A Robust System Monitoring and Control for Battery Energy Storage Systems in Electric Vehicle Charging

Yukun Lou
Electrical Engineering & Computer Science
University of Michigan
Ann Arbor, USA
louyukun@umich.edu

Alireza Ramyar
Electrical Engineering & Computer Science
University of Michigan
Ann Arbor, USA
aramyar@umich.edu

Xiaofan Cui
Energy Science & Engineering
Stanford University
Palo Alto, USA
cuixf@stanford.edu

Jason Siegel
Mechanical Engineering
University of Michigan
Ann Arbor, USA
siegeljb@umich.edu

Anna Stefanopoulou
Mechanical Engineering
University of Michigan
Ann Arbor, USA
annastef@umich.edu

Al-Thaddeus Avestruz
Electrical Engineering & Computer Science
University of Michigan
Ann Arbor, USA
avestruz@umich.edu

2023 IEEE Transportation Electrification Conference & Expo (ITEC) | 979-8-3503-9742-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITEC55900.2023.10186918

***Abstract*—This paper presents a System Monitoring and Control (SMC) strategy for battery energy storage systems (BESS) for electric vehicle (EV) chargers and the grid. With an increasing number of EVs, there is a need to handle the great peak demand for EV charging. BESSs provide a fast energy response to charging demands but must have excellent power and energy utilization, battery lifetime, efficiency, and cost. To ensure safety, reliability, and economy, a hierarchical SMC architecture in managing the complexity of a BESS. A prototype design is demonstrated with a hardware implementation and results.**

***Index Terms*—System Monitoring and Control (SMC), battery energy storage systems (BESS), 2-BESS, Battery Management System (BMS), EV charging**

I. Introduction

The growth of electric vehicles (EVs) cannot continue without Battery Energy Storage Systems (BESS) for EV chargers and the grid. The imminent demand for EV charging is a consequential concern. In EV fast charging, energy storage is needed to supply the peak demand that the grid cannot support [1]. With the number of retired second-use EV batteries increasing, using them for BESS is more sustainable [2]. These second-use battery energy storage systems (2-BESS) add additional economic value before recycling [3].

Safety, reliability, and cost-effectiveness are vital for the mission of the BESS, yet challenging because of its complexity of components including many batteries, power converters, inverters, thermal management, and so forth. A system is needed to monitor and control these components while integrating functionality. This system needs to satisfy the distinct set of requirements for BESSs, while being scalable, flexible, and configurable. For example, the interconnection of batteries and converters in a 2-BESS can be one-to-many, or many-to-many, with interdependent converter operation and setpoints [3]. For example, monitoring and controlling temperature is important because it can dominate life, capacity, and performance in batteries [4], and reliability in power converters. At the same time, monitoring state of health (SOH), state of charge (SOC), and temperature are important in determining battery capability in optimizing power and energy utilization. Historical data is needed for diagnostics, fault prevention, and predictive maintenance, while a fast data stream is requisite to alert and respond to faults [8]–[10]. As the size of BESS systems increase (1–10 MW, 100 kWh–100 MWh), these needs intensify [11].

This paper presents: (1) several essential attributes of an SMC for a BESS; (2) hierarchy abstractions for the architecture; (3) prototyping approach; (4) implementation and results.

A. System Monitoring and Control (SMC)

A well-designed hierarchical System Monitoring and Control (SMC) that is scalable and flexible is needed to meet the current needs and future evolution of battery energy storage systems. The SMC is a unified and hierarchical architecture to organize both distributed BMSs and centralized control, monitoring, and safety of a BESS. At the lowest level, Battery Management Systems (BMS) operate at the cell and module level in battery packs to ensure safe and essential operation [12] [13]. As shown in Fig. 1, an SMC contains several levels of hierarchy with a Central Monitoring and Control Unit (CMCU), which organizes distributed monitoring and control agents (DMCA) containing abstractions for the power converters, batteries, protection, or thermal management, among others. For example, for battery modules, DMCA may include current and voltage monitoring for the whole module and can actuate a contactor, while administering pertinent monitor and control functions in the BMS; for power converters, the pertinent subsystem in the DCMA might be a digital controller instead of a BMS.

As a point of reference, the BMS typically performs cell balancing to ensure even charging and discharging, prevents overvoltage and overcurrent, and monitors temperature. Cell balancing can be either passively through power dissipation often through resistors, or actively through switched power conversion [14]. BMS can be made to monitor and balance SOC. BMS can provide data to infer SOH including capacity loss and the growth of internal resistance, which determine the degradation of energy capacity and fading of power capability over time and usage [8].

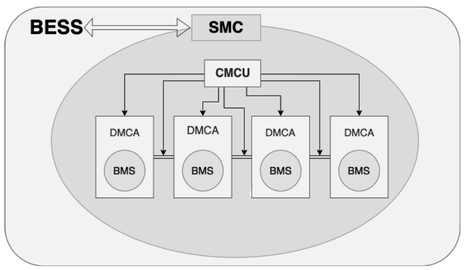

Fig. 1. SMC High-Level Architecture

B. Scalability and Flexibility

Scalability corresponds to the organization of an arbitrary number of additional units, which on a physical level interact with each other (e.g., distributed and interconnected power converters), with an acceptable decrease in system performance; for example, tolerable delays in fault handling and responsiveness to commands. Hierarchy enables scalability. For example, critical faults can be localized to and handled by the DMCA, making this delay independent of the number of DMCA agents. Data can also be locally collected, stored, reduced, and pre-processed.

Flexibility represents incorporating new functionality, capability, or features, which include, for example, new types of hardware, sensors, or interfaces (e.g., BMS). The incorporation of this new functionality must be interoperable with the existing system, without extensive redesign. Vis-a-vis both scalability and flexibility, updates especially to firmware ought to be performed autonomously through the network; for example, reprogramming through a Controller Area Network (CAN) bus bootloader.

C. Hierarchy in the SMC

Hierarchy ought to be designed at various levels of overlapping and interconnected abstractions, which include data, computation, communications, protocol, control, hardware, and software. If instead the design were flat, with for example only one central processor, latencies accumulate as this processor becomes overwhelmed, Also, the design becomes intractable as any choice of task priorities does not accommodate all the scenarios and use cases.

In data abstraction, the hierarchy may require data reduction and pre-processing. For instance, logging of faults can be stored either at the DMCA with electrically-erasable programmable read-only memory (EEPROM), or within the central controller. Raw data such as voltage, currents, and temperature can be reduced to estimates of SOC or system-level warnings (if out of bounds), while historical data can reduced to higher-level information such as SOH [8].

The data hierarchy cannot function without accompanying computation and communication hierarchies. Pre-computing when performed at the DMCA distributes the computation while reducing data storage. Additionally, with reduced data, the communication burden is also reduced.

The communication hierarchy needs to support both time-critical and high-throughput requirements. Communications can be partitioned within the data link layers, for example from the cloud to the central controller through Ethernet, from the CMCU to the distributed MCUs in the DMCA through CAN Bus, and from the MCUs in the DCMA to the BMSs and power converters through SPI or I2C. Fault events are time-critical so data streaming needs to be suspended. Communication protocols like CAN Bus need to be utilized to prioritize fault messages. Streaming data needed for diagnostics are high throughput, requiring high-bandwidth communications between the CMCU and DCMA.

Protocol hierarchy encompasses task scheduling, control, and communication, requiring scalability and flexibility to ensure guaranteed and calculable latency. Round-robin strategies are quantitatively predictable for latency, although the latency performance may be lower than for other strategies. Longer but predictable latency is acceptable for BESS because the timescales pertinent to batteries and thermal management are relatively slow. Faster timescales, for example, related to power converters, are better handled locally at the DMCA, where local protocols can be more specifically organized. Scalability in a round-robin strategy can be addressed to hierarchical round-robin approaches to provide better performance [15] [16].

The control hierarchy organizes the feedback loops that consist of information (from monitoring or derived data) and actions based on criteria. These feedback loops typically have a range of timescales, and the hierarchy organizes the timescale separation along with the coupling among these loops that may traverse multiple timescales. Distributed control can manage the coupling among interrelated information and actions, as well as the classification of urgency, importance, performance, and value metrics.

The hardware hierarchy is an implementation abstraction. It involves the classification of energy storage, computing, and communications, with boundaries determined by physical limitations. For example, the BESS is divided into cells, modules, and packs as layers; cells are the smallest electrochemical unit, modules are collections of cells with a BMS, and packs are collections of modules that may manage multiple BMSs. Fault management strategies are often employed at every layer within a battery pack.

Along with the hardware, hierarchies are employed within the software, including data abstractions, drivers, interfaces, and handlers, often within a real-time operating system (RTOS).

D. Choice of Scope

The implementation of the hierarchies within the SMC depends on whether the target goals are production or prototyping. Ultimately, prototypes must be translatable to a scalable and flexible production implementation. However, some scalability may be sacrificed in a prototype implementation for faster build, test, and iteration. This paper presents an SMC prototype that implements robust monitoring, control, and fault handling that can be scaled. Section II discusses the features of the prototype and its hierarchy. Section III demonstrates the hardware implementation and testing results.

II. SMC PROTOTYPE

Prototypes in research and development prioritize flexibility and agility but need to easily translate and scale to production. Monitoring, control, and fault handling are the three crucial functions that structure the hierarchy in this prototype. Monitoring not only informs whether the system is operating properly but also provides data needed for scaling to a larger system.

The strategy employed in this prototype is to implement as many of the different hierarchies in software within the CMCU, which in this case is a PC programmed in Python. The DMCAs provide mostly raw data and initiate actions that are atomic. This enables better agility in the research and development of BESSs. However, this may incur latencies that may not scale well for large BESSs. When translating to production, these implementations in Python are distributed to implementations in DMCAs (i.e., embedded firmware). It is worth noting that time critical fault handling in the prototype, especially those related to safety and damage, are implemented locally at the DMCAs.

A. Specific Functionality

The main focus of the SMC prototype for BESS is to implement a subset of essential features, which are primarily divided into monitoring, control, and fault handling. The objective is to accurately assess the system's performance for scaling while efficiently developing and testing the prototype. Tasks as well as communication are performed through round-robin strategy.

a) Control: The CMCU consisting of a PC determines both the bus voltage and power output of the BESS, while enforcing predetermined setpoints for the power conversion that ensure optimal battery utilization. This is achieved by transmitting the setpoints for the converter currents [8]. The central controller generates commands and sends them to the DMCAs, which then execute appropriately. Currently, only a subset of functions, such as measuring voltage/current, setting current, and turning the converters on/off, are implemented in the prototype. Flexibility in the design allows additional functionality, for example, systematic thermal management, to be added.

b) Monitoring: The DCMAs directly monitor batteries and power converters, with voltage, current, and temperature sensors. The SMC continuously collects data locally and transmits it to CMCU for decision, analysis, and storage. From the data, the SOC of the batteries can be inferred to determine balancing within the BESS. Local monitoring and control of current and voltage are used to detect control instability, overcurrent (OC), undervoltage (UV), and overvoltage (OV) and actuate a local response, while communicating the events to the CMCU. Additionally, the temperature is currently measured directly; in production, a higher granularity in temperature, which can be available at the cell level from the BMS, will be implemented.

c) Fault Handling: Within the current SMC prototype, faults are categorized as either locally-handled, centrally-handled, or communication-related.

1) Locally handled faults: Time critical faults, such as OC, OV, UV, converter instability, and abnormal local temperature, can be identified by monitoring the transient local data (i.e., current, voltage, and temperature). As shown in Fig. 2, once detected, the CMCU can perform the appropriate action, such as shutting down the power converter or actuating a contactor. Faults are reported to the CMCU during the round-robin process. In the current implementation, a power converter fault results in the CMCU transmitting a broadcast shutdown command. However, an implementation of a controlled and orderly shutdown is more appropriate for a larger system.

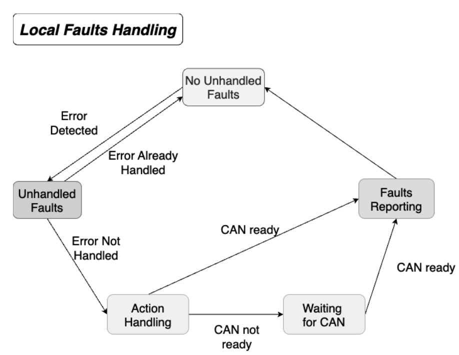

Fig. 2. State Diagram of Local Fault Handling

2) Centrally handled faults: Slower timescale faults, especially those that require data over a large time interval, such as SOC to determine overcharging or depth of discharge. When such a fault is detected, the CMCU generates a message and transmits it to the appropriate DCMAs within the round-robin protocol.
3) Faults in the communication process: Faults can occur during the communication between the CMCU and DCMAs. Different errors can occur in the CAN Bus including synchronization errors and message corruption, among others. As shown in Fig. 3, the CMCU initiates retries with timeouts implemented in the DCMAs

to determine loss of communications, to initiate local shutdown sequences.

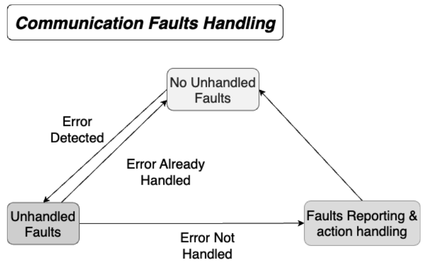

Fig. 3. State Diagram of Communication Fault Handling

B. Functionalities within Hierarchies

In the current prototype, the data hierarchy is such that data is collected at the DMCA and then transmitted to the central controller; all raw data are stored and analyzed in the CMCU without being reduced. This enables straightforward development of the analysis and decision-making algorithms within the SMC. When scaling for production, more data will be stored locally, potentially in the EEPROM of the MCU.

When prototyping the SMC hierarchy, we implemented the various hierarchies within the software, including representations of the hardware hierarchy for DCMAs of batteries and power converters through various data abstractions, interfaces, and handler functions. Monitoring and atomic actuation at the DMCAs were performed through MCUs and communication via CAN Bus. The embedded codes in MCUs had corresponding structures and functions, including drivers for GPIO, PWM, ADC, and so forth.

C. Central Monitor and Control Unit (CMCU)

The CMCU is the centralized controller in the SMC hierarchy, which is implemented in a PC through Python programming and communicating through a USB-to-CAN Bus connection. The CMCU communicates with and controls all the DMCAs, assigns tasks such as measuring current/voltage for monitoring and setting converter current for control, and handles faults through command messages. Within the CMCU, data is logged from the DMCA into a database.

D. Distributed Monitoring and Control Agents (DMCA)

The DMCA is the distributed controller in the SMC hierarchy and is implemented through an MCU. The DMCA monitors and collects local data, executes control commands from CMCU, and protects its constituents against local faults. Various sensors can be used for data collection and monitoring purposes, including voltage, current, and temperature sensors.

E. Communication

A round-robin protocol with its bounded latency is used for communication, as shown in Fig. 4. Each DCMA is assigned tasks in real time, with an upper bound on the time to complete a task. The CMCU communicates with a single DCMA during each time slot and assigns monitoring, protection, and control tasks to that particular DCMA. Once the DCMA completes its assigned tasks, the CMCU moves to the next node and assigns specific tasks. Critical messaging extrinsic to round-robin may be implemented in the future through CAN; however, these must be chosen judicially and sparsely as these incur a complexity that does not scale well.

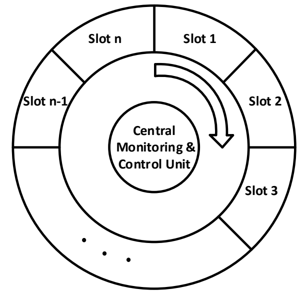

Fig. 4. Round-Robin Protocol

III. HARDWARE DEMONSTRATION

The prototype SMC was implemented in hardware on a 1 kW energy storage testbed as shown in Figs. 5 and 6.

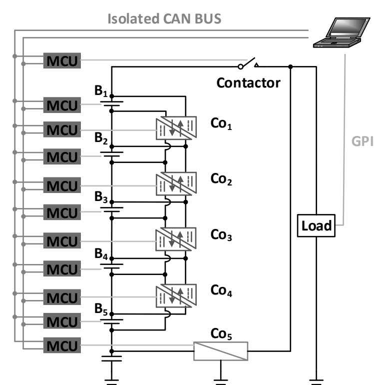

Fig. 5. Prototype BESS with SMC

A. Hardware Description

An SMC was implemented for a partial power processing architecture of a BESS prototype consisting of five batteries, five converters, and a load [8] as shown in Fig. 5. Four converters (Co1, Co2, Co3, and Co4) process the mismatched power among the batteries, and one converter (Co5) regulates the load voltage. To ensure safe operation, all batteries, converters, and the load must be protected against potential faults in real time. Moreover, the converters must be controlled to optimize power flows, while collecting and logging data (such as voltage, current, and faults) for analysis.

The testbed in Fig. 6 used a constant current sink of 1A for the load and regulated the load voltage to 70V through Co5. The power converters and battery management boards connected to the batteries are controlled via an MCU networked over a CAN Bus with isolated transceivers, with each

Fig. 6. 1 kW BESS Testbed

MCU serving as a DMCA. These DMCA agents communicate with a PC acting as the CMCU. While the electrical load currently communicates with the CMCU via General Purpose Interface Bus (GPIB). The MCU selected for this testbed is a Texas Instruments Launchpad (LAUNCHXL-F28379D), with sufficient I/O ports for batteries, power converters, and loads, as well as for expansion with integrated modules and memory. Programming is performed using CCSTUDIO[1] for MCUs in C and Python for the CMCU implementation.

B. Hardware Results

We present a subset of SMC functions that are included in our centralized-distributed hierarchical prototype. The prototype will show the following capabilities: monitoring voltage and current of power converters and batteries, control in both the CMCU and DCMA, handling of local and central faults, and communication between the CMCU and every DCMA. Each DCMA completes the tasks or responds to faults within an acceptable latency.

C. Test Results

We employ continuous monitoring of all the power converters within the BESS prototype. A snapshot of four measured signals from the BESS is shown in Fig. 7, when a generated OV fault occurs at the output of Co1. The relevant DMCA detects the fault and promptly shuts down Co1 by de-asserting the Co1 Enable Signal. After this local fault handling action, the DMCA of Co1 reports this fault to the CMCU during its communication slot (maximum 200 ms latency) in the round-robin. The CMCU broadcasts a command to all the DMCAs, which causes Co2, Co3, Co4, and Co5 to shut down as well. Finally, after a specified period of 3 seconds, the main dc bus contactor opens, effectively isolating the load from the BESS.

[1]CCSTUDIO stands for Code Composer Studio integrated development environment

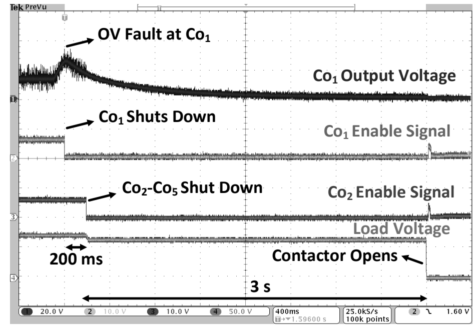

Fig. 7. Hardware Results

IV. Conclusions and Future work

In this paper, we present a prototype of a hierarchical SMC for power conversion and batteries in a BESS that is reliable, scalable, and flexible. Our evaluation of a hardware testbed shows that the essential functions of monitoring, control, and fault handling operate within an acceptable latency in the SMC prototype. In the future, we plan to use the data collected from the prototype to scale the system's higher energy storage with more batteries. We intend to use a real-time operating system (RTOS) for better time-critical control, further enhancing reliability and scalability. Additionally, we plan to integrate the BMSs into the hierarchical control structure to increase the granularity of capability in the BESS.

References

[1] X. Cui, A. Ramyar, J. B. Siegel, P. Mohtat, A. G. Stefanopoulou, and A.-T. Avestruz, "Comparing power processing system approaches in second-use battery energy buffering for electric vehicle charging," *Journal of Energy Storage*, vol. 49, p. 104017, 2022.

[2] H. Engel, P. Hertzke, and G. Siccardo, "Second-life EV batteries: The newest value pool in energy storage," *McKinsey & Company*, 2019.

[3] A. Ramyar, W. Xu, X. Cui, J. Siegel, A. Stefanopoulou, and A.-T. Avestruz, "Lite-Sparse Hierarchical Partial Power Processing for Heterogeneous Degradation of Batteries in Energy Storage Systems," in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*, Detroit, MI, USA, 2022, pp. 1-7.

[4] A. V. Grishin, A. R. Asoyan, A. A. Solntsev, V. V. Guly, and O. V. Fediushkina, "Ensuring the Operation of Electric Bus Batteries in the Optimal Temperature Range," in *2022 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED)*, Moscow, Russian Federation, 2022, pp. 1-4.

[5] X. Luo, D. Obadina, and M. Boddeti, "The roles of energy management system in Texas Nodal Power Market," in *2009 IEEE Power & Energy Society General Meeting*, Calgary, AB, Canada, 2009, pp. 1-7.

[6] H. Shafique, L. B. Tjernberg, D.-E. Archer, and S. Wingstedt, "Energy Management System (EMS) of Battery Energy Storage System (BESS) – Providing Ancillary Services," in *2021 IEEE Madrid PowerTech*, Madrid, Spain, 2021, pp. 1-6.

[7] H. Shafique, D.-E. Archer, R. Eriksson, and L. B. Tjernberg, "Real-time Operation Model for Energy Management System of Battery Energy Storage System - Case Study: The School of Sinntorp," in *2022 17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Manchester, United Kingdom, 2022, pp. 1-6.

[8] X. Cui et al., "Lite-Sparse Hierarchical Partial Power Processing for Second-Use Battery Energy Storage Systems," in *IEEE Access*, vol. 10, pp. 90761-90777, 2022.

[9] D. Chen, M. Huang and A. G. Stefanopoulou, "Discrete Mixed-Integer Shooting (DMIS): Algorithm and Application to Plug-In Hybrid Electric Vehicle Energy Management Accounting for Fuel Cranking and Actual Powertrain Efficiency Maps," in *IEEE Transactions on Control Systems Technology*, vol. 31, no. 1, pp. 221-234, Jan. 2023.

[10] J. S. Edge, S. O'Kane, R. Prosser, N. D. Kirkaldy, A. N. Patel, A. Hales, et al., "Lithium-ion battery degradation: What you need to know", *Phys. Chem. Chem. Phys.*, vol. 23, no. 14, pp. 8200-8221, Apr. 2021.

[11] T. Su, N. Lyu, Z. Zhao, H. Wang, and Y. Jin, "Safety warning of lithium-ion battery energy storage station via venting acoustic signal detection for grid application," *Journal of Energy Storage*, vol. 38, p. 102498, 2021.

[12] H. A. Gabbar, A. M. Othman, and M. R. Abdussami, "Review of battery management systems (BMS) development and industrial standards," *Technologies*, vol. 9, no. 2, p. 28, 2021.

[13] M. T. Lawder, B. Suthar, P. W. Northrop, S. De, C. M. Hoff, O. Leitermann, M. L. Crow, S. Santhanagopalan, and V. R. Subramanian, "Battery energy storage system (BESS) and battery management system (BMS) for grid-scale applications," *Proceedings of the IEEE*, vol. 102, no. 6, pp. 1014–1030, 2014.

[14] P. S. Babu and K. Ilango, "Comparative Analysis of Passive and Active Cell Balancing of Li-Ion Batteries," 2022 *Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, Kannur, India, 2022, pp. 711-716.

[15] D.-s. Back, K. Pyun, S.-M. Lee, J. Cho, and N. Kim, "A Hierarchical Deficit Round-Robin Scheduling Algorithm for a High Level of Fair Service," *2007 International Symposium on Information Technology Convergence (ISITC 2007)*, Jeonju, Korea (South), 2007, pp. 115-119.

[16] A. Baranowska and W. Kabacinski, "Hierarchical round-robin matching for virtual output queuing switches," *Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)*, Lisbon, Portugal, 2005, pp. 196-201.