Virtualizing Energy Storage Management Using RAIBA

Tzi-cker Chiueh, Mao-Cheng Huang, Kai-Cheung Juang, Shih-Hao Liang, and Welkin Ling, *Industrial Technology Research Institute*

https://www.usenix.org/conference/atc18/presentation/chiueh

This paper is included in the Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC '18).

July 11–13, 2018 • Boston, MA, USA

ISBN 978-1-939133-02-1

Open access to the Proceedings of the 2018 USENIX Annual Technical Conference is sponsored by USENIX.

Virtualizing Energy Storage Management Using RAIBA

Tzi-cker Chiueh[1] Mao-Cheng Huang Kai-Cheung Juang Shih-Hao Liang Welkin Ling

Industrial Technology Research Institute

Abstract

Because of the intermittent nature of renewable energy-based electricity generation, a key building block for a sustainable renewable energy-based electricity infrastructure is cost-effective energy storage management, which is largely determined by the cost of electric batteries. Despite substantial technological advances in recent years, batteries used in consumer devices and electric vehicles are still too expensive to be feasible for large-scale deployment. One promising way to reduce the battery cost of an energy storage system is to leverage retired batteries from electric vehicles. However, because the charging/discharging characteristics of retired batteries tend to vary widely from one another, putting these heterogeneous batteries into the same module or energy storage system pose significant safety risks and efficiency challenges. This paper presents the design, implementation and evaluation of a *reconfigurable* battery array system called *RAIBA* that is designed to address the heterogeneity issue in retired battery-based energy storage systems by allowing the inter-battery connectivity to be reconfigurable at run time. In addition, *RAIBA* also enables virtualization of the electrical energy resources in a battery array in the same way as how computing, storage and network resources are virtualized. Empirical measurements on a fully operational *RAIBA* prototype demonstrate that it can effectively increase the discharge service time by more than 80% under a set of real-world electric load traces.

1 Introduction

In light of the global warming and resulting climate change effects triggered by carbon-based fossil energy sources, more and more counties are charging ahead to build an energy infrastructure in which renewable energy plays a major role. Because most important renewable energy sources, such as sun, wind and tidal wave, generate electricity in a way dictated by weather conditions, this intermittent nature renders them unfit as a base load electric energy source, unless they are supplemented by an energy storage system that could bridge the time gaps between energy production and energy consumption. Therefore, for renewable energy to become a significant element of the future clean energy infrastructure, cost-effective energy storage management is a key factor. Most grid-scale energy storage systems today are pumped hydroelectric energy storage, which stores energy in the form of gravitational potential energy of water, which is pumped by off-peak low-cost electricity from a lower-elevation reservoir to one with a higher elevation. However, as electric vehicles become more and more prevalent, the electric batteries they use will increasingly become an important part of the energy storage element of the renewable energy infrastructure.

Despite substantial technological advances in recent years, batteries used in consumer devices and electric vehicles are still too expensive to be feasible for large-scale deployment. In 2017, the LCOE (levelized cost of electricity) of Lithium-based battery [18] is about $0.4 USD per kWh discharge, but US DOE's LCOE target [8] for a cost-effective renewable energy storage is $0.1 USD per kWh discharge. A promising way to facilitate the reduction of the battery cost of an energy storage system is to leverage retired batteries from electric vehicles, as has already been done by several electric or hybrid vehicle manufacturers, such as Tesla, Nissan, Toyota, and BMW. The reason that batteries retired from electric vehicles are still usable as energy storage is because the residual capacity of retired batteries is generally between 70% to 80% of their original full capacity. However, the charging/discharging characteristics of these retired batteries may deviate substantially from those when they were in mint condition.

Conventional battery energy storage systems assume that the charging/discharging characteristics of constituent batteries are homogeneous, as it greatly reduces

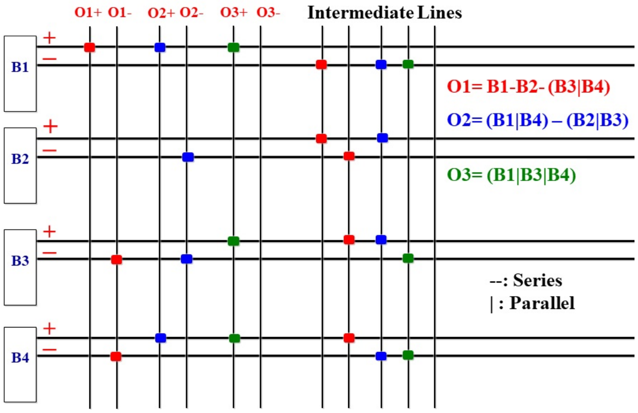


Figure 1: *The most general form of the proposed* RAIBA *architecture, in which each battery could be attached or detached from the interconnect, and any battery could be connected to any other battery in series or in parallel. Intersecting signal lines are connected when they are joined by a dot.*

energy utilization inefficiency. For example, consider a set of batteries that are connected in series, and the capacity of one of them is significantly smaller than the others. In this case, the effective charging/discharging capacity of the entire battery series is dictated by the weakest battery, and the additional capacity of all the other stronger batteries is essentially left wasted. Because retired batteries come from a wide variety of sources, their charging/discharging characteristics are bound to vary widely from one another, and putting these heterogeneous batteries into the same module or energy storage system pose safety risks and utilization efficiency challenges.

In this work, we propose a *Reconfigurable Array of Inexpensive Battery Architecture* (*RAIBA*) [12, 7, 6] to address the heterogeneity issue inherent in energy storage systems that are built from retired batteries. Tesla pioneered the idea of applying commodity batteries used in 3C consumer devices, i.e., 18650 lithium batteries, to building large-scale battery arrays used in electric vehicles. For example, the number of 18650 batteries used in Tesla Model S's battery array is more than 7,000. These battery arrays are provisioned with a certain amount of redundancy to cope with potential battery failures, but when a battery fails, the entire module containing it is impacted because the inter-battery connectivity is fixed. To more effectively minimize the impacts of battery failures and degradations, we propose that a battery array be *reconfigurable* so as to work around failed or degraded batteries at run time.

The most general form of *RAIBA* is shown in Figure 1, which, via software control, controls whether each battery is attached to or detached from the array's interconnect, and how the batteries are connected to one another at run time. The three outputs, O1, O2 and O3, shown in red, blue and green in the figure, represent the outputs of this 4-battery array at three different points in time. The first output (O1) is driven by a series connection among Battery 1 (B1), Battery 2 (B2), and a parallel connection between Battery 3 (B3) and Battery 4 (B4), whereas the second output (O2) corresponds to a series connection between a parallel connection between Battery 1 (B1) and Battery 4 (B4), and another parallel connection between Battery 2 (B2) and Battery 3 (B3). The third output (O3), to which Battery 2 (B2) does not contribute, is simply a parallel connection among B1, B3 and B4. The dynamic reconfigurability afforded by *RAIBA* allows the inter-battery connectivity to be tailored to a given electrical load so as to make the best of available battery resources and minimize unnecessary energy loss, as illustrated by the following three use cases:

- For a series-connected battery array, when the capacity of the weakest battery is exhausted during a discharge operation, *RAIBA* could temporarily put it aside to prevent it from blocking the entire array, and then continues the discharge operation by making the best of the additional capacities of other stronger batteries.
- For a parallel-connected battery array, when the capacity of the weakest battery becomes significantly smaller than that of the others during a discharge operation, *RAIBA* could temporarily put it aside to prevent unwanted inter-battery capacity balancing, which consumes energy, and then continues the discharge operation by making the best of the additional capacities of other stronger batteries.
- Given an electric load request, *RAIBA* makes it possible to use a proper subset of batteries to provide an aggregate voltage and current level which *barely* exceeds those of the request, with their differences and thus the associated down-conversion losses being reduced to the minimum .

The reconfigurability of *RAIBA* opens up myriad battery resource management flexibilities that are not previously possible, and in particular enables virtualization of a physical battery pool in a way similar to how computing and networking resources are virtualized. Given an electric load request $< V, I >$, *RAIBA* dynamically configures a *virtual battery* best fit to satisfy the request by first selecting a proper subset of qualified batteries, and then connecting them in the most appropriate way, so that the virtual battery's aggregate voltage level exceeds V, its aggregate current exceeds I, and the incurred energy loss due to conversion and balancing is minimized.

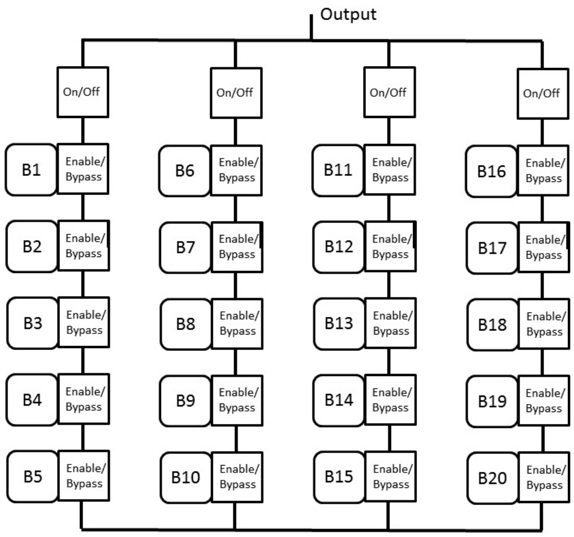


Figure 2: *The system architecture of* RAIBA-1 *consists of N columns connected in parallel, each of which contains M batteries connected in series, with each battery's connectivity to the interconnect via an* enable/bypass *switch.*

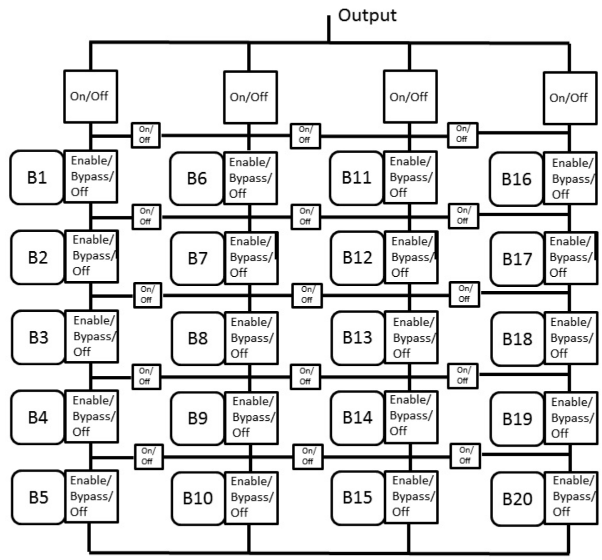


Figure 3: *The system architecture of* RAIBA-2 *is a refinement of* RAIBA-1*, with each* enable/bypass *switch replaced with an* enable/bypass/off *switch, and horizontal inter-column connectivity controlled by* on/off *switches.*

2 RAIBA Levels

The inter-battery interconnect of a *RAIBA* system shown in Figure 1 is deceivingly similar to that of a programmable logic array (PLA) digital logic circuit [14], with a series connection corresponding to an AND operation and a parallel connection corresponding to an OR operation. But the analogy quickly breaks down because of the following two technical challenges:

- The inter-battery interconnect is an energy delivery network that is tasked with carrying much larger electric energy than typical digital logic circuits, and thus require advanced power electronics circuit design techniques to support complex connectivity patterns while guaranteeing operational safety.
- To render inter-battery connectivity reconfiguration completely seamless to an electric load served by a *RAIBA* system, *RAIBA* requires *on-line reconfigurability*, which means the entire system remains functional across each reconfiguration operation and the electric energy pattern delivered to the load before a reconfiguration operation is very close to that after the reconfiguration operation.

Taking into account the significant added circuit complexities involved in supporting dynamic reconfiguration of inter-battery connectivity, we design three RAIBA levels that offer increasing reconfiguration flexibility but also impose growing circuit design challenges.

A *RAIBA-1* array, as shown in Figure 2, consists of N columns connected in parallel, each of which contains M batteries connected in series. At any point in time, each battery is either part of or insulated from a certain column through an *enable/bypass* switch. When the switch is *enable*, the battery participates in the series connectivity of the column; when the switch is *bypass*, the battery takes itself off the series connectivity of the column. Moreover, above all batteries in each column is an *on/off* switch that allows a column to participate in or sit out of the overall inter-column parallel connection. Each battery in a RAIBA-1 system could only be connected in series with other batteries in the same column.

A *RAIBA-2* array, as shown in Figure 3, is basically a *RAIBA-1* array augmented with row-wise inter-column connectivity controllable by *on/off* switches. In addition, each battery is equipped with a *enable/bypass/off* switch that allows a battery to be part of or insulated from its column, or to disconnect the batteries above it in the same column from those below it. That is, when a battery's switch to the battery array's interconnect is off, the battery breaks off the series connectivity of the column to which it belongs. The switchable inter-column interconnects run horizontally and provide additional connectivity flexibility of allowing a battery in the i-th column to be connected in series with another battery in the j-th column, or be connected in parallel with another battery in the j-th column without bypassing all other batteries in the i-th and j-th column. For example, if the horizontal on/off switches above and below B4 and B9 are turned on, B4 and B9 are effectively connected in parallel; if B3 is off and the on/off switch above B4 and B9 is turned on, B4 and B8 are effectively connected in series.

While *RAIBA-1* and *RAIBA-2* are designed to support a single electric load at a time, *RAIBA-3*, as shown in Figure 1, is able to support multiple electric loads simultaneously. In addition, a *RAIBA-3* array allows any battery to be connected in series or in parallel with any other battery in the array. This generality provides much more room for battery resource optimization, but requires each battery to be equipped with approximately as many *on/off* switches as the sum of the numbers of the output lines and intermediate lines, a significant increase hardware implementation complexity.

Although increasing RAIBA levels offer more dynamic reconfigurability, whether the additional hardware complexity associated with higher RAIBA levels is worth the potential gains due to the additional flexibility they provide is an open question. For the rest of this paper, we will only focus on the *RAIBA-1* architecture.

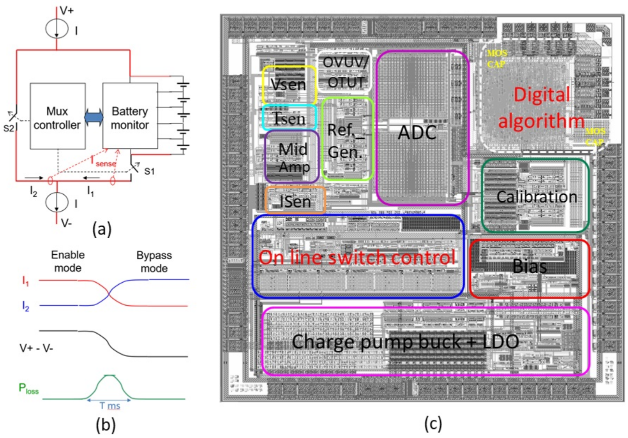


Figure 4: *The (a) circuit architecture and (b) physical layout of the ASIC implementing the* enable/bypass *switch required by the* RAIBA-1 *architecture. (b) shows its ideal transient electric circuit behavior during a transition between the enable and the bypass mode.*

3 Hardware Support

The key building block of the *RAIBA-1* architecture is the *enable/bypass* switch, which either enables a battery to participate in its column, i.e., connecting it in series with the other batteries, or bypasses a battery from its column, i.e., insulating it from the other batteries. While conceptually straightforward, two operational requirements render its design technically challenging:

- The transition between the enable mode and the bypass mode should be as short as possible so as to minimize the energy consumed by each transition.
- The electric current flowing through a *enable/bypass* switch should remain constant during each transition between the enable mode and the bypass mode, so as to minimize the disruption to the electric loads being served.

We have designed and implemented an analog ASIC for the enable/bypass switch, whose architecture and circuit layout are shown in Figure 4. Rather than with an individual battery, this ASIC is designed to work with a battery module, which in turn consists of multiple batteries, in this example, 5 batteries connected in series. An enable/bypass switch includes two on/off switches, S1 and S2, a *battery monitor*, which keeps track of the temperature (Tsen), voltage (Vsen) and current (Isen) of each of batteries in the module, and a *Mux controller*, which controls how S1 and S2 are turned on and off. When S1 is on and S2 is off, the enable/bypass switch bypasses the battery module and the voltage drop across the enable/bypass switch is zero. When S1 is off and S2 is on, the enable/bypass switch enables the battery module and so the voltage drop across the enable/bypass switch is the voltage difference between the two ends of the battery module. When S1 and S2 are both off, the enable/bypass switch cuts off the entire column to which it is connected. When S1 and S2 are both on, the effective circuit becomes a short circuit, and I2 may grow to a dangerously large level. Therefore, this mode of operation is strictly prohibited.

When an enable/bypass switch goes from the enable (bypass) mode to the bypass (enable) mode, the current running through S1, I1, is decreasing (increasing), but the current running through S2, I2, is increasing (decreasing). During a transition between the enable mode and the bypass mode, the *Mux controller* constantly measures I1 and I2, and applies a feedback control mechanism to dynamically tuning the degree to which S1 and S2 are on so that the sum of I1 and I2 remains constant throughput the transition.

To decrease the power consumption incurred by each transition (P_{loss}), the amount of time during which neither S1 nor S2 is off should be minimized, because P_{loss} is equal to the product of the voltage drop and the sum of I1 and I2 during this period, as shown in the lower left of Figure 4. Minimizing the length of the transition period conflicts with the goal of keeping the sum of I1 and I2 constant during the transition period, because, intuitively, it is easier to slowly tune S1 and S2 to keep the total current constant than to try to do so quickly.

When a battery module is enabled by an enable/bypass switch, it may seem that S1 could incur additional energy consumption due to its on-resistance, which is typically very small. However, even for a non-*RAIBA* system, each battery module is typically paired with an on/off switch in order to protect the module from being damaged by unexpected charging currents. This protection on/off switch is no different from S1.

4 Configuration Control Algorithm

Because of its dynamic configurability, *RAIBA* is able to apply a different configuration to each electric load request. Given an electric load $< V, I >$, *RAIBA*'s configuration control algorithm computes a configuration that best serves this load using the following optimization criteria:

- The configuration's delivered current and voltage level exceed I and V, respectively.
- The difference between the configuration's delivered power and $V * I$ is minimized.
- The residual capacities of the batteries in the array are as equalized as possible.

Because every configuration change itself incurs energy loss, *RAIBA* keeps on using its current configuration to satisfy a new electric load request until either the current configuration cannot satisfy the load's $< I, V >$ requirement or the batteries in the current configuration is seriously imbalanced.

The design of *RAIBA*'s configuration control algorithm aims to maximize the energy output of each charge/discharge cycle and the total number of charge/discharge cycles. To squeeze out every bit of the energy accumulated in a charge cycle, it is essential that the left-over battery capacity at the end of a charge cycle be reduced to the minimum. The most likely scenario of squandered battery capacity occurs when one of the batteries in a series-connected battery chain exhausts its capacity and the remaining capacities of the other batteries in the chain are forced to be laid to waste. To avoid this, one should balance the residual capacities of a *RAIBA* array as much as possible. The key to maximizing a battery array's total number of charge cycles is to use each battery in it as gently as possible. Towards this end, when serving an electric load, it is desirable to involve as many batteries and thus draw as little electric current from each battery as possible.

Even though an *enable/bypass* switch in a *RAIBA* system is associated with a battery module, to simplify the exposition below we will assume that each battery modules consists of a single battery. The configuration control algorithm (CCA) used in the current *RAIBA* prototype is shown in Figure 5. Designed with in mind the above optimization objectives, CCA first identifies all possible configurations in an NxM battery array that meet the requirements of the given electric load request,

```
Input: < I,V > of an electric load request, and the
      residual capacity and voltage level of each
      battery in an NxM array;
for (each of the N columns) {
    Sort the batteries in the column according to
    their residual capacity into a list;

    Traverse the list in the sorted order, find all
    possible battery combinations whose
    accumulated voltage level is between V and V∗α,
    and place the resulting combinations into a set;

    Disqualify the column if its set is empty;
}

Form a candidate configuration by picking one
battery combination from each qualified column's
set, and put all candidate configurations into a
list, CCL;

for (each candidate configuration in CCL) {
    Simulate the candidate configuration for one
    time step according to a battery model derived
    from dynamic measurements;

    Compute the candidate configuration's
    eventual output voltage level, V_out, and
    switching cost, Cost_switch;

    Derive the residual capacity of all
    participating batteries one time step later,
    and compute the standard deviation of the
    residual capacities of all batteries, STD_c;

    Disqualify the candidate configuration if
    the current going through any participating
    column is negative or its total power output
    is less than V∗I;
}

Select the qualified candidate configuration that
minimizes β∗(V_out−V)/V_out + γ∗STD_c + δ∗Cost_switch;
```

Figure 5: RAIBA*'s configuration control algorithm, which aims to balance the aggregate voltage levels of participating columns and the residual capacities of all the batteries*

and then picks the one that best balances the residual capacities of all the batteries in the array. Instead of trying out all possible battery combinations, CCA takes a greedy approach by processing each column independently, and within each column, considering only battery combinations that consist of top K batteries in the column's battery list sorted according to their residual capacity and whose aggregate voltage level lies between V and $\alpha * V$. A column is disqualified if the aggregate voltage level of all M batteries in it is below V. The α parameter bounds the search scope and prevents CCA from examining "over-provisioned" configurations. If no satisfactory configuration could be identified for a given α value, CCA increases α and repeats the algorithmic process in Figure 5 again.

When the aggregate voltage level of one column of a candidate configuration is significantly lower than those of the other columns, other columns may charge the weaker column to bring its voltage level up to par with others, in which case the electric current going through the weaker column is *negative*, or opposite in direction to the electric current requirement (I) of the electric load request. Because the overhead incurred by such inter-column charging represents an unnecessary energy loss, CCA disqualifies all candidate configurations that lead to inter-column charging from consideration.

To maximize the life time of an array's batteries, CCA spreads an electric load over as many columns as possible by considering only those configurations that include all the qualified columns. That is, if L columns in an N-column array are qualified, CCA considers only configurations that consists of all L columns, but not those consisting of a proper subset of these L columns. It is conceivable that configurations using fewer than L columns could lead to lower STD_c or V_{out} or both, but including them into consideration would significantly enlarge the search space.

CCA takes into account the following three factors when selecting the best among the candidate configurations. First, to reduce the amount of wasted battery capacity at the end of a charge cycle, CCA strives to balance the residual capacity of an array's batteries, STD_c, by minimizing the standard deviation of their residual capacity after a time period. Second, to reduce the energy loss due to the inverter, which down-converts a candidate configuration's eventual output voltage (V_{out}) to V, CCA also minimizes the difference between V_{out} and V. Finally, because each switching of an *enable/bypass* switch also incurs an energy loss, it is desirable to pick a configuration that is as close to the current configuration as possible. For this, CCA computes the switching energy cost of each candidate configuration, $Cost_{switch}$. Because these factors may conflict with one another, CCA uses three empirically determined parameters, β, γ and δ, to adjust their relative importance or weight.

Given a candidate configuration that comprises L columns, each of which consists of a variable number of batteries, to execute the above algorithm, CCA needs to compute the eventual output voltage V_{out}, the current going through each of the columns, and the residual capacity of each participating battery after a time period. Because of the non-linear discharging characteristics of modern batteries, CCA resorts to a simulation approach to deriving the equivalent electrical circuit behavior of each participating battery. Instead of pre-calibrating each battery in advance, CCA adopts a *trace-based* strategy to build up a simulation model for each battery by periodically measuring the instantaneous *discharge current*, *voltage level* and *used capacity* when it is discharged,

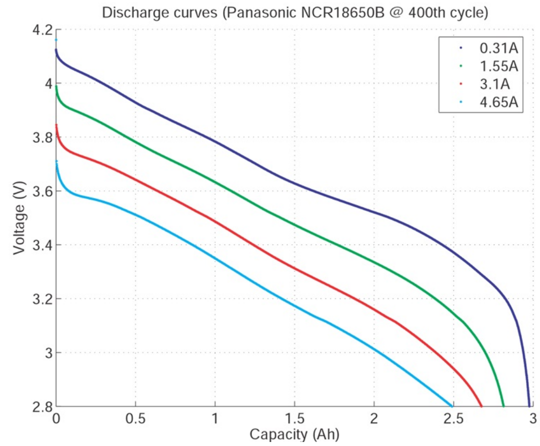


Figure 6: *The discharge characteristic curves for a Panasonic NCR 18650B battery that has been charged 400 times for four different discharge currents, 0.31A, 1.55A, 3.1A, and 4.65A. Measurements were taken in the temperature range between 25 and 35 degrees Celsius.*

generating a sampled version of its discharge characteristic curves (DCC) [17, 19], which describes how a battery's voltage level evolves with its used (not residual) capacity at a discharge current, and then applying linear interpolation to approximating those points on the DCCs that do not have measured values. For example, Figure 6 shows the measured DCCs for a 400-cycle Panasonic NCR 18650B battery for four different discharge currents, 0.31A, 1.55A, 3.1A, and 4.65A, with measurements taken in the temperature range between 25 and 35 degrees Celsius. Note that there is a distinct linear and thus easier to predict range for the DCC corresponding to a particular discharge current. The smaller the discharge current, the larger the corresponding DCC's linear range. Also, suppose the cut-off voltage level is 3V, then the total usable capacity of this battery is around 2Ah when it is discharged at 4.65A, but is about 2.95Ah when it is discharged at 0.31A. This example illustrates that discharging batteries as gently as possible not only lengthens their total lifetime, but also increases their per-charge usable capacity.

CCA models a battery as a voltage source connected in series with an internal resistance. For an L-column candidate configuration, CCA assumes the initial current going through each column and thus each battery in the array is $\frac{I}{L}$. To compute the internal resistance of a battery with a used capacity X and a discharge current of $\frac{I}{L}$, CCA first identifies the two DCC measurements (<discharge current, voltage level, used capacity>) that are closest to $< \frac{I}{L}, , X >$, say $< I_a, V_a, UC_a >$ and $< I_b, V_b, UC_b >$, and then approximates its internal resistance as $\frac{\triangle V}{\triangle I} = \frac{V_b - V_a}{I_a - I_b}$. For example, to calculate the internal resistance of the Panasonic battery whose DCC is shown in Figure 6 when

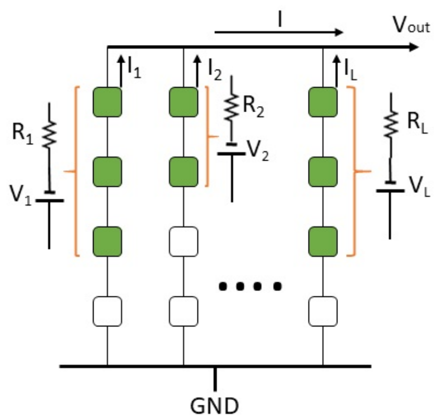


Figure 7: *The equivalent circuit of an L-column candidate configuration, which selects the first 3 batteries of the first column, the first 2 batteries of the second column,... and the first 3 batteries of the L-th column. Each column's effective resistance is the sum of the internal resistances of the participating batteries in that column.*

its used battery is 2Ah and discharge current is 3.6A, we identify the closest measurements $< 3.0V, 4.65A, 2Ah >$ and $< 3.18V, 3.1A, 2Ah >$, and then compute its internal resistance as $\frac{3.18-3.0}{4.65-3.1} = 0.12$ Ohm.

With the internal resistance of every participating battery, CCA computes the aggregate voltage level (V_i)and aggregate internal resistance R_i for each column in the candidate configuration, represents the L-column candidate configuration as an equivalent circuit as shown in Figure 7, and then solves the corresponding linear system of equations as follows, to derive V_{out} and the actual current going through each column, I_i:

$$\begin{aligned} V_1 - I_1 * R_1 &= V_{out} \\ V_2 - I_2 * R_2 &= V_{out} \\ V_3 - I_3 * R_3 &= V_{out} \\ &\dots\dots\dots \\ V_L - I_L * R_L &= V_{out} \\ I_1 + I_2 + I_3 + \dots I_L &= I \end{aligned}$$

If V_{out} is larger than some V_j, then the corresponding current I_j must be negative, the corresponding (j-th) column is disqualified, and the associated candidate configuration is considered unusable. Once the current going through each column of a candidate configuration is known, CCA computes, for each participating battery, the amount of charge that will be discharged within a fixed time period by multiplying the discharge current with the period's length, subtracts the multiplication result from the battery's residual capacity, and finally derives the standard deviation of the residual capacities of all batteries in the configuration, STD_c.

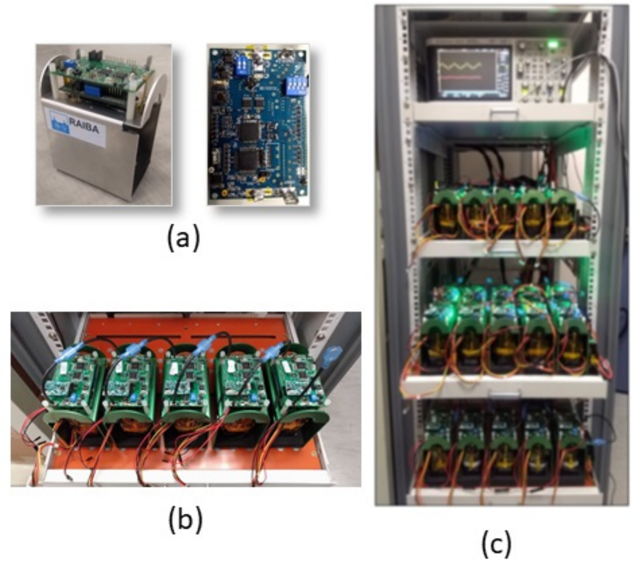


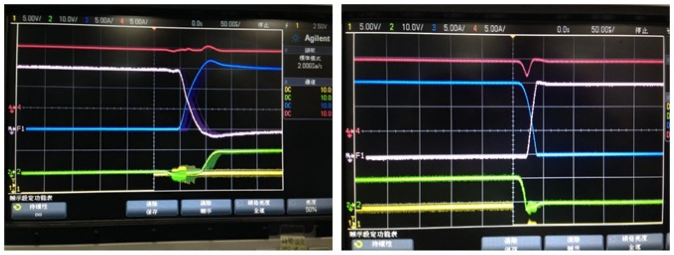
Figure 8: *The hardware implementation of the first* RAIBA *prototype, whose building block is a 4S2P battery module protected by an* ensemble/bypass *switch, shown in (a). Five such battery modules are connected in series to form a column, shown in (b), and the entire prototype contains five such columns connected in parallel and other measurement/control/protection circuits, and is housed in a rack, as shown in (c).*

A nice benefit of the *trace-based* approach to battery modeling is the model derived from measurements taken on a battery is tailored to and ages with the battery, and is thus more likely to better approximate the battery's ground truth than a pre-calibrated model.

5 Prototype Implementation

The first *RAIBA* prototype is a 5x5 array of battery modules, each of which in turn consists of 2 parallel-connected columns with each column comprising 4 Panasonic INR18650GA 3.4Ah batteries connected in series. Therefore, the entire prototype is composed of two hundred 18650 batteries and contains an energy capacity of 2.5KWh. As shown in Figure 8(a), associated with each battery module is an *enable/bypass* switch board and a cooling subsystem that provides thermal load management for the 4S2P (2 parallel-connected columns each having 4 series-connected batteries) batteries in the module. Five of these battery modules are connected in series to form a column, as shown in Figure 8(b), and five of these columns are connected in parallel and placed in a rack to form the complete battery array, as shown in Figure 8(c).

In addition to the battery array, the *RAIBA* prototype also contains a *RAIBA controller*, which is a Linux-based



(a) (b)

Figure 9: *The electric circuit behavior of the enable/bypass switch IC used in the current* RAIBA *prototype for (a) the transition from the bypass to enable mode, and (b) the transition from the enable to bypass mode. Red represents the current traversing the entire enable/bypass switch, White and Blue represent the currents traversing S1 and S2, respectively, and Green represent the voltage across the enable/bypass switch.*

Raspberry Pi 3 board that runs the Configuration Control algorithm based on electric load requests and battery status measurements, a *battery status measurement and communication module*, which uses a Linear Technology LTC6804-1 battery monitor to constantly measure the voltage, current, and temperature of each battery in the array, and an ATmega328P MCU to report these measurements in real time through the UART protocol to the RAIBA controller, and a *surge protector* that offers a safeguard mechanism to limit unexpected transient surge currents during battery array reconfiguration. The charger circuit and the DC/AC inverter for discharging are connected to the *surge protector* to charge and discharge the *RAIBA* prototype, respectively.

6 Performance Evaluation

6.1 Efficiency of Enable/Bypass Switch

Unlike digital logic switch or computer network switch, signals flowing on *RAIBA*'s enable/bypass switches represent energies to be delivered to and consumed by electric loads. Because these signals' magnitude is much larger and their transmission behavior is driven by instantaneous voltage level differences and thus largely analog, whether it is feasible to successfully implement an electrically safe enable/bypass switch that keeps the traversed electric current constant during the transitions between the enable mode and the bypass mode, raised serious doubts in the beginning of the *RAIBA* project. We have two IC implementations for the enable/bypass switch IC. The electric circuit behaviors of the version used in the current *RAIBA* prototype during the transitions of the enable and the bypass mode are shown in Figure 9. Red represents the current traversing the entire enable/bypass switch, White and Blue represent the currents traversing the two constituent switches S1 and S2 (shown in Figure 4(a)), respectively, and Green represent the voltage across the enable/bypass switch.

When the voltage drop across an enable/bypass switch is 16V and the running current is 15A, the total switch time is 50 μsec for this switch to go from the bypass to enable mode, and is 30 μsec for an enable/bypass switch to go from the enable to the bypass mode. The resulting power loss (P_{loss}) is 3mJ for the transition from the bypass to enable mode, and is 1.8mJ for the transition from the enable to bypass mode. Although there are still some glitches in the traversed current during the transitions, as indicated by the dips and bumps in the red curves in Figure 9 (a) and (b), these glitches are relatively small in magnitude and thus seamless to the electric loads.

The above results conclusively demonstrates that an implementation of an electrically safe enable/bypass switch IC which keeps the current traversing it constant during transitions not only is feasible, but also could be made highly efficient in terms of switch time and switching-induced energy loss.

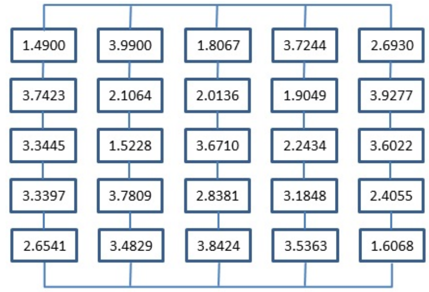


Figure 10: *The initial capacity of each battery in the array used in the test against five electric load traces*

6.2 Gains from Dynamic Reconfigurability

Intuitively, the more heterogeneous the batteries in a *RAIBA* array and the more fluctuated the electric load facing a *RAIBA* array, the higher performance gain *RAIBA*'s dynamic reconfigurability is expected to provide. To empirically assess the gain from *RAIBA*'s dynamic reconfigurability, we tested the first *RAIBA* prototype under a set of electric load traces until it cannot be discharged any further, and measure the total discharge service time, the percentage of wasted battery capacity at the end of the discharge cycle, and the number of enable/bypass mode transitions.

To reduce the amount of time required for each experiment run, we limited the capacity of each battery module to under 4Ah, but still used the entire 5x5 array. Because *RAIBA* is designed to minimize unnecessary energy waste when the constituent batteries in an array ex-

Electric Load	Discharge Service Time (hr)	Wasted Capacity (Ah)	Standard Deviation (Ah)	Transition Count
Simple	4.59 / 8.42 (83.4%)	35.5698 / 2.3747 (-93.3%)	0.8372 / 0.0428 (-94.9%)	0 / 3059
eBus	3.77 / 6.89 (82.9%)	35.5325 / 2.4114 (-93.2%)	0.8372 / 0.0695 (-91.7%)	0 / 1385
Data center	4.16 / 7.66 (84.0%)	35.5472 / 2.0324 (-94.3%)	0.8372 / 0.0931 (-88.9%)	0 / 2083
NYC	2.98 / 5.77 (93.4%)	35.4782 / 2.1170 (-94.3%)	0.8372 / 0.1040 (-87.6%)	0 / 2493
TaiPower	2.93 / 6.09 (108.1%)	35.4609 / 2.4612 (-93.1%)	0.8372 / 0.2510 (-70.0%)	0 / 545

Table 1: *Comparison between the* Fixed *configuration (left) and the* Reconfigurable *configuration (right) in terms of the total discharge service time, the total wasted capacity at the end, the standard deviation in battery capacity at the end, and the number of mode transitions, under five electric load traces. Numbers inside the parenthesis represent the percentage difference between the* Reconfigurable *configuration and the* Fixed *configuration.*

hibit diverse discharging characteristics or have different initial capacities, we focused below on a test case in which the initial capacities of the battery modules in the array vary from 1.49Ah to 3.99Ahm, as shown in Figure 10, where the standard deviation of the individual battery capacity across the array is 0.8377Ah. The total battery capacity of the entire array is 72.4544Ah, and the maximum power is 1790.3365W. To evaluate how the *RAIBA* prototype performs under different electric load patterns, we used the following five electric load traces that represent a wide variety of use cases:

- *Simple*: A constant load at the level of 120W.
- *eBus*: A simplified version of an electric load trace captured from an electric bus that consists of periods each of which includes a 5-sec accelerate phase of 200W, a 50-sec cruise phase of 150W, and a 10-sec decelerate phase of 100W.
- *Data center*: An electric load trace that was collected on 2016/08/02 from a 700+-server cloud computing data center within Industrial Technology Research Institute, and then scaled down in the power magnitude by a factor of 0.02.
- *NYC*: A scaled down version of the electric load trace for New York City on 2016/04/01 by a factor of $3*10^{-8}$.
- *Taipower*: A scaled down version of the electric load trace for the Northern part of Taiwan on 2016/03/29 by a factor of 10^{-7}.

We exercised each of the five electric load traces against the *RAIBA* prototype twice, once when we enabled the array's reconfigurability capability (*Reconfigurable* configuration), and the other time when we enabled it completely (*Fixed* configuration). At the beginning of each run, we charged each battery in the array according to the specification in Figure 10, and then discharged the array using a programmable electric load generator that drew power over time according to a given electric load trace. Each experiment run terminates when the *RAIBA* prototype can no longer continue servicing the corresponding electric load trace.

Table 1 shows the detailed comparisons between the *Fixed* and *Reconfigurable* configuration under the five electric load traces. For each and every of the five electric load traces, the *Reconfigurable* configuration outperforms the *Fixed* configuration in terms of the total discharge service time by between 82.9% and 108.1%. That is, for the same initial array condition and electric load trace, the *Reconfigurable* configuration lasts almost twice as long as the *Fixed* configuration. This gain mainly comes from the reduction in the imbalance of the batteries' residual capacity, as shown in the Standard Deviation column, which that indicates the *Reconfigurable* configuration reduces the standard deviation in residual battery capacity at the end of an experiment run by between 70% to 94.9% when compared with that of the *Fixed* configuration, which is roughly the same as the initial standard deviation because each battery contributes equally during the discharging process. Figure 11 shows visually how an array's batteries' residual capacities evolve over time under an *eBus* trace when dynamic reconfigurability is turned on and off. As expected, the residual capacities of an array's batteries converge over time towards a common value when *RAIBA*'s dynamic reconfigurability is enabled, but progress largely independently of one another when dynamic reconfigurability is disabled.

Hardware-based inter-battery capacity balancing, which transfers charge from larger-capacity batteries to lower-capacity batteries, and inevitably incurs energy loss. In contrast, *RAIBA* balances the residual capacities of an array's batteries by drafting different subsets of batteries to work at different time, and thus is more general because it could balance the residual capacities of those batteries that are not electrically connected, and more energy-efficient because it does not involve any movement of electric charges between batteries.

When the residual capacities of an array's batteries are more balanced, it is less likely that the array is forced to

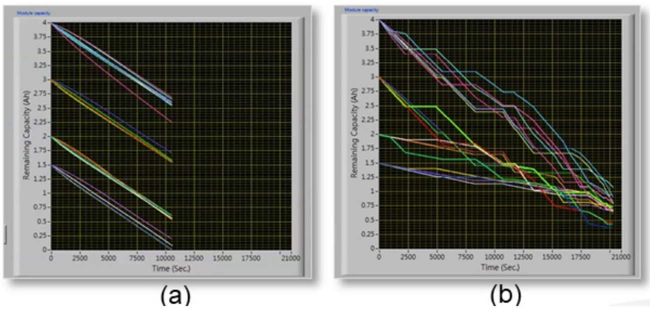


Figure 11: *The residual capacities of a* RAIBA *array's batteries evolve over time under an* eBus *trace when dynamic reconfigurability is turned (a) off and (b) on.*

terminate earlier on because of the capacity exhaustion of some batteries, and the amount of residual capacity lying unused and wasted at the end of each experiment run is expected to be smaller. The Wasted Capacity column of Table 1 shows that the wasted capacity at the end of an experiment run for the *Reconfigurable* configuration is less than 10% of that of the *Fixed* configuration for each of the five electric load traces. This demonstrates the *Reconfigurable* configuration's capability to eliminate energy waste due to fragmentation via more effective inter-battery capacity balancing.

That there is not much correlation between the Transition Count column and the Discharge Service Time column of Table 1 suggests that a higher number of mode transitions does not necessarily result in a higher gain in the total discharge service time. The gain also has a lot to do with the degree of mismatch between the demand patterns of an electric load trace and the energy profile that the *Fixed* configuration could offer.

The price of dynamic reconfigurability is the additional energy consumption due to transitions between the enable and the bypass mode. However, the associated energy consumption is rather miniscule. For example, even if a discharge cycle requires 3000 mode transitions, as in the case of the *Simple* trace in Table 1, the total energy consumption is about $3000 * 3mJ = 9J$, which is about 0.001% of the total energy capacity of the 5x5 battery array used in the test.

7 Related Work

Song Ci et al. [4] provides a detailed survey of the reconfigurable battery techniques, including various reconfigurable battery array designs proposed in the literature, their management and fault tolerance properties, and the design considerations of the associated battery management mechanisms. Baronti et al. [3] and Miyatake et al. [20] explored the effective capacity of a battery array with different inter-battery connectivity configurations. Baronti et al. [2] explored the design space for the bypass/enable switch module. Kim and Shin [15] first proposed a dynamically reconfigurable architecture for large-scale battery arrays used in electric vehicles in order to tolerate battery cell failures. Jin and Shin [13] followed up on [15] with the development of battery pack sizing and reconfiguration algorithms. Kim et al. [16] built the first small-scale (6x3) reconfigurable battery array called self-X, which aimed to tolerate battery failures, balance the capacities among batteries and optimize energy conversion efficiency. He et al. [9] took into account battery conditions, particularly state of health, to dynamically reconfigure a battery array to maximize its total capacity. He et al. [10] improved the performance of charging operations by leveraging various battery state information to best exploit dynamic reconfigurability. He et al. [11] used dynamic reconfigurability to allow weaker cells to rest longer so as to balance inter-battery capacity and increase the battery array's effective capacity.

Badam et al. [1] proposed a software-defined battery system that includes batteries of different charging/discharging characteristics and provides an API for the control software to use the most appropriate batteries to service given electric loads. While dynamic reconfigurability was originally proposed for large-scale battery arrays, Visairo and Kumar [21] and Ci et al. [5] explored the effectiveness of applying dynamic reconfigurability to portable and mobile devices.

RAIBA differs from the research efforts described in the following ways. First, RAIBA features a real implementation of a switching IC that is able to enable and bypass a battery of a battery array while keeping the traversing current constant and the array continuing operating. This real-time dynamic reconfigurability makes it possible to apply RAIBA to applications beyond stand-by energy storage systems, such as electric vehicles. Second, RAIBA supports a dynamic battery array reconfiguration algorithm that takes into account the capacity/state of each battery and the target electric load, and produces in real time a battery array configuration that meets the energy needs of the target load while minimizing unnecessary energy waste due to inter-battery balancing and power conversion. Third, RAIBA adopts a trace-based battery model that removes the need for batteries with homogeneous quality, and is able to accommodate batteries that age over times.

8 Conclusion

This work proposes a *RAIBA* approach to using retired batteries to build cost-effective energy storage systems that make up for the intermittent nature of renewable energy generation systems. The key idea in *RAIBA* is to use dynamic reconfigurability to make the best of heterogeneity in retired batteries, so as to enable continued operations even in the presence of individual battery failures, maximize the energy output of each charge/discharge cycle, and minimize energy loss due to conversion, analog inter-battery capacity balancing and resource fragmentation. The paper describes the design, implementation and evaluation of a fully operational *RAIBA* -1 prototype. More specifically, this paper makes the following contributions to the energy storage management area:

- A taxonomic framework for analyzing varying degrees of flexibility of dynamically reconfiguring the inter-battery connectivity of large-scale battery arrays at run time,
- The first successful and efficient implementation of an *enable/bypass* switch IC that keeps the traversed current constant during transitions between the bypass and enable mode, and
- The completion of a fully operational software-defined virtualized battery array prototype, and an empirical demonstration of the efficacy of its dynamic reconfigurability in increasing the effective discharge service time by more than 80% for a variety of electric load traces.

Notes

[1] Authors are listed in the alphabetical order of their last names.

References

[1] Anirudh Badam, Ranveer Chandra, Jon Dutra, Anthony Ferrese, Steve Hodges, Pan Hu, Julia Meinershagen, Thomas Moscibroda, Bodhi Priyantha, and Evangelia Skiani. Software defined batteries. In *Proceedings of the 25th Symposium on Operating Systems Principles (SOSP '15)*, pages 215–229, October 2015.

[2] Federico Baronti, Gabriele Fantechi, Roberto Roncella, and Roberto Saletti. Design of a module switch for battery pack reconfiguration in high-power applications. In *Proceedings of IEEE International Symposium on Industrial Electronics*, pages 1330–1335, May 2012.

[3] Federico Baronti, Roberto Di Rienzo, Nicola Papazafiropulos, Roberto Roncella, and Roberto Saletti. Investigation of series-parallel connections of multi-module batteries for electrified vehicles. In *Proceedings of IEEE International Electric Vehicle Conference*, Dec. 2014.

[4] Song Ci, Ni Lin, and Dalei Wu. Reconfigurable battery techniques and systems: A survey. *IEEE Acess*, 4:1175–1189, 2016.

[5] Song Ci, Jiucai Zhang, Hamid Sharif, and Mahmoud Alahmad. Dynamic reconfigurable multi-cell battery: A novel approach to improve battery performance. In *Proceedings of Applied Power Electronics Conference and Exposition (APEC)*, pages 439–442, 2012.

[6] Tzi-cker Chiueh, Chia-Ming Chang, Welkin Ling, and Shih-Hao Liang. Power management method for low capacity state of electro-chemical batteries. In *US Patent No. US9229510*, January 2016.

[7] Tzi-cker Chiueh, Shih-Hao Liang, Kai-Cheung Juang, and Shou-Hung Ling. Battery system and control method thereof. In *US Publication No. US20170133865 A1*, May 2017.

[8] Stephen D. Comello, Gunther Glenk, and Stefan Reichelstein. Levelized cost of electricity calculator: A user guide. *Stanford School of Business, Sustaintable Energy Initiative*, May 2017. `http://stanford.edu/dept/gsb_circle/cgibin/sustainableEnergy/GSB_LCOE_User%20Guide_0517.pdf`.

[9] Liang He, Yu Gu, Ting Zhu, Cong Liu, and Kang G. Shin. Share: Soh-aware reconfiguration to enhance deliverable capacity of large-scale battery packs. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems (ICCPS '15)*, pages 169–178, April 2015.

[10] Liang He, Eugene Kim, and Kang G. Shin. -aware charging of lithium-ion battery cells. In *Proceedings of 7th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS 2016)*, pages 26:1–26:10, April 2016.

[11] Liang He, Eugene Kim, and Kang G. Shin. Resting weak cells to improve battery pack's capacity delivery via reconfiguration. In *Proceedings of the Seventh International Conference on Future Energy Systems*, pages 8:1–8:11, June 2016.

[12] Chien-Tung Hsu, Shou-Hung Ling, Kai-Cheung Juang, Tzi-cker Chiueh, and Chuan-Yu Cho. Programmable battery source architecture and method thereof. In *US Publication No. US 20160164315 A1*, June 2016.

[13] Fangjian Jin and Kang G. Shin. Pack sizing and reconfiguration for management of large-scale batteries. In *Proceedings of IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS 2012)*, pages 138–147, 2012.

[14] Yahiko Kambayashi. Logic design of programmable logic arrays. *IEEE Transactions on Computers*, 28:586–590, 1979.

[15] Hahnsang Kim and Kang G. Shin. On dynamic reconfiguration of a large-scale battery system. In *Proceedings of 15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2009)*, pages 87–96, April 2009.

[16] Taesic Kim, Wei Qiao, and Liyan Qu. Power electronics-enabled self-x multicell batteries: a design toward smart batteries. *IEEE Transactions on Power Electronics*, 27:4273–4283, 2012.

[17] Shih-Hao Liang, Tzi-cker Chiueh, and Welkin Ling. Go gentle into the good night via controlled battery discharging. In *Proceedings of the 6th Asia-Pacific Workshop on Systems, (APSys 2015)*, July 2015.

[18] David Linden and Thomas B. Reddy. *Hand book of batteries (3rd edition)*. McGraw-Hill, 2002.

[19] Welkin Ling, Chein-Chung Sun, and Chiou-Chu Lai. Method for checking and modulating battery capacity and power based on discharging/charging characteristics. In *US Patent No. US8823325*, September 2014.

[20] So Miyatake, Yoshihiko Susuki, Takashi Hikihara, Syuichi Itoh, and Kenichi Tanaka. Discharge characteristics of multicell lithium-ion battery with nonuniform cells. *Journal of Power Sources*, 241:736–743, 2015.

[21] H. Visairo and P. Kumar. A reconfigurable battery pack for improving power conversion efficiency in portable devices. In *Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems*, April 2008.