Ref. Ares(2024)4696312 - 29/06/2024

BATTERY2LIFE

D1.3 BMS Specifications and Advanced Algorithms for 2nd Life

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor CINEA can be held responsible for them.

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

DOCUMENT CONTROL TABLE

Work Package	WP1
Task	T1.3, T1.4, and T1.5
Author(s)	I. Aizpurua (CID), Y. Stauffer (CSEM), D. Spanoudaki (SLG), M. Koller (AIT), L. Thumfart (MIBA)
Dissemination Level	Public
Status	Final
Due Date of submission	30/06/2024
Actual Date of submission	28/06/2024
Document Date	27/06/2024
Version	1.0
Project Name	BATTERY Management system and System design for stationary energy storage with 2nd LIFE batteries
Project Acronym & Number	Battery2Life 101137615
Project Website	www.battery2life-project.eu
Project Social Media	@battery2life_eu & BATTERY2LIFE Project

QUALITY CONTROL

	Name	Organisation	Date
Editor	I. Aizpurua	CID	12/06/2024
Reviewer 1	P. Panagiotis	PPC	20/06/2024
Reviewer 2	R. Stachl	MIBA	21/06/2024
Reviewer 3	Friederike L. Kühl	ETE	19/06/2024
Authorised by Quality Manager	E. Karfopoulos	ICCS	26/06/2024
Authorised by Project Coordinator	A. Amiditis	ICCS	28/06/2024

Co-funded by the European Union

DOCUMENT HISTORY

Version	Date	Modification	Partner
V0.1	12/04/2024	Table of Content	I. Aizpurua (CID)
V0.2	11/06/2024	First draft pre-formatting	I. Aizpurua (CID), Y. Stauffer (CSEM), D. Spanoudaki (SLG), M. Koller (AIT), L. Thumfart (MIBA)
V0.3	12/06/2024	First draft pre-review	I. Aizpurua (CID)
V0.4	21/06/2024	First draft post-review	P. Panagiotis (PPC), R. Stachl (MIBA), Friederike L. Kühl (ETE)
V0.5	25/06/2024	Second draft pre-review	I. Aizpurua (CID)
V1.0	28/06/2024	Final version	E. Karfopoulos (ICCS)

LEGAL DISCLAIMER

CONTENTS

Document Control Table 2
Quality control 2
Document History 3
Legal Disclaimer 4
Contents 5
Table of Figures 6
List of Tables 7
List of Abbreviations and Acronyms 8
Executive Summary 10
1 Introduction 11
1.1 Project Introduction 11
1.2 Purpose of the deliverable 12
1.3 Intended audience 12
1.4 Structure of the deliverable/correlation with other WPs 12
2 State of Art of BMS HW, SW and algorithms 13
2.1 HW topology 13
2.1.1 State of Art 15
2.1.2 BMS topology evaluation 18
2.2 SW architecture 22
2.2.1 State of Art 23
2.2.2 Battery2Life SW architecture requirements 23
2.3 BMS algorithms 24
2.3.1 State of Art 24
2.3.2 Battery2Life proposal evaluation 28
3 Comparison between Pillars 32
4 Pillar 1: Industrial load levelling BMS specifications 33
4.1 Selected HW topology specifications 33
4.2 SW architecture specifications 36
4.3 Advanced BMS algorithms specifications 37

Co-funded by the European Union

4.3.1 EIS-based SoC ... 37

4.3.2 SoH ... 39

4.3.3 EIS-based SoT ... 41

4.3.4 EIS-based SoS ... 42

4.3.5 SoW ... 44

5 Pillar 2: Domestic storage BMS specifications ... 49

5.1 Selected HW topology specifications ... 49

5.2 SW architecture specifications ... 52

5.3 Advanced BMS algorithms specifications ... 53

5.3.1 SoC and SoH ... 53

5.3.2 SoS-based SoF ... 55

5.3.3 SoW ... 57

5.3.4 Active balancing ... 61

5.3.5 SoX based on smart sensing ... 62

6 Conclusion ... 65

7 References ... 69

TABLE OF FIGURES

Figure 1: Battery2Life solutions. ... 11

Figure 2: High level view of the Battery pack and the Battery Management System relation. [1] ... 13

Figure 3: A LiFePO4 battery pack. In the front view, we observe the battery management system board. [3] ... 14

Figure 4: The three general topologies for the BMS hardware architecture. [1] ... 15

Figure 5: Battery application vs. voltage level and BMS topology. ... 19

Figure 6: Radar diagram of the different BMS topology comparison based on defined factors. ... 21

Figure 7: Key components of a BMS SW architecture. ... 22

Figure 8: Battery2Life Pillar 1. ... 33

Figure 9: Advanced BMS algorithms in Pillar 1. ... 37

Figure 10: Qualitative states of SoW. ... 46

Figure 11: Battery2Life Pillar 2. ... 49

Figure 12: Software structure for BMS in Pillar 2. 52
Figure 13: Advanced BMS algorithms in Pillar 2. 53
Figure 14: Qualitative states of SoW. 58
Figure 15: Advanced BMS algorithms in Pillar 1. 65
Figure 16: Software structure for BMS in Pillar 2. 67
Figure 17: Advanced BMS algorithms in Pillar 2. 67

LIST OF TABLES

Table 1: BMS topology suitability depending on application.. 19
Table 2: BMS topology comparison based on selected factors. 20
Table 3: Evaluation of BMS algorithms – Pillar 1. 30
Table 4: Evaluation of BMS algorithms – Pillar 2. 31
Table 5: Evaluation of BMS algorithms. 32
Table 6: BMS HW topology of Pillar 1. 34
Table 7: BMS HW topology of Pillar 2. 50

LIST OF ABBREVIATIONS AND ACRONYMS

Abbreviation	Meaning
API	Application Programming Interface
ASIC	Application-specific integrated circuit
BESS	Battery Energy Storage System
BMS	Battery Management System
BS	Battery System
CAN	Controlled Area Network
DRT	Distribution of Relaxation time
ECM	Equivalent Circuit Model
EIS	Electromechanical Impedance Spectroscopy
EKF	Extended Kalman Filter
EoL	End of Life
EP	Enable Pin
ESS	Energy Storage System
EV	Electric Vehicle
FEC	Full Equivalent Cycles
FW	Firmware
GA	Genetic Algorithm
HAL	Hardware Abstraction Layer
HIO	H-Infinity Observer
HW	Hardware
IC	Integrated Circuit
ID	Identifier
IP	Intellectual Property
ISC	Internal Short Circuit
KF	Kalman Filter
LIB	Lithium-Ion Battery
LFP	Lithium-ferrophosphate
LIN	Local Interconnected Network
LO	Luenberger Observer
MCU	Model Control Unit
MMS	Module Management System
N/A	Not Applicable.
NMC	Nickel, Manganese and Cobalt cathode component
NN	Neural Network
OCV	Open Circuit Voltage
OEM	Original Equipment Manufacturer

PCB	Printed Circuit Board
PDE	Partial Derivative Equation
PIO	Proportional Integral Observer
RAM	Random Access Memory
RH	Remaining Health
RTOS	Real-Time Operating System
RUL	Remaining Useful Life
RUW	Remaining Useful Warranty
RW	Remaining Warranty
SEI	Solid Electrolyte Interface
SMO	Sliding Mode Observer
SoA	State-of-Art
SOA	Safety Operating Area
SoC	State-of-Charge
SoF	State-of-Function
SoH	State-of-Health
SoS	State-of-Safety
SoT	State-of-Temperature
SoW	State-of-Warranty
SoX	State-of-X (involving all estimators)
SPI	Serial Peripheral Interface
SVM	Support Vector Machine
SW	Software
TRA	Thermal Runaway

Co-funded by the European Union

EXECUTIVE SUMMARY

This document outlines the specifications for the HW (Hardware) topology, SW (Software) architecture, and advanced BMS (Battery Management System) algorithms for the Battery2Life solutions across Pillars 1 and 2. The objective is to frame an open adaptable smart BMS for 2nd life batteries. In this deliverable, a reliable, interoperable, safe and scalable BMS SW architecture is conceptualised. In addition, HW topology is selected to ensure accessibility, adaptability and customization from 1st life to 2nd life. And finally, safety and life forecasting SoX (State-of-X) and advanced functionalities are specified.

For **Pillar 1**, a modular BMS topology will be implemented where each module is connected to a slave board, and a master board coordinates them. Communication between master and slave boards will be wireless, and an ESP32 module will handle cloud communication. The SW architecture follows AUTOSAR principles, separating HW-dependent Firmware (FW) from HW-independent application SW, ensuring interoperability and facilitating FW-over-the-air updates. Advanced BMS algorithms include EIS-based SoC (State-of-Charge) estimation, SoH (State-of-Health) based on differential-capacity and Coulomb counting, EIS-based SoT (State-of-Temperature) for internal temperature estimation, EIS-based SoS (State-of-Safety) for safety monitoring, and a multi-maintenance approach for SoW (State-of-Warranty). Some of them are computed on the cloud.

For **Pillar 2**, a modular BMS topology with a master-slave structure will be used. The main components include the module management system, BMS master controller, battery junction box, and wireless communication gateway. Additionally, advanced sensing will be integrated at laboratory level for high accuracy SoX estimation. The SW architecture includes two microcontrollers for mutual monitoring and error correction, running on FreeRTOS with Hardware Abstraction Layer (HAL) generated by HALCoGen. Key estimators and functionalities include SoC estimation using an Extended Kalman Filter (EKF), SoH based on capacity difference, SoS-based SoF (State-of-Function) for current limiting, and the same multi-maintenance approach for SoW as in Pillar 1. Additionally, active balancing will be integrated.

Overall, the document presents a comprehensive approach to developing reliable, interoperable, safe, and scalable BMS solutions for both pillars, accommodating both open-source and IP-restricted (IP, Intellectual Property) blocks and ensuring robust battery management through advanced algorithms and flexible software architecture.

1 INTRODUCTION

In this section project introduction, the purpose of the deliverable, the intended audience and the structure of the deliverable are presented.

1.1 Project Introduction

BATTERY2LIFE is a project, funded by Horizon Europe programme that will facilitate the smooth transition of batteries to 2nd life use and boost the innovation of the European Battery Industry by providing enablers to implement open adaptable smart Battery Management Systems (BMS) and improved system designs and proposing methods for the efficient and reliable reconfiguration of used batteries.

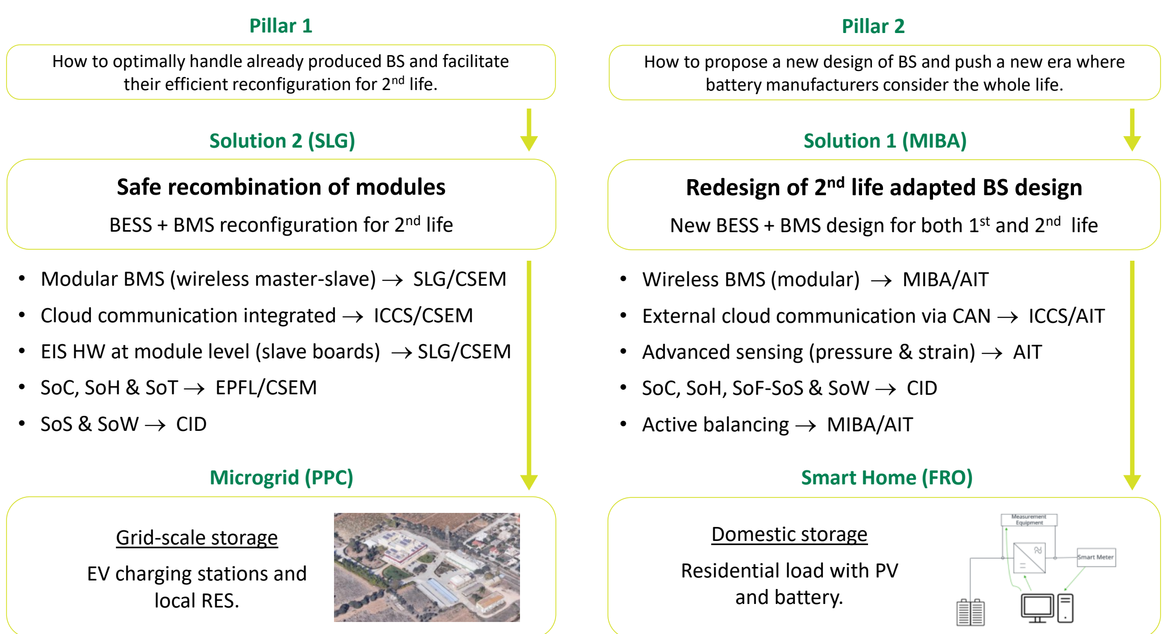


Figure 1: Battery2Life solutions.

Battery2Life introduces two new battery system design frameworks serving the upcoming market needs: the first supports the business transition for the initial market by remanufacturing based on existing battery systems while the second one introduces completely new design principles for 1st and 2nd life of the battery. Furthermore, Battery2Life introduces innovative embedded sensing and more accurate SoX (State-of-X) estimation algorithms, new SoX indicators appropriate for 2nd life use -i.e. SoS (State-of-Safety) and SoW (State-of-Warranty) - and a new EIS (Electromechanical Impedance Spectroscopy) implementation approach by integrating it in the BMS, that will enable the detailed safety and reliability monitoring at both cell and module level during 1st and 2nd life usage. The project will specify an open BMS concept, data formats, considering and extending the battery passport concept, and interoperable communication via the cloud platform to third parties including the future passport exchange system, to facilitate monitoring and assessment.

Two demonstrations that represent two promising and sustainable business cases, serving the two most common stationary applications have been carefully selected: The industrial (grid-scale) and domestic storage (Pillar 1 and Pillar 2), with respect to their operational specificities and requirements.

1.2 Purpose of the deliverable

This deliverable aims to serve three objectives:

- Define the specifications to develop the project BMS hardware (HW) topology for 2nd life.
- Define the specifications to develop the project BMS software (SW) architectures.
- Define the requirements to develop advanced BMS algorithms.

1.3 Intended audience

Deliverable D1.3 is a public document aiming to define the specifications and requirements to develop project BMS HW, SW and advances algorithms which are later being integrated on a domestic and industrial 2nd life Battery Energy Storage System (BESS). It is aimed at the general public, but more specifically at scientific and technological profiles working with BESS.

1.4 Structure of the deliverable/correlation with other WPs

The deliverable comprises four sections:

- **Section 2 – State of Art of BMS HW, SW and algorithms:** First HW topology, SW architecture and BMS algorithms SoA will be presented and evaluated considering application, scale, reliability, safety, and efficiency requirements.
- **Section 3 – Comparison between Pillars:** In this section the Battery2Life solutions are summarized and compared in HW, SW and algorithmic level.
- **Section 4 – Pillar 1, industrial load levelling BMS HW, SW and algorithms specifications:** Once SoA is evaluated, in this section the selected HW, SW and included algorithms' specifications of Pillar 1 are presented.
- **Section 5 – Pillar 2, domestic storage BMS HW, SW and algorithms specifications:** As with Pillar 1, specifications of Pillar 2 HW, SW and algorithms are also presented in a different section since different approach are being developed.

Co-funded by the European Union

2 STATE OF ART OF BMS HW, SW AND ALGORITHMS

Developing a BMS for 2nd life batteries requires a thorough understanding of battery chemistry, skills in circuit design, programming, and data communication. Both, BMS HW topology and SW architecture offering multi-tasking capabilities need to be re-defined in terms of safety, sensitivity, and communication. Together with SoX algorithms and new functionalities, they will enable the development of an open adaptable smart BMS which ensures the reliability, safety and energy efficiency of the system.

In the next section, existing BMS HW topologies, SW architectures and algorithms are presented and evaluated to find the most optimal technical and economical solutions.

2.1 HW topology

The "brains" of a battery pack is named Battery Management System (BMS). Several electronic components are combined to build a BMS, including the Master Printed Circuit Boards (PCBs), the slave PCBs, sensors and the associated software that is responsible for the communication of all these components. Someone could also consider the control electronics, such as the switches, fuses, high-voltage front-end, high-voltage interlock loop and disconnects.

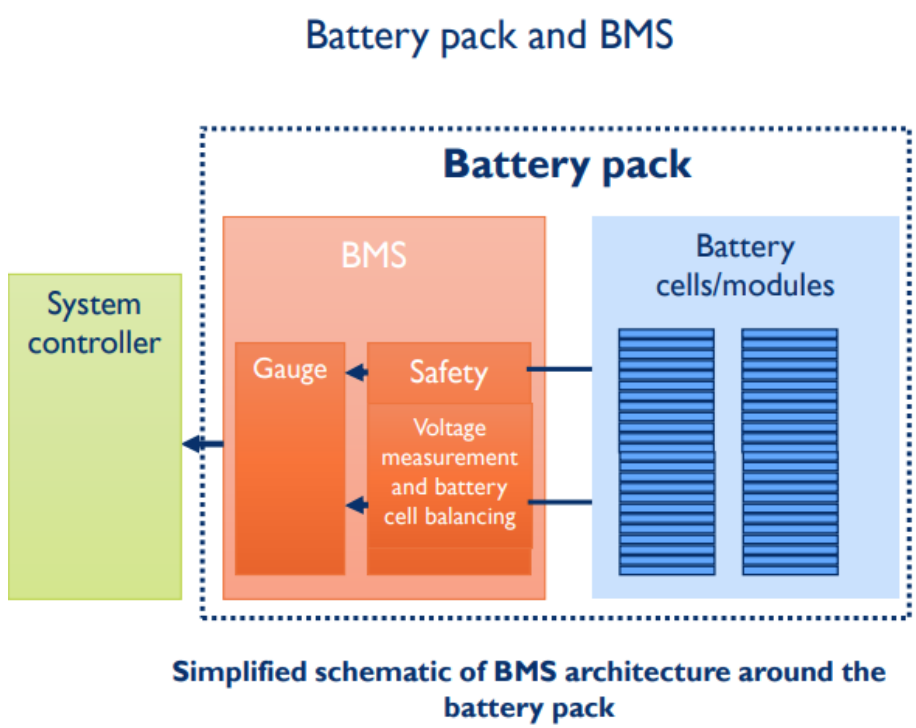


Figure 2: High level view of the Battery pack and the Battery Management System relation. [1]

The purpose of a BMS in a battery is to:

- Protect human safety of device's operator and detect unsafe operating conditions and respond.
- Protect the cells of a battery from damage in abuse/failure cases.
- Prolong the life of the battery in normal operating conditions.
- Maintain the battery in a state in which it can fulfil its functional design requirements.
- Inform the application controller how to make the best use of the pack right now (e.g. by providing power limits, control charger, etc.).

All the above requirements are met, because the BMS is designed to provide protection against overcharging, over discharging, high temperatures, low temperatures, short circuits, and other failure modes. BMS manages to provide protection by monitoring the state of the battery and cells, communicating within and outside the pack with controllers and systems and by maximising the performance of the battery. [2]

While it is quite understandable the purpose of a BMS in a battery, it doesn't answer the question why specific type of batteries like the Li-ion, need a BMS. The main reason is that Li-ion chemistry is more dangerous than other chemistries (e.g. lead-acid) in cases of overcharge or over discharge and extreme conditions in general. For example, when a Li-ion cell over discharges (below 2.1 V) then the electrolyte decomposes, leading to the formation of an additional Solid Electrolyte Interface (SEI) layer, and thus reducing the available capacity of the cell. Other older battery chemistries, such as lead-acid, alkaline batteries (etc.) do not have the same issue.

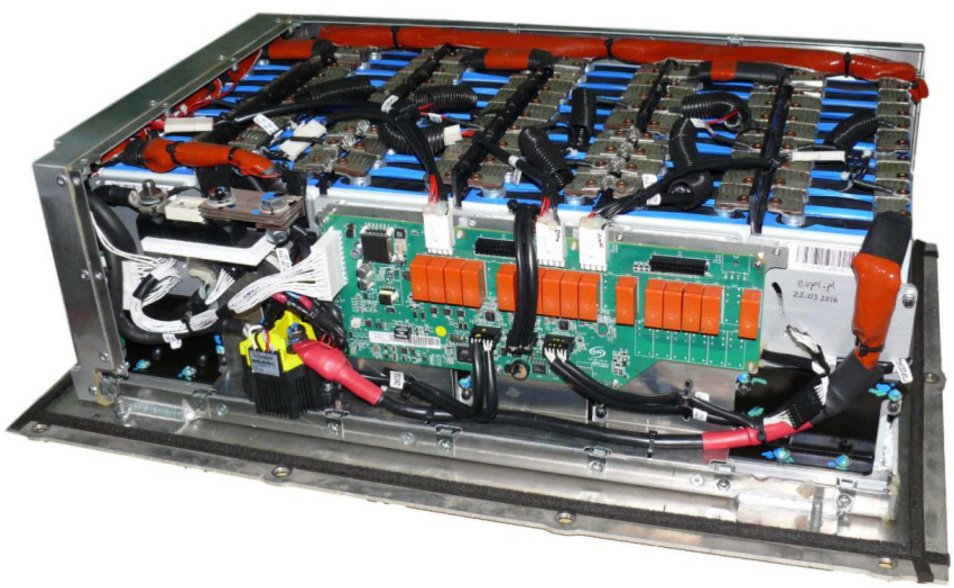


Figure 3: A LiFePO4 battery pack. In the front view, we observe the battery management system board. [3]

2.1.1 State of Art

To assess the State-of-art (SoA) BMS topologies, we first must discuss its main components and their role during the operation of the battery. The BMS hardware includes one or more PCBs that integrates all the components that make up a controller board, including Controlled Area Network (CAN), Local Interconnected Network (LIN), wireless and other communications components, capacitors, resistors, current sensors, and most importantly the Application-Specific Integrated Circuits (ASICs).[2]

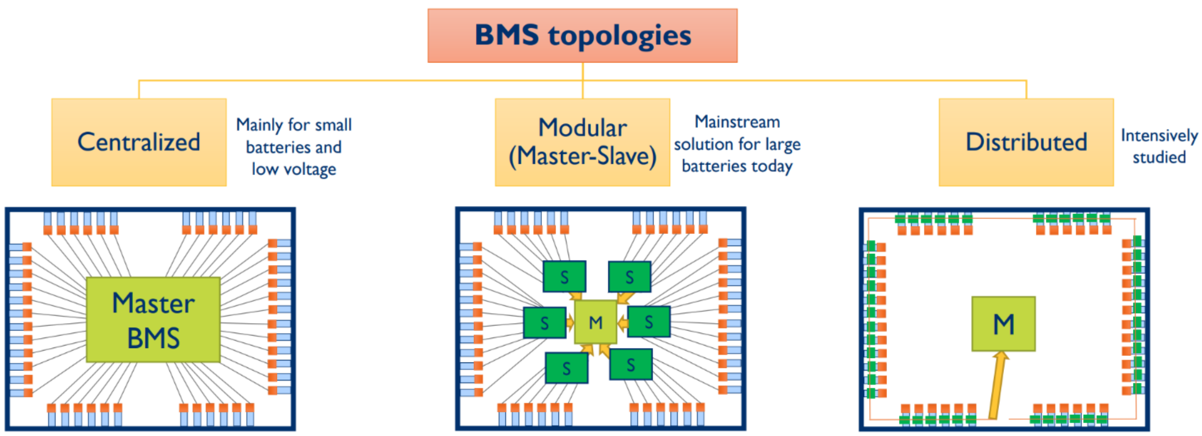


Figure 4: The three general topologies for the BMS hardware architecture. [1]

- **Centralised BMS topology**

In a centralised BMS topology all the cells and modules in a battery pack are monitored and controlled by a single BMS system, that is the Master unit. This topology has its limitations and its advantages.

The limitations are:

1. **Limited scalability:** The centralised topology faces challenges in handling larger and more complex battery packs, as the central control unit may be overwhelmed with a high number of battery cells to monitor and control.
2. **Signal interference and voltage drop:** For larger packs the centralised BMS topology needs extended wiring, which increases the risk of signal interference and voltage drop.
3. **Single point of failure:** Since the entire BMS relies on the central control unit, any malfunction or failure of this unit could potentially lead to a complete system shutdown.

The advantages of a centralised BMS topology are:

1. **Simplicity:** By integrating all tasks under a single controller, the system becomes more straightforward in design, assemble, and maintain.

2. **Cost-effectiveness:** Especially for small battery packs, the centralised BMS topology is more cost-effective. A rule of thumb would be that if a battery pack is cheap, small and contains few components, then a centralised BMS topology should be the first choice.

The limitations and advantages of the centralised BMS topology give a clue about the main use cases it applies, that is applications that require small battery systems, with no complex architecture and assembly, and with a low price relatively to other applications that use batteries. Electric bikes, scooters, and other light electric vehicles are prominent examples of applications for them.

The main components of a Master Unit are:

1. **Central Processor:** The central processor collects data from various battery cells or modules and analyses this information to determine the status of the battery pack.
2. **Cell Monitoring Circuits:** Each battery cell or module is equipped with a monitoring BMS circuit that measures parameters such as voltage, current, and temperature. These circuits transmit data to the central processor.
3. **Communication Interfaces:** Communication interfaces facilitate data exchange between the central processor and the individual cell monitoring circuits. Some of these are CAN interfaces, LIN interfaces and function via wireless communication.
4. **Balancing circuits:** The central processor can command balancing circuits to equalise the charge across the cells by redistributing energy, in case of active balancing, or to convert the excess of energy to heat, in case of passive balancing.
5. **Safety feature:** The centralised BMS includes safety features like overcurrent protection, overvoltage protection, and overtemperature protection to safeguard the battery cells from potential hazards.

- **Distributed BMS topology**

In contrast to the centralised approach, a distributed BMS topology distributes intelligence across each cell or multiple groups of cells. Each cell or group of cells (module) has a unique BMS. These individual BMSes are referred to as “nodes”, and each node individually monitors, balances, and safeguards its own system. The nodes communicate with each other to coordinate their actions and collectively manage the entire battery pack. Typically, they use CAN or LIN protocols to interact via a shared bus. Even though each node functions independently, they work together to maintain the effectiveness, security, and efficiency of the entire system.

The limitations of a distributed BMS are:

1. **Increased complexity:** A distributed BMS requires more advanced communication protocols and coordination between nodes, leading to increased system complexity.

Additionally, as the number of nodes grows, it might be difficult to ensure regular and dependable communication between the nodes.

2. **Higher cost:** The redundancy and additional communication protocols can increase the initial implementation costs compared to a centralised BMS.

The advantages of a distributed BMS are:

1. **Improved reliability:** The distribution of intelligence ensures that even if a node fails, the rest of the system can continue functioning, enhancing reliability. Also, because of each node's ability to accurately regulate the cells it is connected to, it is also possible to accomplish more exact cell monitoring and balancing.
2. **Scalability:** Distributed BMS can accommodate larger battery systems without overwhelming a central processor, making it ideal for applications requiring significant expansion capabilities. The cell may be scaled by simply adding or removing cells and their accompanying controllers because each cell or module has its own controller.

The main components in a BMS, from a HW point of view are:

1. **Node controllers:** Each battery cell or module is associated with its dedicated node controller. These node controllers are responsible for monitoring the individual cells and reporting their status to neighbouring nodes.
2. **Communication Network:** The distributed approach relies on a communication network, such as CAN or Ethernet, which connects the node controllers. Nodes exchange information and collectively manage the battery system.
3. **Coordination and Decision Making:** The node controllers communicate with each other to coordinate actions and make collective decisions for balancing and protection, ensuring that the entire battery pack operates optimally.
4. **Balancing Circuits and Safety Features:** Each node includes balancing circuits and safety features to maintain cell balance and protect the battery cells from potential hazards.

- **Modular BMS topology**

The modular BMS is a middle-solution that combines the advantages of both centralised and distributed designs to offer a scalable and flexible battery management solution.

In a modular BMS configuration, the system is partitioned into multiple identical modules, each tasked with monitoring and overseeing a specific subset of the battery pack's cells or modules. A slave PCB is attributed to every module, and it is responsible for monitoring cell voltage, temperature, current and send this information to the master PCB. In response, from the master PCB it executes control directives ensuring the cell safety.

The limitations of a modular BMS are:

1. **Communication complexity:** The interconnection between modules requires a well-defined communication protocol to ensure proper coordination and control.
2. **Balancing Challenges:** Ensuring proper balancing across modules can be more challenging than within a single centralised or distributed unit.

The advantages of a modular BMS are:

1. **Flexibility:** Modular BMS allows from a flexible design, making it suitable for various battery configurations and accommodating different application requirements.
2. **Easy integration:** Each module operates independently, simplifying integration with other vehicle or system components, such as power inverters or chargers.
3. **Reduced Downtime:** In the event of a BMS failure, only the affected module needs to be repaired or replaced, minimising downtime and ensuring continuous operation.

The main components of a modular BMS are:

1. **Battery modules:** The battery pack is divided into individual modules, each containing a group of battery cells. These modules can be connected in series or parallel configurations based on system requirements.
2. **Module Controllers:** Each battery module is equipped with its BMS controller, responsible for monitoring and managing the cells within that specific module.
3. **Inter-Module Communication:** Modules communicate with each other and a central coordinator to exchange data and coordinate actions, such as balancing and protection, to ensure the entire battery pack operates efficiently.
4. **Central Coordinator:** In some cases, a central coordinator may be included to manage communication between modules and oversee the overall battery management system.

2.1.2 BMS topology evaluation

In this section the presented BMS topologies are evaluated, first from the application point of view, and then based on HW significant factor.

- **BMS topology and application needs**

In Figure 5 different battery applications can be seen, as well as the used BMS topology. The frontier between the use of a centralised BMS or Master-Slave BMS is at the voltage level, at about 48 V to 72 V.

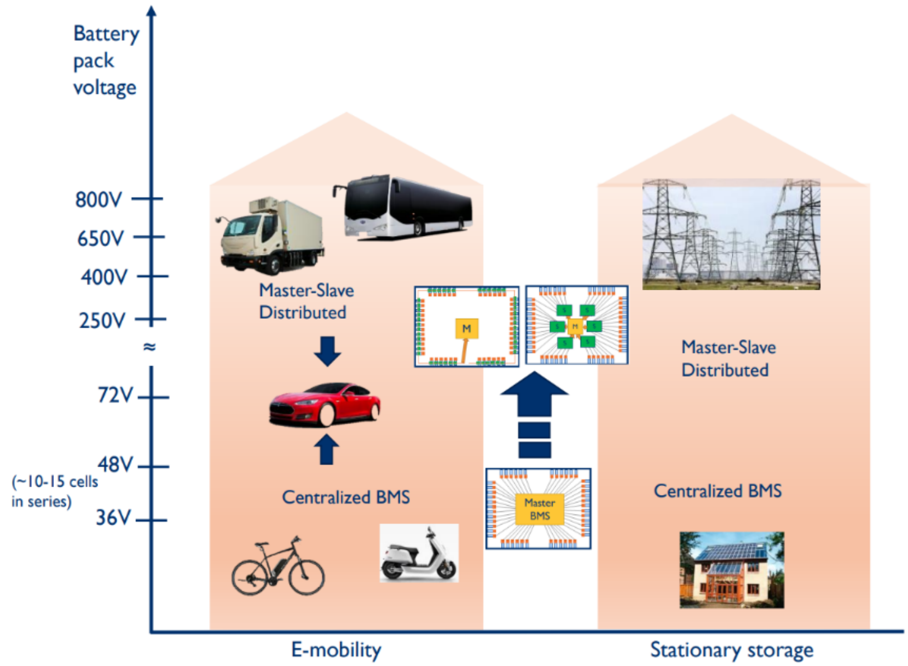


Figure 5: Battery application vs. voltage level and BMS topology.

Considering application necessities, BMS topologies have different suitability as it can be seen in Table 1.

Table 1: BMS topology suitability depending on application.

BMS topology	Suitability	Considerations
Centralised	Centralised BMS is suitable for smaller battery systems with relatively simple architectures. It is commonly used in applications where cost and simplicity are essential factors, such as small electric vehicles, portable devices, and low-power energy storage systems.	For larger and more complex battery systems. The limited scalability and fault tolerance of a centralised BMS may become a concern, as it may struggle to handle the increasing number of cells and communication links.
Distributed	Distributed BMS is ideal for larger battery systems with high scalability requirements, such as electric buses, grid energy storage, and industrial energy storage solutions. It offers excellent fault tolerance and redundancy, making it suitable for critical applications where system downtime must be minimised.	The complexity of communication between nodes may increase implementation costs, and careful coordination is essential to ensure seamless cooperation between distributed nodes.
Modular	Modular BMS is well-suited for applications that require flexibility and scalability. It is commonly used in electric vehicles, data centres, and large-scale storage systems where modules can be added or removed as needed, allowing for easy expansion and maintenance.	While offering high flexibility, the communication complexity between modules and the challenge of balancing across modules should be carefully addressed during the design phase.

- **BMS topology comparison based on factors**

Additionally, various evaluation factors have been chosen to compare the three main BMS topologies. The meaning of the factors is listed below:

Scalability: Refers to the ability of a BMS to adapt and manage larger and more complex battery systems.

Flexibility: Refers to the ability of a BMS to adapt to different types of battery configurations and requirements. For example, to be adaptable to different vehicles or applications or to be compatible with different battery chemistries.

Fault tolerance: Refers to the ability of a system to operate even if a fault occurs. A fault-tolerant BMS often includes redundant circuits and components, it can diagnose various faults in the battery system, and it has mechanisms to handle the fault effectively.

Redundancy: Refers to the inclusion of additional or duplicate components to enhance the reliability and safety of the system. Just like the fault-tolerance, redundancy is about ensuring that a BMS can safely operate even if a fault occurs.

Communication Complexity: It refers to the process of exchanging information between the BMS and other systems or components. It is influenced by several factors, including the size and complexity of the system, required data transmission speed, noise immunity, and cost.

Integration Simplicity: It refers to the ease with which the BMS can be incorporated into other systems and components.

Considering the listed factors, BMS topologies can be evaluated as in Table 2 and Figure 6.

Table 2: BMS topology comparison based on selected factors.

Aspect	Centralized	Distributed	Modular
Scalability	Limited	High	High
Flexibility	Limited	High	High
Fault tolerance	Low	High	Moderate
Redundancy	Minimal	High	Moderate
Communication Complexity	Low	High	Moderate
Integration Simplicity	High	Moderate	High
Cost	Low	Moderate	Moderate

Co-funded by the European Union

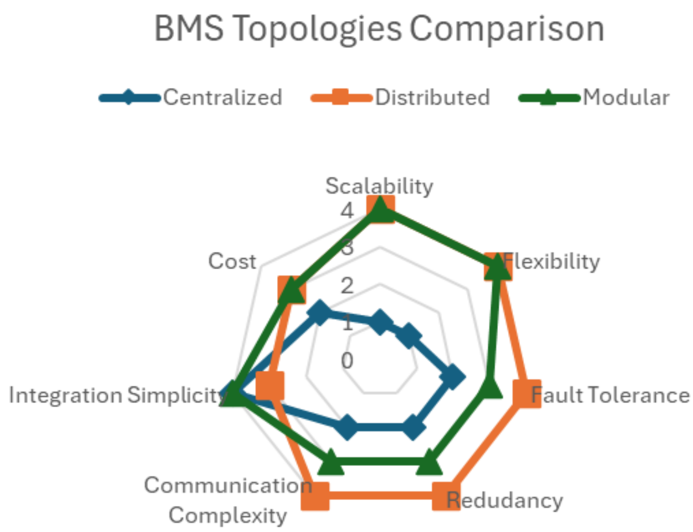


Figure 6: Radar diagram of the different BMS topology comparison based on defined factors.

2.2 SW architecture

The embedded software architecture of a Battery Management System (BMS) is a critical component in ensuring the efficient and safe operation of battery-powered systems, such as electric vehicles or portable devices. The architecture is designed to monitor the battery's state, collect and process data, implement control algorithms, ensure safety, and communicate with other systems.

This architecture is composed of several key components: Battery Monitoring, Data Acquisition, Data Processing, Control Algorithms, Safety Management, and a Communication Interface. The design of the architecture can be made modular to allow for easy upgrades and modifications, catering to the evolving needs and demands of the system. The BMS plays a vital role in optimising the performance of the battery, extending its lifespan, and preventing any potential failures or damages.

The architecture for an embedded software system for BMS can devided into several key components. Here is a basic overview:

- **Battery Monitoring**: This component is responsible for monitoring the battery's current, voltage, temperature, and State of Charge (SoC).
- **Data Acquisition**: This component collects the data from the battery monitoring component and other sensors in the system.
- **Data Processing**: This component processes the data collected by the data acquisition component. This can involve filtering the data, calculating the state of health (SoH) of the battery, and other necessary computations.
- **Control Algorithms**: This component implements the control algorithms that manage the charging and discharging of the battery based on the data provided by the data processing component.
- **Safety Management**: This component ensures the safe operation of the battery. It monitors for any conditions that could lead to battery damage or failure, such as overcharging, overheating, or short-circuiting.
- **Communication Interface**: This component allows the BMS to communicate with other systems, such as the user interface or other parts of the vehicle or device that the battery is powering.

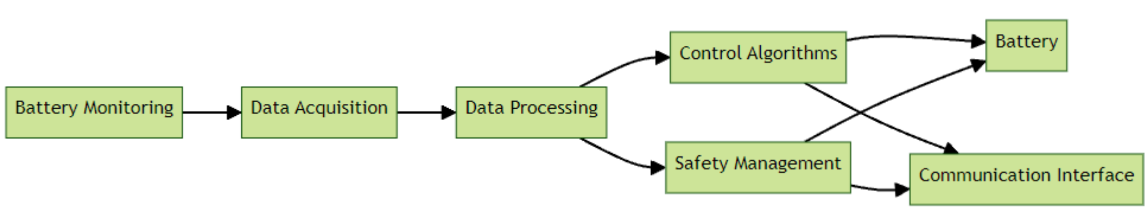


Figure 7: Key components of a BMS SW architecture.

2.2.1 State of Art

SoA Battery Management Systems are leveraging advancements in technology to enhance their performance and capabilities. This includes the integration of more sophisticated sensors for precise monitoring of battery parameters, advanced data processing algorithms for accurate SoH and SoC estimation, and improved control strategies for optimal battery utilisation. Machine learning and artificial intelligence techniques are also being explored to predict battery behaviour and make real-time adjustments to the control algorithms. Furthermore, communication interfaces are being designed to support various communication protocols for seamless integration with other systems. The use of wireless technologies is also becoming more common, allowing for remote monitoring and control. In terms of safety, advanced fault detection and diagnosis techniques are being implemented to prevent any potential battery failures. Overall, the SoA BMS are becoming more intelligent, reliable, and efficient, contributing to the broader goal of sustainable and clean energy utilisation.

To achieve an architecture which offers multi-tasking capabilities the usage of a Real-Time Operating System (RTOS) will be necessary. Both Pillars will use FreeRTOS and depending on safety goals the switch to SafeRTOS will be fulfilled. In modern embedded systems there are several embedded programming languages such as C, C++ and Rust. For a BMS covering safety measures the best practice approach will be using C.

2.2.2 Battery2Life SW architecture requirements

The following defines the characteristics set out to achieve an open-adaptable BMS, regarding the use of Restricted IP-Blocks (IP, Intellectual Property), multi-tasking capability, communication interfaces, and connection to the cloud.

- **Use of Restricted IP-Blocks:** The consideration is to use precompiled libraries which is the most common way to use protected code. The drawback of backboxes is that the implementor is not able to predict the behaviour in terms of stability.
- **Multi-tasking capability:** To achieve this either a “super loop” or an embedded operating system is needed. With therefore a stand of the art approach to use RTOS with further features was chosen.
- **Communication interfaces:** The communication interfaces in both pillars were chosen to meet the requirements of the higher-level system. Also, the interoperability between 1st and 2nd life mainly is based on these interfaces.
- **Connection to the Cloud:** Since the system is going to be a request-response interface (response is at BMS side). Therefore, a buffer for incoming messages is needed to cache the requests and handle them.

2.3 BMS algorithms

In this section, BMS algorithms SoA recompilation is presented, together with the Battery2Life proposal. At the end, Battery2Life algorithms are summarised per pillar on a table.

2.3.1 State of Art

A BMS is an electronic system that monitors and manages multiple critical battery parameters and functions, such as charge and discharge, cell balance, temperature, SoC, SoH and safety. The BMS ensures optimal performance, extended life and safe operation of batteries in a wide range of applications.

The first estimator that refers to the available energy on a battery is the SoC. It indicates the percentage of stored electric energy vs its maximum storage capacity, with 100% indicating a fully charged battery and 0% indicating a fully discharged one.

There are multiple options for the estimation of the SoC [4,5]:

- **Look-up tables** based on experimental data, where SoC is estimated based on the mapping relationship between characteristic parameters such as impedance, internal resistance, or OCV and the SoC.
- **Ampere-hour integral** method, most known as Coulomb Counting.
- **Filter-based** methods can be grouped into two categories: the Gaussian process-based filter approaches (different Kalman Filter, KF, approaches) and the probability-based filter approaches (different Particle Filter approaches).
- **Observer-based** method estimates the state of the battery based on external variables. Luenberger Observer (LO), Sliding Mode Observer (SMO), Proportional-Integral Observer (PIO), and H-Infinity/H_∞ Observer (HIO) have been used extensively for battery state estimation.
- The **data-driven** based methods learn the internal dynamics through large amounts of measurable input and output data. The commonly used methods include Neural Network (NN), fuzzy logic, Genetic Algorithm (GA), Support Vector Machine (SVM), between others.

In order to adjust the estimation of the SoC to the actual capacity during operation, an additional parameter is necessary to include degradation effect. This estimator is the SoH which indicates the current energy storage capacity compared to the original one, also by percentage. Previously mentioned methods can be used to measure the available total energy storage capacity. Both estimators, SoC and SoH can be applied at cell, module or system level.

Co-funded by the European Union

Battery SoH estimation methods can be divided into the empirical-model based, the physical-model based, and the data-driven based[6]. There is also an additional simple method based on the integration of charged capacity:

- The **Ampere-hour integral** method, where the actual capacity is calculated based on the difference between the theoretical charged Ah and the actual charged Ah at the selected SoC range.
- The **empirical model** is an empirical equation based on experiments with temperature, SoC, rate, and the number of cycles as independent variables. However, there are few studies to build an empirical model for 2^{nd} life applications.
- The **physical model** mainly consists of mathematical equations to describe the internal aging mechanism of the battery, such as the growth of SEI film, lithium plating, and active materials crack and electrolyte decomposition. They usually follow linear aging trajectory in 1^{st} life and nonlinear in 2^{nd} life.
- **Data-driven** SoH estimation consist of extracting features from test data during battery operation to train machine learning algorithms and build a mapping model between features and battery SoH.

At cell level, it is also interesting to evaluate the state of each cell in order to avoid unbalances and maintain optimum operation conditions. Cell balancing in battery management refers to keeping individual cells within a battery pack at similar levels of charge and discharge. This is achieved by transferring energy between cells to avoid over-discharging or over-charging. It is crucial to maximise the life and performance of the battery pack, as an imbalance can lead to premature degradation, reduced overall capacity and increased risk of thermal failure or fire. Therefore, cell balancing is essential to ensure safe and efficient operation of battery systems. There are different cell balancing approaches[7]:

- **Passive Balancing:** Uses balancing resistors to dissipate excess energy or bypass diodes to limit the charging current and equalise the cell voltages.
- **Active Balancing:** Transfers energy between cells using converter circuits, capacitors, or inductors/transformers to equalise their charge levels.
- **Hybrid Balancing:** Combines passive and active methods to improve balancing efficiency and speed.

Regarding the equalisation of the battery cells, passive balancing is currently used, which exploit voltage-based signals/algorithms. This mechanism is based on energy dissipation of the cells with higher SoC and/or voltage. Besides being inefficient, algorithms based on voltage signals can create error when under-loaded while cells age differently [8].

In 2^{nd} life batteries, where cells suffer already from a degradation, the heat generation on any charging and discharging process is higher. The state-of-temperature (SoT) refers to the current temperature of the battery and its internal thermal distribution. Monitoring the temperature is crucial to ensure safe and efficient battery operation, as temperature can significantly affect battery performance, life and safety. SoT estimators can also help to prevent overheating and possible damage.

The research on battery SoT estimation is mainly based on thermal models that mimic battery heat generation and heat dissipation. According to the dimension of temperature distribution, it can be divided into lumped-mass thermal models, which are based on ECMs (Equivalent Circuit Models), and 1-D or 2-D thermal models, based on differential equations or PDEs (Partial Derivative Equations) [6].

To optimise the battery operation preventing unexpected failures, state-of-function (SoF) estimator can be used. It describes a battery's instantaneous maximum power output capabilities or, in other terms, it describes if the battery is able to meet the load demands or not [9–12]. The battery needs to perform at the boundaries dictated by the Safety Operating Area (SOA), which is defined by the voltage, current, SoC and temperature boundaries, among others. This SOA window is modified during the cell's lifetime due to aging or environmental conditions, so it needs to be updated during the lifetime [10,13]. Thus, the SoF is strongly correlated to SoC, SoH and internal resistance [159].

Following the safety considerations, battery failure or risky event detection is becoming a critical function, considering that the lifespan of a battery is extended as much as possible to increase its profitability. In 2nd life applications, they are even more critical since the battery has already been degraded and the operation conditions could be limited.

There are several abuse conditions which can produce an Internal Short Circuit (ISC) inside the battery, being this the main cause for battery cells to undergo a Thermal Runaway (TRA) event, as it has been aforementioned. Nevertheless, the TRA process differs depending on many factors, such as the cell chemistry, the aging or the State of Charge (SoC).

TRA is currently the greatest safety concern of a battery cell, and it is defined as the phenomenon of exothermic chain reactions within the battery cell [14]. These reactions cause the internal cell temperature to increase, where the battery materials and inner structures degrade and could potentially become thermally unstable, forcing the failure of the battery eventually.

A TRA event can be induced by several abuse mechanisms, which can be classified into three main groups: mechanical, electrical and thermal abuse.

- **Mechanical abuse**, either through crush or penetration, can lead to electrode material cracks, ductile fractures, and ISCs. Regardless of method, mechanical abuse converts electric energy into thermal, raising the internal temperature and possibly triggering a TRA. Depending on penetration severity, a total discharge without reaching critical temperatures may occur, potentially avoiding a TRA. [15–17]

- **Electrical abuse** in batteries can result from overcharge, over discharge, high C-rates, or external short-circuits. Overcharge-induced abuse is particularly dangerous, causing lithium deposition, dendrite growth, and SEI thickening. Over discharge can lead to SEI layer breakdown, copper collector dissolution, and capacity loss. External short-circuits or high C-rates cause internal temperature rise, gas generation, and potential component decomposition. [18–27]
- **Thermal abuse** in batteries can stem from various sources, including over-temperature due to inadequate cooling, poor electrical contact, or internal heat generation from previous abuse conditions. Depending on the severity, thermal abuse can result in capacity loss, resistance increase, or trigger chain exothermic reactions leading to TRA. Each component of the cell has specific thermal limits, dictating when decomposition or shutdown occurs. Additionally, low temperatures coupled with high C-rates can induce lithium plating, leading to capacity loss, resistance increase, and safety hazards like internal short-circuits. [28–32]

As regards safety functions, currently BMS systems overview and ensure a >I, V, T (5 safety index) protection that cannot be adapted to different applications.

Additionally, they [33,34] require historical data of a battery in its whole life cycle to characterise the safety subfunctions. Furthermore, this indicator is very subjective, as it depends on the analysed internal parameters, the analysed battery sample, the general situation of the Battery System (BS) and the approach used to quantify the SoS (product of subfunctions, probability distributions, etc.).

Those are the most common states that a BMS needs to monitor and manage (SoC, SoH, SoT, SoF, balancing, and safety risks), in some cases, more specific estimators could be necessary.

To address the safety of the battery and to evaluate how much time will pass until the battery is not considered safe anymore, a new indicator called SoS is introduced. It is an incipient indicator and there is not a consensus about its definition or implementation strategy for a real application yet.

This indicator was first introduced in 2016 by Cabrera-Castillo et al. [35]. The authors provided a quantitative method between 0 (totally insecure) and 1 (totally safe) to the safety concept of a battery cell. This method does not evaluate the safety in terms of internal mechanisms, but it is rather solely based on an external perception of the battery.

Another approach is to identify the conditions indicating that a battery cell has a high probability of generating a TRA event in the near future [33]. The conditions that could indicate an abnormal behaviour of a battery, among others, are:

- Self-discharge ratio.
- Internal Short-Circuit (ISC).

- External short-circuit.
- Lithium plating generation.
- Oxygen release of the charged electrodes.
- Expansion force.

The quantitative analysis of these features can require historical data of a battery in his whole life cycle inside a battery system. Furthermore, this indicator is very subjective, as it depends on the analysed internal parameters, the battery itself, the general situation of the battery system and the approach used to quantify the SoS.

Regarding the lifespan of a battery, warranty concepts can be introduced. Warranty refers to the need to repair or replace a battery if it does not perform as expected. The warranty period usually covers a certain duration of time or a specified number of charge cycles, ensuring that the battery meets certain performance standards. But there is also maintenance work that can help to extend the warranty period.

The actual maintenance schedules of electric complex systems (whole ESS systems, EVs, etc.) are mostly done in a mix of maintenance types. The maintenance activities of critical components are done in a preventive or predictive way, such as the case of battery systems, while the rest non-critical components are done in a reactive way.

- Reactive maintenance responds to issues after they occur. [36]
- Preventive maintenance, based on proactive measures, prevents issues through regular inspections. [37]
- Predictive maintenance uses data to predict and prevent potential issues before they happen. [38]

All of them have been widely applied in electromobility applications. But they are not specifically studied for 2nd life use [39]. Therefore, accurate SoC, SoH, and SoT estimation become critical as well as covering safety and warranty aspects.

2.3.2 Battery2Life proposal evaluation

Battery2Life will enhance 2nd life battery reliability and safety through new embedded functionalities, including advanced sensing, active balancing and innovative SoX estimation algorithms. These improvements address specific challenges like non-uniform behaviour, or early warning of safety risks.

- **Pillar 1 – Industrial load levelling, grid-scale application:**

In case of **Pillar 1**, LFP modules will be reconfigured for a grid-scale 2nd life towards enabling a more efficient integration of EVs into the electricity grids. In this solution, EIS will be embedded at module-level. Most of the EIS processing will be done at the edge BMS to reduce computational cost, but the definition/readjustment of the parameters are performed at the cloud BMS. The EIS outcome (ex. real and imaginary part at specific frequency/ies, Nyquist curve) is communicated to the cloud and new parameters are computed.

On the one hand, the EIS will be used to solve the problem of hysteresis in LFP chemistry. The solution will rely on the post-processing of the EIS which is decomposed using Distribution of Relaxation Time (DRT) technique, to update the ECM. This way **SoC** estimation accuracy is increased.

On the other hand, high-frequency EIS phase data is used to estimate cell core temperature (**SoT**) (e.g., for thermal-runaway prediction).

The **SoH** estimation will be a differential-capacity study based on dQ/dV and Coulomb counting.

On the other hand, **SoS** will also be based on EIS for the early detection of li-plating and TRA. Reaching a practical and precise SoS estimation is very complicated, but Battery2Life will improve it, by adding more individual subfunctions that affect the SoS of a BS, i.e., the failure probability of each physical individual element of a battery (electrodes, separators, electrolyte, etc.); and adjusting the already existing subfunctions (operating T, V, I, external deformations and internal impedance).

- **Pillar 2 – Domestic storage application:**

In case of **Pillar 2**, NMC modules will be redesigned to cover both 1st and 2nd life, and they will be validated on a home system testbed.

To improve battery control, the idea is to integrate **advanced sensing**. On the one hand, SoX estimators can be improved, by correlating them with the volume changes on charge and discharge cycles. This helps in mitigating safety risks and enables integration into 2nd life applications. On the other hand, they will allow the creation of a comprehensive database for estimating the remaining lifetime. By defining maximum strain and pressure values, end-of-life scenarios can be identified.

Active cell balancing will be implemented by focusing on SoC-based signals transferring energy from the more charged to the less charged cells. This will improve the battery overall efficiency, as the overall capacity will be the average one and not determined by the weakest cell.

As for the safety aspect, in this case, EIS-based approach is not possible since EIS will not be available. At this point, the approach will be to build a **SoF** which limits the operating current to avoid lithium-plating and TRA. For that, a current mapping will be developed, and it will be reinforced with ISC detection. At the same time, **SoS** will be determined during resting time by the evaluation of dV/dt curves.

A Filter-based methods (EKF, Extended Kalman Filter) is used to estimate the **SoC**. The Ampere-hour integral method has been used to estimate the **SoH**.

- **Pillar 1 & 2:**

Additionally, Battery2Life will introduce an innovative algorithm to estimate the **SoW for both Pillars**. It synthesises quantitative information on warranty fulfilment. This algorithm links degradation and usability, addressing a multi-maintenance methodology scenario. Reactive maintenance is addressed by the "remaining warranty" sub-state, determining consumed and available warranty time. Preventive maintenance is referred as the "remaining health" sub-state, quantifying deviation from prior SoH and proximity to End of Life (EoL). Predictive maintenance is represented as the "remaining useful warranty" sub-state, requiring future SoH and referring to remaining warranty until the EoL threshold, quantifying deviations between predicted and expected Remaining Useful Life (RUL). These concepts will be combined to build a single SoW estimator. [40]

In the next table the Battery2Life proposal is summarised and evaluated.

Table 3: Evaluation of BMS algorithms – Pillar 1.

PILLAR 1					
	Included	**Essential**	**Basic/Advanced**	**Developer**	**Approach**
SoC	Yes	Yes	Advanced	CSEM/EPFL	Model-based (assisted by EIS, DRT and KF)
SoH	Yes	Yes	Basic	CSEM/EPFL	Differential-capacity (dQ/dV and Coulomb counter)
SoT	Yes	No (1)	Advanced	CSEM/EPFL	Based on EIS phase shift
SoF	No (2)	-	-	-	-
SoS	Yes	No (1)	Advanced	CID	EIS-based lithium-plating and TRA early detection
SoW	Yes	No (1)	Advanced	CID	Rule based multi-maintenance
Balancing	Yes	Yes	Basic (3)	CSEM	Passive balancing
SoX (smart sensing)	No	-	-	-	-

(1) When an estimator is defined as non-essential, it refers to the level of basic functionality, meaning that the BMS can perform the most basic operations without them. In any case, they are all important from a safety point of view, and each of the "non-essential" estimators is actually very significant in ensuring reliable and safe operating conditions.

(2) Even if an SoF estimator is not directly going to be integrated in Pillar 1, based on the SoH the safe maximum power will be defined during operation.

(4) Although the balancing in Pillar 1 is not addressed in this project, some basic balancing will be integrated into the final solution.

Co-funded by the European Union

Table 4: Evaluation of BMS algorithms – Pillar 2.

PILLAR 2					
	Included	**Essential**	**Basic/Advanced**		**Approach**
SoC	Yes	Yes	Basic	CID	Filter based (EKF)
SoH	Yes	Yes	Basic	CID	Differential-capacity (ampere-hour integral)
SoT	No (1)	-	-	-	-
SoF	Yes	Yes	Advanced	CID	SoS-based to avoid litium-plating and TR
SoS	Yes	No (2)	Advanced	CID	dV/dt curves to detect li-plating
SoW	Yes	No (2)	Advanced	CID	Rule based multi-maintenance
Balancing	Yes	Yes	Advanced	AIT/MIBA	Active balancing
SoX (smart sensing)	Yes	Yes	Advanced	AIT	High accuracy SoX based on strain and pressure sensors

(1) SoT is not defined in Pillar 2, but temperature related risky events are covered by the SoF and SoS.

(2) When an estimator is defined as non-essential, it refers to the level of basic functionality, meaning that the BMS can perform the most basic operations without them. In any case, they are all important from a safety point of view, and each of the “non-essential” estimators is actually very significant in ensuring reliable and safe operating conditions.

Co-funded by the European Union

3 COMPARISON BETWEEN PILLARS

In this section you can find a summary table where both Battery2Life solutions are compares in terms of HW, SW and algorithms.

Table 5: Evaluation of BMS algorithms.

HW topology					
PILLAR 1			PILLAR 2		
	Developer	Approach		Developer	Approach
Wireless master-slave BMS	CSEM/SLG	Modular	**Wireless BMS**	MIBA/AIT	Modular
EIS HW	CSEM	Module level EIS integrated	**Smart Sensing**	AIT	Advanced pressure and strain sensors
Cloud communication	ICCS	ESP32 module	**Cloud communication**	ICCS	CAN interface
SW architecture					
PILLAR 1			PILLAR 2		
	Developer	Approach		Developer	Approach
Language	CSEM/EPFL	C	**Language**	AIT	C
RTOS	CSEM/EPFL	FreeRTOS	**RTOS**	AIT	FreeRTOS
interface	CSEM/EPFL	CANbus	**interface**	AIT	Modbus
Advanced algorithms					
PILLAR 1			PILLAR 2		
	Developer	Approach		Developer	Approach
SoC	CSEM/EPFL	Model-based and ECM update by EIS	**SoC**	CID	Filter based (EKF)
SoH	CSEM/EPFL	Differential-capacity (dQ/dV and Coulomb counter)	**SoH**	CID	Differential-capacity (ampere-hour integral)
SoT	CSEM/EPFL	Based on EIS phase shift	**SoT**	-	-
SoF	-	Safe Maximum Power defined based on SoH	**SoF**	CID	SoS-based to avoid li-plating and TR
SoS	CID	EIS-based li-plating and TRA early detection	**SoS**	CID	dV/dt curves to detect lithium-plating
SoW	CID	Rule based multi-maintenance	**SoW**	CID	Rule based multi-maintenance
Balancing	CSEM	Passive balancing	**Balancing**	MIBA/AIT	Active balancing
SoX (smart sensing)	-	-	**SoX (smart sensing)**	AIT	High accuracy SoX based on strain and pressure sensors

Co-funded by the European Union

4 PILLAR 1: INDUSTRIAL LOAD LEVELLING BMS SPECIFICATIONS

In the next section the selected Pillar 1 HW topology, SW architecture and included algorithm specifications are presented.

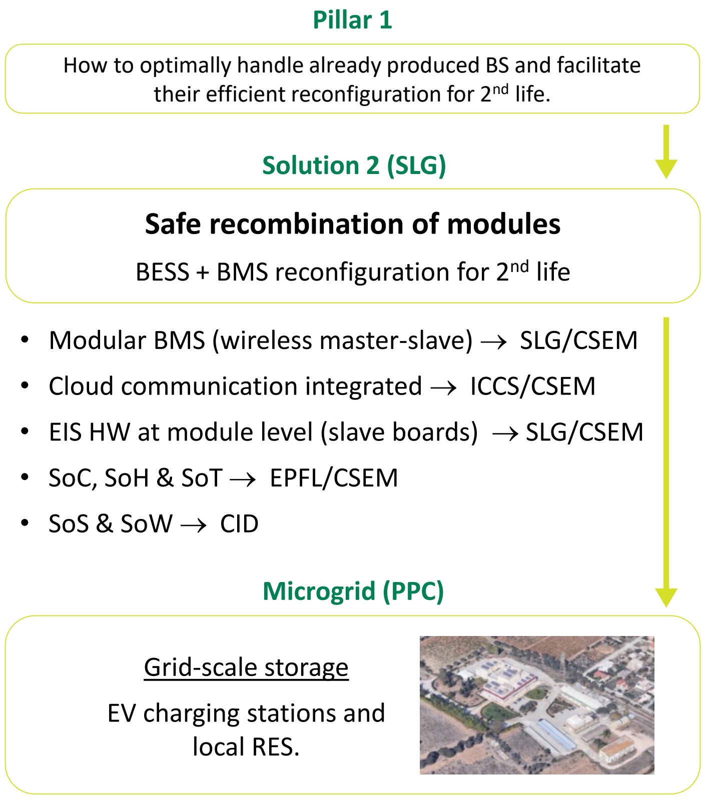


Figure 8: Battery2Life Pillar 1.

4.1 Selected HW topology specifications

After a thorough discussion between the partners of the Battery2Life project, it was decided that the most optimal technical and economical solution for the BMS topology in the Pillar 1 would be the modular topology. The key concepts in the chosen BMS topology are:

1) A slave board (integrated with EIS) will control 12-16 cells (module) and all of the slaves will be controlled by a master board.
2) The communication between the slave and the master will be performed by a wireless communication method due to the parasitic inductance the cables create in the EIS measurements.

Co-funded by the European Union

Other considerations about the BMS topology of the Pillar 1 are that:

1) the dimensions of the cell and the modules were given,
2) the cell balancing will be passive,
3) no flexible PCBs (slave) will be used, and
4) an early consideration was put on the table for the electromagnetic inference and compatibility of the BMS due to the inverter and solutions were proposed.

Table 6: BMS HW topology of Pillar 1.

Pillar 1 HW	
Topology	- Modular topology of the BMS system. - 1 slave in every 12-16 cells (module)
Central Processor	STMicroelectronics
Functional Safety	It will be defined in T2.1
Cell and Module dimensions	**Module:** Two alternatives will be considered: 1. Module with 2P8S 2. Module with 1P16S In both of them the dimensions are the same: Length: 536 mm Total length: 545 mm Width: 157 mm Height: 290 mm **Cell**: CALB 72 Ah Width (Cell thickness): 30mm Height: 215. 8 mm Total height (terminals added): 220.8 mm Distance between the terminals: 67.5 mm Length dimension: 35 mm
Balancing	Passive
Communications	- Master – Slave → Wireless - Other communications → Canbus (e.g. inverter)
PCB Flexibility	No
Electromagnetic Interference and compatibility	Due to the inverter. Considerations for placing the inverter outside the battery. Otherwise, a protective shielding will be needed.

- **Integrated EIS**

EIS is a powerful, fast, non-invasive technique used for impedance analysis in electro-chemical systems at multiple frequencies. Each electrochemical process in a lithium-ion battery has its own time-constant, hence, they can be characterised over a range of frequencies. For lithium-ion batteries, EIS is mainly used for cell electric and electrochemical characterisation and modelling purposes. In Pillar 1, it will be used for the ECM parameter estimation and continuous update, SoT estimation and SoS estimation. That is why EIS will be integrated at module level (slave boards).

The whole module will be excited with a controlled current, and the voltage of each cell will be sensed, along with the excitation current. The EIS will be computed directly on each slave board, and the results will be sent to the master board using a self-developed wireless communication protocol.

- **Integrated wireless cloud communication**

The master board will be the one asking the slave board to perform EIS. The wireless communication from the master board to the cloud will be handled using an ESP32 module.

Co-funded by the European Union

4.2 SW architecture specifications

The proposed SW architecture for the Pillar 1 is divided in three macro blocks: **BMS Master, BMS Slaves and a Cloud Server.**

The most fundamental design principles of the BMS architecture are clear interface definitions and a separation between HW dependent basic software (firmware, FW) and HW independent application software (SW), i.e. an AUTOSAR architecture. On multiple levels the BMS is conceived as a set of blocks that communicate with each other via clearly defined input and output data structures, also referred to as buses.

The cloud and the edge BMS communicate with one another via REST APIs (Application Programming Interfaces). The data structures of each endpoint are fixed, while the algorithms that take these data structures as input are interchangeable. On a hardware level, the communication between the BMS master and the cloud happens wirelessly.

At the BMS Master level interoperability and the combination of open-source and IP-restricted blocks is achieved by adopting an AUTOSAR design philosophy. This means that there is a clear distinction between application software and basic software.

The application software is a set of high-level BMS routines (state estimation, safety routines, system diagnostics, etc.). Each of these routines has a fixed input and output data structure. These fixed input and output buses allow for the routine itself to be interchangeable and/or treated as a black box (in case of IP protected algorithms). Furthermore, any HW-dependent aspects such as communication, data storage and sensor treatment are delegated to the basic SW layer, making the application SW layer completely HW-independent. The routines of the application SW are coordinated by a master BMS.

The basic software is any software in charge of data storage management, the operating system, the communication protocols, input/output and bootloader. Those blocks are hardware dependent on the input side, but their output is clearly defined by input buses of the application SW. Hence to adapt the BMS to varying hardware components, only the application SW needs to change.

Since the BMS architecture consists of a cloud and an edge BMS that communicate with each other wirelessly, this also allows for interoperability after deployment. Just as algorithms can exchange data via clearly defined REST API endpoints, new application software or basic SW can be deployed via FW-over-the-air updates. This is achieved through a hardware-specific bootloader as part of the basic SW.

Co-funded by the European Union

4.3 Advanced BMS algorithms specifications

In the next section, the specifications associated to the definition of the estimators and new functionalities are presented. Figure 9 shows the estimator and functionalities covered in Pillar 1.

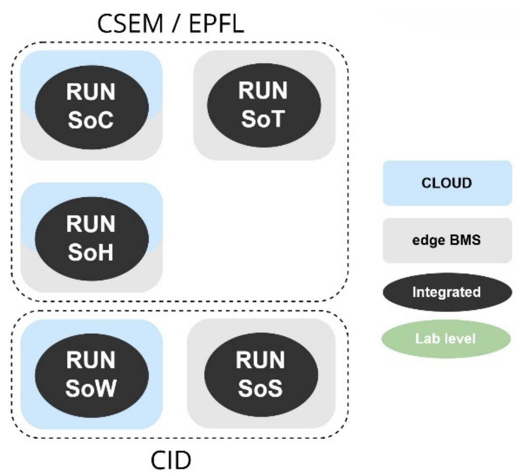


Figure 9: Advanced BMS algorithms in Pillar 1.

The specifications are presented in three blocks: functionality specification, input specifications and output specifications. This way the high-level working principle, input/output variables, execution frequency, necessary memory and other specifications are presented.

4.3.1 EIS-based SoC

EIS measurement integrated on the HW of the Pillar 1 will improve the estimation of the SoX estimators. Additionally, the EIS will be used to solve the problem of hysteresis in LFP chemistry.

- **Functionality specifications**

The SoC estimator runs continuously on the Master BMS, calculated using the model parameters and measurements. Additionally, model parameters will be recalculated regularly by getting EIS data from the cells, running the DRT algorithm on the cloud and updating the parameters to the Master BMS.

Functionality specifications
High level working explanation: SoC prediction based on ECM model (assisted by EIS and DRT analysis) of the battery and then corrected by a Kalman filter.
Methodology: A combination of an ECM model (EIS-based approach) and Kalman Filter.

Integration: The SoC estimator runs on the Master BMS, providing cell level SoC. The parameter updates run on the Cloud.
Execution frequency: SoC will be updated continuously every 1 second (1 Hz), in any operation condition. A recalculation of the ECM parameters will be done at regular intervals (to be determined, but for example every month).
Execution time: < 1 second
Memory usage: 50-65 percent flash memory usage on STM32H7 chip for 90 to 100 cells.
Algorithm type: C-code in a black-box.
Fidelity: High.
Innovative: Innovative, it is none of the classic SoC estimators presented in the SoA.

- **Inputs specifications**

Apart from the cell level measurements for the SoC estimation, EIS is also needed for the ECM parameters adequation.

Inputs specifications
Number of inputs: 2 per cell and 1 at battery pack level.
Input variables: Cell voltage measurement and battery pack current measurement for SoC estimation and cell EIS measurements for ECM parameters adequation.
Data type: Decimal numeric variables.
Data acquisition: For real-time SoC estimation at 1 Hz, cell voltage and battery pack current measurements need to be acquired at less than 1 Hz. Parameter fit via EIS will be done weekly or monthly at max (full EIS takes about 10 minutes).
Accuracy: High – however, input data shall be checked for outliers.
Data storage: Cell/module level voltage and current measurements are stored locally until they are processed/updated for SoC estimation. EIS measurements are directly transmitted to the cloud and stored there. Model parameters are calculated on the cloud, transmitted to the Master BMS and stored locally.

- **Outputs specifications**

SoC value will be the main output of the estimator, but EIS will also be sent to the cloud.

Outputs specifications
Number of outputs: 1
Output variables: SoC value
Data type: Decimal numeric variables.
Data delivery: SoC will be updated every 1 second (1 Hz) and transmitted to the master state machine. EIS is transmitted to the cloud at a much lower frequency and stored there for the later parameters update.
Accuracy: N/A.
Data storage: SoC value is stored locally, and EIS measurements are stored in the cloud.

4.3.2 SoH

Although the SoH estimation is not the main focus of the project, it is necessary for every system, particularly for the SoW estimation.

- **Functionality specifications**

The SoH estimation will be a differential-capacity study which is based on dQ/dV and Coulomb counter. Even if, necessary measurements will be collected locally, the estimation will be carried out on the cloud periodically.

Functionality specifications
High level working explanation: Differential-capacity study to estimate SoH.
Methodology: Based on dQ/dV and Coulomb counter.
Integration: Cell/module level SoH estimation will be partially run in the Master BMS and the BMS cloud. The measurements are collected locally and transmitted to the cloud. The SoH algorithm runs on the cloud.
Execution frequency: The SoH will be updated weekly or monthly.

Execution time: When data are available on cloud database, the calculation takes few seconds and then it is sent to the Master BMS.
Memory usage: SoH calculation will be performed in the cloud BMS.
Algorithm type: C-code in a black-box.
Fidelity: State of the art accuracy will be ensured.
Innovative: Not innovative.

- **Inputs specifications**

Cell level measurements will be the only input. Even if SoH is updated periodically, measurements will be continuously transmitted to the cloud and stored for cumulative data analysis.

Inputs specifications
Number of inputs: 1 per cell and 1 battery pack current.
Input variables: Cell voltage and battery pack current.
Data type: Decimal numeric variables.
Data acquisition: Data logging during battery utilisation will be uploaded to the cloud and then data analysis and SoH calculation done in the cloud.
Accuracy: High – however, input data shall be checked for outliers.
Data storage: Data will be transmitted to cloud and stored there.

- **Outputs specifications**

SoH will be updated periodically.

Outputs specifications
Number of outputs: 1 per cell and module/pack SoH.
Output variables: Cell/module/pack level SoH.
Data type: Decimal numeric variables.

Data delivery: Updated weekly/monthly SoH is delivered for SoW calculation and/or historical record.
Accuracy: N/A.
Data storage: All historical SoH values are stored in the cloud.

4.3.3 EIS-based SoT

EIS will also be used to estimate cell temperature (SoT), which can help on the thermal-runaway prediction.

- **Functionality specifications**

In this case the estimator will focus on the EIS measurement, since the impedance phase-shift can indicate the thermal behaviour.

Functionality specifications
High level working explanation: Internal temperature is estimated from high-frequency phase-shift of EIS.
Methodology: EIS-based approach.
Integration: This estimator run on the Master BMS and provides cell-level SoT
Execution frequency: SoT will be updated continuously every 100 ms (10 Hz), in any operation condition.
Execution time: < 100 ms
Memory usage: Low compared to SoC estimator (simple linear operation).
Algorithm type: C-code in a black-box.
Fidelity: High.
Innovative: Innovative, it is none of the classic SoT estimators presented in the SoA.

- **Inputs specifications**

As input it will only have the EIS phase data.

Inputs specifications
Number of inputs: 1 per cell.
Input variables: High-frequency EIS phase data from cells.
Data type: Decimal numeric variables.
Data acquisition: EIS measurements are acquired from every slave BMS every 100 ms.
Accuracy: High – however, input data shall be checked for outliers.
Data storage: EIS measurements are stored locally until they are processed/updated for SoT estimation.

- **Outputs specifications**

The only output will be the SoT estimator of each cell.

Outputs specifications
Number of outputs: 1 per cell.
Output variables: Cell level temperature.
Data type: Decimal numeric variables.
Data delivery: SoT will be updated every 100 ms (10 Hz), and transmitted to safety algorithms that are in the Master BMS.
Accuracy: N/A.
Data storage: SoT estimator is stored locally until used by safety algorithms. It can also be transmitted to cloud for long-term storage.

4.3.4 EIS-based SoS

EIS-based SoS estimator is directed to the Pillar 1, because EIS measurement will be available. EIS is a powerful, fast, non-invasive technique used for impedance analysis. It can be used to detect internal status of the battery such as internal temperature estimation or aging detection. In this case, EIS measurement will be used for the SoS estimation.

- **Functionality specifications**

EIS-based SoS will be implemented at cell level in any operation conditions.

Functionality specifications
High level working explanation: Estimates the safety level of the battery and gives a warning if an imminent TRA event is taking place.
Methodology: EIS-based. With additional voltage, current, temperature measurements.
Integration: Algorithm implemented in the edge BMS at cell level. Each cell shall have its own SoS indicator. For that, the algorithm will run on the Master BMS with cell level input data.
Execution frequency: Voltage, current and temperature are monitored continuously. While SoT is below a predefined threshold (to be defined during the project) SoS will be updated every 5-10 minutes, so as the EIS measurements. The same procedure will be followed in resting periods. When critical, EIS measurements will be performed every 1 minute if possible. SoS indicator will be updated every minute as well.
Execution time: < 1 ms
Memory usage: At least, the last 3 EIS measurements (magnitude and phase) per cell shall be stored in "single" or "int" type variables. So: **2 bytes * 2 variables * 3 measurements * Number of cells** The last voltage, current and temperature data measurements memory usage are not considered within this algorithm as they are used for monitoring the cells. For the voltage, the voltage slope will be calculated in each step, in "single" or "int" type variables, so the memory usage for this case will be: **2 bytes/cell** Then, each cell will have its own SoS indicator, in "single" or "int" type variable: **2 bytes/cell**
Algorithm type: MATLAB generated C-code as black box for the specified microcontroller.
Fidelity: N/A
Innovative: Innovative, there is nothing yet in the SoA.

- **Inputs specifications**

Apart from EIS measurements, voltage, current and temperature will be also considered for the SoS estimation at cell level.

Inputs specifications (at cell level)
Number of inputs: 6.
Input variables: Cell voltage, current, temperature, EIS magnitude, EIS phase, and SoT.

Data type: “single” or “int” type variables. **2 bytes per variable and per cell.**
Data acquisition: Cell voltage, current and temperature are acquired constantly. EIS data shall be acquired at most every 1 minute, and at least every 10 minutes.
Accuracy: N/A. However, input data shall be checked for outlier data.
Data storage: Necessary. The following data shall be stored in “single” or “int” type variables (**2 bytes per variable and per cell**): 1) last computed dV/dt value, 2) last EIS magnitude and phase values, and 3) a reference EIS phase value.

- **Outputs specifications**

As output, we will have the SoS indicator.

Outputs specifications (at cell level)
Number of outputs: 1.
Output variables: SoS value.
Data type: “single” or “int” type variable. **2 bytes per cell.**
Data delivery: periodical output. Updated every 5-10 min in normal conditions if there is no TRA risk, and every 1 minute if critical.
Accuracy: N/A.
Data storage: Not necessary.

4.3.5 SoW

The SoW is an off-board state that determines the warranty fulfilment level. This state provides a global understandable insight of the warranty state of an evaluated component while providing detailed and complex information about it if required.

- **Functionality specifications**

This algorithm links degradation and usability, addressing a multi-maintenance methodology scenario.

Functionality specifications
High level working explanation: The understandable insight is given by qualitatively determining the fulfilment level of the given warranty in a user-friendly colour code. The detailed and complex information is given by quantifying in three substates related to the most relevant features of each of the different types of maintenance activities: • The Remaining Warranty (RW) for the reactive maintenance. • The Remaining Health (RH) for the preventive maintenance. • The Remaining Useful Warranty (RUW) for the predictive maintenance. The qualitative value that defines the SoW is calculated with an expert system. This calculus is performed off-board. The rule-based logic applied by the expert system determines the SoW value based on the previously determined quantitative substates: the remaining warranty, the remaining health and the remaining useful warranty. That rule-based logic is disclosed in Figure 10.
Methodology: Multi-maintenance methodology.
Integration: The SoW estimation is implemented in the cloud at cell level. Each cell shall have its own SoW indicator.
Execution frequency: it will be executed with a periodicity of a week with an offset of 1 month (a minimum of historic data is required to run the estimator).
Execution time: Increases linearly with cumulatively stored SoH values. It takes 1s for a cell estimation with a SoH vector with 10 values.
Memory usage: The required memory space for the database/historic for 6 months would be around 100 MB. The RAM memory for the execution of the estimator should not exceed 200MB.
Algorithm type: Python. The SoW estimation algorithm will be provided as a black-box fmu.
Fidelity: The fidelity level of this state depends on the level this estimate is done. Therefore, if cell level data would be available, it could be labelled as high fidelity estimation, while at system level low fidelity.
Innovative: This state is an innovative state that provides great insight about the health and warranty evolution of the battery. It provides the metrics needed for a predictive maintenance activity of the battery system and for robust decision-making exercises.

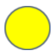
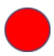
SOW	*RW*	*RH*	*RUW*	*Severity Description*
	*>0.05*	*>0.75*	*>0.5*	*The warranty fulfilment level is correct.*
	*<0.05*	*>0.75*	*>0.5*	*The warranty fulfilment level is correct, but the end of warranty is close.*
	-	*>0.75*	*0.5> RUW >0.4*	***ATTENTION****! Predicted 1 additional replacement. Early advice.*
	-	*0.75>RH>0.5*	*0.5> RUW >0.4*	***ATTENTION****! Predicted 1 additional replacement. Middle advice.*
	-	*0.75>RH>0.5*	*0.4> RUW >0*	***ATTENTION****! Predicted 1 additional replacement. Late advice.*
	-	*0.5>RH>0*	*0.4> RUW >0*	***DANGER****! Predicted 1 additional replacement. Irreversible damages.*
	-	-	*0*	***DANGER****! Predicted 2 additional replacement.*
	-	*0*	-	***DEATH****! The battery has reached the EOL.*
	*¿?*	*¿?*	*¿?*	***Undefined scenario.*** *Something unexpected is happening.*

Figure 10: Qualitative states of SoW.

- **Input parameters specifications**

In case of the SoW the inputs would be a set of parameters which will be previously uploaded to the cloud by the diagnostic-sizing tool. These parameters will not be continuously updated. Initial values will be set at the beginning and will only be updated if necessary.

Input parameters specifications
Number of inputs: 25.
Parameters: Listed below.
Data type: Doubles, being some simple doubles and others vector of doubles. Doubles: - **SoC max.** It is the maximum SoC value at which the operation is limited. - **SoC min.** It is the minimum SoC value at which the operation is limited. - **FECs per day.** It is the amount of Full Equivalent Cycles done per day. - **Days per year.** It is the amount of days of operation expected yearly. - **SoH at EoL.** The defined SoH value as the EoL criterion. - **Enom.** Nominal energy of the pack.

- **Icha max.** Maximum charge current based on datasheet.
- **Icha std.** The standard charge rate used to charge the battery system.
- **Idch max.** The maximum peak discharge current.
- **Idch mean.** The most likely mean discharge current value in operation.
- **lifespan in years.** The lifespan of the battery system in terms of time.
- **lifespan in FECs.** The lifespan of the battery system in terms of discharged energy.
- **number of packs.** The number of packs that the project has.
- **number of modules.** The number of modules that a pack has.
- **number of cells.** The number of cells that a module has.
- **ID of OEM.** A number that defines the OEM.
- **ID of project.** A number that defines the project for the OEM.
- **ID of system.** A number that defines the battery system (in case there are more than a battery system in a project).
- **T max.** Maximum temperature the battery will work in operation.
- **T avr.** The average temperature the battery will work in operation.
- **T min.** Minimum temperature the battery will work in operation.

Vector (doubles):

- **xh0_time [5x1].** The parameters of the aging model that relates the SoH and the time.
- **xh0_FEC [5x1].** The parameters of the aging model that relates the SoH and the FECs.
- **SoH evolution** [40x1]. The SoH values obtained from the aging model to the FECs and time evolution.
- **FECs evolution [40x1].** The FEC values that correlates with the obtained SoH values from the aging model.
- **time evolution [40x1].** The time in years that correlates with the obtained SoH values from the aging model.

Data acquisition: Through the database, which should have stored beforehand using the outcome from the datasheet and sizing tool.

Accuracy: N/A.

Data storage: N/A. Just the parameters at the beginning. Once stored them, they will not change.

- **Inputs specifications**

Usage and degradation related variables and/or parameters will be the inputs for the SoW.

Inputs specifications
Number of inputs: 3.
Input variables: SoH estimation, cumulative discharge capacity and timestamp.

Data type: Vectors of cell number x 1000. SoH and capacity are double. The timestamp is a datetime variable.
Data acquisition: Through the database, which should have stored the data through the cloud gateaway.
Accuracy: The SoH estimation should be +/- 3 % as worst. The cumulative discharge capacity should be +/-2 Ah as worst. The timestamp should be with +/-3 days as worst.
Data storage: Necessary. To Be Determined.

- **Outputs specifications**

Apart from the SoW, each maintenance substates will be also defined on a weekly basis.

Outputs specifications
Number of outputs: 4.
Output variables: SoW, RW, RH, RUW.
Data type: Matrix of strings (SoW) and doubles (RW, RH, RUW).
Data delivery: Weekly.
Accuracy: RW depends on the Timestamp accuracy. RH depends on the SoH accuracy. The RUW depends on the cumulative SoH accuracy. The SoW N/A.
Data storage: Not necessary.

Co-funded by the European Union

5 PILLAR 2: DOMESTIC STORAGE BMS SPECIFICATIONS

In the next section the selected Pillar 2 HW topology, SW architecture and included algorithms' specifications are presented.

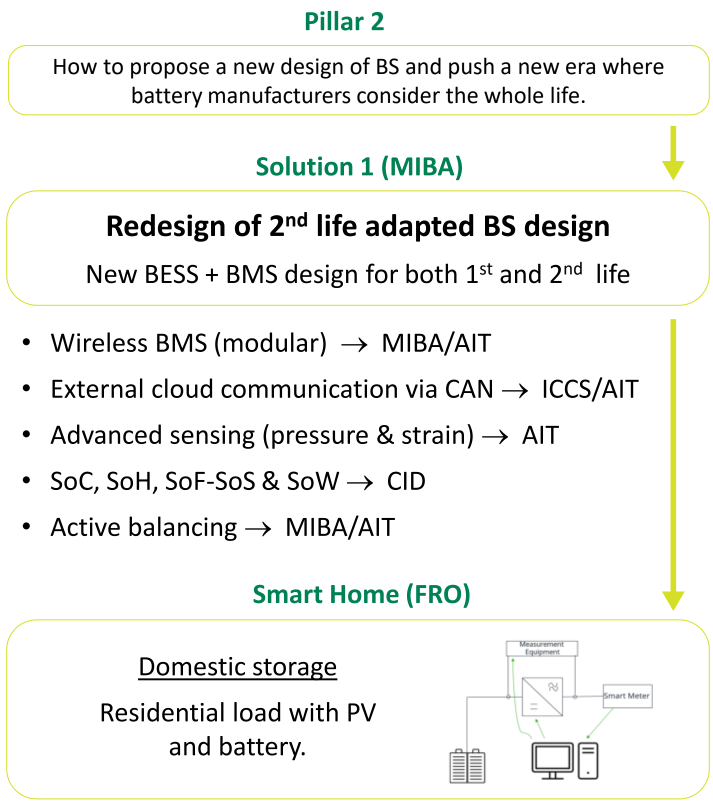


Figure 11: Battery2Life Pillar 2.

5.1 Selected HW topology specifications

Selected BMS HW topology based on optimal technical and economical solution. The hardware consists of the following main components:

- **Module management system (MMS):** The main purpose of a MMS is to measure the voltage of a few cells (12 cells in case of pillar 2) as well as the temperature within a module. Furthermore, it receives balancing commands from the master to equalise the SoC of each cell.
- **The BMS master controller** handles the computation (e.g. thermal management, balancing, diagnosis, estimators, ...) and the MMS communication as well as the communication with the battery junction box and others.

- **The battery junction box** includes necessary fuses and relays and a precharging resistor to disconnect or connect the battery to the load. Furthermore, it includes a current sensor to measure the pack current.
- **Wireless communication gateway:** To communicate with the cloud a gateway is necessary which receives the CAN BMS data and send it to the cloud.

With an additional RS-485 interface the communication with the inverter will be handled by the master BMS.

Table 7: BMS HW topology of Pillar 2.

Pillar 2 HW	
Topology	Modular topology of the BMS system. (Master – Slave)
Central Processor	Texas Instruments
Functional Safety	It will be defined in T2.1
Cell and Module dimensions	18 mm x 650 mm (cylindrical cells)
Balancing	Active
Communications	- CANbus - RF 2.4 GHz (Wireless) - RS-485 (Inverter)
	- TRA → Mechanical strain censors - Integrated circuit → Analog Devices Inc.
PCB Flexibility	No

For a successful transition from 1st life to 2nd life combability needs to get addressed to minimise the effort of reassembly and the quantity of components to get exchanged. To address the first life there is a need to have the following components in common for the cell monitoring unit: the connector for the cell- and temperature measurement harness, the mounting points and hole diameter, the space. Since the 2nd life application will have active balancing, there size of the PCB will increase to the borders of the installation base. For the BMS in general the need to have the same low-voltage range and the necessary interfaces to communicate with a higher-level system.

- **Wireless BMS**

A first preselection of the BMS and MMS main components is already made. The wireless Module Management System (MMS) consists of several ICs:

Monitoring Integrated Circuit (IC) (LTC6811): 12 cell battery stack monitor with Serial Peripheral Interface (SPI) communication interface.

Wireless MCU (CC2652R7): Wireless microcontroller with an Arm cortex processor and an additional RF-core which handles the radio communication.

Balancer IC (ETA 3000): Inductive balancer IC which can balance two cells with a programmable current up to 2 A.

The master BMS contains a Hercules microcontroller from Texas Instruments (TMS570 series) which is suitable for automotive or industrial safety applications.

- **Advances/Smart sensing**

With the help of pressure and strain sensors the volume change of Lithium-Ion Batteries (LIBs) can be measured. These sensors are applied on the surface of cylindrical cells. Due to the volume changes of the active materials, also volumetric changes of the cylindrical housing can be observed. This correlation can be used to estimate the SoC of LIB. Therefore, one goal is to develop a SoX estimator which uses the data of the applied strain/pressure sensors to calculate the SoC of the cell.

- **Active balancing necessary HW**

Despite the most common balancing strategy is passive, there are options to implement active balancing. Therefore, some extra circuit is needed between the cells. To achieve this there will be the use of an IC, which behaves like a DC/DC using inductance to forward energy. Also, a possibility to enable and disable the circuit in a galvanically isolated way is needed.

- **External cloud communication**

AIT will provide a CAN interface to send and receive messages to a gateway.

5.2 SW architecture specifications

The objective is to conceptualise a reliable, interoperable, safe, and scalable BMS SW architecture which can host open-source blocks and IP-restricted blocks without limiting BMS operation.

Figure 3 shows the software structure of the BMS for Pillar 2. The main BMS contains two microcontrollers which are connected via serial bus to check the status of each. In case of an error, the still functional controller can initiate possible error corrections and prevent the system from serious damage.

- The main purpose of the Hercules TMS570 controller (MCU1) is to perform cell monitoring and the estimation of different SoX algorithms to ensure that the SOA of the battery is always satisfied.
- The second controller (MCU2) handles the radio communication with all the MMS slaves. It also serves as a backup controller in case MCU1 has a malfunction.

The BMS applications will be executed on a real time operating system (FreeRTOS). The driver for the Hardware Abstraction Layer (HAL) will partially be created with the help of the auto-code generation tool called HALCoGen. Different driver will be written to communicate with the inverter and the cloud gateway.

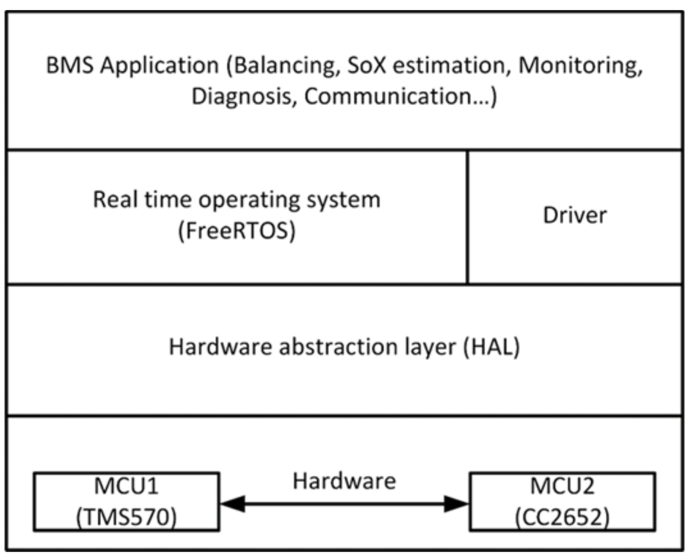


Figure 12: Software structure for BMS in Pillar 2.

5.3 Advanced BMS algorithms specifications

In the next section the specifications associated to the definition of the estimators and new functionalities are presented. In Figure 13 the estimator and functionalities covered by Pillar 2 are shown.

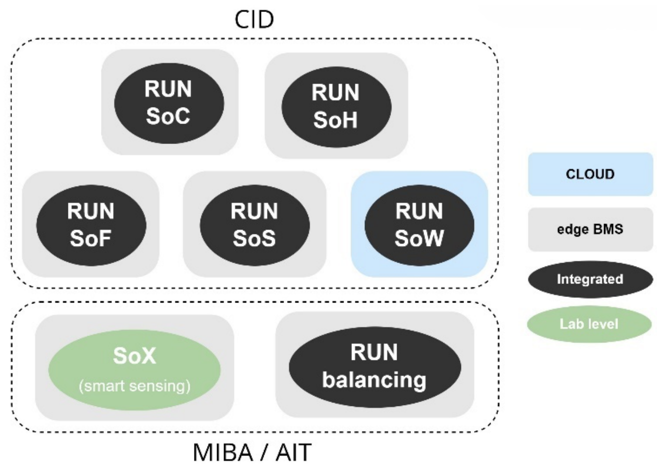


Figure 13: Advanced BMS algorithms in Pillar 2.

As in Pillar 1, the specifications are presented in three blocks: functionality specification, input specifications and output specifications. This way the high-level working principle, input/output variables, execution frequency, necessary memory and other specifications are presented.

5.3.1 SoC and SoH

The SoC and SoH are estimated as the BMS cannot work without the SoC estimation, and the SoW requires to have the SoH estimation.

- **Functionality specifications**

These are not advance estimations but are required. Existing methodologies will be applied such as the Kalman filter approach for the continuous estimation of the SoC and ampere-hour integral for the periodical update of the SoH.

Functionality specifications
High level working explanation: The SoC is estimated through an EKF where the cell voltage measurement is used to adjust the errors committed in the SoC estimation with Ah counting due to current measurement error, initial SoC statement error, SoH estimation error, etc. The SoH is estimated based on the difference between the theoretical capacity and the actual capacity charged at constant current in the evaluated SoC range given by the SoC estimator.

Methodology: Kalman filter for the SoC and ampere-hour integral for the SoH.
Integration: The SoC and SoH estimators are embedded in the edge.
Execution frequency: The SoC is executed each 100ms-300ms. The SoH is executed at the SoC frequency, but the result is obtained when the conditions are meet (from daily to weekly frequence).
Execution time: The execution per cell is below 1ms.
Memory usage: "single" or "int" type variables. - Cell Model: **2 bytes *751 variables = 1502 bytes.** - SoC algorithm: **2 bytes * 59 variables/cell * Number of cells = 118 bytes/cell.**
Algorithm type: c. Both SoC and SoH estimations are provided as c ".a" library generated from Matlab.
Fidelity: Average error of +/-3%.
Innovative: These estimators are not innovative.

- **Inputs specifications**

The input for the SoC estimation will be cell level measurements. While the SoH will be estimated based also on current measurements (ampere-hour integral) and SoC estimation (SoC range).

Inputs specifications (at cell level)
Number of inputs: 3.
Input variables: Voltage, current and temperature.
Data type: Decimal numeric.
Data acquisition: Constantly.
Accuracy: N/A. However, input data shall be checked for outlier data.
Data storage: Necessary. Data should be accumulated for periodical SoH update.

- **Outputs specifications**

The output will be the estimation of both parameters, SoC and SoH. SoC will be constantly updated while SoH will be updated periodically.

Outputs specifications
Number of outputs: 2.
Output variables: SoC and SoH.
Data type: Vector of doubles. Vector length depending on number of cells.
Data delivery: SoC each 100-300ms and the SoH daily-weekly (when conditions meet).
Accuracy: +/-3%.
Data storage: Not necessary.

5.3.2 SoS-based SoF

To address the SoS in the Pillar 2, as we do not have EIS available a different approach will be pursued. In this case, a current limiting SoF will be developed, based on the SoS, which will avoid lithium-plating and TRA conditions during charge. During resting period, dV/dt curves will be calculated to detect anomalies and update the SoS.

- **Functionality specifications**

The SoF will be implemented as a look-up table that limits the charging current based on cell conditions, applicable at any time.

Functionality specifications
High level working explanation: Limits the current of the battery based on the probability of generating Lithium-plating with the current working conditions of the battery. Additionally, during resting period lithium-plating will be also detected.
Methodology: The SoF will be a look-up table. This look-up table will contain the maximum cell charging current value for each SoC and temperature value. Additionally, dV/dt curves will be extracted during resting period for the lithium-plating detection. Necessary measurements: voltage, current, and temperature.
Integration: Algorithm implemented in the edge BMS at cell level. Each cell shall have its own SoS indicator.
Execution frequency: Voltage, current and temperature are monitored continuously. During charging, this algorithm will be updated every 1 second.
Execution time: < 1 ms

Co-funded by the European Union

Memory usage: The last voltage, current and temperature data measurements memory usage are not considered within this algorithm as they are used for monitoring the cells. For the voltage, the voltage slope will be calculated in each step, in “single” or “int” type variables, so the memory usage for this case will be: **2 bytes/cell** The last SoC data is needed. The memory usage for the SoC value of each cell is not considered within this algorithm as it is used for other purposes as well. Then, each cell will have its own SoS indicator, in “single” or “int” type variable: **2 bytes/cell**
Algorithm type: MATLAB generated C-code as black box for the specified microcontroller. C-code or .a library available upon request.
Fidelity: N/A**.**
Innovative: Innovative, there is nothing yet in the SoA.

- **Inputs specifications**

The inputs will be cell level measurements.

Inputs specifications (at cell level)
Number of inputs: 4.
Input variables: Cell voltage, current, temperature, SoC.
Data type: “single” or “int” type variables. **2 bytes per variable and per cell.**
Data acquisition: Cell voltage, current, temperature and SoC are acquired constantly. A minimum of 1 second update is needed.
Accuracy: N/A. However, input data shall be checked for outlier data.
Data storage: Necessary. The following data shall be stored in “single” or “int” type variables (**2 bytes per variable and per cell**): 3 last computed dV/dt value.

- **Outputs specifications**

As output there will be two elements, which are the limited current considering operation conditions and the SoS indicator.

Outputs specifications (at cell level)
Number of outputs: 2.

Output variables: SoS value, and limit charging current.

Data type: "single" or "int" type variable. **2 bytes per variable and per cell.**

Data delivery: during charging, the limit charging current is output every 1 second. For the SoS value, it will be updated at the end of the charging process (charging + 2 hour relaxation).

Accuracy: N/A.

Data storage: Not necessary.

5.3.3 SoW

The SoW is an off-board state that determines the warranty fulfilment level. This state provides a global understandable insight of the warranty state of an evaluated component while providing detailed and complex information about it if required.

- **Functionality specifications**

This algorithm links degradation and usability, addressing a multi-maintenance methodology scenario. These parameters will not be continuously updated. Initial values will be set at the beginning and will only be updated if necessary.

Functionality specifications
High level working explanation: The understandable insight is given by qualitatively determining the fulfilment level of the given warranty in a user-friendly colour code. The detailed and complex information is given by quantifying in three substates related to the most relevant features of each of the different types of maintenance activities: • The Remaining Warranty (RW) for the reactive maintenance. • The Remaining Health (RH) for the preventive maintenance. • The Remaining Useful Warranty (RUW) for the predictive maintenance. The qualitative value that defines the SoW is calculated with an expert system. This calculus is performed off-board. The rule-based logic applied by the expert system determines the SoW value based on the previously determined quantitative substates: the RW, the RH and the RUW. That rule-based logic is disclosed in Figure 14.
Methodology: Multi-maintenance methodology.
Integration: The SoW estimation is implemented in the cloud at cell level. Each cell shall have its own SoW indicator.

Execution frequency: it will be executed with a periodicity of a week with an offset of 1 month (a minimum of historic data is required to run the estimator).

Execution time: Increases linearly with cumulatively stored SoH values. It takes 1s for a cell estimation with a SoH vector with 10 values.

Memory usage: The required memory space for the database/historic for 6 months would be around 100 MB. The RAM memory for the execution of the estimator should not exceed 200MB.

Algorithm type: Python. The SoW estimation algorithm will be provided as a black-box fmu.

Fidelity: The fidelity level of this state depends on the level this estimate is done. Therefore, if cell level data would be available, it could be labelled as high fidelity estimation, while at system level low fidelity.

Innovative: This state is an innovative state that provides great insight about the health and warranty evolution of the battery. It provides the metrics needed for a predictive maintenance activity of the battery system and for robust decision-making exercises.

SOW	*RW*	*RH*	*RUW*	*Severity Description*
	>0.05	*>0.75*	*>0.5*	*The warranty fulfilment level is correct.*
	<0.05	*>0.75*	*>0.5*	*The warranty fulfilment level is correct, but the end of warranty is close.*
	-	*>0.75*	*0.5> RUW >0.4*	***ATTENTION**! Predicted 1 additional replacement. Early advice.*
	-	*0.75>RH>0.5*	*0.5> RUW >0.4*	***ATTENTION**! Predicted 1 additional replacement. Middle advice.*
	-	*0.75>RH>0.5*	*0.4> RUW >0*	***ATTENTION**! Predicted 1 additional replacement. Late advice.*
	-	*0.5>RH>0*	*0.4> RUW >0*	***DANGER**! Predicted 1 additional replacement. Irreversible damages.*
	-	-	*0*	***DANGER**! Predicted 2 additional replacement.*
	-	*0*	-	***DEATH**! The battery has reached the EOL.*
	¿?	*¿?*	*¿?*	***Undefined scenario.** Something unexpected is happening.*

Figure 14: Qualitative states of SoW.

- **Input parameters specifications**

In case of the SoW the inputs would be a set of parameters which will be previously uploaded to the cloud by the diagnostic-sizing tool.

Input parameters specifications
Number of inputs: 25.
Parameters: Listed below.
Data type: Doubles, being some simple doubles and others vector of doubles. Doubles: - **SoC max.** It is the maximum SoC value at which the operation is limited. - **SoC min.** It is the minimum SoC value at which the operation is limited. - **FECs per day.** It is the amount of Full Equivalent Cycles done per day. - **Days per year.** It is the amount of days of operation expected yearly. - **SoH at EoL.** The defined SoH value as the EoL criterion. - **Enom.** Nominal energy of the pack. - **Icha max.** Maximum charge current based on datasheet. - **Icha std.** The standard charge rate used to charge the battery system. - **Idch max.** The maximum peak discharge current. - **Idch mean.** The most likely mean discharge current value in operation. - **lifespan in years.** The lifespan of the battery system in terms of time. - **lifespan in FECs.** The lifespan of the battery system in terms of discharged energy. - **number of packs.** The number of packs that the project has. - **number of modules.** The number of modules that a pack has. - **number of cells.** The number of cells that a module has. - **ID of OEM.** A number that defines the OEM. - **ID of project.** A number that defines the project for the OEM. - **ID of system.** A number that defines the battery system (in case there are more than a battery system in a project). - **T max.** Maximum temperature the battery will work in operation. - **T avr.** The average temperature the battery will work in operation. - **T min.** Minimum temperature the battery will work in operation. Vector (doubles): - **xh0_time [5x1].** The parameters of the aging model that relates the SoH and the time. - **xh0_FEC [5x1].** The parameters of the aging model that relates the SoH and the FECs. - **SoH evolution** [40x1]. The SoH values obtained from the aging model to the FECs and time evolution. - **FECs evolution [40x1].** The FEC values that correlates with the obtained SoH values from the aging model.

- **time evolution [40x1].** The time in years that correlates with the obtained SoH values from the aging model.
Data acquisition: Through the database, which should have stored beforehand using the outcome from the datasheet and sizing tool.
Accuracy: N/A.
Data storage: N/A. Just the parameters at the beginning. Once stored them, they will not change.

- **Inputs specifications**

Usage and degradation related variables and/or parameters will be the inputs for the SoW.

Inputs specifications
Number of inputs: 3.
Input variables: SoH estimation, cumulative discharge capacity and timestamp.
Data type: Vectors of cell number x 1000. SoH and capacity are double. The timestamp is a datetime variable.
Data acquisition: Through the database, which should have stored the data through the cloud gateaway.
Accuracy: The SoH estimation should be +/- 3 % as worst. The cumulative discharge capacity should be +/-2 Ah as worst. The timestamp should be with +/-3 days as worst.
Data storage: Necessary. To Be Determined.

- **Outputs specifications**

Apart from the SoW, each maintenance substates will be also defined on a weekly basis.

Outputs specifications
Number of outputs: 4.
Output variables: SoW, RW, RH, RUW.
Data type: Matrix of strings (SoW) and doubles (RW, RH, RUW).
Data delivery: Weekly.

Accuracy: RW depends on the Timestamp accuracy. RH depends on the SoH accuracy. The RUW depends on the cumulative SoH accuracy. The SoW N/A.
Data storage: Not necessary.

5.3.4 Active balancing

Active balancing to be implemented by MIBA/AIT will require an algorithm in order to distribute the energy unbalance between cells.

- **Functionality specifications**

Active balancing consists of transmitting energy between cells in order to avoid unbalances and optimize energy usage. This will be controlled by a simple algorithm which will indicated when to start the balancing and finish it.

Functionality specifications
High level working explanation: The proposed balancing algorithm controls the Enable Pin (EN) of the ETA 3000 inductive balancer IC. If the voltage difference between two cells is above 100 mV the balancer IC must be enabled to start balancing. As an additional safety feature the maximum allowed balancing time is calculated based on the SoC and compared with the current balancing time. The balancing stops if all cells are within a voltage difference range of <= 30mV.
Methodology: N/A.
Integration: The algorithm is integrated in the master BMS.
Execution frequency: The algorithm will be periodically executed (in the range of seconds).
Execution time: Several microseconds.
Memory usage: < 1kByte.
Algorithm type: C-code.
Fidelity: Low.
Innovative: It is a classic active balancing, but still an improvement compared to the passive balancing.

- **Inputs specifications**

Cell voltage and SoC will be the inputs, since they are the variables to be monitored for the balancing.

Inputs specifications
Number of inputs: 2 per cell.
Input variables: Cell voltage and SoC.
Data type: Integer voltage variable and float SoC.
Data acquisition: Minimum input data frequency: several seconds.
Accuracy: Cell voltage accuracy: +/- 10 mV.
Data storage: Necessary. Current balancing time ($t_{balance}$) must be stored.

- **Outputs specifications**

The only regular output will be the enable of the balancing or not. Additionally, and if necessary only, error flags will be also transmitted.

Outputs specifications
Number of outputs: 2 at least (depending on number of error flags).
Output variables: Balancing enabler and error flags.
Data type: Integer or Boolean variables.
Data delivery: The balancing enabler will be updated periodically during charge, discharge, and resting periods within a time interval lower than a second. And the error flags will be sent to the master BMS, if necessary, during balancing.
Accuracy: N/A.
Data storage: Not necessary.

5.3.5 SoX based on smart sensing

During charging and discharging the mechanical properties of lithium-ion batteries (LIBs) change due to the intercalation processes of lithium-ions in the active materials. One of the most prominent is the volume change of LIBs during cycling. Especially the graphite anode changes its volume up to 10%. With the help of pressure and strain sensors these can be measured on cylindrical cells. The sensors are applied on the surface of the cell and due to the volume changes of the active materials, also volumetric changes of the cylindrical housing can be observed. Studies already showed that there is a correlation between the charging/discharging behaviour and the measured strain/pressure of LIBs. However, not much research was conducted on cylindrical cells with these types of sensors and the conducted studies only showed that there is a relation between the sensor data and the electric behaviour during cycling. In this project one of the main goals is to develop a SoX estimator in particular a state of charge (SoC) estimator based on the sensor data.

- **Functionality specifications**

Because of the novelty of the proposed SoX (SoC) estimator that will be developed in this project it is foreseen that the estimator will not be integrated in the BMS but tested and implemented in a laboratory environment. This means that the necessary data for the estimator will be generated on a single cell level in different environmental conditions (climate chamber). With this approach so-called training data will be gathered and further on used to write the algorithm which determines the SoX (SoC) of the cylindrical LIBs during charging/discharging.

Functionality specifications
High level working explanation: SoX estimator, in particular a state of charge (SoC) estimator, based on the sensor data of the cylindrical LIBs during charging/discharging.
Methodology: Currently it is foreseen that the SoX estimator will be model based (training data acquired from measurements). This might change during development and another approach might be used.
Integration: The developed SoX (SoC) estimator will be applied on different/new cells in a laboratory environment and the accuracy of the estimator will be determined. The programming and implementation will be done locally on a computer using Matlab/Python.
Execution frequency: The estimator can be executed periodically at a specific time interval based on the accuracy and the required state estimation in the final application.
Execution time: Due to the development and testing of the sensor-based SoX (SoC) estimator in a laboratory environment an exact execution time in the real application cannot exactly be defined. In the testing phase an execution time between 1-30 seconds is foreseen.
Memory usage: Can not be defined, due to the above-mentioned reasons.
Algorithm type: MATLAB/Phyton code.
Fidelity: Can not be defined, due to the above-mentioned reasons.

Innovative: Innovative. Not advanced sensing is usually included in classic BMS.

- **Inputs specifications**

Because of the novelty of the proposed SoX (SoC) estimator the number of inputs cannot be defined yet. Possible input data besides the measured strain/pressure can be the cell voltage, the current, the temperature, etc.

Inputs specifications
Number of inputs: Can not be defined yet.
Input variables: Possible cell measurements: strain, pressure, voltage, current, temperature, etc.
Data type: Numerical values (integer/float). Memory usage cannot be defined.
Data acquisition: Can not be defined yet.
Accuracy: Can not be defined yet.
Data storage: Due to the testing in a laboratory environment the data will be stored locally on a computer.

- **Outputs specifications**

As with the inputs specifications, output specifications are still to be defined.

Outputs specifications
Number of outputs: 1.
Output variables: SoX (SoC).
Data type: Numerical values (integer/float). Memory usage cannot be defined yet.
Data delivery: Cannot be defined due to the development and testing of the SoX (SoC) estimator in a laboratory environment.
Accuracy: The minimum necessary accuracy output data cannot be defined yet.
Data storage: In the scope of the project the SoX (SoC) estimator will be developed, tested, and validated in a laboratory environment. Therefore, the data will be stored locally on a computer.

6 CONCLUSION

This document presents all the specifications regarding HW topology, SW architecture, and advanced BMS algorithms of the Batter2Life solutions. After analysing the SoA and the different options, the selected approaches for each Pillar are presented.

- **Pillar 1 specifications**

In case of Pillar 1, a modular BMS topology is going to be implemented, where each module will be connected to a slave board and there will be one master board. At the same time, it will integrate the necessary HW to perform a module-based EIS on each slave board. The communication between the master and slave boards will be wireless. Cloud communication will be handled using an ESP32 module.

The proposed SW architecture for Pillar 1 includes BMS Master, BMS Slaves, and a Cloud Server. It follows AUTOSAR principles, separating hardware-dependent firmware from hardware-independent application software. Communication uses defined data structures, with REST APIs for cloud and edge BMS interaction. The BMS Master ensures interoperability, with high-level routines managed by a master state machine. Basic software handles hardware-dependent tasks, making the application layer hardware-independent. This design supports easy hardware adaptation and firmware-over-the-air updates for continued interoperability.

Advanced algorithms are also specified to ensure interoperability between functionalities and estimators. Additionally, these specifications will help for the later development of the SW and the algorithms. The specifications are presented in three blocks: functionality specification, input specifications and output specifications. This way the high-level working principle, input/output variables, execution frequency, necessary memory and other specifications are presented. A summary of estimators and functionalities integrated in Pillar 1 are presented below.

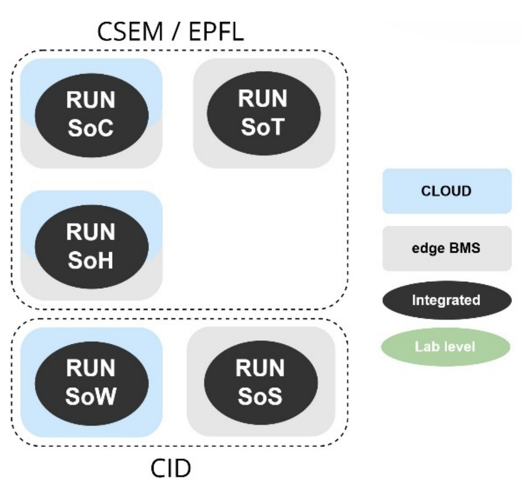


Figure 15: Advanced BMS algorithms in Pillar 1.

EIS-based SoC: SoC estimation is defined at cell level. It is a model-based approach which will run on the Master BMS. Additionally, model parameters will be recalculated regularly by obtaining EIS data from the cells, running the DRT algorithm on the cloud, and updating the parameters in the Master BMS.

SoH: SoH will be a basic differential-capacity approach, based on dQ/dV curves and Coulomb Counter. Even if, necessary measurements will be collected locally, the estimation will be carried out on the cloud.

EIS-based SoT: Internal temperature is estimated from high-frequency phase-shift of EIS. The estimation is carried out in the master BMS but defined at cell level.

EIS-based SoS: Estimates the safety level of the battery and gives a warning if lithium-plating and/or TRA is detected. It will consider cell measurements including EIS from one side, and the SoT from the other. It is also carried out in the master BMS but defined at cell level.

SoW: It is based on a multi-maintenance approach. The rule-based logic determines the SoW value based on quantitative substates: the remaining warranty, the remaining health and the remaining useful warranty. This estimation is periodically run in the cloud. And it needs, among others, the progress of SoH as input.

- **Pillar 2 specifications**

Regarding Pillar 2, a modular BMS topology is also going to be implemented, with a master-slaves structure. The main components are the Module Management System (MMS), the BMS master controller, the battery junction box, and the wireless communication gateway. The MMS consists of several ICs (LTC6811) that monitors 12 cell battery stacks. It also integrates the wireless microcontroller (CC2652R7), and the balancer IC (ETA 3000). The master BMS contains a Hercules microcontroller from Texas Instruments (TMS570 series). Additionally, pressure and strain sensors will be included to evaluate the possibility of SoX estimation based on volume changes.

In order to design a reliable, interoperable, safe, and scalable BMS SW architecture that supports both open-source and IP-restricted blocks, the BMS for Pillar 2 includes two microcontrollers connected via a serial bus for mutual monitoring and error correction. MCU1 (Hercules TMS570) handles cell monitoring and SoX algorithms, while MCU2 manages radio communication and serves as a backup. BMS applications run on FreeRTOS, with HAL generated by HALCoGen, and additional drivers enable communication with the inverter and cloud gateway.

Co-funded by the European Union

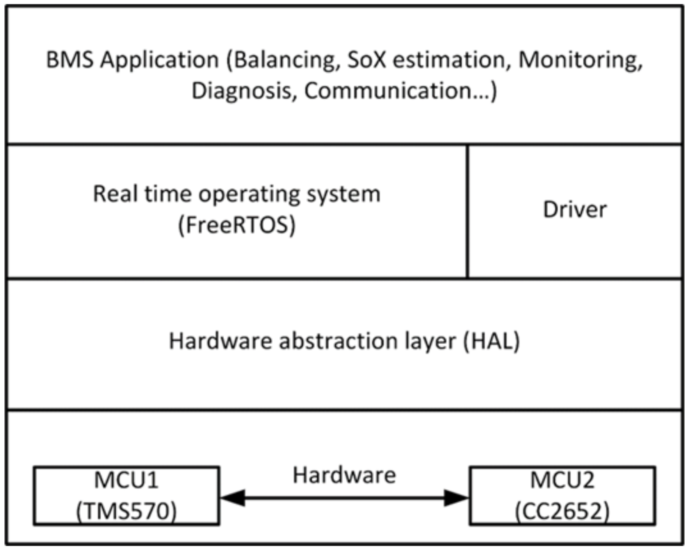


Figure 16: Software structure for BMS in Pillar 2.

In the case of Pillar 2, since it is a different approach, the estimators and functionalities to be included are also different. As in the case of Pillar 1, functionality, input and output specifications have been presented in the deliverable. A summary is presented below.

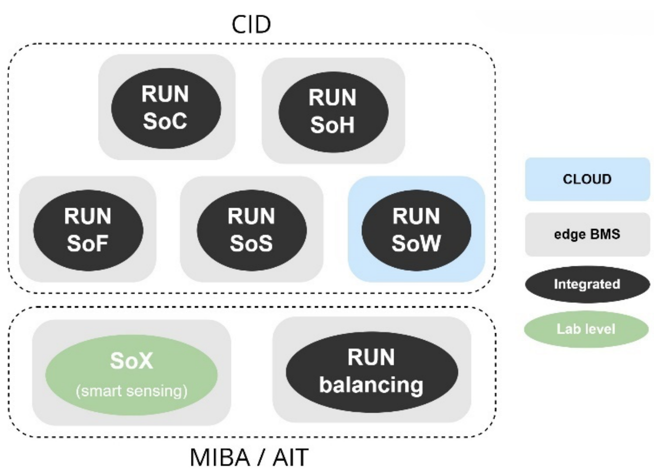


Figure 17: Advanced BMS algorithms in Pillar 2.

SoC: SoC will be estimated based on an EKF. This is an existing robust methodology with good accuracy level on complex battery dynamics. Cell voltage measurement is used to adjust the errors committed in the SoC estimation with Ah counting. It will be integrated in the Master BMS but defined at cell level.

SoH: As in pillar 1, SoH will be a basic differential-capacity approach. It will be updated periodically based on the difference between the theoretical capacity and the actual capacity charged at constant current in the evaluated SoC range given by the SoC estimator. As with SoC, it will be defined at cell level but calculated in the Master BMS.

SoS-based SoF: As EIS is not available in Pillar 2, a current limiting SoF will be developed, based on the SoS, which will avoid lithium-plating and TRA conditions during charge. Anyways, during resting period, dV/dt curves will be calculated to detect anomalies and update the SoS. Integrated in the Master BMS and defined at cell level.

SoW: The same approach is used for Pillar 1 and Pillar 2, the multi-maintenance analysis will be carried out in the cloud, based on remaining warranty, remaining health and remaining useful warranty, considering also SoH tendency.

Active Balancing: It consists of transmitting energy between cells in order to avoid unbalances and optimise energy usage. This will be controlled by a simple algorithm which will indicated when to start the balancing and finish it.

SoX based on smart sensing: With the help of pressure and strain sensors, volume changes of the active materials, and volumetric changes of the cylindrical housing can be observed. The idea is to develop high accuracy SoX estimators, in particular SoC estimator, based on the sensor data. This will be done at laboratory level.

Overall, the document presents a comprehensive approach towards developing reliable, interoperable, safe, and scalable BMS solutions for both pillars, accommodating both open-source and IP-restricted blocks and ensuring robust battery management through advanced algorithms and flexible software architecture.

7 REFERENCES

[1] M. Rosina, S. Agarwal, Li-ion Battery Pack for Automotive and Stationary Storage Applications 2020, 2020. www.yole.fr.

[2] John Warner, The Handbook of Lithium-Ion Battery Pack Design, Elsevier, 2015. https://doi.org/10.1016/C2013-0-23144-5.

[3] A. ŁEBKOWSKI, LIGHT ELECTRIC VEHICLE POWERTRAIN ANALYSIS, Scientific Journal of Silesian University of Technology. Series Transport 94 (2017) 123–137. https://doi.org/10.20858/sjsutst.2017.94.12.

[4] H. Bashir, A. Yaqoob, I. Jawaid, W. Khalid, M.Y. Javed, W. Sultan, A Review of Battery Management System and Modern State Estimation Approaches in Lithiumion Batteries for Electric Vehicle, in: 2022 5th International Conference on Energy Conservation and Efficiency, ICECE 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022. https://doi.org/10.1109/ICECE54634.2022.9758962.

[5] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, Z. Chen, A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems, Renewable and Sustainable Energy Reviews 131 (2020). https://doi.org/10.1016/j.rser.2020.110015.

[6] X. Hu, X. Deng, F. Wang, Z. Deng, X. Lin, R. Teodorescu, M.G. Pecht, A Review of Second-Life Lithium-Ion Batteries for Stationary Energy Storage Applications, Proceedings of the IEEE 110 (2022) 735–753. https://doi.org/10.1109/JPROC.2022.3175614.

[7] E.M.A.G.N.C. Ekanayake, K.T.M.U. Hemapala, U. Jayathunga, Active and Passive Based Hybrid Cell Balancing Approach to Series Connected Lithium-ion Battery Pack, in: MERCon 2022 - Moratuwa Engineering Research Conference, Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022. https://doi.org/10.1109/MERCon55799.2022.9906172.

[8] Z. Chen, W. Liao, P. Li, J. Tan, Y. Chen, Simple and high-performance cell balancing control strategy, Energy Sci Eng (2022). https://doi.org/10.1002/ese3.1243.

[9] R. Swarnkar, H. Ramachandran, S.H.M. Ali, R. Jabbar, A Systematic Literature Review of State of Health and State of Charge Estimation Methods for Batteries Used in Electric Vehicle Applications, World Electric Vehicle Journal 14 (2023). https://doi.org/10.3390/wevj14090247.

[10] S. Park, J. Ahn, T. Kang, S. Park, Y. Kim, I. Cho, J. Kim, Review of state-of-the-art battery state estimation technologies for battery management systems of stationary energy

storage systems, Journal of Power Electronics 20 (2020) 1526–1540. https://doi.org/10.1007/s43236-020-00122-7.

[11] P. Shen, M. Ouyang, L. Lu, J. Li, X. Feng, The co-estimation of state of charge, state of health, and state of function for lithium-ion batteries in electric vehicles, IEEE Trans Veh Technol 67 (2018) 92–103. https://doi.org/10.1109/TVT.2017.2751613.

[12] G.L. Plett, High-performance battery-pack power estimation using a dynamic cell model, IEEE Trans Veh Technol 53 (2004) 1586–1593. https://doi.org/10.1109/TVT.2004.832408.

[13] L. Lu, X. Han, J. Li, J. Hua, M. Ouyang, A review on the key issues for lithium-ion battery management in electric vehicles, J Power Sources 226 (2013) 272–288. https://doi.org/10.1016/j.jpowsour.2012.10.060.

[14] S. Shahid, M. Agelin-Chaab, A review of thermal runaway prevention and mitigation strategies for lithium-ion batteries, Energy Conversion and Management: X 16 (2022). https://doi.org/10.1016/j.ecmx.2022.100310.

[15] G. Zhang, X. Wei, X. Tang, J. Zhu, S. Chen, H. Dai, Internal short circuit mechanisms, experimental approaches and detection methods of lithium-ion batteries for electric vehicles: A review, Renewable and Sustainable Energy Reviews 141 (2021). https://doi.org/10.1016/j.rser.2021.110790.

[16] X. Feng, M. Ouyang, X. Liu, L. Lu, Y. Xia, X. He, Thermal runaway mechanism of lithium ion battery for electric vehicles: A review, Energy Storage Mater 10 (2018) 246–267. https://doi.org/10.1016/j.ensm.2017.05.013.

[17] T. Yamauchi, K. Mizushima, Y. Satoh, S. Yamada, Development of a simulator for both property and safety of a lithium secondary battery, J Power Sources 136 (2004) 99–107. https://doi.org/10.1016/j.jpowsour.2004.05.011.

[18] S. Shahid, M. Agelin-Chaab, A review of thermal runaway prevention and mitigation strategies for lithium-ion batteries, Energy Conversion and Management: X 16 (2022). https://doi.org/10.1016/j.ecmx.2022.100310.

[19] X. Feng, M. Ouyang, X. Liu, L. Lu, Y. Xia, X. He, Thermal runaway mechanism of lithium ion battery for electric vehicles: A review, Energy Storage Mater 10 (2018) 246–267. https://doi.org/10.1016/j.ensm.2017.05.013.

[20] G. Zhang, X. Wei, X. Tang, J. Zhu, S. Chen, H. Dai, Internal short circuit mechanisms, experimental approaches and detection methods of lithium-ion batteries for electric vehicles: A review, Renewable and Sustainable Energy Reviews 141 (2021). https://doi.org/10.1016/j.rser.2021.110790.

[21] J. Ye, H. Chen, Q. Wang, P. Huang, J. Sun, S. Lo, Thermal behavior and failure mechanism of lithium ion cells during overcharge under adiabatic conditions, Appl Energy 182 (2016) 464–474. https://doi.org/10.1016/j.apenergy.2016.08.124.

[22] Y. Zeng, K. Wu, D. Wang, Z. Wang, L. Chen, Overcharge investigation of lithium-ion polymer batteries, J Power Sources 160 (2006) 1302–1307. https://doi.org/10.1016/j.jpowsour.2006.02.009.

[23] J. Liu, Z. Wang, J. Bai, T. Gao, N. Mao, Heat generation and thermal runaway mechanisms induced by overcharging of aged lithium-ion battery, Appl Therm Eng 212 (2022). https://doi.org/10.1016/j.applthermaleng.2022.118565.

[24] L. Zhang, Y. Ma, X. Cheng, C. Du, T. Guan, Y. Cui, S. Sun, P. Zuo, Y. Gao, G. Yin, Capacity fading mechanism during long-term cycling of over-discharged LiCoO2/mesocarbon microbeads battery, J Power Sources 293 (2015) 1006–1015. https://doi.org/10.1016/j.jpowsour.2015.06.040.

[25] R. Guo, L. Lu, M. Ouyang, X. Feng, Mechanism of the entire overdischarge process and overdischarge-induced internal short circuit in lithium-ion batteries, Sci Rep 6 (2016). https://doi.org/10.1038/srep30248.

[26] D. Wang, L. Zheng, X. Li, G. Du, Z. Zhang, Y. Feng, L. Jia, Z. Dai, Effects of Overdischarge Rate on Thermal Runaway of NCM811 Li-Ion Batteries, Energies (Basel) 13 (2020). https://doi.org/10.3390/en13153885.

[27] S. Erol, M.E. Orazem, R.P. Muller, Influence of overcharge and over-discharge on the impedance response of LiCoO 2 | C batteries, J Power Sources 270 (2014) 92–100. https://doi.org/10.1016/j.jpowsour.2014.07.038.

[28] X. Chen, L. Li, M. Liu, T. Huang, A. Yu, Detection of lithium plating in lithium-ion batteries by distribution of relaxation times, J Power Sources 496 (2021). https://doi.org/10.1016/j.jpowsour.2021.229867.

[29] U. Janakiraman, T.R. Garrick, M.E. Fortier, Review—Lithium Plating Detection Methods in Li-Ion Batteries, J Electrochem Soc 167 (2020) 160552. https://doi.org/10.1149/1945-7111/abd3b8.

[30] U.R. Koleti, T.Q. Dinh, J. Marco, A new on-line method for lithium plating detection in lithium-ion batteries, J Power Sources 451 (2020). https://doi.org/10.1016/j.jpowsour.2020.227798.

[31] X. Wang, J. Li, S. Chen, G. Zhang, B. Jiang, X. Wei, H. Dai, Online Detection of Lithium Plating Onset for Lithium-Ion Batteries Based on Impedance Changing Trend Identification During

Charging Processes, IEEE Transactions on Transportation Electrification 9 (2023) 3487–3497. https://doi.org/10.1109/TTE.2022.3224965.

[32] X. Lin, K. Khosravinia, X. Hu, J. Li, W. Lu, Lithium Plating Mechanism, Detection, and Mitigation in Lithium-Ion Batteries, Prog Energy Combust Sci 87 (2021). https://doi.org/10.1016/j.pecs.2021.100953.

[33] J. Zhao, A.F. Burke, Electric Vehicle Batteries: Status and Perspectives of Data-Driven Diagnosis and Prognosis, Batteries 8 (2022). https://doi.org/10.3390/batteries8100142.

[34] X. Hu, F. Feng, K. Liu, L. Zhang, J. Xie, B. Liu, State estimation for advanced battery management: Key challenges and future trends, Renewable and Sustainable Energy Reviews 114 (2019). https://doi.org/10.1016/j.rser.2019.109334.

[35] E. Cabrera-Castillo, F. Niedermeier, A. Jossen, Calculation of the state of safety (SOS) for lithium ion batteries, J Power Sources 324 (2016) 509–520. https://doi.org/10.1016/j.jpowsour.2016.05.068.

[36] C. Wang, H.H. Ghenniwa, W. Shen, Distributed scheduling for reactive maintenance of complex systems, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 38 (2008) 269–273. https://doi.org/10.1109/TSMCC.2007.913916.

[37] A.T. de Almeida, C.A.V. Cavalcante, M.H. Alencar, R.J.P. Ferreira, A.T. de Almeida-Filho, T.V. Garcez, Preventive Maintenance Decisions, in: International Series InOperations Research & Management Science - Multicriteria and Multiobjective Modelsfor Risk, Reliability and Maintenance Decision Analysis, Springer, 2015: pp. 215–232. https://doi.org/10.1007/978-3-319-17969-8_5.

[38] F. Calabrese, A. Regattieri, M. Bortolini, M. Gamberi, F. Pilati, Predictive maintenance: a novel framework for a data-driven, semi-supervised, and partially online prognostic health management application in industries, Applied Sciences (Switzerland) 11 (2021). https://doi.org/10.3390/app11083380.

[39] S. Park, J. Ahn, T. Kang, S. Park, Y. Kim, I. Cho, J. Kim, Review of state-of-the-art battery state estimation technologies for battery management systems of stationary energy storage systems, Journal of Power Electronics 20 (2020) 1526–1540. https://doi.org/10.1007/s43236-020-00122-7.

[40] M. Arrinda, D. Sánchez, M. Oyarbide, H. Macicior, A. Zubiria, Development of the State of Warranty (SOW) for Electric Vehicles, World Electric Vehicle Journal 13 (2022). https://doi.org/10.3390/wevj13080135.