UNIVERSITY OF CALIFORNIA SAN DIEGO

Distributed Control of Second Life Batteries in a Parallel Connected Network

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Michael Bagherpour

Committee in charge:

Professor Raymond A. de Callafon, Chair
Professor John Hwang
Professor Shirley Meng

2020

The thesis of Michael Bagherpour is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California San Diego

2020

DEDICATION

To the wonderful staff and faculty of UCSD's MAE Department.

EPIGRAPH

"[During grad school,] I learned more and more about less and less, until I found I knew everything about nothing at all"

— Professor James Friend

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to Professor De Callafon for his guidance, unending patience, and help in writing this thesis. Thanks to the professors and staff of MAE3 for trusting me with responsibility, and overlooking my failures. Thanks to my classmates for maintaining my (marginal) sanity.

This thesis is a reprint of the material as it is to appear in Preprints of the IFAC 2020 World Congress under the title “Distributed Control of Second Life Batteries in a Parallel Connected Network” by Michael Bagherpour and Raymond A. de Callafon. The thesis author was primary author of this paper.

VITA

2018	Bachelor of Science, University of California San Diego
2018-2020	Graduate Teaching Assistant, University of California San Diego
2020	Master of Science, University of California San Diego

PUBLICATIONS

Michael Bagherpour, Raymond A. de Callafon. Distributed Control of Second Life Batteries in a Parallel Connected Network. To appear in Proceedings of the IFAC 2020 World Congress.

ABSTRACT OF THE THESIS

Distributed Control of Second Life Batteries in a Parallel Connected Network

by

Michael Bagherpour

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2020

Professor Raymond A. de Callafon, Chair

Used batteries can find a second application if connected together to form a battery pack and used to power a load. Battery cells can be combined in series to form battery modules with the desired open circuit voltage, and these modules can be connected in parallel to build up the storage capacity of the battery pack. When connected in parallel, stray currents may arise within the battery pack due to heterogeneous operational parameters such as state of charge or module impedance. Additionally, the impedance of the load being driven can change, and this may also cause stray currents to arise within the pack of heterogeneous modules. The current supplied by the individual modules must be actively balanced to eliminate this. Active balancing of such a configuration has been done using buck regulators to modulate power supplied by

each module, using a centralized processor to calculate the appropriate inputs. However, control algorithms which rely on a centralized processor to calculate inputs are limited by the speed of communication between the modules and the central processor. This paper presents an approach to balance battery modules in such a configuration using buck regulators, and with decentralized calculation of control inputs. The novelty is the ability to rapidly balance the currents as the load impedance changes, independent of communication speed. We separate the control into two components: an occasional update of slow varying parameters such as Open Circuit Voltage (OCV) and module impedances, and frequent updates of control inputs. Estimation of the operational parameters of the battery modules and the impedance of their network interconnection is carried out by a central processor and communicated to the modules infrequently. The modules individually use this information in conjunction with frequent measurement of their individual currents to calculate control inputs that eliminate stray currents within the battery pack. This reduces the required communication between the central processor and the modules, allowing the system to scale up efficiently even as the number of modules grows large. In general our method can achieve any physically realizable balance of currents, but it is often beneficial to make the currents from each battery module equal to each other, so that each module drains at the same rate. This is equivalent to solving the minimization

$$\max_{PWM_j} \beta, \text{ subject to } I_{bat} = \beta \mathbf{1}_{n\times 1}, 0 \leq PWM_j \leq 100. \tag{0.1}$$

To demonstrate our method, we simulate our control algorithm, solving the above minimization on a representative three-module network with a time-varying load.

1 INTRODUCTION

Typically, EV batteries are recycled when they have degraded to about 80% of OEM storage capacity, however this may not be an economical solution [Omar et al., 2014]. As re-purposed Li-Ion EV batteries still have most of their capacity, new applications must be developed to economically utilize second-life batteries [Casals et al., 2019]. Re-purposed batteries of an electric vehicle (EV) may provide economical short-term storage of intermittent electric energy produced by renewable energy sources. Reusing these batteries has economic and environmental benefits [Elkind, 2014, Sathre et al., 2015, Jiao and Evans, 2016], and there are enough EV batteries existing to make large scale energy storage possible [Almeida and Nunes, 2018].

Most existing work has focused on the design of a Li-Ion Battery Management System (BMS) with enhanced State of Charge (SoC) or State of Health (SoH) estimation to ensure a safe range of operation [Huang et al., 2017]. In case of an unbalanced SoC or temperature in the pack, balancing techniques may be used by the BMS to readjust the SoC of the battery pack [Altaf et al., 2014] or adjust minimum and maximum SoC levels during operation [Danko et al., 2019]. SoC/SoH monitoring and balancing control improves the long-term reliability of a battery pack, but they do not account for the internal resistance or impedance of Li-Ion batteries, which is a key parameter in determining power output and energy efficiency [Schweiger et al., 2010].

Although there are many restrictions in electrical codes and standards that limit re-purposed battery pack design for energy storage [Catton et al., 2019], one common challenge is combining batteries with heterogeneous operational parameters that include open circuit voltage,

charge capacity and internal impedance. In this paper, re-purposed Li-Ion battery cells are placed in series to create battery modules. The combination of Open Circuit Voltage (OCV) and module impedance determines the terminal voltage of a module under load conditions. Battery modules with differing impedance and OCV are then connected in parallel to build a battery pack with the desired energy storage capabilities. The parallel connections of heterogeneous modules may create stray currents within the battery pack, so the terminal voltage of the modules must be regulated to eliminate stray currents, see e.g. [Jiang et al., 2019].

This paper is concerned with the connection of battery modules in a parallel network similar to [Zhao et al., 2014], and the control of battery currents without requiring high-bandwidth communication to a central processor. Specifically, we would like to make the currents from each battery module equal to each other, so that each module drains at the same rate. This is equivalent to solving the minimization

$$\max_{PWM_j} \beta, \text{ subject to } I_{bat} = \beta \mathbf{1}_{n\times 1}, 0 \leq PWM_j \leq 100. \tag{1.1}$$

With access to the appropriate parameters of a battery network, a central processor can solve this optimization and send inputs to individual battery modules. However, this control scheme would be limited in the speed at which it can update inputs by the communication bandwidth of the batteries and central processor. We can eliminate the dependence on communication bandwidth by distributing a centralized control algorithm similar to that of [Jiang et al., 2019] to each battery module. In order to distribute the calculation of control inputs to each module, the modules must independently estimate the load being driven by the pack. To facilitate this estimation, we allow each module to measure the current flowing through itself. Using fast sampling and control within each battery module, the central processor is only used to provide infrequent updates to estimates of system parameters. The approach is illustrated on a representative three battery network model in which each battery has a different internal impedance and OCV.

2 Battery Modules Network

2.1 Battery Module Parameters and Signals

As in [Jiang et al., 2019], battery modules are created by placing battery cells in series to create the desired Open Circuit Voltage (OCV), and then adding a buck regulator in series with the cells. A buck regulator uses Pulse Width Modulation (PWM) to reduce the current supplied by the battery module, effectively allowing the battery modules to "simulate" a battery cell of any voltage lower than its OCV. For our analysis, it suffices to describe a module j as a controllable voltage source with a terminal voltage

$$V_j = V_{PWM}^j - R_j I_{bat}^j, \;\; V_{PWM}^j = f_j(V_{OCV}^j, PWM_j)$$

where V_{OCV}^j is the OCV of module j and PWM_j is the PWM of the buck regulator in the module. The function $f_j(\cdot)$ is a module specific , PWM dependent non-linear function with $f_j(V_{OCV}^j, 0) = 0$ and $f_j(V_{OCV}^j, 100) = V_{OCV}^j$. As the internal impedance of a Li-Ion battery cell is a key parameter in determining power output and energy efficiency [Schweiger et al., 2010], the module includes a module impedance R_j in series with the regulated voltage source.

The module current I_{bat}^j varies due to possibly rapid fluctuations in the load powered by the network. The overarching objective is to use inputs PWM_j to control currents I_{bat}^j out of n parallel placed battery modules $j = 1, 2, \ldots, n$ while powering a varying load. Specifically, we equalize the module currents while maximizing power output in the presence of a varying load,

though other criteria can be achieved with the same method we present.

Unlike the rapidly fluctuating load, the open circuit voltage V_{OCV}^j and battery impedances R_j are assumed to depend on the SoC and SoH of the battery module, and thus change more slowly. The distinction between slowly varying parameters V_{OCV}^j, R_j, and the rapidly changing I_{bat}^j, PWM_j motivates a separation of the calculation of control inputs in the individual modules and the communication of parameter information between modules.

Batteries are assumed to each have their own rapid measurement of the current I_{bat}^j, while receiving central information of V_{OCV}^j, R_j of all the battery modules from a central BMS at a much lower update rate. The method through which each battery controls I_{bat}^j via PWM_j will be summarized in the following sections.

2.2 Network Model and Notation

For the presentation of the main concepts behind the distributed control of the battery modules, a network of $n = 3$ parallel battery modules as depicted in Fig. 2.1 is considered. The dashed boxes represent individual battery modules $j = 1, 2, 3$, each with slowly varying parameters V_{OCV}^j, R_j. Each module has a PWM input PWM_j and can measure its own current I_{bat}^j due to its input PWM_j. The open-circuit terminal voltage or no-load voltage $V_{PWM}^j = f(V_{OCV}^j, PWM_j)$ is not measurable, but can be calculated by module j if the OCV V_{OCV}^j, the PWM signal PWM_j, and the function $f_j(\cdot)$ are known. The load driven by the network of battery modules is represented R_{load}.

In addition to the individual battery module impedance R_j and the load impedance R_{load}, the network of battery modules also considers the line impedance R_{ij} between the terminal connection of the modules. When controlling the battery currents I_{bat}^j, the constant or even slowly varying line impedance should not be ignored as it may be comparable in size to the battery module impedance when battery cell impedance is in the order of several milliohms [Mathew

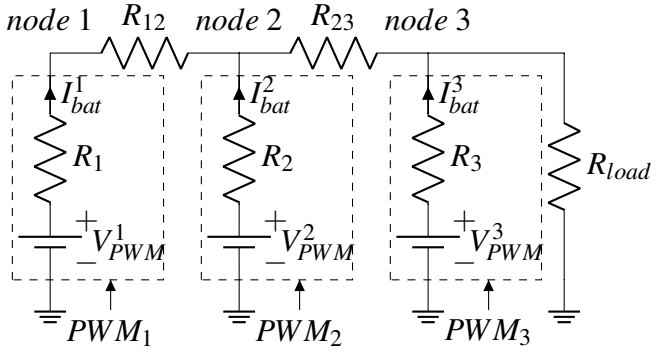

Figure 2.1: Network of $n = 3$ parallel placed battery modules subjected to a load impedance R_{load}, where dashed boxes represent individual battery modules.

et al., 2018].

By switching from impedance R in the units of *Ohm* to admittance $y = 1/R$ in the units of *Mho*, the relation between the battery module terminal voltages V_j and the battery module currents I^j_{bat} can be given in the form of an admittance matrix. For notational convenience we define the $n \times n$ total admittance matrix

$$Y_{total} = Y_{line} + Y_{bat} + Y_{load}$$

where $Y_{line} \in \mathbb{R}^{n\times n}$ is the line admittance matrix given by

$$Y_{line} = \begin{bmatrix} \sum\limits_{j\neq 1} y_{1j} & -y_{12} & \cdots & -y_{1n} \\ -y_{21} & \sum\limits_{j\neq 2} y_{2j} & & \vdots \\ \vdots & & \ddots & \\ -y_{n1} & \cdots & & \sum\limits_{j\neq n} y_{nj} \end{bmatrix} \tag{2.1}$$

Note that in the network model there is no R_{13} connecting node 1 to node 3, so the admittance $y_{13} = y_{31} = 0$. $Y_{bat} \in \mathbb{R}^{n\times n}$ is the battery admittance matrix and $Y_{load} \in \mathbb{R}^{n\times n}$ is the load admittance

matrix, respectively given by

$$Y_{bat} = \begin{bmatrix} y_1 & 0 & \dots & 0 \\ 0 & y_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & & y_n \end{bmatrix}, \quad Y_{load} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & & y_{load} \end{bmatrix} \tag{2.2}$$

Finally, we will use the notation $I_{bat} = \begin{bmatrix} I_{bat}^1 & I_{bat}^2 & I_{bat}^3 \end{bmatrix}^T$ to denote a vector of currents through the batteries and $V_{PWM} = \begin{bmatrix} V_{PWM}^1 & V_{PWM}^2 & V_{PWM}^3 \end{bmatrix}^T$ denotes a vector of no-load voltages in the three battery module network of Fig. 2.1. The actual nodal voltages for the three battery module network of Fig. 2.1 are combined in $V_{node} = \begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix}^T$.

3 Battery Scheduling

3.1 Problem Statement

The objective is to equalize the currents I_{bat}^{j} out of n parallel battery modules $j = 1, 2, \ldots, n$, while maximizing power output to a varying load R_{load}. This can be formulated as the maximization

$$\max_{PWM_j} \beta, \text{ subject to } I_{bat} = \beta \mathbf{1}_{n\times 1}, 0 \leq PWM_j \leq 100 \tag{3.1}$$

where $\mathbf{1}_{n\times 1}$ denotes a $n \times 1$ vector with all elements equal to one and PWM_j of each battery module is scaled between 0% and 100%. The maximization of the current $I_{bat}^{j} = \beta$, $j = 1, 2, \ldots, n$ while enforcing $PWM_j \leq 100$ ensures that at least 1 battery module will always run at 100% PWM, while all other battery modules will be buck regulated down in voltage to ensure equal module currents.

We propose that each battery module j conduct the optimization in (3.1) and apply only their optimal input, PWM_j. The optimization in (3.1) can be solved via a line search, allowing the individual modules to quickly solve for the full vector of control inputs even as the number of battery modules scales up. If the optimization were to be carried out by a central processor, the frequency at which control inputs can be updated would depend on the communication bandwidth between each battery module and the central processor, as sensor readings and optimal control inputs need to be communicated back and forth. This adds cost and complexity as the number of modules increases.

To carry out our distributed optimization, each module needs information on the OCV of the battery modules, the admittances y_j and y_{ij}, the function f_j mapping PWM_j and OCV_j to V^j_{PWM}, and R_{load}. We assume battery j has information on the admittance y_j, y_{ij}, and the OCV of all the battery modules in the network because those values vary slowly, as discussed in Section 2.1. These parameters can be estimated by the BMS of each battery module separately, see e.g. the work by [Schweiger et al., 2010] and [Huang et al., 2017] or [Danko et al., 2019]. The estimation of y_j and OCV V^j_{OCV} is outside the scope of this paper: it is assumed to be generated by the BMS of each battery module and communicated centrally for distribution to each module as often as needed. The function f_j can be approximated by a linear function, which we will address in more detail in Section 3.4. All that remains is for the battery modules to gain information on the quickly varying R_{load} in a decentralized manner.

In the following sections, we will provide a method for module j to estimate R_{load} without communication with other modules or a central processor. As an intermediate step, module j will estimate the full V_{node} using its measurement of the current I^j_{bat} flowing through module j. We require measurement of I^j_{bat}, because it is easy to measure and gives enough information to estimate y_{load}. The method presented could be modified to accommodate a different measurement, for instance measurement of the voltage V^j_{node} at node j. Section 3.2 will address the computation of V_{node} by battery module j, Section 3.3 will address the estimation of R_{load}, and Section 3.4 will address solving the optimization and the application of the optimal control inputs.

3.2 Computation of Node Voltages

In order to implement decentralized control of the battery currents, each battery must have a method to compute the node voltages in the battery network. This section provides a method to achieve this, followed by a derivation of the method.

Theorem 1. Consider the admittance matrices Y_{line}, Y_{bat}, and Y_{load} in (2.1) and (2.2). Then V_{node}

can be calculated with the following two equations:

$$V_{node}^{j} = V_{PWM}^{j} - I_{bat}^{j} R_j \tag{3.2}$$

$$\begin{aligned} I_{-j}^{T} V_{node} = [E(Y_{line} + Y_{bat})(I_n - J_j)I_{-j}]^{-1} \times \\ (EY_{bat}V_{PWM} - E(Y_{line} + Y_{bat})J_j V_{node}) \end{aligned} \tag{3.3}$$

with

$$E = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}_{n-1 \times n}$$

indicating an $n \times n$ identity matrix with the last row removed. J_j is defined for the jth battery as an $n \times n$ 0 matrix, with a 1 in the (j,j) location. I_{-j} is defined for the jth battery as the $n \times n$ identity matrix with the jth column removed, giving an $n \times n-1$ matrix.

Proof 1. From nodal analysis, the module currents are

$$I_{bat} = (Y_{line} + Y_{load})V_{node} = (Y_{total} - Y_{bat})V_{node} \tag{3.4}$$

V_{PWM} is the voltage at the nodes plus the voltage drop across the internal impedance in each module: $V_{PWM} = V_{node} + Y_{bat}^{-1} I_{bat}$. Using (3.4) we can write this as

$$V_{PWM} = Y_{bat}^{-1} Y_{total} V_{node} \tag{3.5}$$

Moving unknowns to the right hand side we have

$$Y_{bat} V_{PWM} = Y_{total} V_{node} \tag{3.6}$$

Although Y_{total} is an $n \times n$ matrix, the battery modules have information on all elements except for the lower right element, due to the y_{load} term. We can eliminate the unknown term y_{load} in (3.6) by defining

$$E = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}_{n-1 \times n}$$

indicating an $n \times n$ identity matrix, with the last row removed. The purpose of the introduction of the matrix E is summarized in the following remark.

Remark 1. If an $n \times m$ matrix is premultiplied by E, the first $n-1$ rows are extracted.

Premultiplying (3.6) by E, we have

$$\begin{aligned} EY_{bat}V_{PWM} &= EY_{total}V_{node} \\ &= (EY_{line} + EY_{bat} + EY_{load})V_{node} \end{aligned}$$

The first $n-1$ rows of Y_{load} are 0, so by Remark 1 we have

$$EY_{bat}V_{PWM} = E(Y_{line} + Y_{bat})V_{node} \tag{3.7}$$

Note that module j can calculate the voltage at node j via

$$V^j_{node} = V^j_{PWM} - I^j_{bat}R_j \tag{3.8}$$

We will partition V_{node} into the voltage V^j_{node} at node j, and the voltages at the other nodes via the definition of the matrix J_j for the jth battery as an $n \times n$ 0 matrix, with a 1 in the (j,j) location. The purpose of the introduction of the matrix J_j is summarized in the following remark.

Remark 2. If an $m \times n$ matrix is postmultiplied by $(I_n - J_j)$, the j'th column is set to zero.

Observe that $(I_n - J_j)V_{node} + J_j V_{node} = V_{node}$. Making this substitution for V_{node} in (3.7) we have

$$EY_{bat}V_{PWM} = E(Y_{line} + Y_{bat})(J_j V_{node} + (I_n - J_j)V_{node})$$

and

$$EY_{bat}V_{PWM} - E(Y_{line} + Y_{bat})J_j V_{node} =$$
$$E(Y_{line} + Y_{bat})(I_n - J_j)V_{node}$$

Note that module j can calculate $JjV_{node} = V_{node}^j$ by (3.8). By Remark 2, $E(Y_{line} + Y_{bat})(I_n - J_j)$ has its jth column set to zeros. That is, the right hand side is unaffected by V_{node}^j. Noting this unused data, and that $E(Y_{line} + Y_{bat})(I_n - J_j)$ is an $n-1 \times n$ matrix by Remark 1, we will eliminate the unused column to get a square matrix as follows.

Define matrix I_{-j} for the jth battery to be the $n \times n$ identity matrix with the jth column removed, giving an $n \times n-1$ matrix. The reason for the introduction of the matrix I_{-j} is summarized by the following two remarks.

Remark 3. If an $m \times n$ matrix is postmultiplied by I_{-j}, all except the jth column are extracted from the matrix.

Remark 4. If an $n \times m$ matrix is premultiplied by I_{-j}^T, all except the jth row are extracted from the matrix.

By Remark 3, post-multiplication of $E(Y_{line} + Y_{bat})(I_n - J_j)$ by I_{-j} will eliminate the column of zeros. Premultiplication of V_{node} on the right hand side by I_{-j}^T keeps dimensions consistent. By Remark 4, this premultiplication removes V_{node}^j which was not being used in the right hand side, so no information is lost by these operations. In summary, the following equation

is obtained

$$EY_{bat}V_{PWM} - E(Y_{line} + Y_{bat})J_jV_{node} =$$
$$E(Y_{line} + Y_{bat})(I_n - J_j)I_{-j}I_{-j}^TV_{node}$$

$E(Y_{line} + Y_{bat})(I_n - J_j)I_{-j}$ is invertible if all line resistances and battery resistances are strictly positive values. Thus we have (3.3) from the beginning of this section:

$$I_{-j}^TV_{node} = [E(Y_{line} + Y_{bat})(I_n - J_j)I_{-j}]^{-1} \times$$
$$(EY_{bat}V_{PWM} - E(Y_{line} + Y_{bat})J_jV_{node})$$

This equation allows module j to calculate the voltage at all nodes except node j, and the voltage at node j can be calculated via (3.2). Thus battery module j can compute the full vector of node voltages V_{node} for the battery network, as described in theorem 1.

3.3 Load Estimation

With the node voltages calculated as described in Section 3.2, the next task is to estimate the load being driven by the battery network. This section provides a method to achieve this, followed by a derivation of the method.

Theorem 2. Consider a module which has obtained the full vector V_{node}. Then y_{load} can be found via

$$y_{load} = \overline{E}(Y_{bat}V_{PWM} - Y_{line}V_{node} - Y_{bat}V_{node})/V_{node}^n \tag{3.9}$$

with $\overline{E}$ defined by

$$\overline{E} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}_{1\times n}$$

Proof 2. The result in (3.6) can be revisited to find

$$Y_{bat}V_{PWM} - Y_{line}V_{node} - Y_{bat}V_{node} = Y_{load}V_{node}. \tag{3.10}$$

which will provide a way to estimate the load impedance. Recall that each battery module j has information on Y_{bat} and Y_{line}, and full access to V_{node} from (3.3) and (3.2). Additionally, each module will run the same algorithm for load following by calculating the full optimal PWM vector, and so has an estimate of the control inputs PWM of all the modules. V_{PWM} can be estimated via the linear approximation of $f(\cdot)_j$ which is laid out in Section 3.4. Y_{load} is entirely 0's, except for the last element of the last row, which is y_{load}. Thus the right hand side of (3.10) is entirely 0's, except for the last row. Defining the matrix

$$\overline{E} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}_{1\times n}$$

provides the ability to extract the last row of an $n \times m$ matrix by premultiplication with $\overline{E}$. This property can be used to write

$$\begin{aligned} \overline{E}(Y_{bat}V_{PWM} - Y_{line}V_{node} - Y_{bat}V_{node}) =& \overline{E}Y_{load}V_{node} \\ =& y_{load}V^n_{node} \end{aligned}$$

Where V^n_{node} denotes the nth element in the V_{node} vector. Since V^n_{node} is a scalar, we now have

$$y_{load} = \overline{E}(Y_{bat}V_{PWM} - Y_{line}V_{node} - Y_{bat}V_{node})\frac{1}{V^n_{node}} \tag{3.11}$$

for an estimate of the load admittance. This can be computed in each battery module $j = 1, 2, \dots, n$ so that information on y_{load} is available via the distributed computation of (3.2), (3.3), and (3.11) in every module.

3.4 Finding Optimal PWM_j

In Section 3.1 we discussed that our goal was to maximize power delivery to the load R_{load}, subject to the constraint that all battery modules provide the same current. This could be desirable if all modules have the same SoC levels and storage capacity. In order to ensure modules are discharged and charged at the same rate despite differences in the internal module impedances R_j or line impedances R_{ij}. The formulation from Section 3.1 is repeated here for convenience:

$$\max_{PWM_j} \beta, \text{ subject to } I_{bat} = \beta \mathbf{1}_{n\times 1}, 0 \leq PWM_j \leq 100 \tag{3.12}$$

where $\mathbf{1}_{n\times 1}$ denotes an $n \times 1$ vector with all elements equal to one and PWM_j of each battery module is scaled between 0% and 100%. If a function can be found which relates PWM to I_{bat}, then (3.12) can be solved via line search to find the optimal input vector PWM.

Although $V^j_{PWM} = f_j(V^j_{OCV}, PWM_j)$ is unknown and may be nonlinear, we note that battery scheduling typically takes place with PWM_j close to 100%. This motivates the initial approximation

$$V^j_{PWM} = \frac{V^j_{OCV}}{100} PWM_j \tag{3.13}$$

which is assumed to be close to $f_j(\cdot)$ when PWM_j is near 100%. A different approximation may be preferred depending on known behavior of the buck regulators. For the sake of the tests conducted for this paper, we will proceed by using

$$V^j_{PWM} = 0.5V^j_{OCV} + 0.5\frac{V^j_{OCV}}{100} PWM_j, \tag{3.14}$$

which is a slight modification of (3.13) with more gradual slope.

V_{PWM} is related to I_{bat} via (3.4) and (3.5), duplicated here:

$$I_{bat} = (Y_{line} + Y_{load})V_{node} = (Y_{total} - Y_{bat})V_{node} \tag{3.15}$$

$$V_{PWM} = Y_{bat}^{-1} Y_{total} V_{node} \tag{3.16}$$

Using (3.15), (3.16), and the approximation in (3.14), we have

$$I_{bat} = (Y_{line} + Y_{load}) Y_{total}^{-1} Y_{bat} \left(0.5 V_{OCV} + 0.5 \frac{V_{OCV}}{100} \odot PWM \right) \tag{3.17}$$

where $\odot$ represents the Hadamard product. Since battery modules can calculate y_{load} via (3.11), they now have information on all the admittances, making (3.17) easy to calculate for a chosen PWM. Thus module j can find the inputs PWM satisfying (3.12) via a line search and the relationship (3.17).

In this way, battery module j calculates the optimal inputs for *all* the modules. Module j then inputs only PWM_j, taken from the full vector of optimal PWM inputs it has calculated.

4 BATTERY SCHEDULING RESULTS

4.1 Simulation Design

To test the control algorithm, we simulate the network with a varying load for the representative three-module network in Fig. 2.1, with the modules running the control algorithm we have described in this paper. A simulated battery module j uses only the reading of it's own current I_{bat}^{j}, calculates it's own estimation of the load R_{load}, and has access to its own PWM input PWM_j. The voltage outputs of the simulated battery modules are used to calculate the currents in the network, and then readings of the current through each module are used by the modules to calculate their next inputs. Each reading of the simulated current has noise added to it, ranging from -5% to 5% of the value measured, with the percentage picked from a uniform distribution. Additionally, each reading is truncated to 8 bits of precision, and the PWM commands of each module are truncated to 10 bits of precision. The module impedances, the line impedance values, and the module OCV's, which the battery modules have information on at the start the simulation, are summarized in Table 4.1 and Table 4.2.The load R_{load} varies through time, with its trajectory indicated by Fig. 4.1

Table 4.1: Numerical values in Ohms for battery module impedance R_j, $j = 1,2,3$ and line impedance R_{ij} for the $n = 3$ parallel placed battery modules shown in Fig. 2.1.

R_1	R_2	R_3	R_{12}	R_{23}
0.47	0.44	0.4	0.09	0.08

Table 4.2: Numerical values in Volts for battery module OCV's V_{OCV}^j, $j = 1,2,3$ for the $n = 3$ parallel placed battery modules shown in Fig. 2.1.

V_{OCV}^1	V_{OCV}^2	V_{OCV}^3
155	160	165

In order to ensure safety during the start of the algorithm, it is desirable to slowly increase the maximum allowable PWM value at the start, so we include this in the simulation.

Three different conditions were simulated which varied in their treatment of the nonlinear PWM mapping. For all simulation conditions, we assume the true mapping is

$$V_{PWM}^j = f_j(V_{OCV}^j, PWM_j) = 10\sqrt{PWM_j}V_{OCV}^j \tag{4.1}$$

In the first simulation condition, the "true" mapping is known to each module, and modules carry out their computations using $f_j(\cdot)$ as shown in (4.1). In this case, there is no need for the approximation in (3.14) or (3.13). This simulation is unrealistic in practice, and is meant only to illustrate the best-case performance we could expect from the control algorithm. The second simulation condition is with the nonlinear $f_j(\cdot)$ approximated by the function described in (3.14), which can be seen in Fig. 4.2. The third simulation condition uses a different approximation for the "weakest" battery module, by which we mean the module which achieves the lowest amperage per PWM commanded. For the "weakest" module k, the nonlinear $f_k(\cdot)$ is approximated by the function

$$V_{PWM}^k = 0.5V_{OCV}^k + \alpha\frac{V_{OCV}^k}{100}PWM_k, \tag{4.2}$$

and the reading of the "weakest" battery module's current is provided by the central processor to each module every thirty samples. The reading provided by the central processor allows each module j to estimate how much it is overpowering the weakest module. By tuning the parameter α in (4.2), the line search conducted by each module changes slightly, and the optimal PWM input each module finds also changes slightly. The goal of this change is to account for mis-knowledge

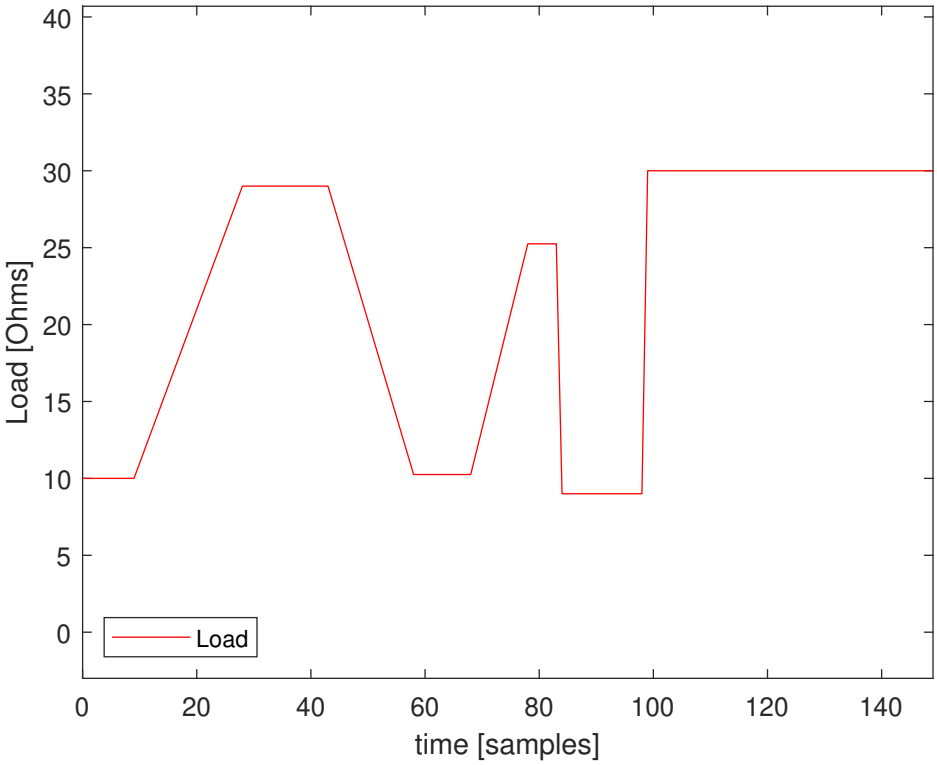

Figure 4.1: The load R_{load} being driven by the power network in Fig. 2.1 varies with time

of the system parameters in a simple way. In addition to the simulations mentioned above, we also construct a simulation to evaluate the effect of mis-knowledge of line and battery impedances. Instead of using the "true" resistances provided in Table 4.1, battery modules carry out their local computations using resistances values that vary uniformly from -0.05 to 0.05 Ohms. The resulting PWM commands are then given as inputs to the simulated circuit, which still uses the "true" resistances from Table 4.1. In this way, we can see the effect of incorrect resistance values on the current balancing, and evaluate the importance of accurate resistance measurements for the control algorithm. Results for the above simulations are presented in the following sections.

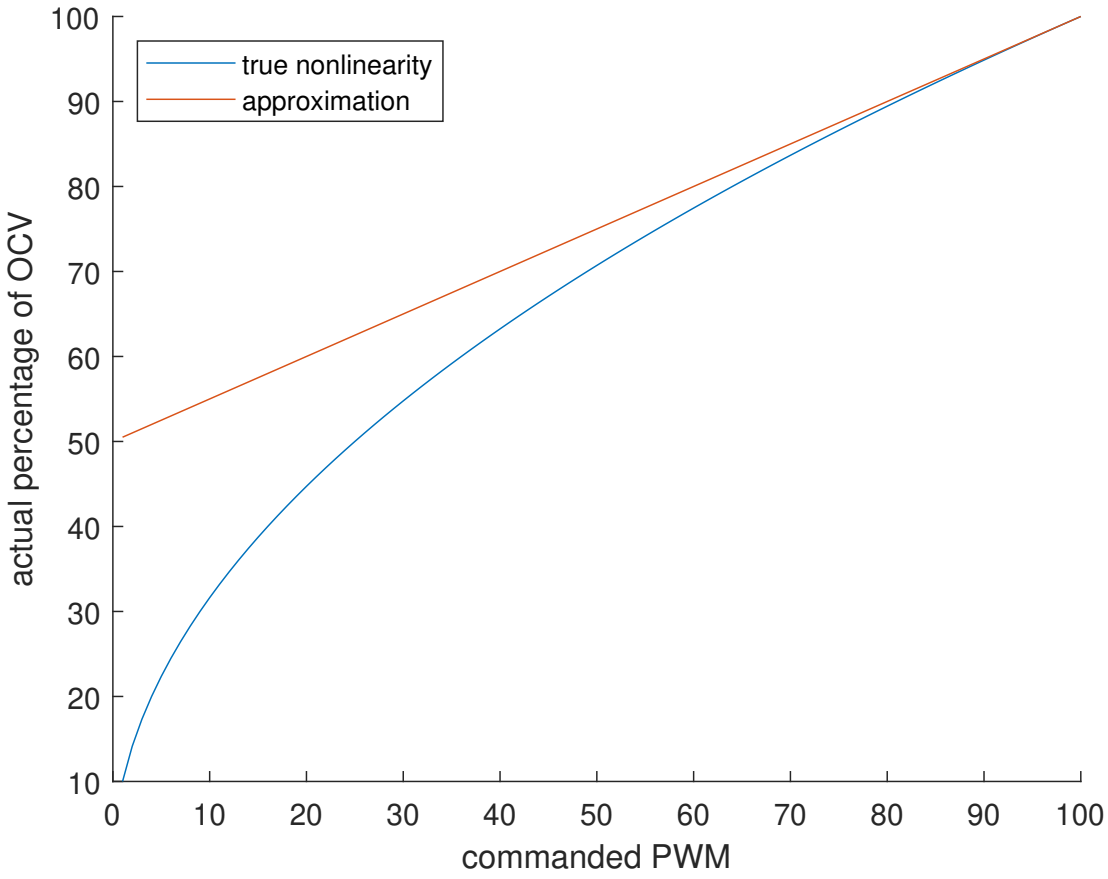

Figure 4.2: The true mapping $f_j(\cdot)$ and the static approximation used by each battery module. The tunable approximation can be thought of as the same function, but with a moving endpoint.

4.2 Fully Distributed Control with Known f_j's

If the f_j's are exactly known, the approximation in (3.14) is unneeded, and the modules can calculate y_{load} and balance their currents with very high accuracy.

Results for the estimation of the load impedance R_{load} using the three network battery system with the initial information of battery module impedance and line impedance values summarized in Table 4.1 are given in Fig. 4.3. It can be observed from this figure that despite noise on the module current and limited resolution measurements of the 8bit conversion, both the linear and abrupt changes in the load impedance R_{load} are tracked by each battery module.

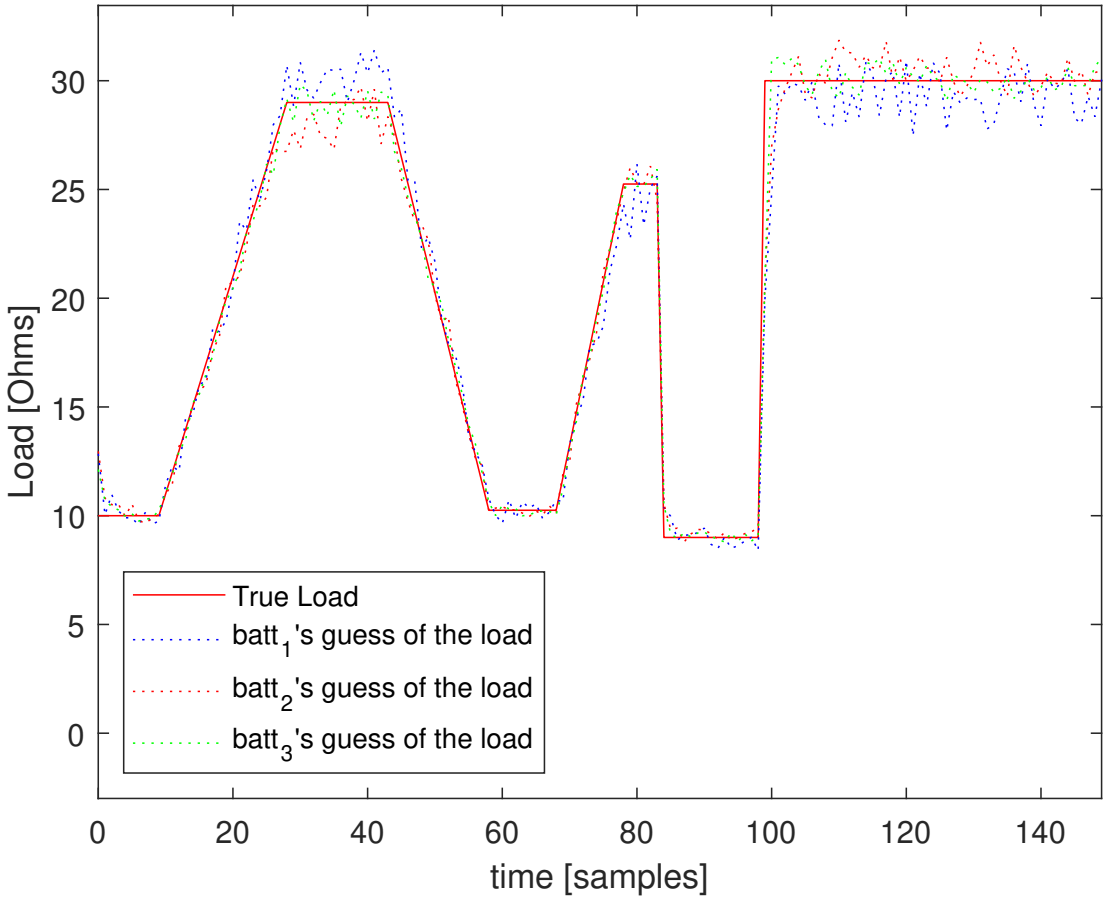

Figure 4.3: Estimation of varying load impedance R_{load} as a function of time in each battery module $j = 1,2,3$ based on distributed noisy observations of module current I^j_{bat} in each module.

The tracking of the varying load impedance R_{load} as a function of time in each battery

module $j = 1,2,3$ provides each module with accurate information on the full network admittance matrix Y_{total} by the computed $y_{load} = 1/R_{load}$, since battery and line impedance have already been initialized at the start of the simulation. The resulting measured current out of each battery module is summarized in Fig. 4.4 where it can be seen that module currents I^j_{bat} $j = 1,2,3$ are equal within the margin of the measurement error and noise. Unbalanced current deviations are observed when there are fast changes in the external load impedance R_{load}, but such imbalances are quickly settled by the fast distributed control algorithm on each battery module.

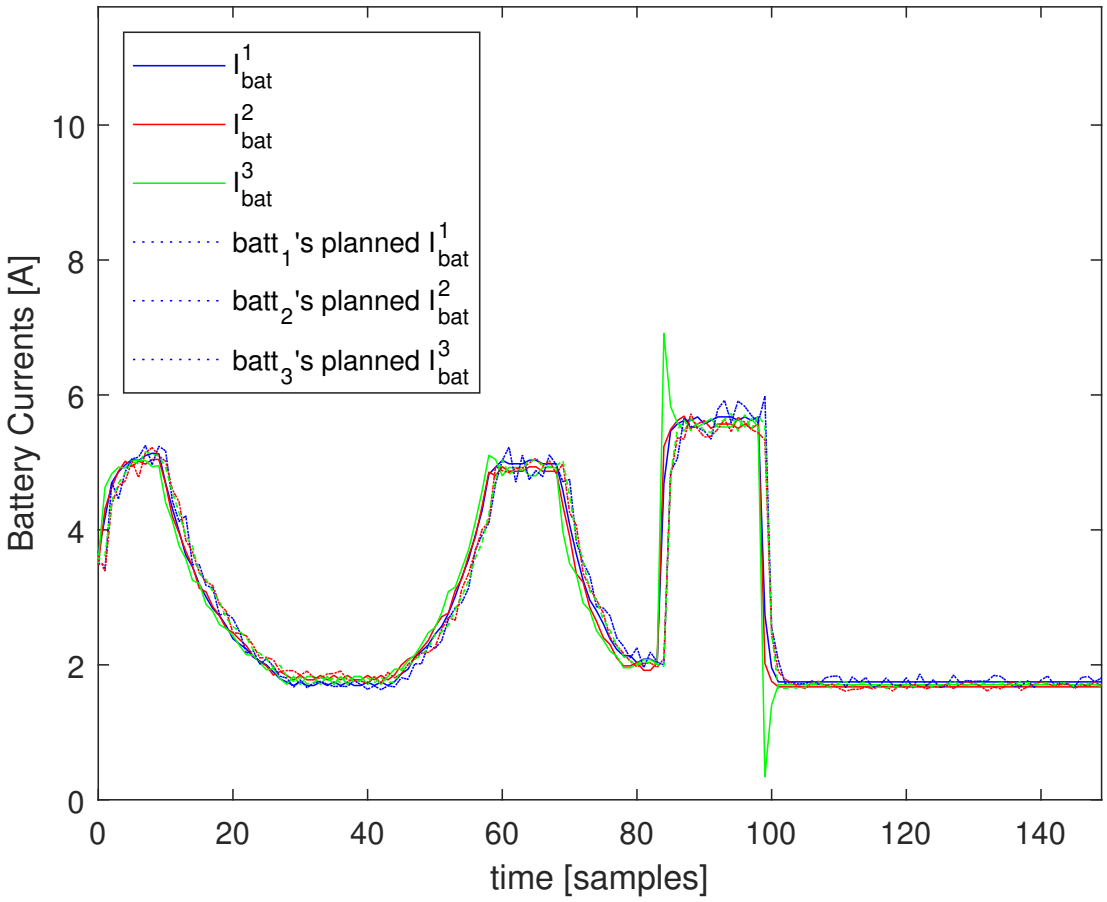

Figure 4.4: Measured battery module currents I^j_{bat} as a function of time in each battery module $j = 1,2,3$ due to the load variations depicted in Fig. 4.3. The spikes near samples 80 and 100 are due to instant changes in R_{load}

It is worthwhile to observe the subtle but important difference in the PWM values PWM_j

for each of the battery modules $j = 1,2,3$ in Fig. 4.5. The PWM of the first module quickly converges to a 100% PWM level, while the other modules are buck regulated to lower PWM values of around 93% and 87% respectively. The abrupt load changes depicted earlier in Fig. 4.3 necessitates the observed subtle changes in the PWM levels of battery module 2 and 3 to maintain balanced module currents, while module 1 remains at 100% PWM for maximum power delivery.

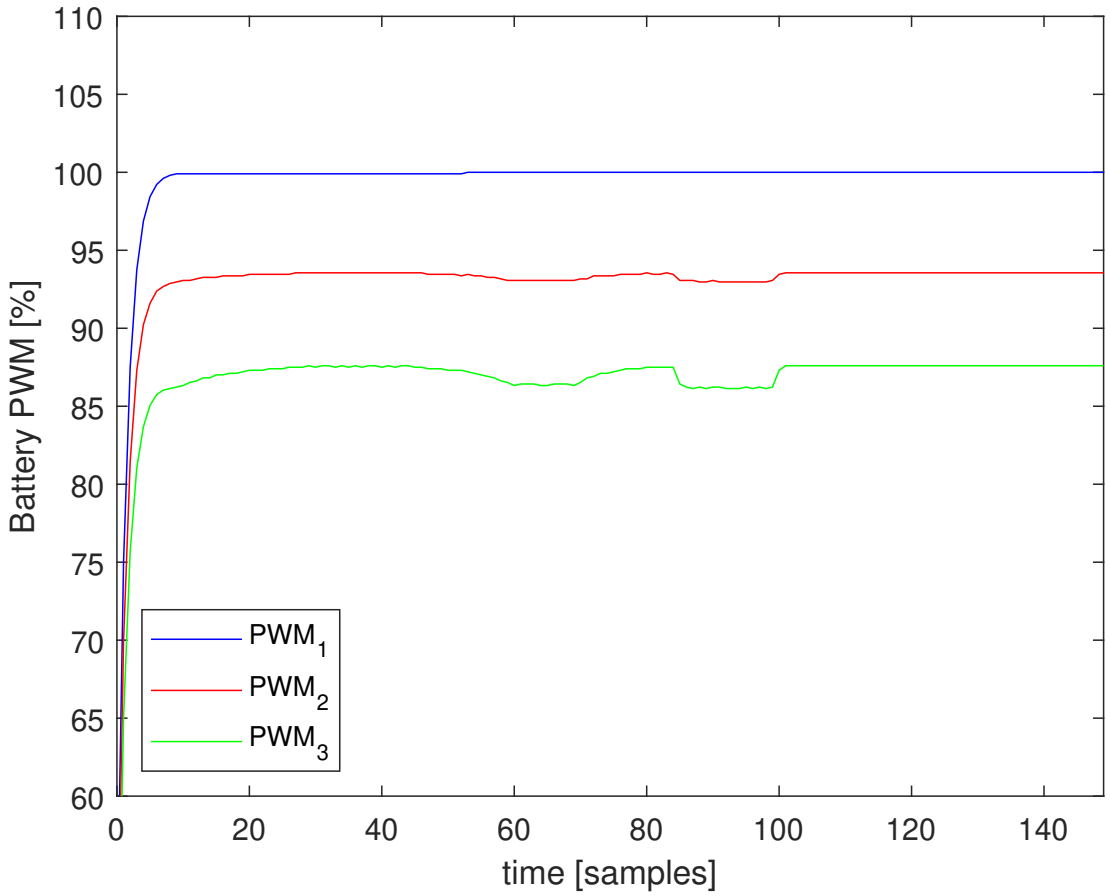

Figure 4.5: Computed scheduled PWM values PWM_j as a function of time for each battery module $j = 1,2,3$ to allow for the balanced battery module currents depicted in Fig. 4.4.

4.3 Distributed Control with Approximated f_j's

If the nonlinear PWM function $f_j(\cdot)$ of each module is approximated by $V^j_{PWM} = 0.5V^j_{OCV} + 0.5\frac{V^j_{OCV}}{100}PWM_j$, a DC offset appears in the current balancing, as can be seen in Fig. 4.6.

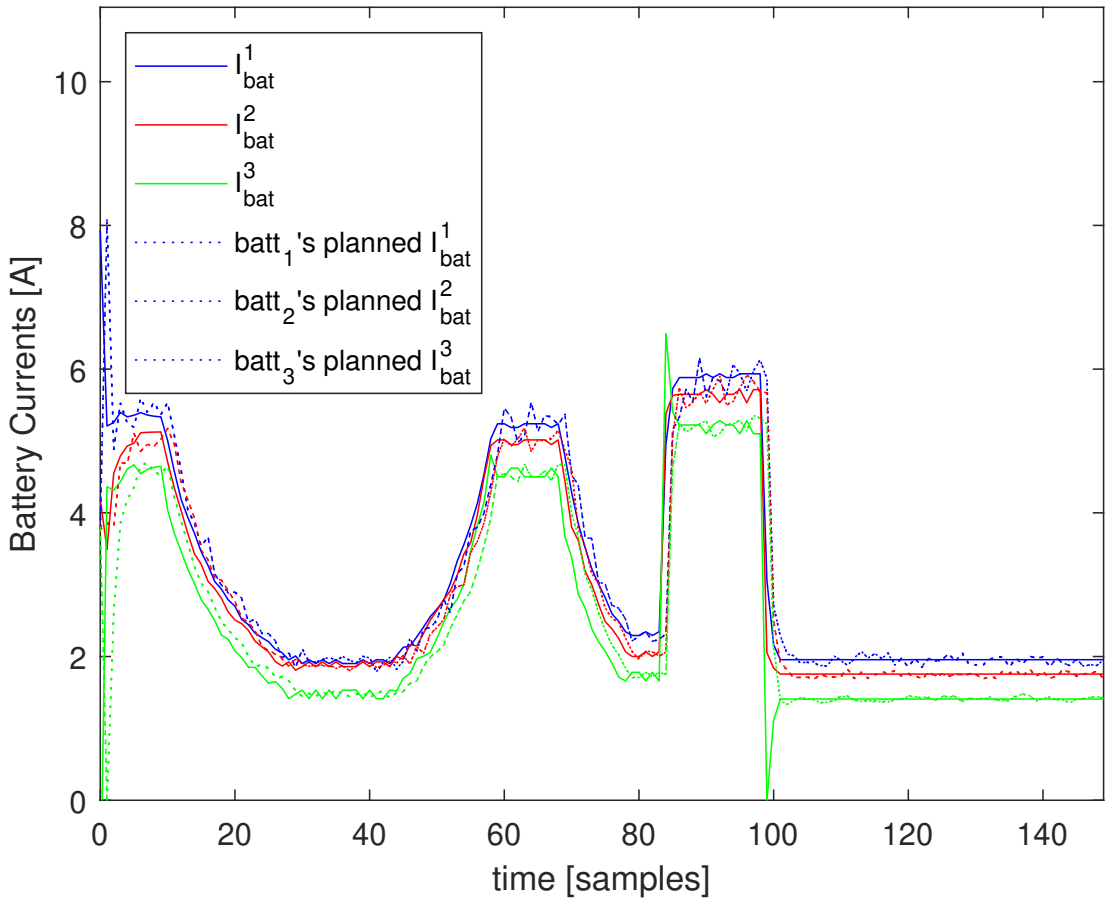

Figure 4.6: Measured battery module currents I^j_{bat} as a function of time in each battery module $j = 1,2,3$ due to the load variations depicted in Fig. 4.3 and with the nonlinear PWM function $f_j(\cdot)$ approximated by a fixed linear function.

Although the DC offset in Fig. 4.6 may be acceptable, further improvements can be gained by adding a tunable parameter to the approximation for the "weakest" battery module k, $V^k_{PWM} = 0.5V^k_{OCV} + \alpha\frac{V^k_{OCV}}{100}PWM_k$, and by allowing a centralized algorithm to intermittently send information on the current through the weakest module, which is used to tune α locally. By tuning the nonlinear approximation f_k, each module tunes its output to the same reference

module (the "weakest" module). Thus the PWM inputs found not only work better with the "weakest" module but also with each other, providing a simple way for the modules to account for mis-knowledge of system parameters. The improved results can be seen in Fig. 4.7.

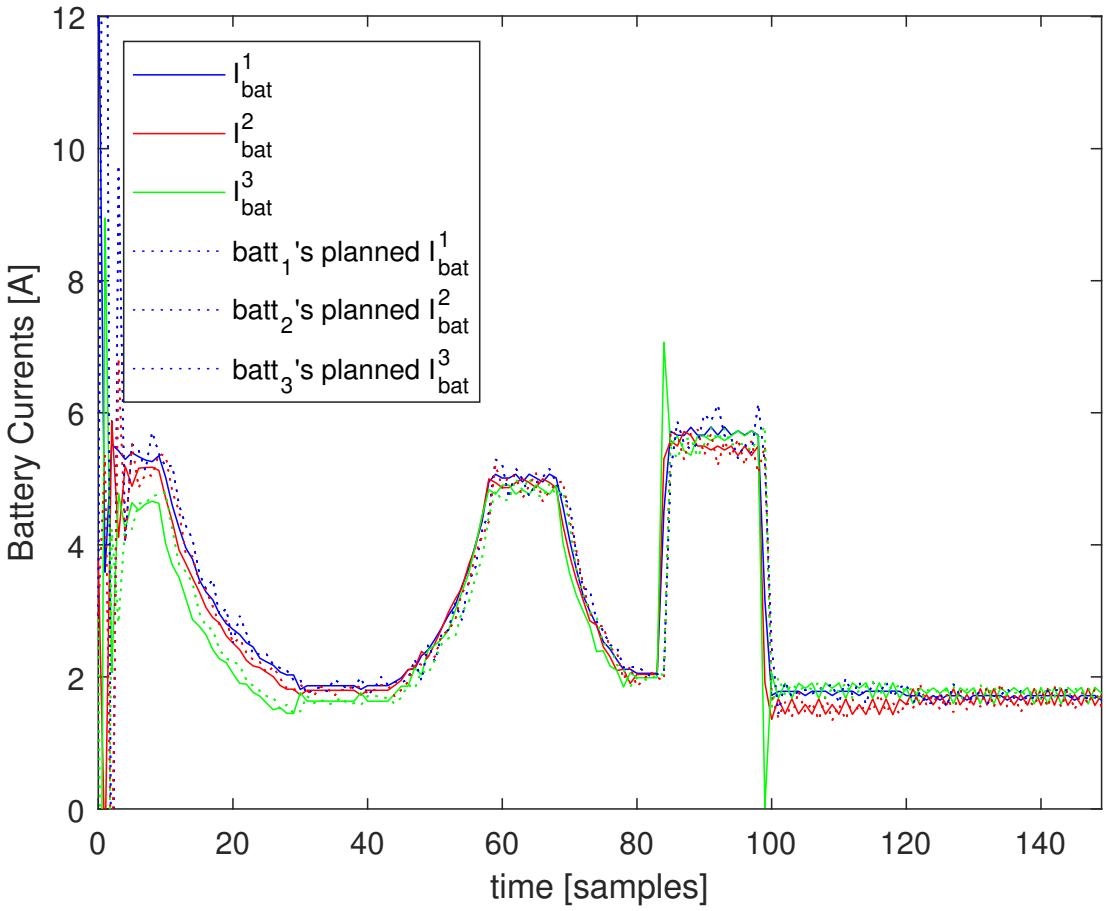

Figure 4.7: Module currents I^j_{bat} as a function of time due to the load variations in Fig. 4.3, with the linear model f_k tuned using centralized updates every thirty samples. Note that the benefits of the tuning persist even as the load changes.

4.4 Control with Mis-knowledge of Impedances

If the line impedances and modules impedances in the circuit are allowed to deviate from those provided to the modules for their distributed computation, more error is introduced into the computation of optimal control inputs. The precise effect will depend on how the resistances are allowed to change, but a representative example can be seen in Fig. 4.8, in which the modules compute their inputs assuming the resistance values shown in Table 4.3.

Table 4.3: Numerical values in Ohms for assumed battery module impedance $\hat{R}_j$, $j = 1,2,3$ and line impedance $\hat{R}_{ij}$ used in the computation of control inputs. The true resistance values can be seen in Table 4.1.

$\hat{R}_1$	$\hat{R}_2$	$\hat{R}_3$	$\hat{R}_{12}$	$\hat{R}_{23}$
0.50	0.48	0.41	0.05	0.07

The current balancing with random mis-knowledge of the impedances isn't much worse than when impedances are exactly known. The worst case for mis-knowledge of impedances seems to be when all the impedances are assumed to be higher than they actually are. In this case, the stronger batteries command a high PWM, and begin to overpower the weaker batteries. An example of this edge-case, in which all control inputs are computed assuming impedance values 0.05 Ohms higher than the true values in Table 4.1 can be seen in Fig. 4.9. An example of this edge-case and with modules using a tunable approximation with centralized updates can be seen in Fig. 4.10.

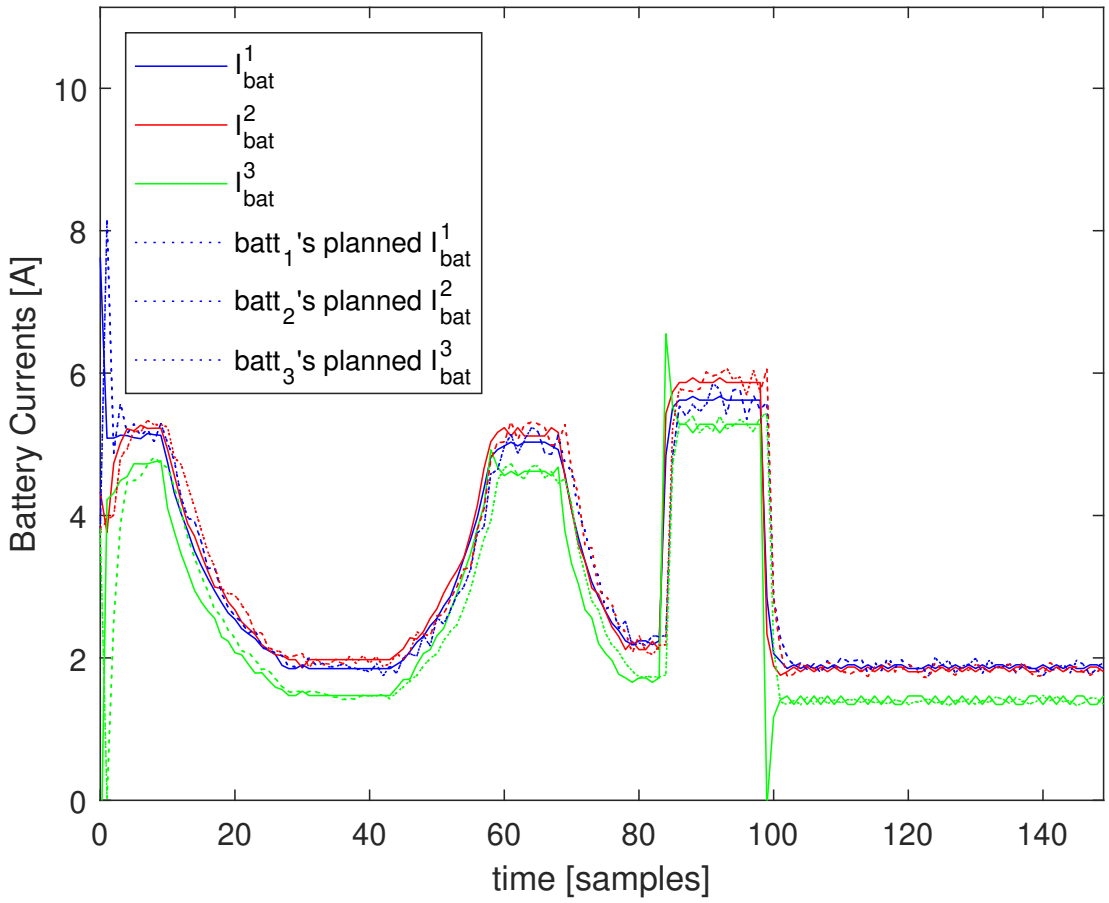

Figure 4.8: Module currents I^j_{bat} with load variations depicted in Fig. 4.3, with the nonlinear PWM function $f_j(\cdot)$ approximated by a fixed linear function, and the circuit impedances approximated by those in Table 4.3.

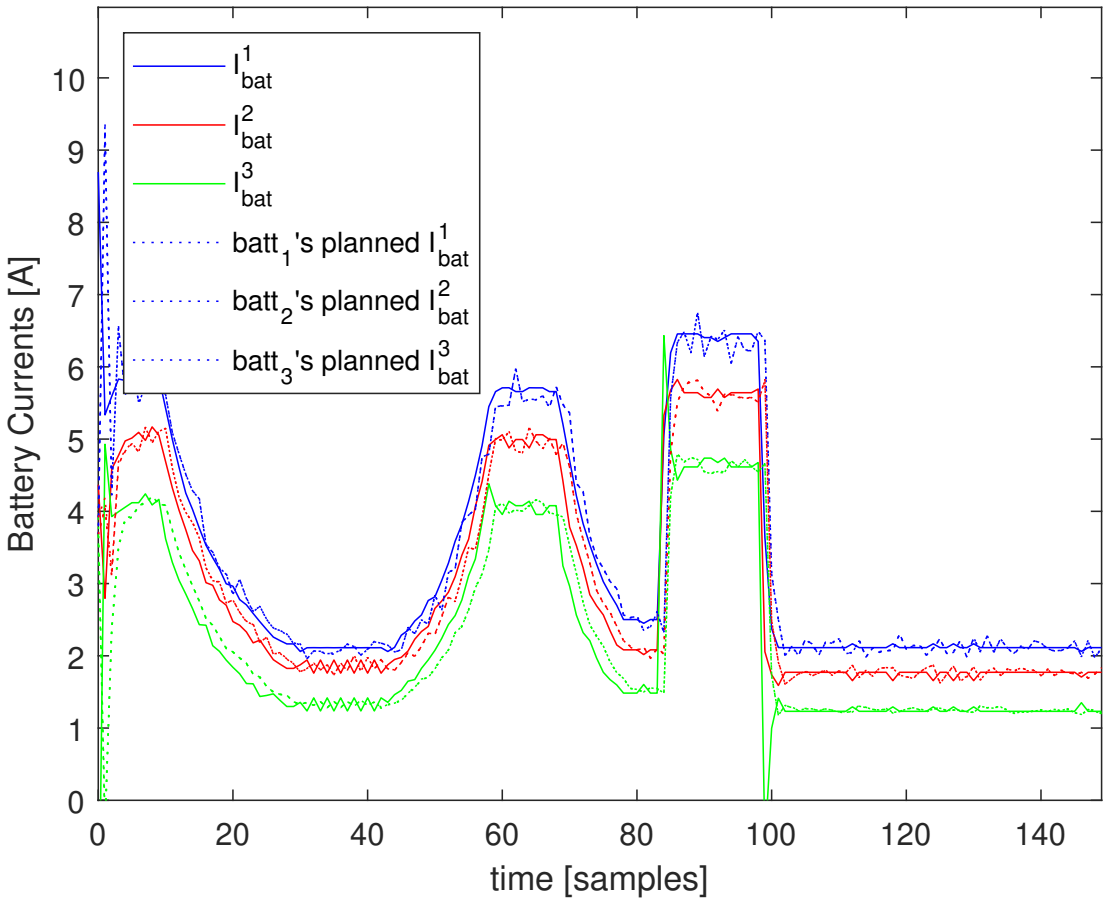

Figure 4.9: Module currents I^j_{bat} with load variations in Fig. 4.3, nonlinear $f_j(\cdot)$ approximated by a fixed linear function, and control inputs calculated using impedance 0.05 Ohms higher than the true values shown in Table 4.1.

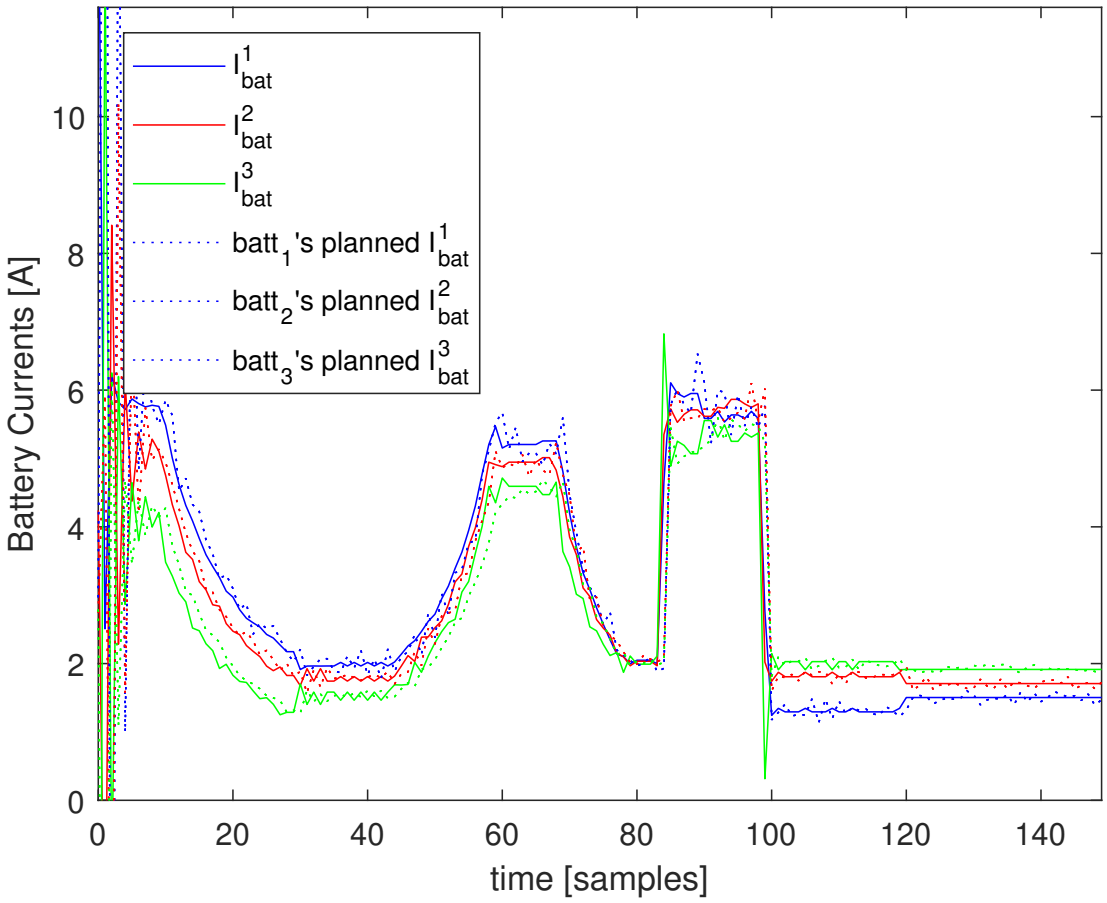

Figure 4.10: Module currents I^j_{bat} with load variations in Fig. 4.3, linear f_k tuned using centralized updates every thirty samples, and inputs calculated using impedances 0.05 Ohms higher than the true values shown in Table 4.1.

4.5 Testing on Hardware

In order to evaluate the proposed control algorithm on hardware, we obtained a prototyping power network, an experimental power network, as well as a load board capable of load scheduling. While these systems are ready for implementation of the control algorithm, we have been unable to complete a test on hardware because of a slow start in translating the matlab code to C and restricted lab access due to Covid-19. Nonetheless, we will describe our experimental setup here in the hopes that it aids future work.

In the prototyping network, proxy battery modules consist of an amperage limited power source in series with a potentiometer and a buck-regulator, which is controlled by an arduino. The potentiometer simulates the internal resistance of the battery module, and the arduino can be used to calculate the optimal control input for the module as outlined in this paper. The amperage limited power source ensures safe operation during the initial implementation on hardware, and allows the user to modify the open circuit voltage of the proxy modules. Three proxy battery modules constructed in this fashion and are connected in parallel using potentiometers, allowing the user to specify line impedances. The result of these connections is the prototyping power network.

In the experimental network, battery modules are constructed by connecting a row of Li-Ion cells and a potentiometer in series with a fuse and a buck regulator controlled by an arduino. As in the prototyping network, we can modify the internal resistance of the battery module through use of the potentiometer, and can use the arduino to calculate the optimal control inputs. The fuse in the battery modules ensures safety during implementation, but makes the experimental power network less suited for rapid iterations than the prototyping power network. Three of these battery modules are connected in parallel using potentiometers. This power network is meant to serve as a close stand-in for a final implementation using second life EV batteries.

The load board consists of multiple resistors connected together using transistors controlled by an arduino. By switching different transistors on and off, resistors can be excluded from the circuit, placed in series with each other, or placed in parallel. In this way, the total resistance of the load board can be scheduled during a test of the control algorithm.

5 CONCLUSIONS

5.1 Conclusions of Present Work

The control approach via optimization presented in this paper enables second-life battery modules in a parallel network to provide balanced currents to a varying load. The decentralized nature of the algorithm enables rapid updates of control inputs without depending on communication bandwidth between modules. The approach is illustrated on a representative three battery network model with buck regulated battery modules that have discrepancies between battery impedance, battery open circuit voltage and uncertainty on the mapping of Pulse Width Modulation of the buck regulator. The best performance is obtained with periodic updates on slowly varying operational parameters of the batteries and the impedance of their network interconnection combined with fast distributed control within the battery modules.

5.2 Recommendations for Implementation

In implementation it may be practical to approximate the nonlinear PWM functions $f_j(\cdot)$ with a static function as in Fig. 4.6, without tuning of the approximation such as in (4.2). If a static function is used, the currents will be slightly mis-balanced, as in Fig. 4.6. The mis-balance is caused by differences between the expected voltages provided by each battery and the actual voltages provided. If the difference between expected and actual voltages is larger, then the

currents will in general be less balanced. Thus, if the approximation cannot be tuned on-line, it is important that the function used to approximate one of the nonlinear PWM functions $f_j(\cdot)$ is close to the true $f_j(\cdot)$ in the range where the module will operate. It is expected that most modules will be operated at high PWM, where buck regulators behave mostly linearly. Thus it is acceptable to characterize the chosen buck regulators and then pick an approximating function which is close at high PWM. However, if a battery module is expected to operate at lower PWM, the error introduced by the nonlinearity can become dangerous. For the approximation displayed in Fig. 4.2 and with the load trajectory prescribed in Fig. 4.1, we have found that currents are balanced within 2 Amps of each other as long as module PWMs remain above 75%. In summary, a static approximation of $f_j(\cdot)$ is safer if module PWMs remain above a specified cutoff. The cutoff above which balancing is "acceptable" will depend on the error between $f_j(\cdot)$ and its approximation, and on the acceptable error in the current balancing.

This thesis is a reprint of the material as it is to appear in Proceedings of the IFAC 2020 World Congress under the title "Distributed Control of Second Life Batteries in a Parallel Connected Network" by Michael Bagherpour and Raymond A. de Callafon. The thesis author was primary author of this paper.

A Data from Simulated Scenarios

A.1 Noisy, Truncated Data

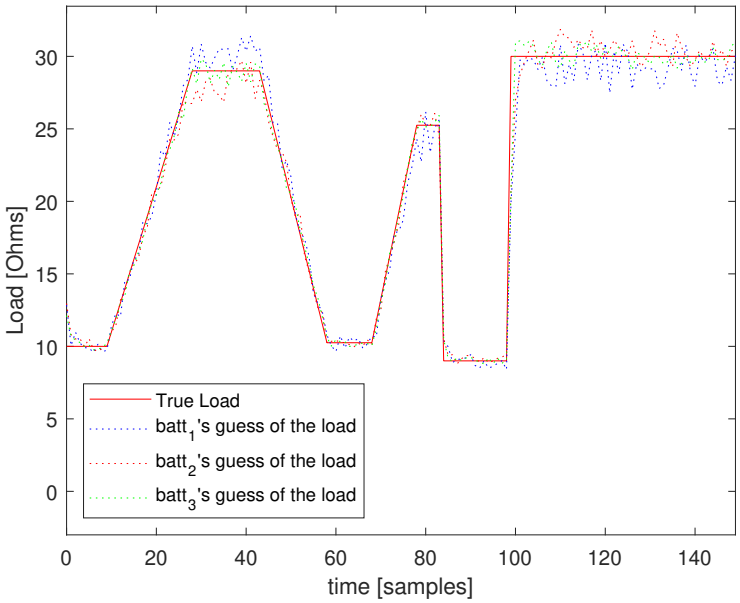

Figure A.1: Each module's estimation of the load under noisy readings and data truncation, but perfect knowledge of the nonlinear *PWM* mapping function

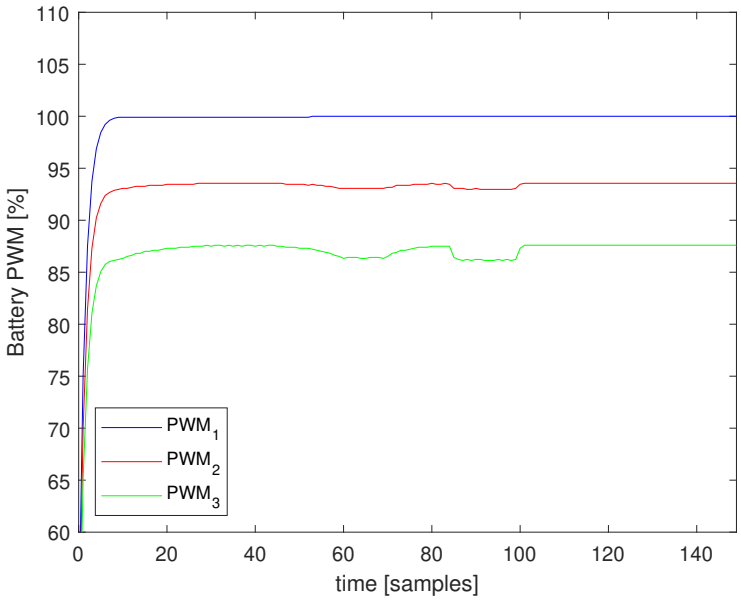

Figure A.2: Each module's PWM commands under noisy readings and data truncation, but perfect knowledge of the nonlinear *PWM* mapping function

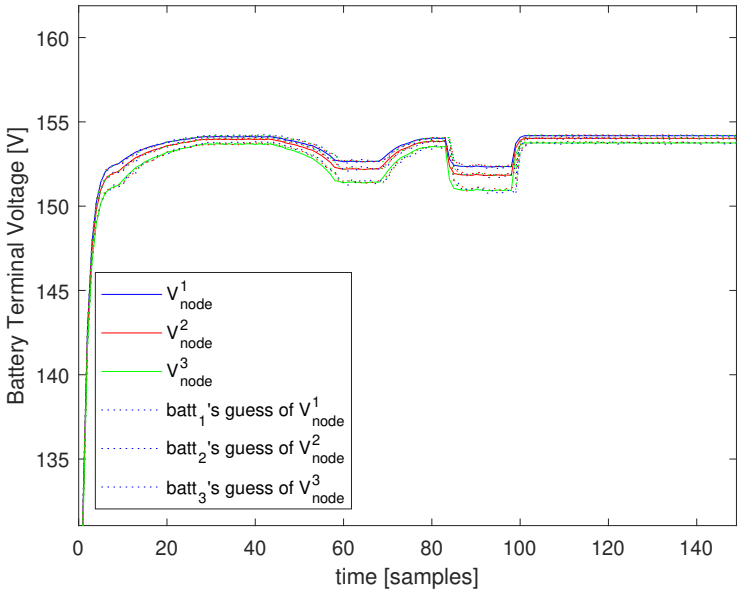

Figure A.3: Each module's terminal voltage under noisy readings and data truncation, but perfect knowledge of the nonlinear *PWM* mapping function

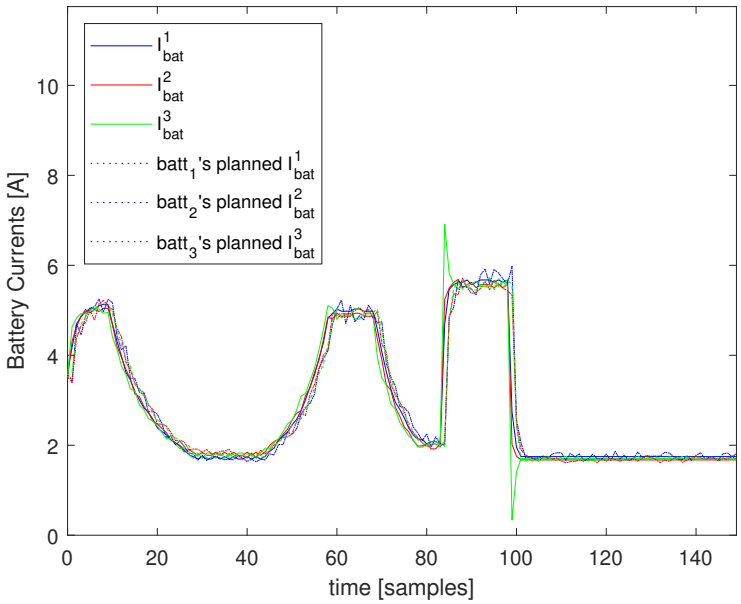

Figure A.4: The current through each module under noisy readings and data truncation, but perfect knowledge of the nonlinear PWM mapping function

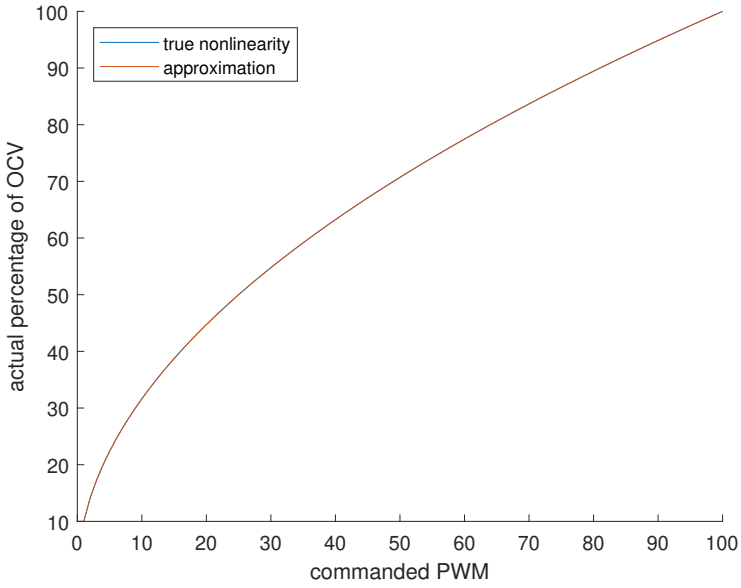

Figure A.5: The nonlinear PWM mapping function, which is exactly known by each module in the case with noisy readings and data truncation, but perfect knowledge of the nonlinear PWM mapping function

A.2 Noisy Truncated Data, and Approximated Nonlinearity

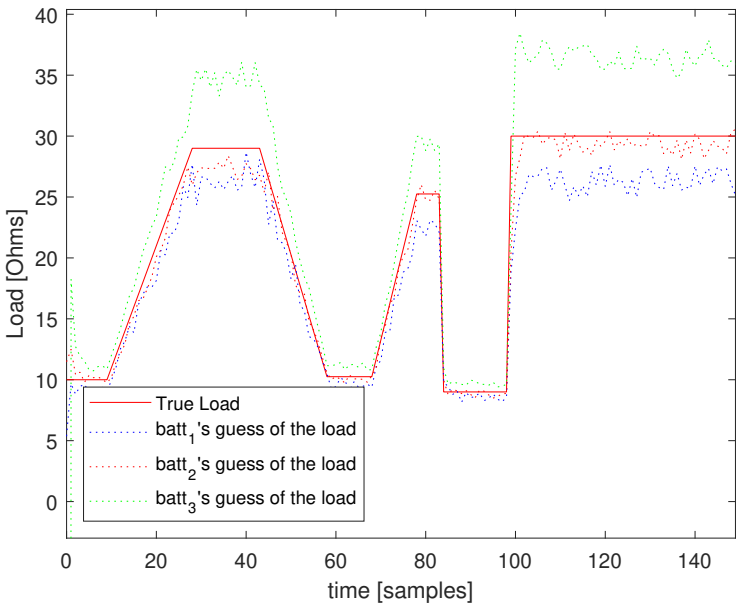

Figure A.6: Each module's estimation of the load under noisy readings, data truncation, and imperfect knowledge of the nonlinear PWM mapping function

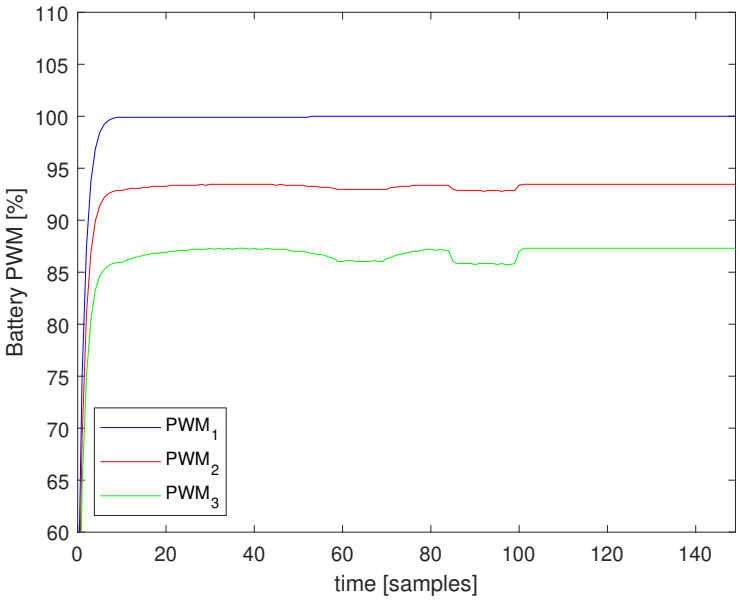

Figure A.7: Each module's PWM commands under noisy readings, data truncation, and imperfect knowledge of the nonlinear *PWM* mapping function

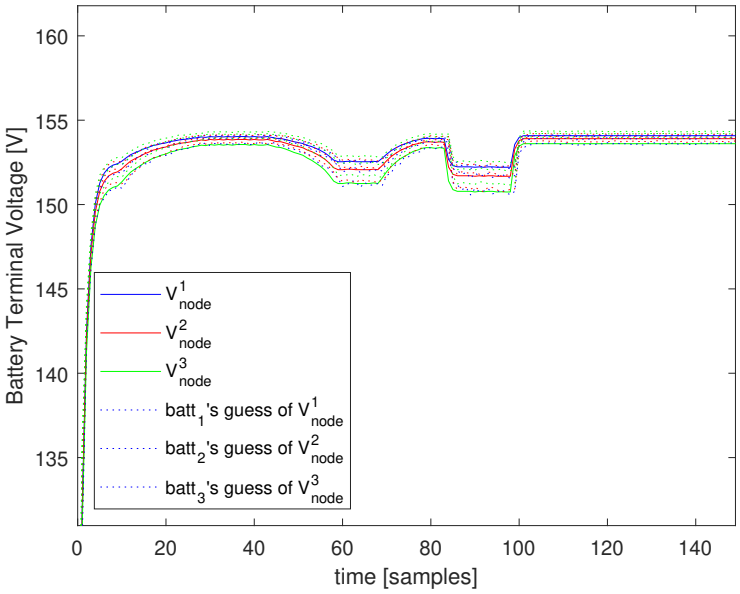

Figure A.8: Each module's terminal voltage under noisy readings, data truncation, and imperfect knowledge of the nonlinear *PWM* mapping function

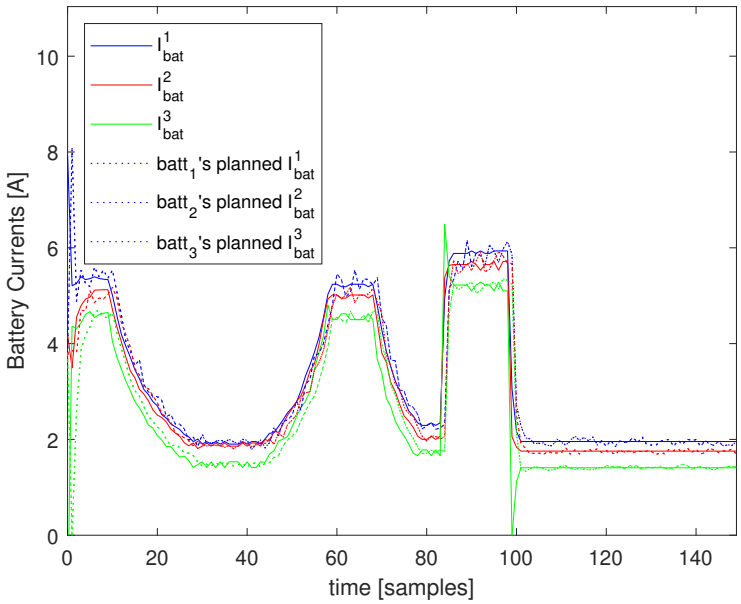

Figure A.9: The current through each module under noisy readings, data truncation, and imperfect knowledge of the nonlinear *PWM* mapping function

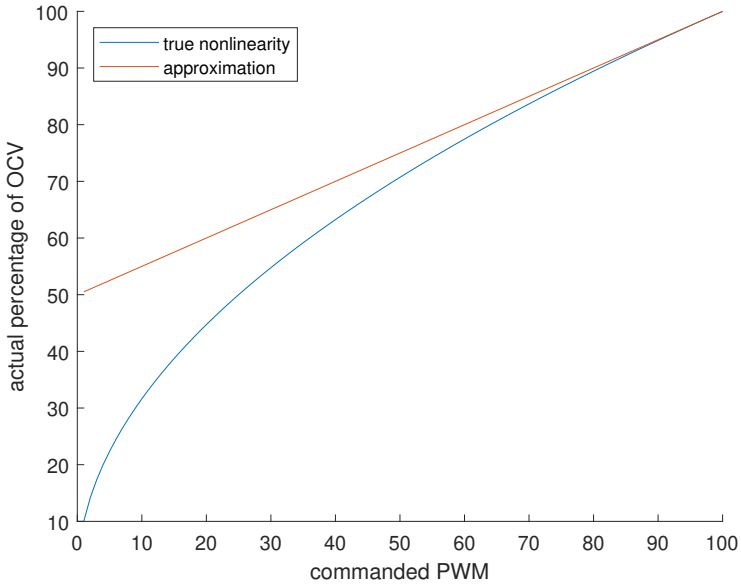

Figure A.10: The nonlinear *PWM* mapping function and its approximation, in the case with noisy readings, data truncation, and imperfect knowledge of the nonlinear *PWM* mapping function

A.3 Noisy Truncated Data, Mis-knowledge of Nonlinearity, and Central Updates

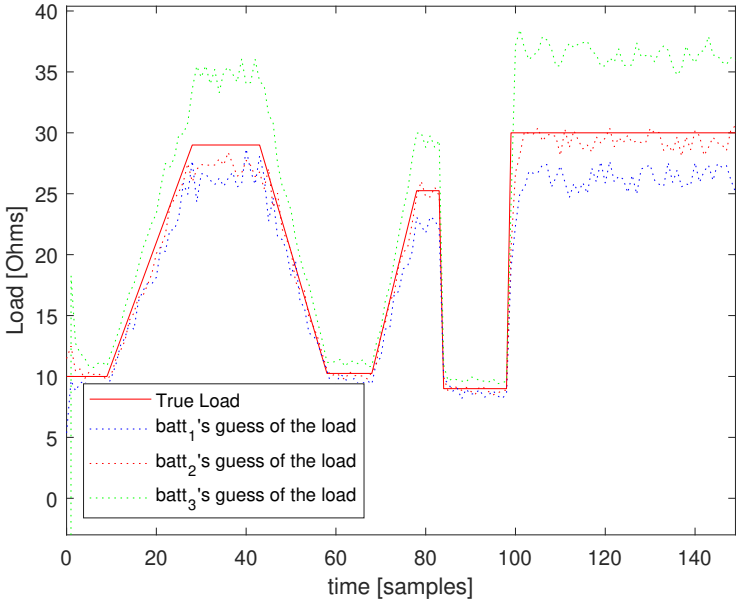

Figure A.11: Each module's estimation of the load under noisy readings, data truncation, and using updates from a central processor every 30 samples to tune the approximation of the nonlinear PWM mapping function

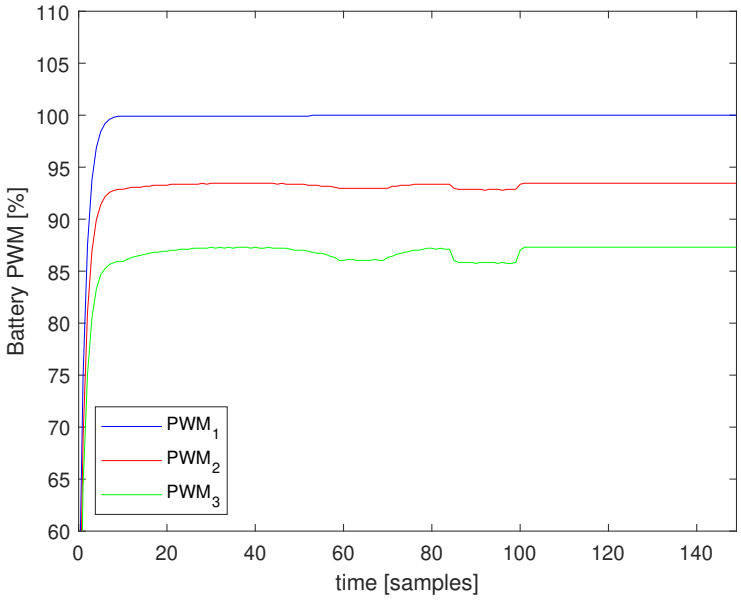

Figure A.12: Each module's PWM commands under noisy readings, data truncation, and using updates from a central processor every 30 samples to tune the approximation of the nonlinear PWM mapping function

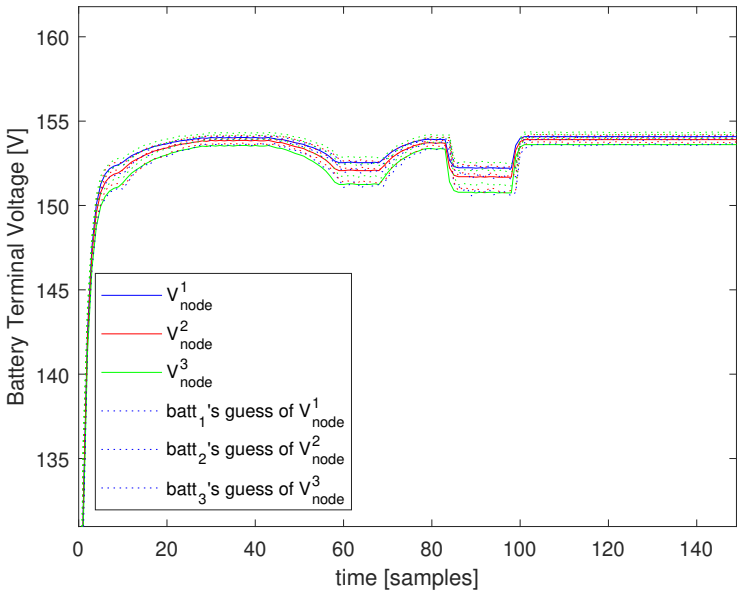

Figure A.13: Each module's terminal voltage under noisy readings, data truncation, and using updates from a central processor every 30 samples to tune the approximation of the nonlinear PWM mapping function

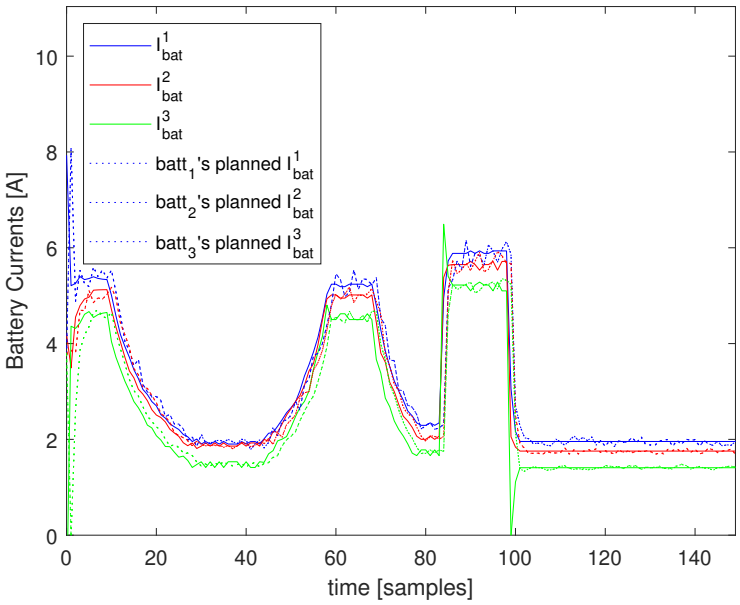

Figure A.14: The current through each module under noisy readings, data truncation, and using updates from a central processor every 30 samples to tune the approximation of the nonlinear PWM mapping function

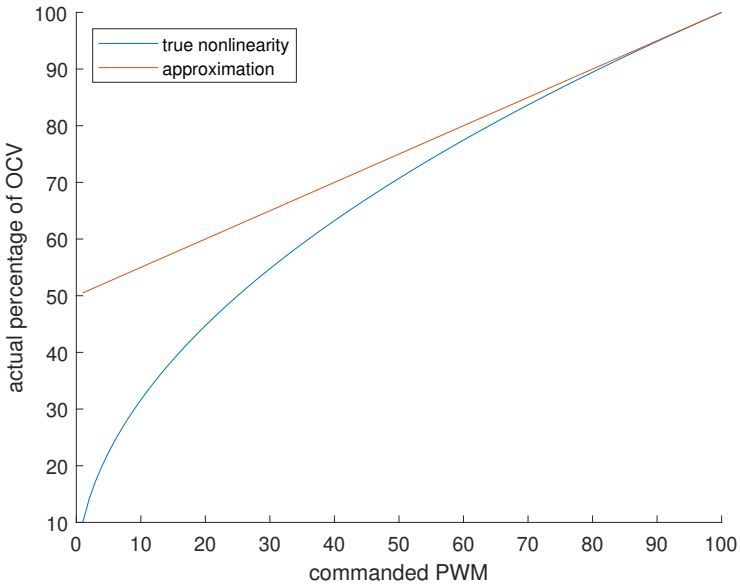

Figure A.15: The nonlinear PWM mapping function and its initial approximation in the case with noisy readings, data truncation, and using updates from a central processor every 30 samples to tune the approximation of $f_j(\cdot)$

A.4 Noisy Truncated Data, Mis-knowledge of Nonlinearity, and Random Mis-knowledge of Impedances

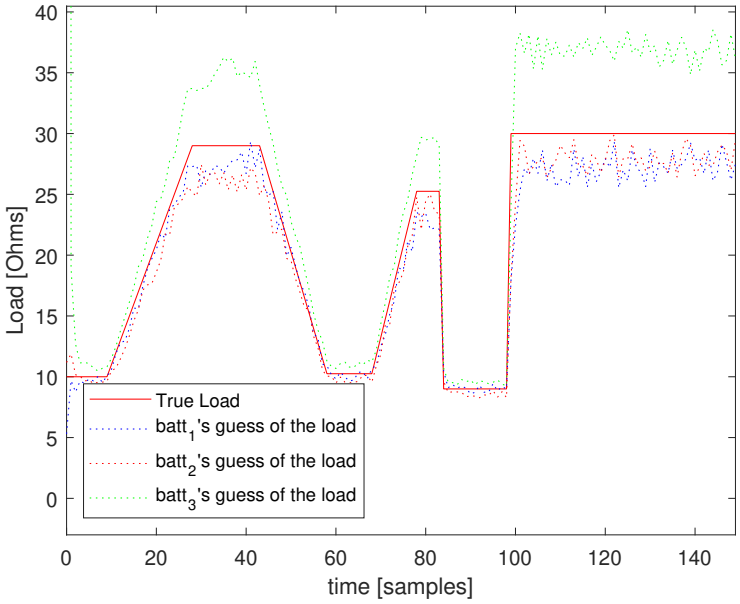

Figure A.16: Each module's estimation of the load under noisy readings, data truncation, imperfect knowledge of the nonlinear *PWM* mapping function, and with control inputs calculated using impedance values described in Table 4.3

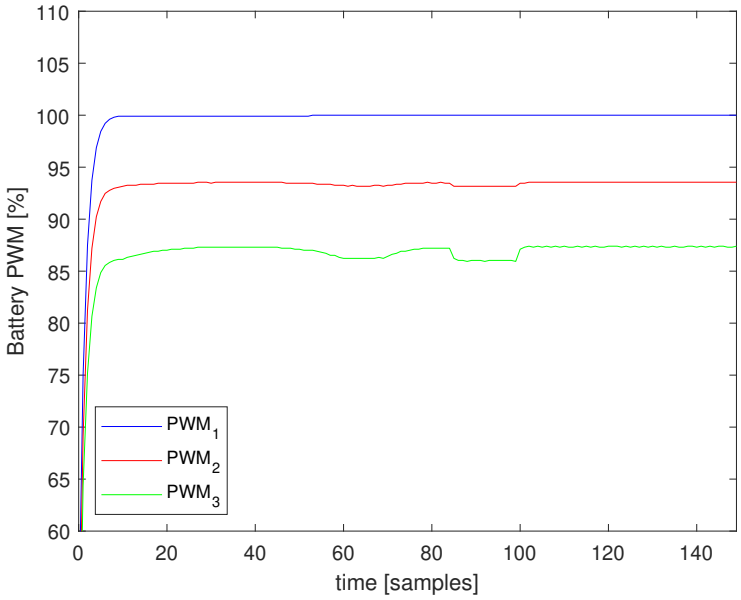

Figure A.17: Each module's PWM commands under noisy readings, data truncation, imperfect knowledge of the nonlinear *PWM* mapping function, and with control inputs calculated using impedance values described in Table 4.3

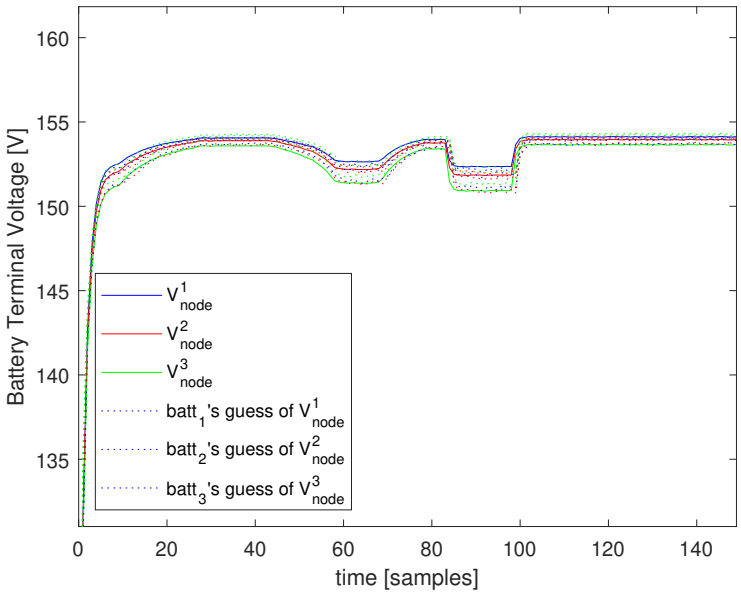

Figure A.18: Each module's terminal voltage under noisy readings, data truncation, imperfect knowledge of the nonlinear *PWM* mapping function, and with control inputs calculated using impedance values described in Table 4.3

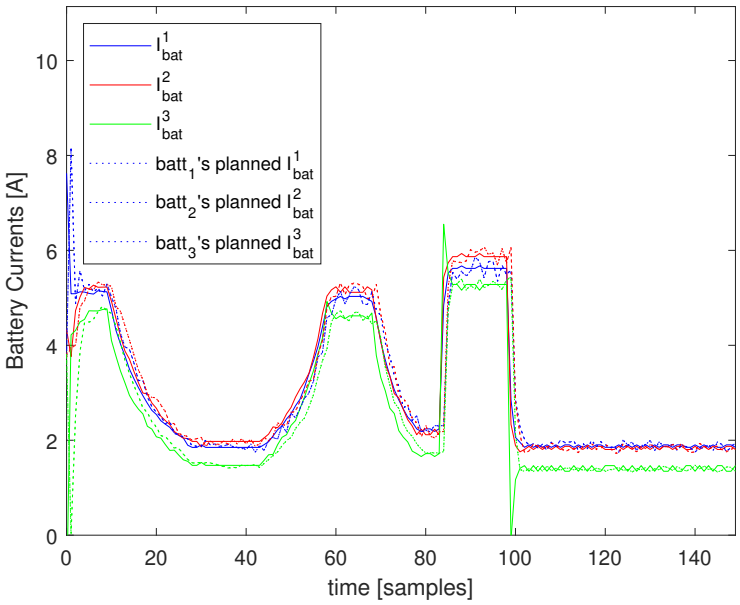

Figure A.19: The current through each module under noisy readings, data truncation, imperfect knowledge of the nonlinear *PWM* mapping function, and with control inputs calculated using impedance values described in Table 4.3

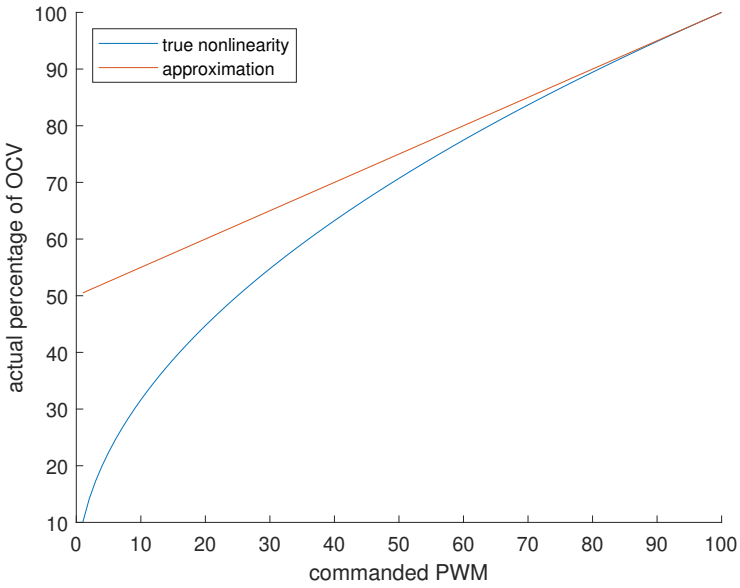

Figure A.20: The nonlinear *PWM* mapping function and its approximation in the case with noisy readings, data truncation, and with control inputs calculated using impedance values described in Table 4.3

A.5 Noisy Truncated Data, Mis-knowledge of Nonlinearity, and Adversarial Mis-knowledge of Impedances

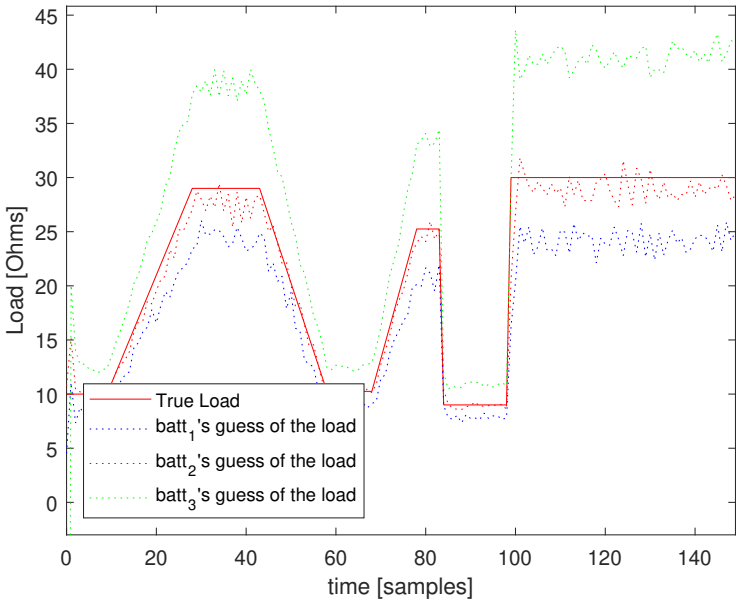

Figure A.21: Each module's estimation of the load under noisy readings, data truncation, imperfect knowledge of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

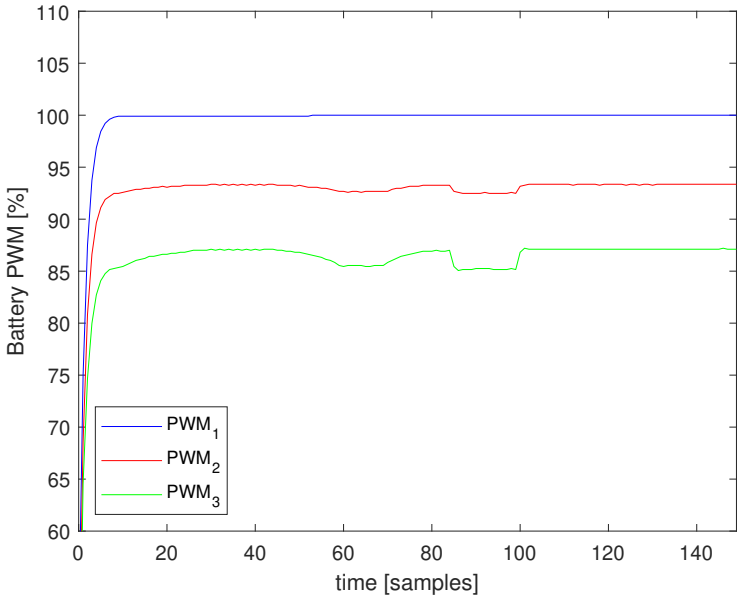

Figure A.22: Each module's PWM commands under noisy readings, data truncation, imperfect knowledge of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

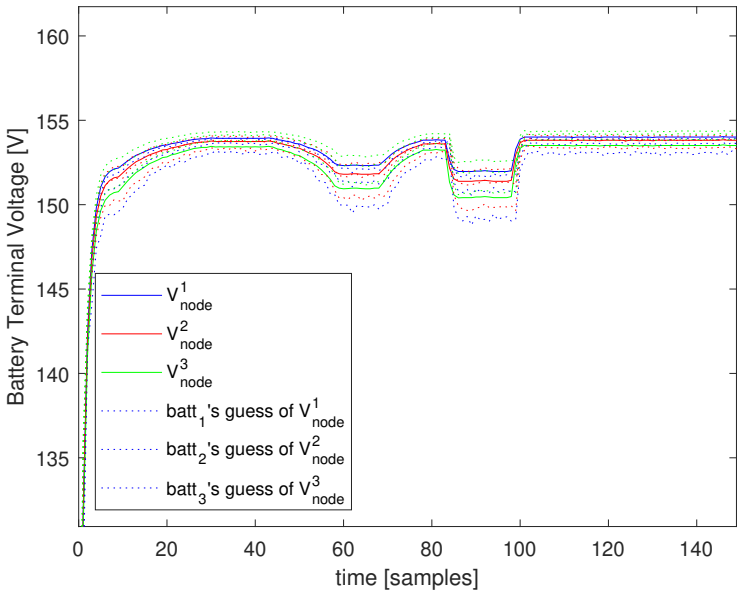

Figure A.23: Each module's terminal voltage under noisy readings, data truncation, imperfect knowledge of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

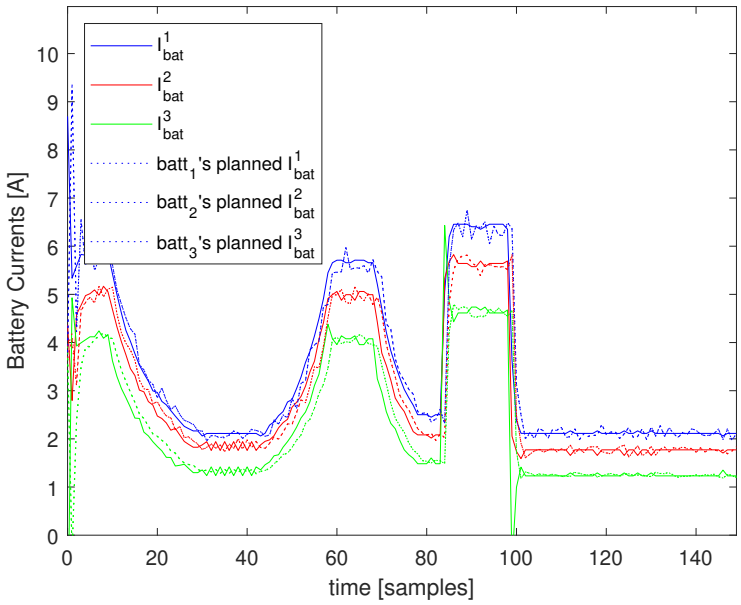

Figure A.24: The current through each module under noisy readings, data truncation, imperfect knowledge of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

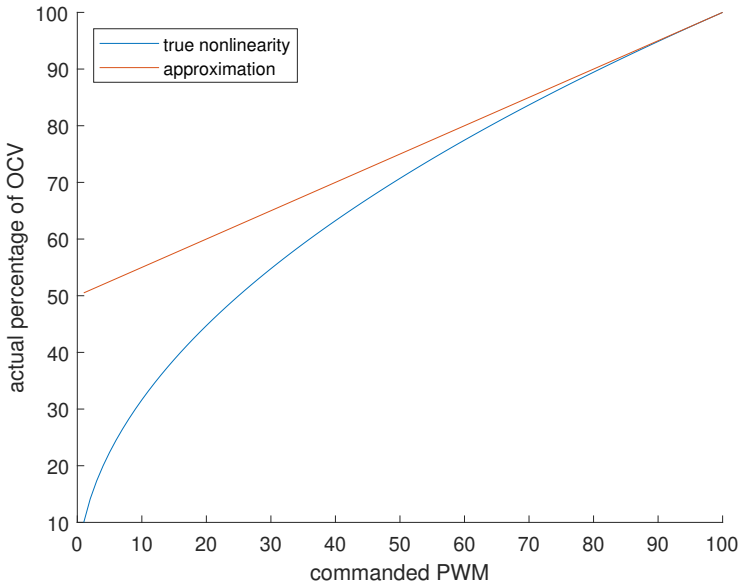

Figure A.25: The *PWM* mapping function and its approximation in the case with noisy readings, data truncation, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

A.6 Noisy Truncated Data, Tunable Approximation of Non-linearity, and Adversarial Mis-knowledge of Impedances

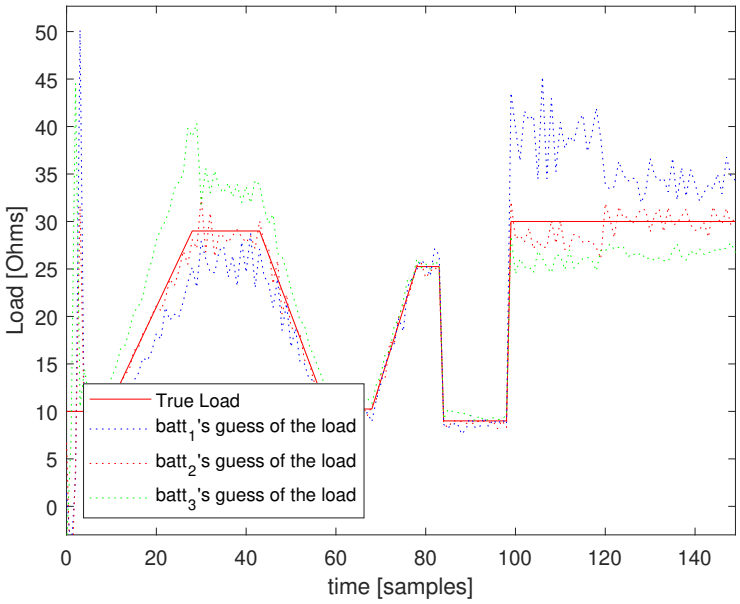

Figure A.26: Each module's estimation of the load under noisy readings, data truncation, a tunable approximation of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

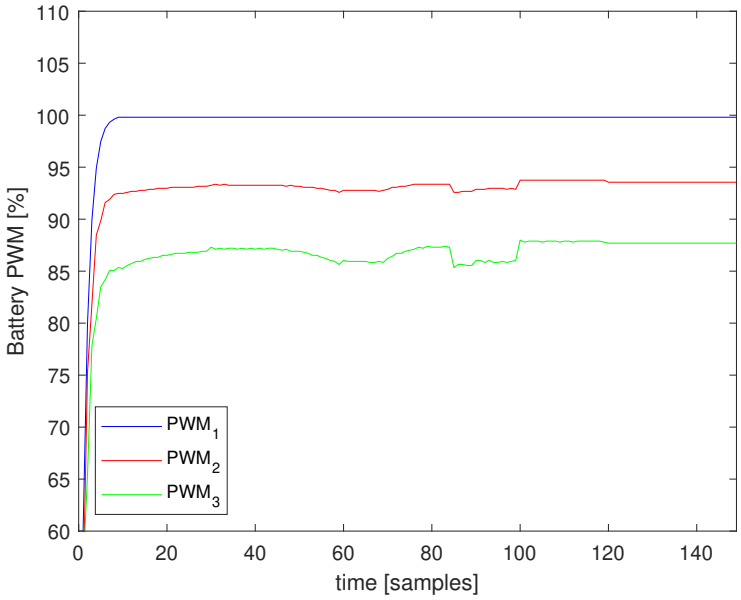

Figure A.27: Each module's PWM commands under noisy readings, data truncation,a tunable approximation of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

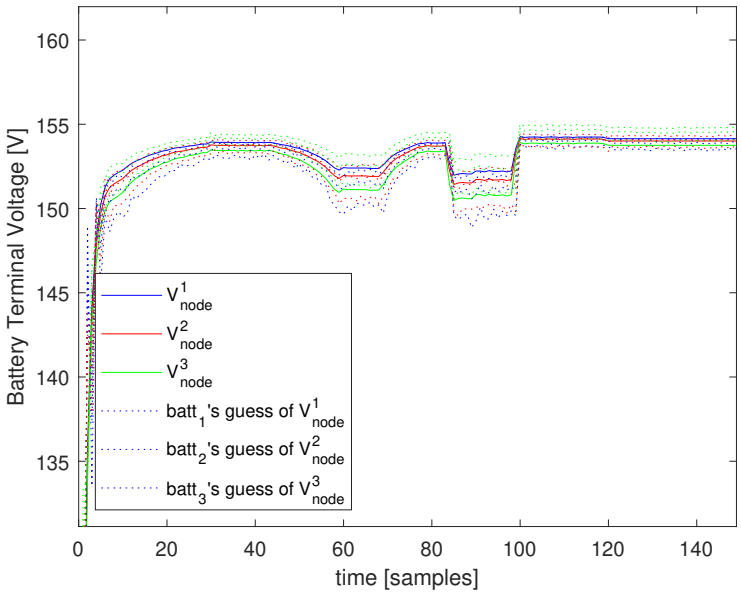

Figure A.28: Each module's terminal voltage under noisy readings, data truncation, a tunable approximation of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

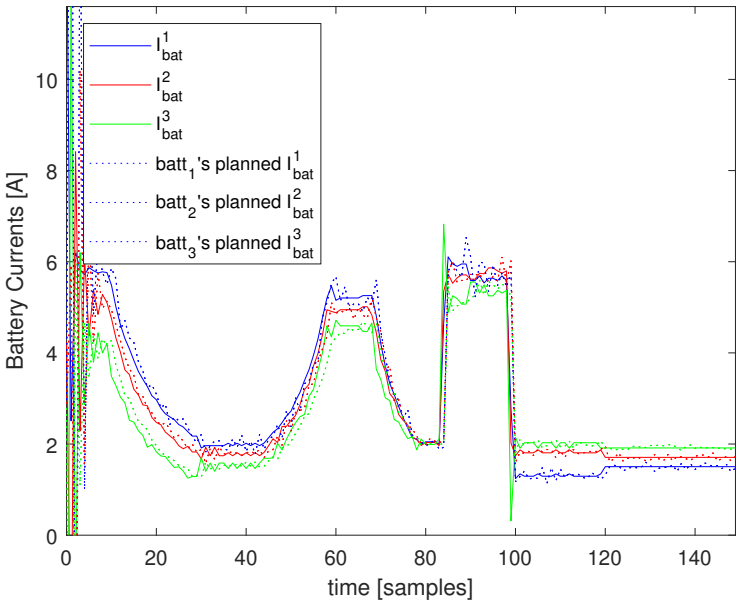

Figure A.29: The current through each module under noisy readings, data truncation, a tunable approximation of the *PWM* mapping function, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

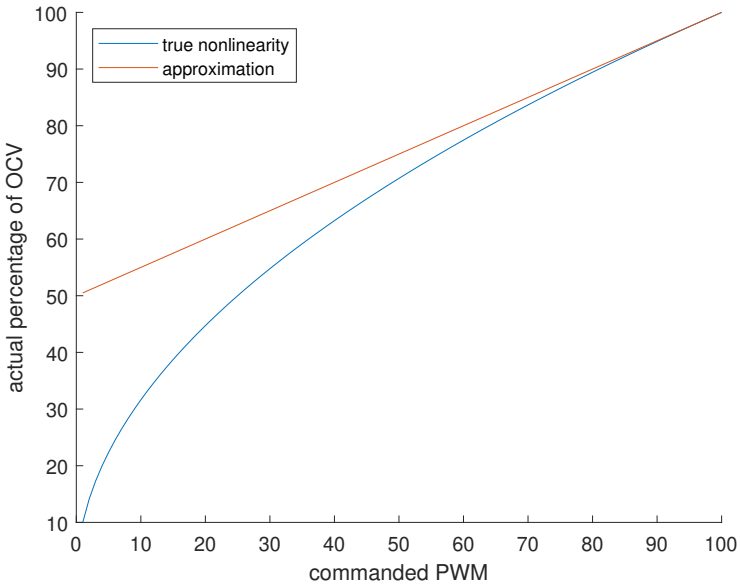

Figure A.30: The nonlinear *PWM* mapping function and its initial approximation in the case with noisy readings, data truncation, a tunable $f_k(\cdot)$, and inputs calculated using impedances 0.05 Ohms higher than the true values in Table 4.1

Bibliography

A. Almeida and P. Nunes. The role of second-life batteries on renewable based power systems. In *13th Conference on sustainable Development of Energy, Water and Environment Systems*, pages 1–15, 2018.

F. Altaf, L. Johannesson, and B. Egardt. Simultaneous thermal and state-of-charge balancing of batteries: A review. In *Proc. IEEE Vehicle Power and Propulsion Conference*, pages 1—-7, 2014.

Lluc Canals Casals, B. Amante García, and Camille Canalc. Second life batteries lifespan: Rest of useful life and environmental analysis. *Journal of Environmental Management*, 232:354–363, 2019. doi: doi.org/10.1016/j.jenvman.2018.11.046.

John W. A. Catton, Sean B. Walker, Paul McInnis, Michael Fowler, Roydon A. Fraser, Steven B. Young, and Ben Gaffney. Design and analysis of the use of re-purposed electric vehicle batteries for stationary energy storage in canada. *Batteries*, 5(14):1–19, 2019. doi: 10.3390/batteries5010014.

Matús Danko, Juraj Adamec, Michal Taraba, and Peter Drgona. Overview of batteries state of charge estimation methods. *Transportation Research Procedia*, 40:186–192, 2019. doi: doi.org/10.1016/j.trpro.2019.07.029.

Ethan N. Elkind. Reuse and repower: How to save money and clean the grid with second-life

electric vehicle batteries. Technical report, UCLA School of Law's EICCE, UC Berkeley School of Law's CLEE, 2014.

Shyh-Chin Huang, Kuo-Hsin Tseng, Jin-Wei Liang, and Chung-Liang Chang Michael G. Pecht. An online SOC and SOH estimation model for lithium-ion batteries. *Energies*, 10(512):1–18, 2017. doi: 10.3390/en10040512.

Yunfeng Jiang, Louis J. Shrinkle, and Raymond A. de Callafon. Autonomous demand-side current scheduling of parallel buck regulated battery modules. *Energies*, 12:1–20, 2019. doi: doi:10.3390/en12110000.

Na Jiao and Steve Evans. Business models for sustainability: The case of second-life electric vehicle batteries. *Procedia CIRP*, 40:250–255, 2016. doi: doi.org/10.1016/j.procir.2016.01.114.

Manoj Mathew, Stefan Janhunen, Mahir Rashid, Frank Long, and Michael Fowler. Comparative analysis of lithium-ion battery resistance estimation techniques for battery management systems. *Energies*, 11(1490):1–15, 2018. doi: doi:10.3390/en11061490.

Noshin Omar, Mohamed Abdel Monem, Yousef Firouz, Justin Salminen, Jelle Smekens, Omar Hegazy, Hamid Gualous, Grietus Mulder, Peter Van den Bossche, Thierry Coosemans, and Joeri Van Mierlo. Lithium iron phosphate based battery – assessment of the aging parameters and development of cycle life model. *Applied Energy*, 113:1575–1585, 2014. doi: 10.1016/j.apenergy.2013.09.003.

R. Sathre, C. D. Scown, O. Kavvada, and T. P. Hendrickson. Energy and climate effects of second-life use of electric vehicle batteries in california through 2050. *Journal of Power Sources*, 288:82—-91, 2015.

Hans-Georg Schweiger, Ossama Obeidi, Oliver Komesker, André Raschke, Michael Schiemann, Christian Zehner, Markus Gehnen, Michael Keller, and Peter Birke. Comparison of several

methods for determining the internal resistance of lithium ion cells. *Sensors*, 10(6):5604—-5625, 2010. doi: 10.3390/s100605604.

Xin Zhao, Raymond A. de Callafon, and Lou Shrinkle. Current scheduling for parallel buck regulated battery modules. In *IFAC Proceedings Volumes*, volume 47, pages 2112–2117, 2014.