Dependable, Efficient, Scalable Architecture for Management of Large-scale Batteries

Hahnsang Kim and Kang G. Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2121, U.S.A.
Email: {hahnsang, kgshin}@eecs.umich.edu

***Abstract*— Conventional battery management systems (BMSs) for electric vehicles (EVs) are designed in an ad hoc way, causing the supply of EVs to fall behind the market demand. A well-designed, combined hardware-software architecture is essential for the management of a large-scale battery pack that consists of thousands of battery cells as in Tesla Motors and Chevy Volt. We propose a *dePendable, effIcient, Scalable Architecture* (PISA) that effectively monitors a large number of battery cells, efficiently controls and reconfigures, if needed, their connection arrangement. PISA is *monarchy-based* and supports hierarchical, autonomous management of battery cells, where a *global* BMS orchestrates a group of *local* BMSs. A local controller on each local BMS autonomously manages an array of battery cells, and the global controller reconfigures the connectivity of such battery-cell arrays in coordination with the local controllers. Configuration of a battery system is controlled by three types of switch—called P-, S-, and B-switches—and an algorithm that configures these switches. Our evaluation results show that PISA effectively tolerates battery-cell failures 10 times—while achieving service cost savings 7.4 times—more than a conventional BMS. This superior performance not only extends the battery life significantly, but also provides the flexibility in supporting diverse electric power demands from a growing number of on-board applications.**

I. Introduction

The global temperature in February 2009 was the 9th warmest by 0.90 degree F above the 20th century mean of 53.9 degrees F [4]. The increase in the global temperature is largely due to greenhouse gas emissions. These emissions could be reduced substantially by the conversion of gasoline combustion vehicles into electric vehicles (EVs), such as hybrid, plug-in hybrid, and battery EVs. For instance, the replacement of 77% of all transport miles with EVs will reduce carbon intensity by 94% over the 1990 numbers [25]. Also, due to soaring fuel prices, EVs are gaining popularity in the global market. To meet this global market demand, we need battery technologies that will make EVs cost- and efficiency-competitive with today's gasoline-powered vehicles.

Cost-effective EVs require not only development of high energy-density battery cells, but also efficient management of large-scale battery packs, each consisting of a large number of battery cells for EVs, e.g., 6,800 lithium-ion battery cells for Tesla Motors' EVs [12] and hundreds of cells for GM 2010 Chevy Volt [16]. In particular, a battery management system (BMS) that monitors and controls battery cells in a pack, must cope with heterogeneous battery-cell characteristics. That is, even if characteristics of all battery cells in a battery pack are initially identical, as they are charged and discharged repeatedly, each cell will exhibit different characteristics. A *weak* cell—that is (charged and/or) discharged faster than others—is likely to be (*over-charged* and/or) *deep-discharged*, i.e., the battery cell continues to be discharged even when its terminal voltage falls below a certain threshold called a *cutoff voltage*. This weak battery cell can eventually become faulty, and if not managed properly, will cause the whole pack to be dysfunctional.

A BMS should be able to cope with weak/faulty cells in such a way that faulty cells are bypassed to keep the pack operational. Bypassing certain cells inside the pack, however, requires switches by which the arrangement of battery cells can be changed as described in [2, 5, 9, 22, 24]. Switches are placed around battery cells, regulating the battery supply power. Furthermore, the reconfigurable battery system we developed earlier [9] offers a way to alter battery connectivity and dynamically adjust supply power to meet application demands. All of these systems require careful system specification, cost-effective incorporation and control of system components like switches and battery cells.

One can conceive two types of battery management architecture: *flat* and *modular*. In the former, a single control module is responsible for monitoring and controlling all components. This architecture is easy to implement, but does not scale well; as the number of components to be monitored and controlled increases, the architectural complexity (e.g., wiring) and management latency grow rapidly. It is not energy-efficient, either. By contrast, in the modular architecture, an individual control module is only responsible for a subset of components independently or in cooperation. A prototypical design of a modular system is presented in [21], consisting of four modules, each of which monitors a series-chain of battery cells. In such a distributed scheme, monitoring is more efficient, and energy-efficiency is higher than the flat architecture. The cost of components, however, increases, and the nature of the battery system requires a global module to orchestrate the others. A mere hierarchical system is neither effective nor efficient in monitoring and controlling battery dynamics. Therefore, we need smart management that makes the most of a reconfigurable framework. This synergetic integration will maximize system performance and reliability at minimum cost.

There are two main challenges in developing a smart battery management architecture. First, there is a tradeoff between the minimum number of hardware components to use and maximum reconfigurability in a BMS. Key components therein are switches that allow a *battery-cell array* to be reconfigurable. The more switches around cells, the more reconfigurable the array becomes, but the costlier. Also, individual components affect directly system reliability. System reliability should be assessed based on the reliability of components and their connections. At the same time, since the cost is the major consideration in realizing a reconfigurable architecture, the components count should be minimized. Second, to maximize both system reconfigurability and reliability, a reconfigurable architecture should be specified with respect to software/hardware components and their inter-relationship. An application (software) may require various battery (hardware) conditions from a BMS. Also, a BMS may request subsystem/local BMSs, if any, for the information on the status of individual battery cells in the case of modular management architecture. Upon receipt of this request, individual local BMSs periodically monitor their battery-cell arrays and reconfigure them, if necessary, in accordance with individual cell characteristics. This interaction between local BMSs also depends upon the underlying hardware system design. A well-designed, hardware-software architecture will provide high dependability, cost-effectiveness, and scalability.

We propose such an integration architecture for the management of a large-scale battery pack, called the *dePendable, efIcient, Scalable Architecture* (PISA). Within PISA, BMSs are differentiated according to their roles and thus classified into a single *global* BMS and multiple *local* BMSs. The local BMS consists of a local controller, a set of switches (referred to as *array-level* switches), and a battery-cell array that includes a group of battery cells and a set of switches (referred to as *cell-level* switches). The global BMS, on the other hand, mainly has a global controller. The global controller 'speaks' to the local controllers so as to configure the array-level switches, based on a switch-configuration algorithm, while individual local controllers configure cell-level switches within their battery-cell array. The relationship between the global and local BMSs is considered *monarchy-based*. That is, the local BMSs govern their battery-cell array autonomously, while the global BMS controls array-level switches via local BMSs as needed. This hierarchical arrangement facilitates (1) switch-configuration management, i.e., the policy for cell-level arrangement (by the local BMSs) is applied to array-level arrangement (by the global BMS), (2) the achievement of system scalability, i.e., effectively coping with large-scale battery cells, and (3) the improvement of power savings for the entire system by putting idle local BMSs into sleep mode.

The main contributions of this paper are three-fold. First, PISA is designed to use the minimum number of switches while achieving reconfigurability that the battery-cell array (array-level) and battery cells therein (cell-level) can be rearranged online in parallel or in series while bypassing any battery cells or battery-cell arrays. Second, PISA achieves scalability for a large-scale battery pack while providing a systematic switch configuration algorithm. This tightly-coupled system provides synergetic performance typical of cyber-physical systems. Third, analytical results give a physical insight into the durability of switches, system reliability, system scalability, and service-cost savings. In particular, a fraction of the charge current load imposed on individual switches varies with cell-level or array-level arrangement. This different fraction dictates different switch lifetimes. The proper choice of switches for different requirements greatly enhances the system reliability with respect to the battery lifetime, achieving service-cost savings.

The rest of the paper is organized as follows. Section II describes PISA we propose, consisting of global and local BMSs, and other key components including switches. We also present configuration commands and an algorithm to control switches. Section III presents the analysis of stress on each switch with respect to cell-level and array-level arrangements. Section IV presents a cost model for a battery-cell array with various battery and switch faults specified. Section V evaluates the performance of PISA. We discuss the related work in Section VI and conclude the paper in Section VII.

II. PISA

This section describes the architecture of PISA and then details its components.

A. Overall Architecture

PISA, whose high-level view is shown in Fig. 1, consists mainly of a global BMS and multiple local BMSs. The global BMS is formed by a global controller, a current meter, and an SMBus (System Management Bus [7]) global. A local BMS is formed by a local controller, a set of *array-level* switches (i.e., P-, S-, and B-switches), a voltage sensor, a battery-cell array, and an SMBus local. Each local BMS monitors battery conditions, e.g., voltage, temperature, *State-of-Charge* (SoC), and *State-of-Health* (SoH), and is connected to the global BMS via their SMBus.

The relationship between the global and local BMSs is *monarchy-based*. The global controller determines a array-level arrangement via a switch-configuration algorithm presented in Fig. 2. The local controllers execute command codes listed in Table I corresponding to the arrangement directed by the global controller. The local controllers are also able to determine their cell-level arrangement autonomously and which cell to be bypassed. In the monarchy-based structure, each local controller is responsible for monitoring its cells and responding to the global controller's interrogation. The monarchy-based structure is scalable to a large-scale battery pack by sharing with local controllers the tasks of arranging, monitoring, and scheduling.

Alternatively, the relationship between the global and local controllers can be viewed as *fully-centralized*, as the global controller determines both cell- and array-level arrangements. In the fully-centralized structure, a local controller's role is minimum and hence unneeded; the global controller may directly monitor individual battery cells, bypass some cells, and perform load-balancing for discharge, cell-balancing for charge, or voltage-balancing for both. In this sense, the fully-centralized structure is agile of detecting and preventing an anomaly caused by some cells. This structure, however, can quickly be overwhelmed by a large number of battery cells to manage. The more the cells, the longer the time to monitor them. Therefore, the monarchy-based structure may become more agile than the fully-centralized beyond a certain number of battery cells.

In what follows are detailed the design of a battery-cell array and the characteristics of switches, a current meter, and an SMBus.

1) Battery-Cell Array: The design of a battery-cell array is dictated by the tradeoff between the low cost and the high reconfigurability, both of which increase monotonically with the number of switches used. By reconfigurability, we mean the capability of bypassing any specific battery cell, enabling effective *voltage-balancing* via selective discharge or charge of cells and extending the pack's operation-time even in the event of random cell failures. Besides, this capability will extend to two types of reconfigurability. First, all cells in an array can be connected in parallel or in series. When they are connected in parallel (series), the array's capacity (terminal voltage) will be a cell's capacity (voltage) × the number of cells. Second, any individual cell can be charged separately, which is important for *cell balancing*.

The number of switches required is determined based on arrangements. In the series arrangement, some weak (short-circuited) cell may have little effect on the array's current except that the overall terminal voltage may drop by no more than the weak cell's voltage. Some weak (open-circuited) cell, however, can block the flow, making the series-chain of battery cells unusable. To bypass the weak cell, two switches, i.e., B- and S-switches, are placed as shown in the right side of Fig. 1. In the parallel arrangement, a single short-circuited cell causes the other cells to be unusable. In this case, a single switch, i.e., P-switch, can make this circuit open. Consequently, three switches per cell are sufficient for *any* cell to be bypassed in both parallel and series arrangements. Certainly, the more customized the arrangement, the smaller the total number of switches.

When it comes to a switch failure, we adopt the *stuck-at fault model* which has been widely used in the literature. In this model, a faulty switch stays in a state permanently, either ON or OFF state, irrespective of inputs to the switch. How to detect switch failures, however, is outside the scope of the paper. Instead, we assume that the local and the global controllers are capable of detecting these switch failures in a timely manner. Based on this fault model, the reliability of the entire battery pack will be analyzed in Section IV.

2) Switch characteristics: Reliable, robust switches are essential to endure high voltages and currents. Generally speaking, three switch modes exist: switch-on, switch-off, and operation. In the switch-on mode, a switch is conductive and must have low impedance, e.g., in the range of milli ohms. The lower the impedance, the lower the power dissipation, and hence the lower the radiative heat. In the switch-off mode, a switch is resistive and must interrupt high currents. High voltages, conversely, can create a damaging *electric arc*, particularly to mechanical contacts. In the operation mode, a switch is in a transition state from the switch-on mode to the switch-off mode, or vice versa. A high voltage that can create an electric arc, possibly destroys the switch circuit.

Accordingly, the selection of proper switches is of great importance. Two types of switches can meet our need: semiconductors and electromechanical relays. In the switch-on mode, a switch's impedance is a key selection criterion. MOSFETs or IGBTs of semiconductors may have relatively low impedance, so that they may not be inappropriate for high current applications [**?**]. For instance, in MOSFETs, given the impedance of 5 milli ohms, a high current of 100 Amperes creates 50 Watts of power dissipation ($P = I^2 \times R$). Conversely, electromechanical relays are known to have a contact resistance on the order of magnitude lower than semiconductors, possibly decreasing down to 0.5 Watts of power dissipation. In the switch-off or operation mode, electromechanical relays can be weakened or even destroyed by significant arcs due to high voltages, while such an effect disappears in semiconductor switches. To suppress such arcs, additional hardware components, e.g., relying on magnetic fields, are required.

Consequently, the analysis of current loads to be imposed on P-, S-, and B-switches is important, especially regarding the cell-level and array-level arrangements. This analysis will be discussed in Section III.

3) A current meter: A large range of time-varying currents requires a fine-grained metering sensor; a larger range leads to higher metering capability. The higher the sample rate, the higher the accuracy in measurements, but the costlier; high output values at a high sample rate requires a high-performance processor/global controller to process. In general, a linear Hall sensor integrated with digital signal processing, such as TLE4998 [8], is widely used for highly-accurate measurements.

Current sensing is performed primarily by the global BMS and used for the estimation of battery SoC. In the array-level parallel arrangement, the global controller measures the current from the entire battery pack (see Fig. 1), and then estimates the current of each local BMS by dividing the measured current by the number of arrays, assuming that all arrays are identical with respect to their voltage and resistance within a certain degree of discrepancy. When the discrepancy exceeds a specified threshold, this estimate is no longer valid. In such a case, individual local BMSs should be able to cope with the measurement; the local controller measures the current from its battery cells in the same way as the global controller does at the array level. Obviously, there is a tradeoff between increase in accuracy and reduction in cost and workload. The measurement frequency is another design parameter for weighing the accuracy against the computation overhead. Our separate paper will handle this issue.

4) Thermistor: Thermal sensing is performed in each local BMS. The main heat sources are battery cells, switches, and controllers. In particular, Battery cells generate the most heat of all. Also, the battery capacity changes, depending on ambient temperatures. Battery cells, e.g., lithium-ion, may not operate as their temperature approaches -58F, whereas they may explode if they approach 167F. Since the operating temperature range is very wide, a temperature sensor with accuracy on the order of ±35F can suffice, and such a sensor is less expensive. Digital temperature sensors, such as thermistors, thermocouples, and resistive temperature detectors [13], can meet our need. Conversely, a heat sink to which the

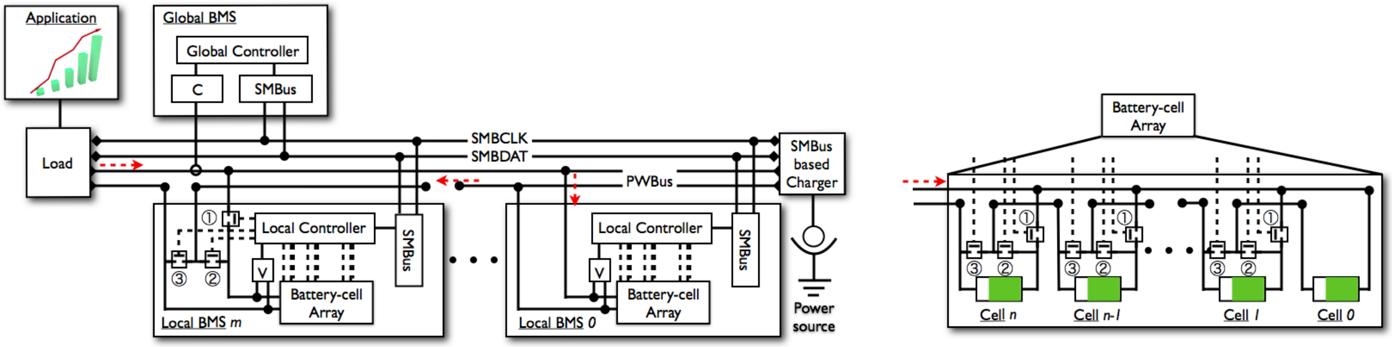


Fig. 1. The schematic diagram of PISA: switch (1) is called P-switch, and switches (2) and (3) are called S- and B-switches, respectively. The arrowed line indicates the discharge of the battery.

dissipated heat is transferred is also required. There are three basic modes of heat transfer: in a solid, in a fluid flow, and through a vacuum [14], but the choice of a means for the heat sink is design-specific.

5) SMBus [7]: It consists of data and clock lines via which the global controller, local controllers, and the charger communicate with each other. Through the data line SMBDAT, messages travel. A message generally consists of address, commend type, control, and data bits. Conversely, the clock line SMBCLK is used to indicate the beginning and the end of a communication in coordination with the SMBDAT signals. That is, a high-to-low transition on SMBDAT with SMBCLK high indicates a start condition, and a low-to-high transition on SMBDAT with SMBCLK high indicates a stop condition. Besides, other functionalities include clock synchronization and arbitration of contention for SMBDAT. Because SMBus is widely used as a protocol for smart batteries, we adopt it for our need.

B. Command Codes for Controlling Switches

Each command code is 3-digit long. The first digit indicates the P-switch's state, and the second and the last digits indicate states of the S- and B-switches, respectively. The value of 0 (1) means that the corresponding switches are switched off (on).

These codes are designed to control P-, S-, and B-switches, and applied for both the array- and cell-level configurations. At the array-level configuration, the global controller issues an appropriate command code in Table I to individual local controllers. A sequence of command codes is issued systematically based on the switch-configuration algorithm shown in Fig. 2. The local controllers then execute the dispatched code to control the array-level switches. At the cell-level configuration, on the other hand, the local controller executes a sequence of command codes to individual cells independently.

TABLE I
COMMAND TYPE AND CODE

Type	NULL	INIT	BYPASS	PARALLEL	SERIES
Code	000	100	001	101	010

Each command code has its own purpose. First, the NULL code is applied to keep a specific battery cell (an array) open, disconnecting all the battery cells (arrays) behind the cell. For instance, setting Cell 2 to the NULL code means that Cells 0 and 1 are not in use irrespective of the configuration of their switches. This bypass can be an option for the case that any switches around Cells 0 and 1 are dysfunctional. Second, the INIT code indicates the beginning of the battery-cell array (a chain of arrays). Thus, the INIT code is applied to the battery cell (array) next to the one to which the NULL code has been applied. Third, the BYPASS code is applied to bypass any battery cell (array) except for the first, i.e., Cell 0 (local BMS 0). Since the first cell (local BMS 0) does not have its own switches, the INIT code is applied to the next cell, resulting in the bypass of the first. Next, the PARALLEL code is applied to make a parallel arrangement. Likewise, the SERIES code is applied to make a series arrangement.

C. Data Aggregation

The global BMS periodically aggregates the information on battery conditions (including mainly voltage, temperature, and current) that individual local BMSs monitor within their battery-cell array. The local controller in an local BMS measures a terminal voltage between the two terminals of the battery-cell array. Conversely, to measure the voltage of individual cells, e.g., Cell i, the local controller applies the INIT code to Cell i, and the BYPASS code to Cell $i+1$ to n. In case of Cell 0, it applies the BYPASS code to all cells. The global controller then fetches the voltage measure from the local controller during the aggregation period. The local controller also measures the temperature of its battery-cell array. When the temperature exceeds a certain threshold, the local controller disconnects it from the load by issuing the NULL code to Cell n. Unlike the voltage measure, the local controller reports this emergency occasion to the global controller at any point in time. For the measurement of currents, as mentioned previously, since costly, it is delegated to the global BMS.

D. Array- and Cell-level Arrangements

The global BMS is responsible for the array-level arrangement, while the local BMS is for the cell-level arrangement. According to the switch-configuration algorithm shown in Fig. 2, the parallel arrangement is changed into the series arrangement or vice versa. First, the global (local) controller takes as an input a bit-array of connectivity, *barr*, where 1 and 0 of the ith bit indicate the connection and bypass of the ith array (cell), respectively. Then, it searches the first available array (cell) and applies the INIT code to the array-level (cell-level) switches. Afterwards, when arrays (cells) are to be connected in parallel or in series, the PARALLEL or the SERIES code is applied to individual arrays (cells), respectively. Consequently, the arrays (cells) are connected as specified.

```
Set switches:
  Input: $barr(n+1)$: bit-array of battery connectivity;
  Input: $ctype$: {parallel, series};
  Output: $tarr(n)$: type-array of switch configuration;

  $i \leftarrow 1$;
  while $barr(i) < 1$ /* search the first available unit */
      $i \leftarrow i+1$;
      switch $barr(i)$
          case 0: /* disconnect the unit from the load */
              $tarr(i-1) \leftarrow$ null;
          case 1: /* connect the unit to the load */
              $tarr(i-1) \leftarrow$ init;
              break;
  $i \leftarrow i+1$;
  while $i \leq n+1$ /* connect the rest of the units */
      switch $barr(i)$
          case 0: /* bypass the unit */
              $tarr(i-1) \leftarrow$ bypass;
          case 1: /* connect the unit in series or in parallel */
              $tarr(i-1) \leftarrow ctype$;
      $i \leftarrow i+1$;
```

Fig. 2. Switch-configuration algorithm

Over the course of battery activity (i.e., charge, discharge, and rest), some arrays (cells) may be bypassed. In this case, the global (local) controller determines which arrays (cells) to be bypassed by setting *barr*, and then runs the algorithm with *barr*. At the same time, the local BMSs that have the arrays bypassed may put into a sleep mode where the monitoring halts for the power savings purpose. The local BMSs in the sleep mode go back to an operation mode upon the global controller's request. The rest period depends on a discharge rate; the lower the discharge rate, the longer the rest period. It is determined based on a battery-activity scheduling mechanism we developed earlier [10].

The local controllers running the switch-configuration algorithm can effectively perform voltage-balancing, based on the scheduling mechanism. Also, the local controller can autonomously decide to make its battery-cell array open-circuited by applying the NULL code to the array-level switches. This is an exceptional case that the array is overheated, overcharged, or deep-discharged.

E. Interaction with applications

The application requests from the global BMS the battery's remaining operation-time, the time for the battery to fully be charged, and the battery's lifetime—in particular, accurate prediction of the lifetime is of great importance to the battery's lifetime warranty. First, for the battery's operation-time, the global BMS feeds the aggregated voltage and current into a reference model [10]. This reference model includes functions of time-dependent charge and discharge rates. Applying the reference model results in the remaining operation-time. Second, as in the calculation of the battery's operation-time, the time for the battery to be fully charged is also obtained from the reference model with the current charge rate. Third, the battery's lifetime is assessed based on the internal impedance of the battery cells, since a high impedance consumes a high power, generating heat. This will ultimately lower the supply voltage and its effect will propagate to other battery cells. Such irreversible degradation will shorten the battery's lifetime. To assess the lifetime, we measure the battery's terminal voltage, leading to the battery's internal impedance, such that $V = \frac{R}{R+r}V_0$, where V_0 is the reference voltage, and R and r are the load and the internal impedance. Note that various methods are available for this purpose.

III. Power Dissipation on Switches

As mentioned earlier, given constant resistance inside a switch, power dissipation on the switch exponentially increases as the current passing through it increases. Thus, we evaluate the current load on each switch in the both parallel and series arrangements.

A. Current Load on Switches at Array Level

The current loads on array-level P-, S-, and B-switches differ, depending on the type of battery-cell array arrangement. To calculate the loads in a local BMS, we first define the following parameters:

- xC: coulombs (per second) required by the load;
- p: probability of a local BMS being bypassed;
- r: number of local BMSs connected to the load for charge/discharge, and $r \leq m+1$;
- k: number of local BMSs consecutively connected among r BMSs, and $k \leq r$;
- i: number of local BMSs among k that precede the $(k+1)$th local BMS.

In the parallel arrangement, the array-level P-switch in each local BMS (Fig. 1) is switched on. Assuming that the differences in voltages of r local BMSs are within an acceptable threshold, the P-switches in r local BMSs equally share the load. Conversely, the array-level S-switches in the parallel arrangement will never be switched on, while the array-level B-switches serve as the conductors. In particular, the current on the B-switch differs depending on j; the greater the value of j, the higher the current on the B-switch. Table II shows the current on each array-level switch in the $(k+1)$th local BMS.

TABLE II
CURRENT ON ARRAY-LEVEL SWITCHES IN THE $(k+1)$TH LOCAL BMS

Arrangement	P-switch	S-switch	B-switch
Parallel	$(1-p)\frac{x}{r}$C	0C	$(1-p)\frac{ix}{r}$C
Series	0C	$(1-p)x$C	pxC

In the series arrangement, the array-level S-switch in each local BMS is always switched on unless the local BMS thereof is bypassed. The current on each of these S-switches is the same. The array-level P-switch in a local BMS, on the other hand, is switched on only if all preceding local BMSs are bypassed. For instance, $p^k(1-p)x\mathrm{C}$ is the current on the P-switch in the $(k+1)$th local BMS. In contrast, the array-level B-switch in a local BMS is switched on only if the local BMS is to be bypassed. Table II also shows the current on these switches in the series arrangement.

B. Current Loaded on Cell-Level Switches

Like array-level switches, cell-level switches are loaded with the same pattern. At the cell level, however, the current fed into the battery-cell array varies with the array-level arrangement. That is, the input current draws directly from the array-level P-switch in a local BMS. When local BMSs are connected in parallel, the current fed into the battery-cell array of corresponding BMSs is $\mathrm{C}^* = (1-p)\frac{x}{r}\mathrm{C}$. On the other hand, when local BMSs are connected in series, the input current becomes $\mathrm{C}^+ = (1-p)x\mathrm{C}$. Given these two input currents, we determine the current on each cell-level switch. To calculate this fraction, we define the following parameters.

- q: probability of a cell in the battery-cell array being bypassed;
- $s \leq n+1$: number of cells connected in a battery-cell array for charge/discharge;
- $l \leq s$: number of cells connected consecutively in a battery-cell array;
- j: number of cells among l cells which precede the $(l+1)$th cell.

In the parallel arrangement, cell-level P-switches within a battery-cell array share the current equally, assuming that voltages of the involved cells are uniform within an acceptable threshold. Cell-level S-switches, on the other hand, are never switched on, just like array-level S-switches. Also, like array-level B-switches, cell-level B-switches are switched on and loaded with the location-varying current. Table III shows the current on three cell-level switches in the $(l+1)$th cell.

TABLE III

CURRENT ON CELL-LEVEL SWITCHES IN THE $(l+1)$TH LOCAL BMS: C^* AND C^+ ARE INPUT CURRENTS DETERMINED BY THE ARRAY-LEVEL ARRANGEMENT, I.E., ARRAY-LEVEL P-SWITCHES.

Arrangement	P-switch	S-switch	B-switch
Parallel (Parallel)	$\frac{(1-q)}{s}\mathrm{C}^*$	0C	$(1-q)\frac{j}{s}\mathrm{C}^*$
Parallel (Series)	$\frac{(1-q)}{s}\mathrm{C}^+$	0C	$(1-q)\frac{j}{s}\mathrm{C}^+$
Series (Parallel)	0C	$(1-q)\mathrm{C}^*$	$q\mathrm{C}^*$
Series (Series)	0C	$(1-q)\mathrm{C}^+$	$q\mathrm{C}^+$

In the series arrangement, cell-level S-switches are always switched on unless the corresponding cells are bypassed. Like in the pack-level series arrangement, all these switches are loaded with the same current. On the other hand, cell-level P-switches are switched on only if all preceding cells therein are bypassed, while cell-level B-switches are switched on only if the corresponding cells are bypassed. Table III shows the current load on cell-level switches in the series arrangement.

IV. Cost Model for a Battery Pack

To assess the cost-effectiveness of a battery pack, we define its total cost C_T as the sum of manufacturing and service costs [20]:

$$C_T = C_M + C_S. \tag{1}$$

where C_M and C_S are the manufacturing and the service cost, respectively. C_M is closely related to the imperfect testing process, whereas C_S depends on the reliability of a battery pack within its warranty period. To assess reliability, we use simple a fault model: a battery cell fails when it becomes o pen-circuited (denoted as $F_B = 0$) or short-circuited (denoted as $F_B = 1$). Similarly, a switch fails when it is *stuck-at* ON state (denoted as $F_{SW} = 1$) or OFF state (denoted as $F_{SW} = 0$) regardless of inputs.

In what follows, we elaborate on the cost model.

A. Manufacturing Cost Model

Before shipping products to customers, it is important to test them, since their failure in the field causes significant expenses and influences the customer's satisfaction or the manufacturer's reputation. Suppose that N battery-cell arrays form a battery pack and the observed yield per pack is y_a, then the manufacturing cost per pack can be modeled as

$$C_M = \frac{(N+1)\cdot C_A}{y_a}. \tag{2}$$

The yield is the probability that a battery pack passes the test. This probability depends upon the fault coverage F; when $F = 0$, no fault occurs. Using a negative binomial yield model [6, 20], this probability is expressed as:

$$y_a(F,\ \lambda_a,\ \alpha) = (1 + \frac{F\cdot\lambda_a}{\alpha})^{-\alpha}, \tag{3}$$

where λ_a is the average number of defects per array and α represents the degree to which defects are clustered. When $\lambda_a = 0$, the battery-cell array is defect-free. When $\alpha \to 0$, defects are strongly clustered, while $\alpha \to \infty$ corresponds to weak clustering.

B. Service Cost Model

A battery pack may fail before the warranty expires. Even when some battery-cell arrays in the pack fail, the pack can "operate"[1] with spare arrays in both PISA and a conventional BMS without any reconfigurable switch). A battery pack is assumed to consist of N battery-cell arrays, each of which is composed of n battery cells. Also, we assume that the battery pack can operate as long as at least M-out-of-N arrays function, and each array operates as long as m-out-of-n battery cells function. This assumption applies to both the PISA and conventional BMS. The failure of a battery pack will incur a service cost for its repair or replacement. The service cost is then directly related to the pack's reliability and can be modeled as

$$C_S = (1 - R_P(t)y_a)C_F \tag{4}$$

where C_F is the service cost per pack, and $R_P(t)$ is the reliability (probability) that at least M battery arrays in the pack are still operational at time t. $R_P(t)$ depends on the reliability of arrays. Let $R_A(t)$ be the reliability that a battery-cell array operates at time t, subject to individual components i.e., battery cells and switches. Also, let X_B (X_{SW}) be an exponentially-distributed random variable for a battery cell (switch) with rate λ_B (λ^*_{SW}). Then, $P\{X_B > t\} = e^{-\lambda_B t}$ and

[1] providing the required voltage, current, or power.

$P\{X_{WS} > t\} = e^{-\lambda^*_{SW} t}$. For simplicity of analysis, all battery cells (switches) are identical.

Since $R_P(t)$ varies with the underlying fault model and battery arrangement, it is calculated with respect to each failure mode such that we have $(F_B,\ F_{SW}) = \{(x,\ y)|x,y \in \{0,\ 1\}\}$ and array-level parallel and series arrangements, resulting in 8 combinations. Each combination is divided into two parts, i.e., cell-level parallel and series arrangements. To indicate these configurations, we use notation CN.xy where N denotes a configuration, and x and y the array-level and the cell-level arrangements, respectively. Table IV lists all the configurations with fault instances.

The reliability of a battery pack is compared with PISA and the conventional BMS with respect to two types of arrangement. In the parallel arrangement, a short-circuited battery cell (i.e., $F_B = 1$) creates a cascade effect, causing the failure of the entire battery pack to be dysfunctional. This effect is a fatal vulnerability to the conventional BMS. Thus, the system reliability in the conventional BMS is subject to that of individual battery cells, resulting in C1.p for the array-level and C1.pp for the cell-level in Table IV. By contrast, PISA effectively deters the total failure by manipulating switches, in which P-switches play a critical role. In a case that switches get stuck-at ON state (i.e., $F_{SW} = 1$), even when a battery cell (array) fails, the entire array (pack) still operates as long as the P-switch functions, and at least m-out-of-n cells (M-out-of-N arrays) operate; only when both the battery cell (array) and the corresponding P-switch fail, the entire array (pack) fails, resulting in the reliability of C1.pp for the cell-level (that of C2.p for the array-level). In a case that switches get stuck-at OFF state (i.e., $F_{SW} = 0$), these switches can limit the reliability of battery cells (arrays). Thus, the minimum of a battery cell's (an array's) lifetime and the corresponding P-switch's lifetime is bound to the reliability of the array (pack), resulting in that of C3.p for the cell-level (that of C3.pp for the array-level). Conversely, an open-circuited battery cell (i.e., $F_B = 0$) seldom affects the operability of the entire array except for the fact that its out voltage slightly drops no less than that of the cell itself. Thus, the entire array (pack) operates as long as at least m-out-of-n cells (M-out-of-N arrays) operate, resulting in the reliability of C5.p for the cell-level (C5.pp for the array-level).

In the series arrangement, on the other hand, a short-circuited battery cell has the same effect as an open-circuited cell in the parallel arrangement on the reliability of the entire array (See C2.s and C2.ss). Rather, an open-circuited battery cell is critical at this point since it can cause the entire array (pack) to be dysfunctional in the conventional BMS. This effect is the same as a short-circuited cell in the parallel arrangement. PISA bypasses the open-circuited cell by switching on the corresponding B-switch and off the S-switch. In a case that switches get stuck-at ON state, the B-switch is critical to the reliability. If this is the case, the cell (array) becomes open-circuited and the entire array's (pack's) voltage slightly drops. This reliability is the same as C1.ps (also C2.ss, C3ps, C4.ss, C5.ps, C6.ss, C7.ps, and C8.ss in Table IV), C2.s for the cell-level (C4.s, C6.s, and C8.s for the array-level) Conversely, when they get stuck-at OFF state, the S-switch becomes critical, resulting in the same reliability as in the case of B-switches. The reliability of every configuration is presented in Table IV.

Since a switch's life varies with the current load imposed on the switch, the mean lifetime of each switch (λ^*_{SW}) is determined, based on the fractions shown in Tables II and III with a normalization factor τ and an exponential random variable λ_{SW} defined. The mean life for the main switch used in each configuration is also listed in Table IV.

V. Evaluation

Our goal is to design a dependable and scalable BMS for large-scale battery systems. To evaluate the dependability and scalability of PISA, we user metrics that include the power dissipation on switches, the reliability of the entire battery system, and the service cost due to failures. Also, the parameters presented in Section IV are listed and specified in Table V.

TABLE V

Parameters and values

Parameter	Description	Value
F	Fault coverage	0.05
λ_a	Average # of defects	3
α	Clustering	2
C_A	Manu. cost per array (normalized)	1
N	# of arrays	[20, 40]
M	Min. # of available arrays	[10, 20]
C_F	Service cost per array	$3 \times C_A$
n	# of cells per array	[15, 20]
m	Min. # of available cells per array	[15, 20]
$1/\lambda_B$	battery-cell lifetime (years)	23
$1/\lambda_S$	3-Ampere switch lifetime (years)	$1.5/\lambda_B$
p	Prob. of bypassing an array	0.05
q	Prob. of bypassing a cell	0.05
r	# of arrays connected to load	[7, 10]
s	# of cells connected to load	[15, 20]
x	Current Ampere (coulombs)	[1, 10]
τ	Normalization factor	$2(C^* + C^+)$

In what follows, using the above metrics and parameters, we demonstrate PISA's superiority to a conventional BMS without reconfigurability.

A. Results

1) A sequence of command codes matches a switch array: Five command codes are defined to turn on/off switches. In the cell-level arrangement, the local controller first determines which battery-cells are to be turned on. After this determination with the *barr* parameter set, it applies the algorithm in Fig. 2, forming a combination of on- and off-switches. Fig. 3 shows the correspondence between a sequence of command codes and the combination of switches. Without loss of generality, we assume that every battery-cell's voltage is 1V. For instance, when setting the total voltage to 1, the local controller turns Switch 16 on, resulting in Cell 6 (i.e., the 7-th cell) to be active. Cells before (after) Cell 6 become open (bypassed). In case of a 2V array, Cells 0 and 10 are connected in series. In general, the command code of INIT indicates the beginning of an active battery-cell array. That is, the preceding battery cells are ignored in spite of their connectivity. Cell bypassing is applied effectively when the switches associated with the preceding battery cells are dysfunctional. This feature minimizes the impact of failure of a single battery cell or switch that might otherwise cause the entire battery-cell array to fail.

TABLE IV

RELIABILITY COMPARISON BETWEEN PISA AND THE CONVENTIONAL BMS

Conf. (F_B, F_{SW})	Reliability of PISA	λ^*_{SW}	Reliability of the conventional BMS
C1.p (1, 1)	$R_P(t) = 1 - \{\sum_{i=1}^{n} \binom{n}{i} (P\{X_{SW} \le t\}(1-R_A(t)))^i (1 - P\{X_{SW} \le t\}(1-R_A(t)))^{n-i}$ $+ \sum_{i=m+1}^{n} \binom{n}{i} (P\{X_{SW} > t\}(1-R_A(t)))^i (1 - P\{X_{SW} > t\}(1-R_A(t)))^{n-i}\}$	$\lambda_{SW} \frac{1}{\tau} \frac{(1-p)}{M} x$	$R_P(t) = \Pi_{i=1}^{N} R_A(t)$
C1.pp (1, 1)	$R_A(t) = 1 - \{\sum_{i=1}^{n} \binom{n}{i} (P\{X_{SW} \le t\} P\{X_B \le t\})^i (1 - P\{X_{SW} \le t\} P\{X_B \le t\})^{n-i}$ $+ \sum_{i=m+1}^{n} \binom{n}{i} (P\{X_{SW} > t\} P\{X_B \le t\})^i (1 - P\{X_{SW} > t\} P\{X_B \le t\})^{n-i}\}$	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{mM} x$	$R_A(t) = \Pi_{i=1}^{n} P\{X_B > t\}$
C1.ps (1, 1)	$R_A(t) = \sum_{i=m}^{n} \binom{n}{i} P\{min(X_B, X_{S_C}) > t\}^i (1 - P\{min(X_B, X_{S_C}) > t\})^{n-i}$	$\lambda_{SW} \frac{1}{\tau} q \frac{(1-p)}{M} x$	$R_A(t) = \sum_{i=m}^{n} \binom{n}{i} P\{X_B > t\}^i (1 - P\{X_B > t\})^{n-i}$
C2.s (1, 1)	$R_P(t) = \sum_{i=M}^{N} \binom{N}{i} (R_A(t) P\{X_{S_A} > t\})^i (1 - R_A(t) P\{X_{S_A} > t\})^{N-i}$	$\lambda_{SW} \frac{1}{\tau} p x$	$R_P(t) = \sum_{i=M}^{N} \binom{N}{i} R_A(t)^i (1 - R_A(t))^{N-i}$
C2.sp (1, 1)	C1.pp	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{m} x$	C1.ps
C2.ss (1, 1)	C1.ps	$\lambda_{SW} \frac{1}{\tau} q(1-p) x$	C1.ps
C3.p (1, 0)	C2.s	$\lambda_{SW} \frac{1}{\tau} \frac{(1-p)}{M} x$	C1.p
C3.pp (1, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{mM} x$	C1.pp
C3.ps (1, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{M} x$	C1.ps
C4.s (1, 0)	C2.s	$\lambda_{SW} \frac{1}{\tau} (1-p) x$	C2.s
C4.sp (1, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{m} x$	C1.pp
C4.ss (1, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} (1-q)(1-p) x$	C1.ps
C5.p (0, 1)	The same as conventional BMS's	—	C2.s
C5.pp (0, 1)	The same as conventional BMS's	—	C1.ps
C5.ps (0, 1)	C1.ps	$\lambda_{SW} \frac{1}{\tau} \frac{q(1-p)}{M} x$	C1.pp
C6.s (0, 1)	C2.s	$\lambda_{SW} \frac{1}{\tau} p x$	C1.p
C6.sp (0, 1)	C5.pp	—	C1.ps
C6.ss (0, 1)	C1.ps	$\lambda_{SW} \frac{1}{\tau} q(1-p) x$	C1.pp
C7.p (0, 0)	C2.s	$\lambda_{SW} \frac{1}{\tau} \frac{(1-p)}{M} x$	C2.s
C7.pp (0, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{mM} x$	C1.ps
C7.ps (0, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{M} x$	C1.pp
C8.s (0, 0)	C2.s	$\lambda_{SW} \frac{1}{\tau} (1-p) x$	C1.p
C8.sp (0, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} \frac{(1-q)(1-p)}{m} x$	C1.ps
C8.ss (0, 0)	C1.ps	$\lambda_{SW} \frac{1}{\tau} (1-q)(1-p) x$	C1.pp

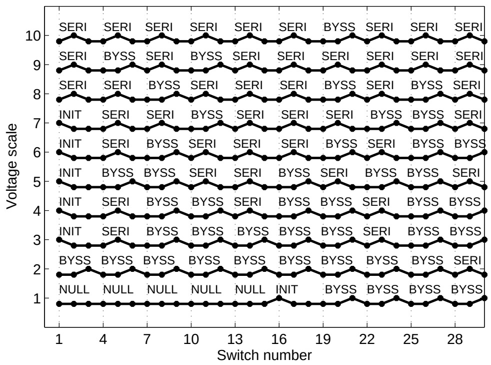


Fig. 3. Switching command array

2) Power dissipation varies with the type of switch: Three types of switch are used in PISA: P-, S-, and B-switches. Each switch incurs power dissipation because of their unavoidable internal resistance. Their power dissipation differs in parallel or series arrangement. In series arrangement, S-switch is turned on. As shown in Fig. 4, it consumes significant power. In contrast, to bypass a battery cell, its B- and S-switches are turned on and off, respectively. Thus, the likelihood of bypassing a battery cell dictates the power dissipation on the switches; a higher bypass probability results in lower power dissipation on the corresponding S-switch and higher power dissipation on the corresponding B-switch. In parallel arrangement, the P-switch is turned on. The power dissipation thereon, however, is negligible in comparison with the S-switch in series arrangement, since the current to the load is shared across the battery cells. By contrast, the B-switch in parallel serves as the conductor for the parallel-connected battery-cells. Its power dissipation, thus, depends solely on the distributed current. In general, it is on the two orders of magnitude higher than that on the P-switch. Note that S-switch is never turned on in parallel arrangement.

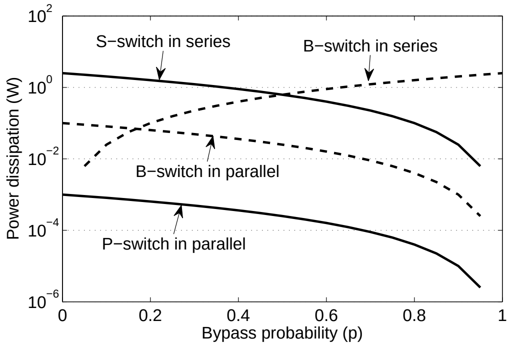
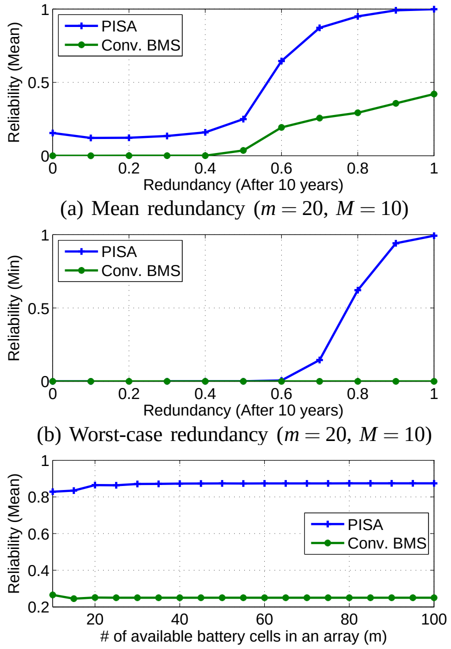


Fig. 4. Power dissipation on each switch in series and parallel arrangements

3) Redundancy greatly improves the battery pack's reliability: The battery pack operates as long as M-out-of-N battery-cell arrays function. Likewise, the array operates as long as m-out-of-n cells function. In other words, $(n-m)M + (N-M)n$ battery cells can be used as backups. The more the back-up cells available, the longer the battery pack will last. PISA is more effective than the conventional BMS in utilizing redundant battery cells. As shown in Fig. 5-(a), PISA improves the pack's reliability an average of 2.7 times more than the conventional BMS. Specifically, the reliability is enhanced most (by 3.2 times) at the redundancy ratio of 0.6. Even in the worst case, as shown in Fig. 5-(b), PISA effectively increases the pack's reliability at 60% redundancy, and the pack's reliability is maximized at 100% redundancy.

Moreover, PISA can effectively handle a large number of battery cells. As shown in Fig. 5-(c), PISA improves the reliability by 20% with a 10-fold increase in the number of available battery cells in an array (m). By contrast, the conventional BMS cannot deal with a large number of battery cells.



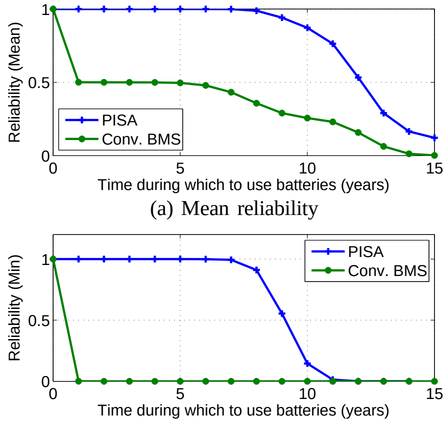
(a) Mean redundancy ($m = 20$, $M = 10$)

(b) Worst-case redundancy ($m = 20$, $M = 10$)

(c) Array size at the redundancy rate of 0.7

Fig. 5. Reliability vs. redundancy ($\frac{1}{\lambda_B} = 23$, $\frac{1}{\lambda_B} = 34$, $t = 10$)

4) PISA allows the battery pack to last longer: Although individual battery cells, on average, last long, e.g., 23 years, the lifetime of the battery pack formed by these cells is not guaranteed to last that long. Actually, it is subject to the arrangement of cells and their (random) failures.[2] For instance, when the battery cells are connected in parallel, a cell that became short-circuited causes the entire battery pack to be unusable. In the case of series arrangement, a cell that became open-circuited has the same consequence. PISA effectively prevents the entire pack from failing due to such a single cell failure. As shown in Fig. 6-(a), on average, PISA offers twice as much reliability as the conventional BMS over the battery's lifetime. In particular, a PISA-managed battery pack to be used for 10 years is 3 times more reliable than the conventional BMS-managed one, whose reliability is only 26%. In the worst-case scenarios such as those mentioned earlier, as shown in Fig. 6-(b), PISA is an order-of-magnitude more reliable than the conventional BMS, which is susceptible to the failure of the entire pack caused by the failure of even a single battery cell. Thus, PISA offers robust battery management regardless of the type of failures that might occur.

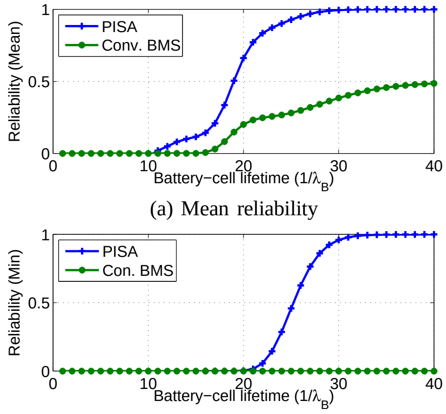


(a) Mean reliability

(b) Minimum reliability

Fig. 6. Change in reliability as the pack is used (redundancy is 0.7)

5) PISA can always offer a dependable and affordable life warranty of battery packs: What type of battery cells do we need to meet the requirement that a battery pack must last for 10 years with 50% reliability? Fig. 7 answers this question. PISA requires battery cells of 20-year life warranty, while the conventional BMS requires those of 40-year life warranty. Theoretically, a PISA-managed battery pack would be twice more affordable than a conventional BMS-managed one. In the worst-case scenarios, as shown in Fig. 7-(b), the conventional BMS cannot provide any warranty for the battery pack, whereas PISA requires battery cells of 26-year life to meet the requirement.



(a) Mean reliability

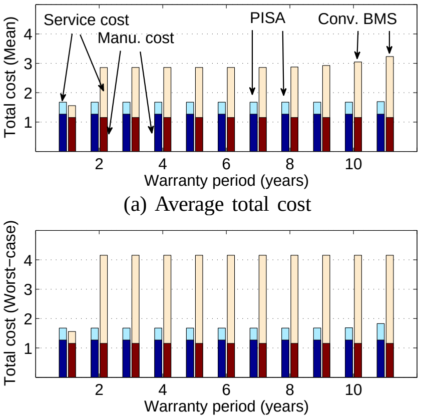
(b) Minimum reliability

Fig. 7. The battery pack's reliability with respect to a battery-cell lifetime (redundancy is 0.7)

6) PISA is cost-effective: PISA requires switches around each cell, increasing the manufacturing cost. However, since a set of switches is greatly less costly than battery cells, the cost for PISA is not much different from that for the conventional BMS, as shown in Fig. 8-(a). Rather, the service cost, which associates with the reliability, has a great impact on the total cost. The service cost is important to reduce because it is usually much costlier than the manufacturing cost in practice. PISA successfully reduces the service cost an average of 4.2 times less than the conventional BMS across various warranty periods. In the worst-case scenarios, PISA makes more than 7.4 times of cost-savings for a longer than

[2] The more cells in th system, the more likely some of them may fail.

10-year warranty period. So, PISA makes a great cost savings, making it affordable and dependable.



(a) Average total cost

(b) Total cost in the worst case

Fig. 8. Change in the total cost for a single battery pack with respect to warranty periods

VI. Related Work

A key function that a BMS incorporates is cell-balancing. Since all battery cells are not created equally, one cell can be charged faster than the other. When the cell is fully charged earlier, the charging process for all cells is stopped, and then either other cells are charged separately or the fully-charged cell is discharged until its SoC reaches that of the second fully-charged one. This discharge separation can be realized using an inductor as described in [18]. That is, the primary inductor is charged from the fully-charged cell, converting the charge into magnetic energy and then storing it in the secondary inductor. Then, the secondary inductor is used to charge the other battery cells. This balancing method, however, seldom avoids energy losses and also is relatively costly. Instead of inductors, switches can be used for separate charge/discharge as in our architecture. Similarly, the prototypical implementation of cell-balancing can be found in [11].

Understanding battery characteristics is of great importance to scheduling of battery charge and discharge. There have been a number of studies on battery characteristics. Szumanowski and Chang [23] presented a linear model as functions of SoC, and Plett [17] extended a linear model to a nonlinear one using an extended Kalman filter. Rong and Pedram [19] presented a closed-form analytical model for predicting the residual energy of a lithium-ion battery. Cloth *et al.* [3] modeled recovery behavior with Markov reward model combined with Kinetic battery model. Kim and Shin [9, 10], and Benini *et al.* [1] also modeled recovery efficiency and discharge efficiency in analysis of battery characteristics.

VII. Conclusion

In this paper, we have presented a combined hardware–software architecture, called PISA, that enables effective monitoring of a large number of battery cells, and efficient control and reconfiguration of their arrangement. By tolerating failures of switches and battery cells, PISA extends the battery life significantly compared to the conventional BMS. Integration of a switch-configuration algorithm and a reconfigurable architecture enhances the reliability, customizability, and extensibility of large-scale battery packs.

References

[1] Luca Benini, Davide Bruni, Alberto Macii, Enrico Macii, and Massimo Poncino. Discharge current steering for battery lifetime optimization. *Trans. on Com.*, 52(8):985–995, Aug. 2003.

[2] Song Ci, Jiucai Zhang, Hamid Sharif, and Mahmoud Alahmad. A novel design of adaptive reconfigurable multicell battery for power-aware embedded network sensing systems. In *Globecom*, pages 1043–1047, Washington, DC, USA, 2007. IEEE.

[3] Lucia Cloth, Marijn R. Jongerden, and Boudewijn R. Haverkort. Computing battery lifetime distributions. In *DSN*. IEEE, June 2007.

[4] Green Car Congress. Noaa: February 2009 ninth warmest february for globe. `http://www.greencarcongress.com/2009/03/noaa-february-2.html`, Mar. 2009.

[5] Amanda Davis, Ziyad M. Salameh, and Stephen S. Eaves. Evaluation of lithium-ion synergetic battery pack as battery charger. *Trans. on Energy Conversion*, 14(3):830–835, 1999.

[6] José T de Sousa and Vishwani D. Agrawal. Reducing the complexity of defect level modeling using the clustering effect. In *DATE*, pages 640–644, New York, NY, USA, 2000. ACM.

[7] Duracell, Energizer, Fujitsu, Intel, Linear, Maxim, Mitsubishi, PowerSmart, Toshiba, Unitrode, and USAR. System management bus (smbus) specification ver. 2.0. Technical report, SBS Implementer Forum, 2000.

[8] Infineon. Linear hall sensor.

[9] Hahnsang Kim and Kang G. Shin. On dynamic reconfiguration of a large-scale battery system. In *RTAS*, pages 87–96, San Francisco, CA, U.S., Apr. 2009. IEEE.

[10] Hahnsang Kim and Kang G. Shin. Scheduling of battery charge and discharge. In *RTSS*, Washington, D.C., USA, Dec. 2009. IEEE.

[11] Gerald Landrum, Thomas A. Stuart, and Wei Zhu. Fast equalization for large lithium ion batteries. In *Oceans*, pages 1–6. IEEE, Sep. 2008.

[12] John Markoff. Nyt: Pursuing a battery so electric vehicles can go the extra miles. `http://www.nytimes.com/2009/09/15/science/15batt.html?scp=1\&sq=electricvehicletesla20ibm\&st=cse`, Sep. 2009.

[13] MAXIM. Temperature sensors. http://www.maxim-ic.com/pl_list.cfm/filter/11.

[14] Donal McNamara. Thermal sensing requires system-level design. *Power Electronics Technology*, pages 28–35, May 2007.

[15] Stephen W. Moore and Peter J. Schneider. A review of cell equalization methods for lithium ion and lithium polymer battery systems. In *SAE*, number 0959 in 01, pages 1–5. Society of Automotive Engineers, 2001.

[16] General Motors. Latest chevy volt battery pack and generator details and clarifications. `http://gm-volt.com/2007/08/29/latest-chevy-volt-battery-pack-and-genera%tor-details-and-clarifications/`, Aug. 2007.

[17] Gregory L. Plett. Extended kalman filtering for battery management systems of lipb-based hev battery packs part 1. background. *J. of Power Sources*, 134(2):252–261, 2004.

[18] Werner Roessler. Lithium-ion battery management integrating active load balancing. In *SAE*, 2008-01-1335, pages 1–10, Warrendale, PA, USA, 2008.

[19] Peng Rong and Massoud Pedram. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *Trans on VLSI Sys.*, 14(5):441–451, 2006.

[20] Saeed Shamshiri, Peter Lisherness, Sung-Jui Pan, and Kwang-Ting Cheng. A cost analysis framework for multi-core systems with spares. In *Int'l Test Conf.*, pages 1–8, New York, NY, USA, 2008. IEEE.

[21] Thomas Stuart, Fang Fang, Xiaopeng Wang, Cyrus Ashitiani, and Ahmad Pesaran. A modular battery management system for HEVs. In *SAE*, pages 1–9, Hyatt Crystal City, VA, USA, June 2002. SAE International.

[22] Vinesh Sukumar, Mahmoud Alahmad, Kevin Buck, Herbert Hess, Harry Li, Dave Cox, Fadi Nessir Zghoul, Jeremy Jackson, Stephen Terry, Ben Blalock, M.M. Mojarradi, W.C. West, and J.F. Whitacre. Switch array system for thin film lithium microbatteries. *J. of Power Sources*, 136(2):401–407, 2004.

[23] Antoni Szumanowski and Yuhua Chang. Battery management system based on battery nonlinear dynamics modeling. *Trans. on Veh. Tech.*, 57(3):1425–1432, May 2008.

[24] H. Visairo and P. Kumar. A reconfigurable battery pack for improving power conversion efficiency in portable devices. In *Int'l Caribbean Conf. on Devices, Circuits and Systems (ICCDCS)*, pages 1–6. IEEE, April 2008.

[25] Christopher Yang, David McCollum, Ryan McCarthy, and Wayne Leighty. Meeting an 80% reduction in greenhouse gas emissions from transportation by 2050: A case study in california. *Transportation Research Part D: Transport and Environment*, 14(3):147–156, 2009.