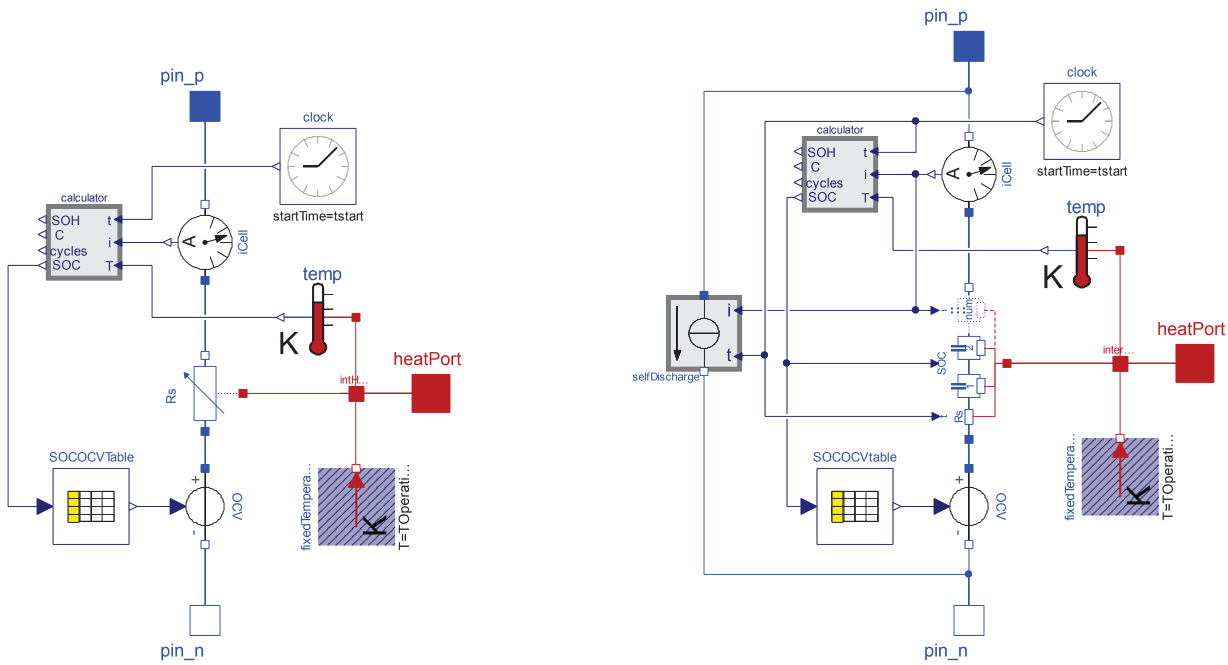
Starting from SOC_{ini} the

$$SOC = SOC_{ini} - \frac{Q}{C}, \tag{1}$$

with the removed charge

$$Q = \int_{t_{start}}^{t_{stop}} I(t)dt. \tag{2}$$

The open circuit voltage OCV of a battery cell changes with SOC and can be extracted from a lookup table $OCVtable$ between the charging voltage limit CVL and the discharging voltage limit DVL. This linearly interpolated lookup table for a lithium ion (Li-ion) battery cell [5] is exemplarily shown in Fig. 4.



(a) `StaticResistance` (b) `LinearDynamicImpedance`

Figure 3: Single cell model with a static impedance (a) and with a variable number of variable RC elements, a variable ohmic impedance as well as with basic self discharge (b)

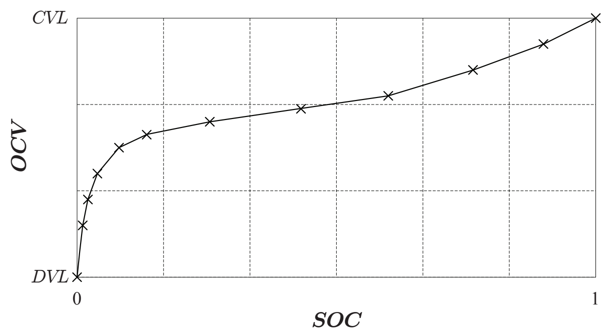


Figure 4: Linear interpolation of the measured open circuit voltage (OCV) for different state of charge (SOC) of a Li-ion battery cell [5].

The total transferred charge between t_{start} and t_{stop} is

$$Q_{abs} = Q_{ini} + \int_{t_{start}}^{t_{stop}} |I(t)| dt \tag{3}$$

and the equivalent number of cycles

$$cycles = cycles_{ini} + \int_{t_{start}}^{t_{stop}} \frac{|I(t)|}{2 \cdot C} dt, \tag{4}$$

with

$$cycles_{ini} = \frac{Q_{ini}}{2 \cdot \frac{C_0 + C(t_{start}, Q_{ini}, T_{heatPort})}{2}}, \tag{5}$$

relates the total transferred charge to the cell capacity C. Therefore, one cycle is equivalent to the charge transfer (regardless in which direction) of one full discharge and one full charge of the current capacity C.

Both cell models consider basic aging, which can be divided into calendaric aging and aging due to cycling. Calendaric aging of a cell is estimated from the time t and the absolute transferred charge Q_{abs} defines the aging due to cycling. Aging of a battery mainly influences the capacity C (decreasing) and the internal impedance (increasing).

The cell capacity

$$C = (C_0 + k_{C\,t} \cdot t + k_{C\,Q_{abs}} \cdot Q_{abs}) \cdot (1 + alphaC \cdot (T_{heatPort} - T_{ref})) \tag{6}$$

is temperature dependent and decreases with increasing time t (calendaric aging) as well as with increasing transferred charge Q_{abs} (aging due to cycling).

The difference between the `StaticResistance` and the `LinearDynamicImpedance` single cell model is the configuration of internal impedance on the one hand and the self discharge on the other hand.

The `StaticResistance` single cell model has just a single, temperature dependent, ohmic

Table 3: Additional input parameters of the `LinearDynamicImpedance` cell model.

name	unit	description
I_{sd0}	A	self discharge current at T_{ref} for $Q_{abs} = 0$ and $t = 0$
$k_{sd\,t}$	A/s	linear t dependency of self discharge current
$k_{sd\,Q_{abs}}$	A/C	linear Q_{abs} dependency of self discharge current
$alphasd$	K^{-1}	linear temperature coefficient of self discharge current I_{sd}
R_{s0}	Ω	series resistance at T_{ref} for $Q_{abs} = 0$ and $t = 0$
k_{R_sSOC}	Ω	linear SOC dependency of Rs
k_{R_st}	Ω/C	linear t dependency of Rs
$k_{R_sQ_{abs}}$	Ω/C	linear Q_{abs} dependency of Rs
num		number of series RC elements
$< R_{d0} >$	Ω	array of length num of R_d at T_{ref} for $SOC = 0$, $Q_{abs} = 0$ and $t = 0$
$< k_{R_dSOC} >$	Ω	array of length num of linear SOC dependency of Rd
$< k_{R_dt} >$	Ω/s	array of length num of linear t dependency of Rd
$< k_{R_dQ_{abs}} >$	Ω/C	array of length num of linear Q_{abs} dependency of Rd
$< alphaRd >$	K^{-1}	array of length num of linear temperature coefficient of R_d
$< C_{d0} >$	F	array of length num of C_d for $SOC = 0$, $Q_{abs} = 0$ and $t = 0$
$< k_{C_dSOC} >$	F	array of length num of linear SOC dependency of Cd
$< k_{C_dt} >$	F/s	array of length num of linear t dependency of Cd
$< k_{C_dQ_{abs}} >$	F/C	array of length num of linear Q_{abs} dependency of Cd
x_Z		factor at which value of the internal, ohmic impedance Z $SOC_Z = 0$

impedance, modeled as

$$R_s = R_{sref} \cdot (1 + alphaRs \cdot (T_{heatPort} - T_{ref})). \quad (7)$$

It does not consider impedance increase due to aging.

In contrast, the `LinearDynamicImpedance` single cell model has an ohmic impedance and num serially connected RC elements for the transient behavior of the electrodes of an electrochemical energy storage as shown in Fig. 5 [6, 7]. All ohmic impedances (R_s, $R_{d\,1} \ldots R_{d\,num}$) are temperature dependent and have a linear dependency on state of charge SOC, on the time t (calendaric aging) as well as on the transferred charge Q_{abs} (aging due to cycling). The serial resistor R_s is modeled as

$$R_s = (R_{s0} + k_{R_sSOC} \cdot SOC + k_{R_st} \cdot t + k_{R_sQ_{abs}} \cdot Q_{abs}) \cdot (1 + alphaRs \cdot (T_{heatPort} - T_{ref})). \quad (8)$$

For $n = 1 \ldots num$

$$R_d[n] = (R_{d0}[n] + k_{R_dSOC}[n] \cdot SOC + k_{R_dt}[n] \cdot t + k_{R_dQ_{abs}} \cdot Q_{abs}) \cdot (1 + alphaRd[n] \cdot (T_{heatPort} - T_{ref})). \quad (9)$$

and since there is no temperature dependency for capacitances considered

$$C_d[n] = C_{d0}[n] + k_{C_dSOC}[n] \cdot SOC + k_{C_dt}[n] \cdot t + k_{C_dQ_{abs}}[n] \cdot Q_{abs}. \quad (10)$$

Table 4: Calculated output variables of the `StaticResistance` and the `LinearDynamicImpedance` cell model.

name	unit	description
SOC		state of charge
OCV	V	open circuit voltage
Q_{abs}	C	total transferred charge
$cycles$		number of equivalent cycles
t	s	calendaric cell time
SOH		state of health
SOS		state of sickness
C	C	capacity
V	V	cell voltage

Moreover, the `LinearDynamicImpedance` single cell model considers basic self discharge which is linear dependent on the temperature, the time (calendaric aging) and the transferred charge (aging due to cycling).

While with the `StaticResistance` single cell model only the basic impedance behavior can be simulated, the `LinearDynamicImpedance` can be used to simulate single cells very accurate if the parametrization work is well done.

The capacity as well as the internal impedance (only with the `LinearDynamicImpedance` single cell model) can change due to aging and the state of health SOH (and accordingly the state of sickness SOS) compares the current condition of a battery cell to its ideal (initial) condition. The SOH is divided into the SOH_C and SOH_Z:

$$SOH = 1 - SOS = SOH_C \cdot SOH_Z, \quad (11)$$

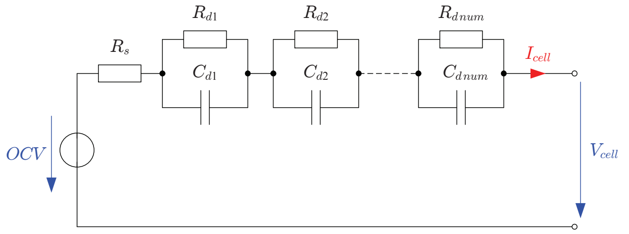


Figure 5: Battery model with one ohmic impedance and num serially connected RC elements.

as shown in Fig. 6.

For a new cell, $SOH_C = 1$ and $SOH_Z = 1$ and therefore also $SOH = 1$. When the capacity C decreases (due to calendaric aging or aging due to cycling) to $x_C \cdot C_0$ (e.g. with $x_C = 0.8$), $SOH_C = 0$. Hence,

$$SOH_C = \frac{1}{C_0 \cdot (1 - x_C)} \cdot C - \frac{x_C}{1 - x_C}. \quad (12)$$

Similarly, when the sum of all internal, ohmic impedances

$$Z = R_s + R_d[1] + R_d[2] + \ldots + R_d[num] \quad (13)$$

increases (due to calendaric aging or aging due to cycling) to $x_Z \cdot Z_0$ (e.g. with $x_Z = 2$), where

$$Z_0 = R_{s0} + R_{d0}[1] + R_{d0}[2] + \ldots + R_{d0}[num], \quad (14)$$

$SOH_Z = 0$. Therefore,

$$SOH_Z = \frac{1}{Z_0 \cdot (1 - x_Z)} \cdot Z - \frac{x_Z}{1 - x_Z}. \quad (15)$$

Fig.7 shows the dependency of SOH_C and SOH_Z from C and Z, respectively.

WithMeasurement

In the `WithMeasurement` package there are two cell models with measurement: The `StaticResistance` and the `LinearDynamicImpedance`. Both contain basic measurement and instances of the corresponding single cell model. Fig. 8 shows for example the `LinearDynamicImpedance` with measurement. The voltage, the current and the temperature of the single cell model are measured and provided with the `singleCellBus` (cf. section 2.7). The cell models with measurement have the same connectors as the single cell models (`pin_p`, `pin_n` and `heatPort`) but have additionally the `singleCellBus`. Therefore several instances of these can be connected

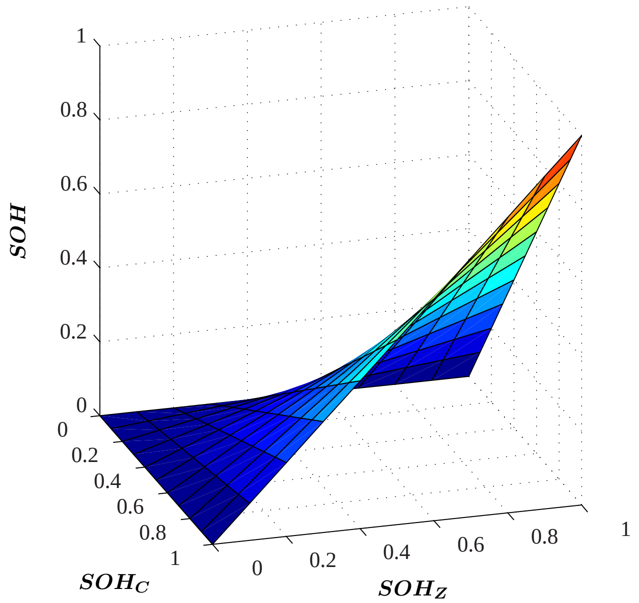
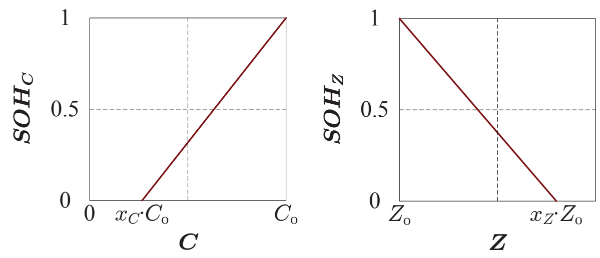


Figure 6: Dependancy of the state of health SOH from SOH_C and SOH_Z.



(a) SOH_C (b) SOH_Z

Figure 7: Partial state of health with respect to the capacity C and to the impedance Z with paramters $0 < x_C < 1$ and $x_Z > 1$.

together (serially and in parallel) and the basic measurement of each cell is done.

The advantage of separating the cell models in `Single` and `WithMeasurement` models shows up when it comes to stacks and the current of each cell should be measured. Fig. 9 shows the icons for the `Single` and `WithMeasurement` for the `StaticResistance` cell as well as for the `LinearDynamicImpedance` cell models with enabled `heatPort`.

2.1.2 Stacks

Stacks are n_s serially connected cells and n_p cells in parallel as shown in Fig. 10. The `Stacks` package is structured in the same way as the `Cells` package. There are `Single` and `WithMeasurement` stacks where each uses either the `StaticResistance` or

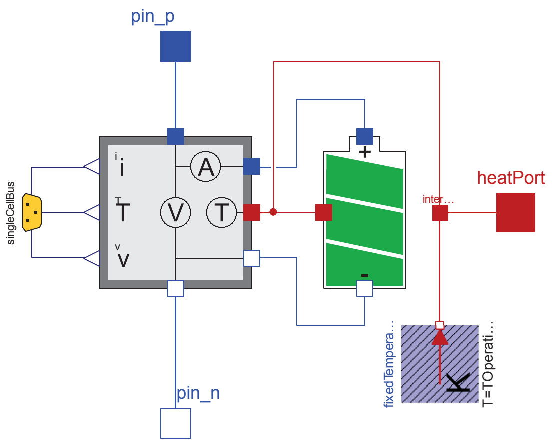


Figure 8: Cell model with measurement. The measured current, temperature and voltage are provided to the outside via the `singleCellBus`

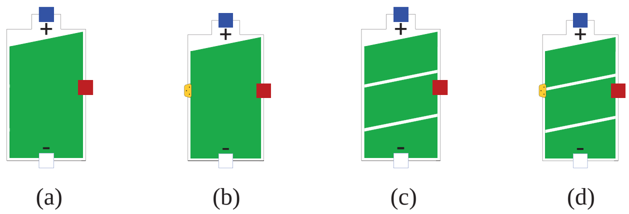


Figure 9: Icons for the `StaticResistance` (`Single` (a), `WithMeasurement` (b)) and for the `LiniearDynamicImpedance` (`Single` (c), `WithMeasurement` (d)) cell model with enabled heat port.

the `LinearDynamicImpedance` cell models.

The `StaticResistance` single stack model is a stack with all equal cells and therefore only one cell needs to be calculated and parameterized. It is basically a `StaticResistance` single cell model that scales the parameters of the model components and behaves like $n_s \cdot n_p$ equally parameterized instances of `StaticResistance` single cell models. For example the value of the series resistance is then $R_{s0} \cdot n_s / n_p$ with the value R_{s0} for a single cell. Hence, it is much faster than a model with $n_s \cdot n_p$ instances of equally parametrized cell models.

The `LinearDynamicImpedance` single stack model has $n_s \cdot n_p$ instances of the `LinearDynamicImpedance` single cell model and the serial and parallel connections are textually generated with loops:

```
equation
//series connection
for s in 1:ns-1 loop
 connect(cell[s,1].pin_n,cell[s+1,1].pin_p);
end for;
```

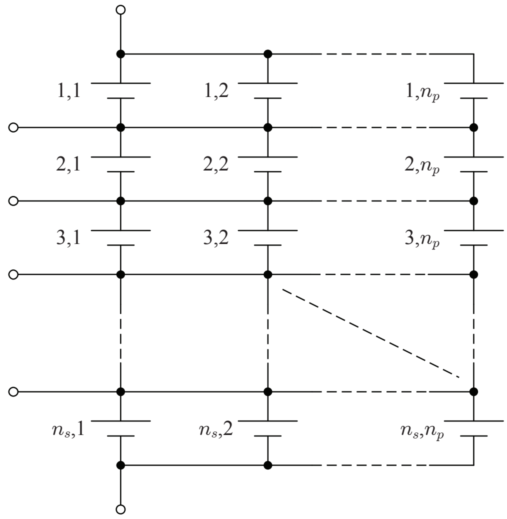


Figure 10: Battery stack with n_s serially and n_p connected cells.

```
//parallel connection
for p in 1:np-1 loop
 for s in 1:ns loop
 connect(cell[s,p].pin_p,cell[s,p+1].pin_p);
 connect(cell[s,p].pin_n,cell[s,p+1].pin_n);
 end for;
end for;
//connector connection
for s in 1:ns loop
 connect(cell[s,1].pin_p,pin_pCell[s]);
 connect(cell[s,1].pin_n,pin_nCell[s]);
end for;
//top connection
connect(cell[1, 1].pin_p, pin_pPackage);
//bottom connection
connect(cell[ns, np].pin_n, pin_nPackage);
//heatPort connection
connect(cell[:,:].heatPort,heatPort[:,:]);
```

Each scalar parameter of the `LinearDynamicImpedance` single cell model is extended to an array of dimension $n_s \times n_p$ and the array parameters of the `LinearDynamicImpedance` single cell model now have the dimension $n_s \times n_p \times num$. Each cell in this stack can be parametrized individually and therefore also cell variance in a stack can be simulated. All single cell connectors and all temperature connector are conditionally available. Fig. 11 shows the icons for all different stack models. Therefore a separate thermal model can be considered. With the single cell connectors and an additional model even cell balancing can be simulated [8].

From each single stack model (`StaticResistance` and `LinearDynamicImpedance`) there is also a version with measurement (`StaticResistance`

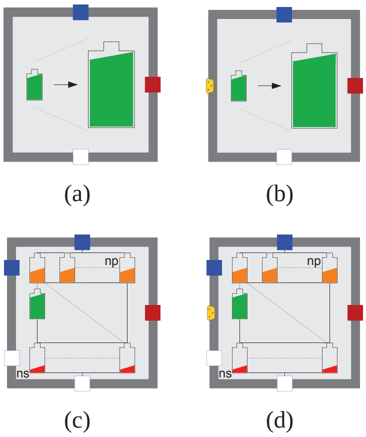


Figure 11: Icons for the `Single` (a), `WithMeasurement` (b) `StaticResistance` stack models with enabled heat ports and the `Single` (c), `WithMeasurement` (d) `LinearDynamicImpedance` stack models with enabled heat as well as single cell ports .

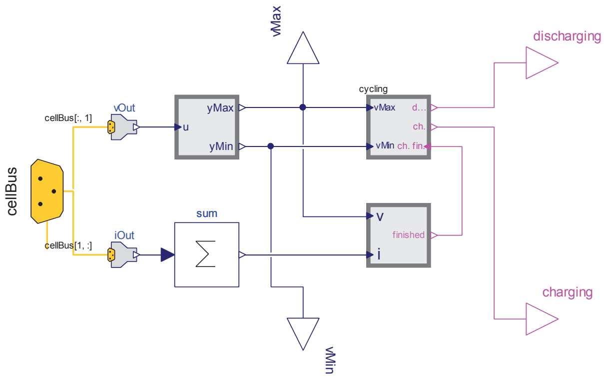


Figure 12: `VoltageCycling` model in the `BatteryManagement` package.

Table 5: Parameters of the `VoltageCycling` battery management model.

name	unit	description
n_s		number of serially connected cells
n_p		number of parallel connected cells
I_{final}	A	final charging switch off current
V_{max}	V	maximal cell voltage
V_{min}	V	minimal cell voltage
t_c	s	delay time after charging
t_d	s	delay time after discharging
ini		true for initial discharging

and `LinearDynamicImpedance`) in the `Stacks` package.

2.2 Battery Management

In the current version of the EES library there is a voltage cycling device (`VoltageCycling`) implemented. Fig. 12 gives an overview about the `VoltageCycling` model and its parameters are given in Table 5

It basically has two boolean outputs to operate with loads and charging devices (cf. section 2.3 and 2.4). The `cellBus` can either be connected to a cell or a stack. During charging if V_{max} of any cell is reached and the charging current of all parallel connected cells is below I_{final}, the boolean output `discharging` gets true after t_c. When V_{min} of any cell is reached during discharging, the boolean output `charging` gets true after t_d. Therefore it is possible to cycle a cell or a stack within its voltage limits V_{min} and V_{max} (in combination with a charging device and a load).

2.3 Chargers

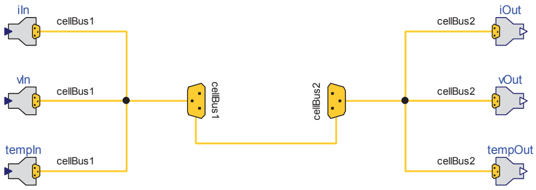
In the `Chargers` package there is a constant current, constant voltage (CCCV) charging device modeled. A CCCV charging device charges a battery with a constant current until its CVL is reached (constant current phase). Then a constant voltage is applied and the current decreases (constant voltage phase). The charging device is switched off when the charging current reaches the final charging switch off current (typically 5% of the 1h discharging current).

When the boolean input of the modeled charging device is true (e.g. the boolean charging output `charging` from the `VoltageCycling` model described in section 2.2), a current source provides a constant current. This current is controlled by a voltage input (e.g. the maximal cell voltage in a battery stack), a reference voltage (parameter V_{max}) and limited by the maximal charging current (parameter I_{max}).

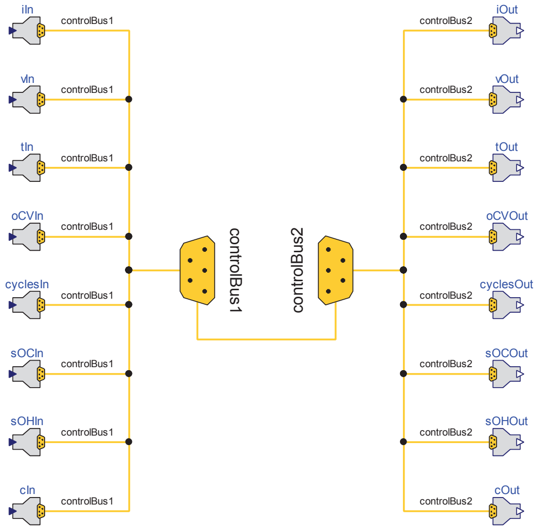
2.4 Loads

In the `Loads` package there are models to discharge a cell or a stack to cover typical lab situations (for example discharging a battery with a defined current or power profile). In extension to the electrical current sources (a current source parametrized with a negative current can be used as a load) there are four different loads considered:

- `BooleanExternalControlledLoad`
- `BooleanConstantCurrent`
- `BooleanConstantPower`



(a) `CellBus`



(b) `ControlBus`

Figure 13: Structure of the `CellBus` (a) and the `ControlBus` (b) with bus adaptors to all variables on the bus (v, i, T for the `CellBus` and v, i, T, OCV, SOC, SOH, $cycles$, C for the `ControlBus`).

- `SignalPower`

All `Boolean` models have a boolean input `on` which for example can be connected to the boolean output `discharging` of the `VoltageCycling` model (cf. section 2.2). If `on` is false the cell or the stack is not being discharged.

2.5 Sensors

The `Sensors` package provides models to estimate the energy, the charge and the absolute charge from/to a cell or a stack. The models provided in the `Electrical.Analog.Sensors` package from the `MSL` are used and consequently extended.

2.6 Icons

In this package there are combined all icons used for the packages.

Figure 14: Test cell at constant temperature in a climate chamber.

2.7 Interfaces

There are two buses available: The `CellBus` and the `ControlBus`. The `CellBus` contains only measurable variables such as the cell voltage v, the cell current i and the cell temperature T. It can be used when lab situations are simulated and only measurable variables are significant. It can also be used to test different battery monitoring systems, based on i, v and T. The `ControlBus` contains the variables from the `CellBus` and additional the following: OCV, SOC, SOH, $cycles$ and C. It could for example be the communication between the battery management system and the control units in an electric vehicle simulation.

In the `Interfaces` package there are also bus adapters to extract/inject all these variables (`Real` values) from/to the `CellBus` as well as from/to the `ControlBus`. Figure 13 shows the structure of the `CellBus` and of the `ControlBus` with the usage of all bus adapters.

3 Examples

Two different examples are presented:

First a `LinearDynamicImpedance` single cell model with one ($num = 1$) RC element is parameterized according to [9]. A realistic current profile, gained from the FTP72 cycle is continuously applied to the parameterized model as well as to the real cell [10]. The FTP72 cycle is a standardized real life driving cycle that simulates an urban route of $12.07\,\mathrm{km}$.

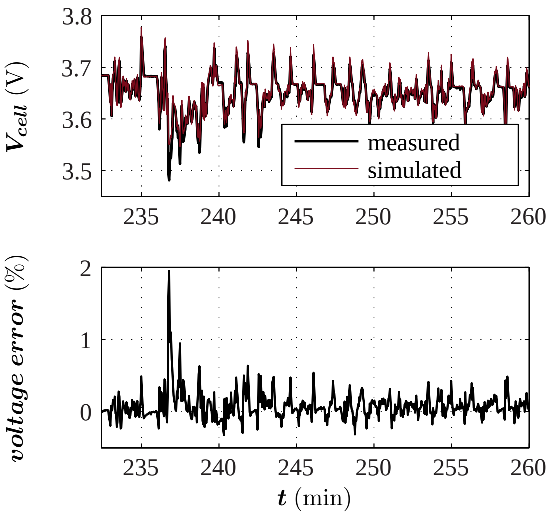


Figure 15: Comparison of the measured and simulated cell voltage when applying the FTP72 current profile to the real cell and to parameterized model (top). The error refers to the measured voltage (bottom).

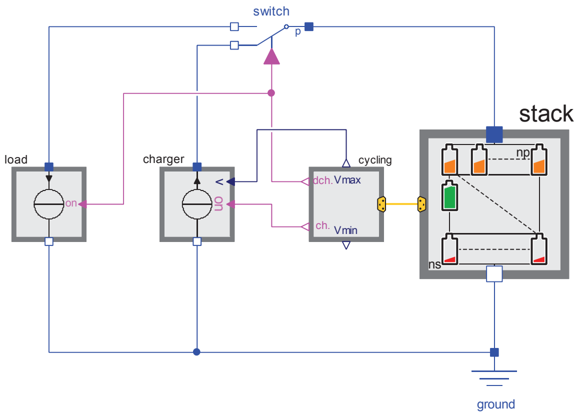


Figure 16: Simulation arrangement for cycling a battery stack.

This cycle is applied to a compact electric vehicle to extract the realistic current profile. The current profile is repeated starting from the full charged cell until it reaches discharging voltage limit DVL. This experiment is performed at constant temperature and Fig. 14 shows the test cell in a climate chamber. Fig. 15 compares the simulated to the measured cell voltage. The error, related to the measured voltage, between the simulated and the measured voltage stays below 2% at $SOC \approx 0.5$ during one complete FTP72 cycle. Therefore the chosen model approach is appropriate.

For the second example, three serially connected cells using the `LinearDynamicImpedance` stack model (without temperature and cell connectors) with different capacities are simulated. The cell capacities are $C_1 = 35\,\mathrm{Ah}$, $C_2 = 40\,\mathrm{Ah}$, $C_3 = 45\,\mathrm{Ah}$ and

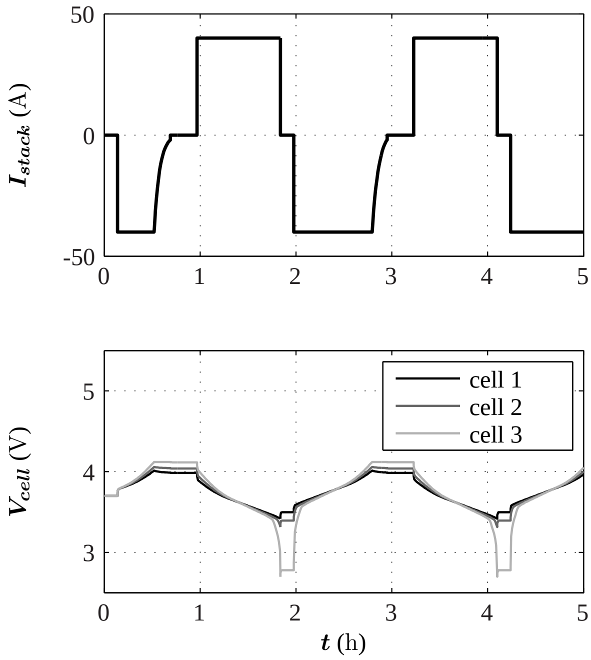


Figure 17: Current profile and voltage response from the three cells from the example simulation.

parametrized using [5, 9]. The cells are cycled with a 40 A charging and a 40 A discharging current between $CVL = 4.1\,\mathrm{V}$ and $DVL = 2.7\,\mathrm{V}$ with a delay after charginging $t_c = 100\,\mathrm{s}$ and after discharging $t_d = 50\,\mathrm{s}$. Cycling means continuously charging and discharging, which is a typical lab application for batteries. Fig. 16 shows the simulation arrangement with the charging device (`charger`), the battery management (`cycling`), the battery stack (`stack`), the discharger (`load`) and a switch (`switch`). Fig. 17 shows the battery stack current and the voltage from all three cells.

4 Conclusions

The Electric Energy Storage library, which is intended to be included in the Modelica Standard Library, has been described. The structure of the library and key parameters as well as the equations of the models for determining the output variables have been presented. Two typical lab tests for batteries have been simulated as an example to show the interaction of the models and to show the correctness of the model approach.

With the Electric Energy Storage library the user has a powerful tool to cover various applications of energy storages with simulation. In combination with existing libraries, (e.g. the `SED`) the behavior of electric energy storages and the interaction for example with the powertrain of an electric vehicle can be analyzed in detail. Future work will focus on including of thermal models of electirc energy storages and the coupling with the electrical models.

References

[1] J. V. Gragger, H. Giuliani, C. Kral, T. Bäuml, H. Kapeller, and F. Pirker, "The smartelectric-drives library - powerful models for fast simulations of electric drives," *Proceedings of the 5th International Modelica Conference*, vol. 2, pp. 571–577, 2006.

[2] Modelica Association, "Modelica - a unified object-oriented language for physical systems modeling, language specification version 3.2," *http://www.Modelica.org/*, 2010.

[3] Dassault Systems, "Dymola 7.4," *http://www.3ds.com/*, 2010.

[4] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. IEEE Press, Wiley-Interscience, 2004.

[5] *EIG ePLB C020 lithium ion polymer cell datasheet*, EIG, 2010.

[6] A. Jossen, "Fundamentals of battery dynamics," *Journal of Power Sources*, vol. 154, pp. 530–538, 2006.

[7] S. Buller, "Impedance-based simulation models for energy storage devices in advanced automotive power systems," *Aachener Beitraege des ISEA*, vol. 31, 2002.

[8] M. Einhorn and J. Fleig, "Improving of active cell balancing by equalizing the cell energy instead of the cell voltage," *Proceedings of the 25th International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium and Exposition (EVS25)*, 2010.

[9] M. Einhorn, F. V. Conte, C. Kral, J. Fleig, and R. Permann, "Parametrization of an electrical battery model for dynamic system simulation in electric vehicles," *Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC)*, September 2010.

[10] *FTP72 Urban Dynamometer Driving Schedule (UDDS)*, U.S. Environmental Protection Agency.