# Power Hardware in the Loop (PHIL) Simulation of Battery Packs

Maitri Uppaluri

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Chemical Engineering
University of Washington
2019

Committee:
Venkat R. Subramanian
Daniel T. Schwartz

Program Authorized to Offer Degree:
Chemical Engineering

University of Washington

## Abstract

Power Hardware in the Loop (PHIL) Simulation of Battery Packs

Maitri Uppaluri

Chair of the Supervisory Committee:
Venkat Subramanian
Department of Chemical Engineering

Power-Hardware in the loop (PHIL) simulation is a powerful technique that uses a real-time simulator to compute a model of a given system in real-time to send power signals from the model to a physical load connected to the simulator. To improve battery designs and battery management systems, PHIL simulations can be used to test their response under dynamic load conditions. The response from the load in connection to the battery model can be sent back to the model as feedback. It provides a novel approach to designing and testing new configurations of batteries from data of a single cell without building their real prototypes.

An equivalent circuit model and a physics-based model of batteries were developed in SIMULINK and tested for PHIL implementation using the OPAL-RT. The single particle model was chosen for its simplicity and to investigate the internal states of the battery with a real load with PHIL simulations.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisors, Prof. Venkat Subramanian and Prof. Daniel T. Schwartz, who have supported me throughout the course of my time working on this project for my Thesis. I would like to thank the staff at the Washington Clean Energy Testbeds who helped me understand and use the OPAL-RT real-time simulator. I would also like to acknowledge the group members from the Subramanian and the Schwartz group, who helped me grow my knowledge in the field and provided a source of motivation. Finally, I would like to thank my friends and family for encouraging me and being my backbone throughout the course of pursuing my master's at the University of Washington.

# CHAPTER 1 INTRODUCTION

There has been a growing demand for innovation in the development of electrochemical energy storage systems such as batteries and supercapacitors for grid energy storage and for powering electric/hybrid vehicles. The usage of these devices for direct power supply from the source to the loads can improve the efficiency of providing power to the user with reducing cost, space and any environmental concern associated with conventional methods of producing electricity [2]. Lithium-ion batteries are an example of electrochemical energy storage devices that have shown promising energy and power capabilities, proving their potential to be used for a wide variety of applications [3].

However, the effort to use battery energy storage systems can result in limited load following capabilities. Power output shortfall can result in damage of these systems and other subsequent components. When cycled over a period, the aging of these systems can result in poor performance resulting in a loss of efficiency of the device. It is important to improve the design of such electrochemical storage devices and test them under extreme conditions to overcome these shortfalls and meet the required load demand without any significant delay.

Battery management systems are used to monitor battery packs to ensure their safety and optimal operation. Different control strategies exist to achieve optimal operation and prevent capacity fade in each cell in the overall pack.

While testing the performance of different battery designs from different battery manufacturers, it is important to address issues that could occur when subjected to different types of power loads. These issues can be observed from the system level to the cell level of these batteries [3]. Understanding the internal behavior of the battery can help comprehend how a specific design of a battery will behave under a range of test conditions. Battery models can be used to predict the performance of different designs when subjected to a load. Several battery models are present in literature that vary in complexity and computational time. There are models that are formulated empirically or with equivalent circuits which treat the battery like a black box and provide information about the final output variables such as voltage, current and temperature [4]. In order to understand the internal dynamics that influence the performance of the battery, physics-based models are used to capture information on the internal states that are determined by its kinetic and transport parameters such as concentration, solid and liquid based potential, etc.

Hardware in the loop (HIL) simulation is a novel technique used to test systems and components in real-time. A mathematical model of the system is inputted in a host computer that can send signals based on the desired output of the model to physical load to be powered. By using a real-time simulator, the computational speed required to solve the model with small time steps (as small as 10μs [1]) can be achieved. The small-time step provides the advantage of observing transient responses to dynamic power profiles from the load to be tested. Transient responses can be useful to analyze the response when subjected to a sudden change in the power demand. This information can help design batteries that can overcome extreme conditions and find deficits in existing designs due to output shortfalls when provided a dynamic load. It is also necessary to design configurations of different such devices to accommodate for these shortfalls. Different designs can be incorporated into the model and tested for their performance using this technique without the need to build and develop a prototype for testing.

HIL can have a powerful impact on the future of battery testing for several applications. It can be used for the testing the accuracy of battery algorithms [5,6]. It is also useful for conducting repeatability experiments, to prevent the aging of battery during testing [7]. HIL simulation for batteries can be used for testing faults in designs without the concern of the safety behind the testing of such devices and provides flexibility for testing designs under extreme conditions [8]. HIL can also be used to validate online parameter estimation algorithms in real-time [9,10,11]. Testing battery management systems (BMS) for batteries can also be done with HIL to study the performance of these batteries when assembled in a pack for different loads [6]. Real batteries can be tested using HIL for their capability to run and store the power generated a grid [12] or a fuel cell [13].

For testing systems with high power demands, an extension of the technique is used, known as Power-Hardware in the loop (PHIL). Here, a power amplifier is put into the loop between the model and the load. Data signals that would normally be transmitted in a HIL simulation are amplified to real power signals in power HIL. PHIL can further improve the understanding of the effect of battery design under extreme conditions and dynamic loads, as real voltage and current is provided to the load based on the calculations from the model.

This thesis focuses on the development of battery models for running PHIL simulations. Real-time simulation of battery models is required for conducting PHIL simulations. A real-time simulator is used to send the response from the model to the power supply to emulate a battery connected to a load. The real-time simulator uses SIMULINK to compile and run the model for PHIL simulations. The equivalent circuit model and the single particle model were explored and built in the SIMULINK environment. The single particle model was implemented to understand the internal states of the battery and how they are affected when subjected to different loads and tests.

In this thesis, Chapter 2 describes the literature available for implementing HIL simulations for batteries. Chapter 3 talks about the different battery models used in this thesis that were developed in SIMULINK. Chapter 4 describes the methods used to develop the models and run a PHIL simulation with the OPAL-RT. Chapter 5 discusses the results obtained and Chapter 6 provides a conclusion and the future work that can be carried out for this work.

# CHAPTER 2 HIL SIMULATIONS OF BATTERIES IN LITERATURE

## 2.1 Models

Typically for HIL simulations of batteries, the battery models are chosen to have less complexity in order to achieve real-time computation. When a model is chosen for hardware in the loop simulations, a trade-off between accuracy and computational speed needs to be considered. Equivalent circuit models (ECM) are widely used as they can be easily parameterized and computed in real-time [7]. The parameters of the circuit normally consist of a voltage source, an ohmic resistance and some number of RC elements. The voltage source normally describes the open circuit voltage as a function of the state of charge of the battery. The RC elements describe the phenomena that occur in the battery such as charge transfer and diffusion. The parameters can be obtained from electrochemical impedance spectroscopic (EIS) measurements of the battery [8]. ECMs give information about the voltage and current of the battery operating at a predefined cycle and temperature.

There have been studies which have incorporated temperature and aging effects to improve the complexity of the model. Barreras et al. presented an electrothermal model ECM of a lithium-ion battery pack coupled with aging effects. The aging parameters for the ECM were extracted separately and needed to be changed during the HIL simulation [7].

Seitl et al. used an extended modified shepherd model (eMSM) for HIL simulations. The model can be compared to an ECM and be described with an open circuit voltage, an internal resistor and one RC network where the time constant holds a constant value. An eMSM model with the initial SOC as the additional parameter was proposed. The parameters of the model were found by taking specific measurements from the battery such as open circuit voltage curves, discharge pulses or a constant discharge measurement to create a lookup table of the parameters [1,14].

To account for the nonlinear dynamics seen in a real battery, Konig et al. proposed a local model network model (LMN) of a battery which consisted of different local models for each region for state space. The dimensions that span each space are the temperature, the state of charge and the load current. An LMN model uses the current history and the present value of current as the input, and the estimated terminal voltage as the output to include the non-linear dynamics [15].

The host computer houses the model of the plant to be tested. For most cases, the model is built in the MATLAB/SIMULINK environment in order to build and load models into a real-time simulator for HIL simulations. The model is built in such a way it is compatible for code generation, and there are several tools used to enable code generation such as real-time workshop or the RT-LAB software.

## 2.2 Targets

The host is connected to a target which behaves as the interface between the host and the device under test. These targets carry out the real-time computation of the model from the host computer. A real-time target comprises of I/O channels and multiple CPUs that can run. There are several targets available for HIL testing: dSPACE, OPAL-RT, xPC Target, etc.

Battery emulators are used for cell level hardware in the loop simulations, more commonly for testing battery management systems. For cell module emulation the dSPACE EV1077 board is commonly used. It has four channels to simulate four cells independent of each other. They have a relatively wide voltage range for testing overvoltage conditions. These boards can be connected in series and parallel to emulate a pack. Another product from dSPACE used for HIL simulations is the DS1104 which can obtain current, voltage and temperature data as well as control the charging and discharging system.

OPAL-RT also has a real-time simulator which utilizes a software (RT-LAB) that interfaces between the real-time simulator and the developed MATLAB/SIMULINK model. One can provide 32 analog inputs and outputs to the real-time simulator.

The xPC target is a product that executes a SIMULINK model in real-time which can automatically generate code from MathWorks's Real-Time Workshop for real-time implementation.

## 2.3 Loads/DUTs

The device under test (DUT) is the real device that is powered by the simulation in the loop. For testing battery performance, the DUT can either be a real battery pack or cell connected to the model of a microgrid or a parameter estimation algorithm, or the DUT can be a load, a BMS or a controller connected to the battery model. When a load is the DUT, HIL simulations are accompanied by a power supply to amplify the voltage and current signals. This is known as Power-Hardware in the loop (PHIL) simulations.

## 2.4 Applications of HIL simulations for understanding battery performance

Hardware in the loop simulation for testing batteries can be broadly categorized into two types of tests; one being the testing of models of the grid where an actual battery is connected to the model of a power grid and tested (Battery in the loop), and the other being where the model of the battery is used, with physical controllers or a load are added to the loop for testing [7]. Both types can be used for different applications. Battery in the loop is used for studying grid models and for automotive components testing. On the other hand, HIL experiments with the battery as the model are used for monitoring pack voltage, pack current, cell balancing. They can also be used for validating the design of the battery by testing the battery model in HIL under different load conditions. It can also be used for testing new control strategies.

For testing battery performance with battery models, HIL has been used in several different ways such as

a) Using battery models with a battery management system in loop with the model to validate the proposed control strategies and functionality of the battery management system.
b) Verifying a proposed battery model or algorithm by testing the model with the specific load conditions and testing a physical battery in loop with the load.
c) Testing parameter estimation algorithms with the algorithm in loop with a physical battery.

Studies that have been carried out for each of the above methods are described below.

### 2.4.1 Testing Battery Management Systems

Battery management systems are important to monitor and control the conditions the battery is charging or discharging under. They are important to monitor the voltages, current and states of each individual cell to ensure cell balancing in a pack. HIL is useful for testing BMS functionality under extreme conditions the battery may face such as low temperatures or for simulating any type of fault in a cell of the pack. Figure 2.1 shows how HIL can be used for testing battery management systems or specific control strategies for the BMS.

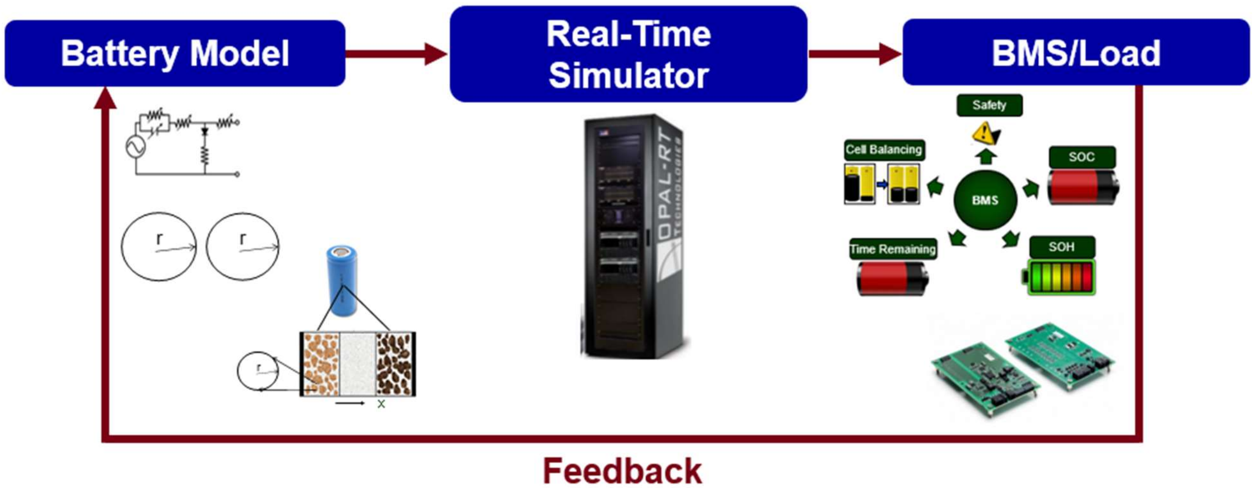


Figure 2.1 HIL implementation for testing battery management systems.

Dai et al. developed an equivalent circuit model for hardware in the loop implementation. HIL testing was conducted to validate the battery management system for voltage detection. The HIL system comprised of host computers which are connected to cell emulators, with a USB-CAN connection. While running the HIL, the model in the host PC first identifies the parameters offline using the parameter identification algorithm, and then sends these parameters to the emulators. Parameters are set constant for the HIL test. For different battery conditions, the model was reconfigured and tested. The emulated cells were connected in series and parallel and could validate the functions of active and passive cell balancing [8].

Haupt et al. tested a battery management system's electronic control unit. The model used a 2 RC network circuit, with the elements representing charge transfer and diffusion effects. The parameters were determined with impedance spectroscopy measurements and the open circuit voltage is a function of the state of charge of the cell. A new approach to model the cell network was proposed, where the model consisted of a reference model and a delta model. The former described the fundamental behavior of the cell, and the latter calculated the difference between the voltage of a cell and the voltage from the reference model. dSPACE cell emulators were used to emulate the model of the battery. The setup consisted of an expansion box which controlled the outputs and simulated the models in real-time. This study was done to observe active and passive cell balancing of the pack [16].

Incorporating aging effects for HIL simulation helps understand the effects of an aged battery on a load. This deems useful for the prediction of when the battery will fail under specific load conditions. Barreras et al. conducted a HIL simulation with a battery model that included thermal and aging effects with parameter estimation done by electrochemical impedance spectroscopy measurement, anisotropy thermal diffusivity measurements, experimental pulse power characterization and capacity checks. To account for aging effects while running the HIL simulation, parameters corresponding to a cycle test done earlier were extracted and changed offline. The battery model also included models of other components such as temperature sensors, fuse and switch boxes. Voltage emulations were done by dSPACE EV1077 battery cell voltage emulation boards. First, the model was validated with a dynamic load profile to monitor the measured voltage and temperature. Then, the BMS was tested for low-level functionality and fault testing [7].

Battery management system testing is crucial to test the components and the overall design before a prototype is developed, to test the design's performance under extreme conditions such as overcharge/over discharge of cells, the occurrence of different voltages in cells in the module or pack, extreme temperatures, etc. without compromising on safety of these tests. New components of the battery management system can be designed and tested using HIL simulations. Battery management systems can be tested for specific applications such as designing hybrid electric vehicles or electric vehicles that use battery packs.

### 2.4.2 Verification of Battery Models

HIL simulations can be used to validate new battery models, by comparing the results of simulation with a dynamic load and HIL tests with a physical battery in loop.

He et al. used a cell level hardware in the loop experiment to develop state of charge and power predictions to verify battery algorithms. Profiles from vehicles are provided to the HIL controller to the cycler that simulates the vehicle energy management system. The measured voltages, current and temperatures are measured and provided to the HIL controller as input, and SOC and power over time are predicted. The model was validated by comparing the model outputs, to the HIL controller outputs [5].

Marco et al. conducted cell in the loop HIL simulations. The cell model was an equivalent circuit model with one RC network and each parameter as a function of state of charge. The parameters of the cell were estimated using the Hybrid Pulse Power Characterization (HPPC) standard. dSPACE real simulation platform was used to for HIL implementation. The setup consisted of two cells in series: one cell being the simulation model and the other being a physical cell. The voltage profiles from the simulated cell and the physical cell were almost identical and comparable to the results obtained when each cell was run offline. Feedback was used to control the load profile [6].

Bazargan et al. compared a pure simulation and a HIL simulation of wind generation scheme. Power HIL simulations were conducted to exchange power signals between the physical battery and the simulation via a power amplifier. RTDS hardware is used as the target to simulate a wind generator for a wind profile. The model used for comparison with the HIL simulation was the shepherd model. Two models describing lead acid and Li-ion batteries respectively were studied. The state of charge of the battery model and the real battery were compared. For both types of batteries, the SOC of the actual battery was lesser than that of the model indicating that ~~the~~ average voltage of the battery was less than the model [12].

Verification of battery models using HIL simulations is useful to compare the performance of a real battery in loop with the model of the load with the predictions of state of charge from the model.

### 2.4.3 Testing Parameter Estimation Algorithms

HIL testing can be used to test new parameter estimation algorithms for battery models. Unlike the testing of equivalent circuit models where the parameters of the circuit are evaluated offline before the HIL simulation is run, estimation of quantities such as state of charge, state of energy, etc. are tested based on the measurements from a real battery.

Chen et al. validated an extended Kalman filtering algorithm with HIL simulations. A real-time controller from dSPACE was used with a programmable power supply as the charger and a programmable electronic load to discharge. Different driving cycles were simulated to test the battery model and the state of charge of the battery was estimated. Several experiments were conducted to carry out SOC estimation, which included estimation with a known and unknown initial SOC, estimation with added noise and SOC estimation with different battery parameters. Different estimation algorithms were compared to the proposed algorithm, and the proposed algorithm provided a more accurate estimation [10].

Zhang et al. proposed an adaptive H filtering method for SOC estimation for predicting parameters. HIL experiments were used to validate the algorithm for its accuracy and robustness. The xPC target was used as the real-time simulator which was connected to an electrical load and a battery pack consisting of two cells in series. The host computer would send current signals to the load which would discharge the battery and the real-_time data of the battery voltage, current and temperature was collected and sent back to the host PC which hosted the algorithm for SOC estimation. The proposed algorithm was compared to existing algorithms for validation [11].

Xiong et al. used HIL simulations to develop a method of continuous peak power capability estimation that is more accurate than the proposed hybrid pulse power characterization method. An electrochemical polarization model was used, where the OCV-SOC relationship is defined by the Nernst model. The model consists of an ohmic resistor, and an RC element that describes polarization characteristics. The HIL test bench consisted of electrical load equipment which could charge and discharge the battery and the host computer which contained the estimation model. Since the described method can estimate power based on a continuous current input, HIL tests were useful for estimation when there is a sudden change in current [17],[18].

### 2.4.4 Battery in the loop simulations

HIL simulation is useful to test a physical battery in loop with a model of a load, grid or another system. These tests are done to evaluate the design of the grid or component with the model of the same.

Gauchia et al. conducted a HIL simulation of a hybrid fuel cell/battery vehicle, with the model of the fuel cell in the host computer, the driving profile in a programmable load and a physical battery in the loop [13].

Khazaei et al. carried out real-time simulation of a microgrid consisting of a PV array and a wind farm with a real battery as the hardware component in the HIL simulation. This approach was used to test a control strategy for the battery, for the implementation of the battery with the PV grid [19].

## 2.5 Implementing Power-HIL simulations for battery models

PHIL simulations of batteries are used to evaluate the power the battery pack can supply based on the power requirements of the load. The signals in a regular loop are amplified with an additional component in the loop. Figure 2.2 demonstrates the workflow for a PHIL simulation for testing batteries.

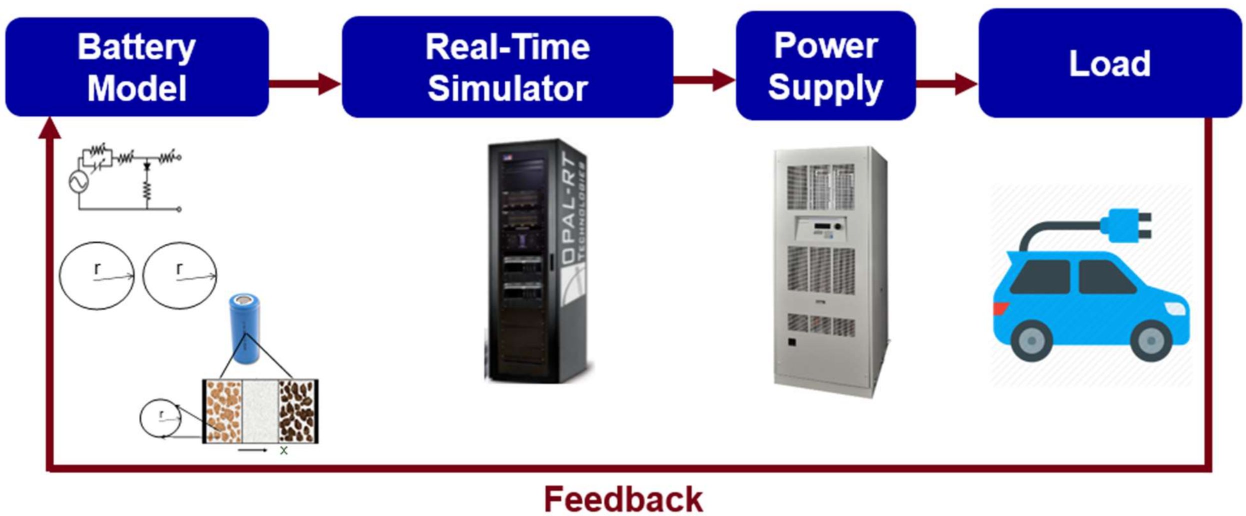


Figure 2.2 PHIL implementation for testing their performance under real loads

Konig et al. used power hardware the loop simulation to test the powertrain of an electric vehicle system with a battery emulator which was run by a local model network model (LMN) of a battery described in the previous section. An impedance controller was used to tightly couple the hardware under test and the battery hardware that emulated the model. A DC power supply coupled with a digital controller was used as the battery emulation hardware. Stability analysis of the closed loop system was conducted [15]. The model was tested against an EV drive system in the loop. They concluded that using an MPC-based approach, they were able to achieve faster closed loop response.

Power Hardware in the loop simulation can also be used for validating battery models. Song et al. used power HIL to verify their battery model when the model is subjected to load data from current requirements from an electric vehicle. The model used was constructed from EIS data of a battery that consisted of resistors, capacitors and inductors that described the dynamic properties of the battery such as ohmic, charge transfer and diffusion losses. An actual battery system with charger/discharger was placed in loop with the model of the driving profile to assess the input and output of the model. The model was included in the model of the plug-in hybrid electric vehicles and this model was tested with PHIL. The voltage estimated from the battery algorithm was used to calculate the power. These power signals were sent to the charger/discharger and the voltage response from the physical battery pack was compared to the same from the model. An error rate of 2% was observed [20].

Seitl et al. used PHIL for battery inverter testing, using an equivalent circuit model as the battery model for the simulation. The model was implemented in MATLAB/SIMULINK. Opal-RT was used as the real-time simulator, with 10µs as the time step to increase the accuracy. For power HIL, Regatron TC.GSS.32.600.400400 was used as the power amplifier. A programmable current source with a constant load was incorporated in the loop, controlled by LabVIEW. The current out of the power amplification was measured and fed back into the simulation to close the loop. The set value used for the experiments were according to the FTP-72 cycle, and the resulting current of the battery simulator was analyzed. Feedback is done to increase the stability of the system. The error of voltage between the offline simulation of the battery model and the PHIL simulation obtained was 0.16% RMS [1,14].

# CHAPTER 3 BATTERY MODELS FOR PHIL SIMULATIONS

In literature, the model used to describe a battery is an equivalent circuit model. These types of models are widely popular due to their reduced complexity which results in faster computational speeds to solve these models making them suitable for real-time implementation [21]. An equivalent circuit model that describes a battery is shown in Figure 3.1. The model normally consists of a voltage source, defined by the open circuit voltage of a cell as a function of the state of charge; one resistor that emulates the internal resistance of the cell; three parallel RC (resistors and constant phase elements) networks, one describing the solid electrolyte interface (SEI) layer on the anode of the cell, one which describes the kinetics of the cell and the other describing diffusion dynamics of the cell. One can assume the constant phase elements in the model as capacitors for determining the overall voltage across the circuit to determine the voltage across a cell.

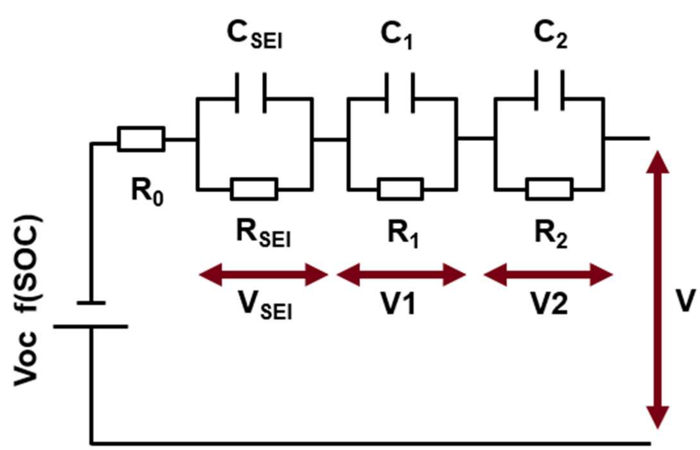


Figure 3.1 Equivalent circuit model of a cell with 3 RC network elements.

To emulate a battery pack, the voltage across the each of the circuit elements in series is multiplied with the number of cells in series; and the current passing through the parallel elements is multiplied by the number of cells in parallel.

The parameters of the equivalent circuit can be obtained from a Nyquist plot, which is obtained by performing Electrochemical Impedance Spectroscopy (EIS) on the cell. EIS tests can give insight into the internal dynamics of the cell. The impedance data obtained from the EIS measurements can be fit into an equivalent circuit whose parameters are described by a Nyquist plot. The impedance data of the cell is collected across different frequencies. The real component of the impedance is plotted against the imaginary component to obtain the Nyquist plot. Figure 3.2 shows the plot and the corresponding equivalent circuit that describes the impedance data and the physical information that can be obtained from the plot about the cell.

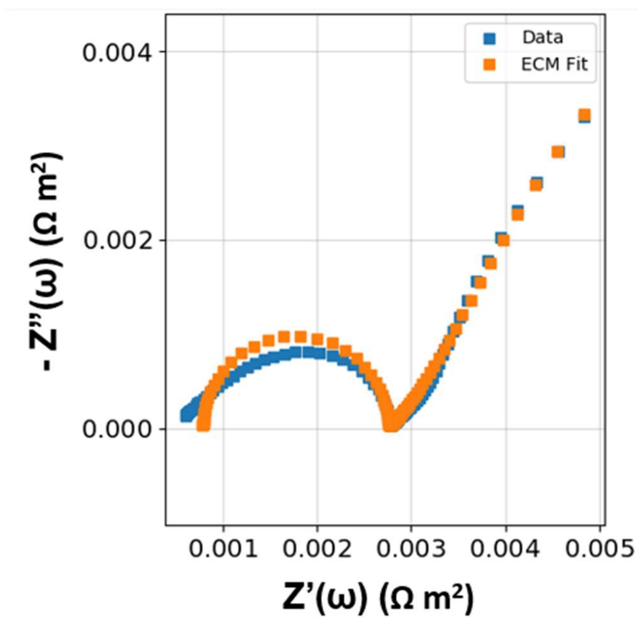


Figure 3.2 Nyquist plot of a battery obtained from EIS data

The Nyquist plot can be fitted using a circuit that uses one parallel RC network and one Randle's circuit, as shown in Figure 3.3.

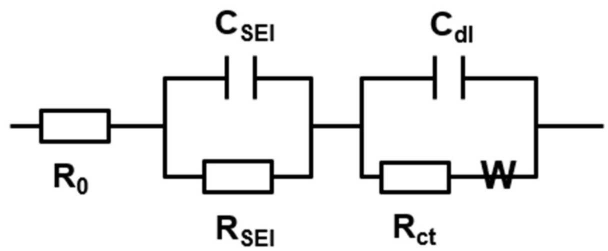


Figure 3.3 Equivalent Circuit to fit the EIS data.

R0 represents the internal resistance of the cell that is observed at high frequencies. The charge transfer resistance and double player capacitance are represented by the semicircle in the plot, and the diffusion dynamics is represented by the tail. The kinetics of the cell is captured by the parallel RC network while the diffusion is represented by the Randle's circuit. To translate the fitted parameters obtained from impedance of the cell to an equivalent circuit model, the Warburg element in the Randle's circuit is approximated to be a series of infinite parallel RC elements [22]. The capacitance in the Randle's circuit is neglected. This results in the circuit and model shown in Figure 3.1.

The voltage output of the cell can be determined by calculating the voltage across each RC network. The voltage across each RC network can be defined as

$$\frac{dVn}{dt} = \left(-\frac{1}{Rn*Cn}\right)Vn + \frac{1}{Cn}I \qquad 3.1$$

Where n is the corresponding RC network, Rn is the resistance, Cn is the capacitance, Vn is the voltage across the RC network and I is the input current to the cell.

The terminal voltage across the model is described as

$$V = Voc + IR0 + \sum_1^n Vn \qquad 3.2$$

Where Voc is the open circuit voltage of the cell.

The open circuit voltage is a function of the state of charge of the battery. The state of charge of a battery at any given time can be described by coulomb counting [21].

$$SOC = SOC_0 + \frac{1}{Q}\int_0^t I(t)\, dt \qquad 3.3$$

Where $SOC_0$ is the initial state of charge of the cell, Q is the nominal capacity of the cell and I is the input current to the cell.

To gain insight into the internal states of the cell, a physics-based model can be incorporated into the power hardware in the loop simulation. The single particle model solves for the diffusion and intercalation in a single electrode particle. This can be extended to a cell model by assuming the anode and the cathode to be a single particle with the same surface area as the electrode. The concentration in the solution phase is assumed to be constant [3,23].

The diffusion of lithium in a particle can be described as

$$\frac{\partial C_i}{\partial t} = \frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2 D_i \frac{\partial C_i}{\partial r}\right] \qquad 3.4$$

i = positive, negative electrode

with the boundary conditions

$$\frac{\partial C_i}{\partial r}_{r=0} = 0 \qquad 3.5$$

$$\frac{\partial C_i}{\partial r}_{r=Ri} = \frac{-j_i}{D_i} \qquad 3.6$$

Where j is the pore wall flux, described by the Butler Volmer kinetics

$$j_i = 2k_i c_e^{0.5}(c_{max,i} - c_{surf,i})^{0.5} c_{surf,i}^{0.5} \sinh\left(0.5\frac{F(\phi_{1,i} - U_i)}{RT}\right) = \frac{I_{app}}{a_i l_i F} \qquad 3.7$$

The model is solved by discretizing the governing equations spatially to obtain a system of differential algebraic equations (DAEs) of index 1. The equations are formulated using 15 node points for each electrode. The DAEs that are derived need to have consistent initial conditions to obtain the solution.

# CHAPTER 4 METHODS TO CONDUCT PHIL SIMULATIONS

In this section, the methods used to run a power-hardware simulation test is described. For a model to be implemented in hardware in the loop environment, the solution of the model for each time step needs to be obtained in real-time. Real-time simulation of the model is necessary to emulate a real system and send real signals to a load without any delay. A real-time target can be used to help simulate any given model for each time step in real-time.

## 4.1 Using SIMULINK to build models for Power Hardware in the Loop implementation in real time

SIMULINK is a tool within MATLAB that uses blocks to emulate the model. Equations of any system can be described with the help of blocks and subsystems predeveloped in the SIMULINK library. It is an interactive tool that can help interface between any hardware and the MATLAB environment.

The model is developed in the SIMULINK environment, which is built and loaded into the OPAL-RT target with the help of the RT-Lab software. The target is connected to a power supply, which behaves as the battery emulator which is connected to the load to be tested. A programmable load is used to specify the power demand. The programmable load can be programmed using SCPI (Standard Commands for Programmable Instruments) to emulate any power profile. A representation of the incorporating the model in SIMULINK is shown in Figure 4.1 The model describing the battery is present inside a subsystem, and an “analog out” block is used to send the signals to the power supply. The “analog in” block receives the signals from the power supply and feeds it back to the model.

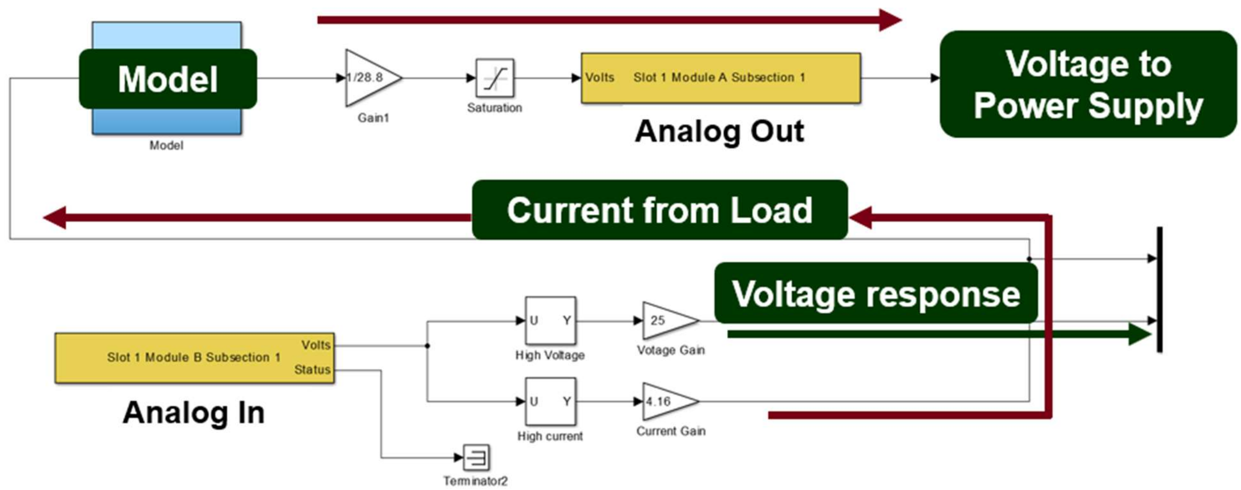


Figure 4.1 SIMULINK model for communication between the model and the hardware.

For a real-time target such as the Opal-RT to be used to solve the given model, the model needs to be loaded into the target through the software that enables building and loading models into the target. Opal-RT uses RT-Lab software to build and load models into the target. The software required models to be in the SIMULINK environment. To incorporate models with more complex equations, SIMULINK offers tools to create custom blocks such as the s-function block and the embedded MATLAB function block. Details on these methods are described in Appendix 1.

Once the model is created in SIMULINK, the model should be able to compile to generate C code for the RT-Lab software to build and load the model to the Opal-RT. The model configuration parameters enable the user to set up the type of solver SIMULINK should use. SIMULINK offers two types of solvers, variable-step solver and fixed-step solver. Each of these types offer implicit and explicit solvers depending on the model to be solved. For hardware in the loop in real-time, the model should be solved using a fixed-step solver. SIMULINK enables the user to also set the time step the model should be solved for. The OPAL-RT has the capability to solve models as small as time steps of 10 μs in real-time. However, for battery models, a time step can be chosen based on the requirements of the load to achieve real-time simulation to observe the transient responses from the load.

## 4.2 Using RT-Lab for Building and Loading the model to OPAL-RT

The equivalent circuit model described in section 2 is built in SIMULINK with the help of blocks that are present in the Simscape library that describe circuit elements such as resistors and capacitors.

The values of the parameters that describe the equivalent circuit model of the battery can be determined with the help of a Nyquist plot taken from a cell. A Nyquist plot is obtained with the electrochemical impedance spectroscopy technique for a range of frequencies from 0.5 mHz to 30000 Hz. The imaginary impedance values are plotted against the real impedance to obtain the plot. The plot was generated using the executable freeware developed by Pathak et al. when the physical parameters of the battery are inputted to obtain the EIS data [24]. This is done to ensure the equivalent circuit and single particle model have the same specifications, making it suitable for comparison.

The internal resistance of the cell is obtained by extracting the real-time impedance at the highest frequency value. The first semi-circle that describes the charge transfer resistance and double layer capacitance can be represented as a parallel RC circuit element and a Randle's circuit describes the diffusion dynamics. The second semicircle represents the dynamics caused by the SEI layer, which describe the face of the cell over cycling the battery. It is important to understand the aging effects of the battery pack when subjected to real loads.

The parameters describing the equivalent circuit are obtained using the python package impedance.py [25], where the fit of the circuit elements is used to describe the parameters.

### 4.3 Solving index-1 DAEs in SIMULINK

Physics based models of batteries use a combination of ordinary differential equations and differential algebraic equations (DAEs) which have been obtained from reducing the partial differential equations (PDEs) that describe the thermal, kinetic and diffusive behavior of the battery. The PDEs were reduced using discretization in space using finite difference scheme [23]. The system of DAEs were obtained in the Maple environment and implemented into the SIMULINK environment that runs in the OPAL-RT for conducting a PHIL simulation.

MATLAB offers a range of solvers that can be used to solve a system of differential equations. They are broadly classified into two types; variable step solvers and fixed step solvers. Variable step solvers change the time step during simulation to minimize accuracy. A solver is chosen based on the stiffness of the solution and the type of equations; a system of ordinary differential equations or a system of DAEs. In MATLAB, ode15s is a solver that is commonly used for solving a system of DAEs. It is an adaptive time step solver but if the time span is mentioned and the set tolerances of the solver are reduced, it can be used to conduct fixed-time step simulations.

The variable-step solvers available in MATLAB are also available for solving models in the SIMULINK environment. The model can be incorporated into the SIMULINK environment either through blocks or creating a custom block describing the equations. The fixed-step solvers available in SIMULINK work by updating each time step to the current time.

For running a model in MATLAB/SIMULINK, the code written needs to be compatible for code generation; i.e. conversion to C code that the OPAL-RT can understand.  To solve the system of

DAEs in SIMULINK for the OPAL-RT for PHIL, the equations need to be solved with a fixed time step solver. This is to ensure that the signals from the output of the model are sent to the hardware and updated for every time step. Figure 4.2 shows the algorithm used to incorporate a system of DAEs in SIMULINK for real-time simulations, which involves the model being updated for every time step, based on the feedback from the load.

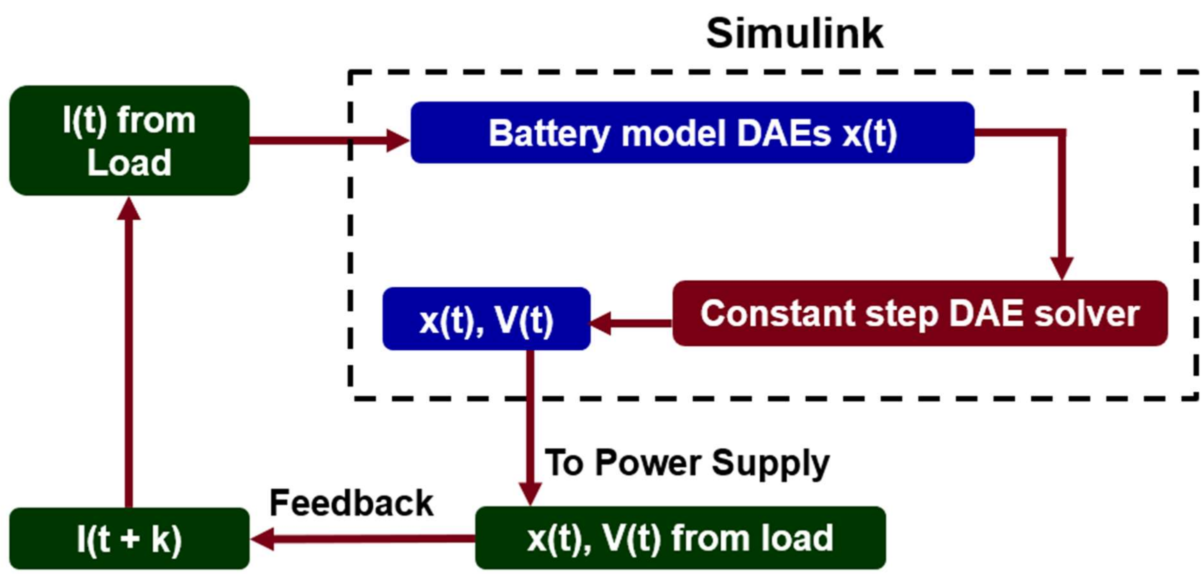


Figure 4.2 Algorithm to solve a system of DAEs in SIMULINK.

In SIMULINK, there are several methods to incorporate a system of DAEs. SIMULINK offers a range of solvers, depending on the type of problem. The user can choose the solver in SIMULINK. One method to describe the equations involves the use of SIMULINK blocks that when put together describe the governing equations. Algebraic equations are solved by taking the result from the solution to the ODEs of the system. In SIMULINK, the solver solves for the system of equations for each specified time step when the fixed-step mode is specified. This can result in an algebraic loop in the system, where the output of the system has a direct correlation with the input.

This results in an error when the model needs to be compiled for implementation in real-_time computation as the feedback for the next time step depends on the response of the present time step. Using blocks from the SIMULINK library can be cumbersome, especially when many equations are involved.

To solve these models in the SIMULINK environment, the governing DAEs can be incorporated by creating a custom block. S-function is a block in SIMULINK that allows the user to input a system of equations that can be written in MATLAB, C, C++ or FORTRAN. If the s-function is written in C, C++ or FORTRAN, the function is compiled as a MEX file. S-functions can be built in different ways. For the purpose of using SIMULINK for real-time applications such as PHIL simulations, the model needs to be compiled in C for building and loading the model to the hardware. S-function models can be written in different ways in C; handwriting the C-MEX files that can be added to the s-function block, using S-function builder and using the legacy code tool. MATLAB files can be used, but a .tlc file that describes the methods for conversion to C code needs to be included for code generation. S-function builder is a graphical tool which allows the user to add the derivatives, updates and outputs in C along with the parameters and initial conditions. The s-function builder creates a C-MEX s-function file that contains the equations, a .tlc file for code generation. The user can also specify whether to generate the MEX file required for C code generation. The legacy code tool allows the user to convert legacy code in C or C++ to an s-function file. However, this method cannot be used for continuous and discrete states.

S-functions can help integrate complex model into the SIMULINK environment. While they are useful when the model is run offline in the SIMULINK environment, it is important to note that the C-compiler can differ with the host PC and this can result in incomplete building of the model into the simulator or compilation errors when the same model is used across different systems.

MATLAB function is another feature of SIMULINK which allows the user to add a function to the model using a .m script that describes the function. This method was used to add the system of DAEs that can be solved using a constant-step Euler backward method written as a MATLAB function. The constant step Euler backward method helps to update the voltage response in real-time based on the feedback provided by the current from the load. The set of equations that describe the battery that are generated from Maple can be solved using this method.

The advantage of using this method is the compatibility across different systems, allowing code generation of the model across different PCs. The functions used to write the script must be code generation compatible. Functions in MATLAB such as feval or using function handles are not suitable for code generation. This includes using the typical DAE solvers used in MATLAB such as ode15s, forcing the user to implement a handwritten solver in the function for code generation of the model.

Table 4.1 summarizes different methods that can be used to incorporate a system of DAEs in the MATLAB/SIMULINK environment and whether they are compatible for building and loading into the real-time simulator. A model can be built into the OPAL-RT if it is suitable for code generation and can be built in MATLAB.

Table 4.1 Summary of different methods to solve a system of DAEs in SIMULINK for PHIL simulations

| Method | Suitable for building and loading to the real-time simulator | Issue |
|---|---|---|
| DAE solvers in MATLAB | No | Variable step solver if no minimum step specified; Not suitable for code generation. |
| SIMULINK blocks to describe DAEs | No | Can be tedious to incorporate a large set of equations. |
| Building an s-function | Yes (For code generation, requires a. tlc (Target Language Compiler). | Can raise compilation errors across different hosts. |
| MATLAB function with hand-written solver | Yes | |

## 4.4 Tools used for HIL Simulations of Batteries

a) Hardware in the loop simulations comprise of these main elements:

b) The host computer, which contains the model to be tested.

c) The target, the tool used for real-_time simulation of the model and the interface for communication between the model and the hardware.

d) The device under test, the hardware component in the loop that receives signals from the model.

The loop can also contain sensors for voltage or current measurement and controllers when feedback is employed. When Power HIL simulation is conducted, a power supply is used for the exchange of power signals between the components of the loop.

The OPAL-RT is used as the simulator to interface between the model and the power supply to achieve real-time simulation for hardware in the loop simulations. The Opal-RT uses RT-Lab as the software to read models and load them into the simulator. RT-Lab requires models from the SIMULINK environment. The models are then built and loaded into the simulator by using RT-Lab that generates C code that describes the model.

The load used for testing the PHIL experiments is the BK Precision 8600 programmable load. The programmable load can operate in constant voltage, constant current and constant resistive mode. For PHIL experiments, the load can either be set to constant current mode or constant resistive mode to emulate a load that discharges a battery.

# CHAPTER 5 RESULTS AND DISCUSSION

To demonstrate the need for running hardware in the loop tests for batteries, a comparison of reduced order models and physics-based models were done, to observe the necessity of implementing physics-based models for hardware in the loop applications. Physics-based models can capture the internal states of the cell. This can help understand the optimal rates at which the battery can operate for a specified load. Different models were built in the SIMULINK environment that are compatible with RT-Lab for building and loading the model onto the Opal-RT.

## 5.1 Equivalent Circuit Models

The equivalent circuit parameters used were obtained from using impedance.py [25]. The parameters obtained are fitted to the EIS data provided based on the desired circuit to be fit. The EIS data was obtained using the executable freeware developed by Pathak et al. when the physical parameters of the battery are inputted to obtain the EIS data.

Figure 5.1 shows the fit when the equivalent circuit from Figure 3.3 is fit to the EIS data obtained and the equivalent circuit parameters used to describe the model obtained from fitting the Nyquist plot. Three RC elements were considered for the fit, where the first RC element represents the SEI layer on the anode of the cell, the second represents the charge transfer kinetics and the third captures the diffusion dynamics of the cell.

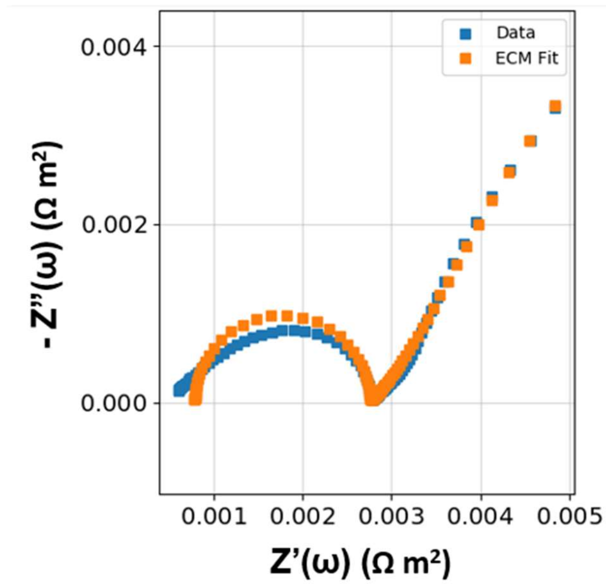


Figure 5.1 EIS Data for fitting the equivalent circuit model

The equivalent circuit model was run in SIMULINK using the s-function described in chapter 4, with equations. The coulomb counting method was used to determine the state of charge of the battery. The obtained state of charge was fed into a look-up table to determine the open circuit voltage of the cell during the charge/discharge process. The Federal Urban Drive Schedule was given as an input to the model and the voltage response of the cell was obtained. For a time-step of one second, the equivalent circuit model can capture the transient changes to the voltage when the load is changing. The equivalent circuit model is fast and computationally simple to implement in the real-time environment for PHIL simulations. However, the parameters of the model are set to be constant values. To obtain a physical interpretation of the model, the parameters need to be functions of state of charge and temperature. To model a battery pack, the output voltage and current are multiplied based on the number of cells in series and parallel.

## 5.2 Single Particle Model

The single particle model was incorporated into SIMULINK using a MATLAB function block that solves the system of DAEs with a constant step Euler backward solver, where the voltage is updated for each time step based on the feedback current from the load. The parameters and the expression for open circuit voltage of the battery as a function of state of charge used in this work to describe the model are referred from Northrop et al. [26]

The single particle model was incorporated into the SIMULINK environment, where the input fed to the model is the C-rate from the load, and the all internal variables are calculated inside the MATLAB function block, which solves the model for each time step. The model represented in SIMULINK is shown in Figure 5.2.

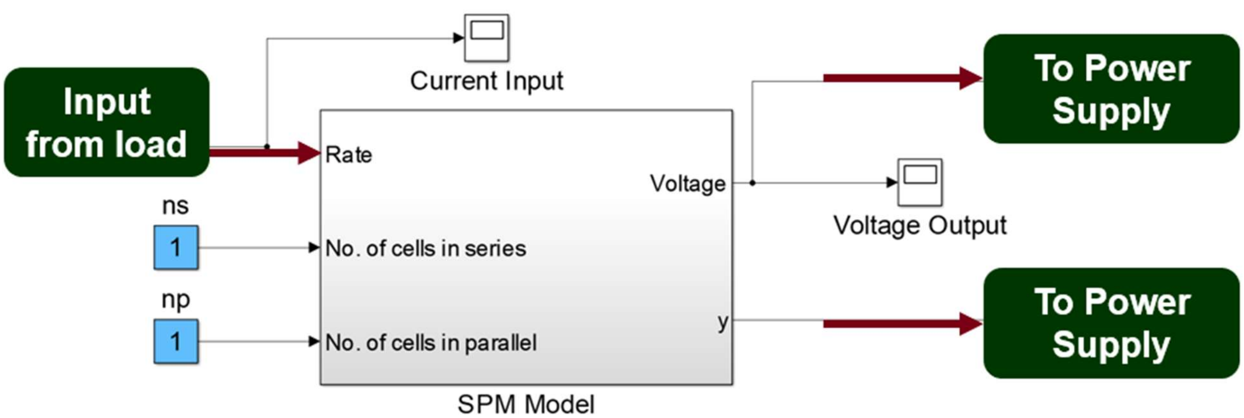


Figure 5.2 SIMULINK model of the Single Particle Model

To demonstrate the effects of the model when subjected to dynamic loads, the model was run in SIMULINK with the Federal Urban Drive Schedule profile as the input. With this model, the effect of the internal states of the battery along with voltage profile can be seen. Figure 5.3 shows the comparisons of the voltage responses obtained from the FUDS drive cycle profile from

SIMULINK. The model can capture the transient response of the cell at a time-step of one second when there is a change in the load. To model a battery pack, the number of cells in series and parallel can be adjusted. The cells in the battery pack will be assumed to have the same output voltage and current and the cells are balanced in the pack.

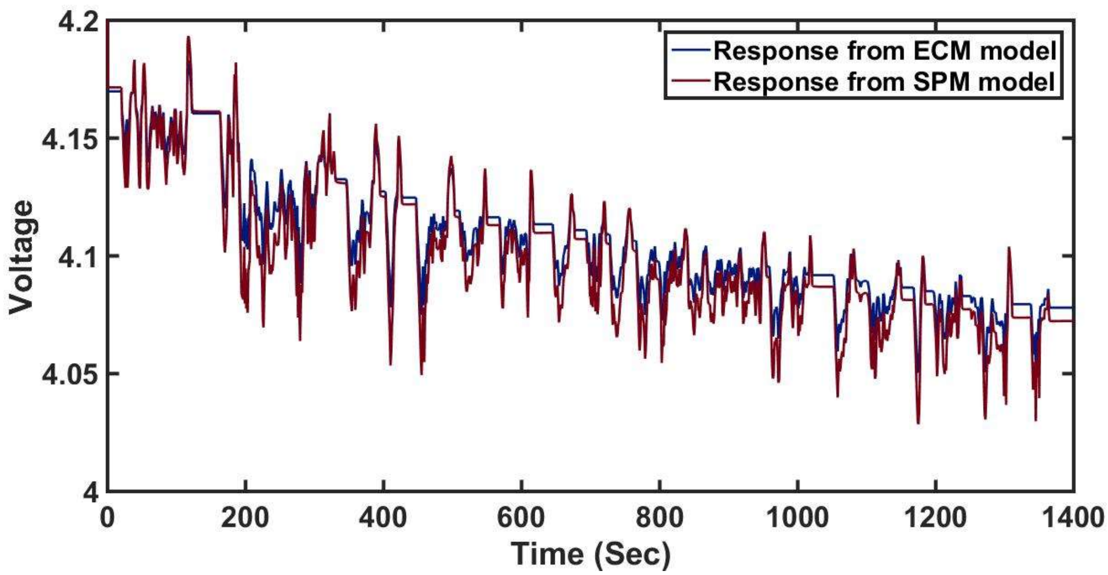


Figure 5.3 Voltage response from the FUDS cycle input obtained from the ECM and SPM model built in SIMULINK.

The advantage of using a physics-based model is to capture the internal states of the battery under a variety of conditions, while an empirical model treats the battery like a black box and captures only variables such as voltage and current from the cells. While the parameters of equivalent circuit models capture the kinetics and diffusion phenomena that occur in the cell, excessive data needs to be collected to establish a relationship that possesses physical interpretation between each of the parameters and variables such as temperature and state of charge. When one needs to study the effects of aging of batteries under the dynamic conditions that PHIL can be used for, an equivalent circuit model requires an additional RC parallel network, where the parameters are functions of temperature and state of charge. For the model to be able to describe the changes in the cell with changes in state of charge and temperature, the parameters need to be fit to available data for a range of temperature and states of charge. This results in the collection of EIS data for a variety of states of charge and temperatures. A physics-based model can solve the governing equations that solve for the variables that are affected by the aging process. For example, the thickness of the SEI layer that is formed during capacity fade of the battery can be determined by solving the equations. When these equations are solved using PHIL, such variables can be solved for in real-time under the dynamic loads that are used in these PHIL simulations. Figure 5.4 shows the thickness of the SEI layer as a function of time in a given cycle when the input to the model is the FUDS.

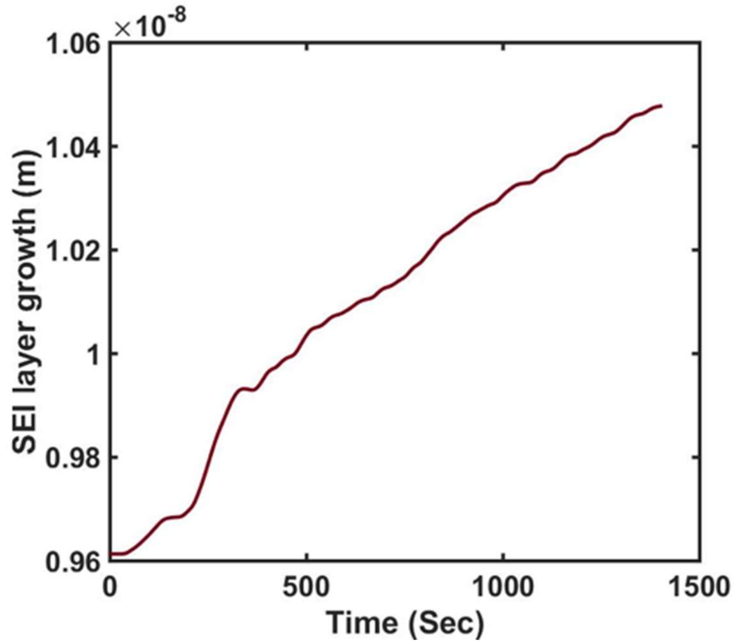


Figure 5.4 SEI layer growth obtained from SIMULINK.

## 5.3 Power Hardware in the Loop Simulations

shows the voltage response obtained when a step change of current was provided to the model from the load. a) shows the current input and b) shows the voltage response obtained. The voltage was recorded for half an hour. The input step change offered from the load was a change from a 1C to a 2C discharge rate to the battery. The equivalent circuit model was developed in SIMULINK and loaded into the OPAL-RT with the help of the RT-LAB software. The signals from the OPAL-RT were sent to the power supply to emulate the battery specified by the model.

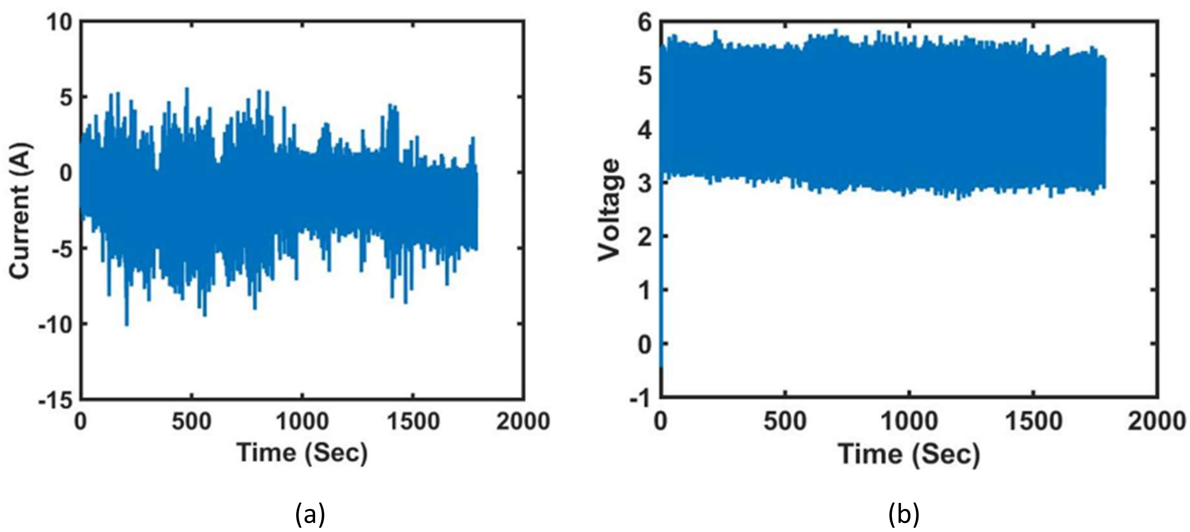


Figure 5.5 (a) The current from the programmable load in the PHIL setup. (b) The voltage response for a single cell from the PHIL simulation using an equivalent circuit model.

The voltage response obtained is for a single cell. For PHIL simulations, the power supply used needs to be able to record low voltages, as that observed from the response from a single cell. It is observed that the noise levels from the load are significant, and one possible factor is the low range of voltages in the experiment.

The time step used to solve the model in SIMULINK when running a PHIL simulation is determined by the time step of the load in the loop. For a battery, the highest frequency used to capture the cell dynamics using the EIS technique is around 10000Hz. This translates to a change in the kinetics of the cell to change at a time step of 10 milliseconds. This means that it is reasonable to study the response of the battery with a time step of 10 milliseconds using the OPAL-RT for running a PHIL simulation. However, the time-step used in the models is determined by the time-step of the hardware in the loop such as the power supply and the load under test.

# CHAPTER 6 CONCLUSION AND FUTURE WORK

Power hardware in the loop simulations can be a powerful tool to test battery designs under extreme conditions, highly dynamic profiles such as a drive cycle, data centers, etc. without building a real prototype of the battery. For the model to provide signals to the load in real-time, the computational speed of the model must be fast. This led to the exploration of empirical models such as the equivalent circuit model for testing PHIL simulations. However, these models provide little information about the internal state of the battery and offer little physical interpretation to the response from the load. Incorporating physics-based models for these types of simulations was explored. The challenge that occurs with physics-based models is with their complexity, which reduces the computational speed of the model, making the model unsuitable for real-time implementation. In this work, the single particle model was explored for PHIL implementation. The model was incorporated to the SIMULINK environment for use with the OPAL-RT for PHIL simulations.

While the proof-of concept of PHIL simulations can be proven to run a battery model in real-time, methods to prevent the noise present in the signals while running the simulations needs to be explored. The model was tested for a single cell in the SIMULINK environment. For implementing PHIL simulations for loads such as EV powertrains and the grid, the model needs to be extended to the pack level.

For future work, the pseudo 2-dimensional physics-based battery model will be implemented with the PHIL setup to demonstrate this simulation technique. The challenge with physics-based models comes the implementation of a large set of DAEs to obtain the solution. A reformulated model should be implemented to ensure the model can run in real-time. This can be incorporated into SIMULINK with the same mathematical algorithm to test if these detailed models can be run in real-time for PHIL simulations. A detailed model can help capture effects that occur in the battery such as capacity fade. Aging of the battery can be well understood with a technique such as PHIL, and how aging of the battery can occur in presence of dynamics loads that are used in EVs and grid energy storage.

# REFERENCES

1. C. Seitl, J. Kathan, G. Lauss, and F. Lehfuß, in *IEEE 23rd International Symposium on Industrial Electronics (ISIE)*,, p. 2285–2290, Istanbul, Turkey (2014).

2. Y. Li, D. Wang, S. Ghose, J. Liu, S. Govindan, S. James, E. Peterson, J. Siegler, R. Ausavarungnirun, and O. Mutlu, *Proc. - Int. Symp. High-Performance Comput. Archit.*, **2016–April**, 444–456 (2016).

3. V. Ramadesigan, P. W. C. Northrop, S. De, S. Santhanagopalan, R. D. Braatz, and V. R. Subramanian, *J. Electrochem. Soc.*, **159**, R31–R45 (2012).

4. B. M. T. Lawder, B. Suthar, P. W. C. Northrop, S. De, C. M. Hoff, O. Leitermann, M. Ieee, M. L. Crow, F. Ieee, S. Santhanagopalan, and V. R. Subramanian, *Proc. IEEE*, **102**, 1014–1030 (2014).

5. Y. He, W. Liu, and B. J. Koch, *J. Power Sources*, **195**, 2969–2974 (2010).

6. J. Marco, N. Kumari, W. Dhammika Widanage, and P. Jones, *Energies*, **8**, 8244–8262 (2015).

7. J. V. Barreras, C. Fleischer, A. E. Christensen, M. Swierczynski, E. Schaltz, S. J. Andreasen, and D. U. Sauer, *IEEE Trans. Ind. Appl.*, **52**, 5086–5099 (2016).

8. H. Dai, X. Zhang, X. Wei, Z. Sun, J. Wang, and F. Hu, *Electr. Power Energy Syst.*, **52**, 174–184 (2013).

9. Y. Zhang, R. Xiong, S. Member, H. He, and S. Member, *IEEE Trans. Power Electron.*, **32**, 4421–4431 (2017).

10. Z. Chen, A. Member, Y. Fu, and C. C. Mi, *IEEE Trans. Veh. Technol.*, **62**, 1020–1030 (2013).

11. H. Hongwen, X. Rui, Z. Xiaowei, S. Fengchun, and F. JinXin, *Veh. Technol. IEEE Trans.*, **60**, 1461–1469 (2011).

12. D. Bazargan, in *3rd International Conference on Electric Power and Energy Conversion Systems*,, p. 1–5, Istanbul, Turkey, (2013).

13. L. Gauchia and J. Sanz, *IEEE Trans. Ind. Electron.*, **57**, 1186–1194 (2010).

14. C. Seitl, J. Kathan, G. Lauss, and F. Lehfub, in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*,, p. 5412–5417 (2013).

15. O. König, C. Hametner, G. Prochart, and S. Jakubek, *IEEE Trans. Ind. Electron.*, **61**, 943–955 (2014).

16. H. Haupt, M. Plöger, and J. Bracker, *Hardware-in-the-Loop Test of Battery Management Systems*, p. 658-664, IFAC, (2013).

17. R. Xiong, H. He, F. Sun, and K. Zhao, *Energies*, 1455–1469 (2012).

18. H. He, R. Xiong, and J. Fan, *Energies*, **4**, 582–598 (2011).

19. J. Khazaei, L. Piyasinghe, V. R. Disfani, Z. Miao, L. Fan, and G. Gurlaskie, in *Proc. IEEE PES General Meeting*,, p. 1–5 (2015).

20. H. Song, T. Kim, J. Jeong, B. Kim, D. Shin, B. Lee, and H. Heo, *IET Power Electron.*, **6**, 417–424 (2013).

21. K. S. Hariharan and V. S. Kumar, *J. Power Sources*, **222**, 210–217 (2013).

22. G. L. Plett, *Battery management systems, Volume I: Battery modeling*, (2015).

23. P. W. C. Northrop, V. Suthar, Bharatkumar Ramadesigan, S. Santhanagopalan, R. D. Braatz,

and V. R. Subramanian, *J. Electrochem. Soc.*, **161**, E3149 (2014).

24. M. Pathak, M. D. Murbach, C. Pathak, T. Jang, Y. Qi, D. T. Schwartz, and V. R. Subramanian, *J. Electrochem. Soc.*, **165**, 1324–1337 (2018).

25. M. D. Murbach and D. T. Schwartz, *J. Electrochem. Soc.*, **165**, A297–A304 (2018).

26. P. W. C. Northrop, V. Ramadesigan, S. De, and V. R. Subramanian, *J. Electrochem. Soc.*, **158**, A1461 (2011).