Distributed Control of Second Life Batteries in a Parallel Connected Network

Michael Bagherpour * Raymond A. de Callafon **

* University of California San Diego, La Jolla, CA 92093-0411 USA
  (e-mail: mbagherp@eng.ucsd.edu).
** University of California San Diego, La Jolla, CA 92093-0411 USA
  (e-mail: callafon@ucsd.edu)

Abstract: Used batteries that are no longer fit for their original applications can be combined to form usable battery packs. Cells can be connected in series to form modules with higher voltage, and these modules can be connected in parallel to build up the capacity of the battery pack. Parallel connections may cause stray currents within the battery pack due to heterogeneous operational parameters of the modules, so the current output by each module must be controlled to eliminate this problem. We present an approach to control such a configuration by buck regulating the terminal voltage of each module. The novelty of the proposed control algorithm is to separate the control into two components: an occasional update to estimates of slow-varying system parameters, and a frequent update of control inputs to accommodate fluctuations in the load. The estimated system parameters are communicated infrequently by a central processor, while the modules individually calculate control inputs. The required communication between modules and the central processor is greatly reduced by distributing the calculation of control inputs, allowing the system to scale up efficiently even as the number of modules grows large.

Keywords: Energy storage; Networked systems; Decentralized control

1. INTRODUCTION

Re-purposed batteries of an electric vehicle (EV) are a possible economical solution to provide short-term storage of intermittent electric energy produced by renewable energy sources. The economic and environmental benefits of such second-life batteries is well recognized (Elkind, 2014; Sathre et al., 2015; Jiao and Evans, 2016), along with the potential of large scale energy storage capabilities for Lithium-Ion (Li-Ion) batteries (Almeida and Nunes, 2018). Simply recycling the materials of a Li-Ion EV battery when its typical 80% of OEM storage capacity has been reached 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). Most of the applications focus 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). Although SoC/SoH monitoring and balancing control improves the long-term reliability of a battery pack, the internal resistance or impedance of a Li-Ion battery is a key parameter in determining power output and energy efficiency (Schweiger et al., 2010).

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 increase energy storage capabilities. This requires regulation of the terminal voltage to avoid stray currents between battery modules, see e.g. Jiang et al. (2019). 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.

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 current by buck regulating the terminal voltage of each battery, without requiring high-bandwidth communication to a central processor. This is achieved by distributing the centralized control algorithm of (Jiang et al., 2019) to each battery module. The proposed control algorithm enables frequent updates of control inputs to accommodate a changing load, despite the low communication bandwidth between battery modules. The control algorithm takes advantage of fast sampling and control within each battery module, while 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

\[ V_j = V_{PWM}^j - R_{PWM}^j \cdot I_{PWM}^j, \]

where \( V_{PWM}^j \) is the OCV of module j and \( PWM \cdot M_j \) 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 \( f_j(V_{OCV}^j,100) = V_{PWM}^j \). 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 \( I_{PWM}^j \) varies due to possibly rapid fluctuations in the load powered by the network. The overarching objective is to use inputs \( PWM \cdot M_j \) to control currents \( I_{PWM}^j \) 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 \( I_{PWM}^j \) 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 \( I_{PWM}^j \), while receiving central information of \( V_{OCV}^j \) and \( R_j \) of all the battery modules from a central BMS at a much lower update rate. The method through which each battery controls \( I_{PWM}^j \) via \( PWM \cdot M_j \) 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. 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 \( PWM \cdot M_j \) and can measure its own current \( I_{PWM}^j \) due to its input \( PWM \cdot M_j \). The open-circuit terminal voltage or no-load voltage \( V_{PWM}^j = f(V_{OCV}^j, PWM \cdot M_j) \) is not measurable, but can be calculated by module j if the OCV \( V_{OCV}^j \) and the PWM signal \( PWM \cdot M_j \) are.

\[
\begin{bmatrix}
I_{PWM}^1 \\
I_{PWM}^2 \\
I_{PWM}^3
\end{bmatrix}
= \begin{bmatrix}
R_1 & 0 & 0 \\
0 & R_2 & 0 \\
0 & 0 & R_3
\end{bmatrix}
\begin{bmatrix}
V_{PWM}^1 \\
V_{PWM}^2 \\
V_{PWM}^3
\end{bmatrix}
= \begin{bmatrix}
R_1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & R_3
\end{bmatrix}
\begin{bmatrix}
I_{PWM}^1 \\
I_{PWM}^2 \\
I_{PWM}^3
\end{bmatrix}
\]

The actual nodal voltages for the three battery module network of Fig. 1 are combined in \( V_{node} = [V_1 \ V_2 \ V_3]^T \).
3. BATTERY SCHEDULING

3.1 Problem Statement

The objective is to equalize the currents $I^j_{bat}$ 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

$$\max \beta, \text{ subject to } I^j_{bat} = \beta 1_{n \times 1}, 0 \leq PW M_j \leq 100$$ (3)

where $1_{n \times 1}$ 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 $I^j_{bat} = \beta$, $j = 1, 2, \ldots, n$ 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) and apply only their optimal input, $PWM_j$. The optimization in (3) 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, high bandwidth communication would be required between each battery module and the central processor in order to communicate sensor readings and the optimal control inputs, which adds 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 $y_i$ and $y_{ij}$, the function $f_j$ mapping $PWM_j$ and $OCV_j$ to $V^{PWM}_j$, 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 $V^{OCV}_j$ 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 $V_{load}$. 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

**Theorem 1.** Consider the admittance matrices $Y_{line}$, $Y_{bat}$, and $Y_{load}$ in (1) and (2). Then $V_{node}$ can be calculated with the following two equations:

$$V^j_{node} = V^{PWM}_j - I^j_{bat}R_j$$ (4)

$$V^{PWM}_j = [E(Y_{line} + Y_{bat})(I_n - J_j)I_{n,j}]^{-1} \times (EY_{bat}V^{PWM}_j - E(Y_{line} + Y_{bat})J_jV_{node})$$ (5)

with

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

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

**Proof.** From nodal analysis, the module currents are

$$I_{bat} = (Y_{line} + Y_{load})V_{node} = (Y_{total} - Y_{bat})V_{node}$$ (6)

$V^{PWM}_j$ is the voltage at the nodes plus the voltage drop across the internal impedance in each module: $V^{PWM}_j = V_{node} + Y_{bat}^{-1}I_{bat}$. Using (6) we can write this as

$$V^{PWM}_j = Y_{bat}^{-1}Y_{total}V_{node}$$ (7)

Moving unknowns to the right hand side we have

$$Y_{bat}V^{PWM}_j = Y_{total}V_{node}$$ (8)

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 (8) by defining

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

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

Premultiplying (8) by $E$, we have

$$EY_{bat}V^{PWM}_j = EY_{total}V_{node}$$

$$= (EY_{line} + EY_{bat} + EY_{load})V_{node}$$

The first $n - 1$ rows of $Y_{load}$ are 0, so by Remark 2 we have

$$EY_{bat}V^{PWM}_j = E(Y_{line} + Y_{bat})V_{node}$$ (9)

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

$$V^j_{node} = V^{PWM}_j - I^j_{bat}R_j$$ (10)

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 $j$th 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 3. If an \( m \times n \) matrix is postmultiplied by \((I_n - J_j)\), the \( j \)’th column is set to zero.

Observe that \((I_n - J_j)V_{\text{node}} + J_jV_{\text{node}} = V_{\text{node}}\). Making this substitution for \( V_{\text{node}} \) in (9) we have
\[
EY_{\text{bat}}V_{PWM} = E(Y_{\text{line}} + Y_{\text{bat}})(J_jV_{\text{node}} + (I_n - J_j)V_{\text{node}})
\]
and
\[
EY_{\text{bat}}V_{PWM} - E(Y_{\text{line}} + Y_{\text{bat}})J_jV_{\text{node}} = E(Y_{\text{line}} + Y_{\text{bat}})(I_n - J_j)V_{\text{node}}
\]

Note that module \( j \) can calculate \( J_jV_{\text{node}} = V_{\text{node}} \) by (10). By Remark 3, \( E(Y_{\text{line}} + Y_{\text{bat}})(I_n - J_j) \) has its \( j \)th column set to zeros. That is, the right hand side is unaffected by \( V_{\text{node}} \). Noting this unused data, and that \( E(Y_{\text{line}} + Y_{\text{bat}})(I_n - J_j) \) is an \( n \times 1 \times n \) matrix by Remark 2, we will eliminate the unused column to get a square matrix as follows.

Define matrix \( I_{-j} \) for the \( j \)th battery to be the \( n \times n \) identity matrix with the \( j \)th 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 4. If an \( m \times n \) matrix is postmultiplied by \( I_{-j} \), all except the \( j \)th column are extracted from the matrix.

Remark 5. If an \( n \times m \) matrix is premultiplied by \( I_{-j} \), all except the \( j \)th row are extracted from the matrix.

By Remark 4, post-multiplication of \( E(Y_{\text{line}} + Y_{\text{bat}})(I_n - J_j) \) by \( I_{-j} \) will eliminate the column of zeros. Premultiplication of \( V_{\text{node}} \) on the right hand side by \( I_{-j}^T \) keeps dimensions consistent. By Remark 5, this premultiplication removes \( V_{\text{node}}^{j'} \) 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
\[
EY_{\text{bat}}V_{PWM} = E(Y_{\text{line}} + Y_{\text{bat}})J_jV_{\text{node}} = E(Y_{\text{line}} + Y_{\text{bat}})(I_n - J_j)I_{-j}^T V_{\text{node}}
\]
\[
E(Y_{\text{line}} + Y_{\text{bat}})(I_n - J_j)I_{-j}^T V_{\text{node}} = (E(Y_{\text{line}} + Y_{\text{bat}})(I_n - J_j)I_{-j}^T V_{\text{node}})^{-1} \times (EY_{\text{bat}}V_{PWM} - E(Y_{\text{line}} + Y_{\text{bat}})J_jV_{\text{node}})
\]
Together with (4), this allows for the computation of the node voltages \( V_{\text{node}} \) for each battery module \( j \).

3.3 Load Estimation

Theorem 6. Consider information on \( V_{\text{node}} \). Then \( y_{\text{load}} \) can be found via
\[
y_{\text{load}} = E(Y_{\text{bat}}V_{PWM} - Y_{\text{line}}V_{\text{node}} - Y_{\text{bat}}V_{\text{node}}) / V_{\text{node}}^{n}\]
with \( E \) defined by
\[
E = [0 \ 0 \ldots \ 0 \ 1]_{1 \times n}
\]

Proof. The result in (8) can be revisited to find
\[
y_{\text{bat}}V_{PWM} - Y_{\text{line}}V_{\text{node}} - Y_{\text{bat}}V_{\text{node}} = y_{\text{load}}V_{\text{node}}.
\] (12)

which will provide a way to estimate the load impedance. Recall that each battery module \( j \) has information on \( Y_{\text{bat}} \) and \( Y_{\text{line}} \), and full access to \( V_{\text{node}} \) from (5) and (4). 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 \( f() \) which is laid out in Section 3.4. \( Y_{\text{load}} \) is entirely \( 0 \)’s, except for the last element of the last row, which is \( y_{\text{load}} \). So the right hand side of (12) is entirely \( 0 \)’s, except for the last row. Defining the matrix
\[
\begin{bmatrix}
0 & 0 & \ldots & 0 & 1
\end{bmatrix}_{1 \times n}
\]
provides the ability to extract the last row of an \( n \times m \) matrix by premultiplication with \( E \). This property can be used to write
\[
E(Y_{\text{bat}}V_{PWM} - Y_{\text{line}}V_{\text{node}} - Y_{\text{bat}}V_{\text{node}}) = EY_{\text{load}}V_{\text{node}} = y_{\text{load}}V_{\text{node}}
\]

Where \( V_{\text{node}}^{n} \) denotes the \( n \)th element in the \( V_{\text{node}} \) vector. Since \( V_{\text{node}}^{n} \) is a scalar, we now have
\[
y_{\text{load}} = E(Y_{\text{bat}}V_{PWM} - Y_{\text{line}}V_{\text{node}} - Y_{\text{bat}}V_{\text{node}}) V_{\text{node}}^{n}
\]

for an estimate of the load admittance. This can be computed in each battery module \( j = 1, 2, \ldots, n \) so that information on \( y_{\text{load}} \) is available via the distributed computation of (4), (5), and (13) 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 \( R_{\text{load}} \) 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 \( R_{\text{i}} \) or line impedances \( R_{\text{l}} \). The formulation from Section 3.1 is repeated here for convenience:
\[
\max_{\beta} \beta, \text{ subject to } I_{\text{bat}} = \beta \mathbf{1}_{n \times 1}, 0 \leq PWM_{j} \leq 100
\]

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 \( I_{\text{bat}} \), then (14) can be solved via line search to find the optimal input vector \( PWM \).

Although \( V_{PWM} \) is unknown and may be nonlinear, we note that battery scheduling typically takes place with \( PWM \) close to \( 100 \% \). This motivates the approximation
\[
V_{PWM} = \frac{V_{PWM}}{100}
\]

Which is very close to \( f_j(V_{\text{OCV}}, PWM_{j}) \) when \( PWM_{j} \) is near \( 100 \% \). \( V_{PWM} \) is related to \( I_{\text{bat}} \) via (6) and (7), duplicated here:
\[
I_{\text{bat}} = (Y_{\text{line}} + Y_{\text{load}})V_{\text{node}} = (Y_{\text{total}} - Y_{\text{bat}})V_{\text{node}}
\]

Using (16), (17), and the approximation in (15), we have
\[
I_{\text{bat}} = (Y_{\text{line}} + Y_{\text{load}})Y_{\text{total}}^{-1} Y_{\text{bat}} 100 V_{\text{OCV}} \odot PWM_{j}
\]

where \( \odot \) represents the Hadamard product. Since battery modules can calculate \( y_{\text{load}} \) via (13), they now have information on all the admittances, making (18) easy to calculate for a chosen \( PWM \). Thus module \( j \) can find the inputs \( PWM \) satisfying (14) via a line search and the relationship (18).

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 \( PWM \) it has calculated.
4. BATTERY SCHEDULING RESULTS

4.1 Fully distributed control with known \( f_j \)’s

If the \( f_j \)’s are exactly known, the approximation in (15) is unneeded, and the modules can calculate \( y_{\text{load}} \) and balance their currents with very high accuracy.

<table>
<thead>
<tr>
<th>( R_1 )</th>
<th>( R_2 )</th>
<th>( R_3 )</th>
<th>( R_{12} )</th>
<th>( R_{23} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.47</td>
<td>0.44</td>
<td>0.4</td>
<td>0.09</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Results for the estimation of the load impedance \( R_{\text{load}} \) using the three network battery system with the initial information of battery module impedance and line impedance values summarized in Table 1 are given in Fig. 2. Results were obtained with battery modules in which battery current is measured with an 8bit AD converter. It can be observed from this figure that despite noise on the module current and limited resolution measurements of an 8bit AD converter, both the linear and abrupt changes in the load impedance \( R_{\text{load}} \) are tracked by each battery module.

The tracking of the varying load impedance \( R_{\text{load}} \) 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 \( Y_{\text{total}} \) by the computed \( y_{\text{load}} = 1/R_{\text{load}} \). Battery and line impedance have already been provided centrally and the module current scheduling control runs fully distributed in each module. The resulting measured current out of each battery module is summarized in Fig. 3 where it can be seen that module currents \( I_{\text{bat}}^j \) \( j = 1,2,3 \) 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 \( R_{\text{load}} \), but such imbalances are quickly settled by the fast distributed control algorithm on each battery module.

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. 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. 2 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.

4.2 Distributed control with approximated \( f_j \)’s

If the nonlinear PWM function \( f_j(\cdot) \) of each module is approximated by \( V_{PWM}^j = \frac{V_{\text{OCV}}}{100} \times PWM_j \), a DC offset appears in the current balancing, as can be seen in Fig. 5. Although this DC offset may be acceptable, further improvements can be gained by adding a tunable parameter to the approximation, \( V_{PWM}^j = \frac{V_{\text{OCV}}}{100} \times PWM_j \), and by allowing a centralized algorithm to intermittently send information on the error between module currents, which is used to tune \( \alpha \) locally. Because the PWM range of any individual module is fairly small when currents are...
balanced, tuning this approximation to the local behavior of the nonlinear function $f_j$ is enough to give significant improvement. The improved results can be seen in Fig. 6.

![Graph showing battery currents](image)

**Fig. 5.** Measured battery module currents $I_{bat}^j$ as a function of time in each battery module $j = 1, 2, 3$ due to the load variations depicted in Fig. 2 and with the nonlinear PWM function $f_j(\cdot)$ approximated by a fixed linear function.

![Graph showing battery currents](image)

**Fig. 6.** Measured battery module currents $I_{bat}^j$ as a function of time in each battery module $j = 1, 2, 3$ due to the load variations depicted in Fig. 2 and with the linear model of $f_j$ tuned by each module based on information received from central updates every ten samples. Initial values of the current are unbalanced due to incorrect initial guess of load impedance $R_{load}$.

5. CONCLUSIONS

The optimization and control approach presented in this paper enables second-life battery modules in a parallel network to provide balanced currents to a varying load. 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.

REFERENCES


Elkind, E.N. (2014). 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.


