Faster than Real-Time Electro-Thermal-Aging Emulation of Multiple Batteries within a Modular Multilevel Converter

Lars Leister[1], Niklas Katzenburg[1], Kai Kuhlmann[2], Lukas Stefanski[1], Marc Hiller[1]

[1]Karlsruhe Institute of Technology (KIT)
Elektrotechnisches Institut (ETI)
Karlsruhe, Germany

[2]Aschaffenburg University of
Applied Sciences
Aschaffenburg, Germany

E-Mail: lars.leister@kit.edu

ACKNOWLEDGMENT

The authors acknowledge the financial support by the Federal Ministry for Economic Affairs and Climate Action of Germany in the project LeMoStore (project number 03EI4031A-E). Furthermore, they acknowledge the data provided by Mainsite GmbH & Co. KG and Batemo GmbH within the project and used in this paper.

***Index Terms*—Modular Multilevel Converters (MMC), Energy Storage, Hardware-In-the-Loop (HIL), Energy Management, Aging.**

***Abstract*—A real-time capable emulation of a modular multilevel converter with integrated batteries for hardware-in-the-loop applications is presented. The modeling and interfacing of electrical, thermal, and aging behavior of multiple batteries are shown. The capability for emulation faster than real-time is demonstrated, which benefits the development of energy management strategies.**

I. INTRODUCTION

Stationary battery energy storage systems (BESSs) become increasingly relevant for the electrical grid, especially due to volatile renewable energy sources. Traditional BESS, consisting of a fixed arrangement of batteries are simple yet inflexible. Especially for the combination of differently aged second-life batteries, a reconfigurable battery topology offers advantages. Based on the topology of the modular multilevel converter (MMC) [1], multiple concepts with integrated batteries have been discussed [2]–[5]. In addition to the capacitors in the submodules of the MMC, battery cells or battery modules are integrated. When integrated into the power grid, the battery-integrated MMC not only serves as a power converter but also as an energy storage system. The topology's modularity gives flexibility, so that the combination of batteries with different capacities and characteristics is possible, allowing the economic and sustainable use of second-life batteries. Compared to other reconfigurable battery topologies, this comes at lower additional costs, assuming the converter is required in the grid anyway.

Combining power converter and energy storage in this modular way requires an adaptation of the MMC control algorithms. In comparison to submodules with solely capacitive energy storage devices, submodules with batteries greatly reduce the voltage volatility. Thus, the typically complex energy control of an MMC is simplified. In turn, the energy stored in the batteries must be managed. Possible objectives of the energy management include peak shaving or provision of balancing power for the grid. It can also include the balancing of the state of charge (SoC) [6] or the state of health (SoH) [7]. Generally, the battery-integrated MMC requires both, MMC control strategies and energy management strategies.

A hardware-in-the-loop (HiL) testbench simplifies the development and testing of such systems under realistic conditions. It allows safe and fast testing of both, control and energy management algorithms while using the actual control platform. A disadvantage lies within the limitation to real-time, making it time-consuming to test energy management algorithms over longer time scales. The approach of faster than real-time (FTRT) emulations reduces this limitation [8]. Models running faster than real-time accelerate testing. Additionally, those models can potentially be used for predictions in control algorithms. For battery cells, faster than real-time has been discussed in [9] and is applied for HiL in [10]. For an MMC it has been discussed in [11].

In contrast to the above, this paper focuses on the emulation of multiple battery modules simultaneously, including thermal and aging models while being interconnected by the MMC. The presented approach closes the gap between realistic real-time emulations and simplified yet fast simulations. This is achieved by means of electrical and thermal equivalent circuit methods and a real-time capable rainflow-counting algorithm for aging. The implementation relies on a combination of models running on a field-programmable gate array (FPGA) and models running on a processor. An emulation with an exemplary control method concludes this paper.

II. Emulated System and Interface Definition

The scope of the emulation is defined according to a real setup currently in development. The control structure is hierarchical, shown in Fig. 1. It is centered around a central control unit (CCU) implemented on a system-on-chip platform. From there, six FPGAs are controlled via universal asynchronous receiver transmitter (UART) protocol over full-duplex optical fiber. Each of the FPGAs is responsible for the control of one MMC arm.

The FPGA in turn connects via full-duplex optical fiber to all power electronic storage blocks (PESBs) in the corresponding arm. Each arm has 20 PESBs as shown in Fig. 2, which consist of a MOSFET H-bridge, a battery module, capacitors and peripherals. In every PESB, the control of the power electronics and the communication with the battery module's battery management system (BMS) of this PESB is realized by means of a microcontroller.

The HiL emulator is intended to emulate the complete PESB, including the microcontroller. Therefore, the emulator inputs are defined as optical inputs given by the described arm FPGAs. For the sake of simplicity, only one of the six arms is emulated completely. For the remaining five arms, the optical signal coming from the CCU is directly fed to the emulator, allowing the emulation complexity of those arms to be greatly reduced as not all 20 but only one reference battery model is calculated for those arms. Moreover, a redundant high-speed GTX interface over SATA between the HIL-system and the MMC CCU is used as a fallback and for certain FTRT-scenarios requiring reduced latency. To summarize, the combination of CCU and arm FPGA is considered as the device under test (DUT).

III. Emulator Platform and Modelling

The emulator platform is based on the Zynq 7030 System-on-Chip, combining an FPGA and two ARM processors [12]. Therefore, models can be implemented

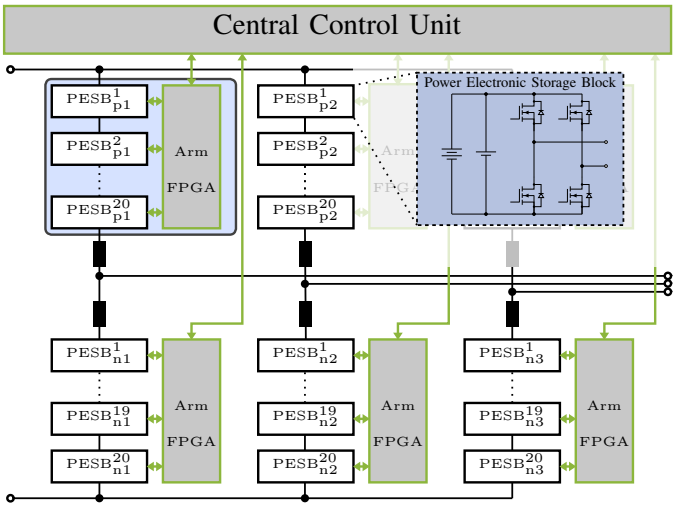


Fig. 1. Topology of the Emulated System.

on either the FPGA or on one of the ARM cores and can be interconnected as required. While the FPGA is suitable for models with short time constants, the ARM core is typically chosen for mathematically more complex models with larger time constants. The platform also offers the flexibility to choose between various interfaces by using different expansion cards. For this paper, the connection is realized via GTX over SATA cable and UART over optical fiber. The emulator also features digital and analog electrical outputs to emulate measurement signals, e.g., current and voltage sensors. However, those are not considered in this paper. The emulated components including an MMC represented in the state-space, an H-Bridge with the respective modulator, capacitors, and batteries are shown in Fig. 2. Their respective modeling is addressed in the following.

A. Electrical Models

The electrical models are implemented on the FPGA. The maximum clock rate used is $100\,\mathrm{MHz}$. For models running at lower clock rates, resource-sharing between models is applied. In the following, the models of the power electronics and the energy storage are shown.

1) Power electronics: First, the models of the power electronics are discussed, shown in green in Fig. 2. The reference voltage values $U_{\mathrm{PESB,ref}}$ given by the CCU or the arm FPGA of the DUT are fed into the modulator model, representing functionality implemented on the PESB microcontroller in the real system. The modulator is implemented with a carrier signal frequency of $8\,\mathrm{kHz}$ and a triangular counter resolution of $10\,\mathrm{ns}$ to generate the pulse-width modulation (PWM) signals. Those gate signals D are fed into an H-bridge model. The

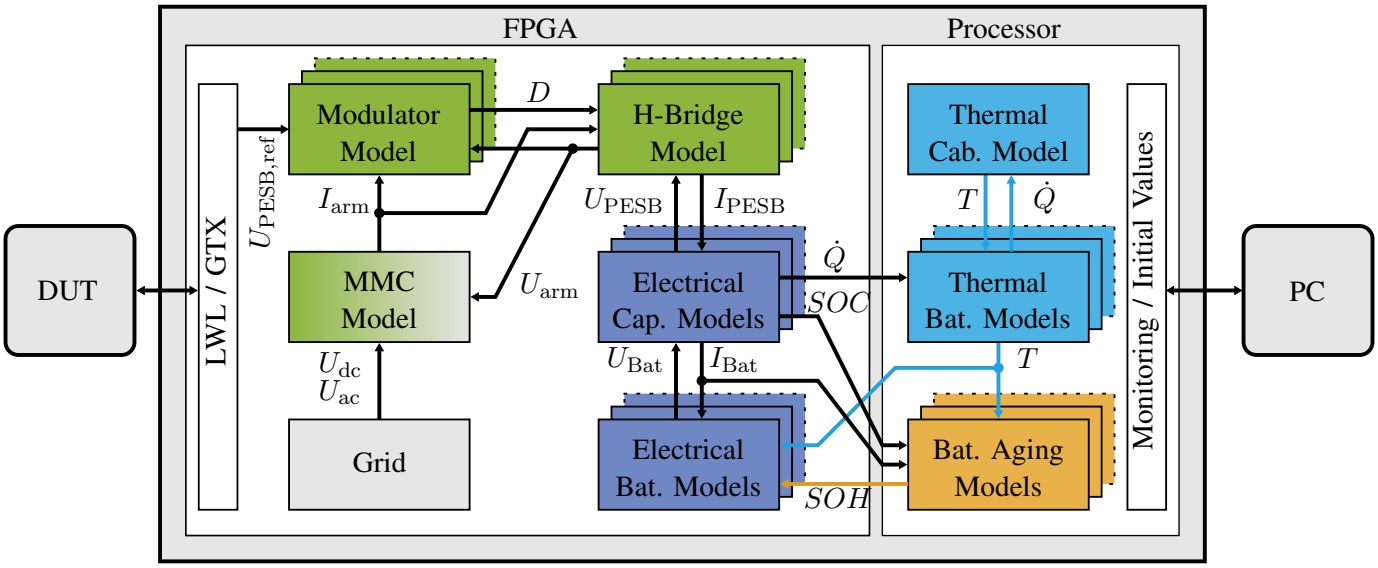


Fig. 2. Schematic overview of the Emulator. Signal Paths to DUT and to Monitoring Block are not shown for clarity.

MOSFETs in the H-bridge are not modeled in detail. Their behavior is represented by idealized switching of voltages and currents at times given by the modulator model. Blanking times and others are neglected. Due to its simplicity, the H-bridge model can be executed at 100 MHz in parallel on the FPGA, taking only four clock cycles (40 ns) from input to output.

The MMC is modeled in state-space according to [13] and calculates the arm currents I_{arm} from the arm voltages U_{arm}. The latter are given as the sum of all voltages calculated by the H-bridge models in the respective arm. They are downsampled from 100 MHz to 10 MHz at which the MMC model is running. The MMC model requires 270 ns open-loop time from input to output. AC and DC voltages are fed into the MMC model from a basic grid model. The grid is implemented as an open-loop model with fixed values, meaning that the voltages U_{ac} and U_{dc} are not affected by the MMC. The MMC model and its implementation on an FPGA are presented in [5].

2) Energy Storage: The electrical storage model consists of a capacitor model and a model of a battery module, both shown in blue in Fig. 2. To model the battery modules, 1RC cell models as shown in Fig. 3 are used. For general statements, 1RC models are sufficient, while for more precise dynamic modeling, 2RC models are also implemented but not considered here. The models are parameterized based on measurement. The measurement results are stored within multi-dimensional lookup tables (LUTs) for the open circuit voltage U_{ocv}, internal resistance R_{i} and polarisation capacity C_{p} and

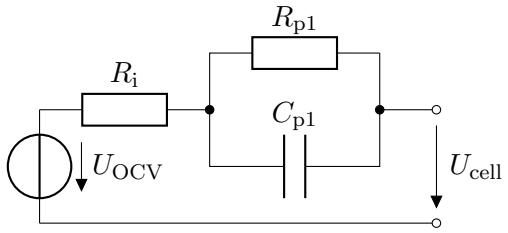


Fig. 3. 1RC Equivalent Circuit of a Battery Cell.

resistance R_{p}. The LUTs represent dependencies on SoC, current and temperature with

$$U_{\mathrm{ocv}} = f(SoC, T), \tag{1}$$

$$R_{\mathrm{i}}, R_{\mathrm{p1}}, C_{\mathrm{p1}} = f(SoC, T, I). \tag{2}$$

The individual battery model's capacity and internal resistance R_{i} are additionally dependent on the Capacity-SoH ($\mathrm{SoH_C}$) and Resistance-SoH ($\mathrm{SoH_R}$) as they are later defined in section III-C. The SoC is calculated by coulomb counting. To scale the model from cell to module level, it is assumed that all cells behave equivalently. Based on this, scaling of voltage and current is sufficient for module modeling. Additionally, module resistances introduced by BMS components and connectors are considered. The battery model interacts with a capacitor model governed by the equation

$$U_{\mathrm{C}} = \frac{1}{C} \cdot \int I_{\mathrm{C}} \, \mathrm{d}t. \tag{3}$$

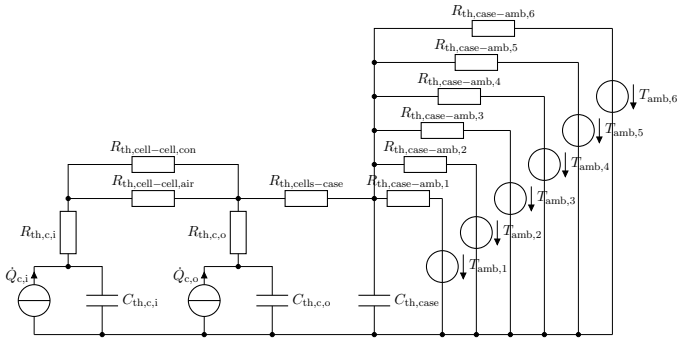
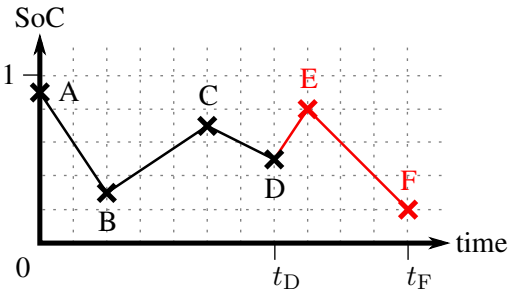


Fig. 4. Equivalent Circuit of the Thermal Battery Module Model.



Evaluation at t_{D}	Evaluation at t_{F}
A-B: Half Cycle	A-F: Half Cycle
B-C: Half Cycle	B-E: Full Cycle
C-D: Half Cycle	C-D: Full Cycle

Fig. 5. Demonstration of the Rainflow-Counting Algorithm.

Due to the model's complexity and multi-dimensional LUTs, parallel execution of multiple capacitor and battery models is hardly possible. One of the models requires around 15 % of the available 125,000 logic cells on the FPGA. Therefore, the LUTs stored in block RAM are reused by serializing the access to them. In this paper, 20 models of battery and capacitor combined share identical LUTs, registers and DSP slices, thus effectively reducing the resource usage at the expense of calculation time. The models are executed at a clock rate of 10 MHz, each of the 20 serialized models is therefore executed at 500 kHz. The open-loop time of this model is 430 ns.

B. Thermal Model

For the thermal behavior of a battery module, the power loss generated per cell is considered. With

$$\dot{Q} = I \cdot (U_{\mathrm{cell}} - U_{\mathrm{ocv}}) \tag{4}$$

the heat flow generated is calculated as the power loss in the internal battery resistance from the battery's open circuit voltage U_{ocv} and terminal voltage U_{cell}. It is assumed, that each battery module can be separated into two groups of cells. One of those groups is fully surrounded by other cells, while the other is only partially surrounded. This results in a compromise of precision and computational complexity as shown by [14], [15].

The heat transfer is modeled using the equivalent circuit model shown in Fig. 4. The heat flow generated by the two groups of cells is given as $\dot{Q}_{\mathrm{c,i}}$ and $\dot{Q}_{\mathrm{c,o}}$, respectively. The thermal capacity and thermal resistance of the individual cells are derived from measurements of single cells, which are not discussed here. The two groups of cells are interconnected by air and electrical connectors, whose thermal resistance is calculated based on their dimensions and material properties. The battery module includes a metal housing with a heat capacity given by $C_{\mathrm{th,case}}$. Each of the six module surfaces is thermally connected to its surrounding air. The ambient temperatures of those virtual air volumes are given by $T_{\mathrm{amb,x}}$ with $x = 1..6$. The 20 modules of one arm are placed in one cabinet, where they are thermally connected by air. The thermal network has not yet been validated by measurement.

C. Aging model

To emulate the aging of the lithium-ion batteries, a rainflow-counting algorithm is adopted and parameterized based on literature [16]. Rainflow-counting is an established approach to determine the damage introduced by load cycles. The algorithm determines the number of full and half cycles including their respective depth of discharge (DoD), current, and temperature. Based on this data, damage tables are used to estimate the resulting total damage. To determine whether a cycle is a half or full cycle, the original algorithm requires a complete dataset. To use it in real-time, an adaptation is necessary. In this paper, the concept of residues is used [17], [18].

The residues describe all past half cycles that have not yet been included in a full cycle. The resulting aging factor is not continuous, as the load cycles that have been previously counted as half cycles can become full cycles as soon as the conditions are met. This can lead to a discontinuous increase in aging as shown in the table in Fig. 5. Evaluating the algorithm at time t_{D} results in B-C being considered a half cycle, while it becomes a larger full cycle B-E after reevaluation at t_{F}. For each half and full cycle, the DoD, root mean square (RMS) current I_{RMS} and average temperature $\overline{T}$ are calculated.

Based on those values, the effect of aging is derived as the weighted current throughput $Q_{\mathrm{w},i}$, where the

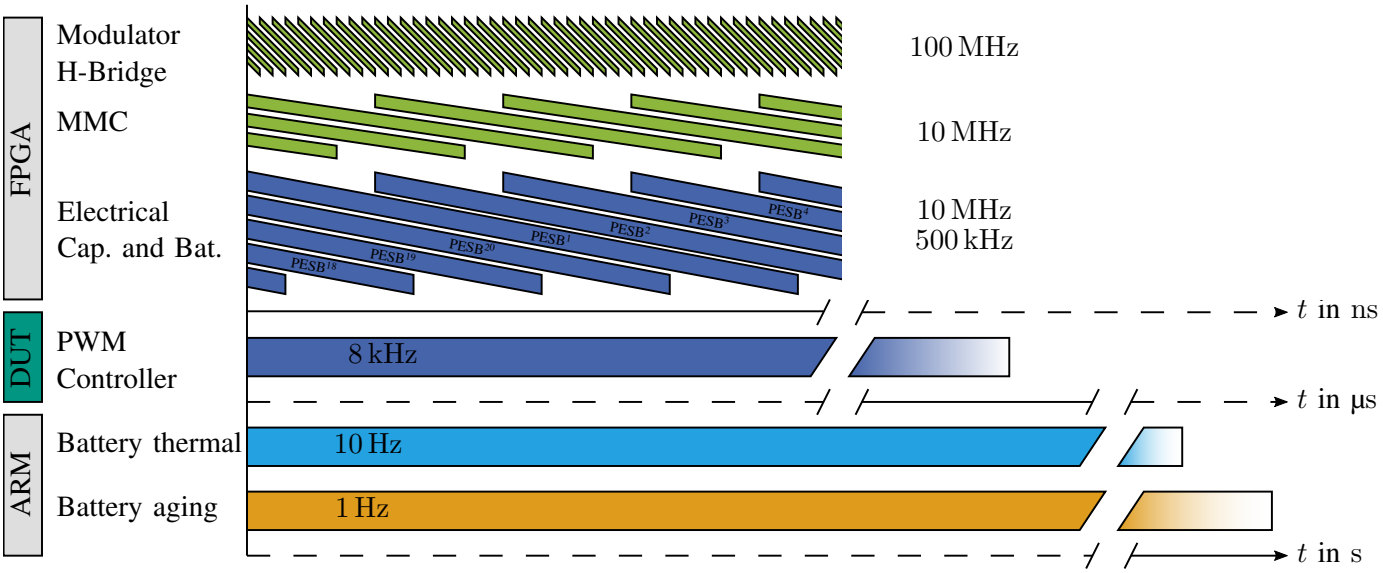


Fig. 6. Overview of Timing Relations between the Model Components.

weighting is achieved with the factors $w_{\mathrm{DoD},i}$, $w_{\mathrm{I},i}$ and $w_{\mathrm{T},i}$ depending on DoD, RMS current and averaged temperature, respectively. For every half or full cycle with the index i, the function as given by

$$Q_{\mathrm{w},i} = n_i \cdot Q_i \cdot w_{\mathrm{I},i}(I_{\mathrm{RMS},i}) \cdot w_{\mathrm{T},i}(\overline{T}_i) \cdot w_{\mathrm{DoD},i}(\mathrm{DoD}_i)\,, \tag{5}$$

where n_i depends on the cycle being a full (n_i = 1) or half cycle (n_i = 0.5) [16]. With increased aging, the battery's capacity decreases and its internal resistance increases. This is represented by feeding the values for $\mathrm{SoH_C}$ and $\mathrm{SoH_R}$ back into the electrical model. Both are functions of $Q_{\mathrm{w},i}$, details can be found in [16].

IV. Control and Energy Management

Despite control and energy management not being the focus of this paper, a brief introduction is given here for comprehensibility of the validation results shown in section VI. The controller is adapted from [13]. Although the MMC considered there does not include batteries, the same method is used to distribute the energy between the six arms of the MMC here. The main idea is to use internal currents within the MMC to influence the distribution of power between the arms while the in- and output power is unaffected.

Additionally, for the distribution of energy within one arm, the method shown by [6] is implemented and adapted. In short, this method works by varying the duty cycles of the individual PESBs while keeping the overall duty cycle per arm as set by the controller. To balance the SoCs, batteries with a high SoC are used more than batteries with a low SoC while power is drawn from the MMC and vice versa.

For the validation data shown later, both controllers are set up to balance variations between SoCs over time. It must be noted, that this is not generally the optimal approach, as it does not consider possible differences in the SoH between the PESBs or other factors. However, for the purpose of validation of the emulator, this controller approach has the advantage of being comprehensible and unexpected behavior can be easily identified.

V. Faster than Real-time

The individual models are running at different clock rates, limited by the available resources. In Fig. 6, an overview of the different time domains is shown. For each model, the clock rate at which it is executed is given. Additionally, the open loop time is represented by the length of the corresponding bar. For example, the open loop time of the MMC is more than double its clock period. While the modulator and H-bridge models are executed at 100 MHz, the MMC model runs at 10 MHz. The more complex battery and capacitor models are limited to 500 kHz, as they are executed in series in groups of 20 models.

To increase the emulation speed above real-time, the clock rates on the FPGA cannot be increased further. Therefore, deterioration of accuracy is accepted to a certain degree. Essentially, the integration steps become larger proportionally to the increase in emulation speed. This can be implemented efficiently by an arithmetic shift, i.e., a multiplication by a power of two. The

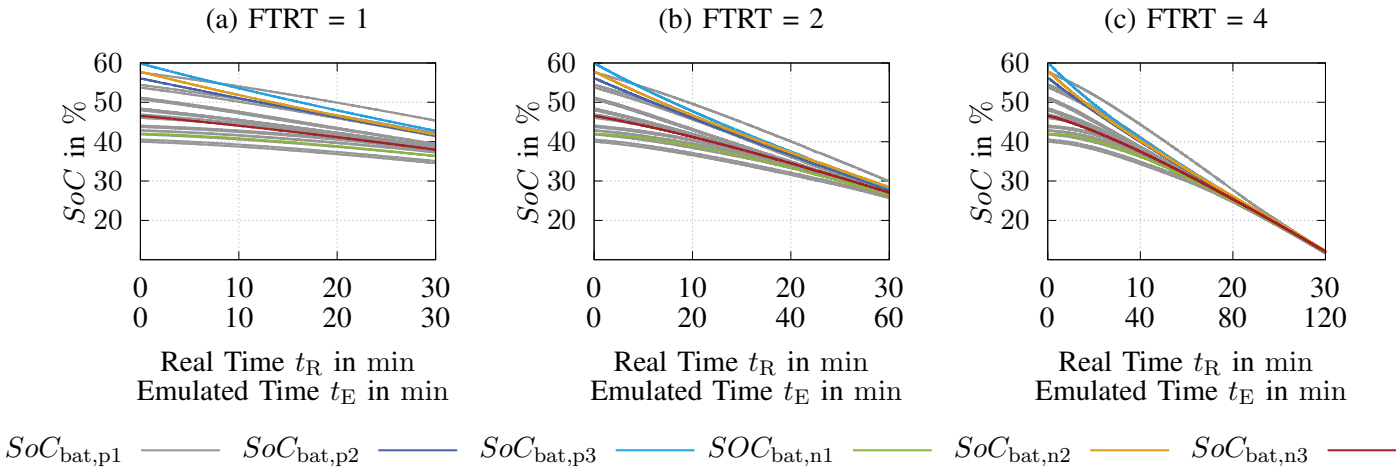


Fig. 7. Demonstration of SoC balancing at different rates faster than real-time.

FPGA code, therefore, does not need to be altered when changing the FTRT factor. The number of bits to shift is received from the processor. This is important as it reduces implementation time and the risk of inconsistent implementations of real-time and FTRT emulations.

The thermal and the aging models do not require execution at fast rates, as they include large time constants. Therefore, they are executed at 10 Hz and 1 Hz, respectively. However, their open loop time is considerably shorter than 1 s. As these models are running on the processor, the clock rates of the thermal and aging models can be proportionally increased, as long as their open loop time is shorter than the clock period.

A. Controller Requirements

The controller is running at 8 kHz. For rates faster than real-time, two possibilities are applicable: The clock rate can be increased, as long as the calculation time of the controller stays below the clock period. If this is no longer possible, the effective clock rate can be reduced. This is possible, as the emulator can use shorter dead times than the real system by using the faster GTX interface. However, it can become necessary to adapt control parameters, especially those determined empirically. With the implementation discussed here, the clock rate can be increased to 16 kHz, above this, a lower effective control frequency must be accepted.

B. Comparison with Simulation

Comparisons between real-time systems and simulations are problematic as their objectives usually differ. However, as the shown approach aims to close the gap between simplified yet fast simulation and more detailed real-time emulation, an estimation is given here. In general, the implementation and specific hardware setup together determine the speed of a model. Although the models in this paper are implemented in Matlab and Matlab Simulink, the models used for the FPGA calculation are not optimized for fast performance on a processor. The FPGA-optimized model parts are therefore re-implemented in a more Matlab/CPU-optimized fashion, e.g., pipeline delays are removed and signals are propagated as vectors, where applicable. Doing so results in the model running approximately six times slower than real-time on an AMD Ryzen 7 7700X 8-Core CPU. From a performance perspective, it is therefore justified to implement them on the real-time platform.

VI. Validation

For validation purposes, the MMC is configured as follows: While all batteries are identical, their initial SoCs differ. Each battery is randomly assigned an initial SoC between 40 % and 60 %. The values are listed in Tab. I. As discussed before, all 20 battery modules within the upper arm of phase 1 (arm p1) are modeled, while for each of the other arms one battery module is modeled, respectively. The initial temperature for the setup is defined to be 25 °C, and the initial SoHs are set to 100 %. The energy management's objective is to balance the SoCs, while an AC power of 100 kW is delivered by the MMC from the batteries. This rather simple scenario without changes in the direction of the power flow is chosen for comprehensibility. In section VI-D, this chapter is complemented with a realistic scenario including changes in the direction of the power flow.

A. Balancing SoC and FTRT

In Fig. 7, the SoCs of the battery modules over time are shown. For arm p1, the SoCs of all 20 modules are shown in grey, and the voltages of the averaged modules

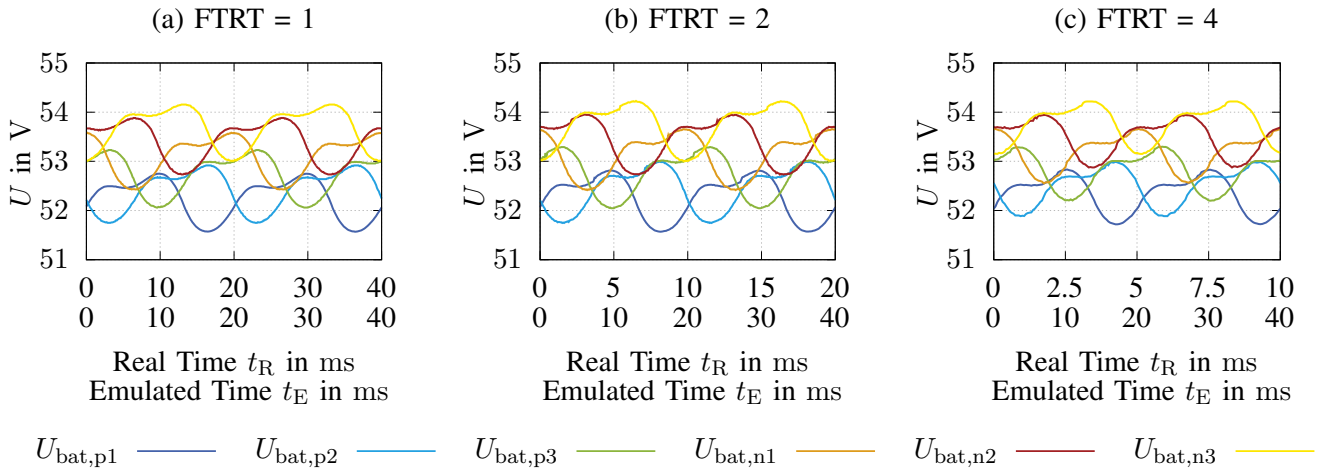


Fig. 8. Demonstration of battery voltages at different rates faster than real-time.

TABLE I
INITIAL SOCS

$p_{1,01}$: 48 %	$p_{1,02}$: 54 %	$p_{1,03}$: 40 %	$p_{1,04}$: 46 %
$p_{1,05}$: 43 %	$p_{1,06}$: 42 %	$p_{1,07}$: 44 %	$p_{1,08}$: 47 %
$p_{1,09}$: 48 %	$p_{1,10}$: 51 %	$p_{1,11}$: 48 %	$p_{1,12}$: 54 %
$p_{1,13}$: 44 %	$p_{1,14}$: 58 %	$p_{1,15}$: 41 %	$p_{1,16}$: 53 %
$p_{1,17}$: 48 %	$p_{1,18}$: 51 %	$p_{1,19}$: 43 %	$p_{1,20}$: 44 %
p_2: 56 %	p_3: 60 %		
n_1: 42 %	n_2: 58 %	n_3: 47 %	

in the remaining arms are shown in colors according to the key. For Fig. 7a, the model is run in real-time. In Fig. 7b, the execution speed is doubled, resulting in 60 min emulated time only requiring 30 min in real-time. As expected, the SoC after 30 min emulated time are identical in both cases. In Fig. 7c, the execution speed is quadrupled compared to real-time. It must be noted, that the behavior of the SoC is slightly different from the behavior in real-time and double real-time. The reason is, that the controller can only run at double of its designed frequency. It is therefore adapted for the effectively lower frequency, resulting in slight differences.

B. Interaction between battery and MMC model

In Fig. 8 the average voltages of the batteries in each arm are shown. They are not to be confused with the arm voltages. As before, three diagrams a)-c) are shown for real-time, double real-time, and quadruple real-time, respectively. However, in these plots always the same emulated time is shown. From this, several things become clear: The general behavior is identical in all three cases. The time step is in all cases small enough

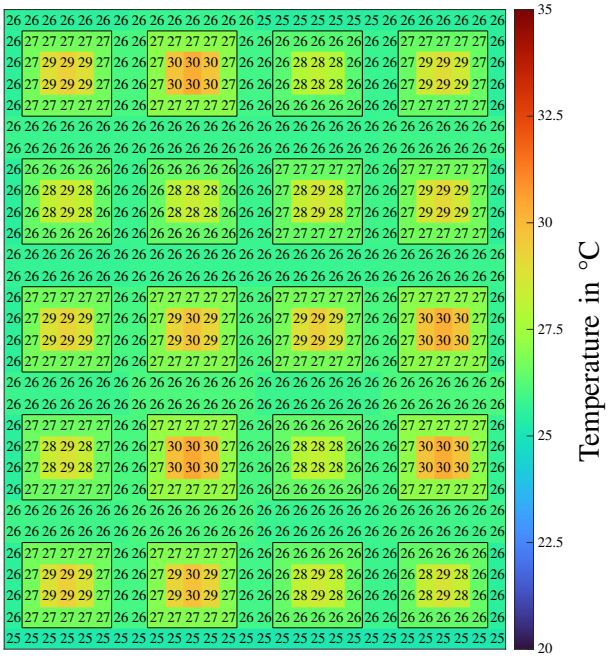


Fig. 9. Temperatures in the switch cabinet after 1 h.

to not result in visible artifacts. At PWM frequency, this can become an issue. Nevertheless, this can be bypassed by neglecting switching events and averaging the current as discussed before. Slight variations between the voltages in the three diagrams are mostly caused by them not being captured at the exact same emulated point in time.

C. Thermal behavior

In Fig. 9, the temperatures of the 20 battery modules in the switch cabinet of arm p1 after 1 hour of running in real-time are visualized. The batteries in this cabinet are arranged in the same way the initial SoCs are given in the

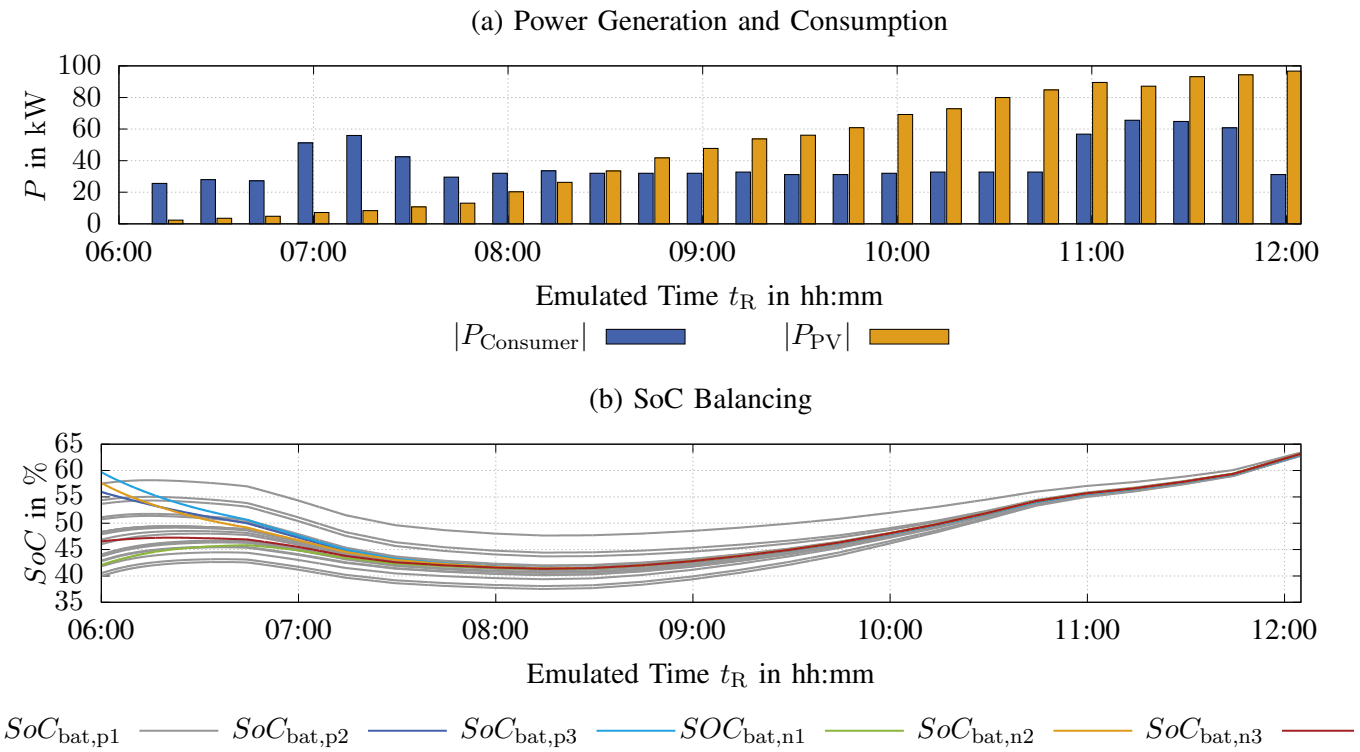


Fig. 10. Application scenario for an industry plant.

upper part of Tab. I. It can be expected, that the battery modules with a higher initial SoC are discharged faster than battery modules with a lower SoC as this results in the SoC approaching each other during discharging. From Fig. 9 it is evident that the temperature of the modules with a higher initial SoC is higher compared to other modules due to a higher average current.

D. Application scenario

The FTRT emulation is employed with a realistic scenario of the battery integrated MMC. In Fig. 10a, the electric power generation and consumption of a medium-sized building equipped with a rooftop photovoltaic system located at an industrial plant are shown. The data was captured on a sunny day in April 2022.

The emulation executed four times faster than real-time over 6 hours of emulated time results in the SoCs shown in Fig. 10b. Between 6:00 a.m. and 8:15 a.m., more power is consumed than provided by the photovoltaic system. Therefore, power is drawn from the BESS. However, due to the different initial SoC emulated in this scenario, not all batteries are discharging. The batteries in the arms p1, n1, and n3 start with SoCs below the average of the BESS. The controller, therefore, supplies internal currents to charge those batteries. After 6:45 a.m., the variance between SoCs has decreased, so that charging is no longer employed by the controller. Balancing is now achieved by different discharging rates.

At 8:30 a.m., the amount of photovoltaic power generated surpasses the power required by the consumer. From this moment, the BESS is charging. The balancing continues so that at 12:00 p.m., all SoC are approximately equal. In this example, the balancing within one arm is implemented slower than between the arms.

Based on this test, improvements for the algorithm can be derived. In this example, it can be concluded that the behavior of the algorithm in the morning can be improved. The mixed charging and discharging in the system lead to additional losses and cyclic battery aging. As the balancing is fast enough to balance in the course of one day anyways, measures should be taken to inhibit this early fast balancing. Further improvements can be derived but are not the focus of this paper.

In comparison to real-time HiL, this test was executed within a quarter of the time. In comparison to a real test, the time saved is even greater, as the batteries would first have to be charged or discharged to the initial SoC required.

VII. Conclusion

An approach to model an MMC with integrated batteries on a real-time platform utilizing both processor and FPGA is shown. Electrical, thermal and aging models are implemented in such a way that they are scalable to the requirement of multiple battery modules within the MMC. Where compromises in precision are acceptable, the models can be executed at rates faster than real-time. This opens up new possibilities for the development of energy management strategies. In comparison to simulations, the advantages are given by smaller step sizes resulting in the possibility to model dynamic effects and realistic interfaces thus simplifying transfer to the real system. In comparison to real-time simulations, this approach allows the evaluation of thermal and aging effects with reduced required runtimes. The shown approach is intended for the testing of energy management and lifetime optimization algorithms as well as the simplified deployment of such for a real system.

References

[1] R. Marquardt, A. Lesnicar, and J. Hildinger, "Modulares Stromrichterkonzept für Netzkupplungsanwendungen bei hohen Spannungen," in *ETG-Fachtagung*, Bad Nauheim, Germany, 2002.

[2] M. Schroeder, S. Henninger, J. Jaeger, A. Raš, H. Rubenbauer, and H. Leu, "Integration of batteries into a modular multilevel converter," in *2013 15th European Conference on Power Electronics and Applications (EPE)*, Sep. 2013, pp. 1–12.

[3] A. Hillers, M. Stojadinovic, and J. Biela, "Systematic comparison of modular multilevel converter topologies for battery energy storage systems based on split batteries," in *2015 17th European Conference on Power Electronics and Applications (EPE'15 ECCE-Europe)*, Sep. 2015, pp. 1–9.

[4] M. Gommeringer, F. Kammerer, J. Kolb, and M. Braun, "Novel DC-AC Converter Topology for Multilevel Battery Energy Storage Systems," *PCIM Europe : International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, Nuremberg, 14 - 16 May 2013 ; proceedings*, p. 699, 2013.

[5] L. Leister, A. Kalk, B. Schmitz-Rode, L. Stefanski, D. Bräckle, and M. Hiller, "Hardware-in-the-Loop Setup for a Modular Multilevel Converter with Integrated Batteries," in *2022 IEEE 23rd Workshop on Control and Modeling for Power Electronics (COMPEL)*, Jun. 2022, pp. 1–7.

[6] F. Gao, L. Zhang, Q. Zhou, M. Chen, T. Xu, and S. Hu, "State-of-charge balancing control strategy of battery energy storage system based on modular multilevel converter," in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2014, pp. 2567–2574.

[7] N. Li, F. Gao, T. Hao, Z. Ma, and C. Zhang, "SOH Balancing Control Method for the MMC Battery Energy Storage System," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6581–6591, Aug. 2018.

[8] X. Liu, J. Ospina, I. Zografopoulos, A. Russel, and C. Konstantinou, "Faster than real-time simulation: Methods, tools, and applications," in *Proceedings of the 9th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, ser. MSCPES '21. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 1–7.

[9] S. Mazumder and J. Lu, "Faster-Than-Real-Time Simulation of Lithium Ion Batteries with Full Spatial and Temporal Resolution," *International Journal of Electrochemistry*, vol. 2013, pp. 1–10, 2013.

[10] F. Lacressonnière, A. Varais, and X. Roboam, "Scaling electro-thermal model of a lithium-ion battery for time-accelerated experiments in a HIL system," *Symposium de Génie Electrique, Université de Lorraine [UL]*, Jul. 2018.

[11] C. Wang, W. Li, and J. Belanger, "Real-time and faster-than-real-time simulation of Modular Multilevel Converters using standard multi-core CPU and FPGA chips," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2013, pp. 5405–5411.

[12] B. Schmitz-Rode, L. Stefanski, R. Schwendemann, S. Decker, S. Mersche, P. Kiehnle, P. Himmelmann, A. Liske, and M. Hiller, "A modular signal processing platform for grid and motor control, HIL and PHIL applications," in *2022 International Conference on Power Electronics and ECCE Asia (IPEC-Himeji 2022 -ECCE Asia)*, 2022.

[13] D. Braeckle, P. Himmelmann, L. Groll, V. Hagenmeyer, and M. Hiller, "Energy Pulsation Reduction in Modular Multilevel Converters Using Optimized Current Trajectories," *IEEE Open Journal of Power Electronics*, vol. 2, pp. 171–186, 2021.

[14] A. Samanta and S. Williamson, "A Comprehensive Review of Lithium-Ion Cell Temperature Estimation Techniques Applicable to Health-Conscious Fast Charging and Smart Battery Management Systems," *Energies*, vol. 14, p. 5960, Sep. 2021.

[15] Y. Shi, K. Smith, E. Wood, and A. Pesaran, "A multi-node thermal system model for lithium-ion battery packs," in *2015 American Control Conference (ACC)*, Jul. 2015, pp. 723–727.

[16] T. Grün, "Passiv gekoppelte Lithium-Ionen-Superkondensatorsysteme für 48V Hybridfahrzeuge," https://publikationen.bibliothek.kit.edu/1000127761, 2020.

[17] S. Loew, A. Anand, and A. Szabo, "Economic model predictive control of Li-ion battery cyclic aging via online rainflow-analysis," *Energy Storage*, vol. 3, no. 3, p. e228, 2021.

[18] G. Marsh, C. Wignall, P. R. Thies, N. Barltrop, A. Incecik, V. Venugopal, and L. Johanning, "Review and application of Rainflow residue processing techniques for accurate fatigue damage estimation," *International Journal of Fatigue*, vol. 82, pp. 757–765, Jan. 2016.