Exploring Adaptive Reconfiguration to Optimize Energy Efficiency in Large-Scale Battery Systems

Liang He[1], Lipeng Gu[2], Linghe Kong[1], Yu Gu[1], Cong Liu[3], Tian He[4]
[1]Singapore University of Technology and Design, Singapore
[2]Shanghai University of Engineering Science, Shanghai, China
[3]The University of Texas at Dallas, Dallas, TX, USA
[4]University of Minnesota, Minneapolis, MN, USA

***Abstract*—Large-scale battery packs with hundreds/thousands of battery cells are commonly adopted in many emerging cyber-physical systems such as electric vehicles and smart micro-grids. For many applications, the load requirements on the battery systems are dynamic and could significantly change over time. How to resolve the discrepancies between the output power supplied by the battery system and the input power required by the loads is key to the development of large-scale battery systems. Traditionally, voltage regulators are often adopted to convert the voltage outputs to match loads' required input power. Unfortunately, the efficiency of utilizing such voltage regulators degrades significantly when the difference between supplied and required voltages becomes large or the load becomes light. In this paper, we propose to address this problem via an adaptive reconfiguration framework for the battery system. By abstracting the battery system into a graph representation, we develop two adaptive reconfiguration algorithms to identify the desired system configurations dynamically in accordance with real-time load requirements. We extensively evaluate our design with empirical experiments on a prototype battery system, electric vehicle driving trace-based emulation, and battery discharge trace-based simulations. The evaluation results demonstrate that, depending on the system states, our proposed adaptive reconfiguration algorithms are able to achieve $1\times$ to $5\times$ performance improvement with regard to the system operation time.**

I. Introduction

Large-scale battery systems with hundreds or thousands of batteries are now widely used in electric vehicles [33], [36], energy storage in both macro [9] and micro [35], [42] smart grids. For many of these applications, the load requirement on the battery system is dynamic and could significantly change over time [3], [28]. For example, depending on the driving states, the required voltage output of electric vehicles may vary from around 70 V to more than 700 V [3], [4], [23]. Such dynamic loads make the problem of optimizing the energy efficiency of large-scale battery systems even more critical and challenging, which is attracting increasing attentions of funding agencies [2], [8] (e.g., ARPA-E has awarded USD 43 million to 19 energy storage projects in 2012 [2]), and research efforts [25], [27], [29].

A traditional method of handling dynamic loads is to adopt voltage regulators to accept voltages supplied by the battery pack and adjust them to the required levels as the input to the loads [29], [41]. Unfortunately, the energy efficiency of voltage regulators may degrade significantly under two scenarios: (*i*) the difference between the supplied and required voltages is large [23], [29], [34], and (*ii*) the load is light and the system operates in a low power modes [41].

In contrast, dynamically adjusting the connections among batteries inside a battery system based on the real-time load requirements, referred as the *adaptive system reconfiguration*, is an alternative approach to handle the mismatch between the supply and the requirement [19]. The adaptive reconfiguration not only avoids the low efficiency issue of the traditional regulator-based approaches, but also increases the system robustness in that failed batteries can be by-passed without significantly degrading the system performance [23], [26]. Much research has been conducted to design battery systems that offer higher configuration flexibility with fewer supplementary electronic components such as connectors and switches [10], [20], [24], which has already been implemented in many off-the-shelf battery packs [8], [19], [24], [26], [39].

Besides offering configuration flexibility, there yet exists another open challenge in maximizing energy efficiency, which is to optimally determine battery system configurations in accordance with real-time load requirements. Motivated by this, in this paper, we advance the state-of-the-art by addressing the following research question: *with a given configuration flexibility of a battery system, how to adaptively identify the optimal system configuration based on real-time load requirements.* We first prove that this problem is *NP-Hard* in general, and then we effectively solve it based on two empirical observations on battery systems. We propose a near-optimal adaptive reconfiguration algorithm based on the classic 0-1 integer programming problem for the single load change scenario. For the scenario where multiple loads change simultaneously, we extend our design with a greedy heuristic to identify the desired system configurations.

Our major contributions in this paper include

- We propose a generic graph representation of large-scale battery systems, which facilitates the optimization of battery system reconfigurations.
- For the scenario where only a single load changes, we transform the problem of identifying the optimal system configuration to a *path selection* problem in the corresponding graph. We first prove that the problem is *NP-Hard*. We then propose a practically feasible solution based on two empirical observations, which is able to return a near-optimal system configuration through Depth-First-Search with pruning method and 0-1 integer programming formulation.
- Extending our investigation to the scenario of multiple load changes, we propose a greedy solution to identify

the desired system configuration by greedily selecting the load to be processed and progressively achieving the desired configuration.
- We extensively evaluate our design with empirical experiments on a prototype battery system, electric vehicle driving trace-based emulation, and battery discharge trace-based simulations. The evaluation results demonstrate that the proposed adaptive reconfiguration algorithms can achieve $1\times$ to $5\times$ improvement in the system operation time.

The paper is organized as follows. The problem statement is presented in Section II. The scenarios with a single and multiple loads changes are investigated in Section III and Section IV, respectively. Our prototype implementation and simulation/emulation results are presented in Section VI and Section VII, respectively. Section VIII reviews the literature. Section V briefly discuss practical issues relevant to our design. Section IX concludes.

II. System Model and Design Principles

A. System Model

We consider large-scale battery systems that can support multiple loads simultaneously in this work [24], as shown in Fig. 1. The battery pack in the system has multiple output terminal pairs, and each terminal pair is connected to a different load. For the i^{th} terminal pair, we denote the load's required voltage and power as $[V^i, (1+\sigma)V^i]$ and P^i respectively, where σ is the tolerable jitter voltage ratio. The battery pack consists of a total number of N batteries, and the voltage of the i^{th} battery at the decision time is $v_i \in [v_c, v_f]$, where v_c and v_f are the battery cutoff voltage [1] and full charge voltage, respectively.

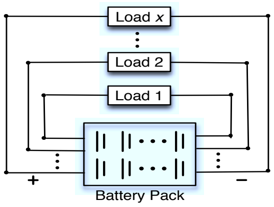


Fig. 1. System model.

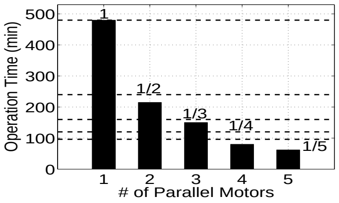


Fig. 2. Reducing discharge current improves energy efficiency.

In practice, the probability for multiple loads to change at the same time is relatively low. Thus we first simplify our investigation by assuming a single load change in Section III. We then tackle the case where multiple loads change simultaneously in Section IV.

B. Design Principles

The ultimate goal of adaptive system reconfiguration is to maximize energy efficiency by adopting the optimal configurations in accordance with real-time load requirements. Energy efficiency of a battery system is jointly determined by many factors such as adopted electronic components, battery temperature, loads, battery chemical properties, etc. To hide these complex factors from practical implementation, we propose the following rule-of-thumb design principles.

1) Matching Supplied and Required Voltages: Using voltage regulators to convert the battery pack supplied voltages to the load's required levels is a common approach in practice. However, the voltage regulators introduce additional energy loss when converting voltages, and the energy loss on regulators increases as the difference between the supplied and required voltages increases. This fact is also reported in [23], [34]. Thus, to optimize the system energy efficiency, it is key to *match the supplied voltage with the load's required voltage as much as possible.*

2) Minimizing the Discharge Current of Individual Batteries: Large discharge current degrades battery performance in many ways, e.g., increasing the internal energy loss [12], reducing the deliverable battery capacity [14], causing significant temperature rise [31], and introducing additional energy overheads due to a higher system monitoring frequency [27]. As a result, the theoretical relationship among the operation time T, the battery capacity Q, and the discharge current I (i.e., $T = \frac{Q}{I}$) is in fact $T < \frac{Q}{I}$ in practice.

To understand the impact of discharge current on battery performance, we conduct a set of measurements as follows. We adopt two series connected and initially fully charged 2450 mAh AA batteries to power several parallel connected motors with an operation voltage of 6 V. In this way, the battery discharge current increases with a larger number of motors. We record the time that the batteries can support the loads, i.e., the operation time, with the motor numbers varying from 1 to 5. The measurement results are shown in Fig. 2. It is intuitive that the operation time decreases with heavier loads. Furthermore, the operation time decreases faster than the increase of loads: normalizing the operation time with one single motor as the unit time 1, the operation time with c parallel motors (and thus a $c\times$ battery discharge current) is smaller than $\frac{1}{c}$. This super-linear decreasing speed of the operation time indicates that it is highly desirable to minimize the battery discharge current to optimize the performance. To support a given load requirement, the battery pack supplied current is normally limited to a certain range; however, we can reduce the discharge current of individual batteries in the pack by optimizing the system configuration.

III. Reconfiguration with Singe-Load-Change

We investigate the adaptive system reconfiguration in the scenario with only a single load change in this section. We first abstract the battery system into a weighted directed graph, with which the problem of identifying the optimal system configuration is transformed to find as many as possible disjoint paths conforming to a given weight requirement. We then show that the problem can be solved in a near-optimal way by a combination of a Depth-First-Search (DFS) with pruning and 0-1 integer programming.

[1] The cutoff voltage generally defines the *empty* state of the battery.

A. Problem Formulation

Given the two design principles described in Sec. II, our problem in identifying the optimal system configuration can be formulated as [2]

$$\begin{aligned} & min \quad \frac{P}{n_p \cdot V_{\text{out}}} \qquad (1) \\ & s.t. \qquad V \leq V_{\text{out}} \leq (1+\sigma)V, \end{aligned}$$

where n_p is the number of series battery strings that are connected in parallel to supply the load requirements and σ is the tolerable jitter voltage ratio. The objective in Equation (2) reflects the design principle in minimizing battery discharge current, and the constraint in Equation (2) is guided by the design principle in matching the supplied and required voltages. Note that in order to provide the corresponding power output P, a current draw of $\frac{P}{V_{\text{out}}}$ from the battery pack is required (we consider the case that it is possible for the battery pack to provide such a current draw in this paper). Because the jitter voltage ratio is normally small for the considerations of both energy efficiency and system safety, e.g, 2.5% in [24], thus with a given P and V, the objective in Equation (2) can be approximated by

$$max \quad n_p. \qquad (2)$$

We need to consider the voltage imbalance issue when the parallel connection of multiple series strings are adopted, which may cause the reverse charging of batteries if the voltage of these strings deviate too much from each other [25]. This imbalance issue in our problem is bounded with the jitter voltage ratio σ in (2).

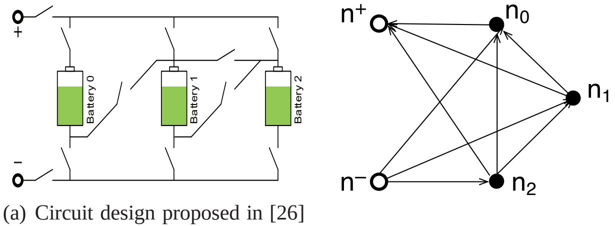
B. Graph Representation

We propose an abstracted graph model for the battery system to facilitate in optimizing its performance. Given a battery pack and the battery voltages at the decision time, we construct a corresponding weighted and directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ in the way that

1) the vertex set $\mathcal{V}$ represents the batteries in the pack, and denote $\mathcal{V} = \{n_1, n_2, \cdots, n_N\}$;
2) the edge set $\mathcal{E}$ represents the configuration flexibility of the pack, i.e., how the batteries can be connected. A directed edge $n_i \rightarrow n_j \in \mathcal{E}$ if and only if the discharge current can flow from n_i to n_j without passing any other batteries;
3) the weight of each vertex is the voltage of the corresponding battery at the decision time: $\forall w_i \in \mathcal{W}, w_i = v_i \ (i = 1, 2, \cdots, N)$.

The above constructed graph only incorporates the batteries in the system. To further include the terminal pairs on which the load has changed into the graph representation, we further extend the graph with the following three steps. First, we add two vertices n^+ and n^- to $\mathcal{V}$, representing the two output terminals. Then to capture the connectivity of the two terminals, we add edge $n^+ \rightarrow n_i$ and $n_i \rightarrow n^- \ (i = 1, 2, \cdots, N)$ to $\mathcal{E}$ if the output terminals can be directly connected to the ith battery. Note that for many existing battery packs, the output terminals can be directed connected to any batteries through backbone buses [26]. In the following, we assume this full connectivity of the output terminals. At last, we extend the weight set $\mathcal{W}$ by setting $w(n^+) = w(n^-) = 0$.

Figure 3 illustrates an example of the extended graph with $N = 3$ based on the battery pack design proposed in [26], as shown in Fig 3(a). With this circuit design, we can draw the current by series connecting battery-0, battery-1, and battery-2. As a result, directed edges $n^- \rightarrow n_2$, $n_2 \rightarrow n_1$, $n_1 \rightarrow n_0$, and $n_0 \rightarrow n^+$ exist in the corresponding graph, as shown in Fig. 3(b) [3].



(a) Circuit design proposed in [26]

(b) Abstracted graph

Fig. 3. An example on the graph representation of battery systems.

The extended graph captures all the potential system configurations, and the out-degree of vertices is a direct metric to quantify the configuration flexibility offered by the system. Note that any specific battery pack can be mapped to only one corresponding graph, while one graph may have multiple battery pack implementations. This is because the edges in the graph only reflect the logical connectivity between batteries, but do not specify how to physically achieve such connectivity.

C. Problem Transformation

With the constructed graph, the problem formulation (2) can be transformed to *identifying the maximal number of disjoint simple paths connecting n^+ and n^- with weight sum in the range of $[V, (1+\sigma)V]$*. Specifically, we say a simple path connecting n^+ and n^- is *feasible* if the weight sum of involved vertices is within $[V, (1+\sigma)V]$. The requirement on the disjoint paths is to avoid involving the same battery in multiple series strings, which increases its discharge current and unbalances the battery utilization.

D. Identifying the Optimal Configuration

We first show that identifying the optimal configuration is NP-hard, then based on two important observations on battery system, we propose a solution which is feasible in practice and achieves the near-optimal performance.

1) NP-hardness:

Theorem 1: *Given a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ and an interval $[V, (1+\sigma)V]$, finding the maximal number of disjoint simple paths in $\mathcal{G}$ with weight sum in $[V, (1+\sigma)V]$ is NP-hard.*

Proof: The decision version of the above problem can be stated as: *given a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$, can we find $n \ (n = 0, 1, 2, \cdots)$ disjoint simple paths in $\mathcal{G}$ with weight sum in $[V, (1+\sigma)V]$?* For the ease of description, denote this

[2] Because we focus on the scenario of a single load change in this section, we remove the superscript representing the ith load for notation convenience.

[3] We use the flow direction of positive charges as the edge direction.

decision problem as Υ. We can prove that the *Longest Path Problem* in graph theory (i.e., finding a simple path of the maximum length in a given graph), which is a classic NP-hard problem, can be reduced to a special case of Υ by assuming: a) $n = 1$ and $\sigma = 0$; b) the weights of all vertices are 1; c) V is an integer larger than 1. If a polynomial time algorithm Ψ exists for the special case of Υ, we can apply Ψ on the Longest Path Problem with path length increasing from 1 to $|\mathcal{V}|$, until no solution can be returned. In this way, we solve the Longest Path Problem in polynomial time, which contradicts with its NP-hardness. As a result, we show that no polynomial time algorithm exists for the special case of Υ, and thus prove its NP-hardness. ■

Our approach to solve the problem consists of two steps: first, we identify all feasible paths in the graph, then we find their largest disjoint subset. Each path in the returned subset represents a series string of the corresponding batteries, and all these strings are connected in parallel to support the load. Although the original problem is NP-hard, our solution is feasible in practice based on two important observations on battery systems.

2) Finding All Feasible Paths: We implement a DFS with pruning method to identify all the feasible paths in the graph. If using the basic DFS idea to identify all the feasible paths, we need a computational time of $\mathcal{O}(N^{N-1})$ to identify all the feasible battery strings (note that we assume fully connected output terminals, and the computation time to identify all the paths is quite different from graph traversal). However, the following two observations on battery systems assist in reducing the computation complexity in practice.

Observation 1: Given a battery pack, the set of supported configurations is limited.

For example, the AHR32113 Power Modules only support 10S(1-4)P connectivity configurations [1] [4]. This is because an higher configuration flexibility is achieved at the cost of significantly increased system implementation complexity. For example, the number of required electronic components increases in a quadric relationship with the number of batteries that a given battery can connect to [23]. With the graph representation, this is reflected by the fact that the vertices out-degree will not be large. Denote the average out-degree of vertices as d, the worst case computational complexity of the DFS search can be reduced to $\mathcal{O}(Nd^{N-1})$, where the item N accounts for starting the search at each of the N vertices, and d^{N-1} accounts for the space to find all the paths starting at a given vertex.

Observation 2: Because all the vertices weights are within range $[v_c, v_f]$, for a given load requirement, the number of vertices involved in a feasible path is limited.

Specifically, the minimal and maximal number of vertices in any feasible paths are $\lceil \frac{V}{v_f} \rceil$ and $\lceil \frac{(1+\sigma)V}{v_c} \rceil$, respectively. With this observation, the DFS only needs to examine a depth of at most $\lceil \frac{(1+\sigma)V}{v_c} \rceil$, and thus the worst case computational complexity can be further reduced to $\mathcal{O}(Nd^{\lceil \frac{V_{\max}}{v_c} \rceil -1})$.

Besides reducing the worst case space complexity, the second observation also assists in pruning the DFS branches and thus reducing the average computational complexity from the following two aspects. First, if the weights of the first i vertices in the current search are too small, it indicates that they cannot be part of any feasible paths. Specifically, denote $p(j)$ $(j = 1, 2, \cdots, i)$ as the first i vertices that have been included in the current search, then

$$\begin{aligned} &\text{if} \quad V - \sum_{j=1}^{i} w(p(j)) > v_f \cdot (\lceil \frac{(1+\sigma)V}{v_c} \rceil - i) \\ &\quad \rightarrow \quad \text{terminate the current search.} \end{aligned} \tag{3}$$

Likewise, the current search can also be terminated if the weight of the first i vertices is too large

$$\begin{aligned} &\text{if} \quad (1+\sigma)V - \sum_{j=1}^{i} w(p(j)) > v_c \cdot (\lceil \frac{V}{v_f} \rceil - i) \\ &\quad \rightarrow \quad \text{terminate the current search.} \end{aligned} \tag{4}$$

With this DFS with pruning, we can practically identify all the feasible paths in the graph.

3) Finding the Largest Set of Disjoint Feasible Paths: Due to the requirement on disjoint paths for a balanced battery utilization, if we include a specific path into the system configuration, other paths with overlapping vertices will not be able to be added to the configuration later. Thus our next step is to find the largest disjoint subset of all these feasible paths. Assuming M feasible paths have been identified, which are denoted as $\mathcal{P} = \{path_1, path_2, \cdots, path_M\}$. We refer two paths as *conflicted* if they share at least one common vertex, specifically, define a 0-1 matrix $A_{M \times N}$ as

$$a_{i,j} = \begin{cases} 1 & \text{if } n_j \in path_i \\ 0 & \text{otherwise.} \end{cases}$$

Then we define a *conflict matrix* $C_{M \times M}$ is defined as

$$C_{M \times M} = \{conflict(i, j)\}.$$

and for any $path_i$ and $path_j$,

$$conflict(i, j) = 1 \Longleftrightarrow \exists k, a_{i,k} \cdot a_{j,k} = 1.$$

Thus our problem in finding the largest set of disjoint paths can be formulated as

$$\begin{aligned} &\max |\mathcal{P}^*| \quad (\mathcal{P}^* \subseteq \mathcal{P}) \\ &s.t. \quad \forall\, path_i, path_j \in \mathcal{P}^* \Rightarrow conflict(i, j) = 0. \end{aligned} \tag{5}$$

where $\mathcal{P}^*$ is the to be obtained largest path set. By further defining

$$x_i = \begin{cases} 1 & \text{if } path_i \in \mathcal{P}^* \\ 0 & \text{otherwise,} \end{cases}$$

we can transform the problem formulation in (5) to

$$\max \sum_{i=1}^{M} x_i \quad s.t. \quad \forall j, \quad \sum_{i=1}^{M} a_{i,j} \cdot x_i \leq 1.$$

This transformed problem is a classic 0-1 integer programming problem. As the DFS with pruning method identifies all the feasible paths in the graph, we can see that the optimality of the identified configuration only depends on how *optimal* a solution the 0-1 integer programming can return.

[4] The notation of αSβP represents that a total number of $\alpha \cdot \beta$ batteries are configured into β parallel connected strings, and each string consists of α series connected batteries.

Fortunately, efficient 0-1 integer programming solvers exist in the literature [13], and thus the near-optimality of the identified configuration can be guaranteed.

IV. Reconfiguration with Multi-Load-Changes

We have investigated the scenario where only a single load changes in the previous section. To complete our investigation, in this section, we extend our design to the scenario where multiple loads may change simultaneously.

A. Problem Formulation

Let U denote the number of changed loads. Specifically, for $i = 1, 2, \cdots, U$, we denote the required voltage and power output as $[V^i, (1+\sigma)V^i]$ and P^i, respectively.

The graph representation of the battery pack can be extended by adding $2U$ vertices in $\mathcal{V}$ and extending the edge set $\mathcal{E}$ and weight set $\mathcal{W}$ in the same way as in the single load change case.

An important difference between the scenarios of a single and multiple loads changes is that the battery discharge currents are likely to be heterogeneous in the latter case. This is because the loads on different terminal pairs are quite likely to be heterogeneous, and the number of series strings for each load in the adopted system configuration may be heterogeneous as well. As a result, we need to *minimize the maximal discharge current of batteries* in this multiple loads changes case to improve the overall battery utilization efficiency.

We can use the same DFS with pruning method as in the single load change scenario to identify all the feasible paths for each load, denoted as $\mathcal{Q} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_U\}$. Define the 0-1 matrix $A_{M\times N}$ in the same way as Equation (5), except that M now is the total number of paths identified for all U loads. We thus have

$$M = \sum_{i=1}^{U} |\mathcal{P}_i|.$$

Define another two 0-1 matrixes $B_{M\times U}$ and $Y_{M\times U}$ as

$$b_{i,k} = \begin{cases} 1 & \text{if } path_i \text{ is feasible for the } k\text{th load} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$y_{i,k} = \begin{cases} 1 & \text{if } path_i \text{ is selected for the } k\text{th load} \\ 0 & \text{otherwise.} \end{cases}$$

Our problem in the multiple load changes scenario can be formulated as

$$\begin{aligned} & \min\max\{\frac{P^k}{V^k \sum_{i=1}^{M} y_{i,k}}\} \ (k \in \{1, 2, \cdots, U\}) \qquad (6) \\ s.t. \quad & \forall i,k, \quad y_{i,k} = 1 \Rightarrow b_{i,k} = 1, \\ & \forall i, \quad \sum_{k=1}^{U} y_{i,k} \le 1, \\ & \forall j, \quad \sum_{i=1}^{M}\sum_{k=1}^{U} a_{i,j}\cdot y_{i,k} \le 1. \end{aligned}$$

The first constraint means that a path can be selected only for the load it can support, the second constraint requires that each of the M paths can be adopted by at most one load, and the final constraint requires the same vertex to be involved in at most one selected path. This *min-max* optimization problem cannot be efficiently solved by 0-1 programming. We thus propose a greedy algorithm to identify the desired system configurations in this case.

B. Greedy Solution

After identifying the path set $\mathcal{Q}$, the greedy solution adds paths into the system configuration in a *step-by-step* manner until no more paths can be added. Because the inclusion of a specific path prevents other paths sharing common vertices from being included in the future, the sequence with which the paths are added into the configuration plays a critical role in determining the final results. Thus, two sub-questions need to be addressed are: (*i*) *which load should be selected to explore for each step*, and (*ii*) *which path should be included into the configuration for the selected load*. Our solution greedily addresses these two sub-questions: we greedily select the load with the largest battery discharge current for each step, and then greedily include the path with the least conflict on other paths into the system configuration.

1) Select the Load with the Largest Current: Given the load requirements and the paths that have been already included into the system configuration, in each step, we can calculate the discharge current of the batteries supporting each load. Specifically, let $n_p^1, n_p^2, \cdots, n_p^U$ denote the number of selected paths for each load. With a small σ, the discharge current of batteries supporting individual loads can be approximated by

$$I^i \approx \frac{P^i}{n_p^i \cdot V^i}. \qquad (7)$$

Then we select the load with the largest discharge current for individual batteries for each step, specifically, the selected load is

$$\arg\max_i \{I^i\}. \qquad (8)$$

2) Select the Path with the Least Conflict: After selecting the load, the next step is to select a feasible path for the load and include the path into the system configuration.

The selection of a specific path prevents other overlapping paths to be selected in the future due to the disjoint paths requirement. Thus, we select the path that has the least negative impact on other paths. We define the conflict matrix C in the same way as (5). When the kth load is selected in a specific step, we select (and include into the configuration) the path according to

$$\arg\min_i \{\sum_{i,j} conflict(i,j) \mid b_{i,k} = 1, \ path_i, path_j \in \mathcal{Q}_a\}, \qquad (9)$$

where $\mathcal{Q}_a \subset \mathcal{Q}$ is the set of paths that are still available for selection before this step.

Note that the inclusion of a specific path into the configuration changes the values of (7), and thus we need to re-select the load to be processed for each step. If no path can be selected for the load under consideration, we mark the load as *saturated*, and re-select the *non-saturated* load with the largest discharge current for individual batteries as the next to process. This process continues until no paths can be selected for any loads.

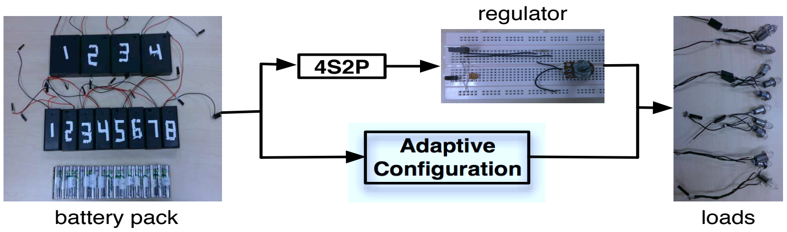


Fig. 4. Experiment method.

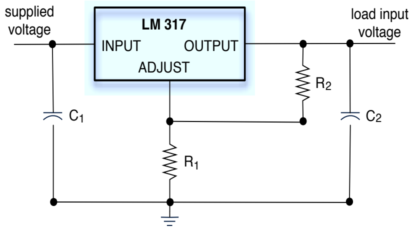
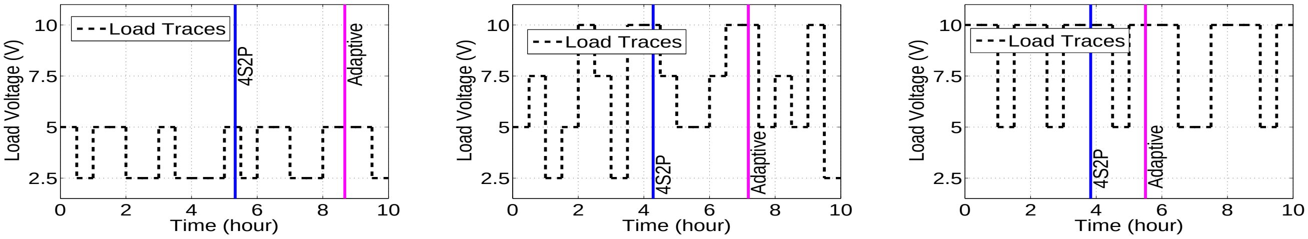


Fig. 5. Adjustable voltage regulator.



(a) Light Loads

(b) Mild Loads

(c) Heavy Loads

Fig. 6. Operation time with different loads.

V. Practical Issues and Further Discussion

Energy Overhead of Reconfiguration Overhead The adaptive system reconfiguration is achieved by the operation of supplementary electronic components such as switches and connectors, which also consumes energy. Because the energy consumption on these supplementary components are normally smaller than the battery capacities in orders of magnitudes, we can separate the identification of optimal configurations from minimizing the operation costs of supplementary components as two independent problems in practice.

Time Overhead of Reconfiguration Besides energy loss, the system reconfiguration also requires certain time duration to accomplish, i.e., the reconfiguration latency. This reconfiguration latency introduces two additional challenges: how to minimize the latency and how to supply the load during the transient phase. A possible approach to address the first question is through the coordinated supplementary component operations, and the second question can be addressed by incorporating super-capacitor based secondary power supply systems.

Battery Charging through System Reconfiguration Battery charging is another important issue for battery-powered systems, which is desired to be fast (i.e., accomplished in short time) and efficient (i.e., more energy is charged into the batteries). It is also possible to apply the idea of system reconfiguration to assist the charging of batteries: the charging current/voltage can be controlled by adjusting the way in which the batteries are connected.

VI. Experiment Evaluations

In this section, we evaluate the proposed adaptive reconfiguration algorithms based on a prototype battery system.

A. Experiment Settings

1) Battery Pack: We build a prototype battery pack with sixteen 2450 mAh AA rechargeable batteries, which are organized into eight modules each with two series connected batteries. Similar to the configuration flexibility offered by the existing off-the-shelf products [1], our prototype supports a configuration set of $\{$1S8P, 2S4P, 4S2P$\}$. For the ease of implementation, the prototype only has a single pair of output terminals, and thus can only support a single load at any time.

2) Loads: We adopt twelve 0.5 A 3.6 W flashlight bulbs as the loads, which are organized into four 1S3P bulb modules. We randomly generate load traces in the form of $\{t_1, a_1\}, \{t_2, a_2\}, \cdots$, where t_i is the lasting duration and a_i is the number of series connected bulb modules during that time ($a_i \in \{1, 2, 3, 4\}$). Because the bulbs can operate under a wide voltage range (but with different lightness), for a fair comparison, we impose an additional requirement that each bulb module requires a 2.5 V input voltage, and the tolerable jitter voltage is 1 V. As a result, the load required voltages vary from $2.5 \times 1 = 2.5$ V to $2.5 \times 4 = 10$ V.

3) Baseline: We adopt the 4S2P battery configuration as a baseline, which trades-off between the highest voltage the system can supply (i.e., 4S) and the preference on small battery discharge current (i.e., 2P). To match the supplied voltage with the load requirement, we implement an adjustable voltage regulator with the LM317 IC [6], and connect it between the battery pack and the load to adjust the load input voltage to the required level. Figure 4 presents an overview of our experiment methods, and the circuit diagram of the adjustable voltage regulator is shown in Fig 5.

4) Evaluation Metric: We adopt the operation time, defined as the time from the start of the measurement to the time when the bulbs cannot be lightened anymore, as the metric to evaluate the system performance. The batteries are initially fully charged for each measurement.

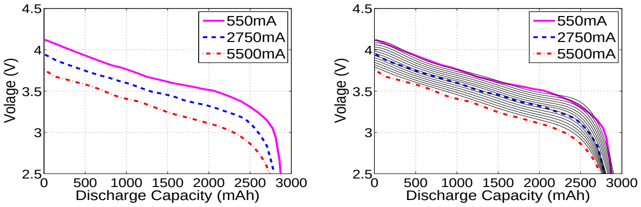
B. Experiment Results

To investigate the performance of the adaptive configuration algorithm with different load conditions, we randomly generate three load traces with light, mild, and heavy loads respectively. Specifically, with light load, only 1-2 bulb modules are series connected as the load. The numbers of bulb modules adopted with the mild and heavy loads are randomly chosen from 1-4 and 2-4, respectively. In this way, the number of bulb modules used in the light, mild, and heavy loads are 1.5, 2.5, and 3 in average, respectively. The load lasting time t_i is set to 30 minutes.

The operation time obtained with the two configuration methods for each load trace are shown in Fig. 6. The advantage of the adaptive configuration over the baseline is obvious, and an average operation time increase of 3.06 hour is obtained over the three loads conditions. This operation time improvement is due to two reasons: first, by adaptively converting the supplied voltages to the load required levels, the energy loss on the voltage regulator is reduced; second, by minimizing the discharge current of individual batteries, more battery capacity can be delivered and the heat dissipation on other system components is also reduced.

Furthermore, we can observe that the lighter the loads, the more improvement can be obtained, which can be explained by the following two facts. First, with the 4S2P configuration, the lighter the loads, the larger the gap between the supplied and required voltages, which degrades the regulator efficiency. Second, the lighter the loads, the fewer batteries are needed to form a single series string to support the loads. This in turn offers more space for the adaptive configuration to identify more parallel connected strings, and thus reduces the battery discharge current.

Another observation from our experiment result is that with the 4S2P configuration, the temperature of the LM 317 IC easily rises to $44°C$ at the maximum. Such a high temperature not only indicates significant energy loss (and thus supports our design principle in matching supplied and required voltages), but also reduces the system stability.



(a) Provided in [7]

(b) Approximated

Fig. 7. Battery discharge traces.

VII. Simulation/Emulation Evaluations

In this section, we evaluate the proposed adaptive reconfiguration algorithms through extensive trace-based simulations. We first evaluate the adaptive reconfiguration based on battery discharge traces obtained from the data sheet of *off-the-shelf* battery products. Then, the efficiency of the adaptive reconfiguration is further verified base on two sets of electric vehicle traces collected during driving.

A. Simulation based on Battery Discharge Traces

1) Trace-based Battery Model: Analytical modeling of battery properties is computational expensive, and thus we use a trace-based method to track the battery states. We simulate a battery pack consisting of $2,900$ mAh Panasonic NCR18650 Li-ion batteries [7], whose discharge curves with discharge currents of 550 mA, $2,750$ mA, and $5,500$ mA are provided in its data sheet. The full and cutoff voltages are $v_f = 4.25$ V and $v_c = 2.5$ V, respectively.

To obtain more fine grained battery discharge traces, we divide the current interval $[550, 5500]$ mA into 99 intervals with a gap of 50 mA each, and proportionally approximate the corresponding discharge traces based on the three traces provided in the data sheet. Note that we could further reduce the current gap to improve the approximation accuracy. A subset of the obtained discharge curves is shown in Fig. 7 (not all the curves are shown for figure clarity).

2) Simulated Battery Packs: The simulated battery pack consists of 64 batteries and can support three loads simultaneously. The current drawn from each battery can directly reach two other batteries on average (i.e., an average vertices out-degree of two in the graph). The simulation follow these settings unless otherwise specified. The initial battery voltages are randomly generated in the range of

$$[\alpha \cdot v_c, \quad v_f], \tag{10}$$

where α is a control parameter that determines the battery voltage diversity and is set to 1.2 unless otherwise specified.

3) Load Traces: Similar to the traces in the experiment, we randomly generate load traces in the form of $\{t_i^j, V_i^j, P_i^j\}$ for each loads, where t_i^j is the lasting duration of the ith trace for the jth load, and V_i^j and P_i^j are the required voltage and power of that trace, respectively. A unit time interval of 10 minutes is adopted for loads' lasting time, i.e., t_i only takes the values of 10 min, 20 min, and so on. The system configuration is updated every 10 minutes by first updating the battery voltages according to the traces in Fig. 7, and then adaptively reconfiguring the battery pack. The required voltage V is randomly generated from 15-20 V unless otherwise specified. The tolerable jitter voltage is 2.5 V (i.e., v_c) by default. The required power P is randomly generated from $15\ V \times 550\ mA = 8.25\ W$ to $20\ V \times 5500\ mA = 110\ W$.

4) Baselines: We implement the following three system configurations as baselines.

<u>Serial</u>: The batteries are evenly assigned to individual loads, and for each load, the assigned batteries are series connected.

<u>Parallel</u>: $\sqrt{N}$ series strings with $\sqrt{N}$ batteries each are formed, and then these strings are assigned to individual loads in a round-robin manner.

<u>Oracle</u>: Given the range of all possible load required voltage, we can calculate how many batteries are needed for a string to be able to support any potential loads, and then we form as many as possible such strings. The remaining batteries (those are not enough to construct another such string) form the last string. This oracle configuration maximizes the number of parallel connected strings while guaranteeing each string is able to support the load requirement. These strings are then assigned to individual loads in a round-robin manner.

5) Performance Evaluation: We evaluate the impact of different system parameters on the performance of adaptive reconfiguration.

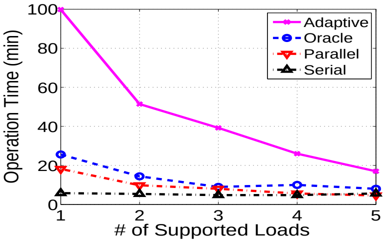


Fig. 8. Impact of supported loads #.

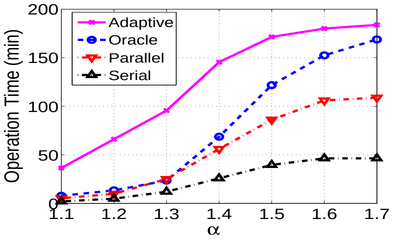


Fig. 9. Impact of initial battery voltages.

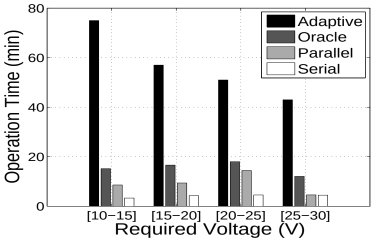


Fig. 10. Impact of required voltage.

TABLE I
THREE CASES OF CONFIGURATION FLEXIBILITY DISTRIBUTIONS.

Vertices Out-degree	1	2	3	4	5
Case A	0.5	0	0	0	0.5
Case B	0.2	0.2	0.2	0.2	0.2
Case C	0	0	1	0	0

Number of Loads: We first examine the impact of the number of supported loads on the system operation time. The results with 1 to 5 simultaneously supported loads are shown in Fig. 8. Note that it is not necessary for all these loads to change at the same time. The operation time decreases as more loads have to be supported, which is intuitive because increasing the number of supported loads essentially indicates a heaver loads for the battery system. Furthermore, we can see that compared with the baselines, the adaptive reconfiguration achieves around $4\times$ gain in operation time when only one load is supported. Although the gain decreases as more loads need to be supported, the adaptive reconfiguration still obtains a $2\times$ operation time even with a load number of 5.

Battery Voltage Diversities: We then explore the impact of battery voltages on the operation time. The battery voltages are controlled by the parameter α in (10), and a smaller α indicates both a higher voltage diversity among batteries and a lower average battery voltages. The operation time with α varying from 1.1 to 1.7 are shown in Fig. 9. Note that with $\alpha = 1.7$, it indicates that the battery voltages are randomly generated from $[2.5 \times 1.7, 4.25] = \{4.25\}$, meaning all the batteries are initially fully charged. The operation time increases as α increases because of higher initial battery voltages, and the adaptive reconfiguration outperforms the baselines in all the explored cases. Furthermore, we can see the advantage of adaptive reconfiguration is more obvious with smaller α. This is because the smaller α is, the more likely that certain batteries are close to depletion. Depleted batteries significantly degrade the system performance if the configuration is not adjustable. On the other hand, the adaptive reconfiguration can bypass low-voltage batteries when necessary, which reduces the impact of the near-depletion batteries on system performance.

Loads Requirements: The operation time with different load required voltages and powers are shown in Fig. 10 and Fig. 11, respectively. The load required power is set to 55 W in Fig. 10 and the required voltage is 20 V in Fig. 11. Again, significant improvement on the operation time can be observed with the adaptive reconfiguration, which is about $3\times$-$5\times$ of those obtained with the baselines. Furthermore, the operation time decreases as the loads become heavier, as a result of the increase in either the required voltage or power. Note that Fig. 10 shows that when the required voltage is relatively low (e.g., 10–25 V), the operation time obtained with the non-reconfigurable baselines slightly increases with a higher voltage. This is because when the required voltage increases, the load current decreases with a fixed required power, which in turn leads to a longer operation time. However, as the required voltage continuously increases (e.g., $[25, 30]$), the probability for the fixed configuration to not able to support such required voltage increases, and thus the operation time is reduced.

Configuration Flexibilities: We investigate the impact of system configuration flexibility on the adaptive reconfiguration in the following. The operation time with different configuration flexibilities (i.e., the vertices out-degree in the graph) are shown in Fig. 12, with an average out-degree of 1, 2, and 3, respectively. Significant increase in system operation time can be observed when the configuration flexibility increases. However, the increase in operation time slows down with larger average out-degrees. This implies that an excessive high configuration flexibility may not be desirable, especially when considering the fact that the configuration flexibility does not come without costs, e.g., the implementation complexity would be dramatically increased.

Besides the average configuration flexibility, the distribution pattern of these flexibilities also affects the system performance. Fixing the average vertices out-degree as 3, we explore three cases on how the configuration flexibilities are distributed among batteries, as shown in Table I. The resulting operation times under these three cases are shown in Fig. 13. We can see that the flexibility distribution significantly affects the performance. An important observation is that when the configuration flexibility is more evenly distributed among batteries, the performance becomes better. This is because such an evenly distribution pattern alleviates the negative impact due to bottleneck batteries on the system performance. This observation serves as a guidance in practical battery system design.

B. Emulation based on Electric Vehicle Driving Traces

We further evaluate the adaptive reconfiguration based on empirical electric vehicle driving traces. We collect two driving traces of around 900 s and 2400 s each, containing the corresponding operation voltages and powers during that time period, as shown in Fig. 14 and Fig. 15, respectively.

We generate the load traces for our emulation based on these two raw traces. First, both the discharging and charging of battery pack happen during the driving of electric vehicles.

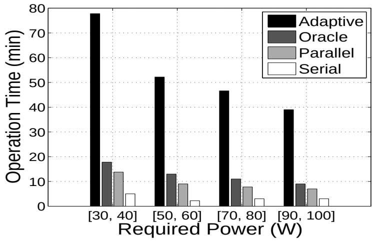


Fig. 11. Impact of required power.

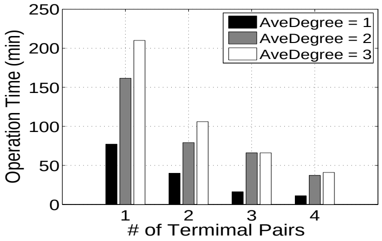


Fig. 12. Impact of ave. configuration flexibility.

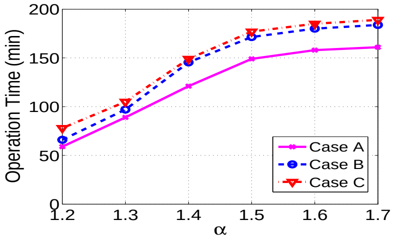


Fig. 13. Impact of flexibility distributions.

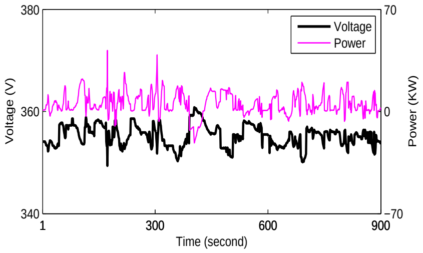


Fig. 14. Electric vehicle driving traces #1.

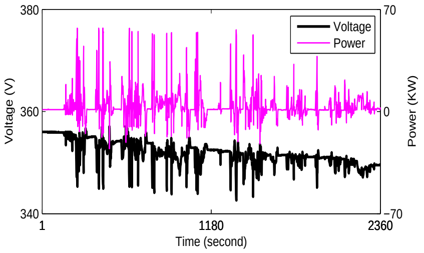


Fig. 15. Electric vehicle driving traces #2.

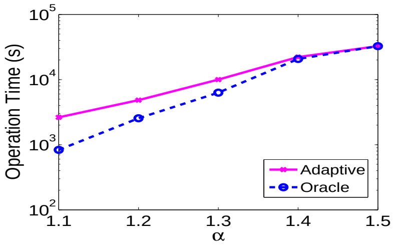


Fig. 16. Operation time.

This is reflected in the traces that the both positive (i.e., discharging of battery pack) and negative (i.e., charging of battery pack) values exist in the power trace. Because we only focus on the discharge management in our design, we set all the negative powers and the voltages at the corresponding time instance in the traces to zeros. This can be interpreted as no load is imposed on the battery pack during those time periods. Second, because the raw traces are relatively short in time, we combine them sequentially to form a $900 + 2360 = 3260$ s trace, and then repeat it for 10 time. As a result, a $\frac{3260\times10}{60} \approx 550$ minutes load trace is generated for our emulation.

The battery packs for electric vehicles normally take the hierarchical architecture: it can be divided into a set of battery modules which in turn are consisted of individual batteries. In our emulation, we form a battery pack consisting of 64 modules each with $16S4P$ connected batteries. The battery discharge property conforms to the discharge traces shown in Fig. 7.

Again, we take the non-reconfigurable Oracle baseline for comparison. The operation time with varying initial battery voltages (i.e., by adjusting α in (10)) is shown in Fig. 16. Obvious advantage of the adaptive reconfiguration can be observed, especially when the battery initial voltages are low, which agrees with the observation in Fig. 9. The operation time obtained with the two configuration methods converge as α increases, because in this case the battery pack has sufficient energy supply to survive the load even without the assistance of adaptive reconfiguration.

VIII. Related Work

Large-scale battery systems are commonly adopted in practice, and many research efforts have been devoted to improve the system performance focusing on the battery discharge scheduling [11], [14], [17], [25], [32], the effective system monitoring [27], the design of battery management systems [16], [30], [38], etc.

Due to the load dynamics in large-scale battery systems, traditionally, the battery supplied voltage is adjusted to the load required level by adopting additional electronic components, e.g., voltage regulators [41]. However, the additional components introduces additional energy consumption/loss [5], [22], and thus degrades the battery energy utilization efficiency.

Another approach to provide the dynamic load requirement is to adaptively adjust the battery connections in the system. Investigations on this adaptive system reconfiguration have been reported in [20], [41], targeting on small multicell battery systems such as mobile devices. In our work, we extend the investigation to large-scale scale battery systems.

Two necessary conditions must be satisfied to effectively and adaptively reconfigure the system. First, the system has to offer certain configuration flexibility on which the adaptive reconfiguration can operate. However, the system configuration flexibility is achieved by adopting more electronic components such as connectors and switches, which not only introduces additional energy costs, but also increases the system implementation complexity. Research efforts have been devoted to effectively offer configuration flexibility with less additional costs [10], [23], [24]. Based on the system design in [24], six switches are enough to connect a battery in any manner: series, parallel, or by-passed. Our work advances the state-of-the-art by proposing adaptive reconfiguration algorithms that return the desired system configuration based on the offered configuration flexibility and the real time load requirements.

Second, the system has to be aware of individual battery conditions to carry out the adaptive reconfiguration. Many works on battery modeling and simulation exist in the literature [15], [18], [21], [37], [38], [40]. However, most of these models are computational extensive, and the simulators require practical parameters to implement. Furthermore, most of these models/simulators are for specific battery types, and thus their universalities are limited. Our proposed adaptive reconfiguration algorithms hide the complex low level battery properties by focusing on two rules of thumb in identifying the desired system configurations: matching the supplied and required voltages and minimizing the discharge currents.

The most similar works are [19], [23], [28]. In [23], a *power tree* representation of the battery pack is proposed to assist the effective system reconfiguration when the battery failures happen. We tackle the system reconfiguration with a different objective, i.e., optimizing the system energy efficiency, and our solutions can also effectively handle the case with battery failures. An optimization formulation w.r.t the energy efficiency is presented in [19], which requires low level battery properties such as the *state of charge*, the *state of health*, etc. Our work hides the complex battery properties from engineering and thus facilitates its practical implementation. A reconfigurable series-connected battery string is proposed in [28] to adjust the supplied voltage to the load required level. We advance the investigation by further exploring minimizing the battery discharge current to improve the system energy efficiency.

IX. Conclusions

In this paper, we have explored the adaptive reconfiguration of large-scale battery systems to dynamically provide the load's required voltages, which avoids the low efficiency issue of the traditional voltage regulator-based solutions. Based on two empirically observed design principles, our approach hides the complex battery properties from engineering, and thus makes it more practical for implementation. Specifically, by abstracting the battery system into a graph representation, we have investigated both scenarios with a single and multiple load changes, and proposed corresponding adaptive reconfiguration algorithms. Through prototype implementation and extensive simulation, we have shown the proposed adaptive reconfiguration algorithms can significantly improve the performance w.r.t. system operation time. In the future, we will investigate the trade-off between the system energy efficiency and implementation complexity.

Acknowledgement

This work was supported by Singapore-MIT IDC IDD61000102a, SUTD-ZJU/RES/03/2011, NRF2012EWT-EIRP002-045, US National Science Foundation (NSF) grants CNS-1117438, a start-up grant from the University of Texas at Dallas, and NSFC-61303202.

References

[1] AHR32113 Power Modules. http://www.a123systems.com/products-modules-power.htm.

[2] ARPA-E. http://www.greencarcongress.com/2012/08/arpae-20120802.html.

[3] Electric Car Motor. http://www.electric-cars-are-for-girls.com/electric-car-motor.html.

[4] Elithion Battery Pack. http://elithion.com/battery_packs.php.

[5] Energy Loss on Regulators. http://www.dimensionengineering.com/info/switching-regulators.

[6] LM317 Chip. http://www.ee.buffalo.edu/courses/elab/LM117.pdf.

[7] Panasonic NCR18650 Li-ion Battery. http://industrial.panasonic.com/www-data/pdf2/ACA4000/ACA4000CE240.pdf.

[8] Reconfigurable Battery Packs. http://arpa-e.energy.gov/q=arpa-e-projects/reconfigurable-battery-packs.

[9] Synchronized Resource Service. http://www.pjm.com/markets-and-operations/ancillary-services/synchronized-service.aspx.

[10] M. Alahmad, H. Hess, M. Mojarradi, W. West, and J. Whitacre. Battery switch array system with application for JPL's rechargeable micro-scale batteries. *Journal of Power Sources*, 177(2):566 – 578, 2008.

[11] L. Benini, G. Castelli, A. Macii, B. Macii, and R. Scarai. Battery-driven dynamic power management of portable systems. In *ISSS'00*, 2000.

[12] L. Benini, D. Vruni, A. Macii, E. Macii, and M. Poncino. Discharge current steering for battery lifetime optimization. *IEEE Transactions on Computers*, 52(8):985–995, 2003.

[13] S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge University Press*, 2004.

[14] G. Castelli, A. Macii, E. Macii, and M. Poncino. Current-controlled policies for battery-driven dynamic power management. In *ICECS'01*, 2001.

[15] G. Ceder, M. Doyle, P. Arora, and Y. Fuentes. Computational modeling and simulation for rechargeable batteries. *MRS BULLETIN*, pages 619–623, 2002.

[16] C. F. Chiasserini and R. Rao. Energy efficient battery management. *IEEE Journal on Selected Areas in Communications*, 19(7):1235–1245, 2001.

[17] C. F. Chiasserini and R. Rao. Improving battery performance by using traffic shaping techniques. *IEEE Journal on Selected Areas in Communications*, 19:1385–1394, 2001.

[18] P. Chou, C. Park, J. Park, K. Pham, and J. Liu. B#: a battery emulator and power profiling instrument. In *ISLPED'03*, 2003.

[19] S. Ci, J. Zhang, H. Sharif, and M. Alahmad. Dynamic recongurable multi-cell battery: A novel approach to improve battery performance. In *APEC'12*, 2012.

[20] S. Ci, J. Zhang, H. Sharif, and M. Alahmadu. A novel design of adaptive reconfigurable multiple battery for power-aware embedded networked sensing systems. In *GLOBECOM'12*, 2012.

[21] L. Cloth, M. Jongerden, and B. Haverkort. Computing battery lifetime distributions. In *DSN'07*, 2007.

[22] D. Graovac and M. Purschel. IGBT power losses calculation using the data-sheet parameters. *Infineon Application Note*, pages 1–16, 2009.

[23] F. Jin and K. G. Shin. Pack sizing and reconfiguration for management of large-scale batteries. In *ICCPS'12*, 2012.

[24] H. Kim and K. G. Shin. On dynamic reconfiguration of a large-scale battery system. In *RTAS'09*, 2009.

[25] H. Kim and K. G. Shin. Scheduling of battery charge, discharge, and rest. In *RTSS'09*, 2009.

[26] H. Kim and K. G. Shin. Dependable, efficient, scalable architecture for management of large-scale batteries. In *ICCPS'10*, 2010.

[27] H. Kim and K. G. Shin. Efficient sensing matters a lot for large-scale batteries. In *ICCPS'11*, 2011.

[28] T. Kim, W. Qiao, and L. Qu. Series-connected self-reconfigurable multicell battery. In *APEC'11*, 2011.

[29] T. Kim, W. Qiao, and L. Qu. A series-connected self-reconfigurable multicell battery capable of safe and effective charging/discharging and balancing operations. In *APEC'12*, 2012.

[30] D. Lim and A. Anbuky. A distributed industrial battery management network. *ACM Trans. on Indus. Elec.*, 51(6):1181–1193, 2004.

[31] D. Linden and T. B. Reddy. Handbook of Baterries (3rd ed.). *McGraw-Hill*, 2001.

[32] J. Luo and N. Jha. Battery-aware static scheduling for distributed real-time embedded systems. In *DAC'01*, 2001.

[33] J. Markoff. Pursuing a battery so electric vehicles can go the extra miles. http://www.nytimes.com/2009/09/15/science/15batt.html.

[34] Maxim. Source resistance: the efficiency killer in DC-DC converter circuits. http://www.maxim-ic.com.

[35] S. McLaughlin, P. McDaniel, and W. Aiello. Protecting consumer privacy from electric load monitoring. In *CCS'11*, 2011.

[36] G. Motors. Latest chevy volt battery pack and generator details and clarifications. http://gm-volt.com/2007/08/29/latest-chevy-volt-battery-pack-and-generator-details-and-clarifications/.

[37] S. J. Moura, J. L. Stein, and H. K. Fathy. Battery-health conscious power management in plug-in hybrid electric vehicles via electrochemical modeling and stochastic control. *IEEE Transactions on Control Systems Technology*, 21(3):679–694, 2013.

[38] D. Rakhmatov and S. Vrudhula. Energy management for battery-powered embedded systems. *ACM Transactions on Embedded Computing Systems*, 2:277–324, 2003.

[39] D. Raychev, Y. Li, and W. Shi. The seventh cell of a six-cell battery. In *WEED'11*, 2011.

[40] P. Rong and M. Pedram. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *IEEE Transactions on Very Large Scale Integration Systems*, 14(5):441–451, 2006.

[41] H. Visairo and P. Kumar. A reconfigurable battery pack for improving power conversion efficiency in portable devices. In *ICCDCS'08*, 2008.

[42] W. Yang, N. Li, Y. Qi, W. Qardaji, S. McLaughlin, and P. McDaniel. Minimizing private data disclosures in the smart grid. In *CCS'12*, 2012.