UNIVERSITÀ DI PISA
DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

ADVANCES IN ALGORITHMS AND MANAGEMENT SYSTEMS FOR LITHIUM-ION BATTERIES

DOCTORAL THESIS

Author
Rocco Morello

Tutors
Prof. Roberto Saletti
Prof. Federico Baronti

Reviewers
Prof. Walter Zamboni
Prof. Ali Sari

The Coordinator of the PhD Program
Prof. Marco Luise

Pisa, November 2018

XXXI

Summary

THE use of the lithium-ion technology in energy storage systems has enabled the spreading of small portable devices, like smartphones and laptops, but also of medium-power devices, electric mobility, and renewable energies, where batteries are built by connecting together a large number of lithium-ion cells. This thanks to higher energy and power densities, longer cycle life, and lower self-discharge with respect to the other battery chemistries. However, they need to be monitored by an electronic system, the so called Battery Management System (BMS), in order to work in a safe and effective way. In fact, lithium-ion batteries can be subjected to degradation or hopelessly damaged if they work outside of their safe operating area. The BMS constantly measures all the cell physical quantities, such as current, voltage, and temperature, and estimates the internal state of all the cells composing the battery pack, including the State Of Charge (SOC) and the State Of Health (SOH). The required accuracy of the measurement and of the estimate depends on the final application. As an example, in automotive applications, an accurate estimation of the charge stored into the battery allows a better knowledge of the remaining driving range. Furthermore, the estimation of the grade of degradation in terms of internal resistance increasing and capacity fading enables the possibility to reuse the cells that have reached the end of life in this application (around the $80\,\%$ of the nominal capacity) for building second life batteries for other fields, such as the stationary energy storage. This because second life batteries must be composed by cells with a homogeneous degradation level to guarantee the best performance.

Algorithms for an accurate estimation of the internal battery state require a significant computational power to be executed online during the battery operation, especially when the number of cells that compose the battery is large. For SOC estimation, an effective solution is the use model-based estimation algorithms since they are characterised by a good trade-off between accuracy and complexity, particularly when an equivalent circuit model is used. The model is used in a closed-loop to correct the estimated value, and the accuracy of the estimation depends on the model capability to reproduce the cell behaviour. To improve the model performance, an online model parameter estimation algorithm can be flanked to the model-based technique to track

the parameter variations with the operating conditions of the battery, such as SOC, temperature, and current. We developed two promising solutions for SOC and parameter co-estimation: the Adaptive Mix Algorithm (AMA) and the Dual Extended Kalman Filter (DEKF). The AMA is based on the joining of the mix algorithm for SOC estimation and the moving window least squares for parameter identification, while in the DEKF these two tasks are performed by two cooperative EKFs. Both the algorithms have been implemented as hardware accelerators on a Field-Programmable Gate Array (FPGA) to improve the estimation performance in terms of execution time with respect to their implementation in software and to be used besides a controller that simultaneously performs the other BMS safety and management functions. The performance in terms of execution time are promising and the DEKF performs better than the AMA. This because the AMA is more complex than the DEKF even if allows a better tuning of the algorithm parameters. To this aim we also developed a procedure that allows the designer to choose the best set of AMA parameters starting from the current and voltage profiles representative of the target application.

During the development of these algorithms and the verification phases we employed a communication-based Hardware-in-the-Loop (HiL) platform that has been developed to simulate the behaviour of a complete configurable battery used in an electric vehicle under many different operating conditions and to analyse the estimator implementation executed on a real BMS. Thanks to this platform we validated both the hardware estimators, and we have been able to compare their performance in terms of execution time, estimation accuracy, and wrong SOC initialization recovery time. The use of large set of scenarios highlighted the necessity to analyse the algorithms in different conditions, since the algorithms behave in different ways. In particular, the AMA provides better results on urban cycles, whereas the DEKF is more reliable for motorway schedules, when the required electric power is higher and the speed profile more static.

Using an FPGA instead of a microcontroller in a BMS brings to various advantages, besides the increase of the computational power, such as a higher flexibility and reconfigurability of the BMS. In this work, two BMS implementations with an FPGA-based architecture are reported: a basic BMS which uses stack monitors to manages groups of cells, and an advanced one which uses smart cells. The first has been developed for an e-bike, a case study that allowed us to test the BMS on the field during real utilization. The e-bike battery is composed of 10 cells, managed by a single stack monitor, an integrated circuit able to measure all the cell voltages, but also additional signals such as the output of a current sensor, and to send the acquired information to the main unit of the BMS. This is based on an Intel Cyclone® V system on chip FPGA with a hard-core ARM® Cortex™ A9 used to execute the software functions. Results have assessed the BMS implementation and showed the effectiveness of the AMA hardware estimator. The same hardware estimator is also used in the advanced BMS, so called Research BMS, built for the 3Ccar project which aims at investigating a new battery architecture based on smart cells. The latter are cells provided with some electronics which allows the execution of various functions, such as the measurement of the cell current, voltage, and temperature, and to communicate with the Research BMS. The BMS main unit belongs to the same family of that of the BMS described above, where the hard-core is also used to run the estimation algorithms developed by the project partners, such as the

Temperature Prediction Algorithm used to predict the cell core temperatures. Besides the Research BMS, we developed a framework composed of three parts: a LabVIEW user interface, which has been also used during the battery experimental tests to manage the BMS, a LabVIEW smart cells emulator, which emulates the communication with the smart cells, and an emulator of the ARM® hard-core, used by the partners to create their own algorithms. The battery assembly has been carried out during the final tests, where the Research BMS has been connected to the smart cells developed by the project partners. The performed tests demonstrated the correct battery operation, for both hardware and software components.

Sommario

I sistemi di accumulo dell'energia basati sulle batterie agli ioni di litio hanno permesso la diffusione di piccoli dispositivi portatili, come smartphone e laptop, ma anche di dispositivi di media potenza, della mobilità elettrica e delle energie rinnovabili, applicazioni in cui le batterie sono costruite con un gran numero di celle agli ioni di litio connesse tra loro. Questo perché, rispetto alle altre chimiche, esse hanno una densità di energia e di potenza più elevata, un maggior numero di cicli di carica/scarica e un'autoscarica minore. Ad ogni modo, queste batterie necessitano di un monitoraggio costante da parte di un sistema elettronico, chiamato Battery Management System (BMS), in modo da garantire un funzionamento sicuro ed efficiente. Infatti, se le batterie agli ioni di litio lavorano fuori dalla loro area operativa sicura possono degradarsi o subire danni irreparabili.

Il BMS misura costantemente tutte le grandezze fisiche delle celle che compongono il pacco batteria, come corrente, tensione e temperatura, e stima il loro stato interno, incluso lo stato di carica (State Of Charge, SOC) e lo stato di salute (State Of Health, SOH). L'accuratezza richiesta per la misura delle grandezze e per la stima delle variabili di stato dipende dall'applicazione. Per esempio, nelle applicazioni automotive, una stima accurata della carica immagazzinata nella batteria permette una conoscenza migliore dell'autonomia residua del veicolo. Inoltre, una stima del grado di invecchiamento della batteria, in termini di aumento della resistenza interna e di diminuizione della capacità, permette di riutilizzare le celle che hanno raggiunto la fine della loro vita. In applicazioni automotive, questo equivale a dire che le celle hanno raggiunto un valore di capacità pari all'$80\,\%$ della loro capacità nominale. Queste celle sono comunque ancora usabili per costruire batterie utili in altri campi, come l'immagazzinamento dell'energia in applicazioni stazionarie. Una stima accurata dell'invecchiamento è utile in quanto in queste batterie le celle devono avere una degradazione omogenea in modo da consentire le migliori prestazioni.

Una stima accurata dello stato può essere ottenuta con l'esecuzione di algoritmi che possono richiedere una capacità computazionale significativa, soprattutto quando il numero di celle che compongono il pacco batteria è elevato. Per quanto riguarda la stima del SOC, una soluzione efficace è quella di usare algoritmi di stima di tipo

model-based, in quanto caratterizzati da un buon rapporto tra accuratezza e complessità, in particolare quando viene usato un modello elettrico equivalente. Il modello viene inserito in un ciclo di retroazione per correggere il valore della stima e l'accuratezza del risultato dipende dalla capacità del modello di riprodurre il comportamento della cella. Per migliorare le prestazioni del modello circuitale si può affiancare alla tecnica di stima *model-based* un algoritmo d'identificazione online dei parametri del modello in modo da tenere traccia delle variazioni del valore di questi parametri con le condizioni operative della batteria, come SOC, temperatura e corrente.

Sono state sviluppate due promettenti soluzioni per la stima congiunta di SOC e parametri: l'Adaptive Mix Algorithm (AMA) ed il Dual Extended Kalman Filter (DEKF). L'AMA si basa sull'unione del Mix Algorithm per la stima del SOC e dell'algoritmo del Moving Window Least Squares per l'identificazione dei parametri, mentre nel DEKF questi due compiti vengono svolti da due filtri di Kalman estesi che lavorano in maniera cooperativa. Entrambi gli algoritmi sono stati implementati come acceleratori hardware su una Field-Programmable Gate Array (FPGA) per migliorare le prestazioni della stima in termini di tempo di esecuzione rispetto alla loro implementazione in software, ed in modo da poter utilizzare questi acceleratori affianco ad un controllore che è libero di eseguire contemporaneamente tutte le altre funzioni di sicurezza e gestione del BMS. Le prestazioni ottenute in termini di tempo di esecuzione sono promettenti e il DEKF risulta essere più efficiente dell'AMA. Questo perché l'AMA è più complesso, anche se permette di scegliere in maniera più efficace i parametri dell'algoritmo. A questo scopo, è stata sviluppata una procedura che permette al progettista di scegliere la migliore combinazione di tali parametri, partendo dai profili di corrente e tensione acquisiti dall'applicazione di riferimento.

Durante le fasi di sviluppo e verifica è stata utilizzata una piattaforma *hardware-in-the-loop*, realizzata per simulare il comportamento di una batteria completamente configurabile durante il suo utilizzo in un veicolo elettrico e in molte condizioni operative diverse, in modo da analizzare l'implementazione dello stimatore eseguito in un vero BMS che comunica direttamente con la piattaforma. Grazie a quest'ultima sono stati validati entrambi gli stimatori e siamo stati capaci di comparare le loro prestazioni per quanto riguarda i tempi d'esecuzione, l'accuratezza della stima e il tempo di correzione a fronte di un'inizializzazione sbagliata dello stato di carica. Analizzando gli algoritmi in diversi scenari, simulati dalla piattaforma, si è potuto notare come i risultati ottenuti dipendono dalle diverse condizioni operative. In particolare, l'AMA produce risultati migliori durante i cicli urbani, mentre il DEKF è più affidabile nei cicli autostradali, quando la potenza elettrica richiesta è più alta e il profilo di velocità più statico.

L'utilizzo di una FPGA invece di un microcontrollore in un BMS porta a vari vantaggi come una maggiore flessibilità e riconfigurabilità del BMS, oltre all'incremento delle capacità computazionali. In questo lavoro sono riportate due implementazioni di BMS con architettura basata su FPGA: un BMS basilare che utilizza degli *stack monitor* per gestire gruppi di celle e un BMS avanzato che utilizza delle *smart cell*.

Il primo è sviluppato per una bici elettrica, un caso studio particolare che ci ha permesso di testare sul campo il sistema durante un reale utilizzo. La batteria della bici elettrica è composta da 10 celle gestite da un solo *stack monitor*, un circuito integrato capace di misurare tutte le tensioni di cella, ma anche segnali addizionali, come l'uscita di un sensore di corrente, e di inviare le informazioni acquisite all'unità centrale

del BMS. Quest'ultima è basata su un *system on chip* FPGA Intel Cyclone® V con all'interno un processore hardware ARM® Cortex™ A9 usato per eseguire le funzioni software. I risultati ottenuti hanno validato l'implementazione del BMS e hanno mostrato l'efficienza dello stimatore hardware che implementa l'AMA.

Lo stesso stimatore è stato implementato anche nel BMS avanzato, chiamato Research BMS, realizzato per il progetto 3Ccar, che ha lo scopo di investigare una nuova architettura basata su *smart cell*. Le *smart cell* sono equipaggiate con dell'elettronica che permette di eseguire varie funzioni, come la misura della corrente, della tensione e della temperatura della cella, e di comunicare con il Research BMS. L'unità centrale del BMS appartiene alla stessa famiglia di quella descritta in precedenza, dove il processore hardware è anche usato per eseguire degli algoritmi di stima sviluppati dai partner del progetto, come il Temperature Prediction Algorithm usato per stimare la temperatura all'interno della cella. Oltre al Research BMS è stato anche realizzata una piattaforma di sviluppo che comprende tre parti: un'interfaccia utente basata su LabVIEW, utilizzata anche per gestire il BMS durante i test sperimentali finali, un emulatore delle *smart cell* sempre basato su LabVIEW, capace di emulare la comunicazione con le celle, e un emulatore del processore hardware ARM®, usato dai partner per sviluppare i propri algoritmi. L'assemblaggio della batteria è stato realizzato durante i test finali, dove il Research BMS è stato connesso alle *smart cell* sviluppate dagli altri partner del progetto. I test eseguiti hanno dimostrato il corretto funzionamento della batteria, sia per quanto riguarda le componenti hardware, che quelle software.

List of publications

International Journals

1. Morello, R., Di Rienzo, R., Roncella, R., Saletti, R., and Baronti, F. (2018). Hardware-in-the-Loop Platform for Assessing Battery State Estimators in Electric Vehicles. IEEE Access, 6, 68210-68220.

International Conferences/Workshops with Peer Review

1. Morello, R., Zamboni, W., Baronti, F., Di Rienzo, R., Roncella, R., Spagnuolo, G., and Saletti, R. (2015, November). Comparison of state and parameter estimators for electric vehicle batteries. In IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society (pp. 005433-005438). IEEE.

2. Morello, R., Baronti, F., Tian, X., Chau, T., Di Rienzo, R., Roncella, R., Jeppesen, B., Lin, W. H., Ikushima, T., and Saletti, R. (2016, June). Hardware-in-the-loop simulation of FPGA-based state estimators for electric vehicle batteries. In 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), (pp. 280-285). IEEE.

3. Morello, R., Di Rienzo, R., Baronti, F., Roncella, R., and Saletti, R. (2016, October). System on chip battery state estimator: E-bike case study. In IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, (pp. 2129-2134). IEEE.

4. Tian, X., Jeppesen, B., Ikushima, T., Baronti, F., and Morello, R. (2016, November). Accelerating state-of-charge estimation in FPGA-based Battery Management Systems. In 6th Hybrid and Electric Vehicles Conference (HEVC 2016), (pp. 1-6). IET.

5. Morello, R., Di Rienzo, R., Roncella, R., Saletti, R., and Baronti, F. (2017, June). Tuning of Moving Window Least Squares-based algorithm for online battery parameter estimation. In 2017 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), (pp. 1-4). IEEE.

6. Di Rienzo, R., Baronti, F., Roncella, R., Morello, R., and Saletti, R. (2017, June). Simulation platform for analyzing battery parallelization. In 2017 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), (pp. 1-4). IEEE.

7. Litta, G., Di Rienzo, R., Morello, R., Roncella, R., Baronti, F., and Saletti, R. (2018, January). Flexible platform with wireless interface for DC-motor remote control. In 2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES), (pp. 509-514). IEEE.

8. Di Rienzo, R., Roncella, R., Morello, R., Baronti, F., and Saletti, R. (2018, January). Low-cost modular battery emulator for battery management system testing. In 2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES), (pp. 44-49). IEEE.

9. Morello, R., Di Rienzo, R., Roncella, R., Saletti, R., Schwarz, R., Lorentz, V.R.H., Hoedemaekers, E.R.G., Rosca, B., and Baronti, F. (2018, October). Advances in Li-Ion Battery Management for Electric Vehicles. In IECON 2018 - 44st Annual Conference of the IEEE Industrial Electronics Society. IEEE.

Contents

1 Introduction **1**

2 Lithium-ion energy storage systems **3**
2.1 Introduction . . . 3
2.2 Lithium-ion technology . . . 3
2.3 Battery management system . . . 5
2.3.1 Cell monitoring . . . 6
2.3.2 State estimation . . . 6
2.3.3 Safety . . . 13
2.3.4 Charge equalization . . . 14
2.3.5 Communication and data management . . . 14
2.4 BMS architectures . . . 15
2.4.1 Discussion . . . 19

3 Advanced state estimation algorithms **21**
3.1 Introduction . . . 21
3.2 Dual Extended Kalman Filter . . . 22
3.3 Adaptive Mix Algorithm . . . 25
3.3.1 SOC estimation . . . 26
3.3.2 Online parameter identification . . . 29
3.3.3 MWLS tuning procedure . . . 32

4 Hardware acceleration of the battery state estimators **37**
4.1 Introduction . . . 37
4.2 Design flow . . . 38
4.3 Adaptive Mix Algorithm . . . 39
4.4 Dual Extended Kalman Filter . . . 42

5 Hardware-in-the-loop platform for hardware estimators assessment **45**
5.1 Introduction . . . 45
5.2 Hardware-in-the-loop platform design . . . 47

5.2.1 Driving schedules 47
5.2.2 Electric vehicle model 49
5.2.3 Electric traction model 49
5.2.4 Battery model 50
5.2.5 Sensor model 52
5.3 Developed Simulink Library 53
5.4 HiL platform for validating hardware estimators 54
5.4.1 BMS hardware 54
5.4.2 HiL platform instance 55
5.4.3 Noiseless acquisition system 56
5.4.4 Noisy acquisition system 59
5.4.5 Unbalanced cells 59

6 FPGA-based battery management systems implementation and assessment 61
6.1 Introduction 61
6.2 Basic FPGA-based BMS: e-bike case study 61
6.2.1 Validation methodology 64
6.2.2 Experimental results 65
6.3 FPGA-based BMS with Smart Cells 68
6.3.1 Developed framework 72
6.3.2 Experimental results 76

7 Conclusions 81

Bibliography 83

CHAPTER 1

Introduction

Nowadays, electric devices with a local Energy Storage System (ESS) are widespread. Good examples are electronic portable devices like laptops, tablets and smartphones, but also medium-power tools like drills and lawn-mowers. Other fields in which the ESS is a key component are the electric mobility and the stationary applications. The number of electric vehicles is rapidly growing in the last years. In fact, the data published by the International Energy Agency shows that in the world there are about 3.1 millions of electric cars and in the 2017 the sells grew up of about $54\,\%$ with respect of 2016 [65]. The increment of the electric transportation, the decentralization of the electrical energy generation and the digitalization are the causes of the diffusion of smart grids, in which the ESSs are used to support the energy distribution.

The main components used to store energy in an ESS are batteries that can be based on different technologies. The most employed are the lead acid, the nickel-metal hydride (NiMH) and the Lithium-ion (Li-ion) batteries. The Li-ion technology presents various advantages with respect to the others and for this reason it enables the electrification of vehicles and devices used in different fields. In fact, Li-ion batteries have higher energy and power densities, longest cycle and calendar life, and lower self-discharge compared to the other battery technologies. However, this technology has some limitations that require Li-ion batteries to be equipped with an electronic control board called Battery Management System (BMS). The main objective of the BMS is to ensure a safe battery operation and to allow the battery lifetime extension. In fact, the battery can be subjected to an accelerated ageing or serious damages if it works out of its safe operating area in terms of voltage, temperature, and current. The BMS main unit also estimates the internal state variables of the battery in order to avoid these dangerous situations and to obtain useful battery information, such as the remaining stored charge, quantified by the State Of Charge (SOC), or the degradation grade, expressed

by the State Of Health (SOH) variable.

During this Ph.D. program, different state estimation algorithms and BMS architectures for medium and large-format batteries have been analysed and developed. In particular, two model-based algorithms, the Adaptive Mix Algorithm (AMA) and the Dual Extended Kalman Filter (DEKF), have been implemented in hardware on a Field-Programmable Gate Array (FPGA) in order to meet with their computational requirements and to enable the state estimation of a large number of cells. Both algorithms are closed-loop techniques which employ an electrical model of the cell flanked to an online model parameter identification algorithm to improve the SOC estimation accuracy.

Furthermore, a Hardware-in-the-Loop (HiL) platform able to emulate a configurable battery during its use in an electric vehicle has been developed in MATLAB/Simulink® environment and used to test the hardware estimators during their implementation. The model used to emulate the battery behaviour has been made available in order to allow the users to build their own simulation platform that can be adapted to test different BMSs and used for different purposes. The main advantage of the developed platform is that it can be used to verify a BMS in a large number of simulated operating conditions with completely safe tests, since there is no power exchange between the BMS and the simulator.

Besides to a higher computational power, the use of an FPGA as the main unit of a BMS allows the system to be more flexible and reconfigurable with respect to a BMS based on a microcontroller. In this way, a BMS can be easily adapted for other batteries or applications. Furthermore, it can execute more complex and accurate estimation algorithms in order to prevent negative situations and to accurately know the degradation grade of the battery, enabling also the reuse of the cells in other applications. We build two innovative BMSs with the main unit based on an FPGA, where the SOC estimation is executed by the AMA hardware accelerator, and each BMS has been validated by using a particular case-study. The first one has been implemented to test the basic functions of a BMS and the SOC estimator during the real utilization of an electric bike on the road. The second BMS has been developed for a $48\,\mathrm{V}$ smart battery within the 3Ccar European project, which aims at investigating the use in automotive applications of an architecture based on the concept of smart cells, *i.e.*, cells provided with an electronic board and some sensors, and able to communicate information to the BMS. The latter can run complex estimation algorithms developed both as hardware estimator and software applications.

The thesis first reports on the main characteristics of the Li-ion technology and the functionalities and the various architectures of the BMS in Chapter 2. At the end of this chapter, we also introduced the concept of BMSs based on an FPGA. The explanation of the two implemented SOC estimation algorithms is reported in the third chapter. The fourth and the fifth chapters report on the hardware implementation of the SOC estimation algorithms and their verification by using the HiL platform. The development and the verification of two BMSs which use an FPGA have been described in Chapter 6, in which two different demonstrators have been used to analyse their performance. Finally, some considerations are summarised in the conclusion chapter.

CHAPTER 2

Lithium-ion energy storage systems

2.1 Introduction

This chapter aims at reporting on the basic information about the lithium-ion technology. In particular, after a brief introduction on the structure of the Li-ion cells, Section 2.2 describes and compares the advantages and the limitations of the various lithium-ion typologies available on the market. One of the main limitations is the necessity to equip the Li-ion batteries with an electronic control board called BMS in order to ensure a safe functioning of the battery pack. The main functions of the BMS are described in Section 2.3, that highlights the importance of the battery state estimation, especially regarding the state of charge and the state of health variables. These functions can be implemented using different BMS architectures, as explained in Section 2.4.

2.2 Lithium-ion technology

Li-ion batteries are widely used in many applications thanks to various advantages introduced with respect to the lead acid and NiMH chemistries. They present a low self-discharge rate, a higher nominal voltage and a longer cycle life. Furthermore, they have a higher energy and power densities with respect to volume and weight [29, 33]. This allows to build lighter and smaller ESSs and to increase the portability in portable devices and the driving range in Electric Vehicles (EVs).

Depending on the application, batteries are composed by connecting together a certain number of Li-ion cells, in series in order to reach the desired voltage and in parallel to increment the battery capacity. Usually, cells are grouped in modules for a better organization of the battery. The Li-ion cell structure and functioning is shown in Figure 2.1. Each cell is composed by a negative electrode (often referred to as the anode), usually made of graphite, and a positive electrode (often referred to as the cathode) that

can be realized using different materials, characterizing the Li-ion technology as shown in Table 2.1 [55, 137].

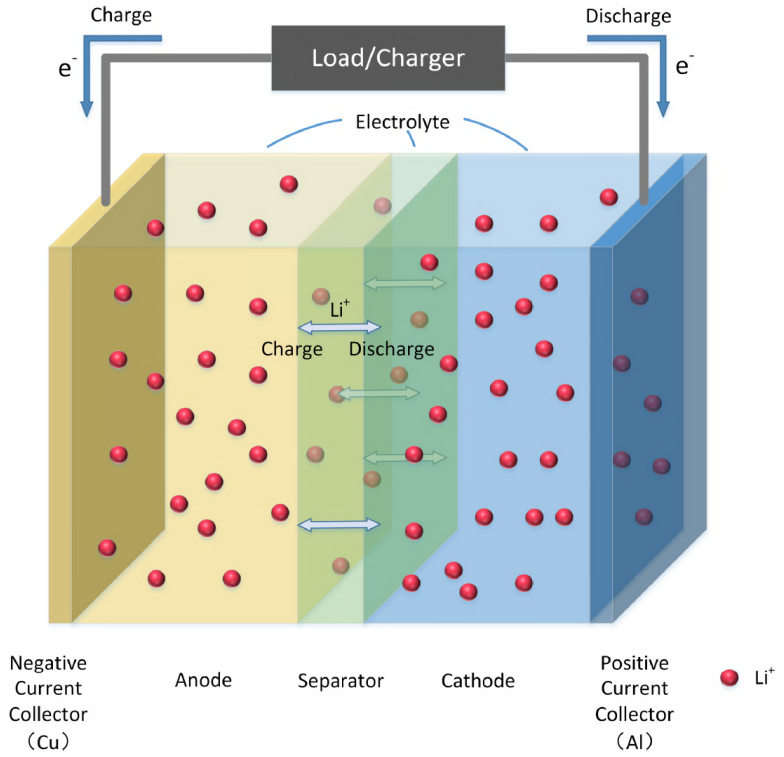


Figure 2.1: *Representation of the internal structure and functioning of a Li-ion cell [151].*

The electronic separation between the two electrodes is guaranteed by a micro-porous separator soaked in an electrolyte in order to allow the ions exchange during the cell charge/discharge. In particular, lithium ions go from cathode toward the anode during the charging phase and vice versa during the discharge. The ions flow causes a current of electrons collected by two current collectors, one made of copper for the anode and one of aluminium for the cathode [29].

Table 2.1: *Lithium-ion chemistries and their properties [55, 137].*

Abbrev	**LCO**	**NCA**	**NMC**	**LMO**	**LFP**	**LTO**
Name	Lithium cobalt oxide	Lithium nickel cobalt aluminum oxide	Lithium nickel manganese cobalt oxide	Lithium manganese spinel	Lithium iron phosphate	Lithium titanate
Positive electrode	$LiCoO_2$	$Li(Ni_0,85Co_0,1Al_0,05)O_2$	$Li(Ni_0,33Mn_0,33Co_0 33)O_2$	$LiMn_2O_4$	$LiFePO_4$	LMO NCA
Negative electrode	Graphite	Graphite	Graphite	Graphite	Graphite	$Li4Ti_5O_{12}$
v_n (V)	3.7-3.9	3.65	3.8-4.0	4.0	3.3	2.3-2.5
Energy density (Wh/kg)	200	250	180	140	120	80
Power	normal	high	normal	normal	normal	very high
Safety	low	normal	normal	good	very good	very good
Lifetime	low	good	normal	normal	good	very good
Cost	high	high	high	normal	high	normal

Table 2.1 highlights the differences among the various chemistries that make them suitable for different applications. The lithium iron phosphate (LFP) cells are widely used in large format ESSs where a high energy density and a good safety are required. The lithium nickel manganese cobalt oxide (NMC) batteries are typically used in low and medium power devices thanks to their very good energy density which allows to build very small ESSs. The lithium titanate (LTO) are used in applications where a high maximum charge current is required [87]. The maximum current is the value that can flows through the cell without damages. It is usually expressed in C-rate, where a current of 1 C is the one that discharges a fully-charged cell in one hour. Therefore, it depends on the cell nominal capacity Q_{n}, *i.e.*, the charge that can be drained from a cell starting from a fully-charged state until to the fully-discharged one in determined operating conditions defined by the manufacturer [148]. Another important parameter is the nominal voltage v_{n}, *i.e.*, the voltage at the equilibrium between terminals when the cell has been charged to an average value, and it is inside the range from $2.3\,\mathrm{V}$ to $4.0\,\mathrm{V}$.

The cell voltage is related to the concentration of lithium particles that changes during the charging/discharging phase. For example, the NMC safe operating voltage is about in the range between $2.7\,\mathrm{V}$ and $4.2\,\mathrm{V}$ and if the cells work outside of this range, they can be subjected to serious damages such as thermal runaway or explosion, or to the degradation of capacity and performance. For this reason, these batteries must be equipped with a BMS that aims at monitoring the cell voltages, but also temperature and current, and to guarantee that all the cells in the battery pack work in their Safe Operating Area (SOA).

2.3 Battery management system

The BMS is an electronic circuit provided with opportune sensors and actuators which allow a control unit to monitor and manage the battery by using dedicated algorithms, in order to guarantee a safe and optimized operation [13]. This system consists of one or more electronic boards, as described in Section 2.4, that communicate with the central unit of the application to optimizing the battery functioning. To perform the mentioned tasks a BMS usually may contain the following features [115]:

- Cell monitoring: measurement of the voltage, current, and temperature of the cells with a suitable accuracy;
- State estimation: estimation of the internal state of the battery, such as the SOC, SOH, Remaining Useful Life (RUL), and application runtime;
- Safety: protection from over-charge, over-discharge, and short circuit;
- Charge equalization: reduction of the energy mismatch among the cells composing the battery pack;
- Communication and data management: integration of the ESS in the application system by using affordable communication links in order to exchange important information, battery data that are periodically stored in a local memory, and commands.

These features have been described in details in the following Sections.

2.3.1 Cell monitoring

The BMS can execute its tasks by using as inputs the physical quantities directly measurable from the cells. As said above, these quantities are the voltages between the cell terminals, the current, and the temperature.

In particular, voltage can be measured with custom solutions that employ an Analog-to-Digital Converter (ADC), especially when the battery contains few cells, or using dedicated integrated circuits, called stack monitors. A stack monitor allows to measure a certain number of cell voltages with an accuracy that can reach a value of $\pm 1\,\mathrm{mV}$ in some conditions. More details about stack monitors have been presented in Section 2.4.

Current is usually acquired using two types of sensor: shunt-resistor or Hall effect based sensors. In the first case, the output of the sensor is the voltage across the shunt-resistor due to the current to be measured, while in the second one the sensor output voltage depends on the variations of a magnetic field caused by the current flow. The required accuracy of current measurement depends on the application. For example in automotive applications, where current can reach hundreds of amps, the target value can go from $0.5\,\mathrm{up}$ to $1\,\%$ [115].

BMSs usually incorporate various temperature measurement points. Different types of sensors can be used to measure temperature, but the most widespread in embedded systems for industry applications, including automotive field, are thermistors, characterised by a resistance value which changes with temperature.

These quantities can be used by the controller of the BMS to perform the estimation algorithms. In this case, the accuracy of the measurements affects the accuracy of the estimation and each measurement must be acquired simultaneously.

2.3.2 State estimation

The internal state of the cells which compose the battery is described by state variables. These are computed by the controller of the BMS by using estimation algorithms and the quantities measurable directly on the cells. The two most important state variables are SOC and SOH. SOC is related to the charge stored into the battery, while SOH is an indicator of the battery ageing. Both variables are used by the BMS to execute all the managing tasks, such as balancing and safety functions. Furthermore, they are inputs for the algorithms that compute other very useful information related to the application, such as the power that the battery can provide for a time interval without exceeding the determined SOC and voltage limits (State Of Power, SOP) [143], the remaining time in which the battery is estimated to be able to work in accordance with some specification (RUL), and the State Of Energy (SOE) used to determine the runtime or the remaining driving range for an EV.

State Of Charge

The SOC is usually defined as the quantity of charge that can be extracted from a cell, normalized to its nominal capacity [104, 155]. The BMS computes the SOC for all the cells contained in the pack in order to define the extractable charge of the battery. In literature, there are different kinds of algorithms for SOC estimation [88, 119, 155].

The discharge test method consists in a complete discharge of the cell in controlled conditions and with a constant current. The obtained results are very accurate, but it is time consuming and can be executed only offline in laboratory (not during the battery functioning on the application), because of the required conditions that are very restrictive.

The Coulomb Counting (CC) is the most simple and diffused technique, since it is easy to implement and requires a low computational power. It is based on the SOC definition, represented in Eq. (2.1), where the SOC is computed starting from its initial value SOC_0 (the SOC at time $t = 0$) and integrating the cell current i_L over time, normalized with respect to the nominal capacity [82, 92].

$$\mathrm{SOC}(t) = \mathrm{SOC}_0 - \frac{1}{Q_\mathrm{n}} \int_0^t i_\mathrm{L}(\tau) d\tau \tag{2.1}$$

The obtained results can be very precise if the current is acquired with a high accuracy [146]. In fact, errors like an offset on the sensor acquisition can cause the divergence of the integration that can be restored only with a periodic calibration of the calculation during a known state of the battery. However, this operation is not possible in some applications. Moreover, the used values of SOC_0 and Q_n also affect the accuracy and these quantities are not easy to estimate, since they depend on the battery ageing.

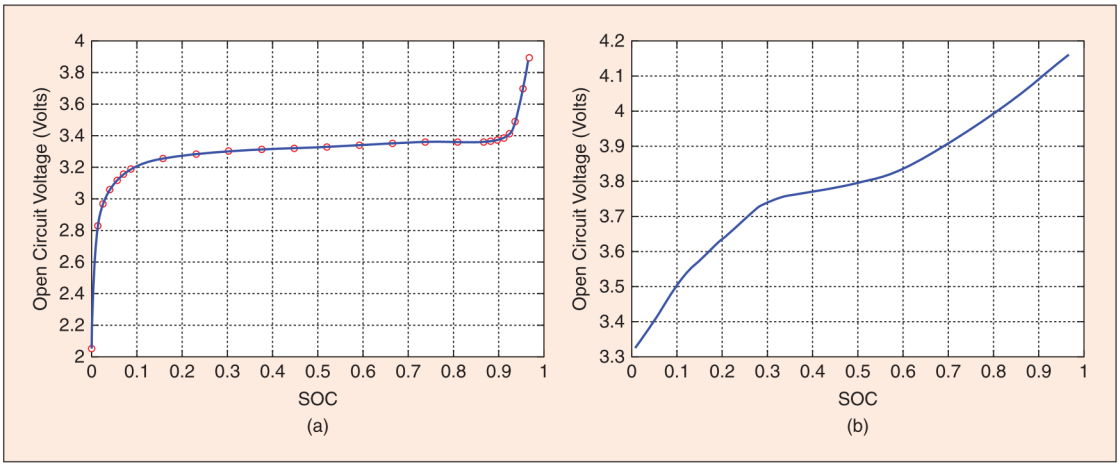


Figure 2.2: *OCV–SOC curves of two different battery chemistries [115]. (a) LFP battery OCV–SOC. (b) NMC battery OCV–SOC.*

Another technique easy to implement and that provides very accurate results is the Open Circuit Voltage (OCV or V_OC) method [55]. It is based on the measurement of the cell OCV that is mainly related to the SOC [102]. The OCV-SOC relationship is a non-linear function that slightly changes with respect to the battery temperature, ageing and manufacturing process tolerances [102]. Moreover, it changes its shape with the battery chemistry [44]. For example, the curve for LFP batteries is flatter than NMC chemistry between 20 and $80\,\%$ of the SOC, which is the typical operating range of the batteries, as shown in Figure 2.2. This fact brings to a higher error in the inversion of the OCV-SOC relationship during the computation of the SOC value. Another drawback of this technique is the time that the cell employs to reach the equilibrium condition when

the current goes to zero [154]. In fact, this time can be very long, making not possible the usage of this technique in applications where the load current has a high dynamic. Furthermore, for some chemistries the OCV measurement must take in consideration also that the characteristic can present the hysteresis phenomena, *i.e.*, the OCV-SOC curve is different between the charge and discharge phases [51, 120].

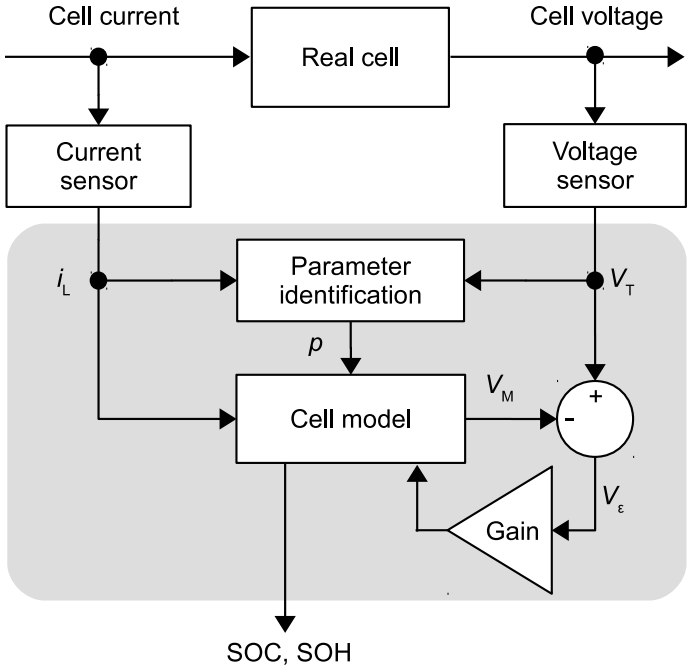


Figure 2.3: *Functioning diagram of the model-based algorithms.*

The most promising solutions that overcome the problems of these techniques are the model-based algorithms. Their functional block diagram is shown in Figure 2.3. The basic idea is to relate the SOC estimate to the measured battery voltage by means of a battery cell model. In fact, they operate in closed-loop where the error signal v_ε is generated, by comparing the cell voltage v_{M} predicted by the model to the measured value, and used to correct the estimate of the state variables. In this way, the estimation error due to inaccuracies in the measured signals is reduced. Examples of model-based algorithms are the artificial neural networks and fuzzy logic approaches, but they usually require high performance because of the use of complex models and require a long time and a large amount of data to perform the training procedure [68, 128, 149]. Another widespread model-based technique is the Kalman Filter, especially in its extension for non-linear systems. It is called EKF and is adopted in [91,94,99,108], while its adaptive version, called AEKF, in [34, 56, 145]. To overcome the limitations of the first order Taylor series linearisation applied in the EKF, derivativeless filters known as Sigma Point Kalman Filter (SPKF) have been considered for SOC estimation. In particular, the use of the Unscented Kalman Filter (UKF) is presented in [5, 59, 152], while the adaptive UKF (AUKF) is implemented in [101] and a Log-normalized UKF (Ln-UKF) in [9]. In [110, 111], the UKF is compared with their square root form, called SR-UKF or SR-SPKF. Another adaptive filter is the H_∞ (H-infinity) filter used, for instance, in [153] and the Particle Filter (PF) [118, 155]. Other solutions employ a

compensator or an observer to correct the SOC computation done in the model using the CC method [35, 73]. In [89, 91, 92], the compensator gain of the Mix Algorithm is designed to counteract the offset affecting the battery current reading, while an optimal observer based on the linear quadratic (LQ) approach is adopted in [114]. In [96], the authors employed instead a Sliding Mode Observer (SMO).

An important factor that determines the performance of the model-based algorithms is the capability of the model to reproduce the cell behaviour. Many models with different characteristics have been presented in the literature [116]. Usually, the more the model is complex, the more it is accurate but difficult to implement in an embedded system like the BMS. These models must take into account also that the model parameters are not fixed and change with the operating conditions, *i.e.*, current-rate, SOC, temperature and ageing of the battery. These parameters can be stored in Look-Up Tables (LUTs) with their dependency on the cell operating point. This solution requires low computational resources but also a very extensive and time consuming offline characterization [12]. Furthermore, it is very difficult to include also the parameter variations due to manufacturing process tolerances and ageing of the battery. These problems are overcome by employing an online identification technique in order to track the model parameter variations with the operating point. This approach leads to a problem where the states and the parameters are jointly estimated, which is solved with an appropriate algorithm, like adaptive filters, such as the Kalman [9, 62, 94] and the H_∞ [153] filters, alone or in combination with a least squares technique [35, 152].

Table 2.2: *Comparison of the SOC estimation methods [55, 82].*

Method	Applications	Inputs	Advantages	Disadvantages
Discharge test	Measure of the cell capacity	Remaining charge	Accurate, easy	Long time needed, offline, energy loss
CC	Any kind of systems	Current, capacity, initial SOC	Accurate, easy	Needs high accurate measurement of the current. Depends on SOC_0. Periodic calibration.
OCV method	Systems that allow to have long rest periods	Rest time, voltage	Accurate, easy, very low resource usage	Needs long periods with zero current. Requires very accurate measurement for some kinds of batteries.
Model-based	Any kind of systems	Current, voltage. Battery model.	Accurate, insensitive of the SOC_0 value, suitable for high dynamics systems. Some of this algorithms also perform the model parameter identification.	Sensitive to the measure noise, not easy to implement. Some of this algorithms require a training procedure.

The different SOC estimation algorithms are compared in Table 2.2 [55, 82]. Another solution for SOC estimation is to combine two or three algorithms to create hybrid methods that exploit their advantages. For example, the OCV method can be used to reset the integration in the CC technique and to find the value of the initial SOC [67]. This combination is used together with the KF in [30], where the KF is used to reduce

the error in SOC estimation.

State Of Health

This variable is an indicator of the Li-ion cell ageing with respect to its fresh state. Both industry and research fields do not give a unique and universal definition of SOH. However, the two most used definitions are based on the cell capacity and internal resistance [85, 95, 109]. For this reason, the estimation of the battery SOH can be reduced to the identification of the battery capacity and/or impedance depending on the employed definition [138].

The SOH is defined as the ratio between the actual cell capacity and the capacity in the fresh state ($Q_{\mathrm{n,new}}$), if the cell capacity fading is taken into consideration, as shown in Eq. (2.2) [72].

$$\mathrm{SOH}(t) = \frac{Q_{\mathrm{n}}(t)}{Q_{\mathrm{n,new}}} \tag{2.2}$$

The cell is considered in a healthy state if the actual capacity has a value greater than a determined percentage of the capacity in the new state. For example, in automotive applications batteries are used until their SOH reaches a value of $80\,\%$ [156]. However, the retired batteries can be employed in other applications such as in stationary field for energy storage and power backup. The second life batteries must be composed by cells with a homogeneous SOH level to guarantee the best performance [86]. For this reason, the state estimation of each cell in the battery pack must be performed with an appropriate accuracy.

The estimation of the cell internal resistance R_0 is necessary to relate the cell SOH to the performance degradation because of the resistance increasing with ageing. In this case SOH is defined as in Eq. (2.3), where $R_{0,\mathrm{new}}$ is the internal resistance in the fresh state and $R_{0,\mathrm{EOL}}$ is considered the resistance at the end of cell life [37].

$$\mathrm{SOH}(t) = \frac{R_{0,\mathrm{EOL}} - R_0(t)}{R_{0,\mathrm{EOL}} - R_{0,\mathrm{new}}} \tag{2.3}$$

In this case, in the healthy state the cell is able to provide the power demanded by the application. Therefore, the application defines the minimum power that the battery has to guarantee and thus the value of $R_{0,\mathrm{EOL}}$.

Cell Models

Models capable of reproducing the Li-ion cells behaviour are very important in estimation algorithms, but also during the offline characterization of a cell or their construction. Various kinds of models have been presented in literature [28, 67], which can be classified as follows:

- Electrochemical models: they describe the mechanisms of power generation by using equations that connect the macroscopic information (*e.g.*, voltage and current) with the microscopic ones (*e.g.*, the electrons exchange during the electrochemical reactions and the ionic concentrations) [100, 122, 147]. The model parameters are hardly identifiable because they require a deep knowledge of the cell structure.

- Mathematical models: in these models the cell behaviour is computed with empirical equations that are usually designed for a specific battery under particular conditions and applications [26,107]. Sometimes there is not correspondence with the physical phenomena.

- Electrical models: they reproduce the cell by using an electric equivalent model with lumped parameters [76, 95].

Highly accurate models, such as the single-particle and multiphysics electrochemical models, tend to be very complex and thus suitable for the offline analysis of the behaviour of a cell [49]. In contrast, Electrical Circuit Models (ECMs) are suitable to be implemented in a real-time embedded system, like a BMS based on a micro controller [57, 94] or a low-cost FPGA device [92]. They can provide good accuracy with affordable complexity [63] by using various combinations of passive components, like resistors and capacitors, and controlled voltage generators to mimic the static and dynamic behaviour of the battery cells with different degrees of fidelity [95].

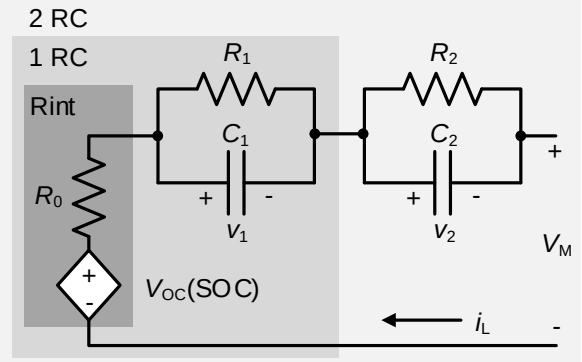


Figure 2.4: *Rint model and Thevenin models with one and two RC branches.*

Very simple ECM configurations are the combined model [153] and the internal resistance or Rint model [58,59]. As shown in Figure 2.4, the Rint model describes the cell behaviour by using just a voltage source OCV and a series resistor R_0. The value of the open circuit voltage OCV depends on SOC. This relationship can be modelled in different ways, for example by using a LUT [12]. The combined model is a particular type of the Rint model, which uses a combination of the Sherpherd model, Unnewehr and Nasar model and the Nernst model to describe the behaviour of OCV:

$$\begin{aligned} v_{\mathrm{M}} =& K_0 - \frac{K_1}{\mathrm{SOC}} - K_2\mathrm{SOC} + K_3 \ln(\mathrm{SOC}) + \\ & K_4 \ln(1 - \mathrm{SOC}) - i_{\mathrm{L}} R_0 \end{aligned} \tag{2.4}$$

where K_0, K_1, K_2, K_0 and K_4 are fitting constants. The SOC is an internal state variable of ECM models, evaluated using the CC method as in Eq. (2.1).

To account for the dynamic behaviour of a lithium-ion cell, one or more RC branches can be added to the Rint model to consider the charge transfer, the double-layer, and the charge diffusion phenomena. This leads to the model with one or two RC branches

shown in Figure 2.4, often referred to as Thevenin model [23,63], which yield different degrees of accuracy and computational complexity.

Thevenin model with one RC branch provides a good trade-off in many applications, especially when fast transients are relevant. It is indeed employed in the state estimation algorithms developed in [46, 56, 57, 99, 145]. A second RC branch can be used to increase the accuracy of the model by reproducing slower dynamics associated to charge diffusion into the electrodes [5, 34, 35, 58]. Voltages v_1 and v_2 on the RC branches model the relaxation effects [95]. A model obtained through the star-delta transformation of the Randles' circuit is used in [101].

The Enhanced Self-Correcting (ESC), which includes effects due to OCV, internal resistance, voltage time constants, and hysteresis has been proposed in [108, 110, 111]. A first-order differential equation is used to describe the hysteresis effects.

The model parameters can be extracted by characterizing the cell in different operating conditions. For each of these conditions, a particular current profile is applied to the cell, while the characterization setup acquires and store the voltage, current, and temperature of the cell. Then, these data are used to identify the model parameters. The Pulsed Current Test (PCT) is used at different temperatures and current C-rate in [11, 12]. In this test the temperature is kept to a constant value by using a thermostatic chamber and the current profile is made of pulses of given duration followed by a rest time [64], as shown in Figure 2.5. The voltage and current profiles are acquired with high accuracy sensors, in order to allow a reliable characterization and SOC computation, by using the CC technique.

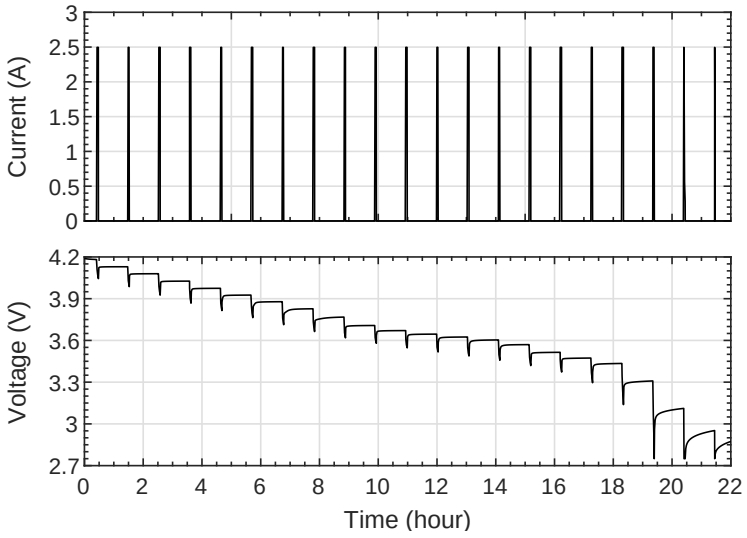


Figure 2.5: *Current and voltage profiles during a pulsed current discharge test.*

The pulse duration allows the system to discharge the cell of a precise SOC value and so to obtain the parameter value for different SOC points by analysing the cell behaviour during the pause. The voltage response during a pause, after a discharging current pulse have been applied, is reported in Figure 2.6. The voltage raise ΔV_0 at the begin of the pause is due to the cell ohmic resistance, while the other model parameters can be extracted by executing an opportune exponential fitting of the voltage during the raise indicated with ΔV_1.

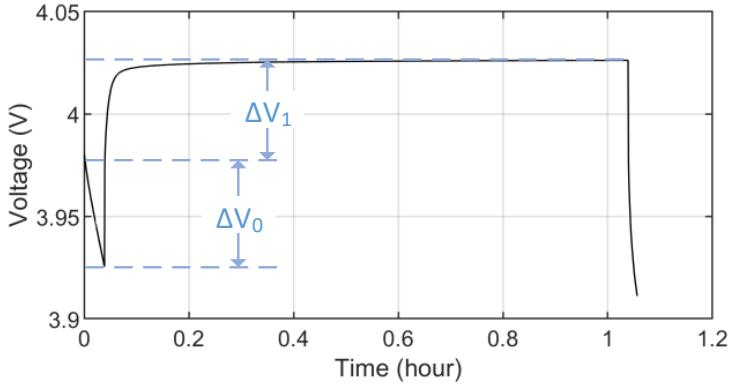


Figure 2.6: *Voltage response during a pause, after a discharging current pulse.*

In order to give an indication of the dependency of the ECM parameters on the operating point of the cell, the behaviour of R_0 as function of SOC and temperature is shown in Figure 2.7 for a Kokam SLPB723870H4 $1.5\,\mathrm{A\,h}$ NMC cell [12]. We can note that the value of this resistance noticeably changes when SOC varies from $20\,\%$ to $100\,\%$ and the temperature from $10\,^{\circ}\mathrm{C}$ to $35\,^{\circ}\mathrm{C}$. This shows that ECMs can provide good accuracy only if their parameters are made vary with the battery operating conditions. Thus, ECMs need to be combined with an online parameter identification technique.

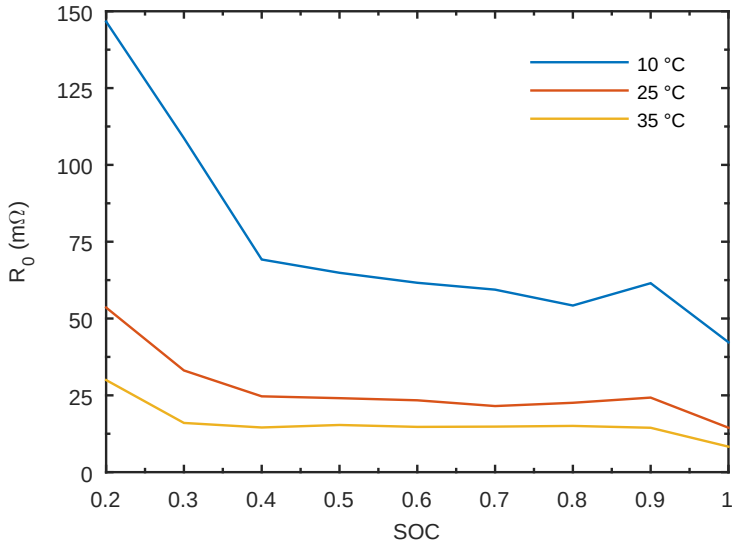


Figure 2.7: *Behaviour of R_0 as function of SOC measured at three temperatures on a Kokam SLPB723870H4* $1.5\,\mathrm{A\,h}$ *NMC cell.*

2.3.3 Safety

One of the main function of a BMS is to protect the battery from dangerous operating conditions due to its chemical characteristics that can be hazardous also for the users. These conditions may occur when a cell works outside of its SOA in terms of voltage and SOC (over-charged or discharged conditions), temperature, and current, when it is

higher than the maximum charge/discharge values. Usually, batteries are provided with a power switch, used by the BMS to protect the cells.

Safety must be guaranteed during the entire battery lifetime, so also the hardware and software components of the BMS must be very reliable and need specific safety features. For example, automotive applications require compliance with the International Organization for Standardization (ISO) 26262 functional safety standard.

2.3.4 Charge equalization

As said in Section 2.2, cells are usually connected in series and in parallel in order to increment the battery voltage and capacity, respectively. This solution may introduce different issues when the used cells have some characteristics that differ from their nominal values. In particular, in series-connected cells, the ageing, the internal resistance mismatch, and the temperature gradient can cause the unbalancing of the stored energy [18, 115]. In this situation, the charge and discharge phases can be executed until one of the cells in the series reaches the upper and the lower voltage threshold, respectively. This does not allow the full charge or discharge of the battery, resulting in a reduction of the energy that can be stored in the battery pack. Various solutions have been implemented on BMSs to equalize the energy among cells and to make their SOC as close as possible. They can be divided in two big categories: the passive and the active balancing [25, 77].

Passive technique equalizes the stored energy by dissipating it on bleeding resistors that are connected to the most charged cells using dedicated switches [8, 157]. This solution is usually adopted in small/medium batteries because of its low cost and simple implementation especially when the currents that flow into the resistors are low. A more complex thermal management is needed for high balancing currents, since the energy is dissipated as heat directly on the BMS board.

On the other side, active balancing techniques allow to improve the equalization efficiency since the energy is moved from the most charged cells to the others [50]. However, the circuital complexity is higher with respect to the passive balancing, and the challenge is to find a good trade-off between complexity and efficiency. Promising solutions are those where energy transfer is executed in a module-to-cell or a battery-to-cell configuration by employing a DC/DC converter [14, 25, 112].

2.3.5 Communication and data management

An important feature of the BMS is the managing and the storage of the measured and computed data, such as cells voltage, current, SOC, SOH, and fault codes [55, 74]. This task can be very difficult, especially in batteries with a large number of cells, because of the large amount of data that must be exchanged and the large amount of memory needed to store these information. These data are then used by the other components of the system to analyse the battery state and to perform the standard tasks. In fact, BMS must guarantee the integration with the internal modules of the application, such as the motor control and the central unit of an EV. Furthermore, it must be able to communicate with external components, such as a charging station to allow the data and commands exchange and the managing of the charging phase. In automotive and industrial applications, the most used communication protocol is the Controller Area

Network (CAN bus) because of its robustness [22, 74]. Other diffused links are the UART, the SPI, and the Ethernet.

2.4 BMS architectures

In battery-powered applications, the various architectures of a BMS mainly differ on the distribution of the BMS functions among different hierarchical levels, and the choice depends on the number of cells composing the battery pack. Each topology has different:

- Complexity: number of boards and components, communication and power cabling, software;
- Flexibility: possibility to easily reconfigure the BMS for different batteries and applications;
- Robustness: ability to cope with failures during the battery functioning. These failures can occur due to issues on the Li-ion cells, on the BMS boards, or on the communication link;
- Computational power: capability of executing algorithms for accurate and reliable cells state estimation;
- Cost: for design, manufacturing, and maintenance.

The most common typologies are usually composed by one or two hierarchical levels. A monolithic architecture is an example of the first typology where all the functions are developed in a single board, while BMSs based on two levels have a distributed architecture, whose most lower level comprises boards that are directly connected to the cells and that can be organised in modules usually managed by a stack monitor or can use cells with some logic inside.

As said in Sections 2.3, stack monitors are dedicated integrated circuits able to monitor a group of cells. They are provided with a series of inputs dedicated to the cell voltages measurement to which each cell terminal must be connected. The acquired data are sent to a control unit through a serial interface and used to execute the BMS algorithms and procedures. Moreover, they can use these information also to alert the main unit about some dangerous situations, such as the detection of under-voltage or over-voltage conditions, by using dedicated error signals. They can also implement other useful functions such as the acquisition of a certain number of analog inputs that can be used to acquire other cell quantities (*e.g.*, the temperature and current), and the control of some digital outputs. Furthermore, they can be able to manage the balancing circuit, which is usually implemented with a passive technique. The main functions and a comparison among the most used stack monitors in research and commercial fields are reported in Table 2.3. Each stack monitor can manage a number of cells belonging to a determined range, where the low limit is due to minimum allowed supply voltage of the chip, since it is directly powered by the cells, while the high limit is due to the number of measurement inputs.

Nevertheless, a battery composed by a large number of cells can be managed by connecting together more stack monitors, since they can be connected in daisy-chain

Table 2.3: *Characteristics comparison of commercial stack monitors.*

Name	BQ76PL455A-Q1	LTC6813-1	ISL78610
Manufacturer	Texas Instruments	Analog Devices	Renesas
Number of cells	from 6 to 16	up to 18	up to 12
ICs daisy-chain	up to 16	up to 17	up to 14
Additional inputs	8 analog + 6 GPIOs	9 GPIOs/analog	4 analog
ADC	14 bit	3 x 16 bit	13 bit
Measurement time for all cells	$2.40\,\mathrm{ms}$	$0.29\,\mathrm{ms}$	$0.24\,\mathrm{ms}$
Accuracy (depends on the ADC configuration and temperature range)	from $\pm 1\,\mathrm{mV}$ to $\pm 7\,\mathrm{mV}$	from $\pm 2\,\mathrm{mV}$ to $\pm 6\,\mathrm{mV}$	$\pm 10\,\mathrm{mV}$
Communication with host	UART ($1\,\mathrm{Mb/s}$)	SPI ($1\,\mathrm{Mb/s}$)	SPI ($2\,\mathrm{Mb/s}$)
Communication through daisy-chain	proprietary differential	isoSPI	proprietary differential
Balancing	Passive	Passive	Passive
Compliance	AEC-Q100	ISO 26262	AEC-Q100
Cost	12€	13€	8€

to allow the control unit to communicate with each single chip. As we can see in Table 2.3, there is also a maximum number of chips that can be connected in daisy-chain because of the limits in the communication. However, the daisy-chain connection of a large number of stack monitors can bring to a problem of data synchronization since messages like the measurement trigger message, sent by the main unit to the nearest stack monitor in the communication bus, can be propagated with a significant delay, and thus the acquisition of the cell voltages is executed in different moments with respect to the measurement of the battery current. The consequence of this problem can be an error in the state estimation algorithms which assume a perfect correspondence between the input samples. To avoid this problem, some new stack monitors, such as the BQ76PL455A-Q1 by TI shown in Table 2.3, allow the broadcast communication of the commands sent by the main unit.

The BMS is a monolithic system in small-format batteries, where all the functions are grouped in a single board in order to reduce costs, that are very important in these cases, since these kind of ESS usually equips low-cost systems [135]. This kind of architecture is used in [8, 78, 157] where the described BMSs manage a low number of cells (from three to seven cells). The main unit is based on microcontrollers that acquires the voltage and the current of the cells with the integrated ADC and uses them to perform the inevitable safety functions and the basic estimation of the SOC with the CC technique. A monolithic architecture for medium-format batteries able to manage 12 series-connected groups of two parallel-connected cells is presented in [41]. In this case, the microcontroller acquires the cell voltages and some point of temperature inside the battery pack by using a stack monitor. In general, balancing techniques are absent or based on very simple passive circuits. These architectures have low complexity and present a low flexibility, since they are built for a defined number of cells to minimize their costs. Furthermore, they introduce a single point of failure since only one central control unit is present.

Different considerations must be done for large-format batteries, where the high cost is due to the large number of lithium-ion cells, to the cabling, and to the large

number of critical components necessary to manage the high required power such as contactors, current sensors, balancing circuits, appropriate enclosures, and isolation systems. In these batteries, the BMS is physically organized in two or more hierarchical levels where the cells are usually grouped in modules provided with a Module Management Unit (MMU) which communicates with a central unit, the Pack Management Unit (PMU). Here, the functions of the BMS can be distributed among the different hierarchical levels, starting from MMUs able to measure only the main cell quantities that are elaborated by the PMU (Figure 2.8.(a)) and arriving to intelligent MMUs (Figure 2.8.(b)) able also to execute different algorithms [135].

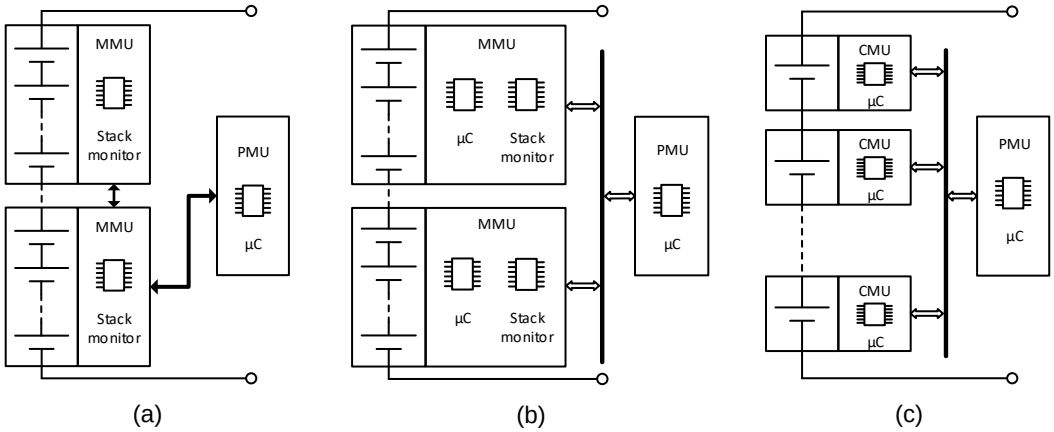


Figure 2.8: *Distributed topologies based on (a) MMUs with a stack monitor, (b) MMUs with a stack monitor and a microcontroller, and (c) Smart Cells.*

In the first case, MMUs can be based on stack monitors able to execute the measurements and to perform a reliable communication with the PMU. The latter needs a high computational power since it has to compute the state of all the battery cells and for this reason it must be based on a microcontroller with a suitable computational power, like in [53].

In the case of Figure 2.8.(b), some of the PMU functions can be demanded to the MMUs in order to distribute the tasks. Here, all the management units can be built using microcontrollers with an appropriate computational power. In [80, 105, 150], the Analog Devices stack monitors or the BQ76PL455A-Q1 by Texas Instruments are used together to a basic microcontroller that has the task to collect the acquired data and to send them to the PMU over CAN communication, while in [22] the MMUs also manage the balancing procedure. A different approach is used in [40, 71], where the MMUs are provided with a more powerful microcontroller, like the LPC1754 by NXP, in order to perform most of the BMS tasks at module level. In fact, each MMU is able to monitor and manage the module cells and to compute their internal state, communicating the results to the PMU over CAN. The possibility to use the CAN communication instead of the daisy-chain connection improves the reliability of the communication and the synchronization of the data acquisition.

Furthermore, in literature there are many works which explore the idea of Smart

Cell (SC) [81, 121, 125, 126], where every single cell is provided with a Cell Management Unit (CMU), *i.e.*, a board with microcontroller able to execute some functions and to communicate with the PMU, as shown in Figure 2.8.(c). In these works, the basic functions of the SCs are the voltage and temperature measurement and the communication with the PMU, but additional tasks can be demanded to the CMU, like the current measurement, and the execution of the state estimation algorithms, reaching a considerable complexity at CMU level. This solution allows the PMU to be very simple and to perform only the data collecting, the communication with external systems, and the managing of the main switches.

Recent works have presented a new architectural approach: a distributed BMS with the main unit based on an FPGA or on a System on Chip in an FPGA device (SoC-FPGA), where the FPGA fabric is flanked to a hard-core processor or used to instantiate a soft-core processor. The use of these devices in industrial and transportation applications is continuously growing [134], such as in motor control [21, 47, 75], robotic arms control and precise positioning [38, 132], control of power quality [54, 141], and fault verification and railway safety [20, 52]. In some cases these devices are going to replace microcontrollers [21] thanks to a substantial reduction of costs and to the introduction of high level design flows for improving the time to market [123, 124], as the design flow introduced by Intel and described in Section 4.2.

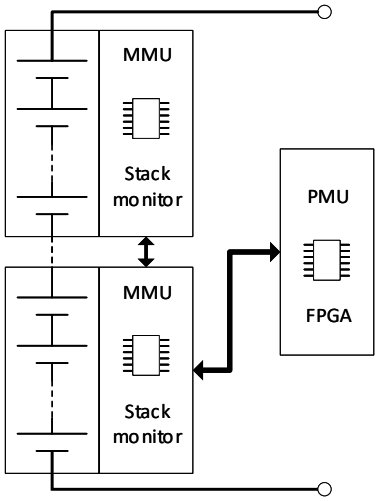


Figure 2.9: *Distributed architecture with an FPGA-based PMU and battery cells organised in modules.*

The use of an SoC-FPGA in a BMS allows us to obtain a more flexible system with a higher computational power. Two different architectures have been analysed by using an FPGA device as PMU control unit: a distributed topology where cells are organised in modules managed by a stack monitor (very similar to the architecture of Figure 2.8.(a)) and an architecture which uses the concept of smart cell (like the one of Figure 2.8.(c)).

A schematic representation of the distributed architecture which uses an FPGA-based PMU and stack monitors to manage group of cells is shown in Figure 2.9. The hard/soft-core processor and its peripherals can be used to implement the basic functions of the BMS, while the FPGA is used to instantiate state estimation hardware

co-processors, to extend the peripheral set of the hard-core processor, and to instantiate further soft-core processors that can be used for redundancy reasons in order to improve the BMS safety and robustness. A distributed architecture which jointly employs the concepts of the use of an SoC-FPGA and SCs is shown in Figure 2.10.

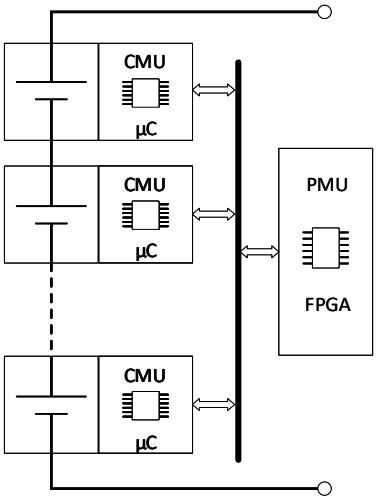


Figure 2.10: *Distributed architecture with an FPGA-based PMU and smart cells.*

2.4.1 Discussion

The choice of the BMS architecture to be used in a specific application depends on many factors. A comparison of the different architectures can be very useful to the designer in choosing the BMS architecture to be used for a specific battery, depending on particular requirements. A monolithic architecture is very convenient for small-format batteries (up to about 20 cells), while for large-format batteries, like those used in automotive applications, the most employed solution is the distributed topology.

The most simple and cost-effective architecture is the one with the MMUs with only a stack monitor and a microcontroller-based PMU. This because of the low circuital complexity dues to the low number of components, the low complexity of the communication cabling, and the necessity to develop the firmware only for one microcontroller. On the other hand, this architecture provides the worst performance in flexibility, computational power, and robustness. In general, architectures which use stack monitors instead of smart cells have a lower scalability with the number of cells since there are some restrictions on the maximum and the minimum number of cells that can be managed and on the number of stack monitors that can be connected in daisy-chain. Furthermore, it is convenient to use module with the same number of cells to avoid different unbalancing of the cells among modules. Another issue of this architecture is that it can only allow the execution of simple algorithms, since complex techniques executed for a large number of cells may require a long execution time if the computational power is not adequate. Better computational power is achieved by the other architectures thanks to the use of more computational units or an FPGA that enables the hardware acceleration, reaching higher performance thanks to parallel computing

and pipeline architectures. Furthermore, the use of various control units brings to more robust BMSs, thanks to the possibility to implement further safety and redundant functions that can allow the battery to keep working even in presence of failures, until the application reaches a safety situation. In this regard, the up-and-coming architectures which use SCs are the most well-performing and present a reduced sense cabling complexity, since each CMU in a smart cell can be directly connected to the cell terminals, but the high costs make it not suitable for many current applications. However, higher integration of some hardware components and higher volume of manufacturing can allow in future a reduction of the costs and so the use in different applications [81].

The use of an FPGA-based PMU offers a very good flexibility thanks to the high grade of reconfigurability of the device that also allows the developer to change its internal peripherals, enabling for example the use of MMUs with a different communication protocol or more instances of the same peripheral to parallelize the data acquisition or the execution of some functions. Moreover, another technique that can be employed to improve the estimation capability and the flexibility of this architecture is the dynamic reconfiguration of the FPGA during run-time as demonstrated in [140], in which various algorithms are executed in time division multiplexing from time to time.

Currently, the architecture whit an FPGA-based PMU and SCs has the advantages of both the SCs and the FPGA, but it is very expensive for industrial applications. However, it can be very useful for research reasons or during the development of new BMS functions and algorithms thanks to the available computational power and the highly flexible architecture. In fact, we developed a BMS based on this architecture within the 3Ccar European research project, which aims at investigating the use of this kind of advanced architecture for automotive applications.

CHAPTER 3

Advanced state estimation algorithms

3.1 Introduction

Regarding the SOC estimation, the advantages introduced by the model-based techniques are very significant, as previously described in Section 2.3.2. This because of their capability to match various computational and accuracy requirements. In fact, the accuracy of the estimation can be improved by using either a complex model or a simple one in combination with an online parameter estimation algorithm.

The brief state of the art description of the model-based algorithms reported in the previous chapter highlights that the most used technique for SOC and parameter estimation is the Kalman filter, used alone or in combination with a least squares technique. A detailed description of this co-estimation algorithm, especially in the dual extended version, is reported in Section 3.2, where either the generic functioning and the specific implementation for estimating the SOC and the model parameter of a Li-ion cell have been reported [91].

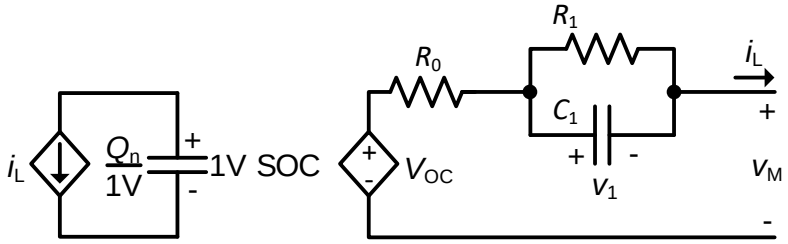


Figure 3.1: *Thevenin model with one RC branch.*

The Li-ion cells have been modelled by using the Thevenin model with one RC

branch introduced in Section 2.3.2 and shown in Figure 3.1, because of its suitability for application with fast transients, such as in automotive field. The left-hand side of the circuit models the cell capacity and the SOC. The latter is calculated as Q/Q_{n}, where Q is the residual charge. On the other side, the cell terminal voltage v_{M} is generated as the sum of the voltages v_1, the open-circuit voltage V_{OC} and the voltage across the ohmic resistance R_0, due to the flow of the cell current i_{L}. The state space equations of the model are reported in (3.1), where $\tau_1 = R_1 C_1$.

$$\begin{cases} \dfrac{dSOC}{dt} = -\dfrac{i_{\mathrm{L}}}{Q_{\mathrm{n}}} \\ \dfrac{dv_1}{dt} = -\dfrac{v_1}{\tau_1} + \dfrac{i_{\mathrm{L}}}{C_1} \\ v_{\mathrm{M}} = V_{\mathrm{OC}} - R_0 i_{\mathrm{L}} - v_1 \end{cases} \tag{3.1}$$

The DEKF presents some issues, such as the choice of some algorithm's parameters, that can be solved in the AMA to the detriment of a major complexity. This algorithm employs the same ECM shown above, as described in details in Section 3.3 together to a procedure to opportunely tune the algorithm parameters in relation with the target application.

3.2 Dual Extended Kalman Filter

The EKF estimates the internal state of a system by using recursive equations and the acquired measurements at each sampling time step. Let the state-space (or process) and the measurement discrete time equations for a time varying non-linear system be:

$$x_{k+1} = f(x_k, u_k) + \xi_k \tag{3.2}$$

$$y_k = g(x_k, u_k) + \psi_k \tag{3.3}$$

where subscript k indicates the discrete time, x_k is the state variable vector, u_k is the input vector, and y_k is the model output vector. Vectors ξ_k and ψ_k are the process noise and the measurement error assumed to be Gaussian random processes with zero mean and covariance matrices R and Q, respectively. Under these assumptions, if the system is linear the KF provides the optimal Bayesian solution for the unmeasured state x_k [70, 106]. However, results presented in literature confirmed that this estimator is very effective also for non-linear systems with non-Gaussian distributions [118].

In the EKF, the non-linear functions g and f are linearised by using the first-order Taylor expansion. This yields the following state space linearised representation:

$$x_{k+1} = A_k x_k + B_k u_k + \xi_k \tag{3.4}$$

$$y_k = H_k x_k + D_k u_k + \psi_k \tag{3.5}$$

where A_k, B_k, H_k and D_k are Jacobian matrices. The state vector is computed by cyclically repeating a prediction and a correction phase as shown in Figure 3.2, after having initialised the state vector and its covariance matrix P_k.

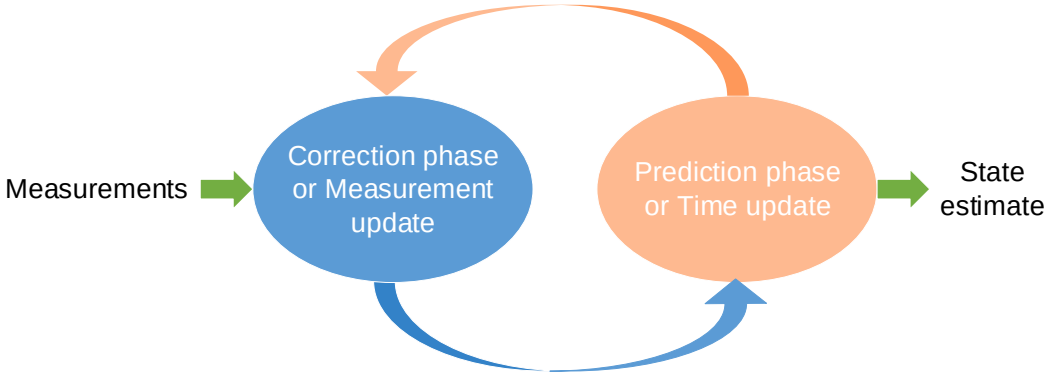


Figure 3.2: *Diagram of the EKF algorithm.*

During the prediction phase, the prediction of x_k and P_k, denoted by superscript "$-$", is computed:

$$x_k^- = A_{k-1}x_{k-1}^+ + B_{k-1}u_{k-1} \tag{3.6}$$

$$P_k^- = A_{k-1}P_{k-1}^+A_{k-1}^{\mathrm{T}} + Q \tag{3.7}$$

In the correction step, the prediction is corrected by using the Kalman gain L_k and the output measurement Y_k, obtaining the estimation of x_k and P_k, denoted by superscript "$+$":

$$y_k = H_k x_k^- + D_k u_k \tag{3.8}$$

$$L_k = P_k^- H_k^{\mathrm{T}}(H_k P_k^- H_k^{\mathrm{T}} + R)^{-1} \tag{3.9}$$

$$x_k^+ = x_k^- + L_k(Y_k - y_k) \tag{3.10}$$

$$P_k^+ = (I - L_k H_k)P_k^- \tag{3.11}$$

The EKF can be used to estimate both the cell state and the model parameters of a cell. This problem can be approached by using two different techniques: the Joint and the Dual estimation [109]. In both cases, u_k is composed only of the cell current i_{L}, while y_k of the model voltage v_{M}. In the Joint estimation, x_k comprises the state variables, such as SOC and v_1, and the parameters. Instead, in the DEKF there is a second filter which is used in cooperation with the first one for the parameter estimation. Usually, the Joint estimation provides more accurate results than the Dual one because it takes into account the correlation between the state and the parameters. It is also more intuitive to implement. Unfortunately, the use of a single state vector leads to larger matrices and to a poor numeric conditioning. These disadvantages are solved in part by using a Dual estimation algorithm, at the expense of a loss of accuracy, due to the neglection of the cross-correlation between state and parameters [110, 111]. The DEKF is built by using the following equations:

$$p_{k+1} = p_k + \chi_k \tag{3.12}$$

$$x_{k+1} = f(x_k, i_{\mathrm{L},k}, p_k) + \xi_k \tag{3.13}$$

$$v_{\mathrm{M},k} = g(x_k, i_{\mathrm{L},k}, p_k) + \psi_k \tag{3.14}$$

The state evolution is computed by equations (3.13) and (3.14), which build the first filter, while the model parameters, contained in vector p_k, are computed by the second filter, composed by Eq. (3.12), where χ_k is the noise on the parameters with zero mean and covariance matrix Q_{p}, and the same measurement equation. Since there is not a state-space description for the ECM parameters, the prediction phase just maintains the last computed value and the update of the parameters is executed in the correction step only. The full DEKF algorithms is as below:

1. Initialization:

$$x_0, P_0, p_0, P_{p_0} \tag{3.15}$$

2. Prediction step:

$$p_k^- = p_{k-1}^+ \tag{3.16}$$
$$P_{p_k}^- = P_{p_{k-1}}^+ + Q_{\mathrm{p}} \tag{3.17}$$
$$x_k^- = f(x_{k-1}^+, u_{k-1}, p_{k-1}^+) \tag{3.18}$$
$$P_k^- = A_k P_{k-1}^+ A_k^T + Q \tag{3.19}$$

3. Correction step:

$$L_k = P_k^- C_k^T (C_k P_k^- C_k^T + R)^{-1} \tag{3.20}$$
$$x_k^+ = x_k^- + L_k(y_k - g(x_k^-, u_k, p_{k-1}^+)) \tag{3.21}$$
$$P_k^+ = (I - L_k C_k) P_k^- \tag{3.22}$$
$$L_{p_k} = P_{p_k}^- C_{p_k}^T (C_{p_k} P_{p_k}^- C_{p_k}^T + R)^{-1} \tag{3.23}$$
$$p_k^+ = p_k^- + L_{p_k}(y_k - g(x_k^-, u_k, p_{k-1}^+)) \tag{3.24}$$
$$P_{p_k}^+ = (I - L_{p_k} C_{p_k}) P_{p_k}^- \tag{3.25}$$

where the subscript p indicates that variables are related to the KF which computes the model parameters. The value of the matrices used in these equations can be derived from the discrete time state space equations of the model [131], starting from (3.1):

$$\begin{aligned} x_k =& f(SOC_k, v_{1,k}, i_{\mathrm{L},k}) = \\ \begin{bmatrix} SOC_k \\ v_{1,k} \end{bmatrix} =& \begin{bmatrix} SOC_k - \frac{T}{Q_{\mathrm{n}}} i_{\mathrm{L},k} \\ v_{1,k-1} e^{-T/\tau_{1,k}} + R_1(1 - e^{-T/\tau_{1,k}}) i_{\mathrm{L},k} \end{bmatrix} \end{aligned} \tag{3.26}$$

$$v_{\mathrm{M},k} = g(SOC_k, v_{1,k}, i_{\mathrm{L},k}) = V_{\mathrm{OC}}(SOC_k) - R_{0,k} i_{\mathrm{L},k} - v_{1,k} \tag{3.27}$$

where T is the sampling time and the OCV-SOC relationship has been mapped by using a degree seven polynomial function, which allows us to obtain a good trade-off between complexity and fitting accuracy:

$$\begin{aligned} V_{\mathrm{OC}}(SOC) =& P_0 + P_1 SOC + P_2 SOC^2 + P_3 SOC^3 + P_4 SOC^4 + \\ & P_5 SOC^5 + P_6 SOC^6 + P_7 SOC^7 \end{aligned} \tag{3.28}$$

From (3.26) and (3.28) we obtain:

$$x_k = [SOC_k; v_{1,k}] \tag{3.29}$$

$$p_k = [R_{0,k}; 1/\tau_{1,k}; R_{1,k}] \tag{3.30}$$

$$A_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{-T/\tau_{1,k}} \end{bmatrix} \tag{3.31}$$

$$C_k = \begin{bmatrix} \frac{dV_{\mathrm{OC}}(SOC)}{dSOC} & -1 \end{bmatrix} \tag{3.32}$$

$$C_{pk} = \begin{bmatrix} i_{\mathrm{L},k} & 0 & 0 \end{bmatrix} + C_k \frac{dx_k^-}{dp} \tag{3.33}$$

where $R \in \mathbb{R}^{1\times1}$, $Q, P \in \mathbb{R}^{2\times2}$, $Q_p, P_p \in \mathbb{R}^{3\times3}$, since they depend on the number of the system outputs, states and parameters of the model, respectively, and:

$$\frac{dx_k^-}{dp} = A_k \frac{dx_{k-1}^+}{dp} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & Te^{-T/\tau_{1,k}}(R_1 i_{\mathrm{L},k} - v_{1,k-1}) & 1 - \tau_{1,k} i_{\mathrm{L},k} \end{bmatrix} \tag{3.34}$$

The DEKF can provide very good results, but the determination of the covariance matrices to be used in the algorithm is very tricky and it is usually performed by using an empirical approach [131]. Their values strongly depend on the application and can influence the numerical stability of the filter and the convergence rate towards the true value of the estimate [9, 24, 139]. They are diagonal matrices, where the *j*-th diagonal element mainly characterises the filter estimation of the state variable or parameter reported in the *j*-th position of the state or parameter vector, respectively. As largely proved in literature, higher values of the covariance usually allow a faster response of the filter. However, the performance improvement is not significant after a certain covariance limit and the filter can become not stable if the covariance value is too high [103]. The covariance values contained in the matrices Q and Q_p used in this work have been reported in Table 3.1, while R matrix is set equal to $3 \cdot 10^{-3}$ V^2.

Table 3.1: *Values of the covariance matrices of the DEKF algorithm.*

State variable/Parameter	SOC	v_1 (V^2)	R_0 (Ω^2)	τ_1 (s^2)	R_1 (Ω^2)
Variance	$8 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	$1 \cdot 10^{-10}$	$4 \cdot 10^{-6}$	$7 \cdot 10^{-10}$

The approach followed in this work is to execute the DEKF algorithm offline on an emulated profile, usually similar to a real profile typical of the considered application, tuning the covariance values in relation to the filter response and the considerations reported above.

3.3 Adaptive Mix Algorithm

The AMA block diagram is shown in Figure 3.3. It is an enhanced version of the Mix Algorithm (blue box), a model-based technique presented by Codeca et al. in 2008 [32], in which the ECM of Figure 3.1 is used with constant parameters.

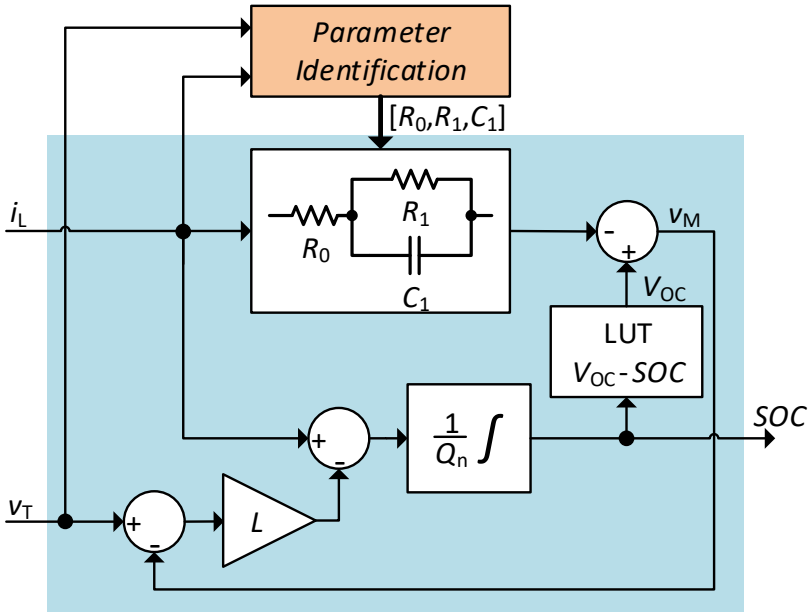


Figure 3.3: *Adaptive Mix Algorithm block diagram.*

In this work, the algorithm accuracy has been improved by taking into account the parameter variations due to changes in SOC, temperature, and current, with an online parameter identification technique, as introduced in Section 2.3.2. The employing of an online identification technique makes the algorithm adaptive, since it allows the model to adapt its parameters to the actual operating point. To this aim, the AMA integrates the Moving Window Least Squares (MWLS) method [113, 114], represented by the orange block in Figure 3.3.

We developed a simple procedure for setting the algorithms parameters in a more appropriate way with respect to the DEKF. In fact, the AMA has a reduced number of parameters that can be tuned in according with the target application by using the procedure described in Section 3.3.3. Moreover, the MWLS offers better stability and has potentially less divergence problems at the expense of a higher computational effort due to the inversion of large matrices [138].

A detailed description of the AMA has been reported in the following Sections, by separately analysing the SOC estimation part and the ECM parameter identification one.

3.3.1 SOC estimation

As said above, the SOC estimation is computed by the Mix Algorithm [32]. The ECM is used to predict the Li-ion cell behaviour. Its output voltage v_{M} is compared with the voltage across the cell terminals v_{T} and the generated error signal is amplified by the observer gain L and subtracted to the measured cell current i_{L}. The resulting signal is then integrated in order to compute the SOC with the conventional CC method.

This algorithm can be seen as an improved CC since it is able to reduce the sensitivity to uncertainties over current measurements and the SOC initial value thanks to the feedback loop. To prove these characteristics, we can analyse the stability of this

algorithm and the sensitivity to different error sources (on current measurement I_{err}, voltage measurement V_{err}, and wrong SOC initialization SOC_{err}), performed in the Laplace domain by linearising the OCV-SOC relationship [31]:

$$V_{\text{OC}}(s) = \alpha_0 + \alpha_1 SOC(s) \tag{3.35}$$

where α_0 and α_1 change with the operating point.

Stability analysis

The system stability can be evaluated by testing the effect of each input (i_{L} and v_{T}) in according to the superposition principle. In fact, the algorithm is asymptotic stable if the transfer functions with respect its inputs are asymptotic stable.

The model voltage and the SOC in the Laplace domain can be obtained from the state space model (3.1):

$$V_{\text{M}}(s) = V_{\text{OC}}(s) - Z_{\text{S}}(s)I_{\text{L}}(s) = \alpha_1 SOC(s) - Z_{\text{S}}(s)I_{\text{L}}(s) \tag{3.36}$$

$$SOC(s) = -\frac{I_{\text{L}}(s)}{sQ_{\text{n}}} \tag{3.37}$$

where $Z_{\text{S}}(s) = R_0 + (R_1/(1 + R_1C_1s))$ is the small-signal output impedance of the linearised cell model, while the coefficient α_0 is neglected in this analysis.

The transfer function $W_V(s)$ between the SOC and V_{T} is obtained by considering $I_{\text{L}} = 0$:

$$SOC(s)|_{I_{\text{L}}=0} = \frac{L}{sQ_{\text{n}}}(V_{\text{T}}(s) - \alpha_1 SOC(s)) \tag{3.38}$$

$$W_V(s) \equiv \frac{SOC(s)}{V_{\text{T}}(s)} = \frac{L}{Q_{\text{n}}}\frac{1}{s + \frac{L\alpha_1}{Q_{\text{n}}}} \tag{3.39}$$

while the transfer function $W_I(s)$ between the SOC and I_{L} is obtained by considering $V_{\text{T}} = 0$:

$$\begin{aligned} SOC(s)|_{V_{\text{T}}=0} &= -\frac{1}{sQ_{\text{n}}}(I_{\text{L}}(s) + L\alpha_1 SOC(s) - LZ_{\text{S}}(s)I_{\text{L}}(s)) = \\ &= I_{\text{L}}(s)\left(\frac{LZ_{\text{S}}(s) - 1}{sQ_{\text{n}}}\right) - \frac{L\alpha_1 SOC(s)}{sQ_{\text{n}}} \end{aligned} \tag{3.40}$$

$$W_I(s) \equiv \frac{SOC(s)}{I_{\text{L}}(s)} = \frac{LZ_{\text{S}}(s) - 1}{Q_{\text{n}}}\frac{1}{s + \frac{L\alpha_1}{Q_{\text{n}}}} \tag{3.41}$$

Both transfer functions have the same denominator where the sign depends on the term $L\alpha_1/Q_{\text{n}}$ and in particular on the sign of L, since α_1 and Q_{n} are always positive. We can note that if $L = 0$ the algorithm works in open-loop and the SOC is computed by the CC method, while if $L > 0$ the system is asymptotic stable.

Sensitivity to a wrong SOC initialization

In order to analyse the algorithm behaviour in presence of an error in SOC initialization, we can add an error $SOC_{0,err}$ at the input of the integration block. In this way, a new transfer function can be obtained:

$$SOC(s) = SOC_{0,err} - \frac{L\alpha_1 SOC(s)}{sQ_{\mathrm{n}}} \Rightarrow \frac{SOC(s)}{SOC_{0,err}} = \frac{s}{s + \frac{L\alpha_1}{Q_{\mathrm{n}}}} \tag{3.42}$$

The steady-state value of the SOC error $SOC_{err} = SOC_{true} - SOC$, where SOC_{true} is the real value and SOC is the estimated one, is obtained from (3.42) using the final value theorem:

$$\lim_{t\to\infty} SOC_{err}(t) = \lim_{s\to 0} sSOC_{err}(s) = \lim_{s\to 0} s\frac{SOC_{0,err}}{s}\frac{s}{s + \frac{L\alpha_1}{Q_{\mathrm{n}}}} = 0 \tag{3.43}$$

This means that the Mix algorithm is capable of fully correcting a bad SOC initialization independently of L, reaching the correct value with an exponential trend with a time constant equal to $L\alpha_1/Q_{\mathrm{n}}$.

Sensitivity to measurement errors

The same procedure previously applied for the analysis in case of an error in SOC initialization can be also used to determine the algorithm performance in presence of a constant error V_{err} on the voltage measurement:

$$SOC_{err} = \frac{V_{err}}{s} W_V(s) = \frac{V_{err}}{s}\frac{L}{Q_{\mathrm{n}}}\frac{1}{s + \frac{L\alpha_1}{Q_{\mathrm{n}}}} \tag{3.44}$$

The steady-state value of the SOC error is:

$$\lim_{t\to\infty} SOC_{err}(t) = \lim_{s\to 0} sSOC_{err}(s) = \lim_{s\to 0} s\frac{V_{err}}{s}\frac{L}{Q_{\mathrm{n}}}\frac{1}{s + \frac{L\alpha_1}{Q_{\mathrm{n}}}} = \frac{V_{err}}{\alpha_1} \tag{3.45}$$

An error in the cell voltage measurement leads to a non-zero steady-state error, which is proportional to the slope α_1 of the OCV-SOC curve in the actual SOC operating point.

Finally, the SOC steady-state error due to a static offset I_{err} in the cell current measurement can be obtained in the same way:

$$SOC_{err} = \frac{I_{err}}{s} W_I(s) = \frac{I_{err}}{s}\frac{LZ_{\mathrm{S}}(s) - 1}{Q_{\mathrm{n}}}\frac{1}{s + \frac{L\alpha_1}{Q_{\mathrm{n}}}} \tag{3.46}$$

$$\begin{aligned}\lim_{t\to\infty} SOC_{err}(t) =& \lim_{s\to 0} sSOC_{err}(s) = \lim_{s\to 0} s\frac{I_{err}}{s}\frac{LZ_{\mathrm{S}}(s) - 1}{Q_{\mathrm{n}}}\frac{1}{s + \frac{L\alpha_1}{Q_{\mathrm{n}}}} = \\ =& I_{err}\frac{LZ_{\mathrm{S}}(0) - 1}{L\alpha_1}\end{aligned} \tag{3.47}$$

where $Z_{\mathrm{S}}(0) = R_0 + R_1$. Equation (3.47) demonstrates that the steady-state error due to errors on current measurement can be cancelled with an appropriate choice of the gain value:

$$LZ_{\mathrm{S}}(0) - 1 = 0 \Rightarrow L = \frac{1}{R_0 + R_1} \tag{3.48}$$

Note that this value is not constant and depends on the resistive components of the ECM which vary during the algorithm execution, since they are computed online by the parameter identification block.

3.3.2 Online parameter identification

The functioning diagram of the parameter identification block is shown in Figure 3.4. It executes the online estimation of the ECM parameters R_0, R_1, and C_1 by using the MWLS technique applied to the AutoRegressive eXogenous (ARX) structure of the ECM. The inputs are the samples of the cell current i_L and terminal voltage v_T. The signal conditioning block filters with a third-order Butterworth low-pass filter and decimates the input samples. This avoids aliasing and allows noise, affecting the measured voltage and current signals, and the dynamics out of interest to be filtered out.

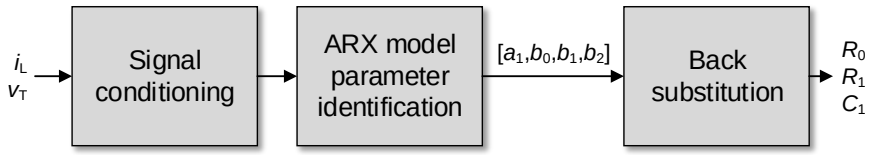


Figure 3.4: *Diagram of the MWLS algorithm.*

The estimation of the model parameters is performed by using a certain number M of conditioned samples, belonging to a temporal window of length $L_\mathrm{W} = M \cdot T$, where T is the sampling time after decimation. At each time step, the temporal window is updated by using the new acquired current and voltage samples. This operation is equivalent to a time shifting of the window and allows to track the parameter variations during the cell operation.

In the ARX model the actual value of the output $y(k)$ depends on its N_a past values and the actual and the N_b past value of the exogenous input $u(k)$:

$$y(k) = \sum_{i=1}^{N_a} a_i y(k-i) + \sum_{i=0}^{N_b} b_i u(k-i-d) \tag{3.49}$$

where a_i and b_i are the coefficients of the ARX model, and d represents the minimum delay between the input and the output. This representation can be obtained by linearising the model around the operating point as in Section 3.3.1, since it slowly changes over time in a Li-ion cell [92]. Then, the discrete-time transfer function of the linearised ECM is obtained using the bilinear transform

$$s \to \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{3.50}$$

from Eq. (3.1):

$$\frac{Y(z^{-1}) - \alpha_0}{U(z^{-1})} = -\frac{b_2 z^{-2} + b_1 z^{-1} + b_0}{a_2 z^{-2} + a_1 z^{-1} + 1} \tag{3.51}$$

where $Y(z^{-1})$ and $U(z^{-1})$ are the z-transforms of the voltage output v_T and current input i_L, respectively [13]. The coefficients of the discrete-time transfer function (3.51) can be written as follows:

$$a_1 = -\frac{4\tau_1}{2\tau_1 + T} \tag{3.52}$$

$$a_2 = \frac{2\tau_1 - T}{2\tau_1 + T} \tag{3.53}$$

$$b_0 = -\left[4R_0 + 2T\left(\frac{\alpha_1}{Q_\mathrm{n}} + \frac{R_0}{\tau_1} + \frac{1}{C_1}\right) + \frac{\alpha_1 T^2}{Q_\mathrm{n}\tau_1}\right]\gamma \tag{3.54}$$

$$b_1 = -\left(\frac{2\alpha_1 T^2}{Q_\mathrm{n}\tau_1} - 8R_0\right)\gamma \tag{3.55}$$

$$b_2 = -\left[4R_0 - 2T\left(\frac{\alpha_1}{Q_\mathrm{n}} + \frac{R_0}{\tau_1} + \frac{1}{C_1}\right) + \frac{\alpha_1 T^2}{Q_\mathrm{n}\tau_1}\right]\gamma \tag{3.56}$$

where $\gamma = \tau_1/(4\tau_1 + 2T)$. The discrete-time transfer function (3.51) is used to obtain the second order ARX model of the system:

$$\begin{aligned} y(k) = &- a_1 y(k-1) - a_2 y(k-2) + \alpha_0(1 + a_1 + a_2) \\ &+ b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) \end{aligned} \tag{3.57}$$

Equations (3.52) and (3.53) yield $1 + a_1 + a_2 = 0$, thus (3.57) simplifies as follows:

$$\begin{aligned} y(k) - y(k-2) = &a_1(y(k-2) - y(k-1)) + b_0 u(k) \\ &+ b_1 u(k-1) + b_2 u(k-2) \end{aligned} \tag{3.58}$$

Equation (3.58) is used to build an overdetermined linear system, which is solved to obtain the vector $x = [a_1, b_0, b_1, b_2] \in \mathbb{R}^4$. The Least Squares (LS) technique has been employed to solve the overdetermined linear system $Ax = b$, where $A \in \mathbb{R}^{M\times 4}$, and $b \in \mathbb{R}^M$. Vector x is obtained by performing the QR decomposition of the matrix A. In this way, the system is decomposed in the following one:

$$Ax = b \Rightarrow QRx = b \Rightarrow Rx = Q^T b \Rightarrow Rx = d \tag{3.59}$$

where $Q \in \mathbb{R}^{M\times M}$ is an orthogonal matrix and $R \in \mathbb{R}^{M\times 4}$ has the higher submatrix $R_u \in \mathbb{R}^{4\times 4}$ upper triangular. The LS solution of the system is those that minimize the quantity

$$\begin{aligned} &||Ax - b||_2 = ||Rx - d||_2 = \\ &\left|\left|\begin{bmatrix} R_u \\ 0 \end{bmatrix} x - \begin{bmatrix} d_u \\ d_l \end{bmatrix}\right|\right|_2 = ||d_l||_2 + ||R_u x - d_u||_2 \end{aligned} \tag{3.60}$$

where $d_u \in \mathbb{R}^4$ and $d_l \in \mathbb{R}^{M-4}$. The minimum of $||Ax - b||_2$ is equal to $||d_l||_2$, and it is obtained for values of x that allow $R_u x - d_u$ to be equal to zero.

There are different methods which can be used to compute QR decomposition including Gram-Schmidt Orthonormalization, Householder Reflections, and the Givens Rotation. In the AMA, the QR decomposition is executed by using the Givens Rotation, since it is considered numerically stable and easy to implement with pipelining and ASIC fabrication [6, 144]. The Givens Rotation algorithm can be described with

Algorithm 1 Givens Rotation algorithm.

$Q = I_m$

for $j = 1 : n$

  for $i = m : -1 : (j + 1)$

    $[c, s] = givens(A(i - 1, j), A(i, j))$

    $A(i - 1 : i, j : n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A(i - 1 : i, j : n)$

    $Q(i - 1 : i, j : n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} Q(i - 1 : i, j : n)$

  end

end

the MATLAB-like code reported in Algorithm 1, where the *givens* function, reported in Algorithm 2, computes the values of c and s such that:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \tag{3.61}$$

Algorithm 2 Givens function.

function$[c, s] = givens(a, b)$

  if $b = 0$

    $c = 1; s = 0$

  else

    if $|b| > |a|$

      $\tau = -a/b; s = 1/\sqrt{1 + \tau^2}; c = s\tau$

    else

      $\tau = -b/a; c = 1/\sqrt{1 + \tau^2}; s = c\tau$

    end

  end

The vector containing the ARX model coefficients is then used during a back substitution phase, described in Algorithm 3, in which the ECM parameters are computed by inverting equations (3.52), (3.53), (3.54), (3.55), and (3.56).

Algorithm 3 Back substitution algorithm.

x=[0; 0; 0;$d(k,1)/R(k,k)$]

for $k = n-1:-1:1$

$x(k,1) = \frac{d(k,1)-R(k,k+1:n)x(k+1:n,1)}{R(k,k)}$

end

The MWLS has some parameters that can be tuned in relation with the target application, such as the length of the identification window L_{W} and the cut-off frequency of the low-pass filter f_{c}. A procedure for tuning these parameters has been presented in the following Section.

3.3.3 MWLS tuning procedure

The procedure for tuning the MWLS parameters has been presented in [90] and uses a typical current profile of the battery in the considered application. The number of samples and the sample time depend on the BMS platform. In fact, the sample time is fixed by the sampling period of the monitoring circuit in the BMS and is typically between $10\,\mathrm{ms}$ and $100\,\mathrm{ms}$. The number of samples M in the identification window is determined by the trade-off between complexity and accuracy that offers the processing unit of the BMS. A better accuracy can be reached by using a high value of M because the estimation is computed starting from a larger set of input samples. However, this value affects the memory utilization and the computational effort of the Givens Rotation algorithm, as described in Section 3.3.2, and so it is fixed by the available hardware resources. The only tunable algorithm's parameters are L_{W} and f_{c}. The procedure is the following:

1. Determine a range of reasonable values of L_{W}:
   (a) The maximum value of L_{W} has to be chosen so that the assumption of constant model parameter values still holds in the identification window;
   (b) Regarding its minimum value, we observe that the equations used to compute the vector $[R_0, R_1, C_1]$, obtained from equations (3.52), (3.54), (3.55) and (3.56), depend on the term $a_1/(2+a_1)$. The value of a_1 should thus be sufficiently far from -2 to obtain a non ill-conditioned problem. Hence, from (3.52), the value of T should not be much less than τ_1, *i.e.*, $T \geq \tau_1/10$. This implies that $L_{\mathrm{W}} \geq M \cdot (\tau_1/10)$.
2. Determine a range of reasonable values of f_{c}:
   (a) The maximum value of the cut-off frequency has to satisfy the Shannon theorem ($f_{\mathrm{c}} \leq 1/(2T)$), thus $f_{\mathrm{c}} \leq M/(2L_{\mathrm{W}})$;
   (b) The minimum value is obtained considering that it cannot be too low in order to preserve the fast dynamics of the battery, so $f_{\mathrm{c}} \geq 0.5/(2\pi\tau_1)$.
3. A tuning phase is carried out by using a load current profile typical of the target application.

4. The best combination of L_{W} and f_{c} is found by evaluating the rms error of the ECM predicted voltage, when L_{W} and f_{c} vary in their defined ranges, and by choosing the couple of values that minimise this error. The latter is computed as the difference between the measured voltage and the one predicted by the ECM, when the parameters R_0, R_1 and C_1 are identified by the MWLS algorithm.

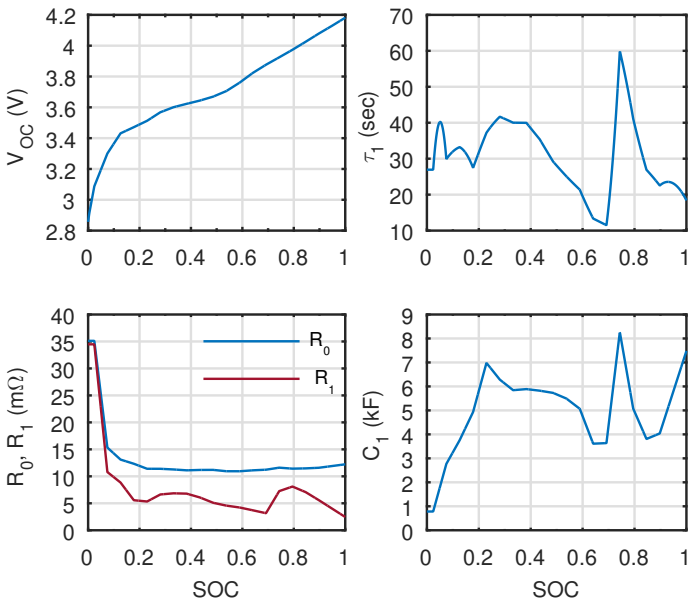


Figure 3.5: *Cell parameters extracted from a PCT test executed at room temperature.*

A gardening application is taken as a case study to verify the tuning procedure, assessing also the algorithm performance. Battery powered gardening tools can provide different types of power profiles, which allow us to explore different battery operating conditions. In particular, two tools with different power requirements are considered [39, 79]. The battery consists of 24 Li-ion cells, manufactured by LG Chem. The cells are arranged in 12 series-connected groups of two parallel-connected cells. The battery is monitored by a BMS, which measures the voltage, current and temperature of each cell group. A cell group has been characterised using a PCT, which has been presented in Section 2.3.2. The parameters measured at room temperature are shown in Figure 3.5. Four tests have been selected to tune the validation procedure. They all start with a fully charged battery, which is completely discharged operating the tool in its normal use. The current profiles acquired by the BMS during a part of the four tests are shown in Figure 3.6. The first two tests (test 1 and test 2) have been carried out on a tool with a low power requirement of $160\,\mathrm{W}$. The others two (test 3 and test 4) come from the second tool, which has a rated power absorption of $320\,\mathrm{W}$. The tuning procedure has been performed on test 1 and the selected parameters are used for the model parameter identification of the 12 series-connected groups in the four tests.

The ranges' bounds are calculated by following the steps 1 and 2 of the presented procedure and considering that a new sample is acquired every $100\,\mathrm{ms}$ and that the identification window is composed by $M = 20$ samples. The values of L_{W} and f_{c} are varied in the ranges of 60 to $540\,\mathrm{s}$ and 2 to $14\,\mathrm{mHz}$, respectively. After the execution of the steps 3 and 4 we obtain a map of the rms error as function of L_{W} and f_{c}. The

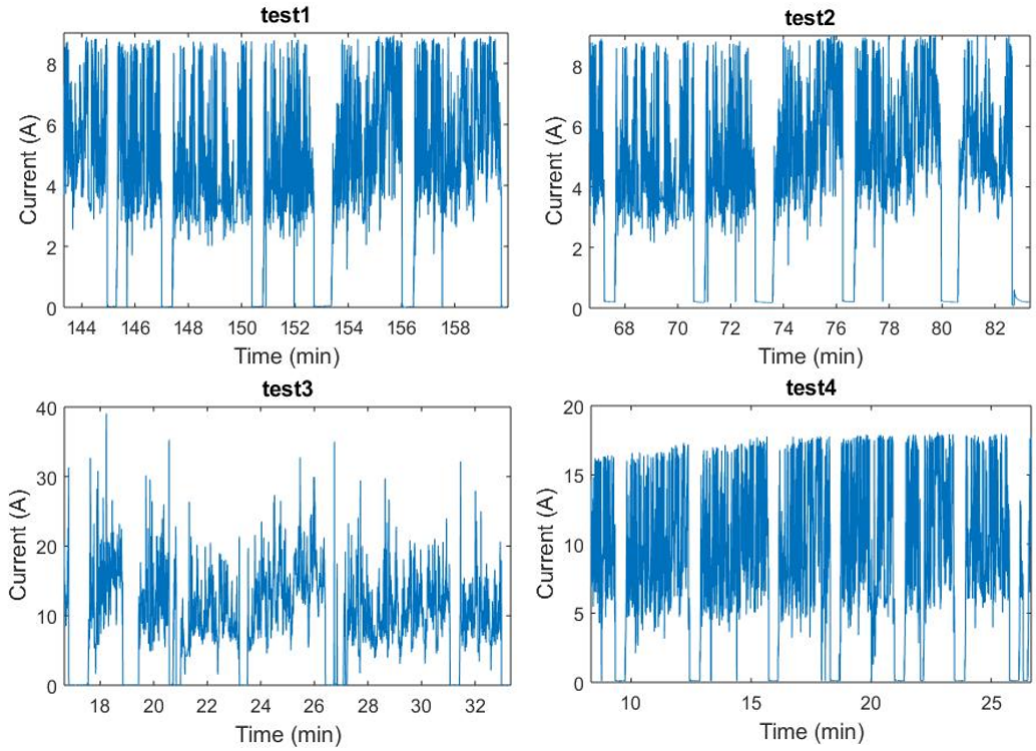


Figure 3.6: *Current profiles acquired by the BMS during the four tests carried out on the two tools.*

error computed considering the first group of two parallel-connected cells, using test 1, is shown in Figure 3.7. This figure highlights an area around $L_{\mathrm{W}} = 210\,\mathrm{s}$ and $f_{\mathrm{c}} = 5\,\mathrm{mHz}$, where the error is low and the values of the MWLS parameters should be chosen.

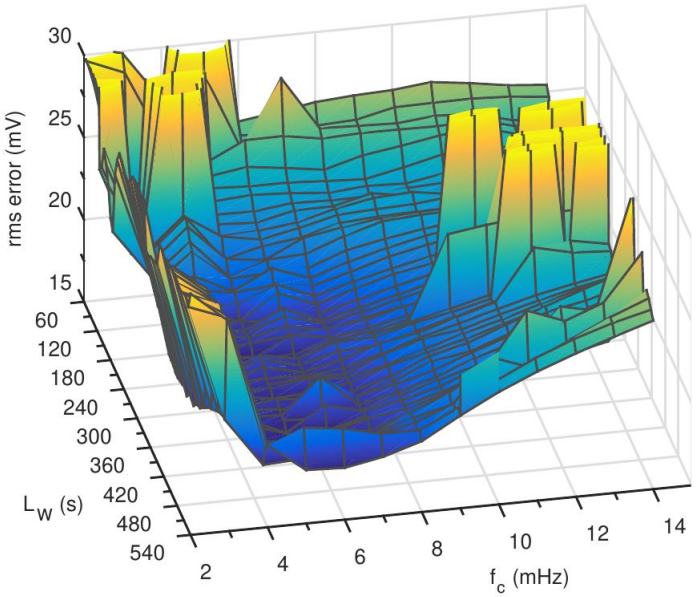


Figure 3.7: *rms error of the voltage predicted by the ECM, as function of the identification window length L_{W} and of the cut-off frequency f_{c} of the filter.*

Therefore, these values have been selected and used to apply the algorithm on all the battery cell groups in each test in order to verify the effectiveness of the procedure. Figure 3.8 shows a comparison between the voltage rms errors obtained by using the parameters identified by the MWLS and those extracted offline from the PCT. The errors obtained with online identification are clearly lower in the third and fourth test.

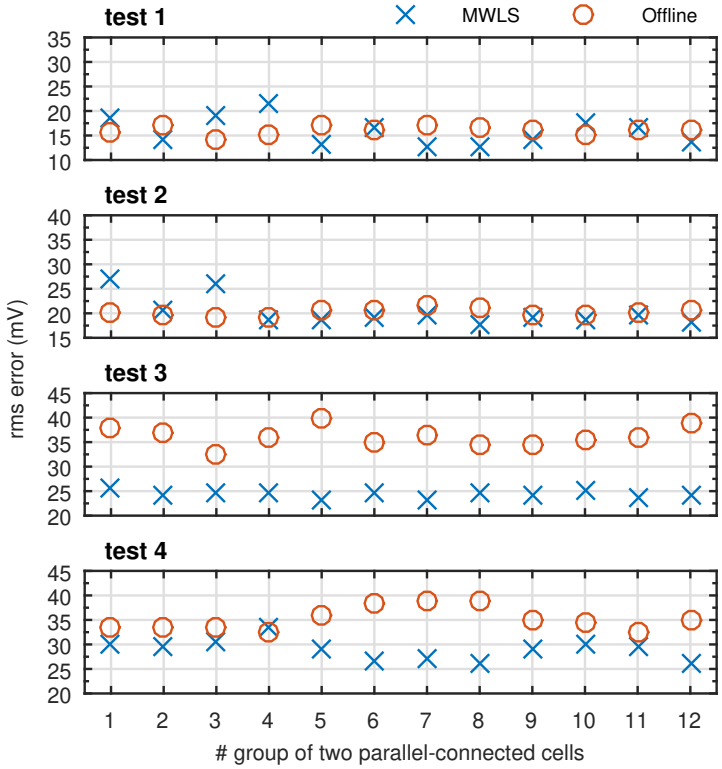


Figure 3.8: *Comparison of the voltage rms error obtained by using the parameters of the MWLS and those extracted offline. Each point corresponds to a group of two parallel-connected cells.*

As these tests have been executed on the tool with the higher mean power absorption, a battery temperature increase of about $30\,^{\circ}\mathrm{C}$ is observed, as shown in Figure 3.9 for test 3, where it is also reported the comparison between the ECM parameters estimated online by the MWLS and those obtained by running the ECM with the parameter extracted offline and the same current acquired during the test, for the first group of two parallel-connected cells. During this test, the current becomes equal to zero after minute 38. For this reason, the temperature slightly decreases and the parameters obtained offline remain constant. It is worth noting that MWLS is able to take into account the variations of the model parameters due to the temperature change during the use, providing better results in the cell voltage prediction. In more detail, the parameters are close to those extracted by the PCT in the first half of the test, but they are quite different at the end of the test where the battery temperature reaches about $50\,^{\circ}\mathrm{C}$. This result clearly shows the benefits of using an online parameter identification algorithm.

The situation is slightly different for the other tool (test 1 and test 2). In this case, the online identified parameters are similar to those extracted from the PCT at room temperature and the rms errors are comparable. Indeed, the battery temperature remains almost constant to the ambient value during these tests, as the power absorbed by the

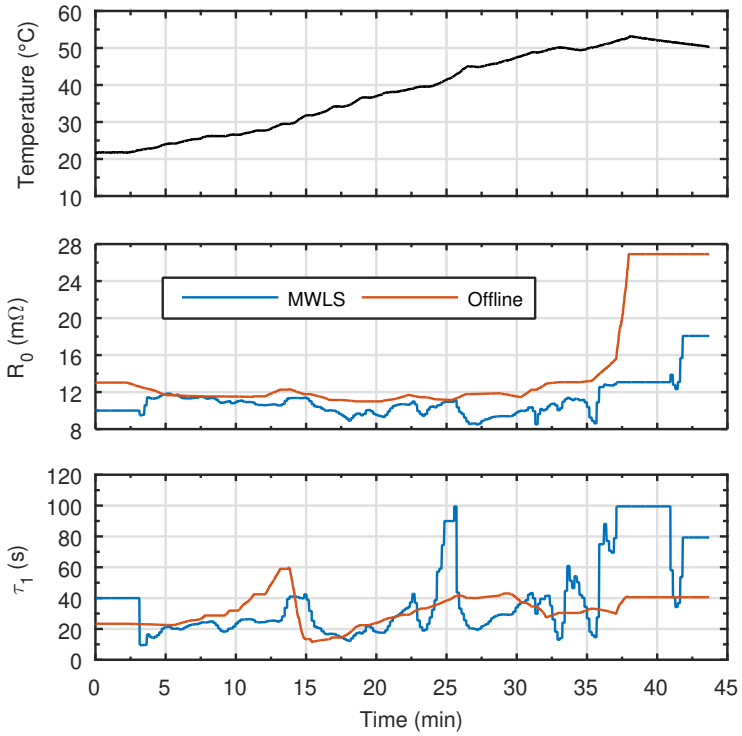


Figure 3.9: *Temperature and ECM parameters estimated online by the MWLS compared with those extracted offline in test 3 for the first group of two parallel-connected cells.*

tool is relatively low.

CHAPTER 4

Hardware acceleration of the battery state estimators

4.1 Introduction

As said in Chapter 2, using FPGA and SoC-FPGA instead of a microcontroller in a BMS enables the possibility to increase its flexibility, reconfigurability, and estimation capability. Thanks to their higher performance with respect to microcontrollers [42, 69, 97] and parallelism, complex and accurate algorithms can be executed for a large number of cells without interfering with the other BMS functions. In fact, a BMS can be equipped with custom peripherals in order to perform specific operations. Many works in literature show the advantages of implementing complex, accurate, and robust state estimation algorithms in hardware by using FPGAs, like different versions of the Kalman Filter for SOC and cell parameter estimation [34, 131], and the Mix Algorithm [16] or other observers [98, 136] for SOC estimation.

In this work, both the DEKF and the AMA algorithms, explained in Sections 3.2 and 3.3, respectively, have been implemented as hardware estimators by using Intel FPGA devices, such as a low cost Intel MAX® 10 FPGA. A high-level design flow provided by Intel has been used to this aim, in order to reduce the developing time and to guarantee an easy and fast portability among all the devices of the producer.

The high-level developing process and its advantages is explained in detail in the next section, while the implementation of the two hardware estimators are explained in sections 4.3 and 4.4.

4.2 Design flow

The high-level design flow provided by Intel and drawn in Figure 4.1 includes the DSP Builder tool that allows us to describe the algorithm by using a Simulink model and the blocks provided in the DSP Builder Advanced Blockset and to automatically generate the low-level hardware description [123].

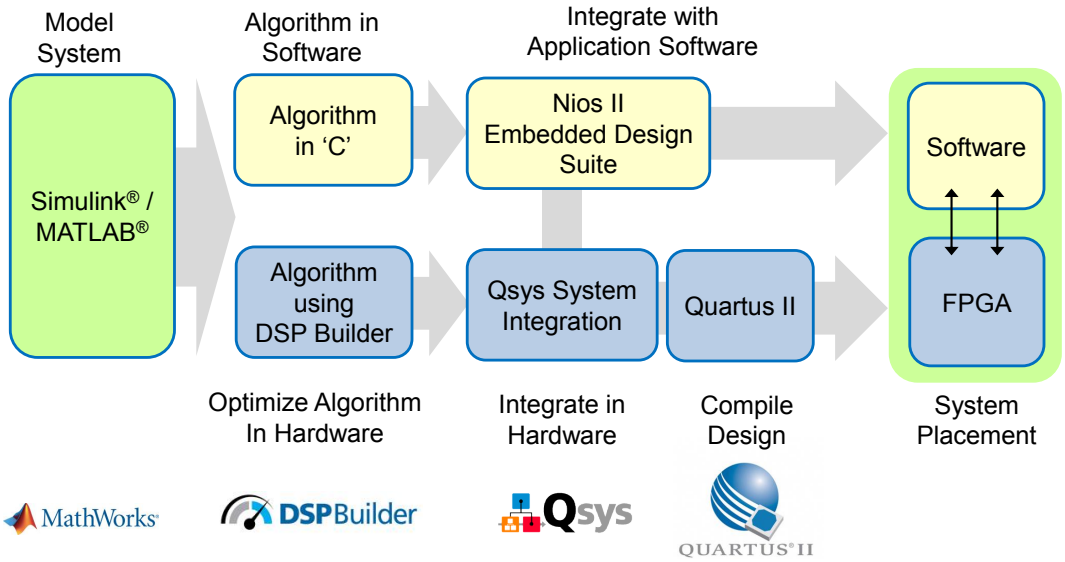


Figure 4.1: *High level design flow based on DSP Builder.*

The generated hardware components are then provided with an interface that allows it to be connected to other components in the system, including a Nios II 32-bit soft processor, JTAG and memory, by using the Qsys system integration tool. In Intel FPGAs, system components provided with different interfaces can communicate through the Avalon bus. As an example, the Avalon Memory Mapped Interface (Avalon-MM) shown in Figure 4.2 allows an address-based read/write communication typical of master–slave connections.

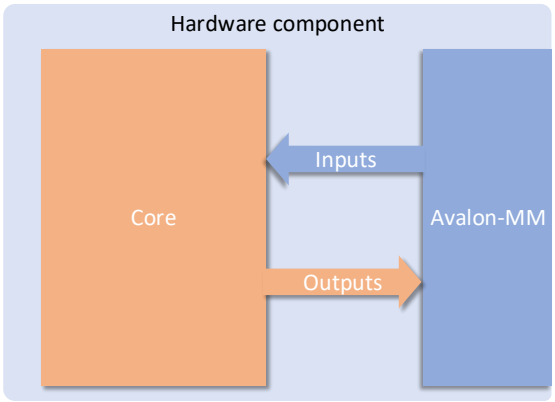


Figure 4.2: *Hardware component provided with the Avalon Memory Mapped Interface.*

The complete design is synthesized and programmed for the target FPGA using

Quartus II design software. For algorithms which includes application software running on Nios II soft-core, Embedded Design Suite compiles the C software and launches the compiled application on the FPGA. The procedure is the same for devices equipped with the ARM® hard-core, where the software is compiled by using the DS-5 suite.

In addition to the reduction of the design effort, this tool allows the automatic generation of the components for user-defined FPGA device. This ensures the portability of the hardware system for various Intel FPGA families. The design is optimized for performance and resource by applying pipelining, time-division multiplexing/folding and customizing precision, optimizing the hardware module performance and resource usage depending on the tool parameter settings. Furthermore, DSP Builder is also optimized for floating-point designs. Usually, FPGAs can achieve high performance, especially when the workloads are highly parallelizable, but the performance are limited in floating-point designs due to long processing latencies and routing congestion. Moreover, the register-transfer-level description in Verilog or VHDL is not suitable to implement complex floating-point algorithms. In case of a series of elementary floating-point operators, DSP Builder overcomes these problems by generating a fused datapath that combines elementary operators into a single and simpler function or datapath.

4.3 Adaptive Mix Algorithm

The hardware component is composed by the estimator which implements the AMA and the a Avalon-MM slave interface used to integrate the component in an embedded system, as shown in Figure 4.2. As an example, the peripheral can be managed by both the Nios II or the hard-core processors which are provided with an Avalon-MM master interface, allowing the user to access to the data stored in the peripheral using simple read/write software functions. The estimator is composed by three blocks (Figure 4.3): the low-pass filter, the MWLS block, and the Mix algorithm one. The input and the output values are collected by the Avalon-MM interface which also allows the user to change the algorithm and the filter coefficients via software, making the estimator suitable for different applications and battery cell technologies.

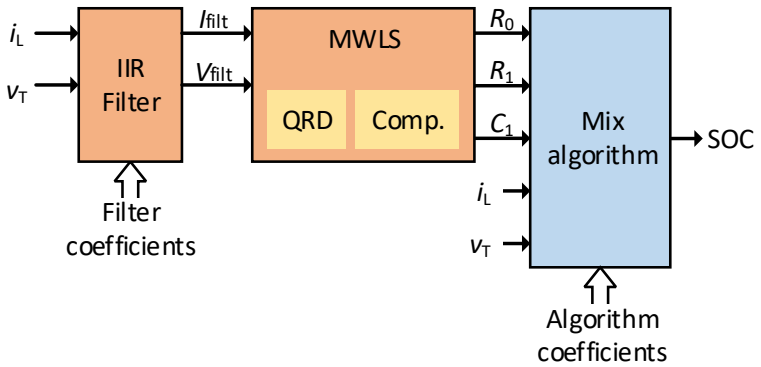


Figure 4.3: *Block diagram of the implemented AMA hardware estimator.*

The input of the MWLS block are the filtered and decimated voltage and current samples. The samples are filtered by a third-order Butterworth low-pass filter. This avoids aliasing and allows noise and the dynamics out of interest to be filtered out. The

cut-off frequency of the filter can be selected by changing the filter coefficients, and it can be obtained with the procedure described in Section 3.3.3. The filter has been realized by using a two-stages biquad architecture.

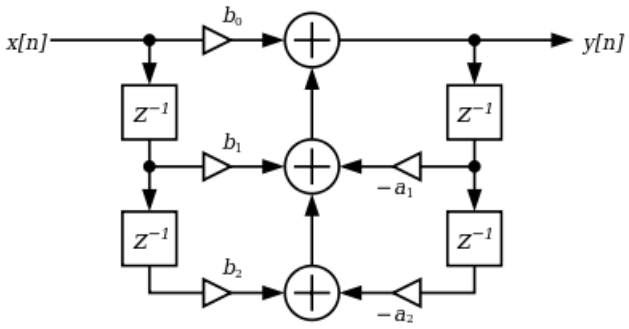


Figure 4.4: *Flow graph of a biquad filter.*

A single-stage biquad filter has the architecture shown in Figure 4.4 and the following discrete-time function:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \tag{4.1}$$

where usually $a_0 = 1$, *i.e.*, the coefficients are normalised with respect to a_0.

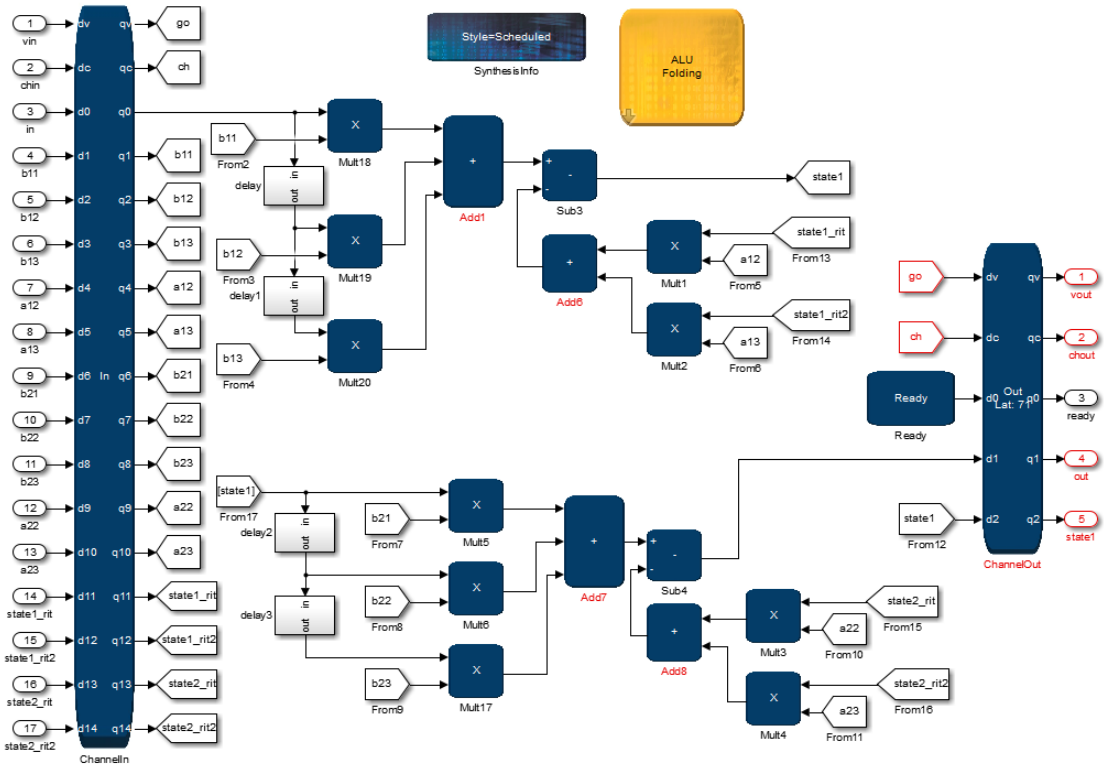


Figure 4.5: *Implementation of the filter in the Intel DSP Builder tool.*

The DSP Builder implementation of the filter is shown in Figure 4.5. The two biquad stages are visible in the figure and they can be used for a configurable number of input

channels. Each channel is selected by using a dedicated input. The filter is optimised for resource usage since the ALU folding block is inserted in the design. It enables the possibility of the architecture folding which reduces the number of implemented elementary operators by using them in time multiplexing.

The MWLS block can be further divided in two sections. The first one, so called QRD in Figure 4.3, composes the linear system with the collected samples and solves it by using the QR decomposition, as shown in equation (3.59). The various phases are managed by a state machine. The output values are the coefficients of the ARX model which are used by the second part of the MWLS block which computes the back substitution by inverting equations (3.52), (3.53), (3.54), (3.55), and (3.56). These sub-blocks implement the algorithms described in Section 3.3.2.

The SOC estimation is computed by the Mix algorithm block. It takes as inputs the samples of the cell voltages and current and the last valid parameter values computed by the MWLS block. The estimator is very similar to that presented in [16] and its DSP Builder implementation is reported in Figure 4.6.

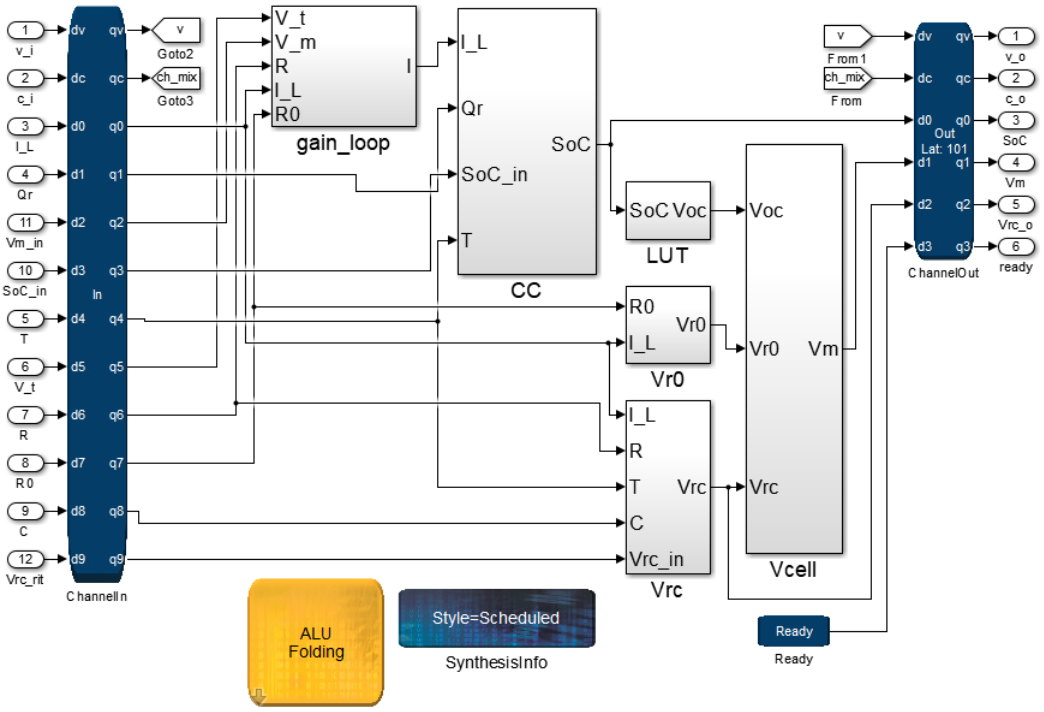


Figure 4.6: *Implementation of the Mix algorithm in the Intel DSP Builder tool.*

The integration of the current is executed by the CC block by using equation (3.37). The latter can be reported in the discrete-time domain using the bilinear transform (Eq. (3.50)):

$$SOC(z) = -\frac{I_{\mathrm{L}}(z)}{Q_{\mathrm{n}}}\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}} \tag{4.2}$$

from which we can obtain the operation computed by the CC block:

$$SOC(z) = SOC(z)z^{-1} - \frac{T}{2Q_{\mathrm{n}}}I_{\mathrm{L}}(z)(1+z^{-1}) \tag{4.3}$$

In this equation, the term $T/2Q_{\mathrm{n}}$ does not change during the operation of the estimator, thus it is provided to the estimator as a fixed value (E_1) during the initialization phase. At each time step, the model cell voltage is computed as sum of the OCV value, obtained by using a LUT addressed by the SOC estimate, the voltage across the series resistance and the voltage across the RC branch. The latter is computed by using equation (3.1), that in the Laplace domain is:

$$sV_1(s) = -\frac{V_1(s)}{\tau_1} + \frac{I_{\mathrm{L}}(s)}{C_1} \tag{4.4}$$

The discrete-time equation is obtain in the same way explained above:

$$V_1(z)\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} = -\frac{V_1(z)}{\tau_1} + \frac{I_{\mathrm{L}}(z)}{C_1} \tag{4.5}$$

and then the final equation can be write as follows:

$$V_1(z) = \frac{1}{2\tau_1}[(2\tau_1 - T)V_1(z)z^{-1} + R_1 T I_{\mathrm{L}}(z)(1+z^{-1})] \tag{4.6}$$

The model voltage is used by the gain_loop block to generate the correction term which is applied to the current measurement, after the multiplication by the gain L.

Table 4.1: *AMA Resource Usage for the 10M50DAF484C6GES FPGA.*

System entity	Logic Elements	9-bit Multiplier	Memory bits
Mix algorithm	7.5 k	23	36 Kb
MWLS	30 k	196	134 Kb
Filter	3.5 k	7	5 Kb
Total	41 k/50 k (82 %)	226/288 (79 %)	175 Kb/1638 Kb (11 %)

As an example, Table 4.1 reports the resource usage by entity of the AMA hardware estimator implemented for a 12-cell battery and developed in a low cost Intel MAX® 10 FPGA (10M50DAF484C6GES device). This device targets low-cost applications and includes non-volatile memory and integrated ADCs. The execution time to update both the state and the parameters of one single cell is $34\,\mu\mathrm{s}$, measured with a clock frequency of $100\,\mathrm{MHz}$. This very short execution time makes it possible to use the same module in time multiplexing for estimating a large number of cells.

4.4 Dual Extended Kalman Filter

The DEKF hardware estimator has been developed in collaboration with Intel. The work has been developed into a reference design that is freely available and can be run on low-cost hardware, such as the FPGA device presented in the previous section. The estimator can be executed by using a HiL platform available in the reference design and that can be used to develop a full FPGA-based BMS [89, 131]. Further details on this platform has been reported in Chapter 5.

The implementation carried out by using the DSP Builder tool in shown in Figure 4.7, where the two EKFs, which implement the discrete-time equations presented

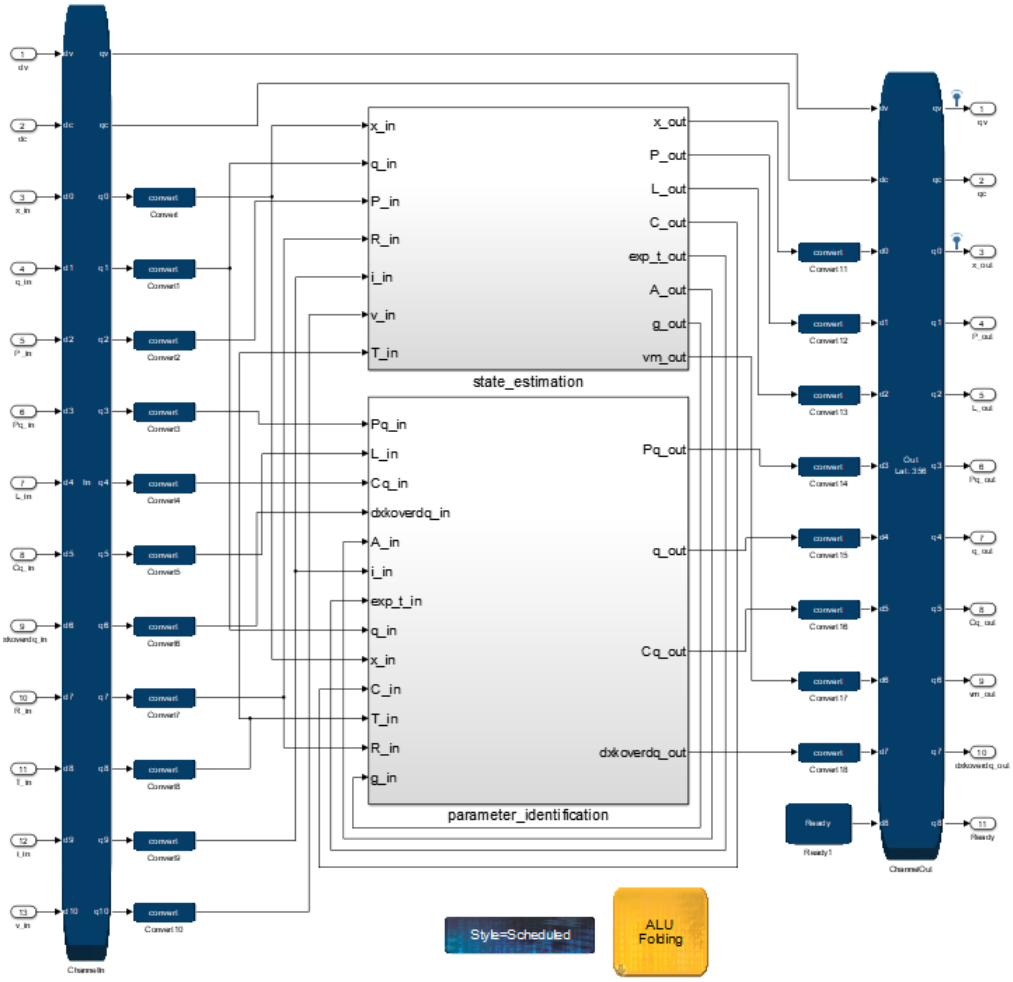


Figure 4.7: *Implementation of the dual extended Kalman filter in the Intel DSP Builder tool.*

in Section 3.2, are clearly visible. Also in this case, the input/output operations are carried out by using an Avalon-MM interface.

The resource usage of the DEKF estimator implemented for a 12-cell battery is reported in Table 4.2 for the Intel MAX® 10 10M50DAF484C6GES device. It is noticeable that the used resources in terms of logic elements and multipliers are lower than the resources employed by the AMA estimator, while the AMA requires less memory bits.

Table 4.2: *DEKF Resource Usage for the 10M50DAF484C6GES FPGA.*

System entity	Logic Elements	9-bit Multiplier	Memory bits
DEKF	23 k/50 k (46 %)	39/288 (14 %)	230 Kb/1638 Kb (26 %)

Besides the fully DSP Builder implementation, the DEKF has been developed in three different ways:

- In software, executed by the Nios II soft-core with floating point acceleration.
- A solution that implements some functions in software, executed by the Nios II soft-core with floating point acceleration, and the matrix operations in hardware by using a floating point matrix processor. The matrix processor has been developed by Intel using the DSP Builder Advanced Blockset. It can perform sequences of different matrix operations in parallel to the Nios II core.
- In software, executed by a custom processor. This solution has been implemented by using the Intel's ASIP tool that converts a C description (with restrictions) of the algorithm into a program which runs on a custom processor.

The time that the various implementations employed to update the SOC and the parameters of a single cell is reported in Table 4.3. As expected, the fully hardware solution has the best performance, while the pure software implementation has the longest execution time. However, this comparison can be very useful in order to allow a designer to choose a balance between calculation time and resource usage [131].

Table 4.3: *Execution time for each implementation method measured at* 100 MHz.

Implementation method	Time (µs)
DSP Builder IP	16.5
Software	44.9
Software with matrix processor	33.8
ASIP core	21.5

CHAPTER 5

Hardware-in-the-loop platform for hardware estimators assessment

5.1 Introduction

During the estimator development and verification phases, the designer should verify that the algorithm provides reliable results in any realistic condition. To this aim, we developed a HiL platform able to reproduce the usage of a battery in an EV. It has been used to test and tune both the AMA and the DEKF algorithms.

HiL simulation platforms generally can be categorised in: Power-HiL (PHiL) and communication-based HiL [17]. In particular, PHiL platforms allow the BMS hardware to be tested when the power fluxes from/to the battery are real and not simulated. Instead, the BMS control algorithms are tested by substituting the battery power path with simulated battery data (*i.e.*, voltage, current, and temperature of each battery cell), provided via a communication link, in the communication-based HiL platforms.

Different PHiL testing platforms are described in [7, 27, 36, 41, 60, 83, 153], where the battery state estimators are tested by using specific current profiles and including in the loop a real battery or an emulator able to reproduce the battery behaviour. In [60], a profile based on the electric power measured on an EV driving the Federal Test Procedure (FTP) has been used, while the current has been generated starting from the combination of the Extra-Urban Driving Cycle (EUDC) and the Economic Commission for Europe urban (ECE) driving schedules in [83]. A current profile typical for the use of a smartphone is employed in [7], whereas a dynamic stress test is applied in [153] and [27]. In these works, the implemented systems include complex instrumentation like cyclers or programmable loads and chargers, in order to apply the selected profiles to the battery, and accurate measurement equipment to acquire the cell quantities. In [36], the HiL system is based on an emulator able to reproduce the battery cell dynamics to

validate the functions of voltage monitoring, active and passive cell balancing. A low-cost battery emulator with basic functionalities is used in [41] for the functional testing of a BMS at the end of the production line. It can be very useful in many medium power applications, characterized by rather low production volumes, where the investments in the design of the system should be limited and classic solutions for HiL testing might not be affordable.

Communication-based platforms allow less expensive, faster and safer tests than those performed on real batteries, because the battery itself belongs to the simulated part of the testing loop. The obvious disadvantage of the communication-based platforms is the impossibility to test some hardware parts of the BMS, such as the measurement and the balancing circuits. On the contrary, any possible load power profile can easily be applied to analyse the BMS behaviour and to validate the battery state estimators in the most demanding situations. Moreover, not using a real battery in the HiL platform also allows the saving of the battery charging time period, so successive profiles can be applied to the battery without pauses. Communication-based platforms are presented in [117, 127]. A simple battery model composed by 96 cells is used in [127] to perform a functional test of the BMS, while a 12-cell SOC estimation algorithm implemented on a microcontroller is tested in [117] by using a cell ECM that does not take into account the parameter variation with SOC and temperature.

The simple models used in these two works are inadequate when high reliability and accuracy of the BMS algorithms is required. Some of these limitations are overcome with the communication-based platform developed in this work in which the modelled EV and its battery parameters are fully configurable and the temperature effects are also considered by using a simple thermal model of the battery and temperature-dependent cell model parameters. In this way, the battery behaviour is reproduced considering a large set of conditions, such as the charge unbalance among the battery cells, the parameter variations due to their operating conditions and ageing, and different current loads. In fact, the HiL platform allows us to simulate an EV in many different driving scenarios and thus with different power dynamics. Furthermore, since the communication-based platforms directly send the acquired quantities to the main unit of the BMS, the acquisition chain and the actuators of the BMS are out of the loop and thus neglected in the tests. This problem has been solved by introducing a model of the voltage and current acquisition chain in this platform. The implemented simulation framework, included the battery and the electric traction models, has been presented in sections 5.2 and 5.3. The HiL platforms presented in literature [117, 127] usually use external devices to perform the test in real-time. To this aim, this instrumentation is provided with high computational power, multicore processors, high speed communication links, and real-time operating systems. In our case, the entire system can be built in Simulink in order to obtain a low cost software platform, without using additional devices. However, this does not guarantee the real-time execution of the model and the simulation performances depend on the computational power of the used computer and on the type of digital communication link used to send the simulated data to the hardware estimator under test.

We reported the capabilities of the developed HiL platform in Section 5.4, besides the analysis and the comparison of the results obtained from the AMA and the DEKF algorithms.

5.2 Hardware-in-the-loop platform design

The communication-based HiL platform (Figure 5.1) has entirely been implemented in the MATLAB/Simulink® environment. The driving schedule block allows us to select a standard speed profile, which is the input of the EV model block. The latter generates the input signals for the estimation algorithms, which are simultaneously executed by both the BMS hardware and the platform software. In fact, the algorithms under test are also implemented with MATLAB scripts and integrated in the Simulink framework, in order to compare their results with those obtained from the BMS and to validate their hardware implementation.

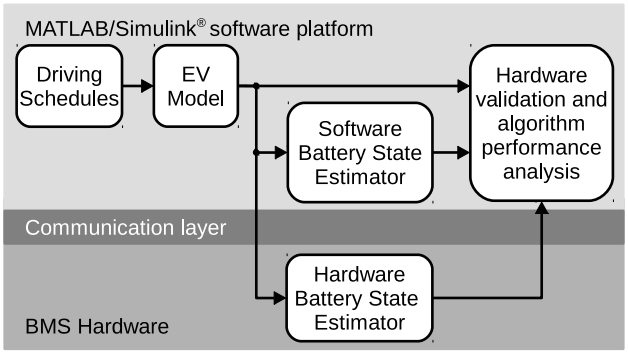


Figure 5.1: *Block diagram of the developed communication-based hardware-in-the-loop simulation platform.*

The communication with the BMS hardware is performed by the communication layer. This part of the HiL platform aims at sending the generated battery quantities to the BMS hardware under test by using a digital link. A large set of communication typologies is supported by the MATLAB environment, such as serial interfaces, CAN-bus, and Ethernet, but also a custom interface available from a third-party can be added.

5.2.1 Driving schedules

The Driving Schedules block provides the speed values relative to the driving cycle selected among the 18 available profiles, however further driving cycles from regulation authorities or custom choices can easily be added to the default portfolio. The selected driving cycles are usually used for vehicle emission and fuel economy assessment and are representative of various driving situations. Their main characteristics are collected in Table 5.1 that shows a significant variation in the average speed and, thus, in the electric power required from the traction battery. These cycles are defined by various specialised organisations, therefore they can be classified according to the geographical areas to which they belong.

Worldwide

The Worldwide harmonized Light vehicles Test Procedures (WLTP) is maintained by the UNECE World Forum for Harmonization of Vehicle Regulations [2]. The class determines the power-to-mass ratio of the vehicles. In particular, class 3 is representative of European and Japanese vehicles.

Table 5.1: *Main characteristics of the driving schedules included in the platform.*

Geographical area	Driving schedule	Duration (min)	Distance (km)	Average speed ($\mathrm{km\,h^{-1}}$)
Worldwide	WLTP class 3	30	23.3	46.5
United States	UDDS	23	12.1	31.5
	HWFET	13	16.8	77.5
	FTP	31	17.6	34.1
	IM240	4	3.1	47.1
	SFTP US06	10	12.9	77.2
	SFTP SC03	10	5.8	34.5
	NYCC	10	1.9	11.4
	LA92	24	15.8	39.6
	LA92 short	16	11.1	41.8
European Union	EUDC	7	6.8	58.6
	NEDC	20	8.5	25.4
	ECE R15	3	0.8	16.5
	ArtUrban	17	5.0	17.6
	ArtRoad	18	17.2	57.4
	ArtMw130	18	29.0	96.8
	ArtMw150	18	29.8	99.5
Japan	J1015	15	6.4	25.6

United States

The Urban Dynamometer Driving Schedule (UDDS), the Highway Fuel Economy Test (HWFET), the Federal Test Procedure (FTP), the two Supplemental Federal Test Procedures (SFTP US06 and SC03), the New York City Cycle (NYCC) and the Inspection and Maintenance (IM240) driving cycle are defined by the U.S. Environmental Protection Agency [4]. The UDDS has been developed to test heavy-duty vehicles. All the FTP cycles are variants of the UDDS one, used to test light-duty vehicles in urban scenarios. For example, the SFTP US06 represents aggressive, high speed and/or high acceleration driving behaviours, whereas the SFTP SC03 also takes into consideration the engine load due to the use of air conditioning. Another cycle useful to test urban profile is the NYCC, whereas the HWFET is used for simulating highway scenarios. The IM240 is usually employed for emission testing. The LA92 and the LA92 short (which consists of the first $969\,\mathrm{s}$ of the LA92) are developed by the California Air Resources Board [1]. They are mostly used for simulating urban driving.

European Union

The New European Driving Cycle (NEDC), the Extra-Urban Driving Cycle (EUDC) and the Economic Commission for Europe urban driving cycle (ECE R15) are maintained by the United Nations Economic Commission for Europe (UNECE) [3]. The Urban cycle (ArtUrban), the Rural road cycle (ArtRoad) and the Motorway cycles (ArtMw130 and ArtMw150, with a maximum speed of 130 and 150 km/h, respectively) are included in the Common Artemis Driving Cycles (CADC), developed within the European Artemis (Assessment and Reliability of Transport Emission Models and Inventory Systems) project.

Japan

The 10-15 cycle (J1015) had been used in Japan for emission and fuel economy testing of light duty vehicles.

5.2.2 Electric vehicle model

The EV model consists of three blocks, as shown in Figure 5.2, which are executed with a $100\,\mathrm{ms}$ integration time step. This value is considered sufficient to capture the system dynamics of interest. The battery current I_b is computed as the ratio of the electric power P_e to the battery voltage V_b, given by the sum of the cell voltages $\boldsymbol{V}$ computed by the Battery Model block. P_e is generated by the Electric Traction block starting from the speed values v coming from the Driving Schedules block according to the selected driving cycle. The Battery Model block also provides the vector of the ECM parameters $\boldsymbol{p}$ and the SOC value for each battery cell, which constitute the reference values to be compared to the estimates computed by the BMS under test. Finally, the Sensor Model block generates a noisy version of the battery current and cell voltages, $\hat{I}_\mathrm{b}$ and $\hat{\boldsymbol{V}}$, respectively, which form the input of the BMS under test. A detailed description of each block of the implemented EV model is reported below.

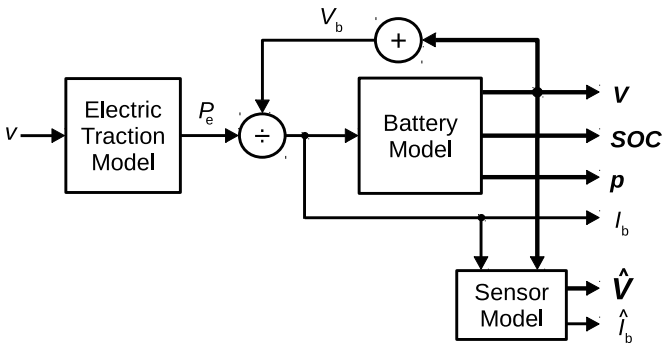


Figure 5.2: *Block diagram of the EV Model.*

5.2.3 Electric traction model

The electric traction model block simulates an EV travelling on a flat road by using the simplified dynamic model of an EV presented in [13, 142]. The use of this model allows us to obtain a likely realistic value of the electric power starting from the vehicle speed profile. The mechanical power P_m is computed by (5.1), where the two frictional forces are due to the air and rolling resistances.

$$P_\mathrm{m} = Fv = \left(M\dot{v} + \frac{1}{2}\rho_\mathrm{air} S C_\mathrm{X} v^2 + \alpha_\mathrm{R} M g \right) v \tag{5.1}$$

The symbols used in this equation are defined in Table 5.2.

Then, the electric power P_e is obtained from P_m by using (5.2), in which two different energy efficiency values are taken into consideration, η_wheel for the traction and η_reg for the regenerative braking. They are considered as constant values instead of a function of the utilization regime in order to not increase the model complexity.

Table 5.2: *Electric traction model parameters for Nissan Leaf.*

Symbol	Description	Value
M	Kerb weight	$1525\,\mathrm{kg}$
S	Frontal area	$2.27\,\mathrm{m}^2$
C_X	Drag coefficient	0.29
α_R	Rolling resistance	0.01
ρ_air	Air density	$1.2\,\mathrm{kg\,m}^{-3}$
g	Gravity acceleration	$9.82\,\mathrm{m\,s}^{-2}$
η_wheel	Efficiency from battery to wheels	0.7
η_reg	Efficiency from wheels to battery	0.5

$$P_\mathrm{e} = \left(\frac{1}{\eta_\mathrm{wheel}} \frac{1 + sgn(P_\mathrm{m})}{2} + \eta_\mathrm{reg} \frac{1 - sgn(P_\mathrm{m})}{2} \right) P_\mathrm{m} \tag{5.2}$$

The default model parameters used in (5.1) and (5.2) are chosen to resemble a commercial electric car (Nissan Leaf) and they are reported in Table 5.2 [13]. These parameters can easily be changed by the user to fit the model to other vehicles.

5.2.4 Battery model

The battery model is capable of simulating a battery composed of M modules, each of them consisting of K series-connected cells, for a total of $N = MK$ cells. The only input is the battery current I_b, which is the same for all the series-connected cells, as shown in Figure 5.3(a).

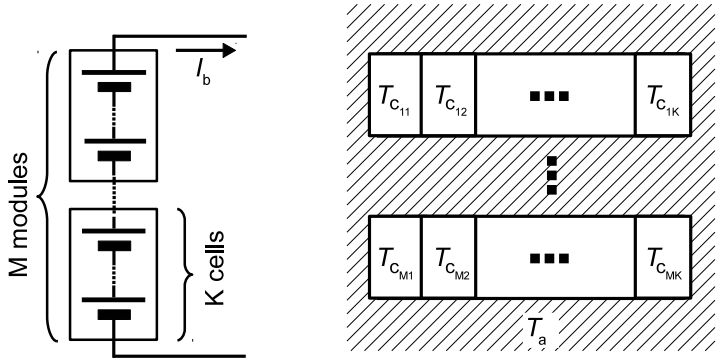


Figure 5.3: *Electrical (a) and thermal (b) organisation of the simulated battery.*

The cell model is composed of two main parts, *i.e.*, the ECM and the Cell Thermal Model (CTM), and their interaction is shown in Figure 5.4. At each time step, the N instances of the ECM generate the cell voltage and SOC value arrays, as well as the present values of the model parameters and, at the same time, the temperature of each cell is determined by the CTM. This allows the simulator to track the temperature distribution inside the battery pack.

The modules are assumed to be thermally isolated to each other in the battery pack (see Figure 5.3(b)). Moreover, the K cells contained in a module are placed side by

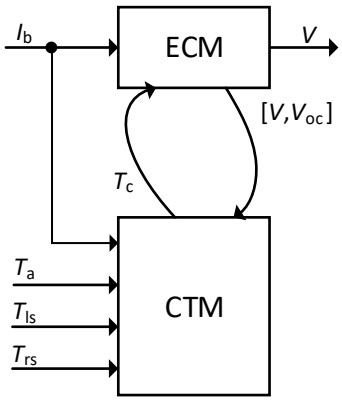
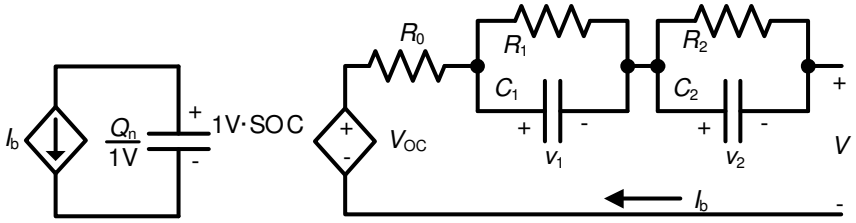
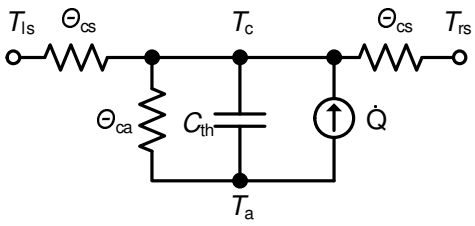


Figure 5.4: *Integration of the electric model with the thermal one.*

side, so each CTM interacts with the nearest two cells only, in order to simulate the cell-to-cell heat transfer. The inputs of the CTM are the left and right surface temperatures T_{ls} and T_{rs} of the considered cell, the ambient temperature T_{a} and the cell current I_{b}, terminal voltage V, and open circuit voltage V_{OC}, provided by the ECM, while the output is the cell core temperature T_{c}. The surface temperatures of each internal cell are equal to the corresponding surface temperatures of the neighbouring cells.



(a)



(b)

Figure 5.5: *Electric circuit model with two RC branches (a) and electrical equivalent thermal model (b) of the battery cells.*

The battery cell electric model is the ECM previously shown in Figure 2.4 with two RC branches, and reported in Figure 5.5(a). The model with two RC branches allows us to increase the accuracy of the emulated battery, tacking into account more battery phenomena with respect to the model with a single RC branch, as also explained in Section 2.3.2. The higher computational demand can be satisfied thanks to the higher

performance provided by the PC processor with respect to an embedded system. The terminal voltage V is the sum of the open-circuit voltage V_{OC} and a dynamic term, which incorporates the voltage across the internal ohmic resistance R_0 and the double layer (V_1) and diffusion (V_2) effects occurring in a Li-ion battery during charging and discharging. As the parameters $[R_0, R_1, R_2, C_1, C_2]$ depend on temperature, SOC and current, their values are stored in 3D LUTs. As far as the open-circuit voltage is concerned, the dependency on SOC is only considered, while the capacity is kept constant during the tests.

The temperature of each cell is determined by the CTM, based on the thermal models developed in [12, 64, 130]. The CTM is an electrical equivalent model of the thermal system, as shown in Figure 5.5(b). It considers the heat exchange between the cells and the external environment and also the cell-to-cell exchange in the battery module. This rather simple thermal model takes into account the main thermal exchanges while maintaining the computational load affordable, also when the number of cells becomes large. The thermal resistances Θ_{cs} modelling the core-to-surface thermal paths have the same value. The cell-to-air thermal path is modelled by the thermal resistance Θ_{ca}. The cell core temperature T_{c} is represented by the voltage at the top terminal of the thermal capacitance C_{th}. The current generator $\dot{Q}$ models the cell self-heating [45]. It is the sum of a reversible contribution, the entropic heat flow, and an irreversible part, due to the ohmic losses inside the cell, as shown in (5.3).

$$\dot{Q} = I_{\mathrm{b}} T_{\mathrm{c}} \frac{\partial V_{\mathrm{OC}}}{\partial T_{\mathrm{c}}} + I_{\mathrm{b}}(V_{\mathrm{OC}} - V) \tag{5.3}$$

Pulsed current tests were performed at different temperatures and pulse amplitudes on a $1.5\,\mathrm{A\,h}$ NMC cell to extract the default ECM parameters that fill the LUTs [12]. The measured parameters are scaled to represent a cell from the same technology but with different capacity, by proportionally scaling the parameter values with the cell capacity, inversely for the resistive elements and directly for the capacitive ones [13]. The different current and voltage profiles measured during these tests have been also used in the cell thermal characterisation in order to obtain the default thermal parameters. The different temperatures are set by using an ad-hoc designed and implemented thermostatic chamber provided with temperature sensors [12]. The initial value of the thermal parameters is optimized with a recursive least square method by using the MATLAB framework [84]. However, different types of cells can be simulated by simply using data provided by characterisation tests performed on other chemistries.

The default cell model has then been validated by using the stepwise current test shown in Figure 5.6. The current measured in the real test is used as input of the model, configured to simulate only one cell. We note in the figure that the predicted cell voltage is in very good agreement with the measured one, resulting in maximum and rms errors of $149\,\mathrm{mV}$ and $20.4\,\mathrm{mV}$, respectively. Good results are also achieved for the cell temperature prediction, as the maximum and rms errors are $1.7\,^{\circ}\mathrm{C}$ and $0.48\,^{\circ}\mathrm{C}$, respectively.

5.2.5 Sensor model

As described in Section 2.3, in real applications, the measurements are performed by an acquisition system composed by current and voltage sensors and one or more ADCs that

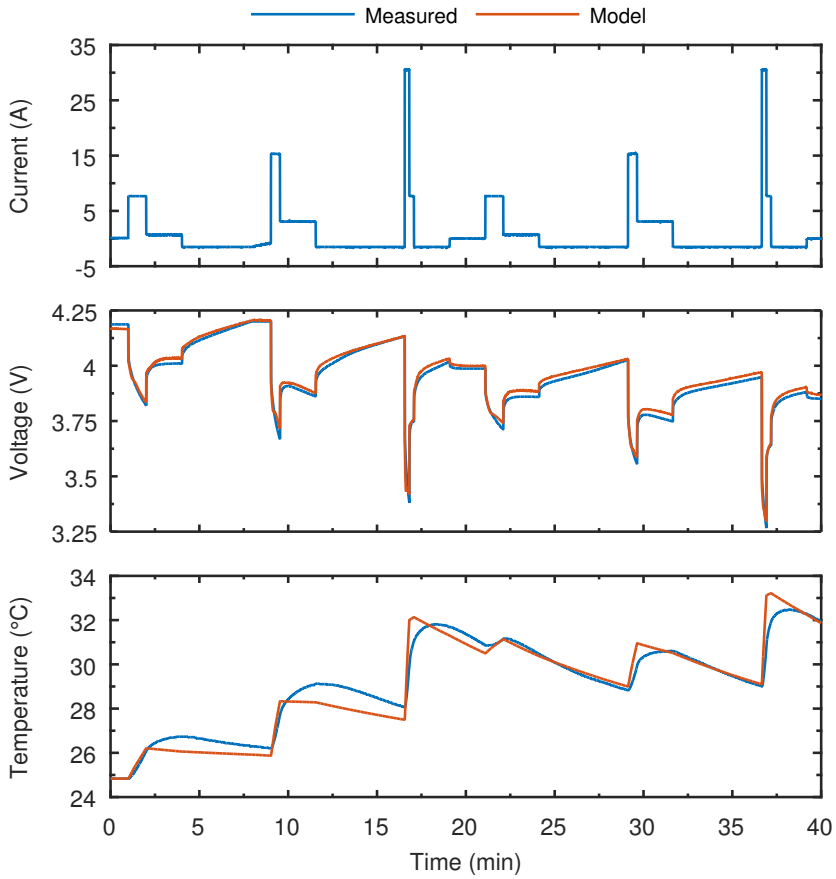


Figure 5.6: *Comparison of the model and measured voltages and temperatures.*

convert the analog signals into digital values. As this process introduces measurement errors and noise, a model of the acquisition system that considers these phenomena has been developed to check the algorithm performance in a real system.

This model takes as input the current I_b and the cell voltages $\boldsymbol{V}$ generated by the battery model and provides as output the acquired current $\hat{I}_\mathrm{b}$ and voltages $\hat{\boldsymbol{V}}$, as shown in Figure 5.2. Measurement noise and offset errors are added to the simulated quantities, then converted into digital values by means of an ADC with a given number of bits and an ideal characteristic. The offset and standard deviation of the measurement noise, as well as the ADC bits are independently configurable for the voltage and current acquisition channels. In this way, the real current and voltage sensors can individually be modelled in accordance with their characteristics.

5.3 Developed Simulink Library

The models described in the previous section have been implemented as configurable Simulink custom blocks and grouped in the library shown in Figure 5.7, which is available online [133].

The Battery Electric model implements the series connection of the ECMs, where the number of cells can be configured by the function block parameters window. The inputs of this block are the battery current and the array of the cell temperatures provided

with the Battery Thermal block. The latter is organised as described in Figure 5.3(b), implementing a configurable number of Module Thermal blocks which emulate the temperature exchange in a module composed by a certain number of CTMs.

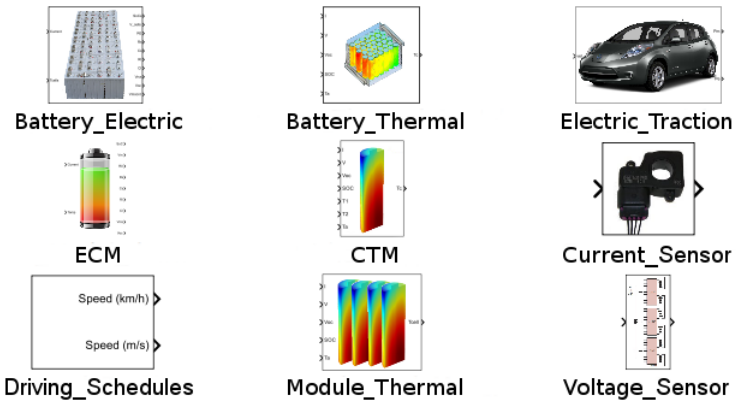


Figure 5.7: *Developed Simulink library.*

The possibility to connect the blocks in different combinations and to change the configurable parameters allows the user to create a simulation platform specific to the BMS under test. For example, the user can choose to not include the thermal model or the model of the sensors. This allows the reduction of the platform complexity and hence of the simulation time.

5.4 HiL platform for validating hardware estimators

5.4.1 BMS hardware

The BMS under test is based on the Intel MAX® 10 FPGA (10M50DAF484C6GES device) introduced in Section 4.3. This BMS has been used to validate the hardware implementation of the AMA and DEKF estimators and to assess their performance without any power path. Besides the estimators, the Nios II soft-core processor and a JTAG module are hardware programmed into the FPGA to allow the communication with the Simulink framework (Figure 5.8). These peripherals are provided with an Avalon-MM interface in order to be connected to the other components of the system via the FPGA Avalon Bus.

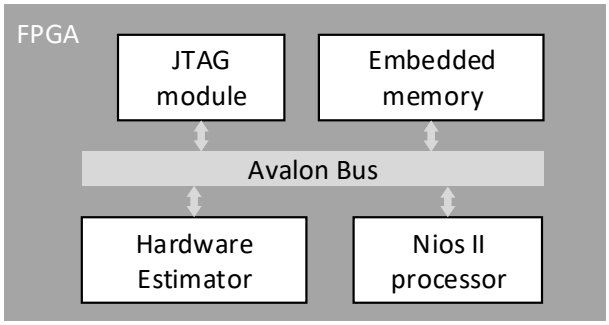


Figure 5.8: *Block diagram of the FPGA-based BMS hardware.*

5.4.2 HiL platform instance

The user interface of the developed HiL platform is visible in the photograph of Figure 5.9, the bottom of which also shows the FPGA-based BMS board.

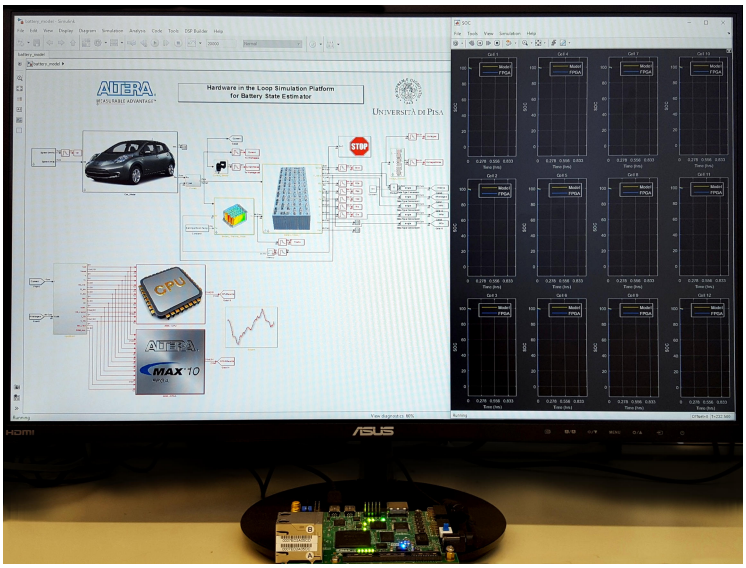


Figure 5.9: *Hardware-in-the-loop platform in action.*

The library blocks shown in Figure 5.7 have been used to assemble a platform able to simulate the Nissan Leaf traction battery. This battery has the same structure described in Section 5.2.4, and it has been build by using the Battery Electric and the Battery Thermal blocks of the library. The block parameters have been changed in order to emulate 96 series-connected NMC cells with a capacity of $66.2\,\mathrm{A\,h}$, obtaining a battery with a nominal voltage of $355.2\,\mathrm{V}$. From a thermal point of view, there are 8 modules of 12 series-connected cells. All the executed simulations consist of the repetition of one of the driving schedules listed in Table 5.1, until the SOC battery reaches $20\,\%$, starting from $80\,\%$. The length of the moving window in the AMA is set to $90\,\mathrm{s}$ and the noise covariance matrices in the DEKF have been reported in Table 3.1.

Table 5.3: *Platform single step execution phases during the test of the AMA estimator.*

Step phase	Execution time per cell
1 - Computation of the battery state and electrical quantities	$208\,\mu\mathrm{s}$
2 - Sending data to the estimator	$2.3\,\mathrm{ms}$
3 - Algorithm execution	$35\,\mu\mathrm{s}$
4 - Reading data from the estimator	$3.53\,\mathrm{ms}$

The simulations have been executed on a computer with an Intel® Core™ i7-4790 processor and $16\,\mathrm{GB}$ of RAM. The platform employs about $6\,\mathrm{ms}$ per cell to execute a single simulation step, with the AMA estimator. This time includes the computation of the cell state and electrical quantities, the sending of the simulated data to the FPGA, the execution of the algorithm on the FPGA and the receiving of the computed results. Table 5.3 shows that the limiting factor of the platform performance is the time

employed by the communication layer. These considerations are the same also for the DEKF algorithm. In this platform instance, the interface between the Simulink framework and the BMS hardware is obtained with a communication layer mapped on the JTAG link. The software that allows the use of this link is contained in the System Console API provided by Intel. This is a set of commands which allows the real-time interaction between the MATLAB environment and the FPGA thanks to the use of a memory mapped interface.

The implementation validation of both estimators is carried out by comparing the results coming from the FPGA and those obtained from the software-executed algorithms. Various scenarios have been simulated in order to demonstrate the capability offered by the HiL platform in assessing the algorithm performance and to analyse and compare the behaviour of the two algorithms.

5.4.3 Noiseless acquisition system

The algorithms have first been tested in case of noiseless sensors, in which the acquisition system only introduces the quantisation error. Here, the simulations are performed on identical cells that all start from the same SOC. For this reason, all the battery modules behave in exactly the same way. As an example, Figure 5.10 shows the simulation results in SOC estimation for one of the two central cells in a battery module during the UDDS cycle. The estimated SOC is in good agreement with the reference one computed by the HiL battery model. The estimate is less accurate in the SOC range from $50\,\%$ down to $25\,\%$ because the SOC is poorly observable from the cell voltage there due to the V_{OC}-SOC curve that is almost flat [91].

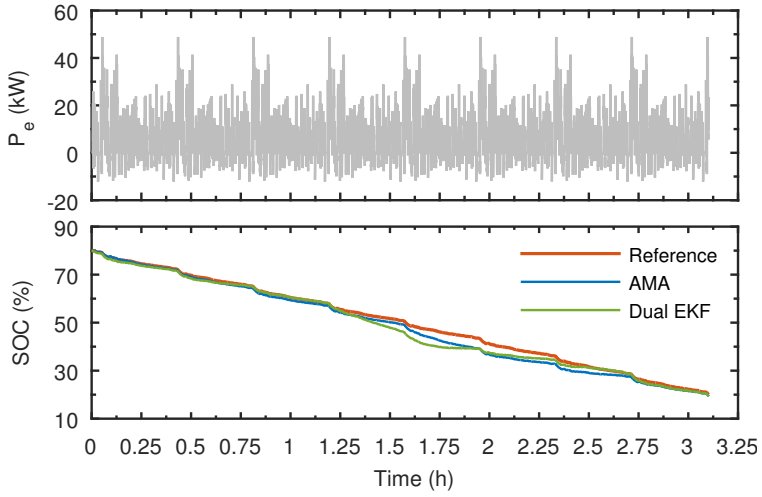


Figure 5.10: *Behaviour of SOC for one of the two central cells of the first module, during a UDDS test.*

The temperature of all the cells has been calculated by the thermal model, starting from a steady-state of $10\,^{\circ}\mathrm{C}$, which is the value chosen for the ambient temperature. This ambient temperature value emphasises the cell parameter variations induced by self heating. Figure 5.11 shows the end of test temperature of the 12 cells contained in a module. We note that the external cells show a temperature lower than the others as expected, because of their larger heat dissipation to the ambient. Furthermore, the higher heating of the central cells with respect to the external ones allows us to observe

the temperature induced variations of the battery parameter values and to test the capability of the algorithms in tracking different parameter changes during the same test.

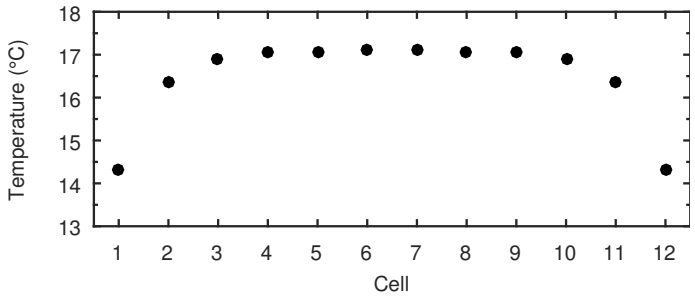


Figure 5.11: *Final temperature of each cell in one module in the UDDS test.*

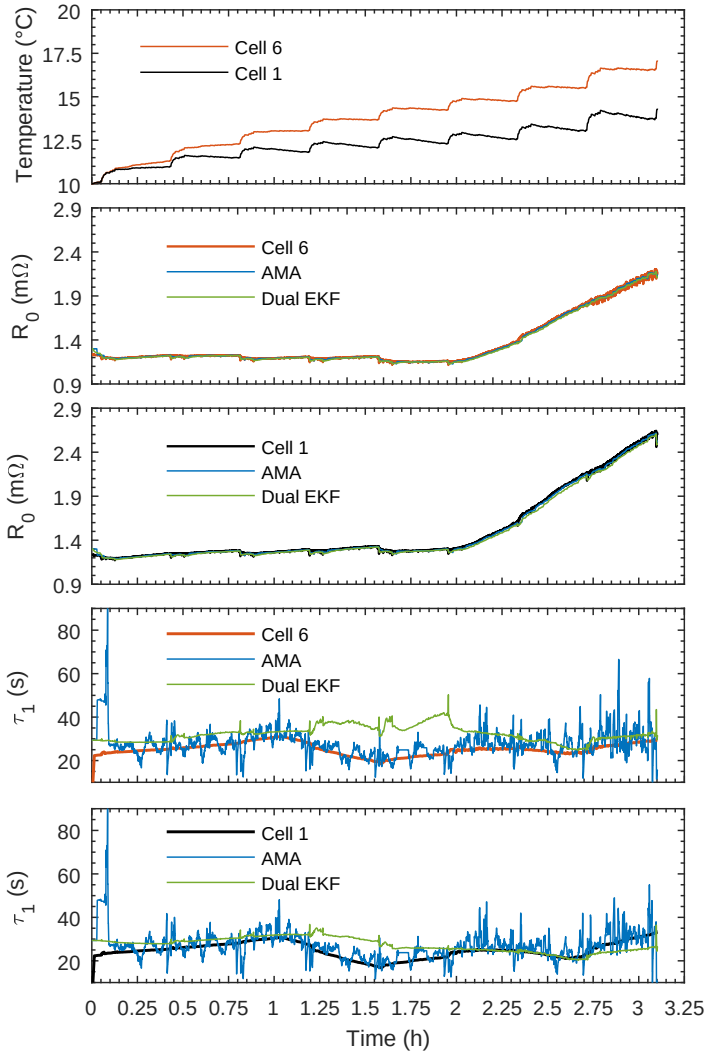


Figure 5.12: *Temperature and ECM parameters for one of the two central cells and one of the two external cells of the first module, during a UDDS test.*

Figure 5.12 shows the temperature and the identified ECM parameters for one of the two external cells (cell 1) and one of two central cells (cell 6) in the module. It can be observed that both estimators well identify the series resistance R_0, also in the last part of the discharge when it changes significantly because of the variation of the electrolyte resistance due to the variation of the ionic concentration [66]. This is a remarkable achievement, as R_0 significantly affects the model accuracy and is a good figure of battery ageing, as explained in Section 2.3.2. The time constant $\tau_1 = R_1 C_1$ of the single RC branch in the ECM is compared to the values used in the HiL battery model (the fastest time constant is considered). We can note that its estimation appears to be more noisy with respect to the series resistance estimation because the battery model response is less sensitive to this parameter [93].

Moreover, the presented platform has been used to test the algorithms implemented on the FPGA by using a wider set of driving cycles in different operating conditions. The maximum and rms SOC errors are reported in Figure 5.13 (empty markers), where the driving schedules are sorted according to their average speed. The rms error is always below $5.1\,\%$ and $2.9\,\%$ for the AMA and DEKF, respectively, indicating that a good SOC estimation is achieved by both estimators in all the driving cycles.

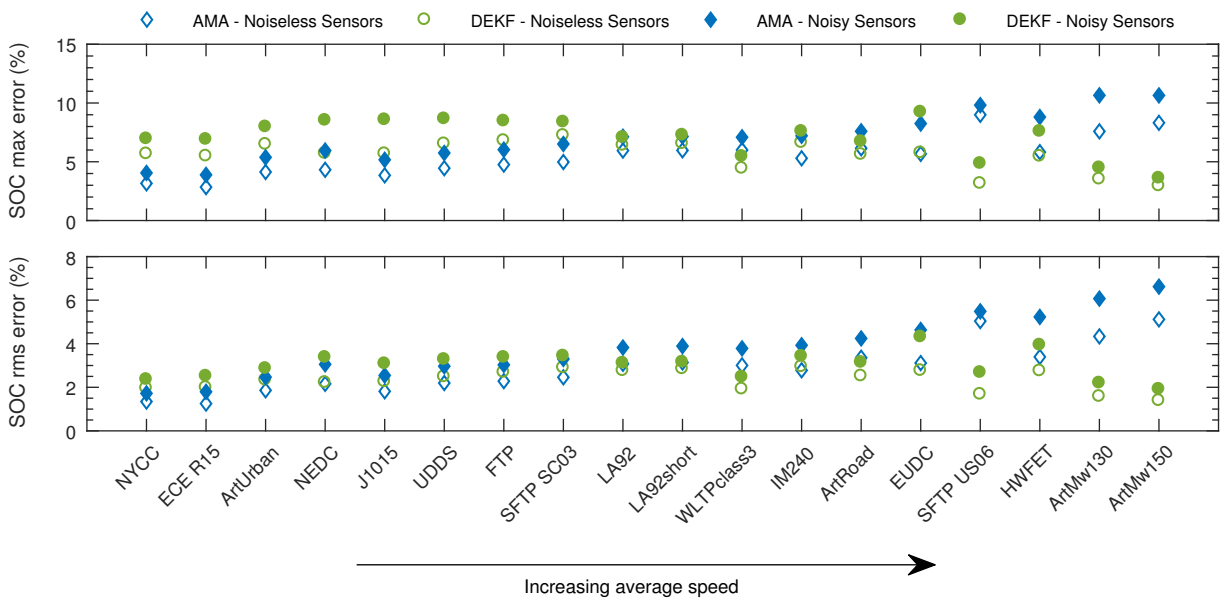


Figure 5.13: *SOC estimation errors for the AMA and DEKF algorithms. The empty markers refer to the case of noiseless sensors, whereas the filled markers to the case of noisy sensors (with standard deviation of* $2\,\mathrm{A}$ *and of* $5\,\mathrm{mV}$ *on the current and voltage measurement, respectively).*

Finally, we note that the properties of the driving cycle and thus of the corresponding electric power and dynamic affect the performance of the state estimator. In fact, the AMA has better results on urban cycles, whereas the DEKF is slightly more reliable for motorway driving schedules where the requested electric power is higher, and the speed profile is usually more static. The algorithms behaviour is attributable to the settings used in these simulated tests.

5.4.4 Noisy acquisition system

As the implemented platform is able to simulate non-ideal sensors, it can also be used to analyse the algorithm behaviour in presence of measurement noise. In our case study, the platform has been configured to simulate two ADCs with a resolution of 16 bits and a full scale value $FS_{\mathrm{I_b}}$ of $400\,\mathrm{A}$ for the current sensor and FS_{V} of $5\,\mathrm{V}$ for the voltage sensor. Furthermore, the sensor models introduce no offset and a white Gaussian noise whose standard deviation is chosen as a reasonable percentage of the full scale values. We considered a standard deviation of $\sigma_{\mathrm{I_b}} = 2\,\mathrm{A}$ $(0.5\,\%\ FS_{\mathrm{I_b}})$ on the current and of $\sigma_{\mathrm{V}} = 5\,\mathrm{mV}$ $(0.1\,\%\ FS_{\mathrm{V}})$ on the voltage measurements.

The estimation errors obtained in these noisy conditions for every driving schedules are reported in Figure 5.13 (filled markers). Both algorithms show a slight degradation of the accuracy in SOC estimation. However, the degradation is not large because the AMA and DEKF methods are capable of rejecting the noise on the current measurement, while they suffer from some sensitivity to errors on the voltage measurement [43].

5.4.5 Unbalanced cells

The series connection of the cells, together with the strict operating ranges, is critical as it introduces the well-known issue of cell unbalancing [15]. This phenomena can easily be reproduced by the developed HiL platform by using different SOC initial values for the cells. This feature allows us to analyse the response of an estimation algorithm to uncertainty in the initial SOC value in a much more effective way than using PHiL testing platforms.

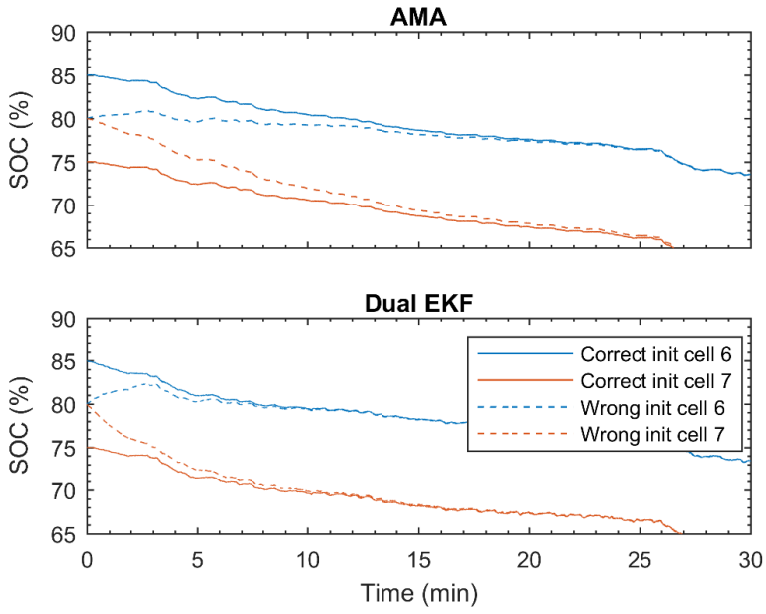


Figure 5.14: *SOC correction after a wrong initialisation compared to the value estimated when the right initialisation value is used.*

To this aim, Figure 5.14 shows the SOC estimation computed by the two algorithms when the SOC of two cells is initialised with different values of $85\,\%$ (Cell 6) and $75\,\%$

(Cell 7) instead of $80\,\%$, as for the other cells in the simulated battery pack. The SOC initialization value of the algorithms are set to $80\,\%$, since the cell unbalancing is an unpredictable state.

However, we can note that both algorithms are able to correct wrong SOC initialisations and that the DEKF is faster than the AMA [91]. Let us consider the wrong estimation to be fully recovered when the difference between the present value and that obtained with the correct initialisation is lower than $1\,\%$. Thus, the DEKF employs $322\,\mathrm{s}$ to recover the wrong SOC initialisation of the cell 6, instead of $798\,\mathrm{s}$ needed by the AMA for the same cell. In any case, the recovering time is small compared to the overall discharge time, which is longer than $3\,\mathrm{h}$.

CHAPTER 6

FPGA-based battery management systems implementation and assessment

6.1 Introduction

In Section 2.4, we have presented two different distributed architectures with an FPGA-based PMU that we have employed to develop two demonstrators. The first one uses the PMU to manage the battery cells organised in modules. As a case study, it has been implemented for the battery of an e-bike that allowed us to validate the BMS implementation and to test the AMA hardware estimator during real utilization on the road. The details of this BMS and the experimental results are reported in Section 6.2. A BMS which employs the concept of smart cells, so called Research BMS, has been implemented within the 3Ccar European project which aims at investigating this new architecture for automotive batteries. In this project, this architecture has been applied to a case study of a $48\,\mathrm{V}$ smart battery. Besides the Research BMS, we also developed a framework to allow the project partners to implement and test their own estimation algorithms. The description of the Research BMS and of the developed framework is reported in Section 6.3. Both the implementations use a device belonging to the Intel Cyclone® V family. These devices are system on chips which are composed of an FPGA used to implement the AMA hardware SOC estimator and a dual-core hardware processor ARM® Cortex™ A9 used to execute the software functions.

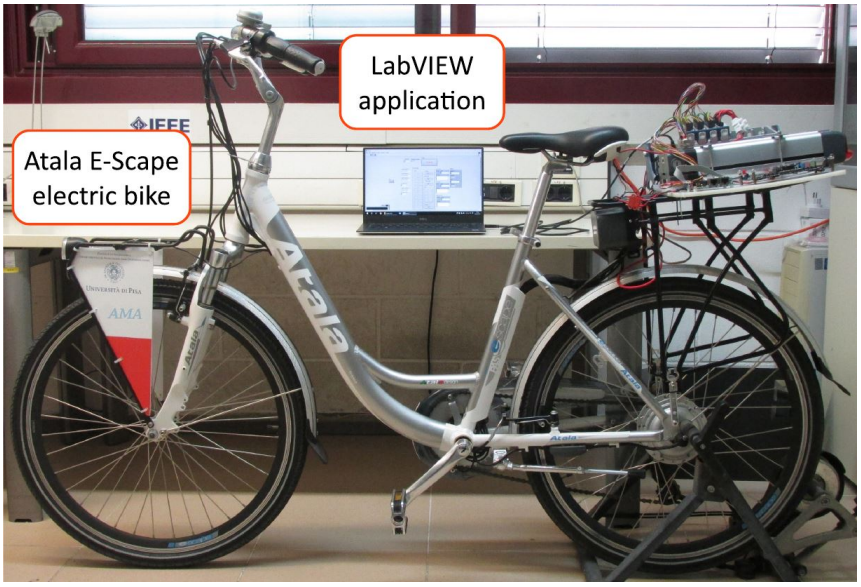
6.2 Basic FPGA-based BMS: e-bike case study

A basic FPGA-based BMS has been built to manage a 10-cell battery pack for an electrically assisted bike. This solution only implements the basic functions of the BMS and allows us to verify the hardware estimator on more than one cell in real road tests. The

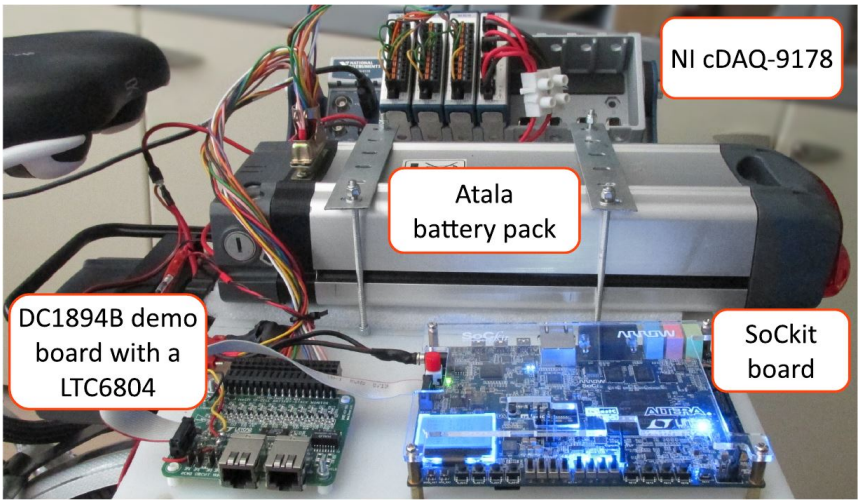
SOC estimation is demanded to the hardware estimator which executes the AMA algorithm in time division multiplexing for all the cells in the battery pack. This is possible thanks to its short execution time ($61\,\mu s$ with a clock frequency equals to $50\,MHz$). The main unit is based on an Intel SoC-FPGA device belonging to the Cyclone® V family which equips the SoCkit development board. The hardware estimator resource usage is reported in Table 6.1.

Table 6.1: *Estimator resource usage implemented on the 5CSXFC6D6F31C6N device.*

Logic utilization (in ALMs)	17244/41910 (41 %)
Variable-precision DSP Block	36/112 (32 %)
Memory bits	723 Kb/5530 Kb (13 %)



(a)



(b)

Figure 6.1: *Photographs of the demonstrator setup.*

Two photos of the demonstrator are shown in Figure 6.1. The electric bike is an Atala E-Scape (see Figure 6.1(a)), a city bike with aluminum frame. The bike is equipped with a brushless DC motor with a nominal power of $250\,\mathrm{W}$. The battery pack consists of 10 Li-ion series-connected cells, yielding a nominal voltage of $36\,\mathrm{V}$ and a rated capacity $10\,\mathrm{A\,h}$. The entire setup, including the battery pack, was arranged on the bike rack, as shown in Figure 6.1(b). A 14-pin connector allows us to access the battery cells' terminals. A ratiometric Hall effect current sensor from Allegro Microsystems is used to measure the battery current. Its full scale range is $\pm 12.5\,\mathrm{A}$, with an internal resistance of $1.2\,\mathrm{m\Omega}$ and a sensitivity of $110\,\mathrm{mV/A}$ (for a supply voltage of $3.3\,\mathrm{V}$). The cells and current sensor output voltages are acquired by a DC1894B demo board. This Linear Technology board is equipped with an LTC6804 battery stack monitor capable of measuring up to 12 series connected cells with a total measurement error of less than $1.2\,\mathrm{mV}$. The LTC6804 is also provided with two auxiliary analog inputs, which are used to accurately acquire the output and supply voltages of the current sensor. All the measurements take $11\,\mathrm{ms}$ to complete.

The battery was preliminary characterized in the laboratory with a PCT profile at $25\,^{\circ}\mathrm{C}$ [91]. The input current and the output voltages during the test are used to determine the AMA initialization parameters, *i.e.*, the V_{OC}-SOC relationship, the capacity of the cells and the initialization values of the ECM parameters. The V_{OC}-SOC curve is considered invariant for every cell and it is shown in Figure 6.2. The extracted cell capacities are very similar to each other and are around the rated value of $10\,\mathrm{A\,h}$, except for the second cell which has a $9.4\,\mathrm{A\,h}$ capacity. These results, together with the average values of the ECM parameters ($R_0 = 24.5\,\mathrm{m\Omega}$, $R_1 = 13.3\,\mathrm{m\Omega}$ and $C_1 = 6945\,\mathrm{F}$), are used to initialize the algorithm implemented on the FPGA. Furthermore, the other algorithm parameters have been tuned by using the procedure described in Section 3.3.3.

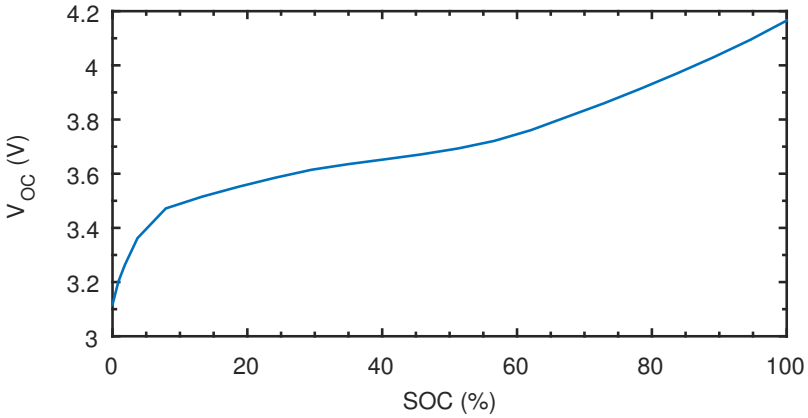


Figure 6.2: *Open circuit voltage V_{OC}, as a function of SOC.*

The Allegro current sensor, the DC1894B and the SoCkit boards form a basic BMS, useful to test the hardware estimator. Figure 6.3 shows a block diagram of it. When the board is turned on, the Linux Ångström distribution, based on a real-time kernel, is booted from a partition on the Micro SD card and the main application is launched. The application manages the acquisition of the cell data, their logging and the communication with the user interface. The current and voltage samples are read from the battery

stack monitor connected to an SPI port. The samples are then sent to the estimator, implemented on the FPGA, via the Avalon-MM interface, and the computation is started. The operation is cyclically executed every $0.1\,\mathrm{s}$, yielding an acquisition sample rate of $10\,\mathrm{Hz}$. When all the cell states are computed, the application saves the samples and the estimation results in a log file, which can be downloaded directly from the SD card or by using the Secure Copy Protocol (SCP) through the Ethernet port. The user interacts with the system via the push buttons on the board. The main battery information, such as the battery SOC, voltage, and current, are shown on the LCD display and LEDs on the development board. The bike acceleration, speed, and position useful for a better analysis of the data, are captured by a GPS unit.

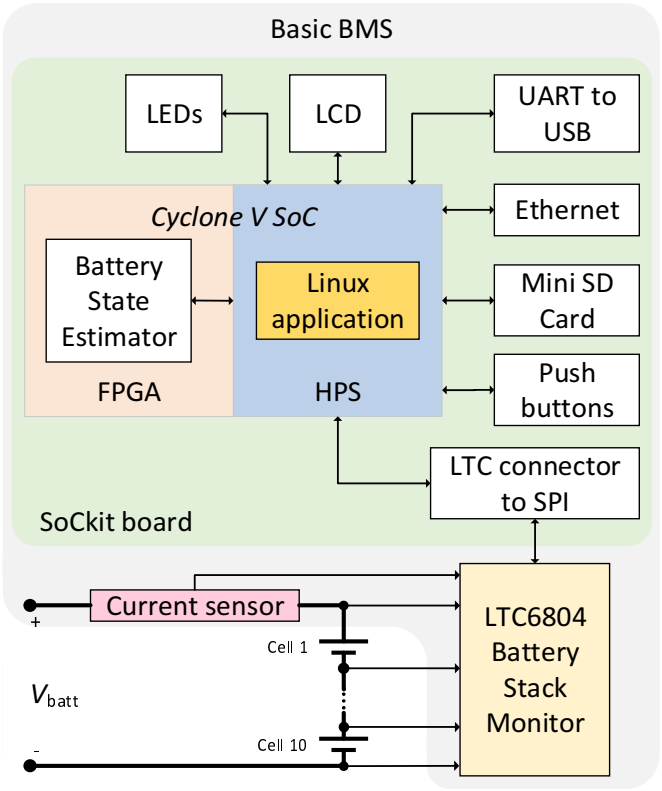


Figure 6.3: *Basic BMS block diagram.*

6.2.1 Validation methodology

The implemented SOC estimator is validated in field tests, *i.e.*, riding the e-bike as in every day use. To evaluate the SOC estimation error, the SOC estimate needs to be compared with a reference value [10]. The common approach is to use the CC technique to obtain the SOC reference value, as shown in (6.1) that is equal to Eq. (2.1) (it has been reported here for simplicity).

$$SOC_{\mathrm{ref}}(t) = SOC_0 - \frac{1}{Q_{\mathrm{n}}} \int_0^t i_{\mathrm{L}}(\tau) d\tau \tag{6.1}$$

As said in Section 2.3.2, Eq. (6.1) can provide a reliable SOC reference SOC_{ref}, assuming that the initial SOC value SOC_0, the normalization capacity Q_{n} and the battery

current i_{L} are known precisely.

The battery current should therefore be acquired by a highly accurate current sensor, different from the one used within the BMS, which feeds the SOC estimator. This is an important aspect, often neglected in literature, which makes it possible to assess the robustness of the estimator against errors in the current measurement effectively. In this work, i_{L} in (6.1) is measured by a National Instrument (NI) 9227 module mounted on a NI cDAQ-9178 chassis. The latter is also equipped with three NI 9215 modules, which are used to acquire the voltage of the 10 cells and to verify the LTC6804 stack monitor behavior (see Figure 6.1(b)). The cDAQ-9178 is powered by a supplementary battery and connected to a laptop, carried by the cyclist in a backpack, which runs a LabVIEW application for data acquisition and logging (see Figure 6.1(a)).

SOC_0 is taken equal to $100\,\%$, as tests are started after a full charge of the battery. Q_{n} is assumed to be equal to either the rated battery capacity or the capacity measured in a preliminary characterization test. These assumptions however may lead to inaccuracies in the computation of SOC_{ref}. Inhomogeneities in the cells of the battery pack can cause not all the cells to be fully charged to $100\,\%$ SOC. Moreover, the maximum charge that can be extracted from each cell during a validation test (we refer to this value as the *real* capacity) may differ from that measured during a characterization test, because of different discharge rates and temperature.

To overcome these problems, we propose to combine the CC and V_{OC} estimation techniques for computing SOC_{ref}. These two techniques are used to define two alternative metrics for assessing the SOC estimator errors [10], whereas here they are merged in a unique definition of SOC_{ref}. The basic idea is to set SOC_0 and Q_{n} in (6.1), so that SOC computed by (6.1) coincides with the estimate obtained through V_{OC} at the beginning and end of the validation test. This does not require any specific condition during the execution of the test, apart from the fact that the test starts with the battery in a rested state and ends in an SOC region where it is possible to reliably extract SOC from the V_{OC}-SOC relationship, once the battery has reached the steady state after discharge.

Let $V_{\mathrm{OC}} = f(SOC)$ express the V_{OC}-SOC relationship shown in Figure 6.2. For each cell, we can compute SOC before and after the validation test by inverting the V_{OC}-SOC relationship, *i.e.*, $SOC_0 = f^{-1}(V_{\mathrm{OC}}^{\mathrm{start}})$ and $SOC_{\mathrm{end}} = f^{-1}(V_{\mathrm{OC}}^{\mathrm{end}})$, where $V_{\mathrm{OC}}^{\mathrm{start}}$ and $V_{\mathrm{OC}}^{\mathrm{end}}$ are the cell voltages measured, when the battery is in a steady state, before and after the validation test, respectively. Given the above definition, we can compute the *real* cell capacity related to a validation test, which coincides with the desired normalization capacity to be used in (6.1), by the following equation [48].

$$Q_{\mathrm{real}} = \frac{\int_{t_0}^{t_{\mathrm{end}}} i_{\mathrm{L}}(\tau) d\tau}{SOC_0 - SOC_{\mathrm{end}}} \tag{6.2}$$

6.2.2 Experimental results

Several road tests have been carried out, proving both the functionality of the basic BMS provided with the AMA estimator and the effectiveness of the designed experimental setup. As a representative example, we discuss the results obtained during a validation test, consisting of the repetitions of a cycling period followed by a rest pause. The electric power P_{e} provided by the battery pack and the speed measured by the GPS

data logger during the test are shown in Figure 6.4. First of all, we compute the initial SOC and the *real* capacity of each cell of the battery pack from the data acquired by the NI cDAQ-9178 chassis. Then, we evaluate the corresponding SOC reference using (6.1).

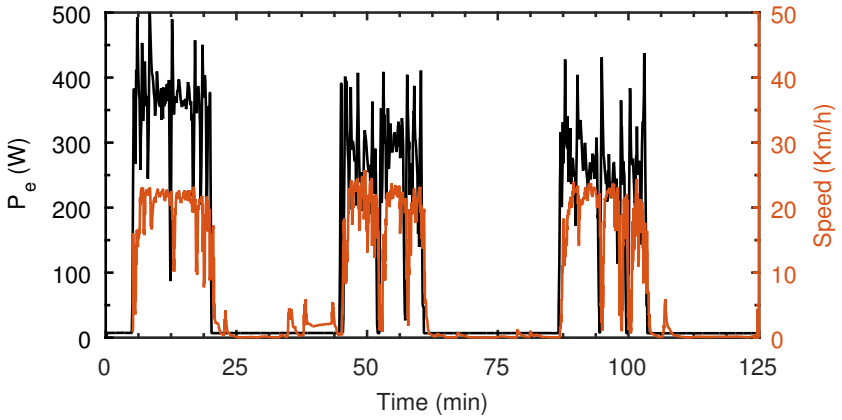


Figure 6.4: *Electric power and speed during the discussed validation test.*

Figure 6.5 shows the *real* cell capacity, for the discussed test, and compares it to the value obtained in the PCT characterization test. Apart from the second cell, which presents a pronounced degradation of the cell capacity, there is only a slight difference between the PCT, rated and *real* cell capacities, as expected by the high Coulombic efficiency provided by the Li-ion battery technology.

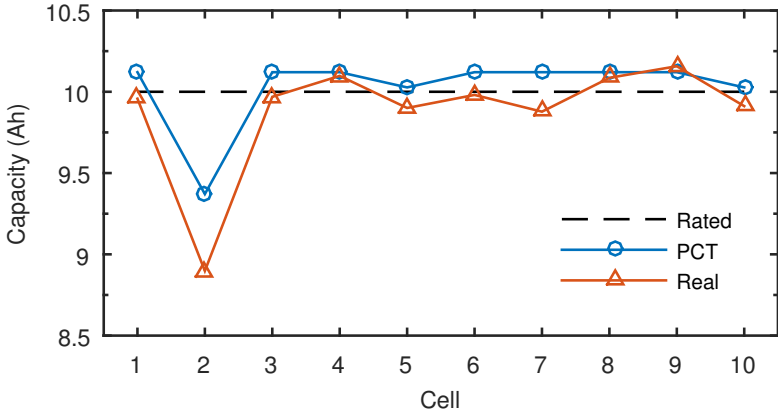


Figure 6.5: *Capacities of the cells extracted using the PCT and the discussed validation test.*

The computed $SOC_{\text{ref}}(t)$ for each cell is reported in Figure 6.6, where it is evident that the SOC of the second cell corresponds to the battery SOC, as the second cell is the first cell to reach the discharge cut-off voltage. With the availability of the SOC reference, we can calculate the estimation error introduced by the AMA. The AMA SOC estimate for the second cell is shown in Figure 6.7 and is very close to the reference value during all the test. On the contrary, the pure uncompensated CC, computed by using the current measured by the BMS, drifts over time. This clearly demonstrates that the AMA is capable of correcting the uncertainties in the current measurement (es-

pecially the offset of the current sensor, as demonstrated in Section 3.3.1) and in the real capacity of the cell.

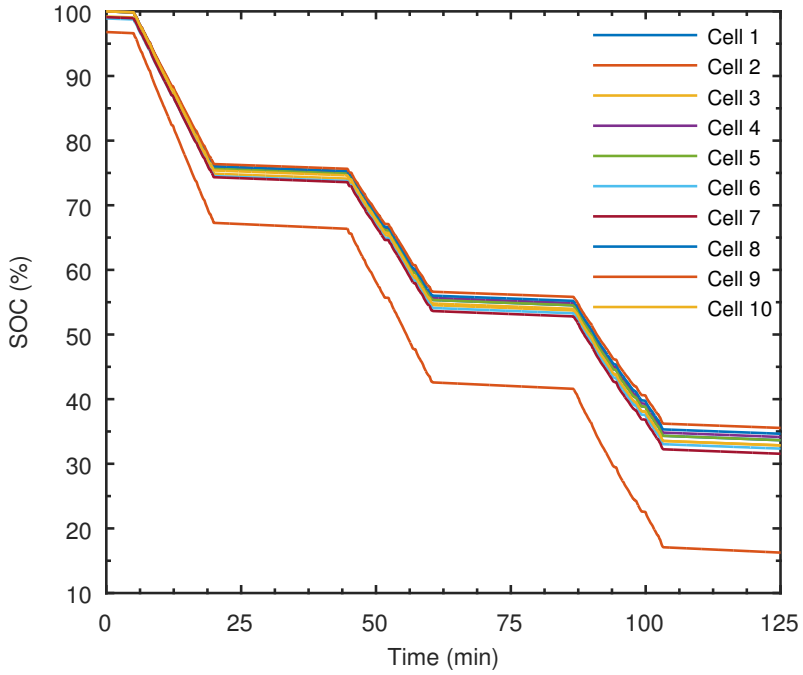


Figure 6.6: *SOC behavior of the 10 cells during the discussed validation test.*

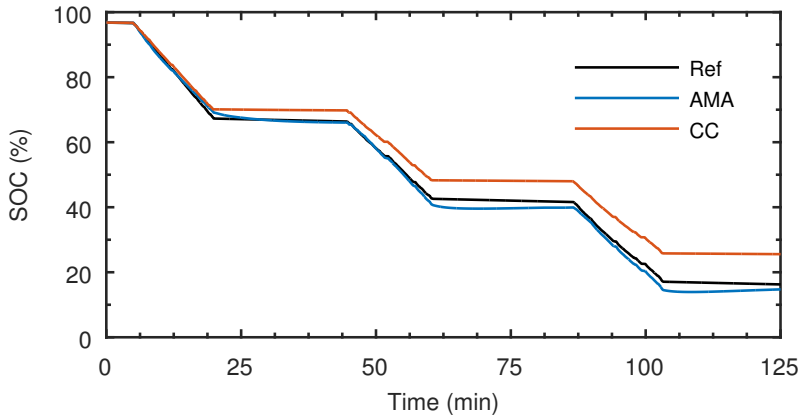


Figure 6.7: *Comparison of the SOC estimation results for the second cell.*

Table 6.2 shows the maximum and rms errors of the AMA and CC SOC estimations. The AMA provides a more reliable estimation than CC, being its rms error below $2\,\%$ and the maximum absolute error around $3\,\%$, for all the battery cells. This is a valuable achievement, as it is obtained in road tests, when the battery is subjected to the real application load. In fact, if we consider the SOC estimation errors reported in [129] for a similar application, AMA outperforms EKF, as the latter introduces a maximum SOC error well above $5\,\%$. Better performance in terms of maximum error is obtained in [129] using an Adaptive EKF, at the expense of higher complexity. It is worth noting that the results reported in [129] refer to laboratory tests and not to road tests.

Table 6.2: *SOC estimation errors.*

	AMA		CC	
	Max (%)	rms (%)	Max (%)	rms (%)
Cell 1	2.4	1.2	2.7	1.5
Cell 2	3.0	1.7	9.3	5.9
Cell 3	3.2	1.7	2.8	1.6
Cell 4	2.2	1.1	3.1	1.8
Cell 5	3.2	1.5	2.4	1.3
Cell 6	3.4	1.7	3.1	1.8
Cell 7	3.1	1.6	3.4	2.0
Cell 8	2.6	1.3	2.5	1.4
Cell 9	2.7	1.3	2.1	1.1
Cell 10	2.5	1.2	3.2	1.9

Finally, Figure 6.8 shows the R_0 and τ_1 ECM parameters identified by the AMA for the second cell. These results refer to a window length of $20\,\mathrm{min}$ and an LS matrix with 30 rows. This implies that the ECM parameters are updated every $40\,\mathrm{s}$. While the R_0 identification seems to be very robust, τ_1 identification is less reliable, especially when the window overlaps a rest period.

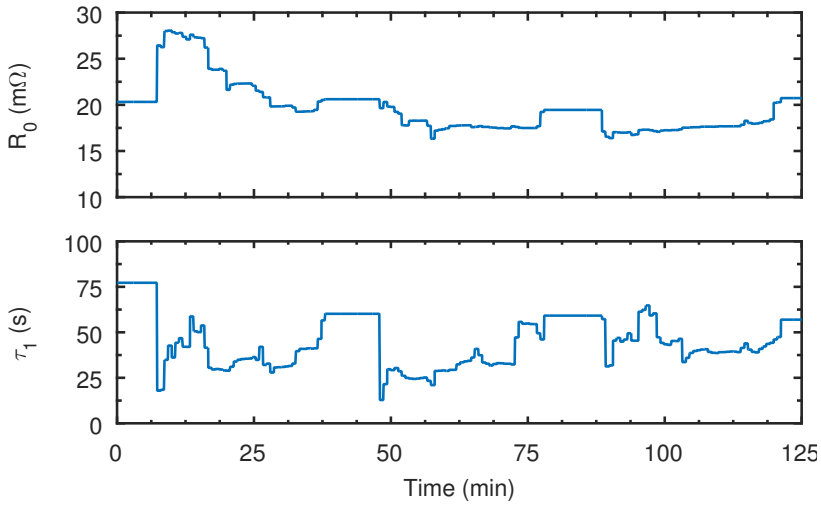


Figure 6.8: *ECM parameters estimated online by the AMA for the second cell.*

6.3 FPGA-based BMS with Smart Cells

A BMS with an FPGA-based PMU and the lower hierarchical level based on smart cells, the so called Research BMS, has been developed within the 3Ccar project, which aims at investigating this new kind of architecture for automotive applications. This section aims at reporting on the hardware and software architecture of the Research BMS implemented for a case study of a $48\,\mathrm{V}$ smart battery system. The latter is composed of 12 SCs that embed electronic circuits and sensing elements to acquire a broad range of cell-related quantities, such as voltage, current, pressure, and impedance, by using the Electrochemical Impedance Spectroscopy (EIS) technique. These data are made available to the Research BMS by means of a Power Line Communication (PLC)

channel. A schematic representation of the smart battery system has been reported in Figure 6.9. The system also integrates a DC/DC converter able to provide a 12 V output voltage. It is a 4-phase synchronous 48 V to 12 V buck converter designed to be integrated in the smart battery housing provided with a heat sink that keeps the DC/DC converter temperature below a safe value under full load condition (1 kW power) without any forced cooling.

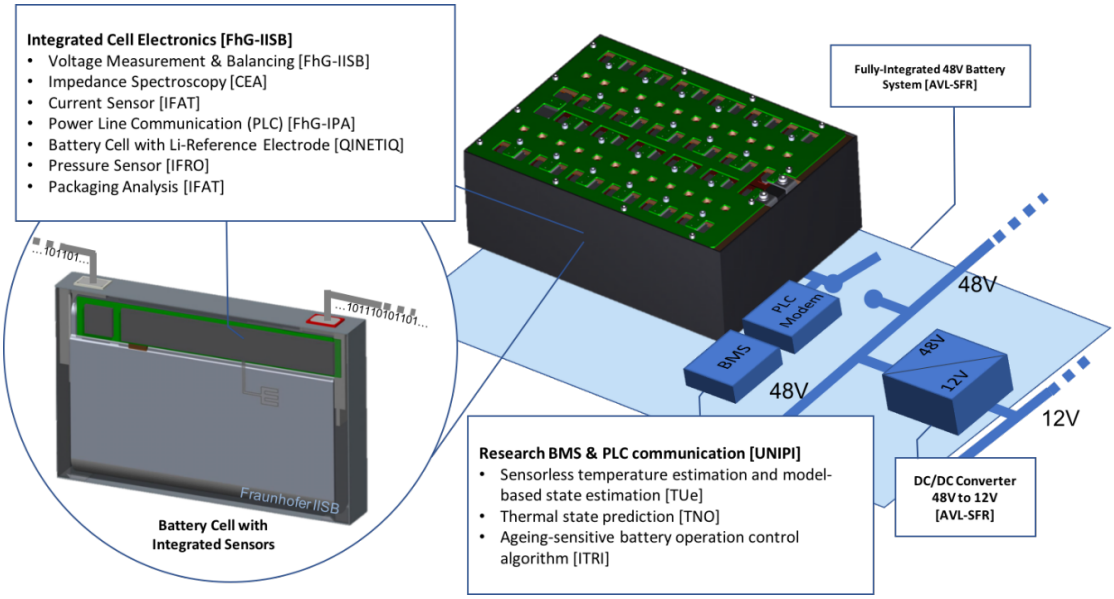


Figure 6.9: *Architecture of the* 48 V *smart battery system.*

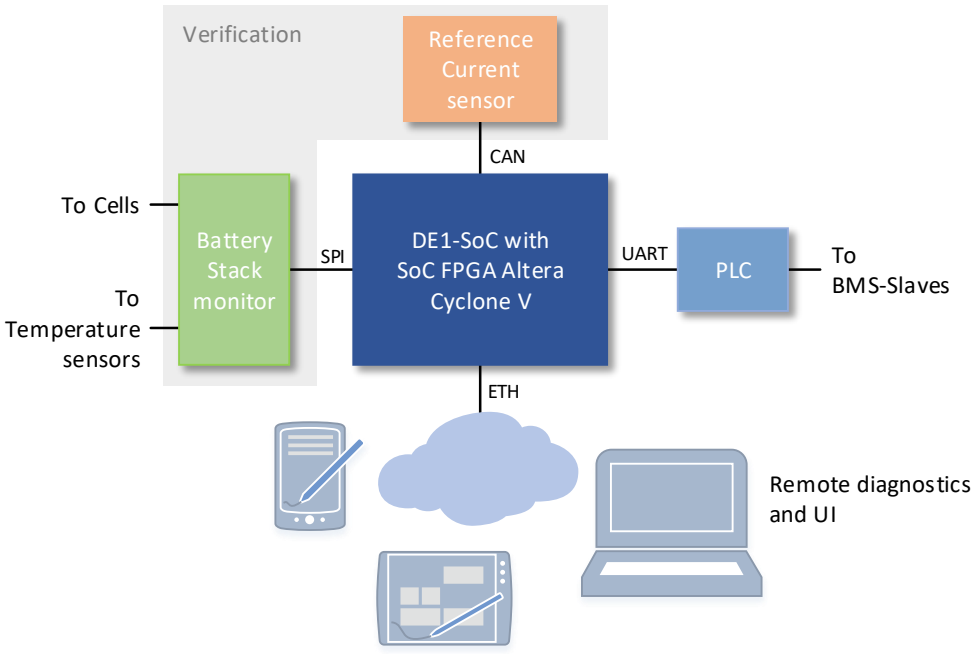


Figure 6.10: *Hardware architecture of the Research BMS.*

The Research BMS must allow the execution of complex algorithms implemented in hardware or in software by the project partners, besides the execution of the basic BMS functions. Our work focused on the development of this BMS, as well as of a framework which allows the partners to develop and test their algorithms. A general representation of the hardware architecture of the Research BMS is shown in Figure 6.10. The main unit is an Intel SoC FPGA of the Cyclone® V family, mounted on a DE1-SoC development board. The use of a development board simplifies the hardware design of the BMS making it possible to focus the efforts on the most innovative aspects of the smart battery system. As said, the devices belonging to this family consist of an FPGA and a dual-core hardware processor. These features provide the required computational resources and the flexibility to implement the estimation and control algorithm effectively. The FPGA fabric can be used to build hardware accelerators in order to execute complex operations in the estimation algorithms. In fact, the system is provided with a hardware co-processor which executes the AMA algorithm for SOC estimation, which has been tuned by using the procedure described in Section 3.3.3. The hardware estimator resource usage is reported in Table 6.3.

Table 6.3: *Estimator resource usage implemented on the 5CSEMA5F31C6N device.*

Logic utilization (in ALMs)	18395/32070 (57 %)
Variable-precision DSP Block	37/87 (43 %)
Memory bits	712 Kb/3970 Kb (18 %)

Using a hardware accelerator for the SOC estimation algorithm, the computational effort is distributed between the hardware accelerators and a processor. In fact, the hardware processor is employed to execute the other estimation algorithms, developed by the project partners, and the basic functions of the BMS. The various software tasks are coordinated by means of a Linux-based OS, whose kernel is stored on a Micro SD card besides to the the software application and the log files.

The FPGA fabric and the processor's peripherals are used to interact with the other components of the Research BMS. In particular, a CAN interface is employed to communicate with the reference current sensor and an SPI interface is used to manage a battery stack monitor. The latter is connected to the board by using a dedicated connector and provides a backup measurement of the cell voltages. The reference current sensor and the battery stack monitor implement the verification function of the Research BMS. The reference current sensor is used to have a reliable current value in order to validate the measured current values of each smart battery cell. Thus, an automotive certified and commercially available shunt based Isabellenhütte IVT-Mod current sensor has been selected. The current sensor is triggered and read by the BMS via CAN. The communication with the SCs through the PLC channel is achieved by means of the PLC modem, which acts as a transceiver, hiding the PLC physical layer by providing a standard UART interface towards the BMS, as well as the SCs.

The CAN and the UART signals are routed by using the two 40-pin expansion headers of the DE1-SoC board which allow the connection to an auxiliary board. The latter has been built to manage the signals routing, especially with the SCs, to connect two NTC thermistors for measuring two temperature points, properly positioned into the battery pack, and to manage the power supply generation and distribution. As an example, the DE1-SoC board and the reference current sensor are supplied by a 12 V

DC input voltage provided by a dedicated DC/DC converter mounted on the auxiliary board, which takes the $48\,\mathrm{V}$ battery voltage as input.

The Research BMS can also exchange data through an Ethernet link. This enables algorithms with a low update frequency to be executed on a remote PC making the tuning of the algorithm easier and more effective, as this procedure can exploit data collected from different batteries. Moreover, the Ethernet link can be used to provide a user interface with which monitoring the BMS behaviour and configuring its parameters, and to download the log files.

The BMS executes its main functions and the estimation algorithms thanks to dedicated software applications. The used OS is the Linux Ångström distribution for ARM platforms based on the LTSI Real Time kernel v3.10 with pre-emption. The use of this OS allows us to establish an SSH or an SCP connection via ETH in order to execute remote commands and to transfer files.

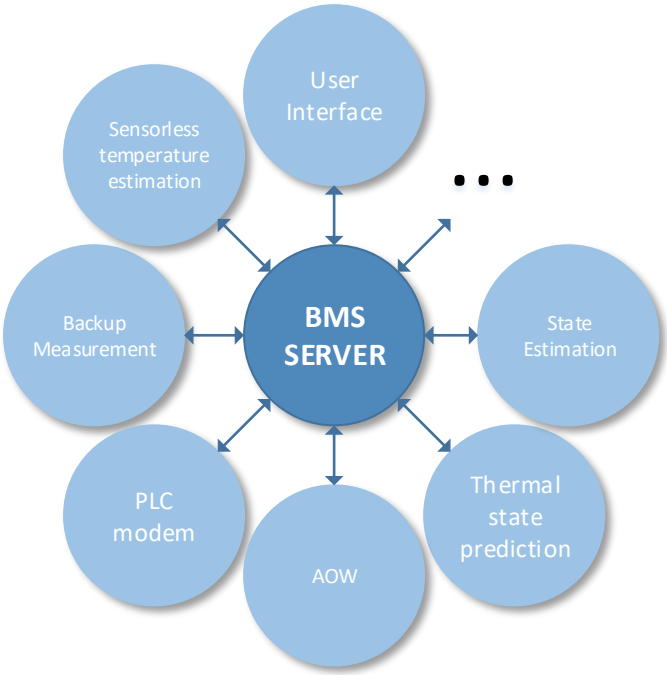


Figure 6.11: *Software organization of the Research BMS.*

The software architecture is based on the client-server paradigm. All the BMS functionalities, including the software estimation algorithms, are developed as separated clients that communicate with a server, called BMS server (Figure 6.11). The latter is lunched as the first process by the OS. The main function of the BMS server is to manage the data flow between the client processes. The communication is based on a Unix domain socket used for exchanging data between processes executing on the same host OS. The input/output values of each client are organised in structures and sent by using the SOCK_SEQPACKET communication semantic, which allows the reliable transmission of ordered datagrams. For example, the Backup Measurement client

manages the acquisition of the cell voltages, temperatures, from the LTC6804, and the current from the Isabellenhütte IVT-Mod current sensor. This process acquires the values every $0.1\,\mathrm{s}$, thanks to a timer and sends them to the BMS server in a dedicated data structure. Furthermore, the partner's estimation algorithms have been implemented as separate clients. The BMS executes the following applications:

- The sensorless temperature estimation algorithm implies deriving the temperature of a cell from its impedance measured at one or more frequencies at the given SOC value of the cell at the moment the temperature is calculated [19].
- The Thermal Prediction Algorithm (TPA) is used to predict the battery core and surface temperature [61].
- The Aging-sensitive Operation Window (AOW) algorithm uses the battery EIS data to determine a battery operation window. The EIS data of a battery will change with respect to its age, either calendar age or cycle age. Therefore, the AOW algorithm will provide an operation window which is aging-sensitive.

Another client application is the User Interface (UI) client that sends all the acquired values and the computed results to a Host PC where a LabVIEW graphical user interface is executed. The UI process has as inputs a fusion of all the structures of the other client processes. The values are converted in strings and sent to the PC by using the client-server paradigm. In particular, this process creates a socket with a static IP address which is used to connect to the LabVIEW UI application. Furthermore, this client also manages various keys and LEDs which have been used to implement a basic user interface of the Research BMS, useful during the debug phase.

6.3.1 Developed framework

The Research BMS has been validated by developing a framework, shown in Figure 6.12, which emulates the smart battery behaviour. The developed framework is mainly composed with two parts, a Smart Cell emulator and the UI, both based on LabVIEW applications.

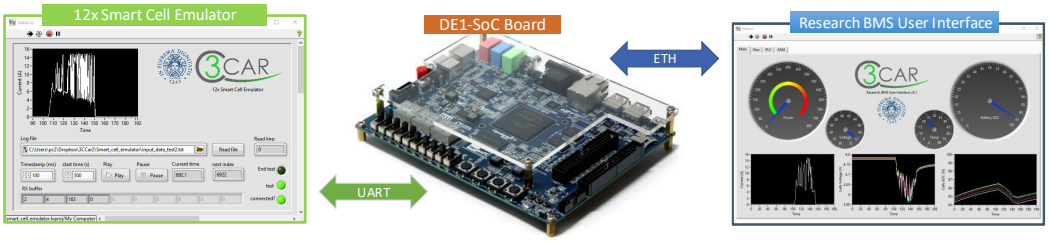


Figure 6.12: *Implemented framework for BMS developing and verification.*

The Smart Cell emulator is used to emulate the communication between the BMS main control unit and the SCs, by implementing the same communication protocol on the UART link. The communication is organised as a master-slave-bus, where the Research BMS is the master and each SC is a slave. The bus features cell addressing as

well as broadcast messages to synchronize, *e.g.*, measurements. Usually, the Research BMS sends commands to the SCs (*e.g.*, trigger voltage measurement) and the SCs send the answers (*e.g.*, measured cell voltage) back to the Research BMS. The cell current, voltage, and temperature samples are stored in a text file which can be selected by using the user interface of the emulator in order to allow us to use different test profiles.

The UI is used for visualising the Research BMS main information, the estimation algorithm results, and for managing the BMS operations. It can be executed on a host PC connected to the Research BMS by using the ETH link. It has been used for both validation and during the final test phases.

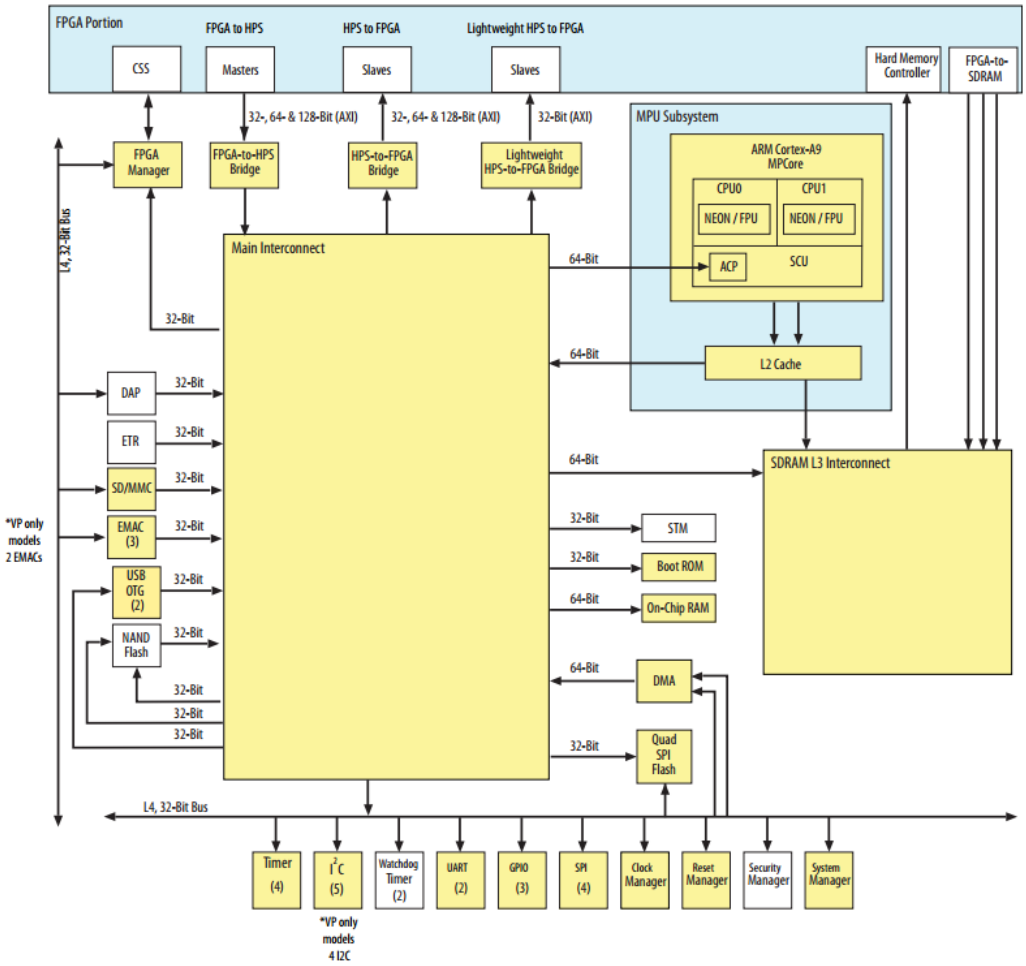


Figure 6.13: *SoC virtual platform block diagram.*

Besides this verification framework, we provided to partners a virtual platform which simulates the SoC-FPGA and reproduces the BMS software applications. It has been used to allow the algorithms to be tested in situations close to the real one. The virtual platform is based on the Mentor Embedded technology and provides the capability to develop and test software for Intel Arria® 10 SoC devices. This family has a structure very similar to the Cyclone® V devices. Figure 6.13 shows the diagram of the structure of these devices, where the blocks highlighted in yellow are modelled in the virtual platform. In particular, the hardware processor is the same of those installed in the

device used for the Research BMS. Thus, the virtual platform allows us also to run an OS, such as the Linux Ångström distribution.

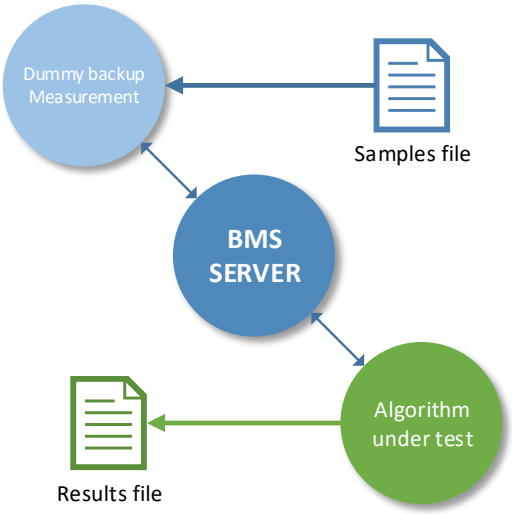


Figure 6.14: *Block diagram of the demo software.*

We also provided a pre-built image of this OS for use with the virtual platform together with a demo software environment which reproduces the final application. It is composed by the BMS server process and a dummy measurement client. The latter is able to read current and voltages samples from a text file and to send them to a consumer, which is the client that executes the algorithm developed by the partner (see Figure 6.14). In this way, the developer has to build an application that receives the inputs from the BMS server, computes the algorithm and saves the data in a log file for validating the algorithm and future processing.

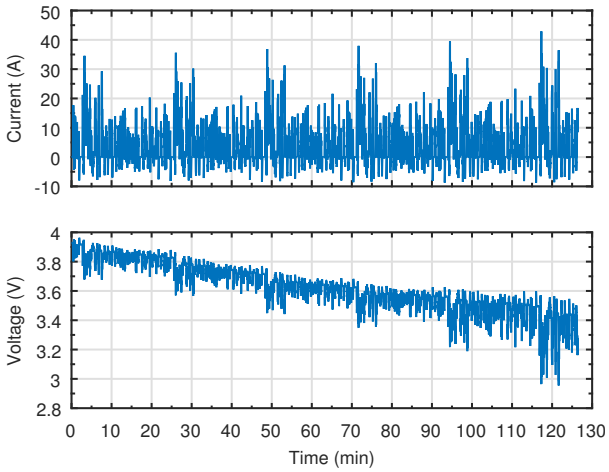


Figure 6.15: *Cell current and voltage profile in five consecutive UDDS cycles.*

As an example, here is reported the validation of the TPA and the AMA hardware estimator implemented on the Research BMS and tested with the setup reported in Figure 6.12, by simulating the battery usage in an automotive environment. The first algorithm is implemented in software by following the procedure described above and

using the provided virtual platform. On the other hand, the AMA is implemented as a hardware accelerator managed by an application client which allows the data exchange with the hardware module. The Smart Cell emulator uses as input a text file containing the current and voltage profiles of the battery cells, which are representative for a battery used in an EV driven in an urban environment, *i.e.*, the UDDS driving schedule described in Section 5.2.1. In particular, the battery starts the test with an SOC equal to $80\,\%$ and the UDDS cycle is repeated until the SOC reaches the value of $20\,\%$. These profiles (see Figure 6.15) are generated by using the battery model including the thermal description of the cells described in Section 5.2.2. The thermal part of the battery model allows us to have a reference for the core temperature prediction, as drawn in Figure 6.16, while the electrical one is used to generate the reference values of SOC and model parameters shown in Figure 6.17.

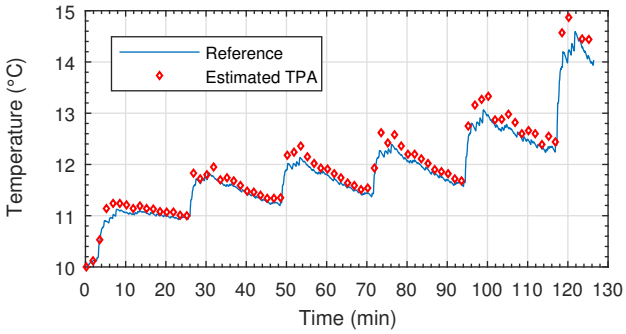


Figure 6.16: *Comparison between the simulated and the predicted cell core temperature.*

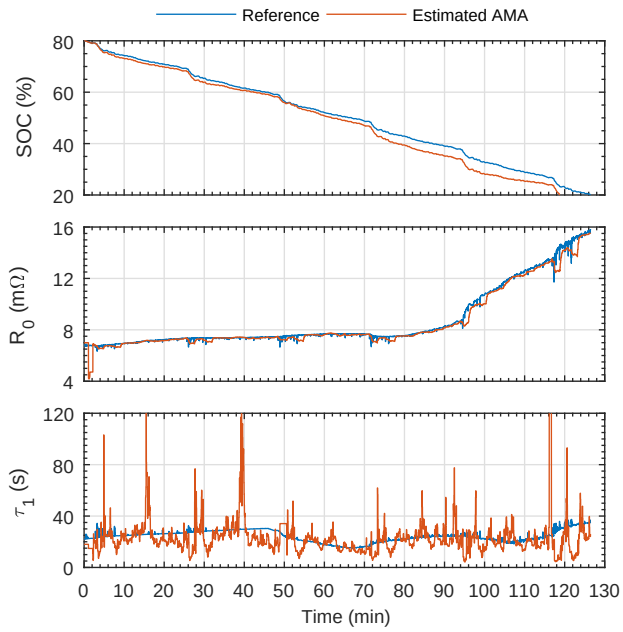


Figure 6.17: *Comparison between the simulated and the estimated cell SOC and model parameters for one of the twelve cells in the emulated battery pack.*

Figure 6.16 shows the temperature predicted by the TPA for one of the 12 simulated SCs. For this test, the update interval has been set to $100\,\mathrm{s}$. From the graph, it is

visible that the predicted temperature is coherent with the simulated one. The results of the AMA are shown in Figure 6.17. The estimated SOC is in good accordance with the simulated one, resulting in an rms error of $2.8\,\%$, and very good results are also achieved in the estimation of the series resistance R_0 and the time constant $\tau_1 = R_1 C_1$, even if the latter is more noisy because the battery model response is less sensitive to this parameter [93].

6.3.2 Experimental results

The final phase of the 3Ccar project focused on the system assembly and a preliminary verification, in order to allow us to validate all the hardware and software components of the smart battery system.

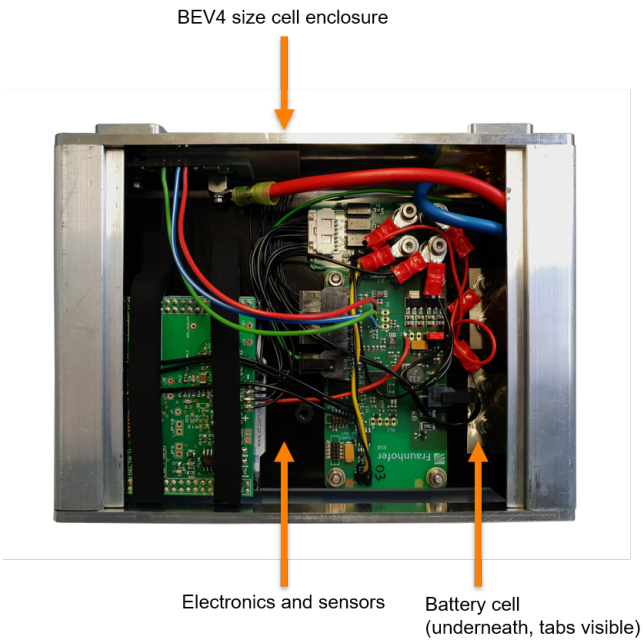


Figure 6.18: *Photograph of the smart cell assembly.*

The SC assembly is shown in Figure 6.18. Each SC comprises a conventional pouch packed Li-ion cell, an MCU with its own communication interface and several sensors into one single BEV4 sized enclosure. To fully take advantage of the SC concept, the battery cell was equipped with several sensors, connected to the local MCU. In particular, cell voltage and reference electrode voltage measurement is performed with the MCU built-in 16 bit ADC that is also used in combination with an analog multiplexer to perform the measurements of six NTC thermistors distributed over the pouch cell surface and on the tabs. In each SC there is also a pressure sensor with an analog interface that detects a possible inner overpressure of the SC, and a current sensor placed in the current path from the pouch cell to the SC enclosure that reads the individual cell current even in parallel connected cells. Moreover, other functionalities have been implemented, such as the power supply circuit (implemented as a boost DC/DC converter) to generate a constant $5\,\mathrm{V}$ output, the impedance measurement system, and a passive cell balancing circuit.

The Research BMS has been mounted on a plate (see Figure 6.19) that has been positioned on the top of the smart battery. In the photograph reported in Figure 6.19 we can identify the three boards that compose the Research BMS: the DE1-SoC development board, the auxiliary board, and the LTC6804 demo board.

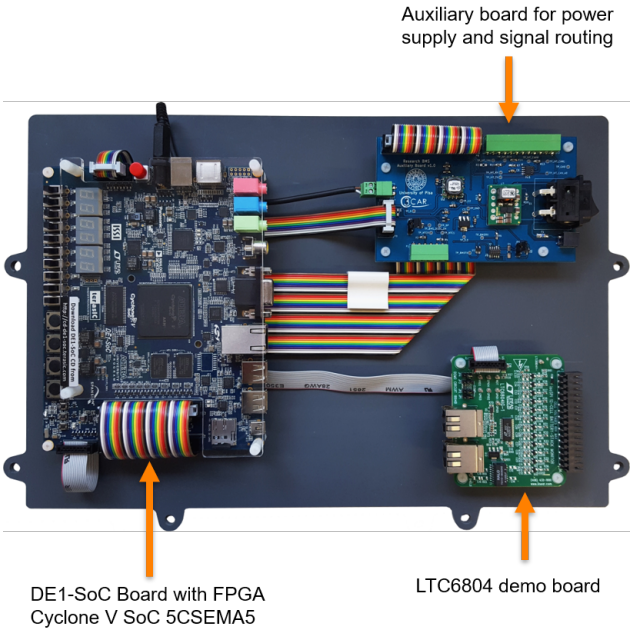


Figure 6.19: *Photograph of the Research BMS assembly.*

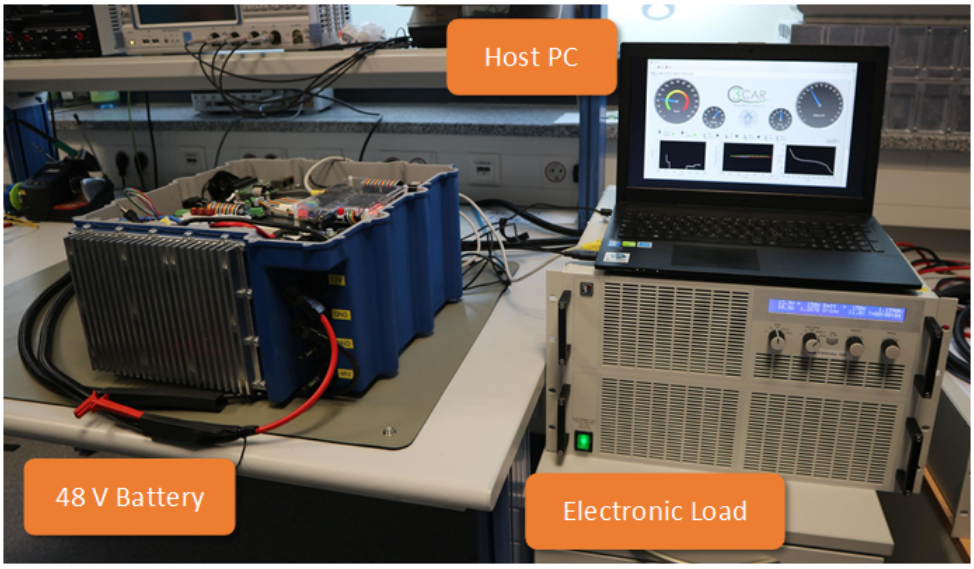


Figure 6.20: *Photograph of the experimental test bed.*

Figure 6.20 shows a photograph of the experimental setup, in which it is possible to note the Research BMS on top of the 48 V battery assembly and the host PC, which executes the developed LabVIEW UI. In the experimental setup is also visible an ELEKTRO-AUTOMATIK EA-EL 9080-400 programmable electronic load used to

perform discharge tests with variable currents. Thanks to this setup, all the BMS have been verified successfully, especially the communication with the SCs that has proven to be reliable, the power cabling, the operation of the sensors embedded in the SCs, the BMS monitoring and control strategies, and the DC/DC converter that has been successfully tested with no errors or signs of overheating of the PCB.

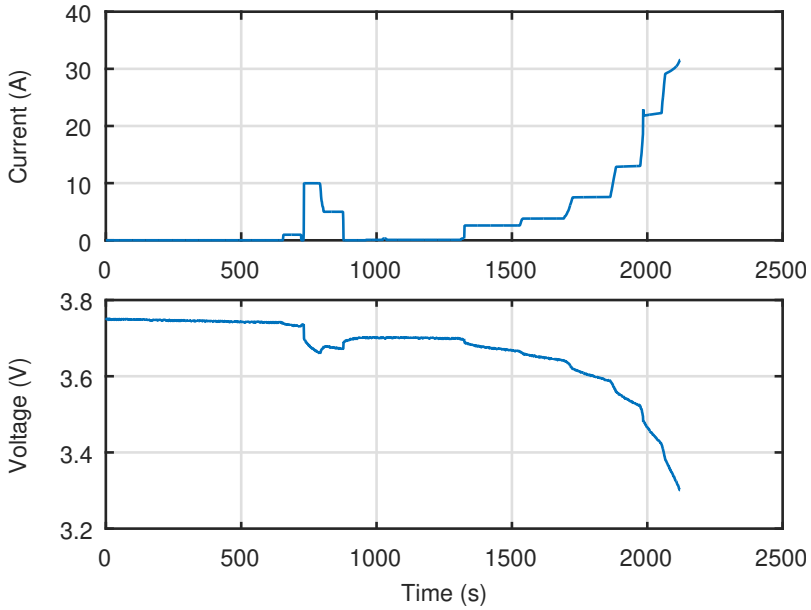


Figure 6.21: *Cell current and voltage profile during one of the experimental tests for the first cell.*

Moreover, the performed tests, consisting of different discharge cycles at different discharge currents, have allowed the acquisition of a wide set of data, consisting of the data sensed by the SCs and the results of the algorithms computed by the Research BMS. For the sake of clarity, we reported in Figure 6.21 the current and the voltage acquired during one of these tests only for one of the twelve SCs. The battery has been discharged with a current that changes in a range from 0 to about 30 A.

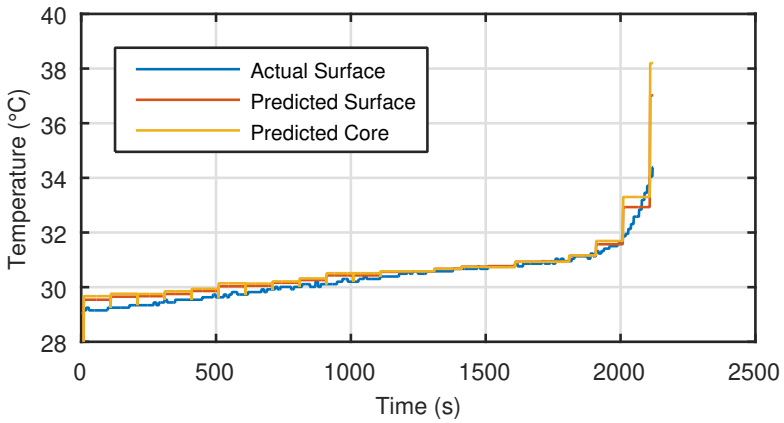


Figure 6.22: *Comparison between the measured and the predicted cell temperature for the first cell.*

Figures 6.22 and 6.23 show the results obtained by the TPA and the AMA in cell temperature and SOC estimation, respectively. Regarding the TPA estimation algo-

rithm, we can only measure the cell surface temperature, therefore we have a reference for the predicted surface temperature and not for the predicted cell core temperature. However, the algorithm seems to work in an effective way. We can also note a sharp temperature increase in the last part of the test, due to the increased power delivered to the load. At the same time, the difference between the estimated core temperature and the measured surface one increases as expected.

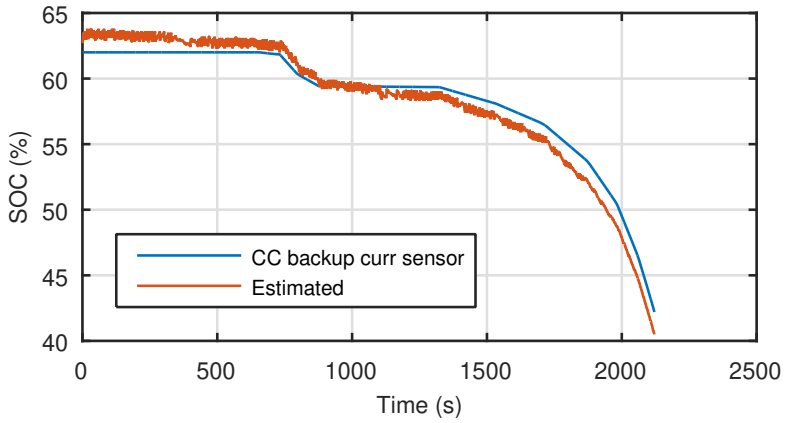


Figure 6.23: *Comparison between the reference and the estimate cell SOC for the first cell.*

Instead, the SOC reference values is computed by using the CC technique applied on the current measured by the reference current sensor which provides a very accurate current measurement. Also in this case, the AMA algorithm works in an effective way since the SOC rms error is of about $1\,\%$.

CHAPTER 7

Conclusions

The aim of the Ph.D. study presented in this thesis was the analysis of the state of the art of the BMSs and the battery state estimation algorithms, and the development of advanced solutions to improve the SOC estimation performance and the functionalities of these systems, by exploring novel architectures based on an SoC-FPGA.

The literature widely reports on the Li-ion technology advantages that make it suitable for a large number of applications, such as the electric mobility and the stationary energy storage. In these applications, batteries are composed of a large number of cells opportunely connected to reach the desired voltage and power levels. However, a safe and effective usage can be possible only with the presence of battery management systems. They have been studied in many works which also presented different techniques for managing the battery cells and to estimate their internal state, such as SOC and SOH. Usually, the solutions presented in literature describes BMS with an architecture distributed on different hierarchical levels which presents on the top a PMU able to manage the cells by using a microcontroller that also executes the estimation algorithms. This kind of architecture can become even more limiting, especially with the growing necessity to have more accurate and complex estimation algorithms that must work for each single cell in the battery pack.

These limitations can be overcome by using an SoC-FPGA instead of the microcontroller in the PMU, that is the solution investigated in this thesis. The resulting BMSs are more flexible and highly reconfigurable, thus they can be easily adapted to different applications or batteries. Furthermore, the possibility to implement algorithms as hardware accelerator in the FPGA fabric enables the use of complex estimation algorithms. Among all, in this Ph.D. study, the SOC online estimation problem has been addressed by employing model-based techniques which are suitable for embedded systems and use an electrical model of a cell, whose parameters are identified online to track their

variations with the operating conditions of the battery, to improve the SOC estimation accuracy.

Two model-based techniques for SOC estimation have been developed and implemented in hardware on an FPGA: the AMA and the DEKF. The hardware implementations of these two algorithms are characterised by good performance in terms of execution time and thus they can be used in time division multiplexing for estimating the SOC and the model parameters of a large number of cells. The development and verification phases have been carried out by using a HiL platform, built for simulating the behaviour of a battery during its usage on an EV in order to test both estimators in various operating conditions. In fact, the HiL platform can provide the current, voltage, and temperature profiles representative of the different simulations that can vary in the driving style, ambient temperature, type of vehicle and battery conditions. The results obtained by both the algorithms are very convincing in both the SOC and the parameter estimation. The achieved results allow us to use the identified parameters, such as the series resistance, to implement in future reliable SOH estimation algorithms. Moreover, the simulated scenarios allowed us to compare the behaviour of the two algorithms which mainly differs in terms of SOC correction time against a wrong SOC initialization and estimation accuracy in function of the used driving cycle. However, thanks to this HiL platform, both estimators have extensively been assessed in different situations without the need of a real battery and in a completely safe environment. The simulations enabled by the HiL platform has allowed us to show that these estimators are suitable for battery state and ECM parameters estimation in EVs.

The AMA hardware estimator has been used in two different BMS architectures with a PMU based on an SoC-FPGA which mainly differ on the organization of the lower hierarchical level. In the first one the cells are organised in module managed by stack monitors which communicate with the PMU using an SPI link. The second uses the concept of smart cell, *i.e.*, cells with a CMU provided with some sensors and able to communicate with the PMU. These two architectures have been applied to two case studies. The first one has been tested on an electric bike during a real utilization on the road, while the second has been used for a $48\,\mathrm{V}$ smart battery developed within the 3Ccar European project, where the BMS was able to execute advanced estimation algorithms implemented in hardware and software by the project partners. Both the implementations have been successfully tested in their hardware and software components, thanks to the tests performed in real application scenarios and in laboratory. During the final review of the 3Ccar project, the $48\,\mathrm{V}$ smart battery has been showed to the project officers and reviewers that provided very good comments.

In conclusion, we have discussed two innovative BMS architectures which use SoC-FPGA to address the more computational effort required by the estimation algorithms implementation. Moreover, we presented a HiL platform, highlighting its benefits in the development and assessment of the BMS functionalities. Finally, the obtained good results demonstrate the capabilities of these architectures and enable further future advances in the developing of ESSs based on the Li-ion technology.

Bibliography

[1] California Environmental Protection Agency, Air Resources Board. [Online]. Available: http://www.arb.ca.gov/homepage.htm.

[2] The UNECE World Forum for Harmonization of Vehicle Regulations (WP.29). [Online]. Available: https://www.unece.org/trans/main/wp29/meeting_docs_wp29.html.

[3] United Nations Economic Commission for Europe (UNECE). [Online]. Available: http://www.unece.org/info/ece-homepage.html.

[4] United States Environmental Protection Agency (U.S. EPA). [Online]. Available: http://www3.epa.gov/.

[5] Dave Andre, Christian Appel, Thomas Soczka-Guth, and Dirk Uwe Sauer. Advanced mathematical methods of SOC and SOH estimation for lithium-ion batteries. *Journal of Power Sources*, 224:20–27, 2013.

[6] Semih Aslan, Sufeng Niu, and Jafar Saniie. FPGA implementation of fast QR decomposition based on givens rotation. In *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 470–473. IEEE, Aug 2012.

[7] G V Avvari, B Pattipati, B Balasingam, K R Pattipati, and Y Bar-Shalom. Experimental set-up and procedures to test and validate battery fuel gauge algorithms. *Appl. Energy*, 160:404–418, 2015.

[8] Ølker Aydõn and Özgür Üstün. A Basic Battery Management System Design with IoT Feature for LiFePO4 Batteries. In *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, volume 4, pages 1309–1313, 2017.

[9] Atsushi Baba and Shuichi Adachi. Simultaneous state of charge and parameter estimation of lithium-ion battery using log-normalized unscented Kalman Filter. In *2015 American Control Conference (ACC)*, pages 311–316. IEEE, Jul 2015.

[10] B Balasingam, G V Avvari, K R Pattipati, and Y Bar-Shalom. Performance analysis results of a battery fuel gauge algorithm at multiple temperatures. *J. Power Sources*, 273:742–753, 2015.

[11] F Baronti, G Fantechi, E Leonardi, R Roncella, and R Saletti. Effective modeling of temperature effects on lithium polymer cells. In *2010 17th IEEE Int. Conf. Electron. Circuits Syst.*, pages 990–993, Athens, 2010.

[12] F Baronti, G Fantechi, E Leonardi, R Roncella, and R Saletti. Enhanced model for Lithium-Polymer cells including temperature effects. In *IECON 2010 - 36th Annu. Conf. IEEE Ind. Electron. Soc.*, pages 2329–2333. IEEE, Nov 2010.

[13] F. Baronti, W. Zamboni, N. Femia, H. Rahimi-Eichi, R. Roncella, S. Rosi, R. Saletti, and M.-Y. Chow. Parameter identification of Li-Po batteries in electric vehicles: A comparative study. In *2013 IEEE International Symposium on Industrial Electronics (ISIE)*, pages 1–7. IEEE, May 2013.

[14] Federico Baronti, Cinzia Bernardeschi, Luca Cassano, Andrea Domenici, Roberto Roncella, and Roberto Saletti. Design and Safety Verification of a Distributed Charge Equalizer for Modular Li-Ion Batteries. *IEEE Trans. Ind. Informatics*, 10(2):1003–1011, 2014.

[15] Federico Baronti, Gabriele Fantechi, Roberto Roncella, and Roberto Saletti. High-Efficiency Digitally Controlled Charge Equalizer for Series-Connected Cells Based on Switching Converter and Super-Capacitor. *IEEE Trans. Ind. Informatics*, 9(2):1139–1147, 2013.

[16] Federico Baronti, Roberto Roncella, Roberto Saletti, and Walter Zamboni. FPGA Implementation of the Mix Algorithm for State-of-Charge Estimation of Lithium-Ion Batteries. In *IECON 2014 - 40th Annu. Conf. IEEE Ind. Electron. Soc.*, pages 5641–5646, 2014.

[17] Jorge Varela Barreras, Christian Fleischer, Andreas Elkjaer Christensen, Maciej Swierczynski, Erik Schaltz, Soren Juhl Andreasen, and Dirk Uwe Sauer. An Advanced HIL Simulation Battery Model for Battery Management System Testing. *IEEE Transactions on Industry Applications*, 52(6):5086–5099, Nov 2016.

[18] A C Baughman and M Ferdowsi. Double-Tiered Switched-Capacitor Battery Charge Equalization Technique. *IEEE Trans. Ind. Electron.*, 55(6):2277–2285, 2008.

[19] H.P.G.J. Beelen, L.H.J. Raijmakers, M.C.F. Donkers, P.H.L. Notten, and H.J. Bergveld. A comparison and accuracy analysis of impedance-based temperature estimation methods for Li-ion batteries. *Applied Energy*, 175:128–140, Aug 2016.

[20] S. Bernardi, F. Flammini, S. Marrone, N. Mazzocca, J. Merseguer, R. Nardone, and V. Vittorini. Enabling the usage of UML in the verification of railway systems: The DAM-rail approach. *Reliability Engineering and System Safety*, 120:112–126, Dec 2013.

[21] S. Boukadida, S. Gdaim, and A. Mtibaa. Hardware implementation of FTC of induction machine on FPGA. *Electronics*, 20(2):76–84, 2016.

[22] M. Brandl, H. Gall, M. Wenger, V. Lorentz, M. Giegerich, F. Baronti, G. Fantechi, L. Fanucci, R. Roncella, R. Saletti, S. Saponara, A. Thaler, M. Cifrain, and W. Prochazka. Batteries and battery management systems for electric vehicles. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 971–976. IEEE, Mar 2012.

[23] Attila Buchman and Claudiu Lung. State of Charge and State of Health Estimation of Lithium-Ion Batteries. In *2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 382–385. IEEE, Oct 2018.

[24] Christian Campestrini, Thomas Heil, Stephan Kosch, and Andreas Jossen. A comparative study and review of different Kalman filters by applying an enhanced validation method. *Journal of Energy Storage*, 8:142–159, 2016.

[25] Jian Cao, Nigel Schofield, and Ali Emadi. Battery balancing methods: A comprehensive review. In *2008 IEEE Veh. Power Propuls. Conf.*, pages 1–6. IEEE, 2008.

[26] M Charkhgard and M Farrokhi. State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF. *IEEE Trans. Ind. Electron.*, 57(12):4178–4187, 2010.

[27] Cheng Chen, Rui Xiong, and Weixiang Shen. A lithium-ion battery-in-the-loop approach to test and validate multi-scale dual H infinity filters for state of charge and capacity estimation. *IEEE Transactions on Power Electronics*, 33(1):1–1, Jan 2017.

[28] M Chen and G A Rincon-Mora. Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance. *IEEE Trans. Energy Convers.*, 21(2):504–511, 2006.

[29] Xiaopeng Chen, Weixiang Shen, Thanh Tu Vo, Zhenwei Cao, and Ajay Kapoor. An overview of lithium-ion batteries for electric vehicles. In *2012 10th International Power & Energy Conference (IPEC)*, pages 230–235. IEEE, Nov 2012.

[30] K W E Cheng, B P Divakar, Hongjie Wu, Kai Ding, and Ho Fai Ho. Battery-Management System (BMS) and SOC Development for Electrical Vehicles. *IEEE Trans. Veh. Technol.*, 60(1):76–88, 2011.

[31] Fabio Codeca, Sergio M Savaresi, and Vincenzo Manzoni. The mix estimation algorithm for battery State-of-Charge estimator- Analysis of the sensitivity to measurement errors. In *Proc. 48h IEEE Conf. Decis. Control held jointly with 2009 28th Chinese Control Conf.*, pages 8083–8088. IEEE, 2009.

[32] Fabio Codeca, Sergio M Savaresi, and Giorgio Rizzoni. On battery State of Charge estimation: A new mixed algorithm. In *2008 IEEE Int. Conf. Control Appl.*, pages 102–107. IEEE, 2008.

[33] F. V. Conte. Battery and battery management for hybrid electric vehicles: a review. *e & i Elektrotechnik und Informationstechnik*, 123(10):424–431, Oct 2006.

[34] Haifeng Dai, Xuezhe Wei, and Zechang Sun. State and Parameter Estimation of a HEV Li-ion Battery Pack Using Adaptive Kalman Filter with a New SOC-OCV Concept. In *2009 International Conference on Measuring Technology and Mechatronics Automation*, pages 375–380. IEEE, 2009.

[35] Haifeng Dai, Tianjiao Xu, Letao Zhu, Xuezhe Wei, and Zechang Sun. Adaptive model parameter identification for large capacity Li-ion batteries on separated time scales. *Applied Energy*, 184:119–131, 2016.

[36] Haifeng Dai, Xiaolong Zhang, Xuezhe Wei, Zechang Sun, Jiayuan Wang, and Feng Hu. Cell-BMS validation with a hardware-in-the-loop simulation of lithium-ion battery cells for electric vehicles. *Int. J. Electr. Power Energy Syst.*, 52:174–184, Nov 2013.

[37] Dai Haifeng, Wei Xuezhe, and Sun Zechang. A new SOH prediction concept for the power lithium-ion battery used on HEVs. In *2009 IEEE Veh. Power Propuls. Conf.*, pages 1649–1653. IEEE, 2009.

[38] M. Deepan Raj, I. Gogul, M. Thangaraja, and V. Sathiesh Kumar. Static gesture recognition based precise positioning of 5-DOF robotic arm using FPGA. In *2017 Trends in Industrial Measurement and Automation (TIMA)*, pages 1–6. IEEE, Jan 2017.

[39] R. Di Rienzo, F. Baronti, R. Roncella, R. Morello, and R. Saletti. Simulation platform for analyzing battery parallelization. In *2017 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pages 1–4. IEEE, Jun 2017.

[40] R. Di Rienzo, F. Baronti, F. Vellucci, F. Cignini, F. Ortenzi, G. Pede, R. Roncella, and R. Saletti. Experimental analysis of an electric minibus with small battery and fast charge policy. In *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*, pages 1–6. IEEE, Nov 2016.

[41] R. Di Rienzo, R. Roncella, R. Morello, F. Baronti, and R. Saletti. Low-cost modular battery emulator for battery management system testing. In *2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*, pages 44–49. IEEE, Jan 2018.

[42] William Diehl, Farnoud Farahmand, Panasayya Yalla, Jens-Peter Kaps, and Kris Gaj. Comparison of hardware and software implementations of selected lightweight block ciphers. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–4. IEEE, Sep 2017.

[43] Tingting Dong, Jun Li, and Haifeng Dai. Analysis on the influence of measurement precision of the battery management system on the state of charge estimation. In *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, pages 1–5. IEEE, 2010.

[44] Dong Tingting, Li Jun, Zhao Fuquan, You Yi, and Jin Qiqian. Analysis on the influence of measurement error on state of charge estimation of LiFePO4 power Battery. In *2011 International Conference on Materials for Renewable Energy & Environment*, pages 644–649. IEEE, May 2011.

[45] Jiuyu Du, Zhe Chen, and Feiqiang Li. Multi-Objective Optimization Discharge Method for Heating Lithium-Ion Battery at Low Temperatures. *IEEE Access*, 6:44036–44049, 2018.

[46] Van-Huan Duong, Hany Ayad Bastawrous, KaiChin Lim, Khay Wai See, Peng Zhang, and Shi Xue Dou. Online state of charge and model parameters estimation of the LiFePO4 battery in electric vehicles using multiple adaptive forgetting factors recursive least-squares. *Journal of Power Sources*, 296:215–224, 2015.

[47] Christoforos Economakos, George Kiokes, and George Economakos. Using advanced FPGA SoC technologies for the design of industrial control applications. In *IISA 2015 - 6th International Conference on Information, Intelligence, Systems and Applications*, pages 1–6. IEEE, Jul 2016.

[48] Markus Einhorn, Fiorentino Valerio Conte, Christian Kral, and Juergen Fleig. A Method for Online Capacity Estimation of Lithium Ion Battery Cells Using the State of Charge and the Transferred Charge. *IEEE Trans. Ind. Appl.*, 48(2):736–741, Mar 2012.

[49] Guodong Fan, Ke Pan, and Marcello Canova. A comparison of model order reduction techniques for electrochemical characterization of Lithium-ion batteries. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3922–3931. IEEE, Dec 2015.

[50] Javier Gallardo-Lozano, Enrique Romero-Cadaval, M Isabel Milanes-Montero, and Miguel A Guerrero-Martinez. Battery equalization active methods. *J. Power Sources*, 246:934–949, 2014.

[51] Thomas Gallien and Georg Brasseur. State of charge estimation of a LiFePO4 battery: A dual estimation approach incorporating open circuit voltage hysteresis. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6. IEEE, May 2016.

[52] Juan Jesus Garcia, Alvaro Hernandez, Jesus Urena, and Enrique Garcia. FPGA-Based Architecture for a Multisensory Barrier to Enhance Railway Safety. *IEEE Transactions on Instrumentation and Measurement*, 65(6):1352–1363, Jun 2016.

[53] M. Giegerich, M. Akdere, C. Freund, T. Fühner, J. L. Grosch, S. Koffel, R. Schwarz, S. Waldhör, M. Wenger, V. R.H. Lorentz, and M. März. Open, flexible and extensible battery management system for lithium-ion batteries in mobile and stationary applications. In *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pages 991–996. IEEE, Jun 2016.

[54] Damien Guilbert, Michael Guarisco, Arnaud Gaillard, Abdoul N'Diaye, and Abdesslem Djerdir. FPGA based fault-tolerant control on an interleaved DC/DC boost converter for fuel cell electric vehicle applications. *International Journal of Hydrogen Energy*, 40(45):15815–15822, Dec 2015.

[55] M.A. Hannan, M.S.H. Lipu, A. Hussain, and A. Mohamed. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renewable and Sustainable Energy Reviews*, 78:834–854, Oct 2017.

[56] Hongwen He, Rui Xiong, and Hongqiang Guo. Online estimation of model parameters and state-of-charge of LiFePO4 batteries in electric vehicles. *Applied Energy*, 89(1):413–420, 2012.

[57] Hongwen He, Rui Xiong, and Jiankun Peng. Real-time estimation of battery state-of-charge with unscented Kalman filter and RTOS μCOS-II platform. *Applied Energy*, 162:1410–1418, 2016.

[58] Hongwen He, Xiaowei Zhang, Rui Xiong, Yongli Xu, and Hongqiang Guo. Online model-based estimation of state-of-charge and open-circuit voltage of lithium-ion batteries in electric vehicles. *Energy*, 39(1):310–318, 2012.

[59] Wei He, Nicholas Williard, Chaochao Chen, and Michael Pecht. State of charge estimation for electric vehicle batteries using unscented kalman filtering. *Microelectronics Reliability*, 53(6):840–847, 2013.

[60] Yongsheng He, Wei Liu, and Brain J Koch. Battery algorithm verification and development using hardware-in-the-loop testing. *J. Power Sources*, 195(9):2969–2974, 2010.

[61] E. Hoedemaekers. Battery Core Temperature Prediction, Including Parameter Estimation for Li-ion Pouch Cells. In *Advanced Battery Power Conference*, Aachen, 2017.

[62] Chao Huang, Zhenhua Wang, Zihan Zhao, Long Wang, Chun Sing Lai, and Dong Wang. Robustness Evaluation of Extended and Unscented Kalman Filter for Battery State of Charge Estimation. *IEEE Access*, 6:27617–27628, 2018.

[63] Cong-Sheng Huang and Mo-Yuen Chow. Accurate Thevenin's circuit-based battery model parameter identification. In *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pages 274–279. IEEE, Jun 2016.

[64] T Huria, M Ceraolo, J Gazzarri, and R Jackey. High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells. In *2012 IEEE Int. Electr. Veh. Conf.*, pages 1–8. IEEE, Mar 2012.

[65] International Energy Agency. Global EV Outlook 2018, 2018.

[66] S. M. Rakiul Islam, Sung-Yeul Park, and Balakumar Balasingam. Circuit parameters extraction algorithm for a lithium-ion battery charging system incorporated with electrochemical impedance spectroscopy. In *2018 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 3353–3358. IEEE, Mar 2018.

[67] Yong-Min Jeong, Yong-Ki Cho, Jung-Hoon Ahn, Seung-Hee Ryu, and Byoung-Kuk Lee. Enhanced Coulomb counting method with adaptive SOC reset time for estimating OCV. In *2014 IEEE Energy Convers. Congr. Expo.*, pages 1313–1318. IEEE, 2014.

[68] Wei Jian, Xuehuan Jiang, Jinliang Zhang, Zhengtao Xiang, and Yubing Jian. Comparison of SOC Estimation Performance with Different Training Functions Using Neural Network. In *2012 UKSim 14th International Conference on Computer Modelling and Simulation*, pages 459–463. IEEE, Mar 2012.

[69] Tino Kahl and Sibylle Dieckerhoff. Comparison of FPGA- and microcontroller-based control of a high-dynamic power electronic converter. In *2017 IEEE 18th Workshop on Control and Modeling for Power Electronics (COMPEL)*, pages 1–6. IEEE, Jul 2017.

[70] R E Kalman. A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME – J. Basic Eng.*, (82 (Series D)):35–45, 1960.

[71] Hahnsang Kim and Kang G. Shin. DESA: Dependable, efficient, scalable architecture for management of large-scale batteries. *IEEE Transactions on Industrial Informatics*, 8(2):406–417, May 2012.

[72] Taesic Kim, Wei Qiao, and Liayn Qu. Online SOC and SOH estimation for multicell lithium-ion batteries based on an adaptive hybrid battery model and sliding-mode observer. In *2013 IEEE Energy Conversion Congress and Exposition, ECCE 2013*, pages 292–298. IEEE, Sep 2013.

[73] Taesic Kim, Wei Qiao, and Liyan Qu. Real-time state of charge and electrical impedance estimation for lithium-ion batteries based on a hybrid battery model. In *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 563–568. IEEE, Mar 2013.

[74] Xiangdong Kong, Yuejiu Zheng, Minggao Ouyang, Xiangjun Li, Languang Lu, Jianqiu Li, and Zhendong Zhang. Signal synchronization for massive data storage in modular battery management system with controller area network. *Applied Energy*, 197:52–62, Jul 2017.

[75] Saber Krim, Soufien Gdaim, Abdellatif Mtibaa, and Mohamed Faouzi Mimouni. Design and Implementation of Direct Torque Control Based on an Intelligent Technique of Induction Motor on FPGA. *Journal of Electrical Engineering and Technology*, 10(4):1527–1539, Jul 2015.

[76] Xin Lai, Yuejiu Zheng, and Tao Sun. A comparative study of different equivalent circuit models for estimating state-of-charge of lithium-ion batteries. *Electrochimica Acta*, 259:566–577, Jan 2018.

[77] Wai Chung Lee, David Drury, and Phil Mellor. Comparison of passive cell balancing and active cell balancing for automotive batteries. In *2011 IEEE Veh. Power Propuls. Conf.*, pages 1–7. IEEE, 2011.

[78] Jianchao Li, Shunli Wang, Carlos Fernandez, Ni Wang, and Hongqiu Xie. The battery management system construction method study for the power lithium-ion battery pack. In *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, pages 285–289. IEEE, Dec 2017.

[79] G. Litta, R. Di Rienzo, R. Morello, R. Roncella, F. Baronti, and R. Saletti. Flexible platform with wireless interface for DC-motor remote control. In *2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*, pages 509–514. IEEE, Jan 2018.

[80] Bingqi Liu, Mingzhe Liu, Xin Jiang, Xianguo Tuo, Helen Zhou, and Jun Ren. Design of battery management system based on DSP for BEV. In *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, pages 857–862. IEEE, Jul 2017.

[81] V. R.H. Lorentz, M. M. Wenger, J. L. Grosch, M. Giegerich, M. P.M. Jank, M. Marz, and L. Frey. Novel cost-efficient contactless distributed monitoring concept for smart battery cells. In *IEEE International Symposium on Industrial Electronics*, pages 1342–1347. IEEE, May 2012.

[82] Languang Lu, Xuebing Han, Jianqiu Li, Jianfeng Hua, and Minggao Ouyang. A review on the key issues for lithium-ion battery management in electric vehicles. *J. Power Sources*, 226:272–288, Mar 2013.

[83] Yan Ma, Bingsi Li, Xiuwen Zhou, and Hong Chen. Battery state of charge estimation hardware-in-loop system design based on xPC target. In *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, volume 2016-Septe, pages 1338–1343. IEEE, Jun 2016.

[84] K Makinejad, R Arunachala, S Arnold, H Ennifar, H Zhou, A Jossen, and W Changyun. A lumped electro-thermal model for Li-ion cells in electric vehicle application. *28th International Electric Vehicle Symposium and Exhibition 2015, EVS 2015*, pages 1–13, 2015.

[85] E. Martinez-Laserna, E. Sarasketa-Zabala, D.-I. Stroe, M. Swierczynski, A. Warnecke, J.M. Timmermans, S. Goutam, and P. Rodriguez. Evaluation of lithium-ion battery second life performance and degradation. In *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1–7. IEEE, Sep 2016.

[86] Egoitz Martinez-Laserna, Elixabet Sarasketa-Zabala, Igor Villarreal Sarria, Daniel-Ioan Stroe, Maciej Swierczynski, Alexander Warnecke, Jean-Marc Timmermans, Shovon Goutam, Noshin Omar, and Pedro Rodriguez. Technical Viability of Battery Second Life: A Study From the Ageing Perspective. *IEEE Transactions on Industry Applications*, 54(3):2703–2713, May 2018.

[87] J. Mei, Eric K. W. Cheng, and Y. C. Fong. Lithium-titanate battery (LTO): A better choice for high current equipment. In *2016 International Symposium on Electrical Engineering (ISEE)*, pages 1–4. IEEE, Dec 2016.

[88] Jinhao Meng, Mattia Ricco, Guangzhao Luo, Maciej Swierczynski, Daniel-Ioan Stroe, Ana-Irina Stroe, and Remus Teodorescu. An Overview and Comparison of Online Implementable SOC Estimation Methods for Lithium-Ion Battery. *IEEE Transactions on Industry Applications*, 54(2):1583–1591, Mar 2018.

[89] R. Morello, F. Baronti, X. Tian, T. Chau, R. Di Rienzo, R. Roncella, B. Jeppesen, W. H. Lin, T. Ikushima, and R. Saletti. Hardware-in-the-loop simulation of FPGA-based state estimators for electric vehicle batteries. In *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pages 280–285. IEEE, Jun 2016.

[90] R. Morello, R. Di Rienzo, R. Roncella, R. Saletti, and F. Baronti. Tuning of Moving Window Least Squares-based algorithm for online battery parameter estimation. In *2017 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pages 1–4. IEEE, Jun 2017.

[91] R. Morello, W. Zamboni, F. Baronti, R. Di Rienzo, R. Roncella, G. Spagnuolo, and R. Saletti. Comparison of state and parameter estimators for electric vehicle batteries. In *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pages 5433–5438, 2016.

[92] Rocco Morello, Roberto Di Rienzo, Federico Baronti, Roberto Roncella, and Roberto Saletti. System on chip battery state estimator: E-bike case study. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 2129–2134. IEEE, Oct 2016.

[93] S. Nejad, D. T. Gladwin, and D. A. Stone. Sensitivity of lumped parameter battery models to constituent parallel-RC element parameterisation error. In *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pages 5660–5665. IEEE, Oct 2014.

[94] S. Nejad, D. T. Gladwin, and D. A. Stone. On-chip implementation of Extended Kalman Filter for adaptive battery states monitoring. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 5513–5518. IEEE, Oct 2016.

[95] S. Nejad, D.T. Gladwin, and D.A. Stone. A systematic review of lumped-parameter equivalent circuit models for real-time estimation of lithium-ion battery states. *Journal of Power Sources*, 316:183–196, Jun 2016.

[96] Bo Ning, Jun Xu, Binggang Cao, Bin Wang, and Guangcan Xu. A Sliding Mode Observer SOC Estimation Method Based on Parameter Adaptive Battery Model. *Energy Procedia*, 88:619–626, Jun 2016.

[97] Francisco Ortega-Zamorano, Jose M. Jerez, Daniel Urda Munoz, Rafael M. Luque-Baena, and Leonardo Franco. Efficient Implementation of the Backpropagation Algorithm in FPGAs and Microcontrollers. *IEEE Transactions on Neural Networks and Learning Systems*, 27(9):1840–1850, Sep 2016.

[98] Noemi Otero, Habiballah Rahimi-Eichi, Juan J. Rodriguez-Andina, and Mo Yuen Chow. FPGA implementation of an observer for state of charge estimation in lithium-polymer batteries. In *Proceedings - 2014 International Conference on Mechatronics and Control, ICMC 2014*, pages 1646–1651. IEEE, Jul 2015.

[99] M. Oya, K. Takaba, L. Lin, R. Ishizaki, N. Kawarabayashi, and M. Fukui. Accurate SoC estimation of lithium-ion batteries based on parameter-dependent state-space model. In *2015 15th International Symposium on Communications and Information Technologies (ISCIT)*, pages 37–40. IEEE, Oct 2015.

[100] S. Panchal, M. Mathew, R. Fraser, and M. Fowler. Electrochemical thermal modeling and experimental measurements of 18650 cylindrical lithium-ion battery during discharge cycle for an EV. *Applied Thermal Engineering*, 135:123–132, May 2018.

[101] Maral Partovibakhsh and Guangjun Liu. An Adaptive Unscented Kalman Filtering Approach for Online Estimation of Model Parameters and State-of-Charge of Lithium-Ion Batteries for Autonomous Mobile Robots. *IEEE Transactions on Control Systems Technology*, 23(1):357–363, Jan 2015.

[102] B Pattipati, B Balasingam, G V Avvari, K R Pattipati, and Y Bar-Shalom. Open circuit voltage characterization of lithium-ion batteries. *J. Power Sources*, 269:317–333, 2014.

[103] G. Petrone, G. Spagnuolo, and W. Zamboni. Numerical study of the DEKF parameter identification capabilities in fuel cell EIS tests. In *2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*, pages 50–55. IEEE, Jan 2018.

[104] Sabine Piller, Marion Perrin, and Andreas Jossen. Methods for state-of-charge determination and their applications. *J. Power Sources*, 96(1):113–120, 2001.

[105] Liu Ping, Zhu Hong-hui, Chen Jie, and Li Guan-yan. A distributed management system for lithium ion battery pack. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 3997–4002. IEEE, May 2016.

[106] Gregory L. Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background. *Journal of Power Sources*, 134(2):252–261, 2004.

[107] Gregory L. Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification. *Journal of Power Sources*, 134(2):262–276, 2004.

[108] Gregory L Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *J. Power Sources*, 134(2):277–292, 2004.

[109] Gregory L. Plett. Dual and Joint EKF for Simultaneous SOC and SOH Estimation. *CD-ROM PROCEEDINGS OF THE 21ST ELECTRIC VEHICLE SYMPOSIUM (EVS21*, 2005.

[110] Gregory L. Plett. Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1: Introduction and state estimation. *Journal of Power Sources*, 161(2):1356–1368, 2006.

[111] Gregory L. Plett. Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2: Simultaneous state and parameter estimation. *Journal of Power Sources*, 161(2):1369–1384, 2006.

[112] Jian Qi and Dylan Dah-Chuan Lu. Review of battery cell balancing techniques. In *2014 Australasian Universities Power Engineering Conference (AUPEC)*, pages 1–6. IEEE, Sep 2014.

[113] H Rahimi-Eichi, F Baronti, and M.-Y. Chow. Modeling and online parameter identification of Li-Polymer battery cells for SOC estimation. In *2012 IEEE Int. Symp. Ind. Electron.*, pages 1336–1341. IEEE, 2012.

[114] Habiballah Rahimi-Eichi, Federico Baronti, and Mo-Yuen Chow. Online Adaptive Parameter Identification and State-of-Charge Coestimation for Lithium-Polymer Battery Cells. *IEEE Trans. Ind. Electron.*, 61(4):2053–2061, Apr 2014.

[115] Habiballah Rahimi-Eichi, Unnati Ojha, Federico Baronti, and Mo-Yuen Chow. Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles. *IEEE Ind. Electron. Mag.*, 7(2):4–16, 2013.

[116] V Ramadesigan, P W C Northrop, S De, S Santhanagopalan, R D Braatz, and V R Subramanian. Modeling and Simulation of Lithium-Ion Batteries from a Systems Engineering Perspective. *J. Electrochem. Soc.*, 159(3):R31—-R45, 2012.

[117] Hannes Rathmann, Christoph Weber, Wolfgang Benecke, and Dirk Kahler. Sophisticated estimation of hardly measurable conditions of lithium-ion batteries. In *IECON Proceedings (Industrial Electronics Conference)*, pages 1862–1866. IEEE, Nov 2013.

[118] Rocco Restaino and Walter Zamboni. Comparing particle filter and extended kalman filter for battery State-Of-Charge estimation. In *IECON 2012 - 38th Annu. Conf. IEEE Ind. Electron. Soc.*, pages 4018–4023. IEEE, 2012.

[119] Juan Rivera-Barrera, Nicolás Muñoz-Galeano, and Henry Sarmiento-Maldonado. SoC Estimation for Lithium-ion Batteries: Review and Future Challenges. *Electronics*, 6(4):102, Nov 2017.

[120] Michael A Roscher and Dirk Uwe Sauer. Dynamic electric behavior and open-circuit-voltage modeling of LiFePO4-based lithium ion secondary batteries. *J. Power Sources*, 196(1):331–336, 2011.

[121] Fida Saidani, Franz X. Hutter, Wilhelm Selinger, Zili Yu, and Joachim N. Burghartz. A lithium-ion battery demonstrator for HEV applications featuring a smart system at cell level. In *2017 IEEE International Systems Engineering Symposium (ISSE)*, pages 1–5. IEEE, Oct 2017.

[122] Daniel Sauerteig, Nina Hanselmann, Arno Arzberger, Holger Reinshagen, Svetlozar Ivanov, and Andreas Bund. Electrochemical-mechanical coupled modeling and parameterization of swelling and ionic transport in lithium-ion batteries. *Journal of Power Sources*, 378:235–247, Feb 2018.

[123] Deshanand P. Singh, Bogdan Pasca, and Tomasz S. Czajkowski. High-Level Design Tools for Floating Point FPGAs. *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 9–12, Feb 2015.

[124] Panu Sjovall, Janne Virtanen, Jarno Vanne, and Timo D. Hamalainen. High-level synthesis design flow for HEVC intra encoder on SoC-FPGA. In *Proceedings - 18th Euromicro Conference on Digital System Design, DSD 2015*, pages 49–56. IEEE, Aug 2015.

[125] Sebastian Steinhorst. Design and verification methodologies for Smart Battery Cells. In *2016 International Symposium on Integrated Circuits (ISIC)*, pages 1–4. IEEE, Dec 2016.

[126] Sebastian Steinhorst, Martin Lukasiewycz, Swaminathan Narayanaswamy, Matthias Kauer, and Samarjit Chakraborty. Smart Cells for Embedded Battery Management. In *2014 IEEE International Conference on Cyber-Physical Systems, Networks, and Applications*, pages 59–64. IEEE, Aug 2014.

[127] Rahul Subramanian, Paul Venhovens, and Barry P. Keane. Accelerated design and optimization of battery management systems using HIL simulation and rapid control prototyping. In *2012 IEEE International Electric Vehicle Conference, IEVC 2012*, pages 1–5. IEEE, Mar 2012.

[128] Bingxiang Sun and Lifang Wang. The SOC Estimation of NIMH Battery Pack for HEV Based on BP Neural Network. In *2009 International Workshop on Intelligent Systems and Applications*, pages 1–4. IEEE, May 2009.

[129] Carlo Taborelli and Simona Onori. State of charge estimation using extended Kalman filters for battery management system. In *2014 IEEE International Electric Vehicle Conference (IEVC)*, pages 1–8. IEEE, Dec 2014.

[130] Andreas Thanheiser, Tom P. Kohler, Christiane Bertram, and Hans-Georg Herzog. Battery emulation considering thermal behavior. In *2011 IEEE Vehicle Power and Propulsion Conference*, pages 1–5. IEEE, Sep 2011.

[131] X. Tian, B. Jeppesen, T. Ikushima, F. Baronti, and R. Morello. Accelerating State-Of-Charge Estimation in FPGA-based Battery Management Systems. In *6th Hybrid and Electric Vehicles Conference (HEVC 2016)*, pages 1–6. Institution of Engineering and Technology, 2016.

[132] Wang Tianmiao. Embedded control system for industrial robots. *International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, pages 122–125, Aug 2010.

[133] University of Pisa. EV Battery HiL, 2018.

[134] María Dolores Valdés Peña, Juan J. Rodriguez-Andina, and Milos Manic. The Internet of Things: The Role of Reconfigurable Platforms. *IEEE Industrial Electronics Magazine*, 11(3):6–19, Sep 2017.

[135] Andrea Vezzini. Lithium-Ion Battery Management. In *Lithium-Ion Batteries: Advances and Applications*, pages 345–360. Elsevier, 2014.

[136] G. Vicidomini, G. Petrone, E. Monmasson, and G. Spagnuolo. FPGA based implementation of a sliding-mode observer for battery state of charge estimation. In *2017 IEEE International Symposium on Industrial Electronics (ISIE)*, pages 1268–1273. IEEE, Jun 2017.

[137] Ole K. Vigerstol. A review of the suitability of lithium battery technology in ict energy infrastructure. In *2017 IEEE International Telecommunications Energy Conference (INTELEC)*, pages 254–261. IEEE, Oct 2017.

[138] Wladislaw Waag, Christian Fleischer, and Dirk Uwe Sauer. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *Journal of Power Sources*, 258:321–339, 2014.

[139] Eric A. Wan, Eric A. Wan, and Alex T. Nelson. Dual Kalman Filtering Methods for Nonlinear Prediction, Smoothing, and Estimation. *IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 9*, 1997.

[140] Shaojun Wang, Datong Liu, Jianbao Zhou, Bin Zhang, and Yu Peng. A run-time dynamic reconfigurable computing system for lithium-ion battery prognosis. *Energies*, 9(8):572, Jul 2016.

[141] He Wen, Da Cheng, Zhaosheng Teng, Siyu Guo, and Fuhai Li. Approximate algorithm for fast calculating voltage unbalance factor of three-phase power system. *IEEE Transactions on Industrial Informatics*, 10(3):1799–1805, Aug 2014.

[142] Jiaqi Xi, Mian Li, and Min Xu. Optimal energy management strategy for battery powered electric vehicles. *Applied Energy*, 134:332–341, Dec 2014.

[143] Shun Xiang, Guangdi Hu, Ruisen Huang, Feng Guo, Pengkai Zhou, Shun Xiang, Guangdi Hu, Ruisen Huang, Feng Guo, and Pengkai Zhou. Lithium-Ion Battery Online Rapid State-of-Power Estimation under Multiple Constraints. *Energies*, 11(2):283, Jan 2018.

[144] Wang Xinying, Phillip Jones, and Joseph Zambreno. A Reconfigurable Architecture for QR Decomposition Using a Hybrid Approach. In *VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on*, pages 541–546. IEEE, Jul 2014.

[145] Rui Xiong, Fengchun Sun, Xianzhi Gong, and Chenchen Gao. A data-driven based adaptive state of charge estimator of lithium-ion polymer battery used in electric vehicles. *Applied Energy*, 113:1421–1433, 2014.

[146] Xueyan Li and Song-Yul Choe. State-of-charge (SOC) estimation based on a reduced order electrochemical thermal model and extended Kalman filter. In *2013 American Control Conference*, pages 1100–1105. IEEE, Jun 2013.

[147] Fangfang Yang, Dong Wang, Yang Zhao, Kwok-Leung Tsui, and Suk Joo Bae. A study of the relationship between coulombic efficiency and capacity degradation of commercial lithium-ion batteries. *Energy*, 145:486–495, Feb 2018.

[148] Ruixin Yang, Rui Xiong, Hongwen He, Hao Mu, and Chun Wang. A novel method on estimating the degradation and state of charge of lithium-ion batteries used for electrical vehicles. *Applied Energy*, 207:336–345, Dec 2017.

[149] A. Zenati, Ph. Desprez, H. Razik, and S. Rael. Impedance measurements combined with the fuzzy logic methodology to assess the SOC and SOH of lithium-ion cells. In *2010 IEEE Vehicle Power and Propulsion Conference*, pages 1–6. IEEE, Sep 2010.

[150] Aixiang Zhang, Shizhan Song, Chuanyong Wang, Jian Zhang, Kun Wang, and Liwei Li. Research of battery management system for integrated power supply. In *2017 Chinese Automation Congress (CAC)*, pages 3178–3181. IEEE, Oct 2017.

[151] Jianan Zhang, Lei Zhang, Fengchun Sun, and Zhenpo Wang. An Overview on Thermal Safety Issues of Lithium-ion Batteries for Electric Vehicle Application. *IEEE Access*, 6:23848–23863, 2018.

[152] Jinlong Zhang, Yanjun Wei, and Hanhong Qi. State of charge estimation of LiFePO4 batteries based on online parameter identification. *Applied Mathematical Modelling*, 40(11):6040–6050, 2016.

[153] Yongzhi Zhang, Rui Xiong, Hongwen He, and Weixiang Shen. Lithium-Ion Battery Pack State of Charge and State of Energy Estimation Algorithms Using a Hardware-in-the-Loop Validation. *IEEE Transactions on Power Electronics*, 32(6):4421–4431, Jun 2017.

[154] Linfeng Zheng, Lei Zhang, Jianguo Zhu, Guoxiu Wang, and Jiuchun Jiang. Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model. *Applied Energy*, 180:424–434, Oct 2016.

[155] Yimin Zhou and Xiaoyun Li. Overview of lithium-ion battery SOC estimation. In *2015 IEEE International Conference on Information and Automation*, pages 2454–2459. IEEE, Aug 2015.

[156] Chunbo Zhu, Jianing Xu, Kai Liu, and Xiaoyu Li. Feasibility analysis of transportation battery second life used in backup power for communication base station. In *2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, pages 1–4. IEEE, Aug 2017.

[157] Fangfang Zhu, Guoan Liu, Cai Tao, Kangli Wang, and Kai Jiang. Battery management system for Li-ion battery. *The Journal of Engineering*, 2017(13):1437–1440, Jan 2017.