

Model-Driven Software Development and Verification Solutions for Safety Critical Battery Management Systems - A Quantitative Evaluation of Probabilistic Inference & Artificial Intelligence Methods

Von der Fakultät für Elektrotechnik und Informationstechnik
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von
Christian Fleischer, M.S.
aus Dettelbach

Berichter:
Univ.-Prof. Dr. rer. nat. Dirk Uwe Sauer
Associate Professor Dr. Erik Schaltz
Tag der mündlichen Prüfung: 22. Januar 2020

Diese Dissertation ist auf den Internetseiten
der Universitätsbibliothek online verfügbar.

AACHENER BEITRÄGE DES ISEA

Vol. 142

Editor:

Univ.-Prof. Dr. ir. h. c. Rik W. De Doncker

Director of the Institute for Power Electronics and Electrical Drives (ISEA)

RWTH Aachen University

Copyright ISEA and Christian Fleischer 2020

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the publisher.

ISSN 1437-675X

Institute for Power Electronics and Electrical Drives (ISEA)

RWTH Aachen University

Jägerstr. 17/19 • 52066 Aachen • Germany

Tel: +49 (0)241 80-96920

Fax: +49 (0)241 80-92203

post@isea.rwth-aachen.de

Acknowledgements

Now the time has come to let go this work..., I acted with the means that were available to me. To the end, I understood, that I only could scratch the surface, leaving room to a more worthy and smarter successors to built upon these results.

2008/2009, the recession in most parts of the world is resulting in an complete standstill. As consequence of the European Economic Recovery Plan agreed in December 2008, Germany gained momentum and an economic upturn by means of adopted measures, e.g. extensive state investments for businesses and industries in the sector of electric mobility, resulting into a bold federal government's goal of bringing one million electric cars onto the road by the year 2020 (by now we know that will not happen). I am thankful, not the least of which is the fact that Universities could increase their research plans in this topic. Then, obviously, I could not know where the journey is heading, and that in the end I have really been backing on the right horse. Because, as today, i am right in the middle of the pulsating heart of technological transformation and an active contributor within pioneering innovations to shape mobility towards 100% sustainable future.

Therefore, my gratitude goes first and foremost to my doctoral mentor Prof. Dirk Uwe Sauer, giving me the opportunity to research on the most dominant turning point topic in the automotive industry, as electrification with its battery technology is predestined as replacing conventional propulsion systems. Joining the institute 2009, I had not clear indication of direction in which things are heading: At some point, Prof. Sauer built up an world-wide leading and well known institute in Electrochemical Energy Conversion which I could feel part of. Not least because of his demonstration of a humble leadership, he was always supportive also especially when things have not turned out as intended. He also provided the opportunity to me to become a visiting researcher at DTU / Aalborg University / Lithium Balance in 2012, which prepared the ground for my future workplace. That was the time I could get to know Prof. Erik Schaltz, Associate Professor in the Department of Energy Technology at Aalborg University. During our project in which we installed the first university based battery hardware-in-the-loop test bench, I had the chance of many interesting, exciting, and insightful discussions around battery systems with Erik. I am very thankful to him for taking his time and contributions to my work and for accepting to be my co-supervisor. I would also like to thank all my colleagues at ISEA, i could spend my working time together. Especially I want to mention Wladislaw

Waag, who was my technical mentor when I joined ISEA, who brought me to a topic that he has already kept him busy before, also Nsombo Nlandi, a humble colleague I could always count on, especially in the field of embedded programming. I would also like to take this opportunity to thank the staff of ISEA and my former students for their support, especially Yizhou Zhang, Ze Zhang, Sujitah Davidson for helping getting the toolbox up and running. Especially thanks goes out to Lothar Pfeifer from Esterel Technologies, which is the leading provider of software development and critical systems solutions like SCADE. Evaluating the tools during my first job, he supplied me afterwards with additional evaluation licenses to implement the BMS architecture and algorithms. He also provided the build of the multi-core code (not commercial available yet). What would be work-life, without a strong private-life which charges your batteries when needed. Friendships beyond borders and nationality, here, I would like to thank my friends, the Rönnerfors family for the amazing tacco nights and brotherhood, the Navarro/Cazares family for the refreshing bible study discussions and deepest friendship, all my friends from the CENTRO church community, welcoming us to our new home.

My family, the divine foundation united in the bound of love and respect to each other, I want to thank my parents in law, Frank and Anabella, for the continues prayers, their hospitality when we were escaping the winter month, for guidance and support also in raising children, and the childcare allowing to go out to have dates again, Santiago, for his fun way as a person. My sister Annemarie, who has to bear with my bad teacher jokes, but still always being there for my family, providing educationally well-thought-out kids presents and for making sure that our children learn a correct German grammar.

Meine besondere Danksagung geht an meine Eltern, Georg und Anna, die mich alle Jahre massgeblich unterstuetzt und gefoerdert haben, die viel verzichten haben um ihren Kindern die beste Ausbildung und Start ins Leben zu ermöglichen. Danke fuer eure Liebe, eure harte Arbeit, dass ihr liebevolle Grosseltern seid und stets hilfreich zur Seite steht, ohne euch wäre diese Arbeit nicht gelungen.

In truth, I could not have achieved my work without a strong support at home, my beautiful wife Andrea, for many hours, days and weeks I had to pull back leaving you with lots of responsibilities and work. But you kept your patience during this time, also supported me with advice and gave me faith because you understood me, my strength but also my many weaknesses.

Felipe, Isabella, and Lukas, what drives me forward is the love I have for you!

Thank you God for everything in my life!

Abstract

The automotive industry is reaching an unprecedented level of turning point. The new challenges are wide-ranging, the future of combustion engine has been signed and sealed, as electrical vehicles with their high-voltage (HV) batteries are predestined as replacing propulsion systems. Today, SW methods and algorithms are not mature enough considering that the HV traction battery is a complex, electrochemical device in a new application, with the prerequisite in knowledge and understanding of the working principles including aging mechanism. These cause complex multi-time scale dynamics by superimposed interactions of chemical reactions which are complex to model. In recent years the need of battery systems and its advanced SW has increased significantly in order to reach performance, energy, lifetime and safety requirements. At the same time, the cost must drop to compete with conventional propulsion systems. Therefore, the question which put OEMs at stake is how to use the battery in an optimal way to ensure a competitive HV traction battery system. Therefore, this work discusses model-based battery management design, which does not mean applying visual methods to address the problems, rather we are using qualitative models to represent battery behaviour, e.g. looking at the electrical properties of the voltage response when applying current to the battery. Providing accurate power and energy estimates to the vehicle propulsion system is part of the on-line battery management primary objective. This work provides a set of adaptive algorithms for online battery state- and parameter estimation evaluating different aspects, such as their efficiency, robustness, accuracy and implementation effort on target. Bayesian optimal filters for non-linear systems, linearization methods e.g. extended Kalman filter, and other numerical approximation methods (Gaussian sum, iterative quadrature, and deterministic sampling approximation) are quantitatively benchmarked. The comprehensive discussion involves the definition of the estimation framework which is integrated for state- and parameter estimation using joint and dual estimation. Additionally, new estimation techniques from the field of machine / deep learning are introduced to predict battery lifetime. The Bayesian estimators are benchmarked in regards to various battery equivalent circuit models as they are needed for accurate voltage, state-of-charge, parameter or available power / energy estimation. Battery management systems are safety-critical software systems, containing code which failure or malfunction can trigger or contribute to a hazard jeopardizing people's lives. The quality of released SW is subject to the effort of which level the testing is executed. Therefore, SW testing and hardware-in-the-loop BMS validation on functional and non-functional level is introduced, leading to a novel battery system simulation environment including fault insertion. Additionally, a new way of ISO26262 complaint model-driven software development for BMS is discussed, introducing a new tool-chain for SW architecture and model development on multi-core target

CPUs optimizing computing power for individual cell monitoring. The introduced ecosystem fulfils the requirements during development phase with traceability, validation and verification.

Contents

Acknowledgements	iii
Abstract	v
1 Motivation	1
2 Introduction	5
2.1 The Profound Impact of Electrification: From Computer-like Car to a Car-like Computer	7
2.1.1 Increasing Complexity in Automotive Hardware and Software Systems	8
2.1.2 Trends in Software Defined Cars	10
2.1.3 Thesis Structure	15
3 State-of-the-Art Battery Management Systems	19
3.1 Architectures and Subcomponents	19
3.1.1 High-Voltage BMS HW Components	19
3.1.2 High-Voltage BMS SW Components - Requirements and Functionality	23
3.2 Literature Review and Objectives	34
3.2.1 State of Charge Estimation	36
3.2.2 Impedance Estimation	37
3.2.3 Capacity Estimation	38
3.2.4 Available Power Estimation	39
3.2.5 State of Health Estimation	40
3.2.6 Remaining Useful Life Estimation	41
3.3 Analysis of State-of-the-Art Battery Management Systems	42
3.3.1 BMSUT #1 - System A	42
3.3.2 BMSUT #2 - System B	44
3.3.3 BMSUT #3 - System C	45
3.3.4 BMSUT #4 - System D	46
3.3.5 BMSUT #5 - System E	47
3.3.6 BMSUT #6 - System F	48
3.3.7 BMSUT #7 - System G	48
3.3.8 Benchmark of Evaluated BMS HW Functionality	49
3.3.9 Benchmark of Evaluated BMS SW Functionality	50
3.3.10 Summary	51

4	Principles in Estimation of Signals and Systems	53
4.1	Basic Concepts of Estimation and Problem Statement	53
4.2	Assessing Estimator Performance	60
4.3	Bayesian Philosophy and Estimation	72
4.3.1	Statistical Inference in Estimation	73
4.3.2	Summary	76
5	Probabilistic Inference and Bayesian Recursive Estimation	77
5.1	Hierarchical Classification of Models	77
5.2	Recursive Bayesian Estimation Theory	79
5.3	Approximate Solutions to the Probabilistic Inference Problem	80
5.3.1	Methods Using Gaussian Approximation	81
5.3.2	Methods Using Non-Gaussian Approximation	100
5.3.3	Resampling	110
5.3.4	Different Resampling Methods	112
5.4	Estimation Framework - Systems, Problem Statement and Implementation	123
5.4.1	Different Estimation Problems	123
5.4.2	Methods for Implementation	127
6	Evaluation Methodology for Bayesian Estimation and Machine Learning	133
6.1	Battery State-Space Models	134
6.1.1	Equivalent circuit/semi-empirical models	134
6.1.2	State-Space Models for Battery State and Parameter Estimation . .	137
6.2	Implementation of Non-Linear Filtering Concepts for BMS	145
6.3	Benchmarking Criteria	147
6.4	Neural Networks and Deep Learning	153
6.4.1	Shallow learning	156
6.4.2	Deep learning	159
6.5	Convolutional Neural Networks	161
6.5.1	Input Images	161
6.5.2	Convolution	162
6.5.3	Nonlinear Activation	163
6.5.4	Pooling	163
6.5.5	Fully Connected Layer	163
6.5.6	Architecture	164
6.6	CNNs on Battery States Estimation	164
6.7	Numerical Benchmarking of Estimators	183
6.8	Results	187
6.8.1	Application Example - Non-linear Battery State Estimation	189
6.8.2	Results - State Estimation	190
6.8.3	Results - Joint Estimation	192
6.8.4	Results - Dual Estimation	194
6.8.5	Final Verdict	196

7	Hardware-in-the-Loop Testing for Battery Management System	199
7.1	Hardware-in-the-Loop Simulator	202
7.2	HIL Battery System Simulation Framework	204
7.2.1	Lithium-Ion Electro-Thermal Multi-Cell Model	204
7.2.2	Hardware-in-the-Loop System Model	205
7.3	Commercial BMSs Under Test	210
7.4	Functional and Non-functional Target Testing	211
7.4.1	Fault insertion environment	212
7.4.2	The Threats: Faults, Errors, and Failures	213
7.4.3	Fault Classes	215
7.5	Battery Management System Testing Using Fault Insertion Techniques . .	216
7.5.1	Background to BMS Testing	216
7.5.2	Strategy on HiL BMS Testing	219
7.5.3	Experimental Setup and Results	221
7.5.4	Summary	239
8	Developing ISO 26262–Compliant Safety-Critical BMS Application	241
8.1	ISO 26262 - Functional Safety Concepts and Testing for BMS SW Development	241
8.2	ISO26262-Complaint Model-Driven Software Development	247
8.3	Model-Based Software Development	252
8.3.1	TargetLink Ecosystem	255
8.3.2	SCADE - An Aerospace Solution for Highest Level of Safety in SW Development	255
8.3.3	Comparison: Traditional vs Full Safety Critical Toolchain	256
8.4	A Model-Based Approach for Multi-Core - Multiple Battery Cells Estimation	257
8.5	Parallel Optimization for Computational Efficiency	263
9	Conclusions and Outlook	269
A	Appendix	273
A.1	BMS HW evaluation	274
A.1.1	System #1	274
A.1.2	System #2	276
A.1.3	System #3	277
A.1.4	System #4	278
A.1.5	System #5	279
A.1.6	System #6	281
A.1.7	System #7 (Lithium Balance)	282
	List of Figures	321
	List of Tables	327
	Bibliography	329

1 Motivation

Rethinking the Car New: From Nuts and Bolts to a Software-Driven Electronics Device.

Future perspective development leads to world-wide scarcity of resources and ecological damage which forces necessary rethinking about unsustainable development, about the urgent need to take action on climate change. Research and development of new, future-oriented mobility systems is becoming increasingly important. The development of sustainable technologies, requires rethinking, especially with regard to the vehicle propulsion. New drive and battery concepts must be developed while the broad infrastructure has to be redesigned [1]. This causes the automobile industry to experience the biggest transformation process since its invention in 1886 (combustion). This is enabled by the fact that new technological requirements and stringent legal regulations (CO₂ reduction) force the market participants to face a rather revolutionary rethinking on developing new propulsion systems. In the past, OEMs have focused their efforts in the area of comfort, passive safety and efficiency, while the optimization of consumption was treated as a minor issue. However, in order to meet these regulations, changes must drive technological progress of new propulsion systems away from fossil fuels. Here, the electro-mobility with its use of sustainable generated electric power, and advantages in "Well-to-Wheel" efficiency is seen as the answer to climate change as it will reduce the 20% CO₂ pollution caused by traffic as well as in other applications (e.g. residential battery storage). Additionally, urbanization also drives this development, as the needs for individual mobility and clean air (megacities) is self-evident. Electro-mobility, with its electric buses, tramway or modern electric subways using KERS (kinetic energy recovery system with high voltage battery), will prove that mobility and sustainability can go hand in hand. Generally, propulsion systems are designed using either internal combustion engines and/or solely electric drive-trains. The intelligent combination of electric motor and traditional combustion engine can be differentiated by the grade of hybridization, starting at micro-, mild-, full- (HEV), or plug-in hybrid (PHEV). Additionally, from the architectural design viewpoint, the setup of the power-train components is also important and categorized into serial, parallel or power-split hybrid. The drive train of a battery electric vehicle (BEV) is propelled by a purely electric motor, while the energy can be provided from a variety of battery systems which may differ in capacities, energy and power density. Lastly, to complement this discussion, the concept of fuel cell vehicles (FCEV) shall be mentioned, as this propulsion concept carries a battery on-board for recuperation purposes. While

the combustion engine was optimized over the last century and is therefore well understood and industrialized, electric propulsion systems (EPS) in automotive applications still remain at its beginning (standardization – > modular approach; industrialization – > away from prototype concepts) and has a long research haul ahead, as also energy supply has to be guaranteed. However, to get to the point where electro-mobility has an important contribution to sustainable transport system, the industry still must solve particular difficult challenges. Therefore, the focus of the measures lies in the improvement and reduction of broad variety of EPSs, resulting in minimal efforts in re-designing the systems for entire platforms and wide applications (stationary), leading to overall component costs reduction. From the ecological point of view, the overall global carbon footprint must be reduced to justify EPSs. However, the key challenge and the main field of action lies within the drive-train with its high-voltage traction battery system (HVTBS), and this must be identified as the biggest problem as further optimizations in safety, costs, specific energy and performance, internal and external condition handling (e.g. temperature), reliability and longevity must be undertaken to also maximize the electric driving range. The efforts must start at the cell level, here, lithium-ion batteries (LIB) are the dominating technology fulfilling the performance requirements in automotive industry. In comparison to older technologies, e.g. lead-acid (PbA) or Nickel–metal hydride (NiMH) batteries, LIB allow for good energy/power densities, relatively high full cycle numbers under higher efficiency. In contrary, improvement in LIB to achieve these goals is depending on complex development of new chemistries including new cathodes, anodes, electrolytes (currently a yearly average of 8% increase of energy density) and cell design without jeopardizing battery safety. Connecting individual cells together and grouping them into modules until sufficient voltage to the drive-train can be provided defines a battery system. To operate a LIB, hardware (HW) as well as software components (SWC) are required, including electronic circuits which together combine to a battery management system (BMS). It ensures actively and passively that connectors/fuses (dis-)connect the battery from the load, sensors to measure e.g. temperature, voltages, current, isolation, thermal system to control battery temperature, embedded control units for battery monitoring, state estimation and controls to coordinate and manage the activities of the HVTBS. Also, the mechanical and electrical integration of the LIB is of great challenge, structural integrity when integrated in the vehicle body as well as low and high voltage harnesses (EMC challenge) must be understood. Beyond the need of basic functions of a BMS (sensing), intelligent SW components for tracking and monitoring the battery condition are required. It must be pointed out that SW methods or algorithms of the BMS has the most important task to correctly diagnose the measured signals and to estimate current HVTBS operational state and parameters to predict future performance limits (short: power boost in HEV, long: remaining energy for range in BEV) which are parameters that cannot be measured directly [2]. On the other hand, today, SW methods and algorithms are not mature enough considering that the HVTBS is a complex, electrochemical device in an new domain/application, with the prerequisite in knowledge and understanding of the working principles including degradation and aging mechanism of HVTBS. The critical information to extract consists of the state of charge (SoC), power & capacity fade. Electrification, particularly HVTBS, has a big impact on SW development in the automotive industry. The exponential growth of lines of code

within an intelligent BMS is a result of the increased functionality, safety requirements and variants of HVTBS.

But, a successful integration of electric mobility requires also cost reduction. As the battery is the single most expensive system in an battery electric vehicle, while it is both the most complex and most difficult system to integrate, also in terms of size and weight, these new batteries are still the least understood system and therefore there is a lot of room for optimization between these domains, materials, system design (electrical & mechanical design), thermal system and battery monitoring - estimation - controls (SW). As of today (May 2018) VW announced that it just ordered \$48 Billion in electric car batteries (Source: Fortune.com). If we can achieve with intelligent software methods an increase of usable energy/power of the battery system, which may vary from new system to aged system between 1-5%, this will prevent also an over-dimensioning of the pack and in the end reduces TCO and high potential cost savings thanks to an intelligent battery management system, no other current SW application offers this potential in cost reduction in todays automotive industry.

The automotive industry is reaching an unprecedented level of turning point. The new challenges are wide-ranging, the future of combustion engine has been signed and sealed, as electrification was predestined as replacing propulsion systems, also the question who or what will drive our cars later, questions which put ordinary OEMs at stake now as mobility concepts turn into software-defined propulsion systems. In the past, OEMs have increased demands on the suppliers of SWC, but, as BMS SW becomes a strategic product and driver for innovation, driven by increased battery lifetime by intelligent control schemes for more exciting driving experience, the development will take place more in-house. As electro-mobility and electric propulsion systems are relatively new, this pioneering role in BMS SW development brings a technological advantage and separates manufacturers. To sum up, BMS intelligent software functions implemented with the best knowledge of LIB working principles must increase, resulting in a higher demand on computing power in future xEVs.

2 Introduction

Besides the continuous efforts in advancing battery cell materials to ensure safe and reliable operation, the most important safety mechanism of a battery pack in xEVs is the BMS. It is designed to monitor signals, estimate SoC, SoH (capacity, impedance), SoF (available power/energy), as well as control (thermal, charging...) and protect (stay within the battery limits (power, voltage, current)) the battery system. BMS can be defined as a system of electronic safety circuitry (HW) and SW components performing tasks, a software-driven electronics device for estimation and control, installed in the battery, which uses sensors to monitor the systems (voltage, current, temperature, humidity,...) with the task to control the performance of a battery according to certain set of principles or procedures, while dealing with measurement errors, noise, and uncertainties [3]. The system to be controlled is dynamic, a system whose state evolves with time over a state space where the output cannot be clear identified by its current input, while the state space cannot be directly measured. In principle, as the battery is a complex, electro-chemical device, from a controls point of view, the battery system is defined as a highly non-linear dynamic system, depending on temperature and usage profile, which changes its characteristics during lifetime due to degradation. Hereby, the major task is to ensure that individual cells of a battery system are charged and discharged according to the manufacturer's specifications, meaning, that cells have to be operated always inside a safe voltage, temperature and current window. However, to guarantee for safe operation, an advanced BMS must not only monitor and control the described limits. In fact a wide range of functions have to be included in a BMS, including, in particular, measuring, protection, balancing, analysis, external communications, logging and telemetry. All these management functions and BMS key objectives can be defined as followed [4]:

- Increase safety, reliability and optimized usage.
- Single cell monitoring and protection.
- Intelligent energy usage efficiency for optimized user case scenarios.
- Reduce battery degradation to extend battery lifetime.

Therefore, to be a leader in any application with large LiB, e.g. home-storage or automotive, it requires sophisticated software methods and algorithms to estimate battery condition and unmeasurable parameters to control the battery short- and longterm.

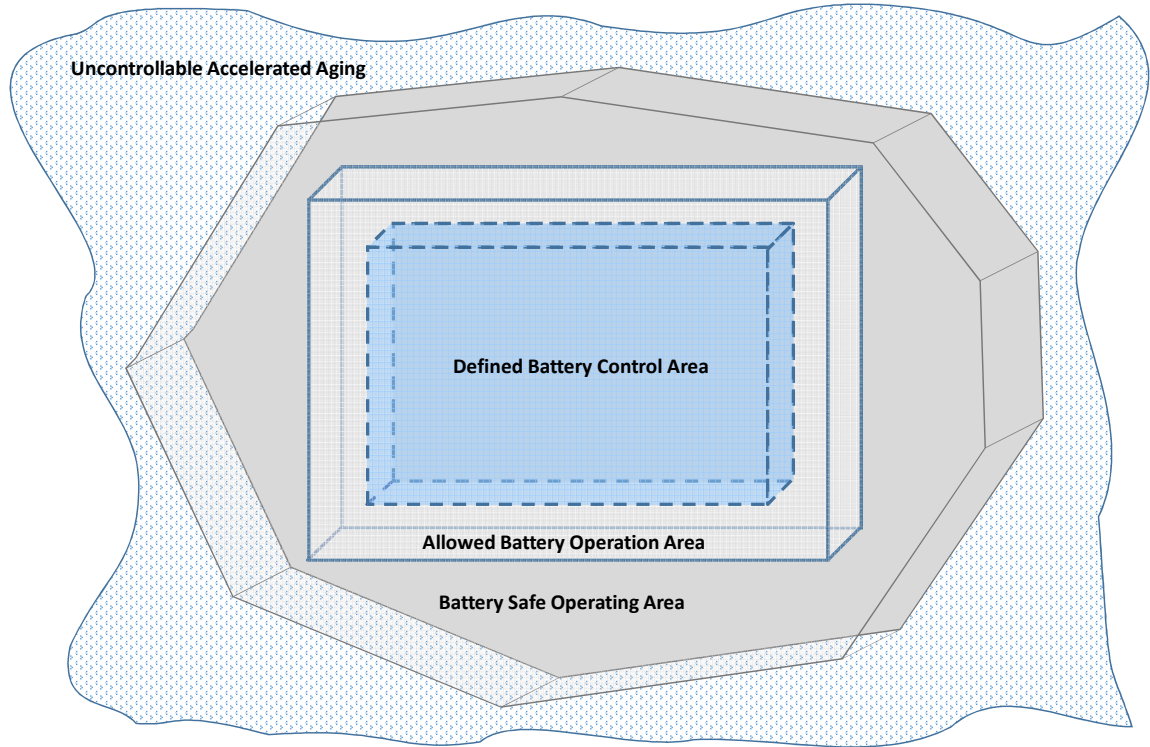


Figure 2.1: From battery monitoring to battery management - an illustration (modified from [5])

Looking back at the BMS tasks, monitoring is important, but rather low-level functionality with minor gain and potential for improvement. On the other hand, estimation & control has a big impact when optimized, as on the vehicle level many functionalities are connected and the driver will feel the impact of a poorly designed behavior. It is therefore the key part of our discussion as these SW methods objectives are to enable for all systems in the car which depend on the battery, to use the battery optimally, such as the supervising energy management unit. As mentioned earlier, the BMS SW has to perform certain actions, such as estimation of internal states, which are unknown and unmeasurable. As an example, Fig. 2.1 was presented at the Vehicle Electronics & Connected Services 2018 [5], showing an overall random cloud with sub-areas representing different operation areas and fading borders. The BMS supervises the system to stay within the battery safe operating area (BSOA), as long as the system remains in this state (short time) it is feasible. Staying within the allowed battery operation area (ABOA) will guarantee for the calculated and expected lifetime requirements. However, this border between BSOA and uncontrollable accelerated aging area (UAAA) cannot be defined as it depends on so many factors (temperature, history, DOD, C-rates,...). It

is a moving target, non-measurable, and crossing this border regularly will cause faster degradation and results in neglecting the requirements, in terms of range (energy fade due to capacity loss) and vehicle performance (power fade due to increase of internal resistance). The usage window can be defined by cell testing and SW method verification comparing system tests with estimated values based on real measurements. *But what does it mean for the vehicles drive-ability?* The necessary information within these areas is based on how much energy and power is available and will be available for the short and long term time-horizon. Therefore, the better our defined battery control area (DBCA) is moving towards the ABOA, the less pessimistic the estimations is hence the battery utilization rather optimized. As we are crossing the ABOA with the DBCA, that means we were too optimistic, we estimated more performance than what is actually available, resulting in a sudden power restriction noticeable to the vehicle passengers. This does not mean that we will damage the battery long term, as the control algorithm will correct the overshoot quickly to be inside the DBCA within milliseconds or up to seconds, until the DBCA is close to ABOA. *But what makes this so difficult?* From the control theory point of view, the battery is a dynamic system with memory, it is highly non-linear, depending on C-rate, DoD, or temperature, which for example leads to higher losses at lower temperatures, as the stored chemical energy cannot be utilized efficiently causing available energy and power to drop. In addition, based on how we are using the battery, its degradation will follow a different path. The BMS also has to follow the rapid change in chemistry change, almost every 2 years we see new updated material combinations in cells which leads to different behavior and higher energy density. An intelligent BMS with an expected lifetime of 10+ years shall account for system changes. It must also be pointed out, that the system is rather complex as it contains several hundred of cells, each works as a system for electrochemical energy conversion/storage and by redox reactions electrical energy is then generated through conversion of chemical energy. This reflects a heavy burden on the BMS and its computational requirements in the future, as no infinite CPU performance can be justified. Hence, smart SW methods must be applied to handle the complexity throughout the battery system. Summarizing the BMSs tasks, and its interface to the vehicle, its providing short-/long-term power and energy prediction, may control the thermal management fully or requests the vehicle supervising unit for heating/cooling needs and interacts with the charging systems. On the module level downwards, understanding the charge levels and battery states such as SoH, and how the system is going to be used, is of highest importance and challenge.

2.1 The Profound Impact of Electrification: From Computer-like Car to a Car-like Computer

The automotive ecosystem experiences a wind of change towards technology driven environment, electric drive-train, battery technology and artificial intelligence developments are software centric mechatronic domains with few moving parts. Before this disruption,

the car was a mechanical product with mechatronic and SW functions (~ 1 million lines of code), today, extremely complex systems (e.g. PHEV) of connected SW and mechatronic functions ($\sim$ hundreds of million lines of code). *Advanced battery management systems, what does it mean for the electric system and software?* In battery management systems, many cells are interconnected, lots of sensors providing lots of data which has to be fused. This requires coordination and timing across a distributed system with many ECUs. This data needs to be analyzed while complex and intelligent algorithms decide and control the vehicle in a safe manner for all times. High performance computational resources are needed to process high-bandwidth data in dependency of many known and unknown variables.

2.1.1 Increasing Complexity in Automotive Hardware and Software Systems

Over the last years, the complexity in automotive E/E systems including battery management systems is increasing. Typical key figures regarding complexity are the number of electronic control unit (ECUs) and the communication between these embedded systems leading to a very high number in of on-board signals. Also, for an successful penetration of electric vehicles on the market, one important factor is the topic of cost efficiency, therefore, at the highest increase of these numbers is the increase of legacy ECUs, meaning that system designers making use of the fact that ECUs which are used in current systems are taken from the previous models with minor changes. On the other hand, this limits the development of new functions when major parts are not allowed to be changed and leads to logistical challenges in the software integration. Additionally, an increase of bandwidth on busses and networks for connecting the vehicle to the back-end has to be addressed. This will be of new importance especially for China since the Chinese standard of remote monitoring system for electric vehicles has been circulated in draft form including 5 parts: GB/T for electric vehicle lithium-ion battery safety requirements. This standard will allow remote control by operators and therefore presents new challenges in regards to complexity, and security leading to an increase in demands in all domains. Additionally, the current communication topologies like CAN-bus will encounter limits in bandwidth considering functions for big data, connection to back-end with many BMS sensors generating a lot of data. To decrease the complexity of BMS architecture a topology with few domains has to be considered. The architectural development has to cluster the domains after functions rather than hardware nodes. The goal is to minimize the amount of communication between domains (CAN, LIN, FlexRay) with very large design space of parameters. This requires that the configuration of the network takes place in an controlled way in regards to the distributed ECUs, gateways and routings. Good documented processes is a must during development and integration into the battery system. The key to successful documentation considering ISO 26262 lies in the development of good tools, which might not always to be obvious. The complexity diagram for software functions and tasks of an multi-core engine control unit is illustrated in terms of a call

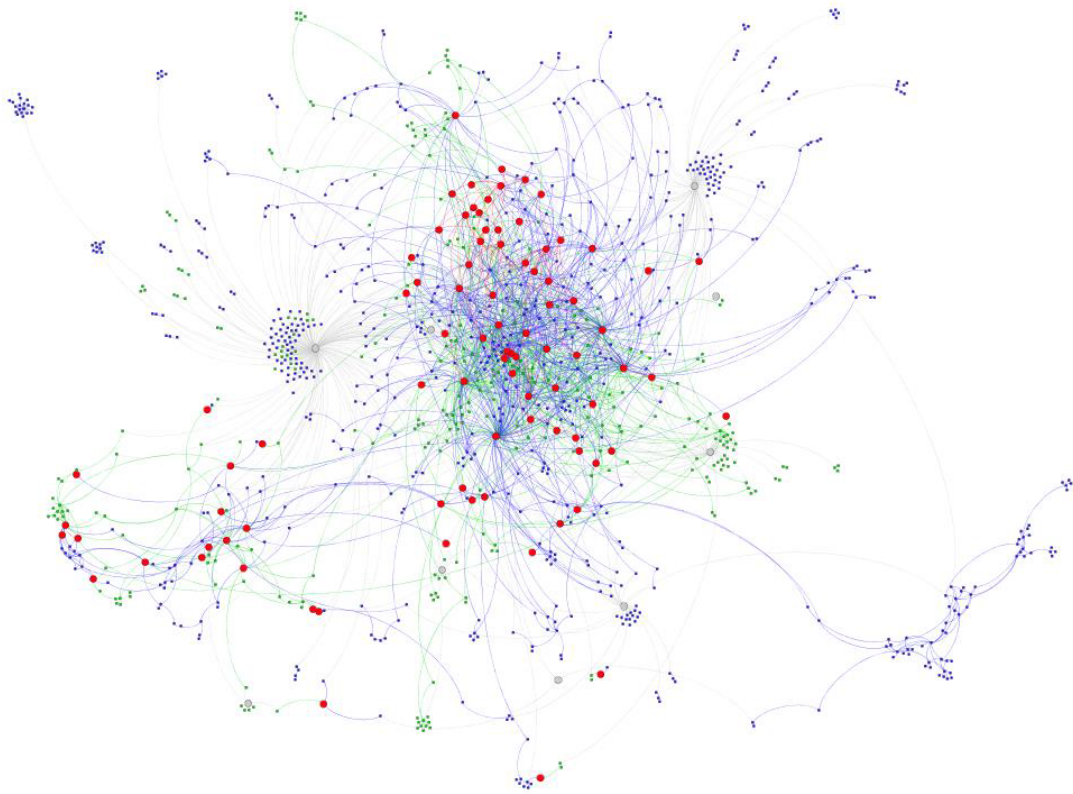


Figure 2.2: Complexity of automotive software functions in terms of tasks execution (Trends in Embedded Systems and Vehicle Networks — Gunnarsson, BMW AG — 2016-05-11)

graph in Fig. 2.2. The figure shall illustrate the different software tasks represented by blue dots running on core 1 while the green dots representing the tasks running on core 2. The red dots are non-movable tasks and fixed on one of the cores handled by the operating system. A software architect has to create an optimal SW architecture to be able to utilize the computing power and robustness while safety and security must be guaranteed. System integration nowadays already struggles with high complexity to fulfill requirements. On the other side, bringing features quicker to the customer requires shorter innovation cycles. Already the pressure of expectation generated by consumer products is building up where updates are on a regular base forcing OEMs to major changes. This especially affects the BMS systems as well, such as remote updates over the air to provide new features such as aging models for batteries, updates of characteristic maps, new operation parameters extracted from laboratory test, or data which was not available during start of production. BMS manufacturer have to enable innovations as quickly as possible to make it available to the fleet, and provide the customer a new experience every day.

2.1.2 Trends in Software Defined Cars

To reduce costs for BMS, new technology has to be developed in means of OEM cooperation without competitive advantages. To give an example, AUTOSAR is a worldwide development alliance to establish an standardized software architecture for embedded systems in automotive applications. It is intended to reduce the legacy ECU costs by increased transferability of software and safety requirements. One aim would be the same for a hardware platform, from which similar benefits may result. These challenges and trends have to be well understood with a number of new players on the market, such as Apple where rumors suggest that it is developing an electric iCar within the Titan project. Apple has gained experience with CarPlay, connectivity, and user interfaces, Google brings highest competence and knowledge in big data and maps, Tesla as a complete new OEM with over-the-air remote updates (OTA) which is well received by its customer. These trends lead to a paradigm shift from evolutionary development of solving problems to rather revolutionary approach. EVs accelerate the evolution from the electric system fail-safe to fail-operational architecture, allowing the driver to be able to control the vehicle even if a failure (e.g. in battery system) occurs. Nowadays, typically software updates are scheduled via mandatory or voluntary recall due to problems with vehicle functionality. In most cases this can be resolved by a software update. Depending on the servility and urgency of the update, the recall has to take place immediately or as part of a upcoming maintenance service. The traditional method is quite complex, may take couple of hours, expensive since flashing equipment is costly and therefore only a limited number of specialized workbenches is available. Modern methods for remote software update may solve this problem. For today, most OEM are only able to do so for some off-board functions, this is mainly caused by the integration complexity for entire software logistic process, where the firmware is built for hundreds of ECUs, while the suitable ECUs for updates have to be defined to assure the consistency of the vehicle configuration rather than the technical challenge of updating. Another aspect is the local processing of all the measurement data to control the system as in classical automated control approach. There might be a change to big data (BD) processing, where vehicles and their BMS communicate sensor data to the back-end and receive feedback with control signals and system state updates. The services will lead to an increase of communication requiring a shift from limiting factor of single based communication to IP based networks as a complement. Several fundamental legal, economical and technical questions as well as liability need to be answered. Topics for the BMS architecture need to meet safety standards in an intelligent way because OEMs cannot afford to use the amount of redundancy and development of functions with different teams like happening in the avionics industry. Therefore, basic functions and their high versions together with the generated redundancy shall be used without the need to develop multiple instances of the same function. An important aspect has to be answered, such as how is it possible to guarantee requirements in case of degradation for safe-operation. Under all circumstances, strategies need to be in place to fulfill a working system also in presence of failure. As mentioned earlier, new trends in terms of E/E architecture will hit the market for more advanced connectivity for battery control for big data analytics, to understand how batteries de-

grade in the fleet. Hereby, all layers of the E/E architecture including BMS is affected by these trends. From commercial products well known, software as a product/service (SaaS) featuring updates even on demand for some time shall be hosted in the cloud which leads to additional integration challenges for non-AUTOSAR software components. Even open software platforms may be profitable to increase software system integration to achieve safety requirements. Therefore, the system architecture design is more critical to prevent overload the ECUs with topics of buffer size or point-to-point communication to isolate high bandwidth functions and separate control signals from measurement data. On the BMS ECU side, multi-core ECUs become more and more common trend which assures more computing power for BMS for on-line monitoring and safety functions, however, if the software is not designed for multi-core purposes, it may even be counterproductive, at the cost of increasing complexity for software integration. Today, individual attempts on multi-core task coordination between different cores is investigated, where cores are used independently from each other, but to get the benefit of computational power of all cores the cores with scheduling need to be coordinated in a effective way. Traditionally the ECUs used in vehicles for BMS are general purpose processors, depending on the overall BMS system, might still be sufficient, such as for intelligent battery disconnect unit (BDU) which contains lower software complexity, or safety critical calculations, but the need for a new family of special purpose processors will raise quickly for more intelligent BMS platforms. Special purpose processors such as GPU (~ 3 TFLOPS = 25x Iphone 6s) are not necessarily used for graphic calculations since they perform well in sensor fusion, DSPs can take over complex signal processing algorithms (~ 5 GFLOPS) as well as application specific ICs or system-on-chips including already proprietary algorithms. The obvious way to write code is normally done in a sequential manner, but this cannot be applied to multi-core ECUs especially the communication between task will become very complex and important also for the timing, it needs to be taken care that timing of tasks is well known and deterministic for all components in the E/E system. The classical older versions of AUTOSAR were not suitable for multi-core applications but because of general direction of requirements especially for intelligent and adaptive BMS to move towards high computing platforms introduced the AUTOSAR adaptic platform for multi-core. It is quite controversial, the software component development increases in complexity, on the other side agile development shall lead to faster cycles of development. In distributed battery management systems combining different sensor data (sensor fusion) gives different sets of data, accurate time stamps and synchronisation is required. On top of all trends, reliability, safety and tolerance to fulfill certain automotive safety integrity level (ASIL) means significantly increased demands on development processes, increase of complexity by introducing HW and SW redundancy resulting in significant effort in testing and validation. In terms of sensor redundancy, the mainstream used e.g. dual current measurement methods in case one breaks down, HW redundancy such as dual processors to grant battery monitoring, or even several nodes cooperating with each other, in case one fails, the substitute can take over or the vehicle control unit may degrade the functionality of the system. Additional power supply redundancy or in communication link to guarantee that any kind of data or signals may reach the receiver via a second path.

Much of the technology we see in the current car is already outdated by the time they

are released to the market, caused by the long development and manufacturing time. In order to experience new technology and software updates, mostly the only way is to buy a new car. OEMs don't want to risk to ship software that is not entirely functional and proven. We have seen that in order to handle the increasing complexity in functions and software development combined with the requirements in shorter development cycles to reduce the time to market structured methods for BMS and E/E system are necessary to allow HW/SW development to keep up with cell chemistry releases. New OEMs from other domains enter the market changing the conventional automotive rules with increased technology. To achieve that, structured methods for a revolutionary approach is needed to decrease the amount of legacy as much as possible. A paradigm change will take place with service-based communication and computing platform using many-cores. Safety aspects in battery systems are also of predominant concerns. Different layers of safety measurements have to be addressed, functional safety (FS) is one part of overall safety concept, and amongst others, such crash safety, unauthorized access (security), passive safety (airbag). That is the complexity for many developers, in FS we are not talking about specific algorithms, hardware or implementation, rather than referring to a safety of a function, such as for example, charging, is charging safe, is charging unsafe, is the operation of the battery system safe? In the beginning, it's just on the abstraction layer where we don't have implementation rather just looking at functions, can fast charging be unsafe? The answer is yes, because if some rely on it, and it fails, and it does not tell or warn the actor that the maximum state-of-charge, voltage, or temperature is critical, it can lead to a dangerous situation if the system does not change its behavior according to the incidence. But why do we need it now, how could we manage in the past? The difference is, that the systems and including battery systems get more complex, in hardware and especially software including connectivity, systems of systems, resulting in the computer to take control over the systems in the vehicle and 20 years down the road unforeseeable complexity in energy storage systems will arise. Therefore, a blue screen on the dashboard on the highway losing pack control with a message: *"Thank you very much for driving our car, unfortunately the system is out of control"* is by no means an option, therefore, we have to secure the software and the functions and safe guard the driver. Whatever is done for BMS SW, it will be connected to the cloud for diagnostic and service reasoning and hence to the Internet, making safety and security especially important. Secondly, complex architectures are developed for various systems increasing the complexity, while these systems which are deployed here must fulfill the tasks of functional safety which someone could argue to be part of the diagnostic system, it is not autonomy rather interconnected to other functions.

All three parts, on-board-diagnostics, functional safety, and cyber security have the purpose to detect and isolate (diagnose) faults or misbehavior of the system. What differentiate them is the reaction in the end. While the OBD for BMS is the safe guard in the battery pack up to vehicle level, e.g. hot ambient temperature with high speed may lead to overheat of the battery pack resulting in lowering vehicle power. Functional safety for example is, the battery impedance estimation algorithm is failing or the battery controller has a SW fault resulting in uncontrollable or extra ordinary vehicle speed. The consequence must be reaching safe state in BMS by switching off e.g. parts of the electronics

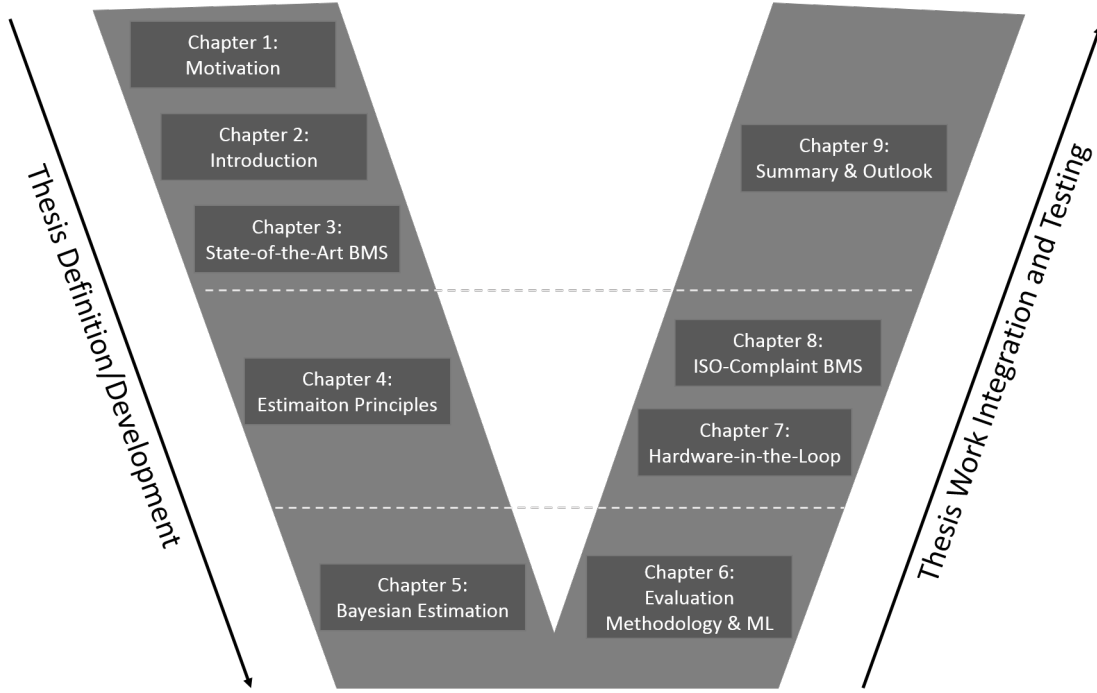


Figure 2.3: Content of work.

and get the system into safe operational state for the driver. Cyber security deals with threats such as wireless carjackers and is in the grey area between safety and security. It was shown that two hackers remotely hacking into a Mitsubishi Outlander with cell phone app to open the doors via an open WiFi access point which was unsecured implemented using only a 4 bit key, making it relatively easy to get access to the car. This key gives you access to the CAN bus and by knowing the commands everything is possible to do with the Outlander. Another example, a Jeep Cherokee which was driving on the highway many miles away. The outcome was that the brakes were disabled sending the car with its driver into a ditch [6]. None of the functions such as firewalls, authentication routines, HW encryption or hacking detection are available in most cars and is therefore a new field in security related diagnostics and makes security also safety problematic, especially for the BMS remote monitoring system. All the three methods have to be combined to secure and safe our BMS and car. The E/E architecture needs to be modified with shared local high performance computers directly connected to backend data center environment. The BMS will support over the air updates with updated parameters, resulting in the biggest challenge, how does the manufacturer handle the constantly increasing complexity of software, only when the industry can give an answer to this question, a large number of xEVs will successfully enter the mass market. Why do we need it, first we want to safe the driver and secondly the integrity of the company in terms of product liability.

How do we handle all this? Basic systems use characteristic maps, trying to describe the dynamics of a battery via extensive lab tests and storing these results as a look-up table in the car. Then during operation, real data is compared with the predic-

tive values from the look-up tables. A mismatch for these methods is expected as lab tests cannot describe all the operation space in real application. For this method, the DBCA will be far away from the ABOA and hence rather inefficient. Therefore, to get a hands-on user experience in driving xEVs and understanding the current solution on the market in terms of user experience/interface (UX/UI) and SW maturity/functionality are tested for Model S, i-MiEV, e-Golf, Zoe, Leaf, i3/i8, Fisker Karma, e-tron, and B-Class as shown in Fig. 2.4 and Fig. 2.5. The test drive was decided beforehand according to google maps and entered into the cars navigation system. It was of interest how well the software is capable in predicting the energy consumption for this test drive (not all cars had this function -;range anxiety), how the software can handle the driving behavior, how the system behaves when load is connected (air conditioning, heating-, seat and rear screen heating...), basically, how sophisticated current display functions are. It was a rather sobering realization that the SW functions were partly very basic (i-MiEV, Zoe) not capable in updating predictions based on current and past driving behavior. E.g. switching on the fan dropped the range by more than 50 %. Surprisingly, the so-called "best" system from Tesla also uses not very complex methods for routing estimation, as illustrated, the predicted value based on past driving history leads to an error of $\sim 10\%$ for range estimation (Zoe $\sim 13\%$, I-MiEV $\sim 14\%$). The BMW i3 uses on the other hand routing information and reflected a more positive feeling. The overall experience puts the EV UX/UI expectation back down to the reality of hard facts that the SW must be refined and improved, as the customer expect equal or better functionality in comparison to the conventional vehicles (as they pay more).

Instead? In recent years the need of battery systems and its advanced SW has increased significantly in order to reach performance, energy, lifetime and safety requirements. At the same time, the cost must drop to compete with conventional propulsion systems currently on the market. This trend has triggered funding shifts from governments and companies towards battery system research, e.g. towards new chemistry. However, the need is especially on further advanced control schemes which require more insight regarding the hidden inner states of a battery, which on a contrary cannot be provided by additional sensors. Instead, most solutions focus on proving an intelligent observer algorithm as an inexpensive alternative to estimate unmeasurable information.

Using! Model-based battery management design, which does not mean applying visual methods to address the problems, rather we are using qualitative models to represent battery behavior, e.g. looking at the electrical properties of the voltage response when applying current to the battery, as well as thermal properties to predict future cooling or heating needs caused by charging and discharging. These kind of challenges have been investigated in different domains using methods from statistical signal processing and automatic control theory. These methods shall be reviewed and implemented to provide a quantitative evaluation of different algorithms. As the described BMS study later shows, common architectural designs for BMS must be considered for SW development, e.g. centralized BMS, where cell monitoring units (CMU), also referred as to BMS slaves, battery electronic control module (BECM) referred to BMS master, are integrated on one single PCB. In contrary, the modular BMS architecture consists of individual module management PCBs placed close (normally on top) of the battery modules with normally reduced complexity, namely measuring, sensing and balancing function. The data is sent to the

BMS master to execute remaining functions such as analysis and system control.

2.1.3 Thesis Structure

First the *Motivation* chapter describes the profound impact of electrification on OEMs, and how traditional mechatronic solutions focused OEMs are forced to become a software driven company. The second chapter, Introduction, discusses new trends in software defined cars, and that the BMS control unit is a central component for all future vehicle functions (even in automated driving).

In *Chapter 3*, 7 BMS of different manufacturers were evaluated, to mainly understand the maturity of implemented SW, from which only 1 is commercial available. Due to the sensitive data, it is not possible to reveal the manufacturer names, rather they will be called BMSUT #1...#7 (BMS under test). 6 of the BMS systems are intended/partly already operate in battery electric vehicles (BEVs) while one system has is a multi-purpose solution (Lithium Balance A/S sBMS). It is important that during BMS HW and SW development many guidelines and aspects have to be considered to ensure usability for safety-critical battery systems, amongst other guidelines, following the ISO 26262 standard "Road vehicles - Functional Safety" is here considered during the own BMS SW development and for benchmarking these 7 BMS systems. Additionally, around 350 papers from 2014-2018 in regards to algorithms for SoX were grouped into categories to understand the research trends, which is used as input for our SW methods evaluation.

Chapter 4 introduces the principles in estimation and its problem, of signals and systems using nonlinear stochastic filtering, and what on-line, real-time estimation methods requires to monitor batteries. The objective is to introduce the way of conducting the evaluation of the state-of-the-art estimators for non-linear systems, to identify appropriate filter and model combination which were rated under accuracy, complexity and robustness, while the set of filters, their classification and verification are covered.

Chapter 5 discusses the Bayesian optimal filters for non-linear systems, and the number of methods which have been developed for these kind of problems, linearization methods e.g. extended Kalman filter, and other numerical approximation methods (Gaussian sum, iterative quadrature, deterministic sampling approximation). Also, artificial intelligence algorithms, such as neural networks and deep learning are discussed.

Although the importance of state and parameter estimation is discussed in the previous chapter, in *Chapter 6* by describing the multi-variable problems, the profound and complete evaluation for battery systems is provided. The comprehensive discussion involves the definition of the estimation framework and integrated for state- and parameter estimation using joint and dual estimation with respect to their underlying attributes. Also, new estimation techniques from the field of Machine Learning are introduced. All are benchmarked in regards to various battery equivalent circuit models as they are needed

for accurate voltage, power and aging prediction. Due to the batteries properties and non-linear behavior, the investigation concentrates on local and global non-linear methods for State-of-Charge and ECM parameters, while Machine Learning methods are used for Remaining Useful Life / State-of-Health estimation.

Chapter 7 deals with the SW testing and hardware-in-the-loop BMS validation on functional and non-functional level, designing a new battery system simulation environment including fault insertion.

The final *Chapter 8* discusses the challenges in safety-critical systems design, introducing a new way of ISO26262 complaint model-driven software development taking into account a new tool-chain for SW architecture and model development for BMS multi-core target CPUs. The introduced ecosystem fulfills the requirements during development phase with traceability, validation and verification, for which new model-based battery management tool-chain was proposed while computing power was optimized, before the work ends with a summary and outlook in *Chapter 9*

2.1 The Profound Impact of Electrification: From Computer-like Car to a Car-like Computer



a) TESLA Model S (SW v.6.0)



b) i-MiEV



c) VW e-Golf



d) Renault Zoe



e) Nissan Leaf

Figure 2.4: Conducted test drives of different xEV's for evaluating user experience and SW quality.



a) BMW i3



a) BMW i8



a) Fisker Karma



a) AUDI e-tron



f) Merceds B-Class

Figure 2.5: Conducted test drives of different xEV's for evaluating user experience and SW quality.

3 State-of-the-Art Battery Management Systems

3.1 Architectures and Subcomponents

The HVTBS and its design depends on the application and its requirements (e.g. aerospace vs. home storage systems) which vary in terms of power and energy demand, environmental condition (ambient temperature) and safety level. The following Chapter is summarized from e.g. [7][8] [9] [10]. This freedom of design is not standardized and hence, not limited in terms of mechanical, electrical and thermal implementation, each system has to be developed starting from almost white paper and just taking "lesson learned" into consideration. A main component of the HVTBS is the cell module, which consists of the individual cells integrated in macro cells (modules) e.g. below 60V. The interconnection between modules may vary between serial and/or parallel and is supervised by the BMS. The thermal systems' task is to keep the battery pack within its defined temperature range. The system can be active (e.g. water cooled) or passive by air. Some systems also have the opportunity to pre-conditioning the battery pack by heating up the system e.g. in harsh winter condition to prevent lithium plating. All of the components are tightly placed within a battery tray & lid (housing) to protect the components from the environment, which also includes all interfaces (signal/power/thermal) to other nodes in the vehicle.

3.1.1 High-Voltage BMS HW Components

The scope of a BMS typically consists of the following main components, BECM (Master) - Battery Electronic Control Module which is the dedicated hardware for communication and data collection and processing from CMUs, external communication, performance management with diagnostics and safety I/O functions. CMU (Slaves, or CSC)- Cell Monitoring Unit/Cell Supervising Controller, which is a dedicated hardware for individual cell voltage measurement, cell temperature measurement, balancing resistors for balancing for pack equalization and a battery disconnect unit (BDU) sometimes referred in the literature as switchbox. As mentioned earlier, there is no standards architecture on BMS and for each new vehicle platform a new BMS must be designed based on the pre-defined requirements. Therefore, it is necessary to understand the traction battery

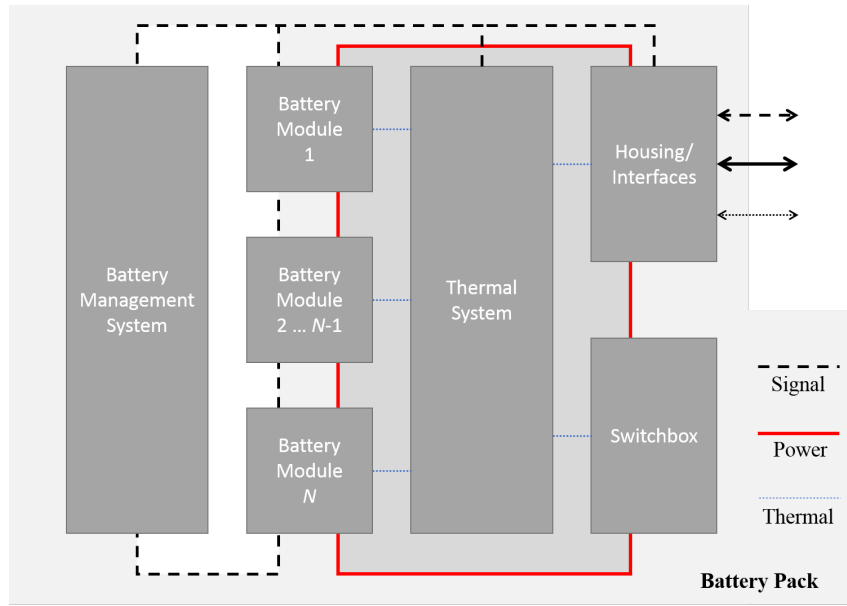


Figure 3.1: Schematic illustration of a battery pack and its main components. (modified from [3])

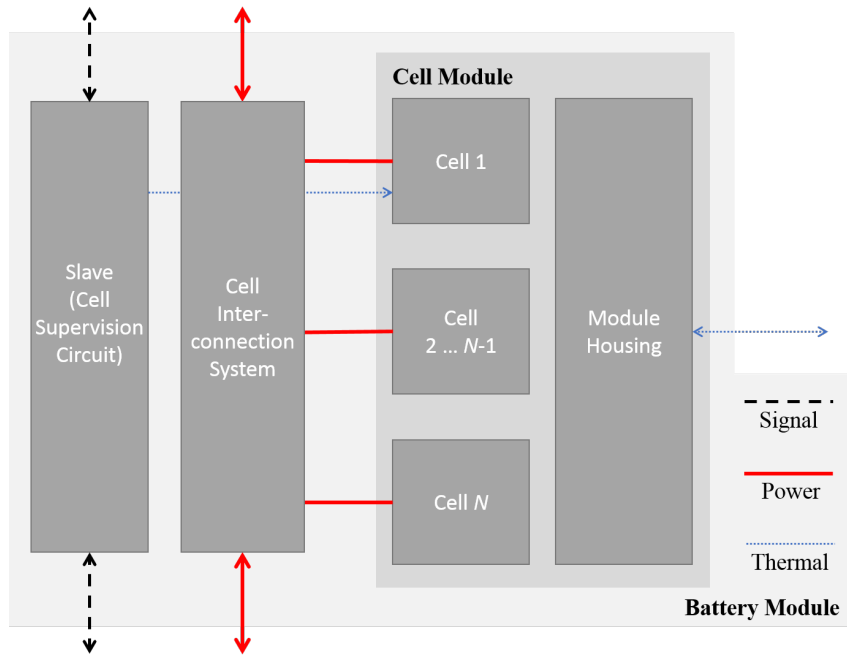


Figure 3.2: Schematic illustration of a battery module and its components. (modified from [3])

system components and its structure. As illustrated in Fig. 3.2, typical components are defined. Besides the battery modules, the system consists of the BMS, a thermal system for heating and cooling of the pack, a BDU, connectors for HV and LV. In case of a modular system, the BMS slaves are placed typically on top of the modules, to perform cell

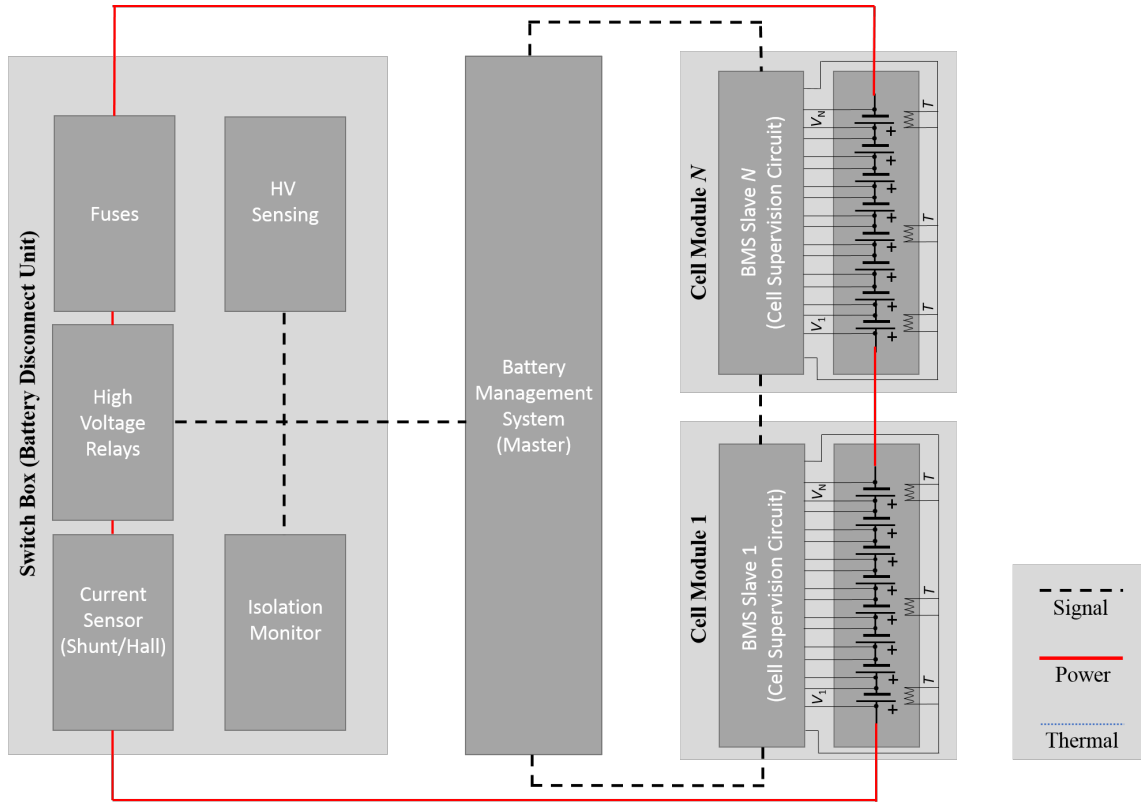


Figure 3.3: Schematic illustration of battery management system. (modified from [3])

monitoring. The slaves are commonly interconnected via an isoSPI or CAN communication line ending at the BMS master. The BDU consists of HW and SW functions, e.g. HV contactors, fuses with short circuit monitoring using over-current fault handling, fuse blown monitoring and fuse blown fault handling, in some cases the current measurement board (CMB) is integrated in BDU, for current measurement, current offset calibration, pack voltage measurement, DC link voltage measurement, voltage comparison, passive discharge check, current sensor rationality check, a pre-charge resistor which limits the inrush current and isolation resistance monitoring to measure if the vehicle is sufficiently isolated from high voltage components. In Fig. 3.3 a schematic illustration of a battery management system is illustrated. Additionally, some BMS are able to control the thermal management, e.g. pre-condition the battery at low temperatures or control thermal balancing with passive/active cooling/heating, valve control, pump control, coolant level monitoring, coolant temperature monitoring, coolant flow estimate or wake up to cool. Also the BMS is managing the charging sequence, including charge handshake/request, and charge control. In order to process the information of a vehicle crash, the BMS must supply additional CAN inputs and fault handling. With regard to the architecture, BMS are classified according to the installation mode and its modularity being the choice for the given application a compromise between safety, costs, reliability and integration. The BMS is realized using general purpose micro-controllers reading in measurements, as well as more or less distributed power and control PCBs with the task of controlling, calculating, estimating and predicting functions executed by the monitoring algorithms running

on the embedded system. Connecting the battery management system electronics to the battery cells must be accomplished with a very high degree of reliability. This connection between BMS slave and modules performs two functions, carrying the cell balancing current, as well as providing the voltage signal to the measuring circuit. The wire must be sized appropriately to carry the balancing current without excessive temperature rise. The wire selected must have a temperature rating exceeding the maximum service temperature plus the expected temperature rise during balancing. At failure mode of short-circuit currents flowing in the sense harness is an important consideration. Wire harnesses have to be constructed using conductors rated for the maximum expected potential difference between both the battery pack terminals when fully charged, and between the high voltage system and earth ground. In the following, different BMS HW architectures are discussed, as it is an important input into the SW methods design [7][8][9][10].

Centralized Architecture The simplest solution in terms of distributed system and installation has all his functions defined on a centralized single board reducing the need and costs of inter-module interfaces. On the other side, this single point setup requires wiring to the central location and limits the scalability of the system due to fixed number of installed cell monitoring ICs, decreasing the HW effort, while increasing the wiring within the battery system. At the same time, it has to handle and support the entire system voltage which means that all connectors must be designed and rated towards the pack voltage. Although some designs are driven by the fact that the potential difference of successive modules provide isolation barriers, in case of a system fault, the chance of running into higher voltage faults is given. Since balancing will take place on that PCB, the local heat generation and cooling has to be considered.

Modular Architecture Distributed systems offer the highest degree of modularity since all individual cell modules are equipped with a measurement unit board which is responsible for monitoring the voltage, carries balancing resistor, and digital communications which offers a simpler design and tends to offer higher reliability while all slaves are interconnected by e.g. serial communication network ending at the master. This topology is designed as Master-Slave architecture, where the master represents the single central management module, which is fully responsible for calculating unknown battery states and system controls. The slaves monitor and control a group of cells and are responsible for measuring cell temperatures, voltages and are in bidirectional communication to master, which initiates balancing. In modular system designs, the amount of installed battery modules and slave boards are identical, and when highly integrated the slaves are mounted on top of the battery module, where the slaves measure via sensing lines, or directly via the PCB. The communication between master and slaves can be based on different protocols, CAN, isoSPI, or Ethernet. In general, modular systems are heavier and more expensive due to extra circuits which consume more energy from the cells. Broadly speaking, the modular architecture combines the (dis-)advantages of centralized and distributed systems.

Semi-Modular Architecture The slaves of semi-modular systems can generally monitor more cells and are not integrated within the modules. In this context, the BMS can be scaled and expanded throughout the system development, when the final size or design layout of the battery pack is not frozen (cell supplier is changing cell chemistry/layout), and is therefore designed to be less depended on the sensing circuits. Also, some may prefer or require "off-the-shelf-BMS" products (e.g. Lithium Balance) which are developed for generic applications and fit for purpose.

Computing Architecture Here the measurement or control circuits are installed on multiple HW components/circuits throughout the different battery modules. In general, the master is responsible for most functions toward safe functionality of the battery pack, including contactor control, communication, pack temperature/voltage monitoring, requests cooling/heating requirements, and most important, calculates, estimates and supervises the SoC, SoH, SoF (SoX) functions. Besides the hardware, the key function of the BMS are carried out by an increasingly large amount of coding lines, consisting of complex algorithms, operating system with costly software architecture such as AUTOSAR that ensures safe operation of the battery. Depending on the necessary computing power for a BMS, the software architecture and computing power holds free selection of design [9]. For instance, in modular systems, the slaves may consists of a computing unit to monitor individual cell voltages, temperatures, and initiates the cell balancing.

3.1.2 High-Voltage BMS SW Components - Requirements and Functionality

The major BMS task is to ensure that individual cells of a battery system are charged and discharged according to the manufacturer's specifications, meaning, that cells have to be operated always inside a safe voltage, temperature and current window, limited by certain current, voltage and temperature. However, to guarantee for safe operation, an advanced BMS must not only monitor and control the described limits. The superset of individual functions are represented in the Table 3.1 and differs from BMS to BMS due to different requirements (e.g. automotive, aerospace, heavy duty machinery and more). The general requirements can be grouped into [4][7][8][9][10]:

1. Measuring, sensing and balancing
2. Protection
3. Analysis and high-voltage control
4. Thermal management

5. External communication and data logging
6. Power and energy management with diagnostics
7. Basic Software Module which defines basic software functionality close to the HW (System, Memory, Communication, Diagnostics, Complex Device Drivers...)

Due to the immense variety in battery cells (dimensions, sizes, chemistries) and battery pack vehicle integrations restrictions, an infinite possibility of combinations and architectural designs are possible. The BMS comprised of HW and SW must be designed in a way, to safely manage and control energy transfer as well as enables overall functionality of battery systems for automotive applications. The main task is safe operation within the specified working range and maintenance throughout the battery cells lifetime. Recent years have shown, that the trend towards modular and scalable architecture is favored. However, most systems are not independent from battery system or even module design and therefore don't offer generic interfaces which makes it impossible for easy mechanical and electrical integration. The same holds for SW part, changing chemistry or replacing cells later is not possible since adaptive SW is rarely used. As stated and categorized before, the functions for the BMS are to measure individual cell voltages, module temperatures and system current, additional safety functions such as the isolation measurement, contactor control for safe dis-/connecting the battery system from the propulsion system and thermal management must be considered. In regards to protection, electronics monitor all signals against all over-voltage and temperature, violation of charging and discharging currents or even short circuit detection. Communication with the overall vehicle control unit (VCU), as well as vehicle energy management (VEM) must be established, to communicate constantly the battery status, such as available and future energy and power limits. Also for warranty propose and lifetime estimation, critical states are logged and saved in the BMS EEPROM for future analysis. The task of the power and energy management is to calculate the battery system state, estimate available power and charge, and equalize the cells in the battery system. Besides the battery diagnostics (monitoring) where the SoH or the remaining useful life in cycles or capacity is estimated, additionally, an overall battery diagnostic strategy must be implemented. The basic aim of the diagnostic system is to detect and report faults in the battery system and all its components. The goal is to have a self-diagnostic system which is capable to identify and locate all faults (including external circuits connected) in the system throughout the entire lifetime by means of analyzing diagnostic trouble code (DTC).

Safety Management The key task of a BMS is safety management and protection against thermal and electrical abuse outside of specified operation window. Methods must be implemented to supervise these parameters and protect the battery pack against these conditions. These methods include:

- power and energy management (e.g. limiting continuous or peak power capability under certain operating conditions)

- thermal management to equalize temperature gradients between cells (i.e. controlling heating/cooling system)
- to disconnect the battery pack from the load
- to prevent dangerous states such as deep-discharged or overcharged
- control the switches located in the BDU with also includes other components such as current sensor, isolation measurement and interlock enable lines
- pre-charge and soft start control are required to allow the battery pack to be connected to capacitive load/transients

Measuring Accurate measuring is of high significance of digital BMSs. A sophisticated digital BMS has to be able to measure pack current (typically 0.5 – 1% up to 500A accuracy) and pack voltage (typically 0.1% accuracy), voltages of cells (3 – 5 mV) connected in series and temperature ($\pm 1^\circ C$ accuracy). For cell chemistries with a especially flat OCV characteristic curve a very high accuracy is required. Besides voltage and current, several temperatures within the battery pack best placed at hot and cold spots, have to be measured. Additionally, supplementary sensors, such as environment sensors like humidity sensors, GPS for location using analog and digital I/O [3]. Pack current can be measured using a Hall effect sensor or/and a current shunt. Hall effect sensors, unlike shunts, will be isolated, non-resistive and stable over time and temperature, but suffer from offset at zero current. Pack voltage will be measured or calculated from the sum of single cell voltages, whereas cell voltages can be measured either using one analog to digital converter per cell or a multiplexer. It should be noted that isolation of voltage signals from control signals is an issue in large packs and has to be addressed. In automotive application and the LIB chemistry certain accuracy and sampling rates are required, depending on the used cell chemistry, these requirements may be adjusted also for precise state estimation. At the same time, the measurement circuits PCBs must be robust against electromagnetic interference (EMI).

Balancing If cells in a pack are connected in series, in order to maximize battery capacity, BMS has to correct cell-to-cell unbalances through active (non-dissipative) or passive (dissipative) systems. Passive systems are usually made up of simple resistors, different power ratios are possible, while a great number of topologies have been proposed for active systems using some type of DC-DC converter with or without transformers. Furthermore different balancing algorithms shall be proposed, a smarter method is required. Heat dissipation in passive systems and cost, reliability, volume and standby power consumption in active systems are issues to be considered. The origins of the unbalance are interconnection of cells with different SoC levels when the pack is built and/or differences between single cells that show distinct self-discharge rates or capacity under certain operating conditions. Thus, balancing is a key function for large packs if the cells are not preselected for the pack

(choosing cells with similar capacity/internal resistance with a minimum standard deviation). Charging strategies (CC/CV) , final voltage, and thermal management have to be included in the system design. Balancing faults in regards of high self-discharge, voltage measurement deviation have to be considered.

Analysis Analysis of measured variables allows the BMS to estimate and predict the pack status and therefore to optimize its performance. More advanced algorithms are required to address the management challenges of battery technologies for each specific application, for a LIB (eg NMC, NCA) must include accurate SoC and remaining run-time indication for user convenience and extension of battery lifetime. Also, prediction of power capability is also a must for any BMS used in LIB EV applications due to high power charge and discharge pulses. For LFP chemistry, a voltage hysteresis model has to be included.

External Communications BMS systems incorporate communications between the battery and other embedded systems of the vehicle, such as the charger and fast charging interface to transmit information such as SoC, SoH or energy or power prediction, while at the same time the BMS received data as well. Communication interfaces also need to allow the user access to the battery for modifying the BMS control parameters, for diagnostics or testing. The BMS system devices requires communication with external equipment. This is a data link used for performance monitoring, for data logging, for diagnostics, or for setting system parameters and communications channel carrying system control signals. The choice of the communications protocol is determined to be CAN to the external devices. External communications include the transmission of a wide variety of data from and to the BMS through different channels.

Power and Energy Management (PEM) Common features are: charger control (also wireless), different power plans or operating modes (e.g. maximum runtime, maximum battery life...) and ability to be charged during operation, as well as safety hazards and impact in the grid operation and stability. Therefore it is necessary the highest accuracy of the input data used to compute the PEM algorithms, i.e. the state variables estimated and predicted in the analysis module (SoC, SoH, remaining useful life (RUL),...). Using PEM in these applications, the cost of the energy production, distribution, maintenance and operation can be reduced, increasing as well the reliability, safety and lifetime of the battery.

Power supply The battery management system slaves shall be powered directly from the cells in the battery stack, if not power could be completely provided by an external control voltage supply. Alternatively the high voltage bus, either on the battery side or the load side, could be used as a potential power supply. Combinations of the above implementations are also possible. The master is powered by the SLI battery 12V board net. Regardless of its source (internal or external), real-time measurement of this supply voltage is required. It is a must that on-board backup power for at least the microprocessor and supporting circuitry such that a loss of

power leads to a controlled shutdown with an orderly save of data to nonvolatile memory.

For a state-of-the-art BMS, the function attributes consider following requirement set:

Diagnostics System diagnostic implementation with diagnostics interface, diagnostics conditions, storage of diagnostics data (snapshot data), diagnostics calibration, notification concept, and others.

Battery Mode Manager with functions for power up/power down, restart after fault.

Battery monitoring, estimation & controls SoC estimation, including SOC limits with a required certain accuracy, cell SoC/voltage balancing, short-/long-term power prediction considering SoC/temperature dependence for dynamic voltage limit determination, health condition estimation for capacity/resistance determination.

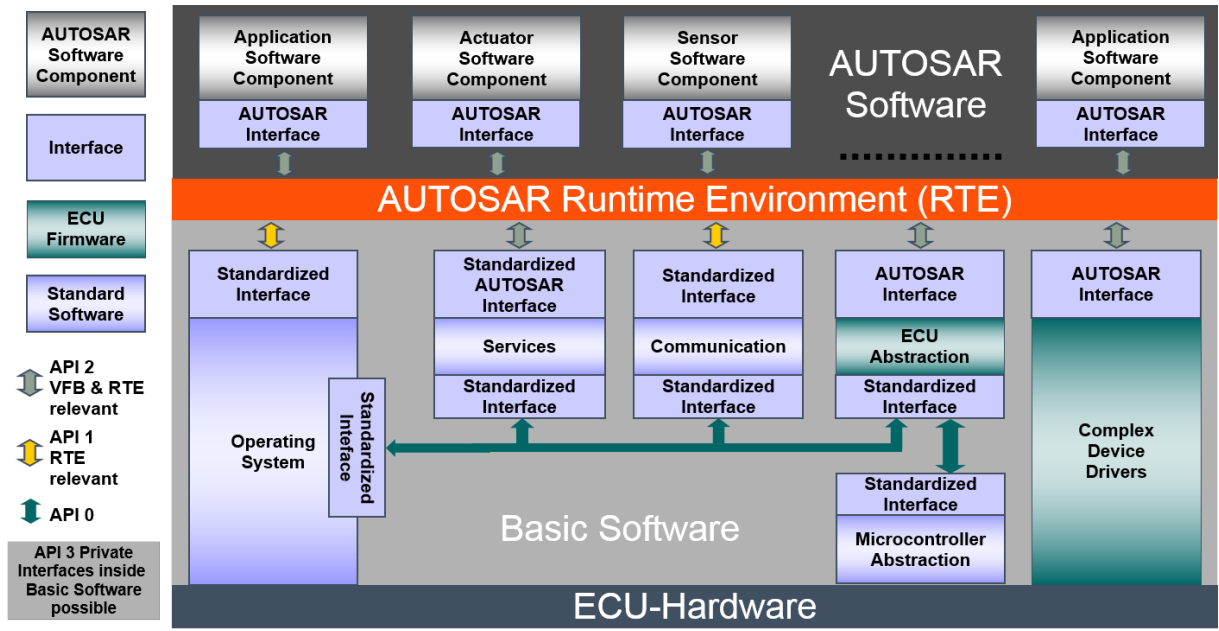
Safety In terms of safety, voltage and current monitoring is of highest priority, it must account for current measurement, including current offset calibration, single cell/link/pack voltage measurement with voltage comparison for plausibility checks, passive discharge and current sensor plausibility check. All the functional safety related SWC are based on the results of the hazard and risk analysis and functional and technical safety concepts.

Base Tech Additional SW modules include base technology with HW drivers, to make sure software download is possible, network management and calibration of parameters, memory read/write, reset and failure handling. Regarding the communication, the system must be robust and be able to handle the system in case of lost communication, as well as measurement signal synchronization, analog filtering or quality factor signals handling.

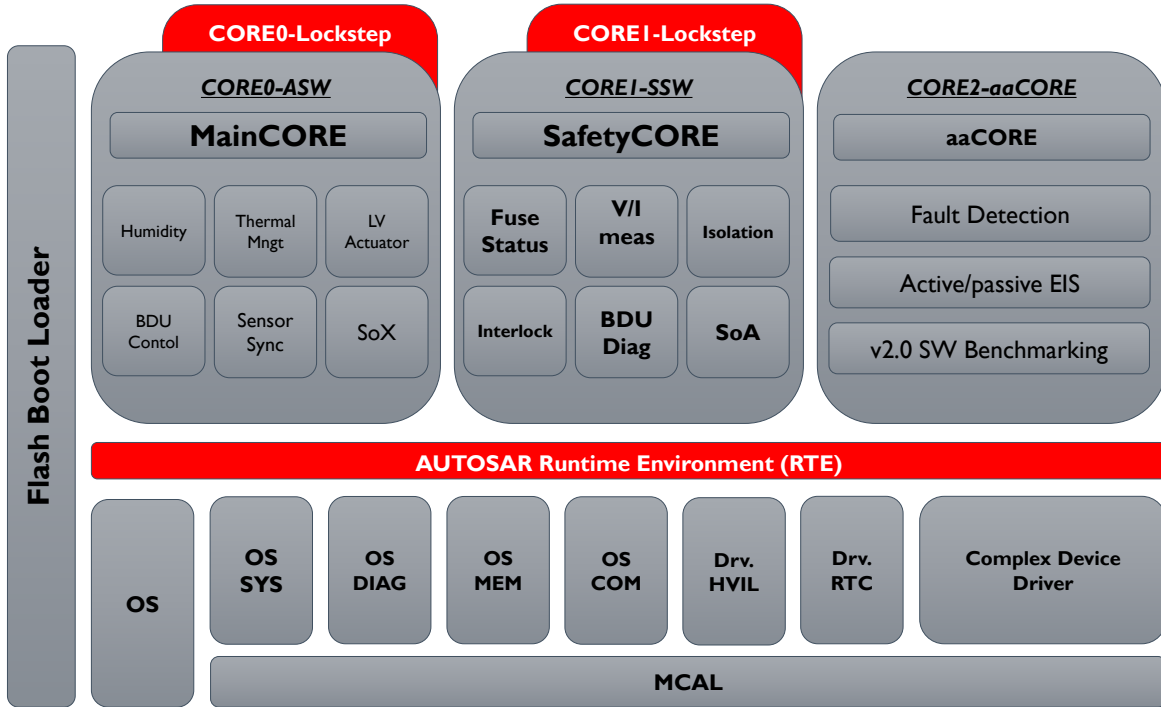
SW Quality Assurance During the SW development phase, SW quality assurance must be account for, including documentation, full testing over all layers (unit, software-in-the-loop (SIL), hardware-in-the-loop (HIL), vehicle...) Static code checks and formal reviews of design and code, functional unit tests, continuous integration and testing using e.g. Jenkins, automated functional tests on HIL test benches, regression testing, which must be reviewed amongst all stakeholders for final delivery.

Looking closer into the SW architecture as shown in Fig. 3.6, it can be seen that there are different level of features. The SW architecture is designed after the AUTOSAR standard, with the idea to standardize basic SW modules to manage the increasing complexity. The modular structure makes it possible to re-use legacy code and SW modules from previous projects, to improve efficiency and handling the shorter lead times, when ECUs are changed in the automotive industry. By decoupling application layer from basic layer with a clear interface, the reference architecture can be migrated to new E/E architectures. Again, different options are available depending on the chosen system architecture, e.g

centralized systems use single ECU approach containing all the functionality. As chosen for this work, a modular BECM-CMU architecture is chosen with distributed functionality, for which ICs on the CMUs are responsible for voltage/temperature measurement as well as balancing. Also, there are BMS available with CAN communication between CMU and BECM, in return the MCU can be used for data-preprocessing to add additional functionality to CMUs.



(a) AUTOSAR layered architecture provided by AUTOSAR.



(b) Developed SW architecture for advanced, safety-critical BMS.

Figure 3.6: Proposed and developed SW architecture following AUTOSAR standard.

In the following, only for this work relevant AUTOSAR working principles are mentioned, further information can be found in [12]. The fully-preemptive multi-tasking framework is designed to account for that the safety-critical function carried out in time. The ar-

chitecture consists of multiple layers, application layer (top level), runtime environment (RTE) (middle level) and basic software (BSW) (low-level). Within these layers, following are integrated like micro-controller abstraction layer, ECU abstraction layer (ECUAL) and services layer (SL). The functional SW is designed in terms of software components (SWC) parts of low-level code that executes an application not only on the highest level, the virtual function bus manages the interaction within all SWC and maps the SWC to the ECUs with their configuration of base tech SW. The application is the most top layer and divided into SWCs (e.g. SoC, SoF, SoH...). The SWC using ports to interface to others, also the data types are defined. The SWC are decoupled and independent from the used ECU hardware. The application layer communicates through the RTE with the BSW. MCAL is the lowest SW layer of BSW, including internal low-level drivers to access directly the MCU. ECUAL interfaces the external and MCAL drivers. SL represents the top level in BSW, offering OS functionality, vehicle network communication, diagnostic services or ECU state management. Additionally to the SWC, a flash bootloader is required. The developed platform is implemented on an 32-bit MCU Multicore/Lockstep AURIX from Infineon, based on three independent 32-bit TriCore CPUs. 2 out of the 3 cores feature an additional lockstep core enabling ISO26262 functional safe computational power. Within the functional safety SWC, the error manager supervises and triggers the handler in case the fault exceeds the allowable time of action. Also, the safety system will also run on the corresponding lockstep core of the Aurix, which enhances the overall safety monitoring. Also, functions are split into different core for increased redundancy and freedom from interference. In this work, the ASWC are located on Core0 with lockstep, the SSWC's are also located on a separate core Core1 with lockstep. Core2 is used as a benchmarking core for advanced algorithms, over-the-air new functions may be uploaded and compared to ASWC calculations, in case of improvement ASWC may be flashed. The overall SWC's are summarized in Table 3.1 [7][8][9][10].

Classification of proposed SW components for intelligent BMS functionality		
Application - ASW	Basic SW - BSW	Functional Safety - FSSW
Measurement/Monitoring • Cell voltage • Cell/module temperature • LV supply (KL15/30/31) • Balancing • TS coolant (leackage, T) • Contactor (diagnostics) • Pre-charge • Fuses • Interlock • Crash • Insulation • HV current • HV DC-link/system • Sensor synchronization • Humidity • Slave IDA • Address registration • UDS, update programmable	Support of - A/D measurements (V/I/T) - Improper node/node - State control - Network management - Power-up/down - Contactor control - Fault handling - Cell balancing - Diagnostics - Crash signal - Communication protocols - OBD standard - NVM access/mangmt. - Operating system - RTE with Task scheduler - CPU load monitoring - HAL driver	Monitoring for SC - Voltages - Cell/module temperature - Program flow - End-to-end comm. - TriCore safety features - Crash signal - Fuse status - BDU diagnostics - Interlock - OV/UV protection - SDU status
Controls • Thermal system (pump,fans) • Pre-charge • •	Support/handling - Calibration via XCP - RTC - Restart after fault - Fault event logging	Handling - Diagnostics - Safety errors
Battery State Estimation • Data pre-processing • Plausibility check • SoC • Battery impedance • Diffusion • Adaptive char.maps • Capacity • Available power • SoH • Available energy • Cell equalization • Data logging • Remaining useful life • Charging time • Fault detection & isolation • Active/passive EIS • On-line SW benchmarking		

Table 3.1: Summary of SWC for ASW, BSW, FSSW

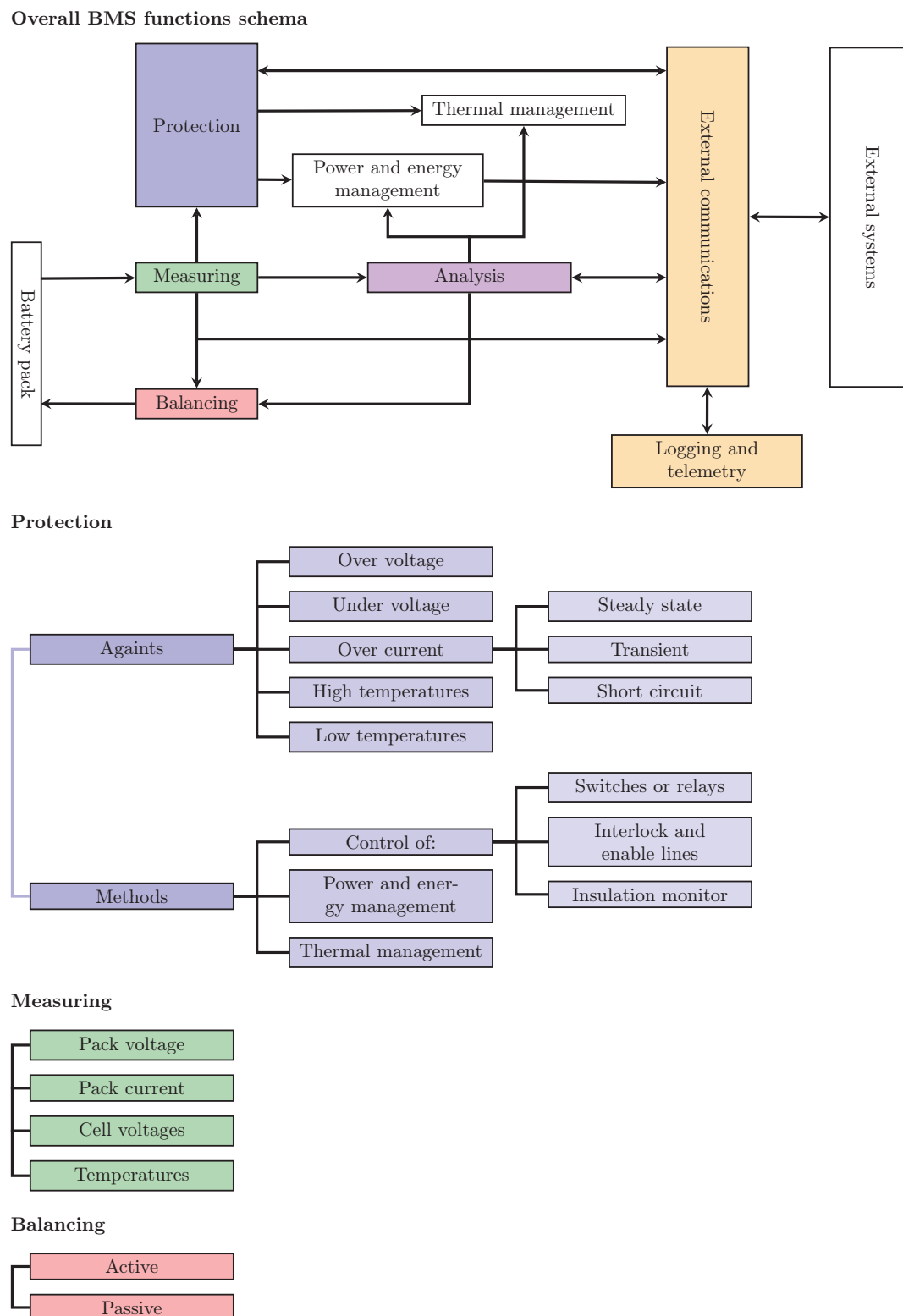
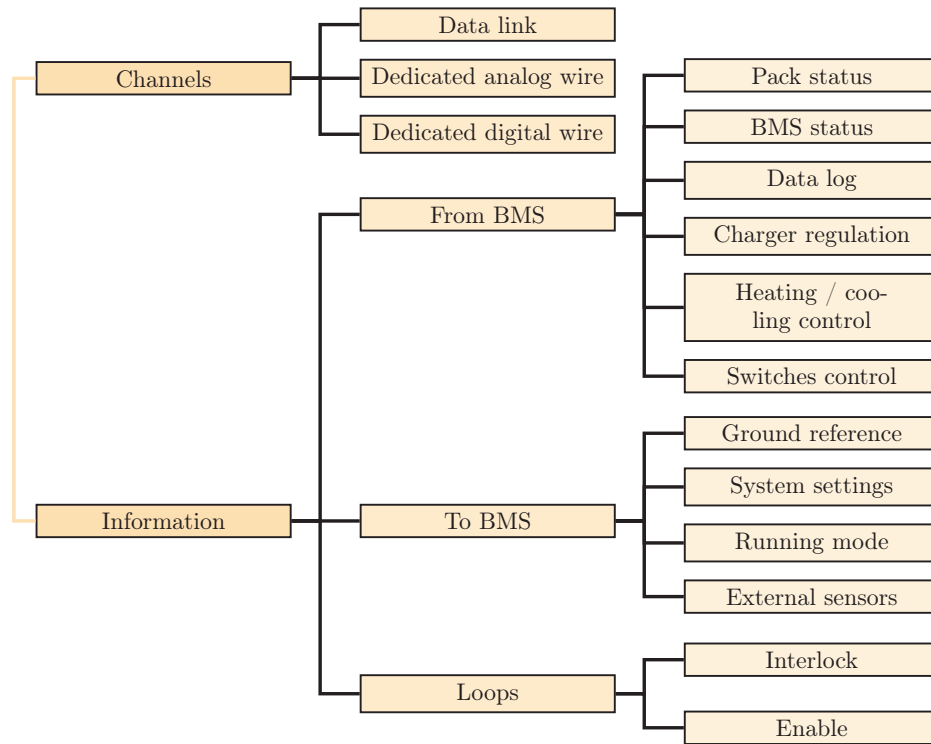


Figure 3.4: Schematic illustration of BMS functions (modified from [4][11]).

External communication



Analysis

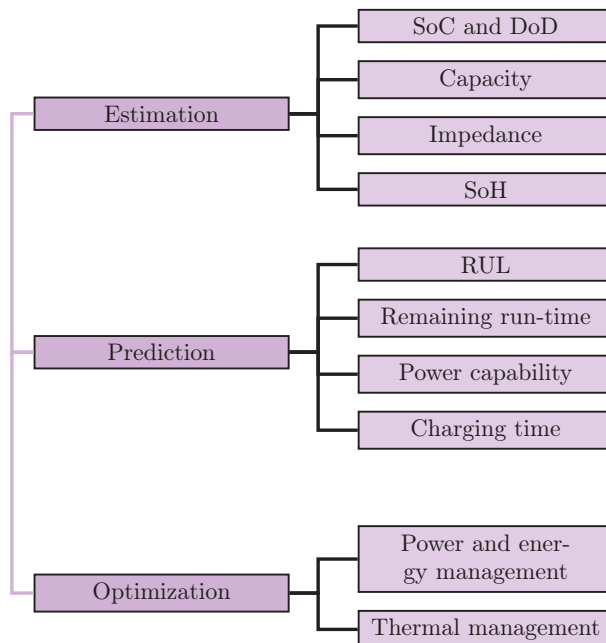


Figure 3.5: Schematic illustration of BMS functions.

3.2 Literature Review and Objectives

This section follows up the critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles in Waag et al. work [13], which has reviewed more than 350 sources until 2014 based on BMS requirements. This work extends this effort including 350 publications in this field from years 2014-2018, but instead of following up with a classical literature review, which includes the evaluation of the individual work. For that, it shall refer to [13][14][15], instead, the classification of methods derived in [13] is further extended. As it shows much research is put into battery diagnostics. *Therefore, this thesis is going to answer the question:*

Which method (filter & model) reaches a certain efficiency $\rho = X\%$ with a certain stability $\zeta = Y\%$ with the least implementation effort and lowest computational cost.

Therefore, it is important to understand the estimation problems to this task, which must consider the following [16]:

- (a) understand the battery dynamics in terms of aging;
- (b) hidden dynamic states must be observed while considering measurement noise;
- (c) for good filter parameterization, knowledge of process and measurement noise covariances are one requirement;
- (d) on-line identification to counterattack model inaccuracies caused by model reduction and change of system behavior.
- (e) recursive or batched based problem
- (f) robustness and convergence towards meaningful values
- (g) equality and inequality constraints handling
- (h) low complexity on derivatives or need for Jacobians or Hessians
- (i) does system- and sensor noise to be considered; if so, what is the underlying noise process (Gaussian / non-Gaussian)
- (j) stochastic, deterministic or optimization problem statement
- (k) robust numerical algorithms available

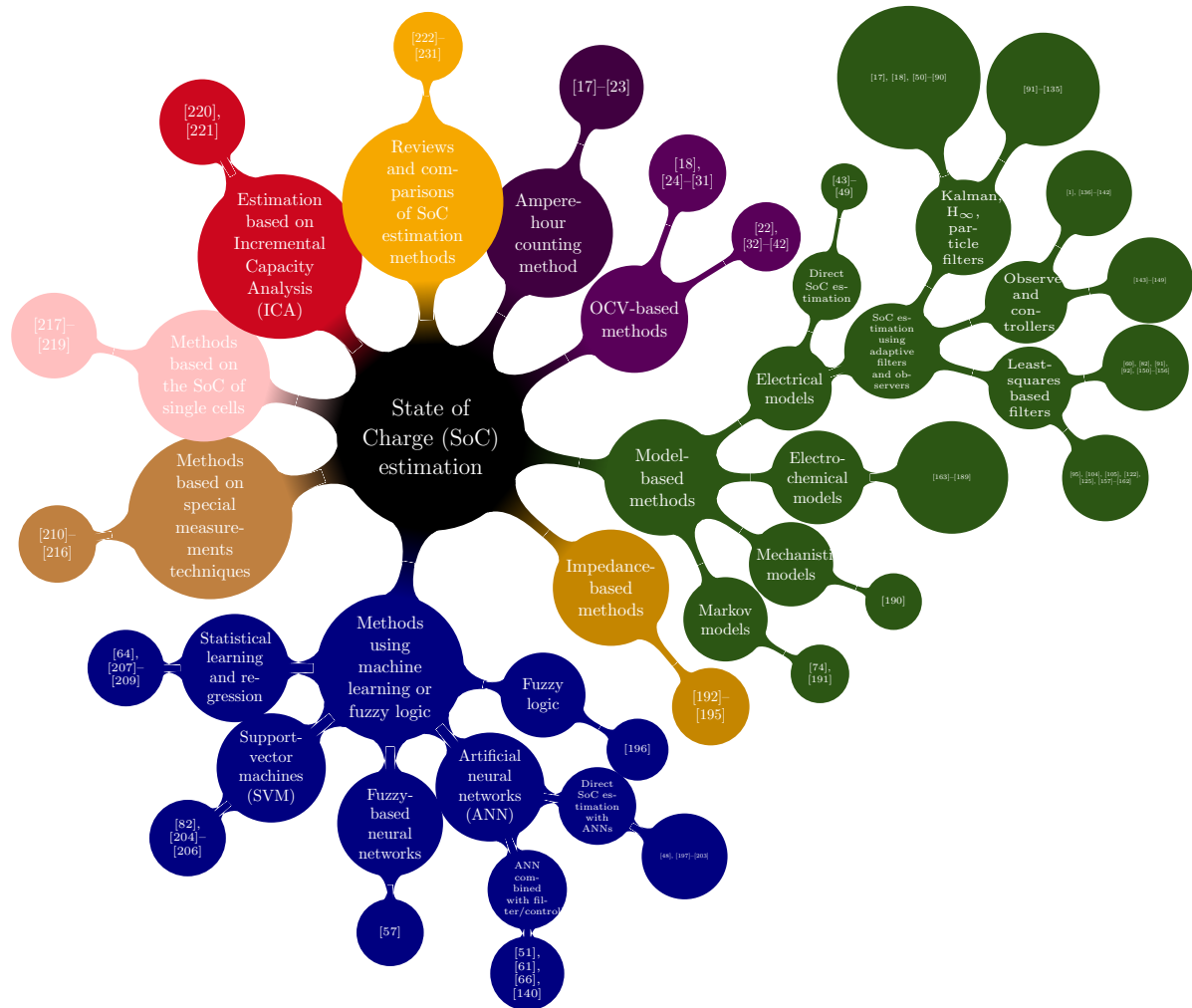
- (1) feasibility of implementation, complexity handling and robust to uncertainties

For the requirement-purpose of battery state- and parameter estimation in batteries, the focus in this work emphasis non-linear, stochastic estimators (deterministic estimators are referred to as observers) which must be adaptive to account for the battery performance change caused by degradation over lifetime.

But starting first, the focus now is to summarize the methods for the battery states and parameters to be monitored and group them into functions (SoC, impedance, capacity, SoH, power, RUL) and categorize the applied filter methods. The general trend uses simplified behavioral models of the battery characteristics (electrical -i voltage response to current); thermal -i heat generation / losses and temperature control) which can be used to estimate and control the battery attributes. Then, most often methods from automatic control theory and signal processing methods (Kalman filter) are applied. Something must be pointed out, as already stated, software in vehicles is becoming a strategic product, especially with BMS ASWC, as a lot of research is invested. But let us not forget that although these algorithms are the main driver, it could not work without the overall SW development and integration processes. Therefore, it is a mistake to just look at algorithms when discussing BMS SWC, it is more than that and therefore all aspects (methods and tools) are discussed in this thesis.

3.2.1 State of Charge Estimation

The State of Charge at all times is a proportional indication of available charge in relation to the state the cell is fully charged. Its unit is in %, where 100% indicates a fully charged battery, and 0% fully discharged.



3.2.2 Impedance Estimation

The battery impedance is defined by the manufacture and the design of the cell. However, this attribute changes during lifetime as the battery ages and so does its impedance. The information about present impedance is an important input to estimate energy losses during driving, SoC estimation using electrical models and for power prediction. The applied method must be very accurate and consider all dependencies (e.g. current).



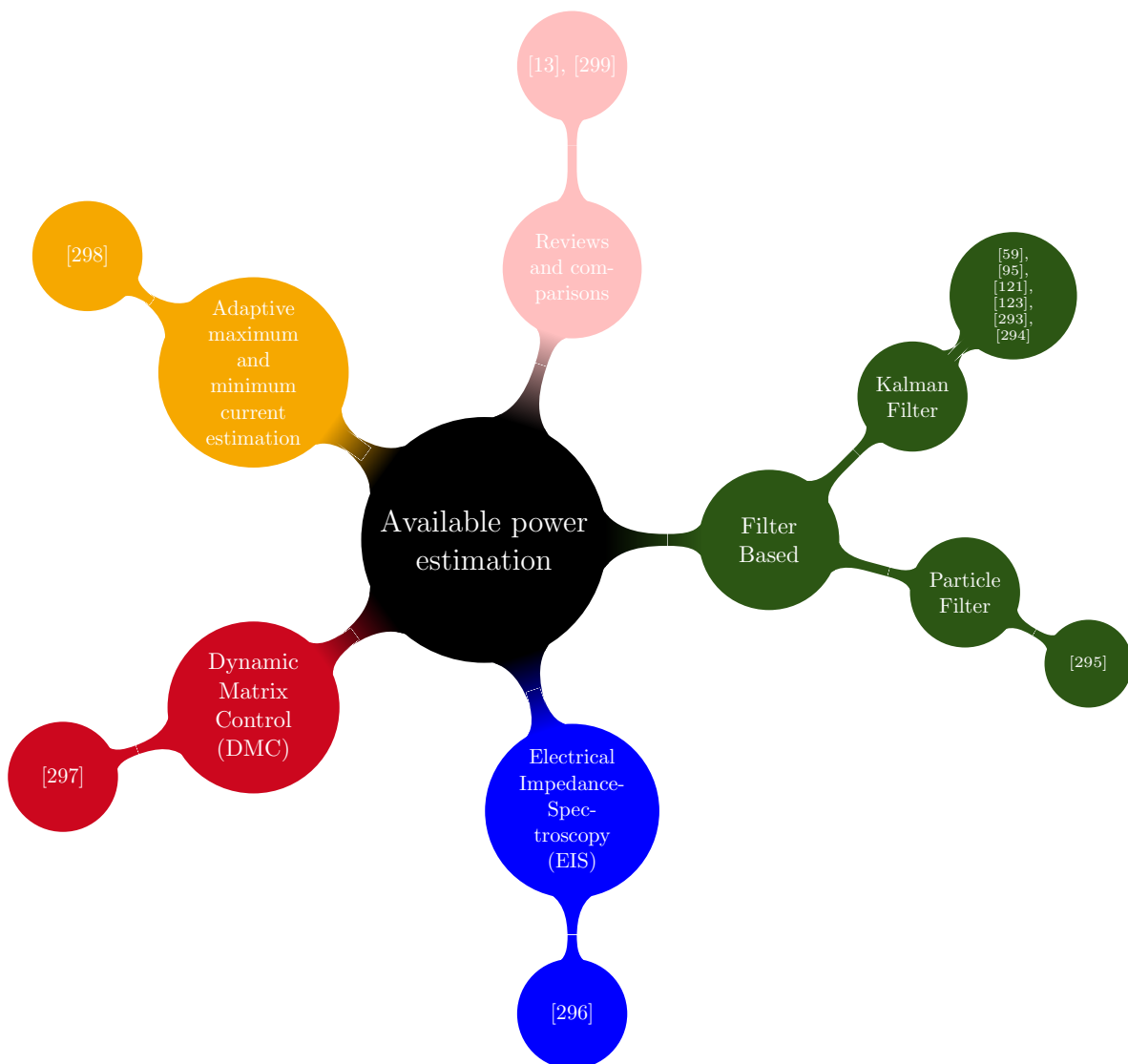
3.2.3 Capacity Estimation

The battery capacity indicates the batteries' stored and available energy when it is fully charged. The cell capacity for a new cell is defined by the manufacturer and its chemistry. However, over the lifetime the available capacity changes due to degradation mechanism leading e.g. loss of lithium. The methods for present capacity estimation must consider aging and needs to be adaptive.



3.2.4 Available Power Estimation

The task of a battery is not only deliver energy to the propulsion system, but it must also be assured that the battery is capable to deliver a certain power at all times. It is crucial to estimate the limiting factors, such as voltage limits, current limits, SoC or temperature in a predictive manner. The applied methods have to consider the dependencies to internal and external conditions, such as load history or temperature. Battery characteristic changes due to aging must be considered as well.



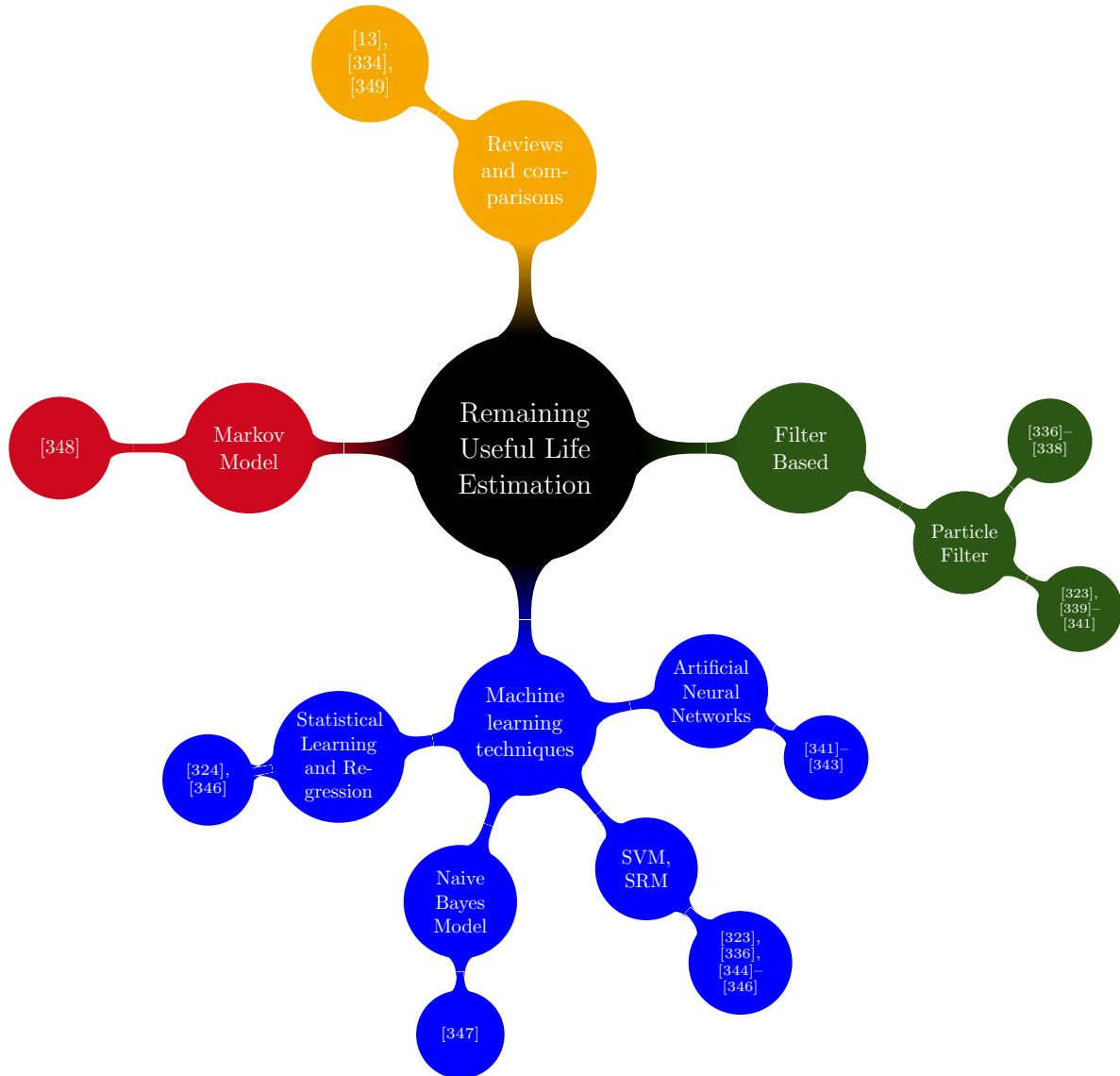
3.2.5 State of Health Estimation

The battery attributes, such as storing energy (SoHe), delivering certain power (SoHp) reduces over its lifetime due to battery degradation. SoH is a figure of merit to indicate this aging phenomena, a new battery has a SoH of 100%, while a battery is considered aged is 0% to a predefined minimum level.



3.2.6 Remaining Useful Life Estimation

Last but not least, RUL defines the remaining time or cycles of charging/discharging until the battery reaches a $\text{SoH} = 0\%$.



Which methods to investigate is clear when evaluating the publication list. It is apparent that methods from automatic control for state- & parameter estimation determines the direction of research and hence, was the motivation for this work.

3.3 Analysis of State-of-the-Art Battery Management Systems

Even though cell technology with its energy density, cycle life or power capabilities keeps improving, alongside with material improvement the control becomes more complex, the innovation has to take place also in software, hardware like embedded system and semi-conductors to provide safe, reliable and efficient management. The complete system is responsible, including controls, thermal system, sophisticated electronic devices which go hand in hand with intelligent software. Only then the system is able under all circumstances to precisely calculate the state of the battery, with its parameters to draw conclusions about its current and future condition, improve range and lifetime and guarantee that the battery does not reach critical levels. Therefore, to understand the needs for novel and next generation battery management systems, state-of-the-art commercially and non-commercially BMS are evaluated. In the following, 7 BMS have been analyzed in regards to key parameters and features, taking HIL tests and surveys into account, to determine the individual features. In general, for BMS components based on complex electronics, the requirements are that these systems are manufactured with high quality and standards to fulfill highest requirements. To understand the requirements for battery monitoring software, the state-of-the-art battery management hardware architecture currently in use in commercial available vehicles is studied. This gives a better understanding for future needs and requirements for intelligent software functions. Because of simplicity and non-disclosure agreements, acronyms are used instead.

3.3.1 BMSUT #1 - System A

The E/E system architecture of supplier A is defined as following. The master is the main controller based on Infineon AURIX family for SoX calculation and battery system control, it offers the interface to the vehicle, generates the high voltage interlock loop (HVIL) as well as relays drivers. The slave, is responsible for cell voltage and temperature measurement. Here, manufacturer uses different amount of temperature sensors per module, starting from 1 up to 6 sensors. The CMU is also equipped with balancing resistors responsible for cell balancing. The cell voltage measurement accuracy is here of importance over the lifetime of the system, and as discussed earlier, the modularity of the cell configuration, here it is based on an BMS IC chipset from Linear Technology allowing modularity and $\pm 3\text{V}$ measurement accuracy over lifetime. The passive balancing system with diagnostic capabilities is based on voltage or SOC and executed during normal system operation or during idle while master is periodically woke up by RTC. The communication with master is implemented using a differential daisy chain setup. To account for diagnostics and safety functions, the CMU offers over-temperature protection, open wire detection and reference voltage for plausibility of voltage measurements. One important component which is mostly not discussed in literature and considered as part of

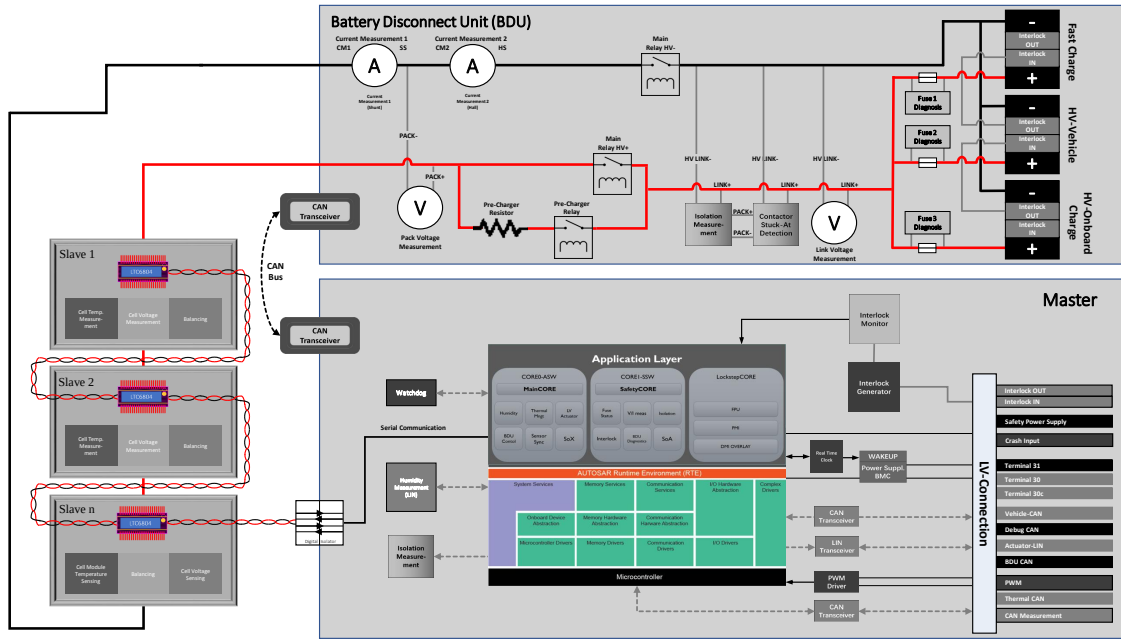


Figure 3.7: Supplier A: BMS E-architecture

the BMS is the battery disconnect unit. The BDU must provide safe and reliable connect and disconnect of high voltage battery pack to the overall electrical drive-train, including inverter, DC/DC converter etc and acts to ensure a safe operation during servicing, crash, or fault conditions. It has a pre-charge function for the DC-link, HV circuit protection and interlock function. For supplier A, the BDU is responsible to measure the battery pack current (ASIL-C) as well as the battery pack as well as link voltage measurement (ASIL-A). The BDU also consists of the integrated internal and external isolation monitoring unit while it is connected to the master via CAN. Regarding diagnostics and safety functions, it offers main fuse and relays diagnostic.

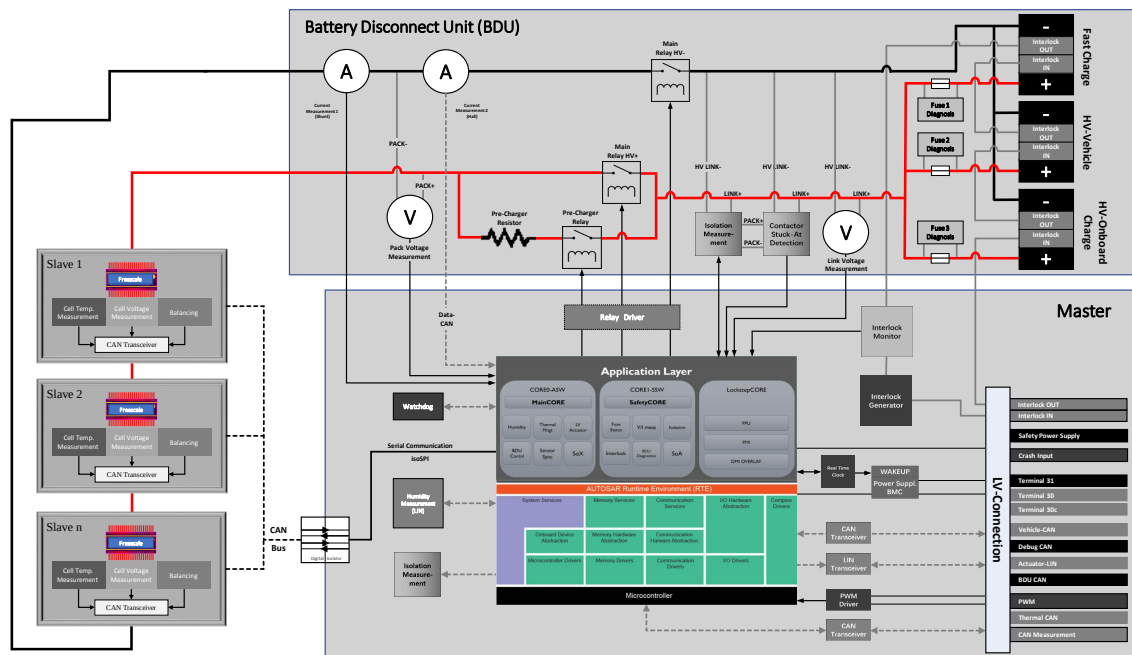


Figure 3.9: System C: BMS E-architecture

3.3.3 BMSUT #3 - System C

The E/E system architecture for Supplier C is illustrated in Fig. 3.9. Similar to Supplier B, there are separate private CAN interfaces to the vehicle and BDU while Ethernet is also available. In comparison to the previous supplier, here, the master calculates, monitors and requests the cooling demand for the battery pack, and it detects any coolant leak inside the battery box.

The slaves are equipped with a 32-bit/48 MHz CPU and CAN module from Freescale. The software architecture consists of third party SW which is controlled by OSEK OS. The system takes care of balancing, diagnostics, and other safety features and is unique within all tested systems. In comparison to other suppliers, the HV current paths are PCB assembled, where main contactors, pre-charge and resistors, shunt and bus bars are directly connected to PCB. This leads to a reduced effort in cable harness for control as well as voltage and temperature diagnostics with no spacial limitations including cooling concept.

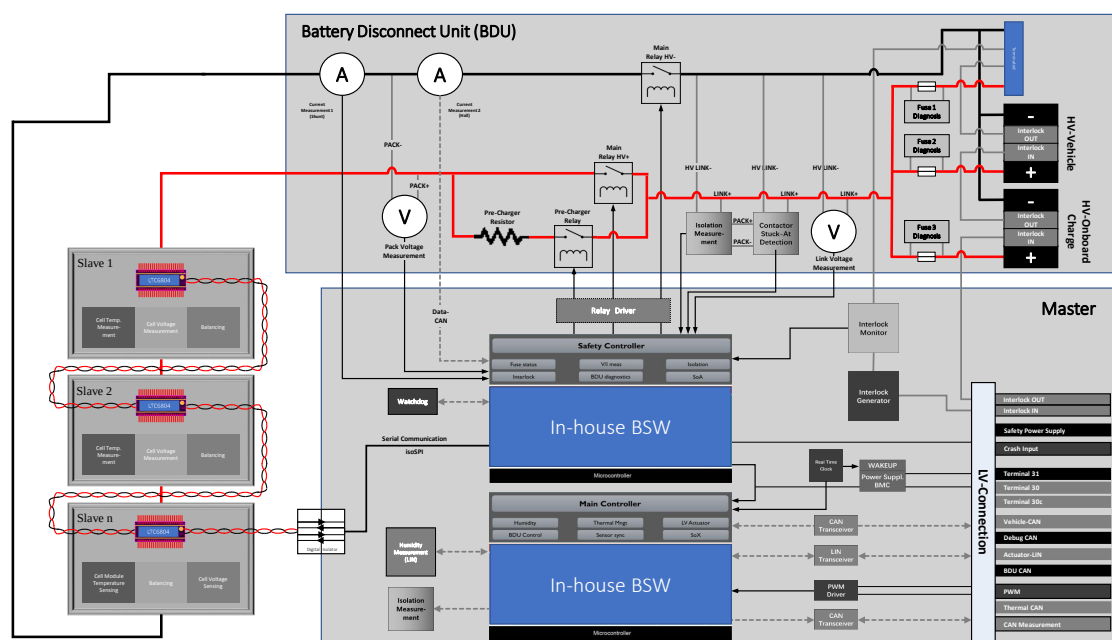


Figure 3.10: System D: System architecture

3.3.4 BMSUT #4 - System D

The modular E/E architecture of supplier D is illustrated in Fig.3.10. It offers a flexible and easy configurable module configuration based on 8, 12, 16 or 18 cells per module allowing cost efficient system architecture using a isoSPI daisy chain and the solution integrates the BMS and BDU into one component, it has integrated contactor drivers and isolation monitoring directly on the master unit. Additionally, system D measures and manages module temperatures up to 5 temperatures per module and differs from other systems, also the fact that DC charge interface is integrated on master, this means BDU includes BMS board and Power Board, including DC fast charge functionality resulting in no need for a voltage sense harness. This modular and integrated system reduces complexity, production costs, and wiring effort. The safety concept includes a separate safety controller which allows efficient safety SW design and is one of two systems in the test which uses 2 micro-controllers, while others continue to focus on multi-core systems. Also, this supplier uses an exclusive OSEK distribution but does not follow the AUTOSAR standard, rather building a lean in-house SW.

Regarding the slaves, most of the systems allow up to 100mA of balancing current, sytem D goes up to 150mA. It is based on a Linear Technology BMS IC current status allowing cell voltages reading with typical < 2 mV error, while the system is capable of pack current/cell voltage synchronization within 1 ms.

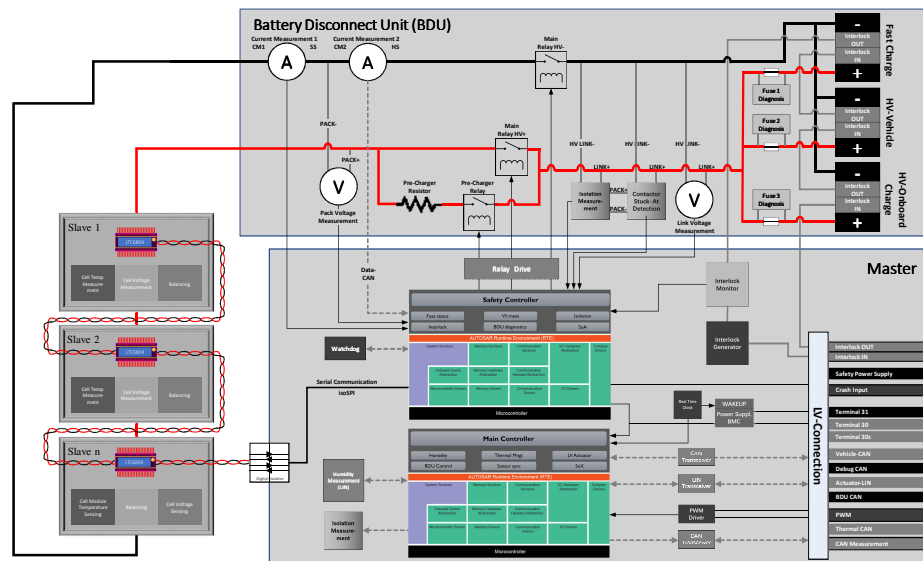


Figure 3.11: System E: System architecture

3.3.5 BMSUT #5 - System E

The BDU is connected to a high voltage battery as an Electronic Control Unit. The Slave is responsible for sensing cell voltages while the master establishes the communication to the primary vehicle management ECU via CAN bus. BDU and master are concatenated to estimate and control SoC, SoF, and SoH of the complete battery pack, the contactors and the precharge state of the intermediate circuit as well as to detect an overload.

The master is also equipped with two controllers, similar to system D, a main controller taking care of SoX calculations, while the safety controller takes care of safety functions. This architecture prevents a single power supply transient from creating a failure at which all contactors are activated.

The BDU consists of a precision shunt resistor, which is temperature monitored to guarantee safe operating area, and a 16bit data acquisition system to measure the data. The integrated micro-controller supplies all required controls to handle and manage the data.

3.3.6 BMSUT #6 - System F

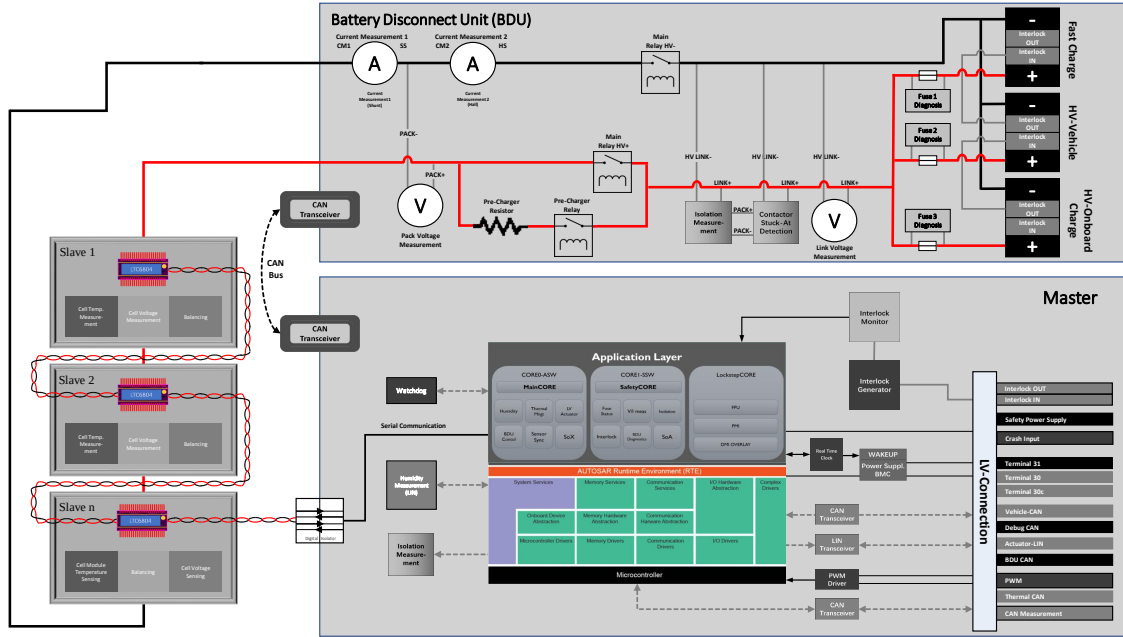


Figure 3.12: System F: Tested EE architectures

System F also builds on an optimized de-centralized architecture, instead of inter-slaves CAN communication, thus a differential communication protocol is applied. The aim of most BMS manufactures is the reduction of electronic components, by ASIC design, removing of micro-controller, removing of CAN communication leading to software free MCU. In comparison to other systems, system F is based on Freescale BMS IC up to 14 cells and 6 temperature sensors per module possible, high passive balancing currents (200mA) and up to 1000V total system voltage. Power supply is generated by ASIC which allows to neglect costly DC/DC architecture.

3.3.7 BMSUT #7 - System G

From the "off the shelf-solution" system G (LithiumBalance Denmark) is claimed to be flexible, robust and cost efficient for many applications, including automotive, industrial and storage stationary battery packs ranging from 12VDC up to 1000VDC. The system has been available for long time and has been validated for extremely harsh environment. This BMS comes with a PC management tool, which allows bit-level setup and configuration of BMS, such as cell-interconnection, communication, or balancing threshold. It is available as integrated PCB boards or packed and IP6x sealed and protected housed product. The system can manage up to 256 battery cells while one module is limited to 16 cells. Since its off the shelf, it may be used for any cell chemistry.

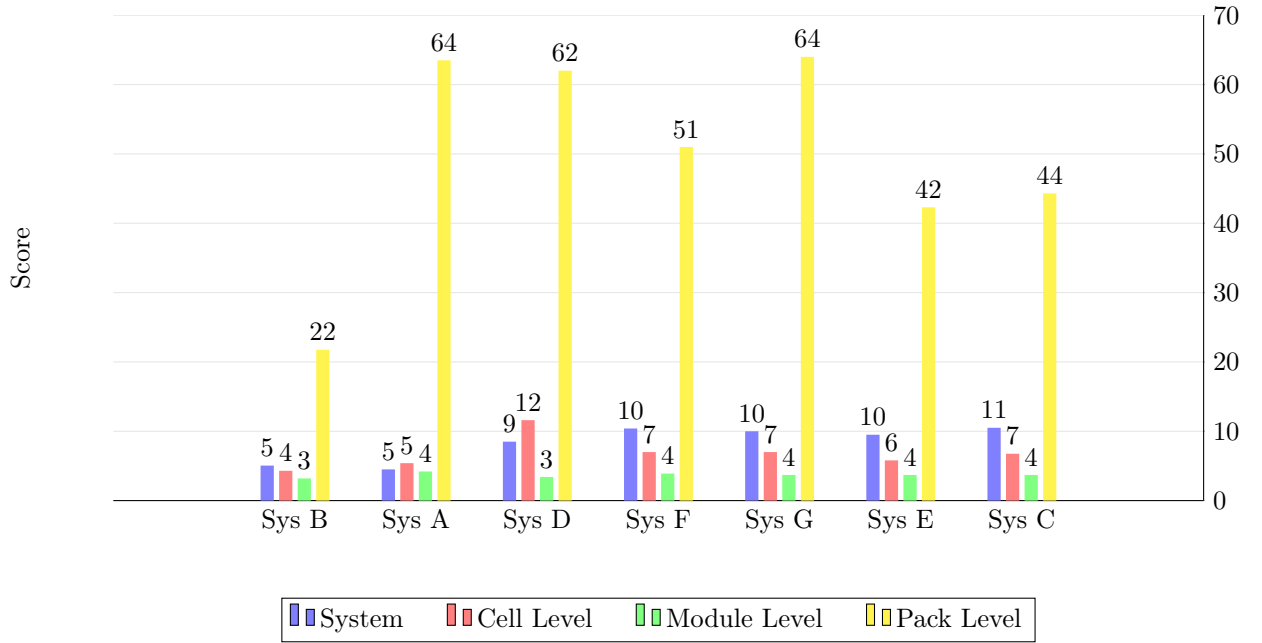


Figure 3.14: BMS HW evaluation diagram

and topologies. However, this will influence the SW design process.

3.3.9 Benchmark of Evaluated BMS SW Functionality

This chapter investigates the state-of-the-art battery management systems, which includes besides the electronics most importantly the software components. Since the software is the key function, this thesis devotes its attention mainly on the mathematical calculations for model-based software to estimate based on noisy sensor data and process noise the operational state of the battery pack and makes predictions with best possible certainty about short (1-5s) and long term (20-30s) performance capability for lithium-ion based battery packs. A survey of relevant supplier unveils different basic methods used to justify the key concepts of a battery management system. The weakness of that simple methods leads the work to focus on more complex but also more robust algorithms. Their implementation will require higher computational costs, but at the same time multi core systems and parallel computing will enhance the opportunity. The software and functions implemented by the supplier are targeted on the corresponding hardware platform. The general requirements focus on performance capabilities and management which combines a set of functions to fulfill SoX (SoC, SoF_P, SoF_E, SoH_P, SoH_E). An important need in BMS is the estimation of available energy and power in the immediate future. The requirement to be able to estimate the available energy is the knowledge of SoC for all cells and the total capacities. The same requirements on SoC has to be met by power prediction, but instead of capacities the individual cell resistances must be known. But,

these values can only be measured indirectly through estimation. During the survey of state-of-the-art BMS the goal was to identify the level of complexity from the applied methods. Generally, two different groups could be identified, basic and weak functions in comparison to current publications about battery state and parameter estimation. Of course, the lack of software engineers in battery technology and principles restrict complex or expert methods for SoX, therefore, simpler ways are implemented with higher errors. As illustrated in the SW evaluation diagram 3.15 Systems A-F lack in advanced battery monitoring SW, such as power prediction or SoH. On the other hand, SoC estimation with the more robust methods (e.g. Coulomb counting, OCV) is industry standard showing similar scores. The methods implemented in this work, the model-based battery monitoring improves commonly used methods as shown in the scores Sys CFL. Often, battery packs are over-dimensioned on purpose to overcome for uncertainty in SoX calculations. The basic battery controls methods try to overcome these disadvantages, sometimes using supporting algorithms for recalibration. A summary of existing basic and advanced battery state algorithms can be found in [13] and shall serve as basis for reviewing software components of state-of-the-art BMS.

3.3.10 Summary

It must be pointed out, that when developing BMS SW, it is important to consider the BMS HW too. For example, depending on the used HW architecture, the SW development has to be adopted, e.g. BSW (basic software and diagnostics). At the same time, the general experience in functional safety (ISO26262), are processes for Automotive-SPICE followed, general SW development tasks (tools, SW-architecture, agile mindset, coding guidelines, model-based design, diagnostics, multicore...) and lastly testing strategy on all levels from unit test up to target testing. The challenge in BMS SW is beyond algorithm development and a interdisciplinary problem. It can be seen, that the quality of state-of-the-art BMS lacks in their SW quality, it was evident that no complex algorithms (e.g. observer, estimators) are fully integrated. Firstly, the applied monitoring algorithms are of basic nature, model-based estimation have not yet become a standard, keeping the implementation rather basic. It was also shown that supplier G uses open-source OS which has not been identified during the HAZOP, which is vital for safety critical applications. In general, experience with safety-critical development is still in their infancy, AUTOSAR is not standard, and automation is limited. Figure 3.15 summarizes the overall supplier score which was partly extracted during HIL tests and partly through data sheets. The details of the evaluation is attached to the Appendix A.

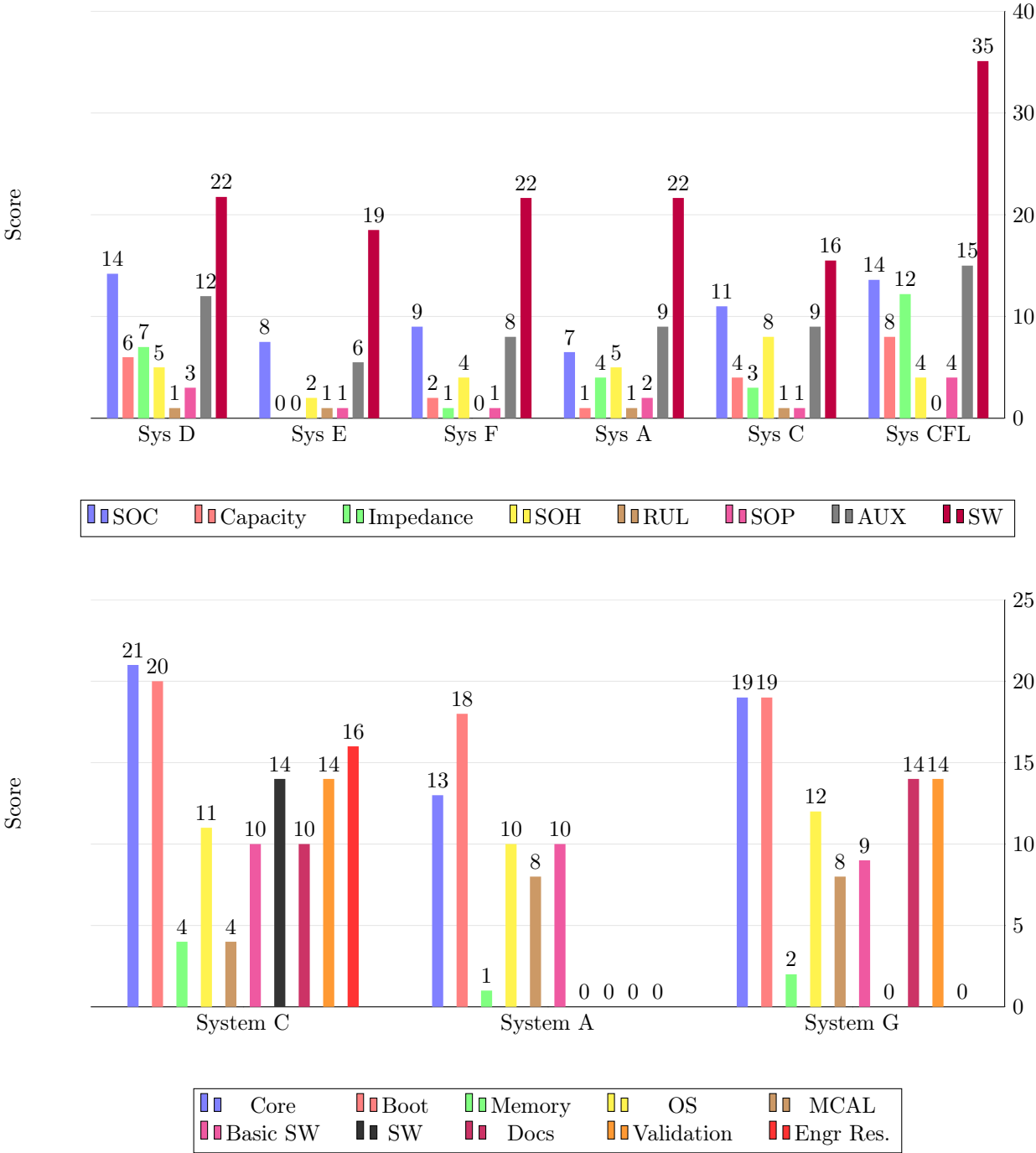


Figure 3.15: BMS SW evaluation diagram

4 Principles in Estimation of Signals and Systems

An optimal estimator is a computational algorithm that processes measurement to deduce a minimum error estimate of the state of a system by utilizing: knowledge of system and measurement dynamics, assumed statistics of system noises and measurement errors, and initial condition information.

– Arthur Gelb

Although linear estimation theory or non-Gaussian Monte-Carlo methods are often studied in the literature, they are widely scattered through journals with different focus, scientific areas and applications and has not summarized in a single work, especially for applications in battery systems. Therefore, this Chapter summarizes different sources such as [350][351][352][353][354][355][356][357][358][359][360]. The parameter problem deals with the question of a given set of measured data $y[0], y[1], \dots, y[k-1]$ which depends on unknown parameters θ , the challenge is to find an estimator or some kind of function such at [361]

$$\hat{\theta} = f(y[0], y[1], \dots, y[k-1]) \quad (4.1)$$

where f is some function which is used to estimate θ . The parameter estimation problem solves the question in finding a probability density function (PDF) of data as a function of θ .

4.1 Basic Concepts of Estimation and Problem Statement

Estimation delivers approximate calculation or judgment based on available evidence but tries to guess it in a educated way based on incomplete information. This process consists

of unknown and uncertain conditions and leads to non-trivial problems. The basic nature of the estimation problem is that its computation is a backdrop of uncertainty, therefore the system has to make reliable guesses, like for classification, prediction or detection. First, the necessary conditions to realize reliable estimation techniques, a mathematical description of the object is required to capture the generalized behavior in terms of a model. To which abstracted level the model describes the physical properties of the object depends on the purpose (need of the developer), the importance of process and the trade-off between complexity and available computational power. It is for these reasons, that

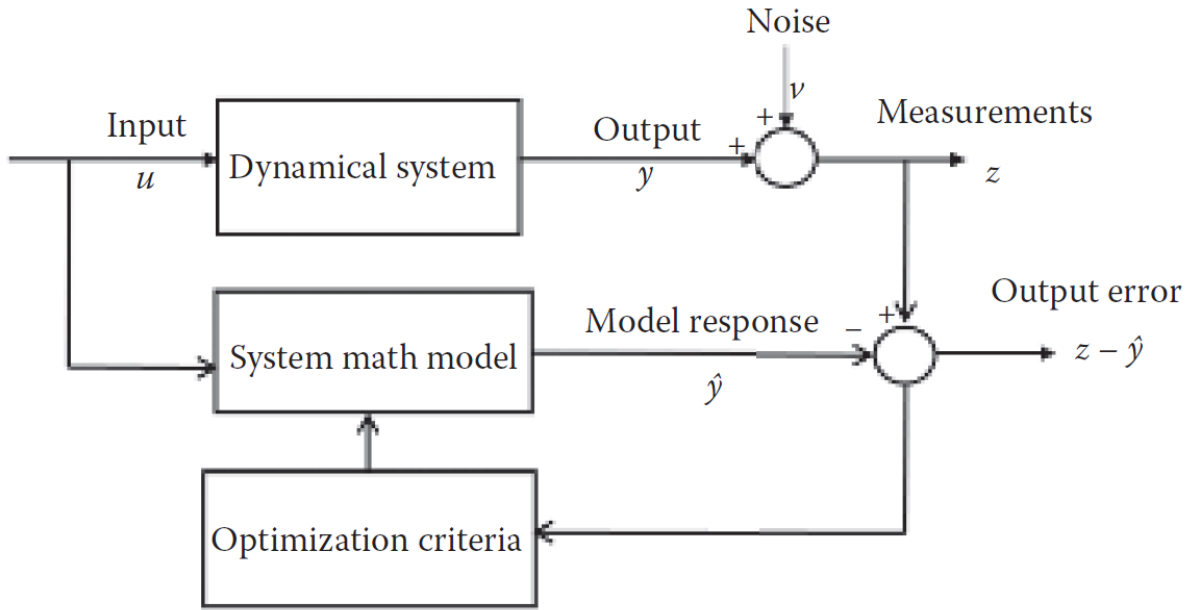


Figure 4.1: Block diagram of adaptive filtering [362].

calculations are in strict sense an approximation, because they use a certain approximate description of the actual physical principles which proceeds and has its origin in stochastic. Certain phenomena are neither describable, measurable or not necessary to describe in such detail. For certain application estimation deals with unmeasurable signals (e.g. to expensive to use measurement sensors, to complex) or if measurable they are superimposed with measurement or system noise with unknown properties. Additionally, dynamic and complex systems are considered and if discussed in estimation, signals and systems are described as models. Estimation can be grouped into two categories:

Off-line estimation (batch learning) collects data while the system is functioning, but the time when the estimation takes place is unrelated to the data collection, meaning the samples are stored and later applied for model estimation.

On-line/real-time estimation the data sets are samples and received through the channel during processing one after another as the model is simultaneously estimated

and is especially important for high level feedback in supervisory control applications making systems adaptable, autonomous, intelligent and fault-tolerant. These kind of systems require continuous monitoring, acquiring signals and check its plausibility, classifies it to make appropriate decisions regarding its control strategies.

On-line estimation requires on-line system monitoring and a method which supervises the control system (guidance). The on-line estimator acts like a sensor for the supervisory controller which provides set-points to the automatic controller with learning abilities (adaptation). It must adapt to a system whose parameters vary or are uncertain. Therefore, adaptive systems interact with their environment, its feedback loops adjusts the behavior to deliver a certain level of performance. The estimators are executed on digital computers extracting states or parameter values form a N -point discrete-time data set such as $y = \{y[0], y[1], \dots, y[k - 1]\}$.

Since batteries are non-linear and time varying systems, it would require that their behavior is described through non-linear dynamic mathematical models and introduces the necessity of non-linear filtering, however, a methodical evaluation of non-linear filtering techniques and their practical implementation improved only hesitantly. That is based on the fact that non-linear filter creates their own challenges:

- Theoretically, the optimal solution to a non-linear inference question is primarily the task of numerically solving in an recursive setup based on measurements the probability density function of the system state vector. The hereby required computing power for this problem statement is significant.
- To overcome the computational burden of calculation the Bayes-law, instead, rather non-linear filters are applied which use the method of moments computation, meaning that the probability density function is approximated by the means of its expected value and covariance matrix. The same is done regarding the non-linearities, they are approximated by some kind of linearizion technique around the operating point.
- The implementation of non-linear filters comes with more complexity in comparison to linear filters, as it requires a deeper understanding of the mathematical and controls theoretical background.

Estimation Problem Statement

In general, in many domains it is desired to extract desired signals and reject unwanted noise signals, such as in state estimation in linear or non-linear stochastically disturbed controlled system. This is generally classified as the estimation problems. The idea behind estimation in battery management systems is, that observable measurements from the

battery are provided, such as voltage, temperature or current, while the battery system is represented as a discrete dynamic process. Based on that data, the hidden states and unobservable battery dynamic parameters shall be estimated directly on-line. In this application, the state of the battery system refers to a set of dynamic variables, e.g. over-voltages, SoC, or SoH, which characterize the system completely. In continuous dynamic processes, the state evolution depends during the time (e.g. battery lifetime) only on the previous state and its transition, and is described by differential equations. Since we implement these estimators on embedded target systems, the differential equations are replaced by mapping the state information at measurement t_k to prior state e.g. t_{k-1} . Hereby, the measurements are superimposed by surrounding noise and make it random and uncertain. At the same time, the dynamic system is exposed to process noise with unknown origin. That is why the uncertainties are modeled via process noise. Based on observations and measurements, the models, which are representing fundamental system behavior, and in which the estimation problem is derived and its hypotheses about its environment, is falsified or verified, on the basis of its newest experiences which may lead to an adaption of the model properties. Depending on the certain quality criteria this adaptation is carried out as long as pre-defined compromise is not satisfied. A model is in principle a representation of e.g. a real phenomena. It approximates the fundamentals behind the real phenomena to a certain extent but especially those which are relevant to the system design, for which the state of a system consists of states and parameter. The state is given by a set of real numbers (a vector) in which each entry is expressing a point in the state space. Combining the state with the system input signal, it is possible to describe completely the systems behavior. The model parameters on the other hand describe the static, dynamic and/or (non)linear system behavior. In terms of a lithium ion battery, the behavior follows a dynamic, non-linear structure. This results in the fact, that the knowledge of the system model, its parameters and states with its excitation is leading to the property of predicting the future system's dynamic behavior. We are using here the term "estimation problem statement", since states and parameters of a real system are superimposed by interfering random signals as mentioned earlier. The dynamic process will be defined as a Markov process, meaning that the knowledge of the next state depends on the previous state. Since we deal with discrete stochastic processes, it is defined as [363]

$$x_k = f_{k-1}(x_{k-1}, u_k, \gamma_{k-1}) = f_{k-1}(x_{k-1}) + u_k + \gamma_{k-1} \quad (4.2)$$

where the state vector is described by x_k , f_{x-1} represents the state transition function which transforms the state x from time t_{k-1} to time t_k while u_k defines the control input sequence which excites the system dynamics. The noise term γ_k is not necessarily fixed to Gaussian distribution and accounts for the processes which are not included in the system model. The difference between the parameter estimation and system identification is, that for parameter estimation an a priori knowledge of the system model is necessary, while at system identification even the system model is unknown and derived together with its parameters. In most cases, however, it is important to estimate the internal states of the system, e.g. in batteries. Hereby the question arises, how can it be assured that based on a noisy current measurement, the information of an accurate SoX can be

provided and how about the confidence regarding the reliability in that estimation? Here, the concept of probabilistic estimation takes place, based on statistical formulation of the described problem, in which uncertainties are considered, such as that the observation are superimposed with noise and therefore imperfect and represented by random variables. The same accounts for the parameter vector, which drives the generation of another random vector [364]. Hence, the aim via probabilistic estimation is, even though the imperfect information, to gain knowledge about internal states and parameters for the underlying model of the battery, while at the same time to encounter for the measurement and process noise to guarantee for stable parameter and state estimates. In the recent years, after first publications in 2000 under Gregory Plett, who first took the extended Kalman filter and applied it to battery state estimation, many publications have followed in that field. Many different filters where applied to battery technology, from extended Kalman filter to sigma point Kalman filter and even particle filter, for each with different variations trying to decrease the complexity, which did not always succeed ending up in higher complexity and implementation effort. However, it is difficult to say based on the countless publications [13], which of these filter algorithms are best for battery state and parameter application, in respect to filter quality, robustness, required theoretical background information to allow smooth implementation and complexity regarding time-to-market, computational costs on embedded system. This fact has raised the question, which algorithm is the best for this application when favoring the accuracy reached by this filter followed by the fact that the filter can be implemented on the target platform and keeps stable throughout operation. Attention should be paid to avoid too complex methods, meaning, that if more complex methods lead only marginal to an increase in accuracy, the more simple algorithm shall be chosen. This motivates in this work to generate a test environment which quantifies based on the defined criteria the best filter/battery model combination. For model-driven battery diagnostic these methods gain in significance due to improved calculation power, while its challenge remains, in fact, the parallel on-line estimation of the models parameter and states.

Objective

In battery packs, which consists of interconnection up to hundreds or thousands battery cells, deviations caused by production leads to part-to-part inconsistency. It is not possible to generate for each individual cell parameter maps for off-line-integration into production, therefore, adaptive algorithms for on-line re-calibration are required. Hence, one part of this work focuses on the evaluation of state-of-the-art estimators for non-linear systems, to provide a tool which helps to identify the most appropriate filter and model combination in terms of filter accuracy, complexity and robustness. Following tasks are fulfilled to provide a quantitative/qualitative evaluation of filters:

- **Set of filters:** a filter library beyond mainstream filter were implemented to provide different optimization opportunities

- **Classification:** to give an extended analysis, the filters pros and cons are investigated for battery management application
- **Verification:** a multi-level verification process was installed to apply the criteria for evaluation. Besides using measurement data extracted by battery tester, also hardware-in-the-loop test bed to grant for reproducibility and real integration on EV prototype to account for real world scenario. First, the simulation environment reads in the laboratory measurements, followed by the HIL and EV integration. This study shall be presented by the example of lithium-ion batteries, which are characterized as complex, highly non-linear and time-variant systems.

Hence, the specific solution is estimating the in general unknown and unobservable state vector x_k based on all measurements in the observation vector $Y_k \triangleq \{y_1, y_2, \dots, y_n\}$. The key assumption for many applications is that the PDFs associated with the estimation problem under consideration is available or defined a priori. This enables to derive all required information based on a set of assumptions. However, in most cases this assumption is the exception rather than meeting the rule. For on-line applications the parameters of stochastic processes have to be collected and analyzed within a finite set of sampled measurements. In this chapter, the necessary concepts of estimation theory for on-line parameter estimation will be introduced summarized from many sources. Let's consider a problem with observed data $y[0] = \theta + v[0]$ with $v \sim \mathcal{N}(0, \sigma^2)$, we seek to determine a set of $\theta = [\theta_1, \dots, \theta_p]^T$ from a set of observed and noisy data points $y = [y[0], y[1], \dots, y[k-1]]^T$ such that the values of these parameters would lead to the highest probability of obtaining the observed data, $\max_{\theta} p(x, \theta)$ [360][365][366][361][367]. The complete solution to the estimation problem is provided by the *a posteriori* density function, $p(x|y)$. Often, the point estimate of the unknown state x is of interests, resulting in an estimator $\hat{x}$. By inputting a measurement data point y into $\hat{x}$, we obtain an estimate $\hat{x}$. There are different methods available to choosing an estimator, and unknown parameters may be seen as deterministic or random variables and have to be estimated from a discrete-time signal. The theory and techniques of statistical estimation are an important approach towards optimal filtering. The techniques can be categorized in two groups, estimation based on PDFs which contain unknown but deterministic parameters is considered classical estimation, following estimators are based on classical estimation (**θ is deterministic**)[366]

- Minimum Variance Unbiased Estimator (MVU)
- Cramer-Rao Lower Bound (CRLB)
- Best Linear Unbiased Estimator (BLUE)
- Maximum Likelihood Estimator (MLE)
- Least-Squares Estimator (LSE)

The second category, Bayesian estimation, provides a framework which deals with situations where decisions have to be made under the state of uncertainty. The unknown parameters are assumed to be random variables with a given PDF and formalized in *a priori* beliefs to lie within some range of allowable parameters or defined performance [360][365][366]. (**θ is stochastic**)

- Minimum Mean Square Error Estimator (MMSE)
- Maximum A Posteriori Estimator (MAP)
- Linear MMSE Estimator
- Kalman Filter

Since the data are random it is mathematically described by its probability density function (PDF) such as $p(y[0], y[1], \dots, y[k-1]; \theta)$. As example consider the problem of DC level in white Gaussian noise [360][365][366], the observed single point data $y[0] = \theta + w[0]$ with $k = 1$ and θ denotes the mean:

$$p(y[0], \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{1}{2\sigma^2} (y[0] - \theta)^2 \right] \quad (4.3)$$

Clearly, the observed value of $y[0]$ impacts upon the likely value of parameter θ [360][365][366].

The parameter which has to be estimated is termed a realization of the random variable θ whose distribution can be obtained from the joint distribution of the random variables, and defined as the sampling distribution and should be concentrated as closely as possible around the true parameter for better results. The data are described in a statistical way where the joint PDF of the data and parameters are [365]

$$p(y, \theta) = \underbrace{p(y|\theta)}_{\text{(conditional PDF)}} \cdot \underbrace{p(\theta)}_{\text{(prior PDF)}} \quad (4.4)$$

where $p(\theta)$ summarizes the knowledge about θ before any measured data is observed and $p(y|\theta)$ indicates the knowledge provided by the measured data y conditioned on knowing θ . The goal is to design an estimator as a rule that assigns a value of θ from each realization of $\underline{y} = y = [y[0], y[1], \dots, y[k-1]]^T$. An estimate of θ is the value determined for a given realization in the form of $\hat{\theta} = g(y)$. The design of the estimator has to fulfill some degree of robustness allowing a mismatch between the measurement and the PDF assumption. In real application the specification of PDF is critical in determining a good estimator, therefore, it has to be chosen in such a way that it not only fits to the problem constraints and any "a priori" knowledge but also the fact that it is also traceable in a mathematical point of view.

4.2 Assessing Estimator Performance

Beforehand, to classify the "goodness" of an estimator, the sampling distribution has to be determined. The intention of a good estimator design is that its estimate should be as accurate as possible to the true parameter value, however, applying a different or reduced subset of the gathered observation data, it is most unlikely that the estimator provides the identical value. An estimator is a random variable, so its performance can only be described by its PDF or statistically (e.g. computer simulations like Monte-Carlo). When estimating a value of a random variable y using linear mean square estimation based on an observation of a related random variable x , the coefficients of the linear equation $y = ax + b$ depend besides mean and variance of x and y also on their correlation. To evaluate how "good" an estimator works, different estimators of the same parameter have to be checked against each other, based on their sampling distribution. However, since equality of exact same sampling distributions may not be possible, lower-order moments such as mean, variance or mean square error are applied.

Bias of estimator [368][365][366]. One attribute of the estimator defines the difference between the expected value of the estimate and the actual value such as

$$B\{\hat{\theta}_k\} = \{\hat{\theta}_k\} - \theta \quad (4.5)$$

where $\hat{\theta}_k$ is defined as estimation over k data samples. A zero bias would lead to the expected value of the estimate to be equal to the actual value, that is

$$\{\hat{\theta}_k\} = \theta \quad \equiv \quad B = \{\hat{\theta}_k\} - \theta = 0 \quad (4.6)$$

making the estimate to be unbiased and the PDF of the estimator centered at the true value θ . If $B \neq 0$ then the estimator $\hat{\theta} = g(y)$ is said to be biased. If an estimator has the following properties, $\lim_{k \rightarrow \infty} \{\hat{\theta}_k\} = \theta$ for $B \rightarrow 0$ when $k \rightarrow \infty$, the estimator is asymptotically unbiased. In practice it is desirable that the estimator is either unbiased (e.g. mean estimator), closely unbiased (e.g. variance estimator) or asymptotically unbiased.

Variance of estimator [368][365][366]. It is also beneficial for the estimator performance to utilize the available statistics efficiently, e.g. considering the variance to measure the spread of the PDF of θ around its average value. That is,

$$\lim_{k \rightarrow \infty} \text{var}\{\hat{\theta}_k\} = \lim_{k \rightarrow \infty} \{|\hat{\theta}_k - \{\hat{\theta}_k\}|^2\} \quad (4.7)$$

Applying Chebyshev's inequality $\forall \epsilon > 0$ and $\{\hat{\theta}_k\} = \theta$

$$\Pr\{|\hat{\theta}_k - \theta| > \epsilon\} \leq \frac{\text{var}\{\hat{\theta}_k\}}{\epsilon^2} \quad (4.8)$$

shows consistency, meaning that probability of estimation error $e = (\hat{\theta}_k - \theta) \neq \epsilon$ will converge to zero, therefore, the estimator $\hat{\theta}_k$ will converge to the actual value θ with probability one. Choosing an estimator with the smallest variance may not fulfill the small bias requirements and resulting in an increase in bias. The mean square error (MSE) is applied to find a balance between this two opposed requirements [368][365][366].

Mean square convergence [368][365][366]. A stronger property of convergence is provided by the mean square convergence. The estimator $\hat{\theta}_k$ converges towards the real value θ if

$$\lim_{k \rightarrow \infty} \underbrace{\{|\hat{\theta}_k - \theta|\}^2}_{\text{mean square error}} = \sigma_{\hat{\theta}}^2 + |B_{\hat{\theta}}|^2 = 0 \quad (4.9)$$

and said to be consistent, is asymptotically unbiased, its variance goes to zero as $N \rightarrow \infty$ and finally converges to the actual value of the parameter. Considering the case of a DC level, A in random noise and modeled as $y[n] = A + w[n]$, where $w[n] \sim \mathcal{N}(0, \sigma^2)$ stands for the zero mean random process. Given measurement samples $\{y[0], y[2], \dots, y[k-1]\}$, the first reasonable estimator to determine the parameter A is the sample mean estimator, $\hat{A} = \frac{1}{k} \sum_{n=0}^{k-1} y[n]$. The estimator is a function of random data, which makes itself to a random variable. Therefore, the performance can only be judged on statistical basis. The mean of parameter $\hat{A}$ is calculated by

$$\{\hat{A}\} = \left\{ \frac{1}{k} \sum_{n=0}^{k-1} y[n] \right\} = \frac{1}{k} \sum_{n=0}^{k-1} \{y[n]\} = A \quad \equiv \quad \text{unbiased} \quad (4.10)$$

Similar, the variance σ^2 of A based on measurement samples $y[n]$ would be

$$\begin{aligned} \{\hat{A}^2\} &= \text{var}\{\hat{A}\} = \text{var}\left\{ \frac{1}{k} \sum_{n=0}^{k-1} y[n] \right\} = \frac{1}{k^2} \sum_{n=0}^{k-1} \text{var}\{y[n]\} \\ &= \frac{1}{k^2} N \sigma^2 = \frac{\sigma^2}{k} \quad \text{as } k \rightarrow \infty \quad \text{variance} \rightarrow 0 \quad \equiv \quad \text{consistent} \end{aligned} \quad (4.11)$$

In a sense of classical estimation, the goal is to design a "good" estimator of unknown, but deterministic parameters. Here, unbiased estimators yield on average the true value of the parameters, for that, the real value of the parameter does not play a role, i.e. $(\hat{\theta}) = \theta$, with $a < \theta < b$ where (a, b) defines the range of possible values for θ . However, since the estimator may only be unbiased for certain values of the unknown parameter θ , this would result in an estimator which is not unbiased. Biased estimators will introduce a largely deterministic, systemic error, and have to be avoided for stochastic properties determination. When several unbiased estimators of the same parameter from an independent set of measured data are available $\{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_L\}$, a better estimator is determined when

averaging over all estimators [368][365][366]:

$$\hat{\theta} = \frac{1}{L} \sum_{l=1}^{L-1} \hat{\theta}_l \quad \rightarrow \quad \{\hat{\theta}\} = \theta \quad (4.12)$$

assuming that all estimators are unbiased, uncorrelated and have the same variance, $\text{var}\{\hat{\theta}\} = \frac{1}{k^2} \sum_{n=1}^{k-1} \text{var}\{\hat{\theta}_k\} = \frac{\text{var}\{\hat{\theta}_k\}}{k}$, with $k \rightarrow \infty$, $\hat{\theta} \rightarrow \theta$ leading to a decrease in variance for increasing sample space. This does not apply to unbiased estimators, no matter how many estimators are averaged. Theoretically, the most logical criterion for estimation is the mean squared error, which is in practice unrealizable, since the estimator may be unbiased for certain values of the unknown parameter θ , resulting in an unbiased estimator. Nevertheless, averaging biased estimators will not generate unbiased properties, $\text{var}\{\hat{\theta}\} = \frac{\text{var}\{\hat{\theta}_l\}}{L}$, as $L \rightarrow \infty$, $\hat{\theta} \rightarrow \theta$ (consistent).

Minimum variance criterion In searching for an optimal estimator we need to define some optimality criterion. A natural criterion is the mean square error (MSE), defined as

$$\text{MSE}(\hat{\theta}) = \{(\hat{\theta} - \theta)^2\} \quad (4.13)$$

However, if we further convert the MSE expression above, it becomes clear, that this criterion leads, however, to unrealizable estimators - namely, ones which are not solely a function of the data. It is most likely, that for any given criterion, which depends on the unknown parameter θ , it leads to an impractical estimator [368][365][366].

$$\begin{aligned} \text{MSE}(\hat{\theta}) &= \left\{ \left[(\hat{\theta} - \hat{\theta}) + ((\hat{\theta}) - \theta) \right]^2 \right\} \\ &= \text{var}(\hat{\theta}) + \left[(\hat{\theta}) - \theta \right]^2 = \text{var}(\hat{\theta}) + B^2(\hat{\theta}) \end{aligned} \quad (4.14)$$

From the implementation point of view, it is not practical to apply the minimum MSE estimator for e.g. battery monitoring, it needs to be adapted by constraining the estimator to be unbiased. Instead of minimizing MSE, the variance of the unbiased estimator can be minimized. Such an estimator is called minimum variance unbiased (MVU) estimator, but does not always exists. Its PDF of the error is concentrating around zero and provides therefore an improved likelihood for lower estimation errors. In case the MVU estimator exists, following approaches may be applied to identify the estimator.

Cramer-Rao lower bound CRLB defines the lower bound on the variance of classical estimators. Hereby, the variance is at least as high as the inverse of the Fisher information.

Rao-Blackwell-Lehmann-Scheffe (RBLS) states that $\forall \theta \text{ MSE}(\hat{\theta}^*) \leq \text{MSE}(\hat{\theta})$ where $\hat{\theta}^*(y[0], y[1], \dots, y[k]) = (\hat{\theta}(y[0], y[1], \dots, y[k]) | T(y[0], y[1], \dots, y[k]))$ is at least as good

as θ in terms of the MSE [369]. Further, if an unbiased estimator is a function of a complete sufficient statistic T , it satisfy MVUE.

Estimator Accuracy Consideration

All information which allows to draw conclusions about the system state or parameters based on an estimator is embodied in the observed data and the underlying PDF. It is firstly evident, accuracy of estimator depends directly on the PDF, therefore, it is worthwhile to investigate the circumstances on how the accuracy of an estimator depend upon the relationship between unknown parameters and the PDF. A key role in estimating parameters form a set of statistics is the likelihood function, the PDF is considered as a function of unknown parameters. The likelihood function has a distinctive property, its "sharpness", which determines how accurate the parameters may be estimated. It is measured by its "curvature" and calculated as [368][365][366]

$$-\frac{\partial^2 \ln p(y[0], \theta)}{\partial \theta^2} = \frac{1}{\sigma^2} \quad (4.15)$$

Clearly, the curvature of the log-likelihood function increases with an decreasing variance σ^2 . To reduce the influence of the second derivative upon $y[0]$, the function is represented in a statistical way and measures the average curvature of the log-likelihood function [368][365][366].

$$-E \left[\frac{\partial^2 \ln p(y[0], \theta)}{\partial \theta^2} \right] \quad (4.16)$$

with respect to $p(y[0]; \theta)$. The larger the quantity in the log-likelihood function, the smaller the variance of the estimator. It can be concluded, since the likelihood function is a random variable that a minimum variance unbiased (MVU) solution might be preferable. But, what would be the sufficient information needed to derive an MVUE estimator?

Minimum Variance Unbiased Estimator

Generally, it is difficult to design the MVU estimator, however, in signal processing linear relations are often first and sufficient approximations to describe the underlying nature of quantities. This enables a linear data model to determine the MVU estimator, if the data observed can be described as [368][365][366]

$$y = \mathbf{H}\theta + v \quad (4.17)$$

where y is an $N \times 1$ vector of observations, $\mathbf{H}$ is a known $N \times p$ observation matrix with $N > p$ and rank p , θ is a $p \times 1$ vector of parameters to be estimated and v defines the $N \times 1$ noise vector with a PDF $\mathcal{N}(0, \sigma^2 I)$. The columns of $\mathbf{H}$ must be linearly independent, so

that $(\mathbf{H}^T \mathbf{H})^{-1}$ becomes invertible. The MVU estimator for the linear model is

$$\hat{\theta} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (4.18)$$

for which the covariance matrix of $\hat{\theta}$ is

$$\mathbf{P}_{\hat{\theta}} = \sigma^2 (\mathbf{H}^T \mathbf{H})^{-1} \quad (4.19)$$

Based on the linear model, the estimator is unbiased and efficient since it fulfills the constraint of CRLB. Its statistical performance is completely specified, and $\hat{\theta}$ is a linear transformation of a Gaussian vector $\mathbf{y}$ and therefore $\hat{\theta} \sim \mathcal{N}(\theta, \sigma^2 (\mathbf{H}^T \mathbf{H})^{-1})$.

Best Linear Unbiased Estimator

Linear estimator are an important class of estimators because of their reduced abstracted form dealing with only the 1st and 2nd moments [368][365][366]. A great advantage is in the ease of implementation, especially when considering real-time filtering problems. Unfortunately, the MVUE is often not computable, for instance CRLB or Rao-Blackwell theorem are not applicable because the PDF of the data is unknown and the randomness is only available up the first and second order moments. In such cases, we have to accept a suboptimal solution by imposing some constraints on the estimator and data model. For the estimator to be unbiased, the mean of the data must obey a linear relationship with the true parameter and has minimum variance, this leads to $\Rightarrow$ **Best Linear Unbiased Estimator (BLUE)**. Again, this criteria might not always be feasible, an alternative could be the Maximum Likelihood Estimation (MLE), however, there is an important subclass of problems where BLUE constraints hold, which are called the Gauss-Markov assumptions. In many applications we need to make decisions about θ from observations $y \sim p(y, \theta)$. The density of X varies within a family of distributions, depending on different choices of θ . In statistical sense, this means, that any information we want to extract from $\mathbf{x}$ depends only on the p parameters $\theta = [\theta_1, \theta_2, \dots, \theta_p]^T$. Therefore, we compress the N measurements into a reduced dimensional statistics that consists of all useful information resulting in the same quality. Nevertheless, even though we are not able to find an efficient estimator, it is still the goal to design, if exists, the MVU estimator. First we find a transformation $T(\mathbf{x})$ of lower dimension that contains all information with respect to the parameter θ . This data may still be too long, a minimal sufficient statistics is extracted, which includes all necessary information that is useful for the parameter estimation. To solve for the above conditional distributions is certainly a challenging task in finding sufficient statistics directly. The following theorem is used to verify sufficient statistics, the Neyman-Fisher factorisation. If possible, it allows to factor the PDF [368][365][366] $p(y, \theta)$ as

$$p(y, \theta) = g(T(y), \theta) h(y) \quad (4.20)$$

where $g(\cdot)$ is a function depending on y only through $T(y)$, while function h only depends on y . This meant that $T(y)$ is a sufficient statistic for parameter θ . Finally, to

solve the problem in finding MVU estimator, first, the sufficient statistics is checked for completeness followed by the verification of uniqueness of $g(\cdot)$ applying the Rao-Blackwell-Lehmann-Sheffe method. The problem of finding the MVU estimator results fundamentally from the unknown data PDF or simply cannot be assessed. This limits the methods and CRLB cannot be applied and a suboptimal solution is derived. The BLUE is restricted and data is assumed linear, further restricts estimates to be unbiased, and lastly BLUE is restricted to minimize the variance of the estimates. All these limitations lead to an estimator which requires only the mean and the variance of the data, but not the PDF. On the other hand, the PDF of the estimates cannot be computed. Consider the measured data $y = [y[0], y[1], \dots, y[k-1]]^T$, whose PDF $p(y, \theta)$ depends on the unknown parameter θ , BLUE [368][365][366] shall be designed such that,

1. Choose a linear estimator for the measured data

$$\begin{aligned} y[n] & \quad , \quad n = 0, 1, \dots, k-1 \\ \hat{\theta} = \sum_{n=0}^{k-1} a_n y[n] = a^T y & \quad , \quad \text{where } a = [a_0, a_1, \dots, a_{k-1}] \end{aligned} \quad (4.21)$$

2. Restrict the estimate to be biased:

$$(\hat{\theta}) = \sum_{n=0}^{k-1} a_n (y[n]) = \theta \quad (4.22)$$

3. Minimize the variance:

$$\begin{aligned} \text{var}(\hat{\theta}) &= \left\{ [\hat{\theta} - (\hat{\theta})]^2 \right\} = \left\{ [a^T - a^T(y)]^2 \right\} \\ &= \left\{ a^T [y - (y)] [y - (y)]^T a \right\} = a^T \mathbf{P} a \end{aligned} \quad (4.23)$$

The BLUE will be optimal if the found MVU estimator is linear in θ . It can be represented as the amplitude estimation of known signals superimposed with noise. Therefore, the optimization task lies in the minimization of the variance subject to the unbiased constraint

Therefore, to determine the BLUE, we only need the information of the scaled mean s and the covariance $\mathbf{P}$. This allows an estimation based on the first two moments without the knowledge of the functional relation of the PDF. The estimation becomes $\hat{\theta} = (\mathbf{H}^T \mathbf{P}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{P}^{-1} y$ with the covariance matrix $\mathbf{P}_{\hat{\theta}} = (\mathbf{H}^T \mathbf{P}^{-1} \mathbf{H})^{-1}$. It becomes obvious that for the case the data is Gaussian the BLUE yields the Gauss-Markov theorem or when noise is Gaussian then BLUE becomes a MVU estimator [368][365][366].

Maximum Likelihood Estimator

The maximum likelihood estimation (MLE) is an approximately optimal estimator for problems, where the MVU estimator does not exist or BLUE cannot fulfill certain constraints, e.g. the models are restricted to be linear [368][365][366][370]. The fundamental concept relies on the property that for large data samples the MLE is asymptotically efficient. It is claimed that for a large number of sample points or as $k \rightarrow \infty$, the MLE is efficient, meaning

$$(\hat{\theta}) \rightarrow \theta \quad (4.24)$$

$$\text{var}(\hat{\theta}) \rightarrow \text{CRLB} \quad (4.25)$$

If the MLE satisfies Equation 4.24 it is said to be asymptotically unbiased and the estimator $\hat{\theta}$ is consistent. Further if it also fulfills Equation 4.25 its asymptotically efficient. The framework consists of a measurement device observation, Y , whose distribution depends on θ and superimposed with noise. In comparison to the later framework, the Bayesian estimation, here the parameter is not modeled as a random variable. Even though θ is deterministic, its estimate is a random variable, because it is a function of the observed data. The distribution of the parameter estimate $\hat{\theta}$ depends on the distribution of Y , which is influenced by θ . For θ we do not know the prior distribution which means in mathematical point of view, that many different probabilistic models are possible. In many cases only a small set of models have to be considered from which we need to distinguish. The MLE chooses a set of model parameter values that maximizes the likelihood function. Let $y_0, y_1, \dots, y_k$ be the measurement data, described as independent and identically distributed random variables (i.i.d) with the PDF $p(y, \theta)$ where θ is a $(k \times 1)$ vector of parameters that characterize $p(y, \theta)$. The joint density of the data is, by independence, equal to the product of the marginal densities such as [370]

$$p(y_0, y_1, \dots, y_k; \theta) = p(y_1, \theta) \cdots p(y_k, \theta) = \prod_{i=1}^k p(y_i, \theta) \quad (4.26)$$

The joint density is an N -dimensional function of the measurement data $y_0, y_1, \dots, y_k$ given the parameter (vector) θ . It satisfies

$$\begin{aligned} p(y_0, y_1, \dots, y_k, \theta) &\geq 0 \\ \int \cdots \int p(y_1, \dots, y_k, \theta) dy_1 \cdots dy_k &\geq 1 \end{aligned} \quad (4.27)$$

The likelihood function is defined as the joint density treated as a function of the parameter θ , hereby the PDF $p(y, \theta)$ when θ is regarded as a non-random variable, while the ML estimate is the value of θ that maximizes $p(y, \theta)$. The likelihood function is defined

as the joint density over the θ

$$\mathcal{L}(\theta|y_0, y_1, \dots, y_k) = p(y_0, y_1, \dots, y_k; \theta) = \prod_{i=1}^k p(y_i, \theta) \quad (4.28)$$

The likelihood function is a function of θ given the data $y_0, y_1, \dots, y_k$ and not a proper PDF [370], it is always positive but not necessary adding to one

$$\int \cdots \int \mathcal{L}(\theta|y_0, y_1, \dots, y_k) d\theta_1 \cdots d\theta_k \neq 1. \quad (4.29)$$

$Y = (y_0, y_1, \dots, y_k)$ defines the observed measurement data, then the joint PDF and likelihood function may be expressed as $p(Y, \theta)$ and $\mathcal{L}(\theta|Y)$. Intuitively, we have to find the value that makes the observed sample the most probable, in other words, ML estimator shall maximize the probability of the selected model with the observed data. Therefore, suppose we take a random sample from the PDF $f(y_i, \theta)$ and looking for an estimate for θ [370].

$$\hat{\theta}_{\text{MLE}} = \arg[\max_{\theta} \mathcal{L}(\theta|Y)] \quad (4.30)$$

Formally, the maximum likelihood estimator $\hat{\theta}_{\text{MLE}}$ is defined as the value of θ that maximizes $\mathcal{L}(\theta|Y)$. That is, $\hat{\theta}_{\text{MLE}}$ solves [370]

$$\max_{\theta} \mathcal{L}(\theta|Y) \quad (4.31)$$

To overcome the difficulty in finding a direct solution for the optimization problem in Equation 4.31, instead the log-likelihood function $\ln \mathcal{L}(\theta|Y)$ is maximized. The $\ln(\cdot)$ is a monotonic function, meaning, determining the value of θ that maximizes $\ln \mathcal{L}(\theta|Y)$ will also maximize $\mathcal{L}(\theta|Y)$. The MLE is defined such that $\hat{\theta}_{\text{MLE}}$ as the value of θ that solves [370]

$$\max_{\theta} \ln \mathcal{L}(\theta|Y) \quad (4.32)$$

with the log-likelihood form

$$\ln \mathcal{L}(\theta|Y) = \ln \left(\prod_{i=1}^k p(y_i, \theta) \right) = \sum_{i=1}^k \ln p(y_i, \theta) \quad (4.33)$$

Numerical Determination of the Maximum Likelihood Estimator

A specific characteristic of the MLE is that it can always be found numerically for given sample points. The procedure of finding the MLE is a maximization problem, therefore first we find the log-likelihood function in $f(y_i, \theta)$, differentiate it with respect to θ , set it

to zero and solve it for θ . The established method to solve for the maximization performs continuously grid search over the allowed values of θ using small step size. However, if the sample space of θ is not constrained to an interval, an iterative maximization procedure may be computationally more feasible, such as the scoring method (SCM), Newton-Raphson (NRM) and the expectation-maximization (EM) algorithm. To prevent the iterative procedures to reach only a local maximum, their initialization have to be close to the true maximum. If the constraints are fulfilled, the MLE may be found by differentiating $\mathcal{L}(\theta|\mathbf{Y})$ and solving the first order conditions like [370]:

$$\frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta} = 0 \quad (4.34)$$

Since θ ($n \times 1$) the first order conditions k , eventually non-linear, equations with n unknown values as [370]

$$\frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta} = \begin{pmatrix} \frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta_1} \\ \vdots \\ \frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta_k} \end{pmatrix} \quad (4.35)$$

For a non-linear equation the methods attempt an iteratively solution. Let

$$g(\theta) = \frac{\partial \ln \mathcal{L}(\theta|Y)}{\partial \theta} \quad (4.36)$$

with an initial guess θ_0 for the Equation 4.34. The function can be approximated around θ_0 as

$$g(\theta) \approx g(\theta_0) + \left. \frac{dg(\theta)}{d\theta} \right|_{\theta=\theta_0} (\theta - \theta_0) \quad (4.37)$$

The iterative NR method calculates the new guess θ_{k+1} using

$$\theta_{n+1} = \theta_n - \frac{g(\theta_n)}{\left. \frac{dg(\theta)}{d\theta} \right|_{\theta=\theta_n}} \quad (4.38)$$

The derivative of the log-likelihood function is defined by $g(\theta)$, the estimator $\hat{\theta}_{\text{MLE}}$ is implemented as [370]

$$\theta_{k+1} = \theta_k - \left[\frac{\partial^2 \ln \mathcal{L}(\theta|Y)}{\partial \theta^2} \right]^{-1} \left. \frac{\partial \ln \mathcal{L}(\theta|Y)}{\partial \theta} \right|_{\theta=\theta_k} \quad (4.39)$$

An alternative to the NR is the *method of scoring*. It recognizes that the vector $\begin{pmatrix} \frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|\mathbf{Y})}{\partial \theta_1} \\ \vdots \\ \frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta_k} \end{pmatrix}$ consists of the derivatives of the log-likelihood functions over all θ and denoted as

$$\mathcal{S}(\theta|Y) = \frac{\partial \ln \mathcal{L}(\theta|Y)}{\partial \theta} \quad (4.40)$$

and satisfies

$$\mathcal{S}(\hat{\theta}_{\text{MLE}}|Y) = 0 \quad (4.41)$$

It indicates how sensitive the log-likelihood is to variation in parameters [370]. The score of a sample from a random process becomes the sum of the scores for each observed data y_i :

$$\mathcal{S}(\theta|Y) = \sum_{i=1}^k \frac{\partial \ln p(y_i, \theta)}{\partial \theta} = \sum_{i=1}^k \mathcal{S}(\theta|y_i) \quad (4.42)$$

with $\mathcal{S}(\theta|y_i) = \frac{\partial \ln p(y_i, \theta)}{\partial \theta}$ is the score regarding y_i . The matrix of the second derivative of the log-likelihood is called the Hessian and denoted as

$$\mathcal{H}(\theta|Y) = \frac{\partial^2 \ln \mathcal{L}(\theta|Y)}{\partial \theta \partial \theta'} = \begin{pmatrix} \frac{\partial^2 \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta_1^2} & \dots & \frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta_1 \partial \theta_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta_k \partial \theta_1} & . & \frac{\partial \ln \mathcal{L}(\hat{\theta}_{\text{MLE}}|Y)}{\partial \theta_k^2} \end{pmatrix} \quad (4.43)$$

Taking the negative expectation of the Hessian will lead to the expected *Fisher information matrix* which is defined as [371]

$$\mathcal{I}(\theta|Y) = -E[\mathcal{H}(\theta|Y)] \quad (4.44)$$

an is the covariance matrix of the score [370]. This method then becomes

$$\theta_{k+1} = \theta_k + \mathcal{I}^{-1}(\theta|Y) \left. \frac{\partial \ln \mathcal{L}(\theta|Y)}{\partial \theta} \right|_{\theta=\theta_k} \quad (4.45)$$

Expectation-Maximization

Under some conditions, this algorithm guarantees to converge at least to a local maximum. It works under the assumption, that some data sets are more appropriate to determine the MLE than the provided set. In some cases the MLE may solve a p -multidimensional minimization problem. Instead, the EM decomposes the problem to a $p \times 1$ -dimensional problem based on a complete (many) $\rightarrow$ incomplete (one) data transformation as $y = g(x_1, x_2, \dots, x_k) = g(x)$. x stands for the new "incomplete" data for which we find the MLE of θ by maximizing $\ln p_y(y, \theta)$. Thereby, the log-likelihood function is replaced by

its conditional expectation such as

$$x|y[\ln p_x(x, \theta)] = \int \ln p_x(x, \theta) p(x|y; \theta) dx \quad (4.46)$$

To find the solution for $p(x|y; \theta)$ the knowledge of the θ is necessary. This will be solved iteratively in two steps with θ_k denoting the k th guess of the MLE estimator [368].

- **Expectation (E)**: determines the average log-likelihood of the complete data

$$U(\theta, \theta_k) = \int \ln p_x(x, \theta) p(x|y, \theta_k) \quad (4.47)$$

- **Maximization (M)**: maximizes the average log-likelihood function of the complete data [368]

$$\theta_{k+1} = \arg \left[\max_{\theta} U(\theta, \theta_k) \right] \quad (4.48)$$

which for convergence may result in MLE.

The Least Squares Approach

When it comes to an excellent estimator, no bias and minimal variance are two aspects we will focus on. By looking for minimizing the discrepancy (on the average) between the estimate and the ground truth implicitly, the variance is set to be a criterion of goodness. The least square (LS) method minimizes the gap between the known corrupted data (as $x[n]$) and the ideal clean data in the squared sense [368]. Before that, a model with unknown parameter θ needs to be determined to actually produce the signal. So, the signal (as $s[n]$) is totally deterministic. Since the existence of uncertainty, which mainly includes the sensor noise and the selection of inaccurate model, the objective observation is more like a polluted version of $s[n]$, marked as $x[n]$. The least squares estimator (LSE) finds the value of the parameter θ that makes the gap between the two signals smallest. The closeness is determined by [368]

$$J(\theta) = \sum_{n=0}^{N-1} (x[n] - s[n])^2, \quad n = 1, 2, \dots, N-1 \quad (4.49)$$

the functional J depends on the parameter θ in the model $s[n]$. For now, no probabilistic assumptions are made, thus this approach is equally feasible for any noise, no matter Gaussian or not. But it is obvious that, same as any model errors, the performance of LSE will be affected by the noise type. The application of LSE is wild but usually related to an absence of well-defined statistical profile of data features, or the lack of better estimator. It is also possible that a better estimator consumes more computing [368].

The simple assumption is that the model is linear, which is the linear LS method. Then the parameter becomes scalar,

$$s[n] = \theta h[n] \quad (4.50)$$

note that $h[n]$ is some known signal. The closeness can be expressed as

$$J(\theta) = \sum_{n=0}^{N-1} (x[n] - \theta h[n])^2 \quad (4.51)$$

To create the LSE, the parameter is set as

$$\hat{\theta} = \frac{\sum_{n=0}^{N-1} x[n]h[n]}{\sum_{n=0}^{N-1} h^2[n]} \quad (4.52)$$

The LS error changes into

$$\begin{aligned} J(\hat{\theta}) &= \sum_{n=0}^{N-1} (x[n] - \hat{\theta}h[n])(x[n] - \hat{\theta}h[n]) \\ &= \sum_{n=0}^{N-1} x[n](x[n] - \hat{\theta}h[n]) - \underbrace{\hat{\theta} \sum_{n=0}^{N-1} h[n](x[n] - \hat{\theta}h[n])}_S \\ &= \sum_{n=0}^{N-1} x^2[n] - \hat{\theta} \sum_{n=0}^{N-1} x[n]h[n] \end{aligned} \quad (4.53)$$

It is easy to see $S = 0$. The above derivation is a minimization that can be also written as [368]

$$J_{min} = \sum_{n=0}^{N-1} x^2[n] - \frac{\left(\sum_{n=0}^{N-1} x[n]h[n]\right)^2}{\sum_{n=0}^{N-1} h^2[n]} \quad (4.54)$$

4.3 Bayesian Philosophy and Estimation

In comparison to the classical methods described earlier, where the parameter vector θ and its prior are unknown but assumed to be deterministic, the variance of the prediction depends on θ , its minimum-variance unbiased estimator may or may not be available, and for MCS, where each iteration runs with a pre-defined θ to compute the mean and variance of individual θ . The Bayesian philosophy on the other hand takes θ as random variable with a known prior PDF and estimates a realization of θ using measurements, where its variance is independent of θ . For MCS, θ is random initialized and then the mean and variance over all θ computed. Since a Bayesian estimate always exists, most methods in parameter and state estimation, either linear or non-linear, are originated in Bayes theorem and differ only in the determination of the a priori. The Bayesian framework has its objective in estimation of successive values of a random state vector x_k given a random measurement vector Y_k . For the state equation receptively parameter vector which describe the dynamics of the process, in our case, the battery dynamic characteristics, the stochastic term is inherent while inadequate un-modeled secondary reactions are added as random noise. For the measurement equation the stochastic nature of e.g. quantization or sensor noise, also random measurement noise is added [372]. For now, we assume that the random vector $\mathbf{x}$ can be represented by its known prior probability $p(x_k)$ while the true parameter/state vector is known, then we can define the probability density of the measurement Y_k is given by the conditional density or likelihood function $p(Y_k|x_k)$. With these assumptions the complete statistical properties of the measurements can be described. With the Bayes Theorem, the posterior probability of x can be calculated as

$$p(x_k|Y_k) = \frac{p(Y_k|x_k)p(x_k)}{p(Y_k)} \quad (4.55)$$

which includes all information about x after taking into account the measurements. The probability density function of the measurement can be found from

$$p(Y) = \int_{-\infty}^{\infty} p(Y|x)p(x)dx \quad (4.56)$$

To solve the estimation framework, the likelihood $p(y_k|x_k)$ and the a-priori density $p(x_k|Y_{k-1})$ or their joint density function are required and define the statistical estimation model [372]. The joint density can be found from the posterior function $p(x_k|Y_k)$ is proper for the case that the integral is present, otherwise, it is said to be improper and $p(Y) = 1$ [354]. The posterior distribution can be seen as the general solution to the statistical estimation problem, additionally with its help any point estimate, if exists, of $\mathbf{x}$ can be extracted. Hence, the latest measurement is used to calculate the posteriori density function with the help of the likelihood-density function.

4.3.1 Statistical Inference in Estimation

In statistical filter theory [353] [373][374][375], [376] [360][367][377][362] the filtering problem describes the general idea to form the best estimate of the internal states and parameter values only from incomplete or noisy observations applying optimal filtering technique. The mathematical model for dynamic systems describes the filtering problem either by continuous, continuous-discrete or fully discrete form. Generally, stochastic differential equations (SDEs) are used to model the continuous evolution of dynamic systems, and in which at least one term is subject to stochastic process [378]. Stochastic processes in physics may be modeled via LANGEVIN equations while in engineering dynamic systems are described by differential equations superimposed by process noise as errors of measurement. Mainly two distinctive classes of equations for random effects in differential equations are discussed using different methods of analysis [379]. Consists an ordinary differential equation of random coefficients or known regular stochastic process, the so called random differential equations and differentiable functions of t can be solved as ordinary differential equations [380]. When the forcing of the system is defined by unpredictable processes such as Gaussian white noise, the second class is called (SDEs) but interpreted as integral equations with Itô or STRATONOVICH stochastic integrals. Their solutions consist of a non-differentiability part from the Wiener processes in the integrals. Continuous, stochastic systems can be described as in following, the oldest representation of SDE and known as LANGEVIN type equation containing a deterministic part (ordinary differential equation) and additional random white noise.

$$\dot{x}(t) = f_{CT}(x, u; t) + K(x, u; t)\kappa(t) \quad (4.57)$$

$$y(k) = h_{CT}(x_k, u_k; t_k) + \Gamma(x_k, u_k; t_k)\gamma(t_k) \quad (4.58)$$

where f_{CT} represents the continuous, deterministic system quantity, K and Γ the distribution matrix of the system and measurement quantity, κ and γ stands for the system and measurement noise respectively, h represents the distribution matrix for the measurement. Since we are operating the BMS and its generated code on microcontroller, hence, the discrete battery system models are mapped by means of discrete stochastic LANGEVIN type equations such as

$$x(k) = f(x_{k-1}, u_{k-1}) + K(x_{k-1}, u_{k-1})\kappa_{k-1} \quad (4.59)$$

$$y(k) = h(x_k, u_k) + \Gamma(x_k, u_k)\gamma_k \quad (4.60)$$

Solving SDEs falls back on the techniques for solving ordinary differential equations, with the change that they consider stochastic dynamics. The conventional approach for deterministic dynamical system behavior is described by a set of first-order differential equations on $\mathbb{R}^n$. It is written in a state space form and emphasizes trajectories which interprets the solution of the differential equations for given initial states [381]. Unlike ODE represented in derivate form, SDEs are given in differential form. That is due to the fact that the random process are continous but non-differentiable. The Wiener process is the earlier version of random behavior and called Brownian motion. It is fundamental in

describing stochastic processes, one of its properties is that it defines the integral of noise which is independent of frequency (white noise) [382] and used when randomness needs to be added for processes which cannot be deterministically modeled.

Applying the concept of stochastic differential equations κ is interpreted as generalized white noise process. To solve for these equations a more advanced concept of integration is necessary. The LANGEVIN equation is interpret symbolically as a stochastic differential

$$dX = a(t, X)dt + b(t, X)dW_t \quad (4.61)$$

where a and b are positive constants, dW_t represents the Wiener process. The SDE described earlier has to be transformed into an equivalent integral equation [382]

$$X(t) = X_{t_0} + \int_0^{t_0} a(s, y)ds + \int_{t_0}^t b(s, y)dW_s \quad (4.62)$$

where the most right integral represents an ITO integral. With $c = t_0 < t_1 < \dots < t_{n-1} < t_n = d$ points on the interval $[c, d]$, we get an integral defined by Riemann

$$\int_c^d f(x)dx = \lim_{\Delta t \rightarrow 0} \sum_{i=1}^n f(t_{i'})\Delta t_i \quad (4.63)$$

which is approximated by Riemann-Stieltjes sums and depends on the choice of the grid interval points [382] $\Delta t_i = t_i - t_{i-1}$ and $t_{i-1} \leq t_{i'} \leq t_i$. The ITO-STRATONOVICH integral is the most common used method

$$\int_c^d f(t)dW_t = \lim_{\Delta t \rightarrow 0} \sum_{i=1}^n f(t_{i-1})\Delta W_i \quad (4.64)$$

with $\Delta W_i = W_{t_i} - W_{t_{i-1}}$ stands for a step of Brownian motion across the grid interval [382].

The SDEs can be solved by the chain rule for stochastic differentials introduced as ITO formula. However, only rarely the SDEs have a closed-form solution and therefore a numerical approximation is required. Complete solutions following methods applied to approximate of ordinary differential equations, eg. EULER method, the analogue for SDEs is the EULER-MARUNYAMA. The same also applies to LANGEVIN equations, it is impossible to analytically solve the system in terms of the elementary processes [382]. A solution is derived from the EULER-MARUNYAMA approximation, a stochastic process called Ornstein-Uhlenbeck process which satisfies conditions for large deviations on MLE [383]. Each calculated approximate path results in a random variable and converges strongly such as EULER-MARUNYAMA or MILSTEIN method, which makes use of higher-order terms analogue to the Taylor series or the RUNGE-KUTTA Method but suffer of

higher complexity. Weak solvers (e.g. Monte Carlo estimates) are simpler because they focus on computing quantities of stochastic processes as the expected value of a random variable (mean) or variance of the probability distribution. It can be summarized, to consider random phenomena and solving SDEs numerical approximations are important. Strong convergence using Taylor and RUNGE-KUTTA is necessary when trajectory-level properties are studied in detail [382]. On the other side, when only the parameters of a probability distribution, such as mean and variance are sufficient, in such cases weak convergence may be sufficient when computing costs are of great importance. Generally, to solve SDF a normal random number generator is required.

Based on that discussion, this work is focusing only on time-discrete and dynamic systems. The generation of reference models is proving to be difficult and the question which approximation gives the correct solution to the integrals. Additionally, it creates the numeric solution is based on simple algorithms such as Heun or RUNGE-KUTTA, but none for full stochastic quantities. Dynamic systems can be described by dynamic state space models (DSSM). A joint finite-dimensional distribution of a time series requires assumptions of independence. Markovian dependence states that the information about the time series included in the observations taken up to time $t - 1$ is the same as the information carried by y_{t-1} alone.

Since the Markovian framework for observations might not be feasible in all applications, state space models may be used. The unobservable state x_k develops with discrete time steps (samples) as a partial or indirect Markov process with an initial probability $p(y_0)$

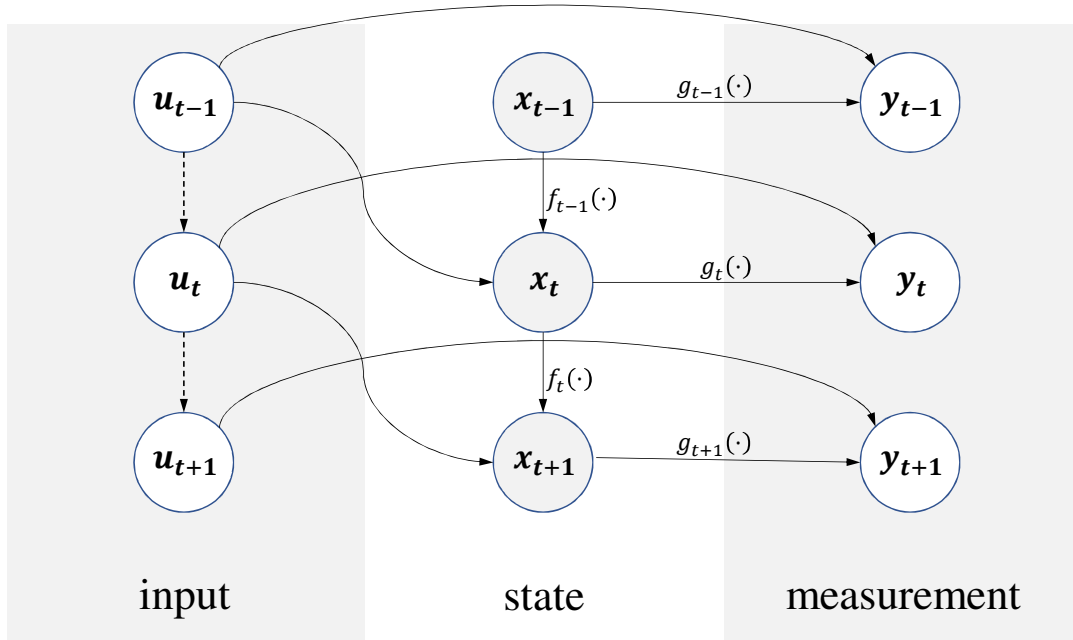


Figure 4.2: Hidden Markov model for battery state and measurement evolution

according to its transition $p(x|x_{k-1})$ while y_k describes the noisy, independent observations according to likelihood $p(y_k|x_k)$. The reference dynamic system framework representing lithium ion battery characteristics shall be defined and used further. As characterized earlier, we assume a discrete time system which is defined by non-linear model represented in difference state space form as

$$x_k = f(x_{k-1}, u_{k-1}) + K(x_{k-1}, u_{k-1})\kappa_{k-1} = \mathcal{F}(x_{k-1}, u_{k-1}, \kappa_{k-1}) \quad (4.65)$$

with the stochastic system equation $\mathcal{F}$. The measurement equation $\mathcal{H}$ is written as

$$y(k) = h_k(x_k, u_k) + \Gamma(x_k, u_k)\gamma_k = \mathcal{H}(x_k, u_k, \gamma_k) \quad (4.66)$$

Where statistical models are available for the system and measurement processes, one seeks the a posteriori conditional probability density function of state x_k

$$p(x_k|Y_k) \quad (4.67)$$

since all statistical information of interests are involved. $Y_k = \{y_1, y_2, y_3, \dots, y_k\}$ represents a sequence of sampled measurements until time k . Is the posterior probability known, it is possible to estimate the unobservable system states using e.g. Minimum-Mean-Square-Estimator (MMSE) or Maximum-A-Posteriori-Estimator (MAP) such as [380]

$$\hat{x}_k \stackrel{MMSE}{=} \mathbb{E}[p(x_k|Y_k)] = \int_{\mathbb{R}^n} x_k \cdot p(x_k|Y_k) dx_k \quad (4.68)$$

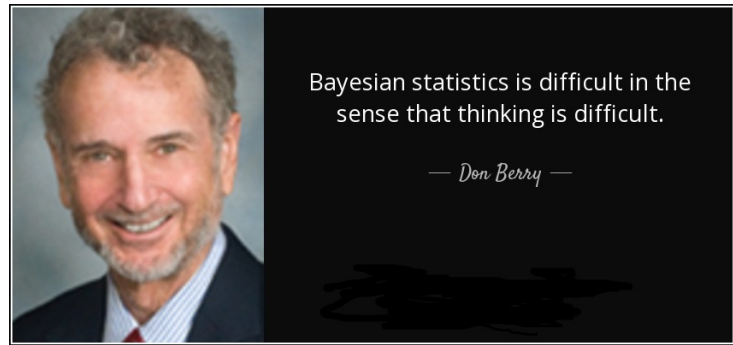
$$\hat{x}_k \stackrel{MAP}{=} \operatorname{argmax}_{x_k} p(x_k|Y_k) \quad (4.69)$$

In the following section the question on how to determine the posteriori conditional probability density function shall be discussed.

4.3.2 Summary

Most engineering problems, and especially in digital signal processing, need to be modeled in a statistical sense. Bayesian theory is applied to statistical inference resulting in a framework where prior information combined with current measurements, where both are treated as random variables, provides an estimation. The posterior density function $p(x_k|Y)$ includes all the information gained from up to the current measurement. However, in most cases these integrals and optimizations have analytical solutions. The Bayesian solution inherent in a recursive framework consists of sequentially updating the posterior density based on new measurements for each sample time. In practice, discrete recursive Bayesian estimation utilizes numerical solutions to high dimensional complex integrations in considerations of the parameters. Moreover, to solve integration problems, the posterior density has to be approximately described so that by successively applying the Bayesian theory a generation of estimate is fulfilled.

5 Probabilistic Inference and Bayesian Recursive Estimation



5.1 Hierarchical Classification of Models

In many engineering and non-engineering disciplines models are generated to mimic the real world behaviors and their parameters into computer systems [367]. General speaking, the more complex the model of the underlying physical system, the better the estimation, of course depending on the algorithm. Therefore, it is important to discuss these models which might be of interest for this work and group them into different classes. First, a general framework will be introduced, from which further common classes are derived and evaluated in regards to estimation problems while this Chapter theory of Bayesian estimation is summarized and further derived from [380][363][384][385][386][387][388][354][355][389][390].

First, the generalized system model is given by the differential-algebraic equation (DAE)

$$\mathcal{F}\left(\frac{dx}{dt}, x(t), u(t), \theta, t\right) = 0 \quad (5.1)$$

where dx/dt represents the differentiation of the internal variable x with respect to time and $\mathbb{R}^{n_x}$, θ with $\mathbb{R}^{n_\theta}$ is the time-invariant parameter vector, w with $\mathbb{R}^{n_w}$ stands for the process noise during time t with $\mathbb{R}^{n_t}$. $u(t)$ represents the input vector used e.g. as control signal. For estimation purposes, the control input signal can be neglected in further discussions. DAE originate from object-oriented modeling, is of stochastic nature and incite the use of this framework to estimate unknown, unobservable internal states and parameters. Nevertheless, the theory and literature focuses on special cases and their

implementations. Due to later complexity, unmodeled dynamics and disturbances influencing the real system, they are only considered in the model equation as an unknown stochastic term. Due to its stochastic characteristics, its implementation and transformation into the corresponding equations bare the risk that the solution also contains noise components. Therefore, some practical degree of freedom regarding the tuneability is important, to control the trade-off between estimation results and noise. Later we will introduce the successful KALMAN filter, whose tuning is realized having a fixed sensor noise covariance matrix while adapting the process noise covariance on-line.

The set of equations is complete by describing the relation between noisy measurements and internal states, this equation is called the measurement equation. The observation samples depend on the measurement system sample frequency and are obtained at discrete time as

$$\mathcal{H}(y_k, x_k, \theta, e_k, k) = 0 \quad (5.2)$$

where y_k with $\mathbb{R}^{n_y}$ represents the discrete measurement, e_k with $\mathbb{R}^{n_e}$, k with $\mathbb{R}$ is the discret time indicator, $\mathcal{H}$ with $\mathbb{R}^{n_x+n_\theta+n_e+1} \rightarrow \mathbb{R}^{n_{\mathcal{H}}}$ stands for the non-linear function for the measurement system.

Classification of Models

Generally, mainly four different models (DAE, ODE, DSSM, DLSSM) are discussed and used in estimation theory [391], and formulated from the original differential-algebraic, for the BMS SW methods, the DSSM model is applied and further derived as following:

Discrete-time Non-linear State-Space-Model with Additive Noise (DSSM)

The discrete-time nonlinear state-space model with additive noise

$$x_{k+1} = \mathcal{F}(x_k, \theta, t) + \kappa_k \quad (5.3)$$

$$y_k = \mathcal{H}(x_k, \theta, \gamma_k, k) + \gamma_k \quad (5.4)$$

where κ_k and γ_k are assumed to be white noises.

It has to be pointed out, that no theory for the discretization of state noise is proposed in the literature, only several different ad hoc assumptions are discussed and can be summarized in [391] An special derivative of the stochastic discrete-time nonlinear state-space model with additive noise uses linear equations which is by far the most studied model cases

5.2 Recursive Bayesian Estimation Theory

In traction battery systems, with up to thousands cells, it is nearly impossible to handle and process the number of measurements, hence, the algorithms are defined in an recursive or iterative manner, also since data is not always available, on-line estimation is crucial. Dynamic systems are defined as discrete processes where the actual state is dependent on prior states. The state dependence and tracking of changes in continuous systems is realized using differential equations. For the discrete case, as such as electronically calculated systems, continuous analog signals are discretized and measurements taken at discrete times, and estimation can only be executed at those time. The differential equations are replaced by finite differences which connect the current state at time t_k to the states at measurement times prior to t_k [372][360] and is defined as first-order Markov processes and only a model for the true transition function. For many engineering problems, the observation and transition function are considered adiabatic, meaning that their dynamics have a higher time constant and slower dynamics making the functions time independent. The posterior density function can be recursively estimated using Bayesian theorem. Following Bayes' law equations are summarized can be rewritten as [380][367]

$$p(x_k|Y_k) = \frac{p(Y_k|x_k)p(x_k)}{p(Y_k)} \quad (5.5)$$

where $p(x_k|Y_k)$ is the pdf of x_k conditioned to the all measurements until the current sample. The set of measurements can be written as individual samples such as

$$p(x_k|Y_k) = \frac{p(y_k, Y_{k-1}|x_k)p(x_k)}{p(y_k, Y_{k-1})} \quad (5.6)$$

Applying the law of the joint density function, this can be written as

$$p(x_k|Y_k) = \frac{p(y_k|Y_{k-1}, x_k)p(Y_{k-1}|x_k)p(x_k)}{p(y_k|Y_{k-1})p(Y_{k-1})} \quad (5.7)$$

Further simplification by applying the law of posterior conditional density and reducing the equations leads to

$$\begin{aligned} p(x_k|Y_k) &= \frac{p(y_k|Y_{k-1}, x_k)p(x_k|Y_{k-1})p(Y_{k-1}|x_k)}{p(y_k|Y_{k-1})p(Y_{k-1})p(x_k)} \\ &= \frac{p(y_k|Y_{k-1}, x_k)p(x_k|Y_{k-1})}{p(Y_k|Y_{k-1})} \end{aligned} \quad (5.8)$$

$$= \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \quad (5.9)$$

A further step is necessary to form a recursive framework for the conditional probability density function. The CHAPMAN–KOLMOGOROV equation relates the prior density

$p(x_k|Y_{k-1})$ and the prior density at time t_{k-1} by

$$\begin{aligned} p(x_k|Y_{k-1}) &= \int_{\mathbb{R}^n} p(x_k|x_{k-1}, Y_{k-1}) p(x_{k-1}|Y_{k-1}) dx_{k-1} \\ &= \int_{\mathbb{R}^n} p(x_k|x_{k-1}) p(x_{k-1}|Y_{k-1}) dx_{k-1} \end{aligned} \quad (5.10)$$

The recursive framework for $p(x_k|Y_{k-1})$ and current posterior $p(x_k|Y_k)$ depends only on the knowledge of the predictive density $p(x_k|x_{k-1})$ and the likelihood function $p(Y_k|x_k)$ [372] [360][380]. The current measurement is used to update the posterior function by the likelihood at t_k

$$p(x_k|Y_k) = \underbrace{\left(\int_{\mathbb{R}^n} p(y_k|x_k) \cdot p(x_k|Y_{k-1}) dx_k \right)^{-1}}_{\text{normalized constant } \Lambda_N} \cdot p(x_k|x_k) \cdot p(x_k|Y_{k-1}) \quad (5.11)$$

with transition and likelihood function as

$$p(x_k|x_{k-1}) = \int_{\mathbb{R}^n} \delta(x_k - \mathcal{F}(x_{k-1}, u_{k-1}, \kappa_{k-1})) \cdot p(\kappa_{k-1}) d\kappa_{k-1} \quad (5.12)$$

respectively,

$$p(y_k|x_k) = \int_{\mathbb{R}^n} \delta(y_k - \mathcal{H}(x_k, u_k, \gamma_k)) \cdot p(\gamma_k) d\gamma_k \quad (5.13)$$

This leads to the optimal estimator, in the sense of statistical inference problem, however, with the restriction of solving the multidimensional integrals only for linear systems with Gaussian noise. The optimal filter for those problems is the linear KALMAN filter (LKF). For all non-linear, non-Gaussian systems, which also lithium-ion batteries belong to, approximations of the recursive Bayesian statistical theorem have to be applied and shall discuss further.

5.3 Approximate Solutions to the Probabilistic Inference Problem

As mentioned earlier, the challenge in solving the probabilistic inference is to estimate internal state variables x and parameters θ based on noisy and missing measurements y . The details are summarized from [372] [390] [380] [392] [367]

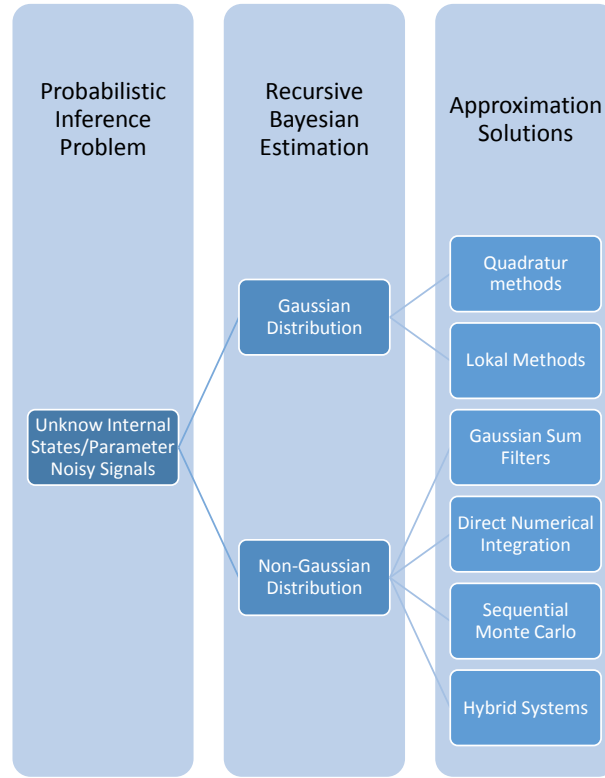


Figure 5.1: Approximation solution for recursive Bayesian estimation [380] [367].

5.3.1 Methods Using Gaussian Approximation

As discussed earlier, the recursive Bayesian framework offers the optimal solution to the non-linear filtering problem. Depending on the problem, the system may consist of multi-dimensional integrals and results in high computational resources for real-time applications executed on embedded systems. Several methods to simplify the integration by approximating are proposed in the literature, such as Sequential Monte Carlo (SMC) [393] [394] Gaussian Closure, Equivalent linearization and Stochastic averaging. SMC with Particle filters transforms the domain of the random variable into a group of limited particles which are passed through the non-linear function to obtain the statistical distribution of the transformed random variable and are therefore the preferred solution for complex non-linear and non-Gaussian system approximations while it has to be taken into consideration that restricting the approximation to the uncertainty propagation, will force a reduction of the number of parameters used.[394] Another method to approximate the conditional pdf is based on the assumption that the stochastic variables are approximated by a sum of Gaussian density functions. The higher the number of Gaussian densities, the better the convergence to any probability function. This leads to the Gaussian Sum Filter (GSF) when considering model case 3, where the first two moments of the Gaussian mixture are recursively passed through the linearized model, under the assumptions, that the new obtained components are equal to the prior weights. This

assumption is reinforced by Bayesian Theorem where new measurements update the approximation of the posteriori probability density function. Generally, the weights of the original GSF are constant, but the uncertainty is passed through the non-linear functions and updated when new observation samples are available. Under the assumption that the random variables are normal distributed, the Bayes filter can be greatly simplifying. Therefore, the filtering problem is limited to determine only the first two moments of the probability distribution [395] best known as the *mean* and its distribution about the central value, the *variance*, which characterizes its variability around that value. Not only the optimal estimate in the minimum mean-squared error (MMSE) sense is given but also any “best” estimator by the conditional mean [367]

$$\hat{x}_{k|k} = \mathbb{E}[x_k|Y_k] \quad (5.14)$$

$$\mathbf{P}_{k|k}^x = \mathbb{E}[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T|Y_k] \quad (5.15)$$

and the evaluation of $\mathbb{E}$, which requires information of the *a posteriori* density $p(x_k|Y_k)$, leading to the “best” estimator. [396] Given the n-dimensional non-linear system with additive noise and uncertain initial conditions given by

$$x_k = \mathcal{F}(x_{k-1}, u_{k-1}, \kappa_{k-1}) \quad (5.16)$$

$$y_k = \mathcal{H}(x_k, u_k, \gamma_k) \quad (5.17)$$

with the normal distributed random variables κ and γ as

$$\kappa_{k-1} = \mathcal{N}(\kappa, \bar{\kappa}_{k-1}, \mathbf{P}_{k-1}^\kappa) \quad (5.18)$$

$$\gamma_k = \mathcal{N}(\gamma, \bar{\gamma}_k, \mathbf{P}_k^\gamma) \quad (5.19)$$

Since the literature only discusses system equations which are reduced as noise is often non-zero-mean, the extension for general stochastic dynamic systems is introduced: The measurement update is done applying Bayes rules:

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \quad (5.20)$$

where

$$p(x_k|Y_{k-1}) = p(x_k|x_{k-1})p(x_{k-1}|Y_{k-1})dx_{k-1} \quad (5.21)$$

and [380]

$$p(y_k|Y_{k-1}) = p(x_k|Y_{k-1})p(y_k|x_k)dx_k \quad (5.22)$$

including the current state density in dependence of the last previous density and the current measurement data which solves the problem of estimating the a-posteriori density

with the help of recursive Bayesian approach as [380],[396]

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \quad (5.23)$$

$$= \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{\int_{\mathbb{R}^n} p(y_k|x_k)p(x_k|Y_{k-1})dx_k} \quad (5.24)$$

To solve this problem, the likelihood function $p(Y_k|x_k)$ as well as the state changes with transition probabilities $p(x|x_{k-1})$ has to be determined by the density functions of the multi-dimensional normal (gaussian) density as [397][380]:

$$p(x|x_{k-1}) = \frac{1}{\Lambda_{N_x}} \cdot \exp \left[-\frac{1}{2} (x_k - \langle \mathcal{F} + K\bar{\kappa} \rangle)^T \{ K\mathbf{P}^K K^T \}^{-1} (x_k - \langle \mathcal{F} + K\bar{\kappa} \rangle) \right] \quad (5.25)$$

$$p(y_k|x_k) = \frac{1}{\Lambda_{N_y}} \cdot \exp \left[-\frac{1}{2} (y_k - \langle \mathcal{H} + \Gamma\bar{\gamma} \rangle)^T \{ \Gamma\mathbf{P}^\gamma \Gamma^T \}^{-1} (y_k - \langle \mathcal{H} + \Gamma\bar{\gamma} \rangle) \right] \quad (5.26)$$

with positive scalars Λ_{N_x} , and Λ_{N_y} with $\int_{\mathbb{R}^m} p(y_k|x_k)dy_k = \int_{\mathbb{R}^n} p(x_k|x_{k-1})dx_k = 1$ with

$$\Lambda_{N_x} = \int_{\mathbb{R}^n} \exp \left[-\frac{1}{2} (x_k - \langle \mathcal{F} + K\bar{\kappa} \rangle)^T \{ K\mathbf{P}^K K^T \}^{-1} (x_k - \langle \mathcal{F} + K\bar{\kappa} \rangle) \right] dx_k \quad (5.27)$$

$$\Lambda_{N_y} = \int_{\mathbb{R}^m} \exp \left[-\frac{1}{2} (y_k - \langle \mathcal{H} + \Gamma\bar{\gamma} \rangle)^T \{ \Gamma\mathbf{P}^\gamma \Gamma^T \}^{-1} (y_k - \langle \mathcal{H} + \Gamma\bar{\gamma} \rangle) \right] dy_k \quad (5.28)$$

According to this, the recursive Bayes rules with initial condition $p(x_0|Y_0)$ can be solved in two steps as following with the analytical solution in the sense of possessing a minimum variance [396]. Therefore, only the conditional mean and covariance needs to be evaluated, and leads to the implemented recursive estimation framework [380]:

1. Prediction step:

$$\begin{aligned} p(x_k|Y_{k-1}) &= \int_{\mathbb{R}^n} p(x_k|x_{k-1}) \underbrace{p(x_{k-1}|Y_{k-1})}_{\mathcal{N}(\hat{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}^x)} dx_{k-1} \\ &= \int_{\mathbb{R}^n} \frac{1}{\Lambda_{N_x}} \cdot \exp \left[-\frac{1}{2} (x_k - \langle f + K\bar{\kappa} \rangle)^T \{ K\mathbf{P}^\kappa K^T \}^{-1} (x_k - \langle f + K\bar{\kappa} \rangle) \right] \\ &\quad \cdot p(x_{k-1}|Y_{k-1}) dx_{k-1} \end{aligned} \quad (5.29)$$

$$\hat{x}_{k|k-1} = \mathbb{E}[x_k|Y_{k-1}] = \int_{\mathbb{R}^n} x_k \int_{\mathbb{R}^m} p(x_k|x_{k-1}) \underbrace{p(x_{k-1}|Y_{k-1})}_{\mathcal{N}(\hat{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}^x)} dx_{k-1} dx_k \quad (5.30)$$

$$\begin{aligned} \mathbf{P}_{k|k}^x &= \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T | Y_{k-1}] \equiv \mathbb{E}[x_k x_k^T] - \hat{x}_{k|k-1} \hat{x}_{k|k-1}^T \\ &= \int_{\mathbb{R}^n} x_k x_k^T \int_{\mathbb{R}^m} p(x_k|x_{k-1}) \underbrace{p(x_{k-1}|Y_{k-1})}_{\mathcal{N}(\hat{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}^x)} dx_{k-1} dx_k - \hat{x}_{k|k-1} \hat{x}_{k|k-1}^T. \end{aligned} \quad (5.31)$$

2. Filtering step:

$$\begin{aligned} p(x_k|Y_k) &= \Lambda_N \cdot p(y_k|x_k) p(x_k|Y_{k-1}) \\ &= \Lambda_N \cdot \frac{1}{\Lambda_{N_y}} \cdot \exp \left[-\frac{1}{2} (y_k - \langle \mathcal{H} + \Gamma \bar{\gamma} \rangle)^T \{ \Gamma \mathbf{P}^\gamma \Gamma^T \}^{-1} (y_k - \langle \mathcal{H} + \Gamma \bar{\gamma} \rangle) \right] \\ &\quad \cdot p(x_k|Y_{k-1}). \end{aligned} \quad (5.32)$$

$$\hat{x}_{k|k} = \mathbb{E}[x_k|Y_k] = \int_{\mathbb{R}^n} x_k \cdot C \cdot p(y_k|x_k) p(x_k|Y_{k-1}) dx_k \quad (5.33)$$

$$\begin{aligned} \mathbf{P}_{k|k}^x &= \mathbb{E}[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T | Y_k] \equiv \mathbb{E}[x_k x_k^T] - \hat{x}_{k|k} \hat{x}_{k|k}^T \\ &= \int_{\mathbb{R}^n} x_k x_k^T \cdot C \cdot p(x_k|x_k) p(x_k|Y_{k-1}) dx_k dx_k - \hat{x}_{k|k} \hat{x}_{k|k}^T. \end{aligned} \quad (5.34)$$

In [397] the integrals can be transformed to

$$\begin{aligned} \hat{x}_k &= (\text{prediction of } x_k) + \mathcal{K}_k [y_k - (\text{prediction of } y_k)] \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + \mathcal{K}_k \cdot (y_k - \hat{y}_{k|k-1}) = \hat{x}_{k|k-1} + \mathcal{K}_k \cdot \tilde{y} \end{aligned} \quad (5.35)$$

$$\mathbf{P}_{k|k}^x = \mathbf{P}_{k|k-1}^x - \mathcal{K}_k \cdot \mathbf{P}_k^{\tilde{y}} \cdot \mathcal{K}_k^T \quad (5.36)$$

and defines the linear recursion, but still the non-linearity of the underlying model is given. The optimal solution in this recursion corresponds to the KALMAN Filter with the gain term

$$\mathcal{K}_k = \mathbf{P}_k^{xy} \cdot (\mathbf{P}_k^{\tilde{y}})^{-1} \quad (5.37)$$

expressed as a function of the posterior covariance matrices, while it is necessary to take expectations of non-linear function of the prior state variable where $\mathbf{P}_{k|k-1}^x$ is the prediction of the covariance of x_k and $\mathbf{P}_k^{\tilde{y}}$ the covariance of $\tilde{y}_k$. These terms are calculated by the KALMAN filter in the linear case.

The most problems occurring in practice, including battery systems too, are of non-linear nature for which no analytical solution to the non-linear recursive solution can be provided. To still apply the Bayesian framework for linear systems to non-linear system

and measurement equations, approximations are required which can be categorized into to different types [380]:

- Local Model Approximations:** The model itself is approximated using a less complex mathematical description of the system dynamics. Multivariate Taylor Series expansions are applied at each time step, to linearize a model about a working point. A discrete-time linear state-space model for first order expansions is obtained, higher orders are also possible but only tend to improve the performance for marginal measurement noise. The non-linear system equation is linearized around the conditional posteriori estimate $\hat{x}_{k-1|k-1}$ at time $k - 1$, while the measurement equation is linearized around the priori estimate $\hat{x}_{k|k-1}$. The KALMAN filter process may be applied to the linearized model, then its known the extended KALMAN filter. The estimation error for the extended KALMAN filter depends on the non-linearities of the underlying process and measurement system. For high non-linearities the linear Taylor-series approximation may result in a poor estimation or negative definite covariance matrices which leads to divergent $\mathcal{K}$. Also, another family is the ensemble KALMAN filter (EnKF), and seen as an alternative to the extended KALMAN filter. The major difference between EKF and EnKF lies within the approximation of non-linearities of the SSM and measurement process. While the EKF applies a linearized error covariance, the EnKF passes around a finite ensemble of model trajectories. As shown in [398], the EKF and EnKF show comparable performance in regards to accuracy and computational effort. Therefore, there was no specific need to do a full scale evaluation, however, the EnKF is implemented for state- and parameter estimation to prove the overall performance.

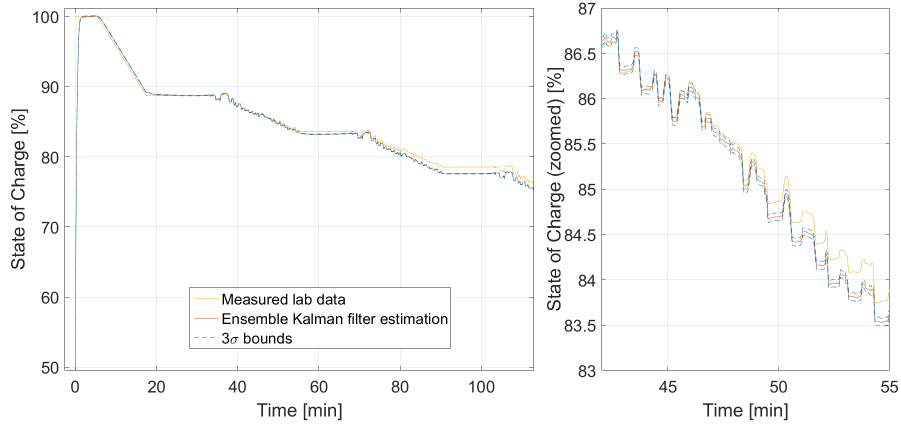


Figure 5.2: State-of-Charge estimation using EnKF. Data provided by [2]

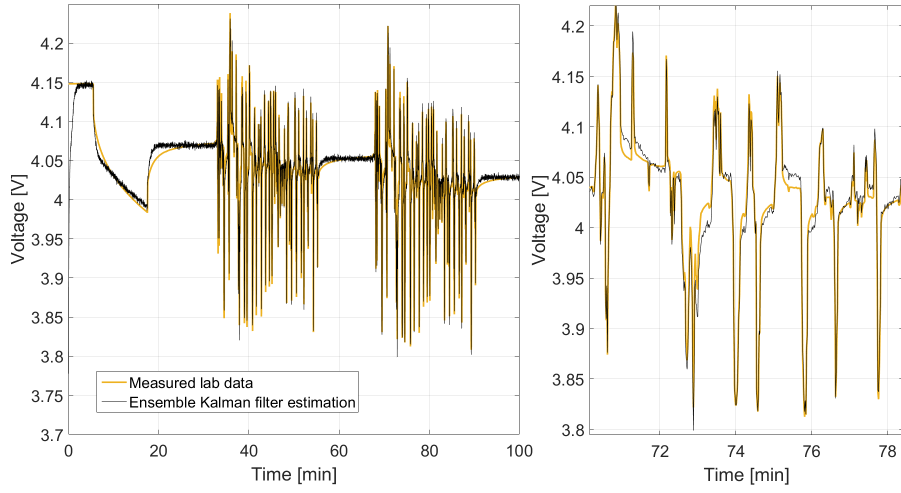


Figure 5.3: Battery cell voltage estimation using EnKF. Data provided by [2]

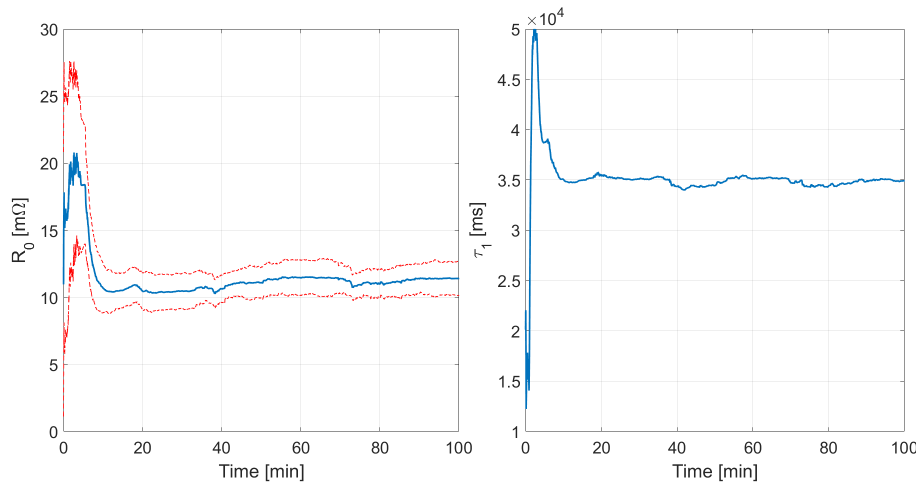


Figure 5.4: Parameter estimation of 1-RC ECM using EnKF. Data provided by [2]

Figure 5.5: Performing joint estimation for first order Li-ion battery ECM using ensemble KALMAN filter.

Following methods address the EKF weaknesses:

Iterated Extended Kalman Filter v1 To advance the "first-order" approximation the iterated EKF was proposed, which improves the convergence behavior of EKF especially for weak initial values and high non-linearities in the measurement equation. At time step k the nominal state estimate is redefined while the measurement equation is re-linearized around the updated state estimate [396]. The prediction works similar to the EKF, but based on the first filter value, primarily the equation is linearized around the predictive state estimate followed by measurement update. Subsequently, the measurement equation is linearized around the newly obtained estimate. This process is iteratively repeated until the absolute difference between two consecutive iteration results fall below the user defined threshold. This advancement does not effect considerably the computation efficiency even though outperforms the standard extended KALMAN filter when non-linear measurements models exist [399].

Iterated Extended Kalman Filter v2 The second version of the iterated extended KALMAN filter goes one step further by minimizing the linearization errors of the system model, thus the corresponding prediction equations are linearized as well. As soon as new measurements are available, the priori state estimation $\hat{x}_{k|k-1}$ as well as the appropriate covariance $\mathbf{P}_{k|k-1}^x$ are corrected iteratively.

Adaptive Iterated Extended Kalman Filter The EKF ignores higher-order terms introducing truncated errors, while the IEKF was designed to overcome that drawback. Additionally, for process noise vector and measurement noise vector, the covariances are described with uncertainty and assumptions, for what the adaptive IEKF was designed to account with a noise statistics estimator.

Additional simplifications focus on the additive noise processes, these are approximated by Gaussian distributions. The model is now linearized with additive Gaussian system and measurement noise. As proposed by Rudolf Kalman, the extended KALMAN filter is the optimal estimator for this problem, which means, the estimator processes observations to compute a minimum error estimate of the state or parameters using the system and measurement equations with its additive noise and initial conditions provided by the designer. On the other side, hidden Markov models (HMM) approximate the non-linearity by keeping the state vector constant within a finite set of values. Nevertheless, for model approximations, the discrete-time Bayesian recursive estimation approach is neglected.

- **Global A-Posteriori-Density Approximations - Non-working-point Linearization** Whenever the local approximations fail or suffer from high level of inaccuracy due to high system non-linearity, bad initial estimates or non-Gaussian noise, the powerful method of Bayesian recursive estimation derived earlier is applied to the original non-linear models, which are based on physical principles to describe the systems behavior, to approximate ordinary differential equations by

solving Bayesian theorem recursively in numerical manners. Many different methods are discussed in the literature and are summarized in Figure 5.1 and can be categorized into 2 groups, where group one uses the parameterized posterior density over a finite values for the parameters and used as inputs to the Bayesian framework, which will be adapted recursively over iterations.

To bypass the linearization process several algorithms are proposed in the literature to increase the effectiveness of the filtering scheme over the original linearization methods such as extended KALMAN filter. The so called Bayesian quadrature is a means of using Bayesian inference with model-based approximation to the value of a potentially intractable function. For a function $\mathcal{F}$ the prior distribution over its values is defined by the mean and covariance matrix as following [400][380]

$$I = \int_{\mathbb{R}^n} F \cdot \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}^x|}} \cdot \exp \left[-\frac{1}{2} (x - \bar{x})^T (\mathbf{P}^x)^{-1} (x - \bar{x}) \right] dx \quad (5.38)$$

where I the context, consists of prior information of mean and covariance functions.

$$I = \int_{\mathbb{R}^n} \tilde{F}(b) \cdot \frac{1}{(2\pi)^{n/2}} \cdot \exp \left(-\frac{1}{2} b^T b \right) db \quad (5.39)$$

with $\mathbf{P}^x = \mathbf{S}^x (\mathbf{S})^T$ and $x = \mathbf{S}^x b + \bar{x}$ which is approximated with different quadrature methods. Later various filtering techniques were introduced under Bayesian framework, such as Central Difference Filter (CDF), Unscented KALMAN Filter (UKF), Cubature KALMAN Filter (CKF), Cubature Quadrature KALMAN Filter (CQKF), Gauss-Hermite Filter (GHF), Sparse Grid Gauss-Hermite Filter (SGHF) and Multiple Square Root Quadrature KALMAN Filter (MSQKF).

1. **Quadrature: Julier & Uhlmann** JULIER & UHLMANN define a new method where a normal Gaussian distribution of a random variable x is approximated by a discrete distribution as [390][380]

$$x^{(i)} = \begin{cases} \sqrt{n + \kappa} e^{(i)} & \text{for } 1 \leq i \leq n \\ -x^{(i)-n} & \text{for } n+1 \leq i \leq 2n \\ 0 & \text{for } i = 2n+1 \end{cases} \quad (5.40)$$

$$p(x^{(i)}) = \begin{cases} \frac{2\kappa}{2(n+\kappa)} & \text{for } i = 2n+1 \\ \frac{1}{2(n+\kappa)} & \text{for } 1 \leq i \leq 2n \end{cases} \quad (5.41)$$

where $\kappa > 0$ represents a constant and $e^{(i)}$ the i th unity vector in $\mathbb{R}^n$ and transform the integral equation into JULIER & UHLMANN approximation [380]

$$I \equiv I_{JU} = \sum_{i=1}^{2n+1} \tilde{F}(x^{(i)}) p(x^{(i)}) \quad (5.42)$$

This filtering method portrayals draws on the Monte Carlo method, with the

idea that its more complex to approximate the arbitrary non-linear function rather than approximating a probability distribution. In comparison to MC, the sampling is not random and therefore the computation is more efficient due to reduced post-processing. Deterministic sample points, called sigma points, which represent the pdf of a multi-dimensional random variable x , are drawn to capture the first two moments of the prior distribution. Each sample is propagated through the non-linear function $y = g(x)$ and transformed via $\Upsilon^{(i)} = g(\mathcal{X}^{(i)})$, and weighted based on the original mean and covariance where afterwards a new Gaussian distribution is fitted to these new transformed weighted points, according to updated mean and covariance matrix. This transformation is known as the *unscented transformation*, and lends its name to the *Unscented KALMAN Filter* (UKF). Since the transformation propagates the mean and covariance precise to the second order of the distribution, the filter lacks on accuracy when applied to non-Gaussian distributions. The matrix $\mathcal{X}$ consists of $2n + 1$ generated sigma points such as [390]

$$\mathcal{X}^{(i)} = \begin{cases} \bar{x}, & \text{for } i = 1 \\ \bar{x} + (\sqrt{(n + \kappa)\mathbf{P}^x})^{(i)}, & \text{for } i = 2, \dots, n + 1 \\ \bar{x} - (\sqrt{(n + \kappa)\mathbf{P}^x})^{(i)}, & \text{for } i = n + 2, \dots, 2n + 1 \end{cases} \quad (5.43)$$

and the following weights $w^{(i)}$:

$$\begin{aligned} w^{(1)} &= \frac{\kappa}{m + \kappa} \\ w^{(i)} &= \frac{1}{2(m + \kappa)} \quad \text{for } i = 2, \dots, 2n + 1. \\ \sum_{i=1}^{2n+1} w^{(i)} &= 1 \end{aligned}$$

with $w^{(i)}$ represents the i -th sigma-point vector, κ is a scaling parameter, $(\sqrt{(n + \kappa)\mathbf{P}^x})^{(i)}$ represents the i th vector of the Cholesky factorization of the weighted covariance matrix. The approximated statistic by means of mean, covariance and cross-covariance are calculated as following [390][380]

$$\bar{y} \approx \sum_{i=1}^{2n+1} w^{(i)} \cdot \Upsilon^{(i)} \quad (5.44)$$

$$\mathbf{P}^y \approx \sum_{i=1}^{2n+1} w^{(i)} \cdot (\Upsilon^{(i)} - \bar{y})(\Upsilon^{(i)} - \bar{y})^T \quad (5.45)$$

$$\mathbf{P}^{xy} \approx \sum_{i=1}^{2n+1} w^{(i)} \cdot (\mathcal{X}^{(i)} - \bar{x})(\mathcal{X}^{(i)} - \bar{y})^T \quad (5.46)$$

Thus, the unscented transform was defined and introduced by JULIER & UHLMANN and further improved the sigma point generation to exploit additional stochastic information including the first three moments of the distri-

bution of a random variable. All these methods face the challenge for higher state space dimensions since the spherical bounds proportions increase with $2^{n-1/2}$ and result in taking non-local effects into account. The sigma point scaling approach (Scaled Unscented Transform (SUT)) [401] overcomes the transformation skewing by propagating a set of scaled sigma points of the form

$$\mathcal{X}^{(i)} = \mathcal{X}^{(0)} + \alpha(\mathcal{X}^{(i)} - \mathcal{X}^{(0)}) \quad (5.47)$$

with α represents the scaling parameter within the interval $0 \leq \alpha \leq 1$ to tune higher order effects and allows an additional degree of freedom to describe further moments. The new set of sigma-point vectors is defined below and influences the fitting in the fourth and higher order moments of symmetric probability functions by means of a additional scaling factor α .

$$\mathcal{X}^{(i)} = \begin{cases} \bar{x}_h, & \text{for } i = 1 \\ \bar{x} + (\sqrt{(n+\lambda)\mathbf{P}^x})^{(i)}, & \text{for } i = 2, \dots, n+1 \\ \bar{x} - (\sqrt{(n+\lambda)\mathbf{P}^x})^{(i)}, & \text{for } i = n+2, \dots, 2n+1 \end{cases} \quad (5.48)$$

and the new weighting factors of this transformed sequence are

$$\begin{aligned} w^{(1),m} &= \frac{\lambda}{n+\lambda} \\ w^{(1),c} &= \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta \\ w^{(i),m} &= w^{(i),c} = \frac{1}{2(n+\lambda)} \quad \text{for } i = 2, \dots, 2n+1. \end{aligned}$$

The scaled sigma-points are propagated through the non-linear function as $\Upsilon^{(i)} = g(\mathcal{X}^{(i)})$ and the approximated stochastic consisting of mean and covariance is [390][380]

$$\bar{y} \approx \sum_{i=1}^{2n+1} w^{(i),m} \cdot \Upsilon^{(i)} \quad (5.49)$$

$$\mathbf{P}^y \approx \sum_{i=1}^{2n+1} w^{(i),c} \cdot (\Upsilon^{(i)} - \bar{y})(\Upsilon^{(i)} - \bar{y})^T \quad (5.50)$$

$$\mathbf{P}^{xy} \approx \sum_{i=1}^{2n+1} w^{(i),c} \cdot (\mathcal{X}^{(i)} - \bar{x})(\mathcal{Y}^{(i)} - \bar{y})^T \quad (5.51)$$

Based on the (scaled) unscented transformation equations, the so called Unscented KALMAN Filter is derived [402].

2. Finite Difference Approximation (Central Difference Kalman Filter)

The idea is based on the polynomial interpolation methods, where the given

function F is approximated by a quadratic function that satisfies [402]

$$I = \int_{\mathbb{R}^n} F \cdot \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}^x|}} \cdot \exp \left[-\frac{1}{2} (x - \bar{x})^T (\mathbf{P}^x)^{-1} (x - \bar{x}) \right] dx \quad (5.52)$$

$$\tilde{F}(\xi^{(i)}) = \mathcal{P}(\xi^{(i)}) \quad \text{for } 1 \leq i \leq \frac{(n+1)(n+2)}{2} \quad (5.53)$$

at the points $\xi^{(i)}$ with the step size $h > 0$

$$\xi^{(i)} = \begin{cases} \langle 0, he^{(i)}, -he^{(i)} \rangle & \text{for } 1 \leq i \leq n \\ he^{(i)} + he^{(j)} & \text{for } 1 \leq i \leq j \leq n \end{cases}$$

which allows the transformation of quadratic function $\mathcal{P}$

$$\mathcal{P}_2(s) = \tilde{F}(0) + \sum_{i=1}^n a^{(i)} s^{(i)} + \frac{1}{2} s^T \mathbf{H} s \quad (5.54)$$

where $s^{(i)}$ is the i -th coordinate of a point s and the vector $a = a^{(i)}$ and the symmetric matrix $\mathbf{H} = (\mathbf{H}^{(i),(j)})_{n \times n}$ [400][380].

$$a^{(i)} = \frac{\tilde{F}(he^{(i)}) - \tilde{F}(-he^{(i)})}{2h}, \quad \text{for } 1 \leq i \leq n \quad (5.55)$$

$$\mathbf{H}^{(i),(j)} = \frac{\tilde{F}(he^{(i)}) - 2\tilde{F}(0) + \tilde{F}(-he^{(i)})}{h^2} \quad \text{for } 1 \leq i \leq n \quad (5.56)$$

$$\mathbf{H}^{(i),(j)} = \frac{\tilde{F}(he^{(i)} + he^{(j)}) - \tilde{F}(-he^{(i)}) - \tilde{F}(-he^{(j)}) + \tilde{F}(0)}{h^2} \quad \text{for } 1 \leq i \leq n \quad (5.57)$$

where the substitution of the stochastic recursive Bayesian integral I is approximated [380]

$$I \approx I_{\text{FDI}} = \tilde{F}(0) + \sum_{i=1}^n \frac{1}{2} \mathbf{H}^{(i),(j)} \quad (5.58)$$

and the integral as

$$J = \int_{\mathbb{R}^n} \tilde{F}_1(b) \tilde{F}_2(b) \cdot \frac{1}{(2\pi)^{n/2}} \cdot \exp \left(-\frac{1}{2} b^T b \right) db \quad (5.59)$$

while J further needs to be approximated by

$$\begin{aligned}
 J &\approx \int_{\mathbb{R}^n} \mathcal{P}_2^{\{1\}}(b) \mathcal{P}_2^{\{2\}}(b) \cdot \frac{1}{(2\pi)^{n/2}} \cdot \exp\left(-\frac{1}{2}b^T b\right) db \\
 &= I_{\text{FDI}} \\
 &= I_{\text{FDI}}^{\{1\}} I_{\text{FDI}}^{\{2\}} + \sum_{i=1}^n \left(a^{(i),\{1\}} a^{(i),\{2\}} + \frac{1}{2} \mathbf{H}^{(i),(i),\{1\}} \mathbf{H}^{(i),(i),\{2\}} \right) \\
 &\quad + \sum_{i \neq j} \mathbf{H}^{(i),(j),\{1\}} \mathbf{H}^{(i),(j),\{2\}}
 \end{aligned} \tag{5.60}$$

where $\mathcal{P}^{\{i\}}$ represents the quadratic polynomial approximation of $\tilde{F}_i$ when $i = 1, 2$. Also possible using the diagonal second order correction $\frac{1}{2} \mathbf{H}^{(i),(j)}$, which results in

$$\begin{aligned}
 J &\approx \int_{\mathbb{R}^n} \mathcal{P}_1^{\{1\}}(b) \mathcal{P}_1^{\{2\}}(b) \cdot \frac{1}{(2\pi)^{n/2}} \cdot \exp\left(-\frac{1}{2}b^T b\right) db \\
 &= \tilde{J}_{\text{FDI}} = \tilde{F}_1(0) \tilde{F}_1(0) + \sum_{i=1}^n \frac{\tilde{F}_1(h e^{(i)}) - \tilde{F}_1(-h e^{(i)})}{2h} \frac{\tilde{F}_2(h e^{(i)}) - \tilde{F}_2(-h e^{(i)})}{2h} \\
 &\quad + \sum_{i=1}^n \frac{1}{2} \mathbf{H}^{(i),(i),\{1\}} \mathbf{H}^{(i),(i),\{2\}}
 \end{aligned} \tag{5.61}$$

Now, the integrals I and J can now be approximated. It is worth mention, by neglecting the second-order correction term the central difference approach may be used for the Jacobian matrix approximation of $\tilde{F}$ for the EKF. The procedure of finding Central Difference KALMAN Filter goes analog to the findings of the UKF, namely

$$\mathcal{X}^{(i)} = \begin{cases} \bar{x}, & \text{for } i = 1 \\ \bar{x} + (h \sqrt{\mathbf{P}^x})^{(i)}, & \text{for } i = 2, \dots, n+1 \\ \bar{x} - (h \sqrt{\mathbf{P}^x})^{(i)}, & \text{for } i = n+2, \dots, 2n+1 \end{cases} \tag{5.62}$$

and the weighting factor

$$\begin{aligned}
 w^{(1),m} &= \frac{h^2-2}{h^2} \\
 w^{(1),m} &= \frac{1}{2h^2} \quad \text{for } i = 2, \dots, 2n+1. \\
 w^{(1),c1} &= \frac{1}{4h^2} \quad \text{for } i = 2, \dots, 2n+1. \\
 w^{(i),c2} &= \frac{h^2-1}{4h^4} \quad \text{for } i = 2, \dots, 2n+1.
 \end{aligned}$$

The approximated moments such as mean, covariance and cross-covariance from $y = g(x)$ are obtained by propagating sigma-point vectors through the non-linear function g with $\Upsilon^{(i)} = g(\mathcal{X}^{(i)})$, similar at the unscented KALMAN

filter approximation

$$\bar{y} \approx \sum_{i=1}^{2n+1} w^{(i),m} \cdot \Upsilon^{(i)} \quad (5.63)$$

$$\mathbf{P}^y \approx \sum_{i=1}^{2n+1} \left[w^{(i),c1} \cdot (\Upsilon^{(i)} - \Upsilon^{(n+i)})^2 + w^{(i),c2} (\Upsilon^{(i)} + \Upsilon^{(n+i)} - 2\Upsilon^{(1)})^2 \right] \quad (5.64)$$

$$\mathbf{P}^{xy} \approx \sqrt{w^{(2),c1} \mathbf{P}^x} \left[\Upsilon^{(2:n+i)} - \Upsilon^{(n+2:2n+1)} \right]. \quad (5.65)$$

The squaring describes the outer vector product such as $(s)^2 = ss^T$. The CDK and UKF are in their structure similar, only differ in the representation of the sigma points and weights. The square-root form is derived in the following.

3. **The square-root central difference Kalman filter** In the same way we optimize the UKF, there is a way to optimize the CDKF to a more efficient way to execute, by means of square-root. Similar, the heavy calculation lies within the square-root covariance matrix for all sigma-points. The work-around is to use the Cholesky factorization of the covariance matrices. In addition, optimized least-squares, QR-decomposition and Cholesky factorization is applied. The filter starts with its initialization as

$$\hat{x}_0^+ = \mathbb{E}[x_0] \quad (5.66)$$

$$\mathbf{S}_{x_0}^+ = \text{CHOLESKY}\{\mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]\} \quad (5.67)$$

$$\mathbf{S}_Q = (\sqrt{\mathbf{Q}})^T = \text{CHOLESKY}(\mathbf{Q}) \quad (5.68)$$

$$\mathbf{S}_R = (\sqrt{\mathbf{R}})^T = \text{CHOLESKY}(\mathbf{R}) \quad (5.69)$$

followed by the time-update equations with little changes to the original central difference KALMAN filter

$$\hat{X}_{k-1}^+ = [\hat{x}_{k-1}^+, \dots, \hat{x}_{k-1}^+], \hat{X}_{k-1}^+ \in \mathbb{R}^{N \times N} \quad (5.70)$$

$$\mathcal{X}_{k-1}^+ = \begin{bmatrix} \hat{x}_{k-1}^+ & \hat{X}_{k-1}^+ + h(\mathbf{S}_{x_{k-1}}^+)^T & \hat{X}_{k-1}^+ - h(\mathbf{S}_{x_{k-1}}^+)^T \end{bmatrix} \quad (5.71)$$

$$^* \mathcal{X}_k^- = F(\mathcal{X}_{k-1}^+, u_{k-1}) \quad (5.72)$$

$$\hat{x}_k^- = \sum_{i=0}^{2N-1} \omega_i^{(m)} (^* \mathcal{X}_k^-)_i \quad (5.73)$$

The next step requires further modification from the original CDKF equations, by writing the Cholesky factorization of the covariance matrix in QR-form,

resulting in the upper Cholesky factor $\mathbf{S}_{x_k}^-$

$$\mathbf{S}_{x_k}^- = \text{qr} \left\{ \begin{bmatrix} \sqrt{\omega_1^{(c1)}} ((\mathcal{X}_k^-)_{1:N-1} - (\mathcal{X}_k^-)_{N+1:2N-1}), \\ \sqrt{\omega_1^{(c2)}} \left((\mathcal{X}_k^-)_{1:N-1} + (\mathcal{X}_k^-)_{N+1:2N-1} - 2[(\mathcal{X}_k^-)_0 \cdot^{\times N-1} \cdot] \right), \mathbf{S}_Q^T \end{bmatrix}^T \right\} \quad (5.74)$$

resulting in adding $N - 1$ times the 0^{th} -column of sigma-points as following:

$$\left[(\mathcal{X}_k^-)_0 \cdot^{\times N-1} \cdot \right] = [(\mathcal{X}_k^-)_0, \dots, (\mathcal{X}_k^-)_0], \quad \left[(\mathcal{X}_k^-)_0 \cdot^{\times N-1} \cdot \right] \in \mathbb{R}^{N \times N-1} \quad (5.75)$$

The measurement equations follows the CDKF implementation,

$$\hat{X}_k^- = [\hat{x}_k^-, \dots, \hat{x}_k^-], \quad \hat{X}_k^- \in \mathbb{R}^{N \times N} \quad (5.76)$$

$$\mathcal{X}_k^- = \begin{bmatrix} \hat{x}_k^- & \hat{X}_k^- + h(\mathbf{S}_{x_k}^-)^T & \hat{X}_k^- - h(\mathbf{S}_{x_k}^-)^T \end{bmatrix} \quad (5.77)$$

$$\mathcal{Y}_k = h(\mathcal{X}_k^-, u_k) \quad (5.78)$$

$$\hat{y}_k = \sum_{i=0}^{2N-1} \omega_i^{(m)} (\mathcal{Y}_k)_i \quad (5.79)$$

$$\mathbf{S}_{y_k} = \text{qr} \left\{ \begin{bmatrix} \sqrt{\omega_1^{(c1)}} ((\mathcal{Y}_k)_{1:N-1} - (\mathcal{Y}_k)_{N+1:2N-1}), \\ \sqrt{\omega_1^{(c2)}} \left((\mathcal{Y}_k)_{1:N-1} + (\mathcal{Y}_k)_{N+1:2N-1} - 2[(\mathcal{Y}_k)_0 \cdot^{\times N-1} \cdot] \right), \mathbf{S}_R^T \end{bmatrix}^T \right\} \quad (5.80)$$

with

$$\left[(\mathcal{Y}_k)_0 \cdot^{\times N-1} \cdot \right] = [(\mathcal{Y}_k)_0, \dots, (\mathcal{Y}_k)_0], \quad \left[(\mathcal{Y}_k)_0 \cdot^{\times N-1} \cdot \right] \in \mathbb{R}^{r \times N-1} \quad (5.81)$$

$$\mathbf{P}_{x_k, y_k} = \sqrt{\omega_1^{(c1)}} \mathbf{S}_{x_k}^{-T} [\mathcal{Y}_{1:N-1} - \mathcal{Y}_{N+1:2N-1}]^T \quad (5.82)$$

$$\mathcal{K}_k = (\mathbf{P}_{x_k, y_k} / \mathbf{S}_{y_k}) / \mathbf{S}_{y_k}^T \quad (5.83)$$

$$\hat{x}_k^+ = \hat{x}_k^- + \mathcal{K}_k (y_k - \hat{y}_k^-) \quad (5.84)$$

$$\mathbf{D}_C = \mathcal{K}_k \mathbf{S}_{y_k}^T \quad (5.85)$$

$$\mathbf{S}_{x_k}^+ = \text{CHOLESKY-UPDATE}(\mathbf{S}_{x_k}^-, d_i, ' - '), \quad i = 0, \dots, r-1 \quad (5.86)$$

with d_i is the i^{th} -column of the Cholesky downdate matrix $\mathbf{D}_C$ applied to $\mathbf{S}_{x_k}^-$

4. **Gaussian–Hermite Quadrature** [402][400] rule is given for the scalar case as

$$\int_{\mathbb{R}^n} g(x) \frac{1}{(2\pi)^{1/2}} \exp(-x^2) dx = \sum_{i=1}^q w^{(i)} g(x^{(1)}) \quad (5.87)$$

with the quadrature points x_i , whereas the equation for the polynomials $g(x)$ holds up to $2q - 1$. $\mathbf{J}$ represents a symmetric, tridiagonal matrix with zero diagonals and $\mathbf{J}^{(i),(i+1)} = \sqrt{i/2}, 1 \leq q - 1$:

$$\mathbf{J} = \begin{bmatrix} 0 & \sqrt{i/2} & & & \\ \sqrt{i/2} & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \sqrt{(q-1)/2} & \sqrt{(q-1)/2} & \\ & & & 0 & \end{bmatrix} \quad (5.88)$$

It follows that the quadrature points x_i obtained from the eigenvalues $\lambda^{(i)}$ of $\mathbf{J}$ and $w^{(i)} = |(v^{(i)})_1|^2$, where $(v^{(i)})_1$ is the first element of the i -th normalized $\lambda^{(i)}$ of $\mathbf{J}$. The normalization is

$$(v^{(i)})^T (v^{(i)}) = \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{1/2}} \exp(-x^2) dx = \sqrt{\pi} \quad (5.89)$$

Considering the one-dimensional case

$$I = \int_{\mathbb{R}^n} f(x) \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{1}{2}\left(\frac{x - \bar{x}}{\sigma}\right)^2\right] dx = \int_{\mathbb{R}^n} \tilde{f}(x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}b^2\right) db \quad (5.90)$$

$$= \int_{\mathbb{R}^n} g(b) \exp(-b^2) db \quad (5.91)$$

$$g(b) = \tilde{f}(x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}b^2\right) db \quad (5.92)$$

$$x = \sigma b + \bar{x} \quad \text{with,} \quad dx = \sigma db \quad (5.93)$$

$$I = \sum_{i=0}^q w^{(i)} g(\lambda^{(i)}) \quad (5.94)$$

This results can be extended to the multi-dimensional case such as

$$I \approx I_{GH} = \sum_{i_1=1}^q \cdots \sum_{i_n=1}^q \tilde{G}(\lambda^{(i_1)}, \lambda^{(i_2)}, \dots, \lambda^{(i_n)}) w^{(i_1)} w^{(i_2)} \cdots w^{(i_n)} \quad (5.95)$$

$$\tilde{G} = \tilde{F}(b) \frac{1}{\sqrt{2\pi}} \exp\left(\frac{1}{2}b^T b\right) \quad (5.96)$$

The Gaussian–Hermite Quadrature rule is applied to design the Gauss-Hermite filter for non-linear Gaussian models. The Gauss-Hermite filter is similar to the Central Difference KALMAN filter and Unscented KALMAN filter, also based on generating a set of sigma-points. The compared algorithms share the same structure, the difference lies within the representation through sigma-point vectors and weights of the probabilities.

- **Sigma Point Kalman Filters** The local model approximation out of which the extended KALMAN filter is the widely accepted standard non-linear estimator, prove especially for high dynamic, non-linear systems deficiency, such as limited "first-order" accuracy of iterated means and covariances due to reduced Taylor series expansion during linearization in the operating point. Moreover, local approximation methods don't take into account the stochastic properties of the corresponding system state and noise during the linearisation. This leads to reduced accuracy, consistency and efficiency of these kind of methods. Advanced algorithms such as the UKF are based on weighted stochastic linear regression, which are collectively called **Sigma-point Kalman filters** address this shortcomings by using deterministic sampling methods, which makes them derivative-free but following still the KALMAN update formulation [390][403][404]. In this work, it is also of interest to shed light on the issue of computational efficiency, based on the motivation to find the appropriate estimator for battery monitoring systems. For the group of Sigma-point KALMAN filters the operation counts in terms of function evaluations can be summarized as follows:

In theory, the algorithm based on Gauss-Hermite rule performs better in the sense of accuracy. For GHF filter, the Gaussian integrals are as far as possible calculated while the UKF and CDKF approximately calculate the Bayes integrals. This results in an strong increase of computational requirements for high-dimensional systems and makes the GHF only partly suitable and can only employed to a restricted degree.

- **Iterated Sigma-point Kalman Filter** In analogy to the Iterated Extended KALMAN filter, iterated Sigma-point KALMAN filter were derived to improve the filter properties. One hybrid method follows the combination of the UKF (prediction step), and the IEKF₁. The IUKF is based using iterated stochastic linearization, which was also implemented for the central difference approximation to ICDKF.
- **Square-root Forms** [405] Matrix square-root calculations for the state covariance to generate the sigma-point vectors may be computational too costly for battery monitoring algorithms, which are executed on embedded controllers which offer limited computing power [396]. To address the need for higher numerical stability and efficiency (guarantees positive semi-definiteness) during the state covariance update the square-root forms for both, the Central Difference KALMAN filter and Unscented KALMAN filter. The most demanding part of the UKF lies within the calculation of the matrix square root of the covariance for computing the sigma-points at each sample time, which means that the state covariance is directly passed and updated in Cholesky factored form $\mathbf{S}$ together with *QR decomposition*, *Cholesky factor updating* and *efficient pivot-based least squares*. To understand the SR-UKF, the equations shall be recalled with emphasis on linear algebra: The Cholesky decomposition can be described as the computation of the matrix square-root of the

positive definite covariance matrix $\mathbf{P}$ which results in an upper triangular matrix $\mathbf{S}$

$$\mathbf{P} = \mathbf{S}^T \mathbf{S} \quad (5.97)$$

However, the more efficient decomposition in the form

$$\mathbf{X} = \mathbf{Q}\mathbf{R} \quad (5.98)$$

is used, where $\mathbf{Q} \in \mathbf{R}^{L \times L}$ is orthonormal and $\mathbf{R} \in \mathbf{R}^{L \times N}$ is an upper triangular. Then the covariance matrix can be calculated as

$$\mathbf{P} = \mathbf{X}^T \mathbf{X} = \mathbf{R}^T \mathbf{Q}^T \mathbf{Q} \mathbf{R} \quad (5.99)$$

resulting in

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{I} \quad (5.100)$$

due to the orthonormality of $\mathbf{Q}$

$$\mathbf{P} = \mathbf{R}^T \mathbf{R} = \tilde{\mathbf{R}}^T \tilde{\mathbf{R}} \quad (5.101)$$

where $\tilde{\mathbf{R}}$ is defined as the upper triangular part of $\mathbf{R}$ resulting in the upper Cholesky factor of $\mathbf{P}$. The Cholesky factor update and $\tilde{\mathbf{P}}$ are defined as

$$\tilde{\mathbf{P}} = \mathbf{P} + v \cdot \mathbf{x}\mathbf{x}^T \quad (5.102)$$

where v is a scalar, $\tilde{\mathbf{S}}$ the upper triangular Cholesky factor of $\tilde{\mathbf{P}}$ and is calculated as following

$$\tilde{\mathbf{S}} = \text{CHOLESKY}(\tilde{\mathbf{P}}) = \text{CHOLESKY-UPDATE}(\mathbf{S}, \sqrt{v} \cdot \mathbf{x}) \quad (5.103)$$

In comparison the the UKF, the initialization considers the Cholesky factors of the covariance and noise matrices as [405]

$$\hat{x}_0^+ = \mathbb{E}[x_0] \quad (5.104)$$

$$\mathbf{S}_0^+ = \text{CHOLESKY}\{\mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]\} \quad (5.105)$$

$$\mathbf{S}_Q = \left(\sqrt{\mathbf{Q}}\right)^T = \text{CHOLESKY}(\mathbf{Q}) \quad (5.106)$$

$$\mathbf{S}_R = \left(\sqrt{\mathbf{R}}\right)^T = \text{CHOLESKY}(\mathbf{R}) \quad (5.107)$$

The sigma-points for the *a posteriori* distribution at time step $k - 1$ is defined via

$$\mathcal{X}_{k-1}^+ = [\hat{x}_{k-1}^+ \quad \hat{X}_{k-1}^+ + \eta(\mathbf{S}_{x_{k-1}}^+)^T \quad \hat{X}_{k-1}^+ - \eta(\mathbf{S}_{x_{k-1}}^+)^T] \quad (5.108)$$

where η defines the spread of the sigma-points, and passed through the non-linear system equation to compute the *a priori* value for the time step k with the estimated

state

$${}^*\mathcal{X}_k^- = f(\mathcal{X}_{k-1}^+, u_{k-1}) \quad (5.109)$$

$$\hat{x}_k^- = \sum_{i=0}^{2N-1} \omega_i^{(m)} ({}^*\mathcal{X}_k^-)_i \quad (5.110)$$

$$(5.111)$$

for which the Cholesky factor is propagated as [405]

$$\mathbf{S}_{x_k}^- = \text{qr} \left\{ \left[\sqrt{\omega_1^{(c)}} \cdot \left(({}^*\mathcal{X}_k^-)_{1:2N-1} - \hat{X}_{k_{sp}}^- \right) \quad \mathbf{S}_Q^T \right]^T \right\} \quad (5.112)$$

$$\mathbf{S}_{x_k}^- = \text{CHOLESKY-UPDATE} \left\{ \mathbf{S}_{x_k}^-, \sqrt{\omega_0^{(c)}} \cdot \left(({}^*\mathcal{X}_k^-)_0 - \hat{x}_k^- \right) \right\} \quad (5.113)$$

Beginning, the QR decomposition is used at $\mathbf{S}_{x_k}^-$ containing the weighted change of sigma-points and the average of the square-root of the process noise covariance matrix. Following, the Cholesky factor update is applied for stability reason as the 0-th sigma-points are taking into account as well. This calculation leads to a similar matrix as represented in 5.64,. The similarity between UKF and its square-root form SRUKF is illustrated for the change of covariance during time as following [405]

$$M = \left[\sqrt{\omega_1^{(c)}} \cdot \left(({}^*\mathcal{X}_k^-)_{1:2N-1} - \hat{X}_{k_{sp}}^- \right) \quad \mathbf{S}_Q^T \right]^T \quad (5.114)$$

$${}^*\mathbf{P}_{x_k}^- = (\mathbf{S}_{x_k}^-)^T \mathbf{S}_{x_k}^- = M^T M \quad (5.115)$$

$$= \left[\sqrt{\omega_1^{(c)}} \cdot \left(({}^*\mathbf{X}_k^-)_{1:2N-1} - \hat{\mathbf{X}}_{k_{sp}}^- \right) \quad \mathbf{S}_Q^T \right] \begin{bmatrix} \sqrt{\omega_1^{(c)}} \cdot \left(({}^*\mathcal{X}_k^-)_{1:2N-1} - \hat{X}_{k_{sp}}^- \right)^T \\ \mathbf{S}_Q \end{bmatrix} \quad (5.116)$$

$$= \omega_1^{(c)} \cdot \left(({}^*\mathcal{X}_k^-)_{1:2N-1} - \hat{X}_{k_{sp}}^- \right) \left(({}^*\mathcal{X}_k^-)_{1:2N-1} - \hat{X}_{k_{sp}}^- \right)^T + \mathbf{Q} \quad (5.117)$$

$$\mathbf{P}_{x_k}^- = {}^*\mathbf{P}_{x_k}^- + \omega_0^{(c)} \cdot \left(({}^*\mathcal{X}_k^-)_0 - \hat{x}_k^- \right) \left(({}^*\mathcal{X}_k^-)_0 - \hat{x}_k^- \right)^T \quad (5.118)$$

$$= \sum_{i=0}^{2N-1} \omega_i^{(c)} \left(({}^*\mathcal{X}_k^-)_i - \hat{x}_k^- \right) \left(({}^*\mathcal{X}_k^-)_i - \hat{x}_k^- \right)^T + \mathbf{Q} \quad (5.119)$$

It is clearly shown, that the square-root form is equivalent to the UKF implementation, however, executed in a more compact way.

During the measurement update, the sigma-points are passed through the non-linear function $h(\cdot)$. This can happen in different ways, option 1 extends the traditional set of sigma-points to an augmented version ${}^{\text{aug}}\mathbf{X}_k^-$ with $4N + 1$ points, while other methods use previous set time $k - 1$ and add $2N$ new points to it processing more data in regards to the odd moments at the cost of more calculation efforts. For

method 1, the new set of sigma-points is generated as following [405]

$$\mathcal{X}_k^- = [\hat{x}_k^- \quad \hat{X}_k^- + \eta(\mathbf{S}_{x_k}^-)^T \quad \hat{X}_k^- - \eta(\mathbf{S}_{x_k}^-)^T] \quad (5.120)$$

The output is estimated as following:

$$\mathcal{Y}_k = h(\mathcal{X}_k^-, u_k) \quad (5.121)$$

$$\hat{y}_k = \sum_{i=0}^{2N-1} \omega_i^{(m)} (\mathcal{Y}_k)_i \quad (5.122)$$

followed by the Cholesky factorization for the measurement covariance and cross-covariance:

$$\mathbf{S}_{y_k}^- = \text{qr} \left\{ \left[\sqrt{\omega_1^{(c)}} \cdot ((\mathcal{Y}_k)_{1:2N-1} - \hat{Y}_{k_{sp}}) \quad \mathbf{S}_{\mathbf{R}}^T \right]^T \right\} \quad (5.123)$$

$$\mathbf{S}_{y_k}^- = \text{CHOLESKY-UPDATE} \left\{ \mathbf{S}_{y_k}, \sqrt{\omega_0^{(c)}} \cdot ((\mathcal{Y}_k)_0 - \hat{y}_k) \right\} \quad (5.124)$$

$$\mathbf{P}_{x_k, y_k} = \sum_{i=0}^{2N-1} \omega_i^{(c)} ((\mathbf{X}_k^-)_i - \hat{x}_k^-) ((\mathcal{Y}_k)_i - \hat{y}_k)^T \quad (5.125)$$

with

$$\hat{Y}_{k_{sp}} = [\hat{y}_k, \dots, \hat{y}_k], \hat{Y}_{k_{sp}} \quad (5.126)$$

For method 2, the new augmented set of sigma-points is represented in the following. It must be pointed out, that the sigma-point weights are also adapted to the new set and size ($N > 2N$) [405]

$$^{(aug)}\mathcal{X}_k^- = [* \mathcal{X}_k^- \quad (* \mathcal{X}_k^-)_0 + \eta \mathbf{S}_Q^T \quad (* \mathcal{X}_k^-)_0 - \eta \mathbf{S}_Q^T] \quad (5.127)$$

$$^{(aug)}\mathcal{Y}_k = h(^{(aug)}\mathcal{X}_k^-, u_k) \quad (5.128)$$

$$\hat{y}_k = \sum_{i=0}^{4N-1} \omega_i^{(m)} (^{(aug)}\mathcal{Y}_k)_i \quad (5.129)$$

$$\mathbf{S}_{y_k} = \text{qr} \left\{ \left[\sqrt{\omega_1^{(c)}} \cdot ((^{(aug)}\mathcal{Y}_k^-)_{1:4N} - (^{(aug)}\hat{Y}_k)) \quad \mathbf{S}_{\mathbf{R}}^T \right]^T \right\} \quad (5.130)$$

$$\mathbf{S}_{y_k} = \text{CHOLESKY-UPDATE} \left\{ \mathbf{S}_{y_k}^-, \sqrt{\omega_0^{(c)}} \cdot ((\mathcal{Y}_k^-)_0 - (^{(aug)}\hat{Y}_k)) \right\} \quad (5.131)$$

$$\mathbf{P}_{x_k, y_k} = \sum_{i=0}^{4N-1} \omega_i^{(c)} ((^{(aug)}\mathbf{X}_k^-)_i - \hat{x}_k^-) ((^{(aug)}\mathcal{Y}_k^-)_i - \hat{y}_k)^T \quad (5.132)$$

with the estimated output

$$^{(aug)}\hat{Y}_k = [\hat{y}_k, \dots, \hat{y}_k] \quad (5.133)$$

Following the KALMAN equation, the gain is calculated following an optimized least-

squares approach

$$\mathcal{K}_k = \mathbf{P}_{x_k, y_k} \mathbf{P}_{y_k}^{-1} = \mathbf{P}_{x_k, y_k} (\mathbf{S}_{y_k}^T \mathbf{S}_{y_k})^{-1} = \mathbf{P}_{x_k, y_k} / \mathbf{S}_{y_k} / \mathbf{S}_{y_k}^T \quad (5.134)$$

giving the *a posteriori* state estimate

$$\hat{x}_k^+ = \hat{x}_k^- + \mathcal{K}_k (y_k - \hat{y}_k^-) \quad (5.135)$$

and the Cholesky factorization of the state covariance as

$$D_C = \mathcal{K}_k \mathbf{S}_{y_k}^T \quad (5.136)$$

$$\mathbf{S}_{x_k}^+ = \text{CHOLESKY-UPDATE}(\mathbf{S}_{x_k}^-, d_i, ' - ') , i = 0, \dots, r-1 \quad (5.137)$$

with d_i is the i^{th} -column of the Cholesky downdate matrix D_C . Below, the steps for the UKF and the SRUKF for the measurement update is summarized for the covariance matrix.

$$\mathbf{P}_{x_k}^+ = \mathbf{P}_{x_k}^- - \mathcal{K}_k \mathbf{P}_{y_k} \mathcal{K}_k^T = \mathbf{P}_{x_k}^- - D_C D_C^T = \mathbf{P}_{x_k}^- - \sum_{i=0}^{r-1} d_i d_i^T = (\mathbf{S}_{x_k}^+)^T \mathbf{S}_{x_k}^+ \quad (5.138)$$

$$\mathbf{S}_{x_k}^+ = \text{CHOLESKY}(\mathbf{P}_{x_k}^+) = \text{CHOLESKY} \left(\mathbf{P}_{x_k}^- - \sum_{i=0}^{r-1} d_i d_i^T \right) \quad (5.139)$$

$$= \text{CHOLESKY-UPDATE}(\mathbf{S}_{x_k}^-, d_i, ' - ') , i = 0, \dots, r-1 \quad (5.140)$$

It can be seen, that the SRUKF evaluates in time the Cholesky factor of the covariance the same way as in the UKF form, however, in a more efficient way saving computational power.

5.3.2 Methods Using Non-Gaussian Approximation

Considering a case of non-Gaussian random variables, taking the methods earlier described, the required pdf's are still approximated by a Gaussian, which may induce distortions towards the underlying structure resulting in filter divergence. Analytical approximations include Gaussian sum filter or estimating the first two moments of the probability density. In this section the developed inference methods are introduced which solve the Recursive Bayesian estimation theory under the assumption of non-linear, non-Gaussian dynamic systems. These are necessary due to the fact that the non-linear transformation of Gaussian distributions result in non-Gaussian distributions. Under the stochastic point of view, the Bayesian framework depends on the concept of processing the integrals over the multi-dimensional parameter space [380]. For general cases, the integrals represent the normalizing factor of the posterior density function together with the MMSE estimate. For the recursive Bayesian estimation a further integral represents the posterior filtering

density function, but as discussed earlier, it leads to the following problem of finding analytical solutions. Therefore, approximations are necessary [380][394], but under the premises that an appropriate choice of the numerical integration approach, also under computational costs consideration, depend on the dimension and form of the integration area. In addition, for recursive Bayesian estimation, standard numerical integration methods do not lead to a mutually acceptable result, since after the filter initialization and first iteration the integrand are not analytically known and can only be approximated but not precisely determined. The posterior density

$$(x_1|Y_0) \approx \sum_{i=1}^n w_i g_i(x_1) \quad (5.141)$$

can be approximated based on local basis functions and their linear combination, which will be used in the recursive estimation. In Grid-based methods, the local basis function are derived from grid mesh, either adaptive or fixed. Grid-based methods (GBF), uses numerical integration to solve the integrals, here the posterior filter density function is discretized and recursively processed. Early work applies uniformly grid mesh which cannot be applied to higher order dimensions. The grid mesh is $\mathbb{R}^2$ Grid-based filter subdivide the probability density function into a deterministic discrete grid as

$$p(x_k|Y_{1:k}) = \sum_{i=1}^n w_k^{(i)} \delta(x_k - x_k^{(i)}) \quad (5.142)$$

The difficulty in this approximation lies in determining the correct position of the grid. For sparse grid, the estimation quality is reduced, and if the mesh is too fine, the numbers of points increase excessive while for both resolutions the grid may be moved and floating. The recursive point mass is now a problem of recursive weights evolution instead [406][380]

$$w_{k|k-1}^{(i)} = \sum_{j=1}^n w_{k-1|k-1}^{(j)} p(x_k^{(i)}, x_{k-1}^{(j)}) \quad (5.143)$$

$$w_{k|k}^{(i)} \propto w_{k|k-1}^{(i)} p(Y_k|x_k^{(i)}) \quad (5.144)$$

The operation counts in terms of function evaluations for $w_{k|k-1}^{(i)}$ is n^2 . The point-mass approach provides also an expansion to non-linear estimation problems.

- **Direct Numerical Integration - Grid-based Methods** Let's define N grid points to approximate $p(x_t|Y_t)$, for which each point has its individual probability mass. The idea is to approximate the posterior by a set of point values on a moving grid to replace the integrals by summations of weighted Dirac delta functions such as

$$p(x_t|Y_t) \approx \sum_{i=1}^N p_i \delta(x_t - x_t^i) \quad (5.145)$$

where the Dirac deltas are fixed and uniformly positioned and the sum is over a finite number of points with probability function p_i located at x^i [407]. Further work concentrated on piecewise constant approximations [392]. However, the bottleneck was considered to be the storing of the high number of grid points. As a consequence, focus shifted towards more advanced interpolation techniques resulting in less grid points but more complex analysis for grid point structure. One of the latest methods was defined by Julier et al (2000 and 2004), the so called unscented transform with the unscented KALMAN filter, in which the unscented transform generated a set of grid points in the state space. Another set of algorithms applies a set of multiple models for estimation, for which each model represents a phenomenon of the corresponding system. The interacting multiple model (IMM), generalized pseudo-Bayesian (GPB) and range parameterized EKF are used especially in target tracking applications. An further improvement to the point mass estimator is the sequential Monte Carlo method [380][394], applied in particle filters. Here the a posteriori density is approximated by a set of statistical chosen points, rather than in a deterministic way, as for point-mass filters [392][367].

- **Gaussian Sum Approximation** Often, on-line estimation is a function approximation problem, whose successful implementation depends on a efficient and reliable measurement and time update. Numerical solution is based on multiplications and convolutions between pdfs generating new Gaussian distributions. Another method to approximate the conditional pdf is based on the assumption that the stochastic variables are approximated by a sum of Gaussian density functions [367] [408][380][394]. If we approximate the a posteriori density using a sum of Gaussian densities as following

$$p(x_t|Y_t) \approx \sum_{i=1}^N w_t^{(i)} \cdot \mathcal{N}\left(\hat{x}_{t|t}^{(i)}, \mathbf{P}_{t|t}^{x,(i)}\right) \quad (5.146)$$

where $\sum_{i=1}^N w_t^{(i)} = 1$ and $\mathcal{N}$ represent the importance weights and normal distribution with mean and covariance respectively of the i^{th} Gaussian kernel with respect to the first measurements, resulting in the Gaussian sum approach [367]. In practice, terms containing small weights are not considered while similar weighted terms are combined to one for each calculation. This gives that for any probability distribution $p(x)$ it is possible when summing up a sufficient number of Gaussian normal distributions (GMM, Gaussian mixture model) the approximation converges uniformly to any probability density function in discrete form.

$$p(x) = p_{\text{GMM}}(x) \approx \sum_{i=1}^N w_t^{(i)} \cdot \mathcal{N}(x_k; \bar{x}^{(i)}, \mathbf{P}_{t|t}^{x,(i)}) \quad (5.147)$$

Based on this Gaussian mixture models, recursive algorithms were derived for dynamical system with additive Gaussian white noise, for which mean and covariances

of the Gaussian components are passed applying the linearized model. The updated Gaussian stochastic parameters with its weights are set equal to the prior weights. Once the measurement is available, the approximation of the posterior probability function by applying BAYES theorem resulting to the so called Gaussian Sum Filter (GSF), which allows to deal with non-Gaussian densities. The higher the number of Gaussian densities, the better the convergence to any probability function. Considering the dynamic state space model with the assumptions that at time $k - 1$ the density $p(x_{k-1}|Y_{k-1})$ and that the filter density functions for system and measurement noise can be approximated by the sum of GMM as following [380]

Probability density at $k - 1$

$$p_{\text{GMM}}(x_{k-1}|Y_{k-1}) = \sum_{g=1}^G w_{k-1}^{(g)} \cdot \mathcal{N}(x_{k-1}; \bar{x}_{k-1}^{(g)}; \mathbf{P}_{k-1}^{x,(g)}) \quad (5.148)$$

System noise

$$p_{\text{GMM}}(\kappa_{k-1}) = \sum_{m=1}^M v_{k-1}^{(m)} \cdot \mathcal{N}(\kappa_{k-1}; \bar{\kappa}_{k-1}^{(m)}; \mathbf{P}_{k-1}^{\kappa,(m)}) \quad (5.149)$$

Measurement noise

$$p_{\text{GMM}}(\gamma_k) = \sum_{j=1}^J \rho_k^{(j)} \cdot \mathcal{N}(\gamma_k; \bar{\gamma}_k^{(j)}; \mathbf{P}_k^{\gamma,(j)}) \quad (5.150)$$

$$(5.151)$$

and subsequent density functions can also be approximated by GMM [380] .

1. **Prediction step:** priori density function at time k

$$p(x_{k-1}|Y_{k-1}) \approx p_{\text{GMM}}(x_{k-1}|Y_{k-1}) = \sum_{g'=1}^{G'} w_{k|k-1}^{(g')} \cdot \mathcal{N}(x_{k|k-1}; \underbrace{\bar{x}_{k|k-1}^{(g')}, \mathbf{P}_{k|k-1}^{x,(g')}}_{\text{prediction}}) \quad (5.152)$$

2. **Filtering step:** Posteriori density function at time k

$$p(x_k|Y_k) \approx p_{\text{GMM}}(x_k|Y_k) = \sum_{g''=1}^{G''} w_{k|k}^{(g'')} \cdot \mathcal{N}(x_{k|k}; \underbrace{\bar{x}_{k|k}^{(g'')}, \mathbf{P}_{k|k}^{x,(g'')}}_{\text{filtering}}) \quad (5.153)$$

$$\text{with } G' = G \cdot M, \quad G'' = G \cdot M \cdot J$$

where G,M,J are the represent the amount of GMM models for pdf of states, system and measurement noise, respectively. Since all the stochastic components of the GMM are of Gaussian property, the prediction and filtering steps are carried by

propagating mean and covariance from time k to time $k - 1$ applying e.g. a filter bank in parallel of EKF, which would then result in a Gaussian Sum (extended) KALMAN filter. Their corresponding weights for the prediction step are calculated in form

$$w_{k|k-1}^{(g)} = \frac{w_{k-1|k-1}^{(g')} \cdot v_{k-1}^{(m)}}{\sum_{g=1}^G \sum_{m=1}^M w_{k-1|k-1}^{(g)} \cdot v_{k-1}^{(m)}} \quad (5.154)$$

and the weights for the filtering step are given by

$$w_{k|k}^{(g'')} = \frac{w_{k|k-1}^{(g')} \cdot \gamma_k^{(j)} \cdot p^{(j)}(y_k|x_k)}{\sum_{g'=1}^{G'} \sum_{j=1}^J w_{k|k-1}^{(g')} \cdot \gamma_k^{(j)} \cdot p^{(j)}(y_k|x_k)} \quad (5.155)$$

Generally, weights of different components of a Gaussian mixture don't change on-line from time k to time $k - 1$ while passing the stochastic values through nonlinear function. The update is obtained when measurement data is available. This property is contradicting to the general non-linear approaches, where new values of weights are necessary for accurate propagation of the density function. This concept is based on the fact that the change and values of covariance is small and the linearization is reasonable around the means [394]. However, this case may be applicable only if the corresponding dynamics are linear or the system behaves conditionally non-linear or the observations are superimposed with marginal noise and free of offset and obtained with high sample rate. In application, especially for battery storage systems, these underlying assumptions lead to unsatisfactorily results for predicting the density function. As batteries have strong non-linear behavior. The amount of GMM's from G to G' during prediction and G' to G'' during filtering leads to an increase of GMM components which restricts the extent of implementation into application by the available computing power. In order to reduce the exponential growing operation counts, GMM's with similar stochastic properties are summarized or those with low weighting factors are neglected. The output of the Gaussian mixture filter is obtained by combining estimates for a bank of KALMAN-like filter, for which each individual filter matches a corresponding distribution of the Gaussian mixture [409].

- **Simulation Based Monte Carlo Methods** - to deal with the Bayesian estimation problem, the numerical integration has to be solved. The restriction lies in the exponential growth for increasing dimension of the integration region. Simulation based (Markov Chain) Monte Carlo integration accounts for a whole spectrum of approaches that provide solutions to non-linear, non-Gaussian complex systems and differ from previous described methods in the relative error which is in general independent from the state dimension but still keeping the requirements on the computing power very high [380]. The idea of Monte Carlo sampling (MCS) approximation is based on stochastic sampling to solve numerical problems. Hereby, MCS uses a set of random variables in a probability space to approximate the corresponding integral. Taking into account, a sequence of random samples from the

probability distribution is drawn, then the integral function $f(x)$ can be estimated as following [410]. It must be remembered, that MCS' accuracy is independent of dimensionality of the state vector [380][394].

$$\hat{f}_N = \frac{1}{N} \sum_{i=1}^N f(x^{(i)}) \quad (5.156)$$

where $\{x^{(1)}, \dots, x^{(N)}\}$ represents the random samples. By the law of large numbers, $\hat{f}_N(x)$ is going to converge toward the expected value for f_x . Now, two questions must be asked, how shall the samples be extracted from the probability distribution (p_x). Lastly, the estimator which is able to determine the stochastic properties of the function with respect to the density.

Perfect Monte Carlo Sampling - All Monte Carlo methods are formed by the perfect sampling idea used to approximate the pdf's where the probability integral $I(g)$ [380][394][367]

$$p(x_k|Y_k) \approx \hat{p}(x_k|Y_k) = \frac{1}{N} \sum_{i=1}^N \delta(x_k - x_k^{(i)}) \quad (5.157)$$

whereas $\{x_k^{(i)}, i = 1, 2, \dots, N\}$ samples are generated from $p(x_k|Y_k)$ with the dirac-delta function δ . It follows that the expectation described by:

$$\mathbb{E}[g(x_k)] = \int g(x_k) p(x_k|Y_k) dx \quad (5.158)$$

$$\text{discrete sum approximation via } \tilde{\mathbb{E}}[g(x_k)] \approx \frac{1}{N} \sum_{i=1}^N g(x_k^{(i)}) \quad (5.159)$$

The samples $x_k^{(i)}$ are independent from each other and identically distributed and for large sample numbers, the estimated value converges towards the true value $\tilde{\mathbb{E}}[g(X_k)] \rightarrow \mathbb{E}[g(X_k)]$ by the strong law of large numbers. Nevertheless, this technique cannot be directly be applied in the later introduced particle filter equations, since the density information is only available up to a proportionally constant. The following importance sampling method makes it possible to work around the problem.

Bayesian Importance Sampling - As the name suggests, the objective of importance sampling (IS) considers to draw samples within the region of "importance" to reduce the necessary of computing power for higher dimensional state space. The beauty of this sampling method is, that no prior information regarding the distribution is necessary, since a proposal distribution $\pi(X_k|Y_k)$ is generated, based on a sequence of states $X_k = \{x_0, x_1, \dots, x_k\}$ and replaces the true distribution. The results are improved by choosing a good proposal distribution. This will be

substituted as [380][394]

$$\mathbb{E}[g(X_k)] = \int g(X_k) \frac{p(X_k|Y_k)}{\pi(X_k|Y_k)} \pi(X_k|Y_k) dX_k \quad (5.160)$$

$$= \int g(X_k) \frac{\frac{p(Y_k|X_k)p(X_k)}{\pi(X_k|Y_k)}}{p(Y_k)} \pi(X_k|Y_k) dX_k \quad (5.161)$$

$$\text{originates from} \Rightarrow \int f(x)p(x)dx = \int f(x) \frac{p(x)}{\pi(x)} \pi(x)dx \quad (5.162)$$

$$\text{important weights } w_k(X_k) = \frac{p(Y_k|X_k)p(X_k)}{\pi(X_k|Y_k)} \quad (5.163)$$

Since the normalizing factor of $p(x)$ is unknown, we eliminate the normalized density $p(Y_k)$ as

$$\mathbb{E}[g(X_k)] = \frac{1}{p(Y_k)} \int g(X_k) w_k(X_k) \pi(X_k|Y_k) dX_k \quad (5.164)$$

$$= \frac{\int g(X_k) w_k(X_k) \pi(X_k|Y_k) dX_k}{\int p(Y_k|X_k) p(X_k) \frac{\pi(X_k|Y_k)}{\pi(X_k|Y_k)} dX_k} \quad (5.165)$$

$$= \frac{\int g(X_k) w_k(X_k) \pi(X_k|Y_k) dX_k}{\int w_k(X_k) \pi(X_k|Y_k) dX_k} \quad (5.166)$$

$$= \frac{\mathbb{E}_\pi[g(X_k) w_k(X_k)]}{\mathbb{E}_\pi[w(X_k)]} \quad (5.167)$$

Once the samples are generated from the proposed distribution $\pi(X_k|Y_k)$, the expectation can be approximated as [380][394]

$$\mathbb{E}[g(X_k)] \approx \tilde{\mathbb{E}}[g(X_k)] = \frac{\frac{1}{N} \sum_{i=1}^N g(X_k^{(i)}) w_k(X_k^{(i)})}{\frac{1}{N} \sum_{i=1}^N w(X_k^{(i)})} \quad (5.168)$$

$$= \sum_{i=1}^N \tilde{w}_k^{(i)} g(X_k^{(i)}) \quad (5.169)$$

with the normalized important weights

$$\tilde{w}_k^{(i)} = \frac{w_k(X_k^{(i)})}{\sum_{j=1}^N w_k(X_k^{(j)})} \equiv \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}} \quad (5.170)$$

The importance sampling is biased, meaning that the expected value is not like the true value. However, the bias disappears rapidly under following assumptions: [380][394]

- $X_k^{(i)}$ are independent with identical distributed samples drawn from $\pi(X_k|Y_k)$
- The data space of $\pi(X_k|Y_k)$ matches the shape of the posteriori and $\mathbb{E}[g(X_k)]$ exists. If $\pi(X_k|Y_k)$ is relatively far off to true $p(X)$ the importance weights are uneven distributed and are insignificant.
- The the expectation of posteriori density of $\tilde{w}_k$ and $\tilde{w}g^2(X_k)$

Now, the a solution to the Bayesian estimation problem is found by solving for the a posteriori density $p(X_k|Y_k)$

$$\hat{p}(X_k|Y_k) = \sum_{i=1}^N \tilde{w}_k^{(i)} \delta(X_k - X_k^{(i)}) \quad (5.171)$$

Sequential Importance Sampling - is derived from the perfect sampling. As already mentioned, for importance sampling it is essential that a good proposed distribution is applied. In general, to find a fitting proposal distribution is especially uncertain in a high-dimensional state space. One way to to circumvent this problem is to sequentially form the proposal distribution, defining the sequential importance sampling (SIS). This means, after the proposal distribution is initialized in a factorized form without the need of changing the states $X_{k-1}^{(i)}$ the distribution is rewritten as [380][394]

$$\pi(X_k|Y_k) = \pi(X_{k-1}|Y_{k-1})\pi(x_k|X_{k-1}, Y_k) \quad (5.172)$$

For our purpose and advanced battery technology, we assume that the states follow a first-order MARKOV process and for the given states are independent form future observations [380].

$$p(X_k) = p(x_0) \prod_{j=1}^k p(x_j|x_{j-1}) \quad \text{and} \quad p(Y_k|X_k) = \prod_{j=1}^k p(y_j|x_j) \quad (5.173)$$

Therefore, particle filter using sequential importance sampling (SIS) to directly an

optimal Bayesian solution, in a recursive manner, the complete posterior state density is approximated [396]. In MC approach, a set of weighted samples, extracted from the posterior distribution is applied to map the integrals to discrete sums [396][380]. As mentioned earlier, due to the difficulty to draw directly samples from the posterior density function, One can get around this problem by using importance sampling or sampling from a known proposal distribution $q(X_0^k|Y_0^k)$. With the proposal distribution, following substitution will lead to the unnormalized important weights [380][394]:

$$\mathbb{E}[g(X_0^k)] = \int g(X_0^k) \cdot \frac{p(X_0^k|Y_0^k)}{q(X_0^k|Y_0^k)} \cdot q(X_0^k|Y_0^k) dX_0^k \quad (5.174)$$

$$= \int g(X_0^k) \cdot \frac{p(Y_0^k|X_0^k)p(X_0^k)}{p(Y_0^k)q(X_0^k|Y_0^k)} \cdot q(X_0^k|Y_0^k) dX_0^k \quad (5.175)$$

$$= \int g(X_0^k) \cdot \frac{w_k(X_0^k)}{p(Y_0^k)} \cdot q(X_0^k|Y_0^k) dX_0^k \quad (5.176)$$

$$(5.177)$$

Applying the equations to the unnormalized importance weighting function, a recursive form can be extracted as

$$w_k = \frac{p(Y_k|X_k)p(X_k)}{\pi(Y_{k-1}|X_{k-1})\pi(x_k|X_{k-1},Y_k)} \quad (5.178)$$

$$= w_{k-1} \frac{p(Y_k|X_k)p(X_k)}{\pi(Y_{k-1}|X_{k-1})p(X_{k-1})} \frac{1}{\pi(x_k|X_{k-1},Y_k)} \quad (5.179)$$

$$= w_{k-1} \frac{p(y_k|x_k)p(x_k|x_{k-1})}{\pi(x_k|X_{k-1},Y_k)} \quad (5.180)$$

with sampling from known proposal distribution $q(X_0^k|Y_0^k)$, and where $w_k(X_0^k)$ represents the important weights such as

$$w_k = \frac{p(Y_0^k|X_0^k)p(X_0^k)}{q(X_0^k|Y_0^k)} \quad (5.181)$$

After canceling the unknown normalizing density $p(Y_0^k)$, the sequential update to the importance weights can be designed by revising the proposal distribution as

$$q(X_0^k|Y_0^k) = q(X_0^{k-1}|Y_0^{k-1})q(x_k|X_0^{k-1},Y_0^{k-1}) \quad (5.182)$$

The assumption for the next step is, that the state k following Markov model, does not depend on future measurements while all measurements are conditionally independent, the recursive form for the weights can be written as

$$w_k = w_{k-1} \cdot \frac{p(y_k|x_k) \cdot p(x_k|x_{k-1})}{q(x_k|X_0^{k-1},Y_0^{k-1})} \quad (5.183)$$

The requirements for a successful sequential update of w_k depends on the choice of appropriate $q(X_0^{k-1}|Y_0^k)$. The solution described by Haykin takes the sampled values from $q(X_0^{k-1}|Y_0^k)$ and analysis the likelihood $p(y_k|x_k)$ with change in state $p(x_k|x_{k-1})$ from which a evaluation set of samples are generated and its updated important weights are calculated iteratively.

$$\begin{aligned}\mathbb{E}[g(X_0^k)] &\approx \frac{N^{-1} \sum_{i=1}^N g(\chi_{0:k}^{(i)}) w_k(\chi_{0:k}^{(i)})}{N^{-1} \sum_{i=1}^N w_k(\chi_{0:k}^{(i)})} \\ &= \sum_{i=1}^N g(\chi_{0:k}^{(i)}) \tilde{w}_k(\chi_{0:k}^{(i)})\end{aligned}\quad (5.184)$$

where $\tilde{w}$ symbolizes the normalized importance weights, and $\chi_{0:k}^{(i)}$ the i th sample points from $q(X_0^{k-1}|Y_0^k)$. While N goes to infinity, the error of approximated posterior density function goes to 0 using point-mass estimate [396]

$$\hat{p}(X_0^k|Y_0^k) = \sum_{i=1}^N \tilde{w}_k^{(i)} \delta(X_0^k - \chi_{0:k}^{(i)}) \quad (5.185)$$

and the posterior filtering density by [396]

$$\hat{p}(x_k|Y_0^k) = \sum_{i=1}^N \tilde{w}_k^{(i)} \delta(x_k - \chi_k^{(i)}) \quad (5.186)$$

The estimation error depends mainly on the amount of particles used for the approximation. In comparison to the minimum-variance Gaussian approximation like the EKF, SIS allows to capture complex structures of higher-order moments. This enables to approximate any posteriori density function (e.g. modal distribution) by the point-mass approximation. The accuracy thereby depends on the amount of samples. and in comparison to GAUSSIAN approximation methods, they offer higher order moments approximation. The measure $\mathbb{E}[(n_k - Nw_k^{(i)})^2]$ points out the spacing between the density peak. The resampling is improved by reducing the spacing within all peaks [411]. In general, the resampling methods are unbiased, however, separate in regards to the underlying variance where n_k copies of the samples $x_k^{(i)}$. Therefore, n_k with low variance improves the results and the two commonly used techniques for variance reduction are discussed. The SIS method may be applied not only for Markov chain framework, but also for non-BAYES problem formulation. For high dimensional state space, the unconditional variance of the weights increase, which results in many weight coefficients close to zero but still taking part in the computation. A re-sampling step in an more advanced MONTE CARLO sampling method, which will be discussed later, is suggested and applied after weight normalization.

Choice of proposal distribution The challenge in implementing a successful SIS lies within the correct proposal distribution $q(x_k|X_{k-1}, Y_k)$, as it will minimize the variance of the weights. On the other hand, the optimal proposal may lead to high complexity and unsolvable integrals. Instead, $p(x_k|x_{k-1})$ may be used as it does not account for the most current measurement information leading the weights calculation

$$\begin{aligned} w_k &= w_{k-1} \frac{p(y_k|x_k)p(x_k|x_{k-1})}{q(x_k|X_{k-1}, Y_k)} \\ &= \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(x_k|x_{k-1})} \\ &= w_{k-1}p(y_k|x_k) \end{aligned} \tag{5.187}$$

Even though the implementation effort can be reduced, the evaluation of the likelihood may lead to only a small number of particles with meaningful weights. Therefore, to improve the results, the particles shall be moved into the area with higher likelihood.

5.3.3 Resampling

As mentioned earlier, SIS struggles with the fact that the variance of the weights increases statistically with time [380] resulting in converting particle towards 1, while the rest decays to 0, losing these particles. The re-sampling strategy shall counteract this problem. Re-sampling is an important step for the later introduced particle filter family. A justified conclusion regarding which re-sampling algorithm shall be used, it defines the complexity and/or improves the re-sampling quality and estimation results. Therefore, in this section different techniques which can be found in the literature are compared analytically and simulative to decide which method shows the most promising results in regards to our application and state space dimension. A re-sampling step adjusts the weighted approximated density $\hat{p}_N$ to an unweighted $\tilde{p}_N$ by rejecting the samples with lower importance. In the literature different re-sampling methods are discussed throughout many articles, patents and books in engineering, finance or biotechnology. However, a comparison and evaluation is still missing and shall be discussed below where the common algorithms are analyzed and compared in regards to the computational efficiency on paper and simulation.

Sequential importance re-sampling overcomes the burden of the proposed sequential sampling methods, where the unconditional variance of the important weights increases in a stochastic manner for one sample towards 1, while the rest converges towards zero having inconsequential effect. This effect is called the weight degeneracy problem [387]. A method which is able to minimize the degeneracy is introduced as the sequential importance re-sampling (SIR), where samples with lower weights are rejected and weights with high importance are multiplied whereas the samples are kept constant. In [392], [367],

the definition of an optimal sample number is approximated by

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (\tilde{w}_k^{(i)})^2} \quad (5.188)$$

where $\tilde{w}$ represents the normalized importance weights. This will lead to a small effective sample size when the variance of the importance weights is large [367]. If the sample size even undergoes a pre-defined threshold, the generic principle of re-sampling is activated to decrease the variance where dedicated algorithms generate particles (samples) $\{x^{(i)}\}$ from an importance density π , then calculate the importance weights $w_k^{(i)}$ for each sample, then find $\tilde{w}_k$ by normalizing the samples. The sample generated from $p(x)$ is just sampled with the replacement from the set of samples $\{x^{(i)}\}$ with probability proportional to the normalized weights [412]. Many different re-sampling methods are discussed in the literature, the most common re-sampling strategies are [380][410]

1. (Adaptive) SIR originates from the Bootstrap methods. More details can be found in [413][386].
2. Multinomial re-sampling
3. Residual re-sampling (RRS) is a two-step selection method, and partially deterministic whereas SIR also included. During the first step, $N_a^{(i)} = \lfloor N \tilde{w}_k^{(i)} \rfloor$ copies of $x_n^{(i)}$ are generated followed by the SIR to generate $\tilde{N}_k = N - \sum_{i=1}^N N_a^{(i)}$ samples and new weight

$$\tilde{w}_k^{(i)} = \tilde{N}_k^{-1} (\tilde{w}_k^{(i)} N - N_a^{(i)}) \quad (5.189)$$

RRS proves to be less demanding in regards to computing power in comparison to SIR an achieves a lower sample rate with reduced sample variance [410],[380].

4. Minimum Variance Sampling (Systematic re-sampling) (MVS) combines different re-sampling methods, firstly the Systematic re-sampling/Stratified and Tree Based Branching method where the weights are processed as continuous random variables within $[0,1]$ with random order, similar distributed grid points with distance $\frac{1}{N}$ [410]. The individual samples is mapped and the new samples set is generated in such a way that the variance $\mathbb{E}[(N - \mathbb{E}[N])^2]$
5. Local Monte Carlo re-sampling is discussed
6. Stratified Sampling is based on probability linearization technique, and unlike the simple or systematic random linearization, it distributes the samples unevenly in regards to their respective variance and is pre-partitioned into different intervals, so called strata regions $\mathcal{D}$ $(0,1]$ that is $\mathcal{D} = \cup_{j=1}^p \mathcal{D}_j$, in which a separate uniform random number is generated. By generating N_j samples x_{jk}^* from the normalized

restricted density for each strata p_j an estimator can be expressed as [414][380]

$$\hat{I}^s(g) = \sum_{j=1}^p \frac{\rho_j}{N_j} \sum_{k=1}^{N_j} g(x_{jk}^*), \quad \text{where } x_{jk}^* \sim p_j \quad (5.190)$$

with ρ_j represents the probability of strata region $\mathcal{D}_j$ and proves to be unbiased as can be seen in [414]. The earlier mentioned proportional allocation where $N_j = N\rho_j$ the integral has following property: $\text{Var } \hat{I}^s(g) = N^{-1}\mathbb{E}[\text{Var}_{\mathbf{P}_j} g] \leq N^{-1}x \text{Var}_{\mathbf{P}_n} g$, thus, the decrease in variance may be possible. In the literature further re-sampling methods are discussed, such as Gibbs Sampling, Bootstrap, Quasi-Monte Carlo, Hybrid Monte Carlo, Metropolis-Hastings Algorithm, but only the re-sampling methods used in this work is briefly discussed.

Remarks on re-sampling quality

The quality of re-sampling can be defined from a theoretical point of view, e.g. the variance of w_i and the estimation error between particle filter estimate and true value, which may be not representative, since an estimate has many influencing factors. On the assumption, reduced variance leads to improved estimate and shall be proven by simulations. The re-sampling methods can be grouped into different techniques, simple random re-sampling, stratified re-sampling and systematic re-sampling, whereas the two latter methods show lower variance and systematic resample general outperforms stratified re-sampling. Based on our assumption that a minimum variance of w_i leads to improved estimates, all but the simple re-sampling method, although the systematic re-sampling is well-disposed.

5.3.4 Different Resampling Methods

In general, defining the needed computation power is difficult to determine, this is owned by the fact of different system architecture, implementation style with toolchain, memory management and more, therefore, the comparison is done in an qualitative manner. From the analytical definition point of view, the re-sampling methods are similar but differ in the how to order the sequence of numbers.

Sequential Monte Carlo Methods - Generic Particle Filter

As discussed in the previous points, out of the combination of Sequential Importance Sampling and re-sampling algorithms, the basic particle filter is derived, called the Generic Particle Filter (GPF). For the implementation, the key element in a successful estimation lies in choosing the proposal distribution $q(x_k|X_{k-1}^0, Y_k^0)$, meaning out of an initial density

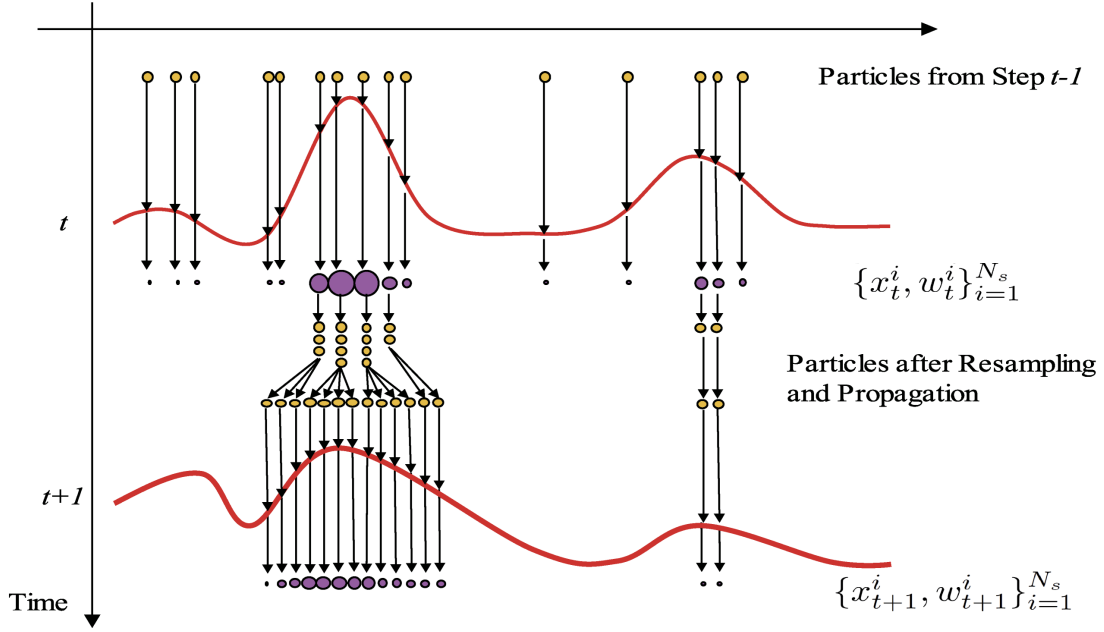


Figure 5.6: A pictorial description of the particle filter. [415]

$p(x_0)$ an initial set of samples is generated. This set is passed forward sequentially by SIS and directly resampled per unit time. The Fig.5.6 below illustrates the idea. The decision determines the overall performance and depends on to assumptions:

- Monte Carlo simulation (MC): Monte Carlo method delivers a numerical solution and optimization where a set of weighted samples (particles) extracted from the posteriori filtering density is applied to transform the BAYES integrals to discrete sums [396].
- Importance Sampling assumption: Random particles are drawn from the posteriori filtering density which can be approximated as $\hat{p}(x_k|Y_k) = \frac{1}{N} \sum_{i=1}^N \delta(x_k - x_k^{(i)})$. Often it is difficult to sample from the posteriori density, importance sampling alternatively draws particles from a known proposal distribution and their weights are sequentially updated.

Are these assumptions fulfilled and the given proposal distribution is appropriate initialized the GPF allowing at least satisfactory or even excellent results, with the following restriction, that point-mass estimates can approximate the general distribution in dependence of resolution (number of particles) used. For e.g. low resolution, the *problem of degeneracy* may arise, where sequentially the weights will concentrate on a few samples. To address the problem of degeneracy the particles are moved to the areas of high likelihood [416]. As a consequence, this may lead to *particle impoverishment*, where other importance weights values are getting close to zero and eventually lead to a weight col-

lapse. Here, a solution has to be found under the premise that we have to live with the compromise of re-sampling only at deterministic steps, such as when the variance of the non-normalized weights exceeds the threshold. The decision of the number of particles is crucial for the filters efficiency and accuracy since the real-time applicability and the estimation error heavily depends on this number. In general, a fixed number is used extracted from MC simulation, statistical bounds or improvised criteria resulting in choosing the particle number higher than needed or selecting the amount too small causing divergence. An intuitive way to choose the particle size would consider the complexity of the true density (complex pdf requires higher amount of particles) and how well shall the proposed density meet the original (big difference wastes samples). In the following, some techniques are introduced which tackle these problems.

Auxiliary Sequential Importance Re-sampling

Considering that particles are re-sampled at the end of the iteration $n - 1$ prior to taking the observation y_n into account seems inefficient. The question arises whether it makes sense to take the next measurement into consideration before the re-sampling algorithm is applied. The Auxiliary Sequential Importance Re-sampling used as auxiliary particle filter (APF) was first introduced by PITT&SHEPARD to reduce the earlier discussed deficiencies of SIR by adding an auxiliary variable (defined as: $k^{(i)} \propto w_k^{(i)} p(y_k | \mu_k^{(i)})$) which allows for more appropriate update of the filter. This approach keeps diversity in the sample interval by employing the next measurement in combination with the present information for the re-sampling selection tuned by the auxiliary variable which stands for the particle index [417]. The essence of this APF is based on the approximation function $\mu_k^{(i)}$, which corresponds to the state x_k for given $x_{k-1}^{(i)}$, while the weights are multiplied by the likelihood function and re-sampled [418]. The approximation of the density $p(y_k | x_{k-1})$ resulting in $\hat{p}(p(y_k | x_{k-1}))$ may give an appropriate weighting. The resampled particles at time $k - 1$ are propagated through the state space model. Assuming that the filtered posteriori function is represented by N weighted samples:

$$p(x_k | Y_k) \approx \sum_{i=1}^N w_k^{(i)} \delta(x_k - x_k^{(i)}) \quad (5.191)$$

The generated particles $x_k^{(i)}$ are calculated and weighted as

$$w_k^{(i)} \propto \frac{p(y_k | x_k^{(i)})}{p(y_k | \mu_k^{(i)})} \quad (5.192)$$

which will update the importance weights to reduce the discrepancy between the likelihood at the current sample and the predicted point $\mu_k^{(i)}$. At each iteration, the following functions are executed, first drawing a sample indexed with i which will be passed at time $k - 1$ into the new step k . The indexes are gathered based on the likelihood function of some

reference points from the approximation function. Using these indexes the conditional samples can be extracted as $x_k^{(i)} \sim p(x|x_k^{(i)})$. In general, the APF can be described as a special type of SIR algorithm with a different approach of importance weights with a marginal increase of computational cost but results obtained for SIR can be used for APF too. For importance sampling framework, choosing a proposal distribution with a heavy-tailed distribution in comparison to the target distribution lead to an estimator with finite variance and is defined by a $\hat{p}(y_n|x_{k-1})$. Choosing $\hat{p}(y_n|x_{k-1})$ which more diffuse than $p(y_n|x_{k-1})$ is therefore crucial [418]. It can also be interpreted that at low system noise the ASIR is more robust than the GPF in terms of outliers. However, it is not necessarily the case that the APF algorithm outperforms the SIR, e.g. for high system noise when the re-sampling step is started with faulty $p(x_k|x_{k-1}^{(i)})$. This is also the case when for $k < n$, $p(x_k|y_{k+1})$ is closer to $p(x_k|y_n)$ than $p(x_k|y_k)$ is. It can be assumed that this holds for the DSSMs used in this work, therefore, the APF will outperform the GPF under the circumstances that a good approximation of the predictive likelihood is provided. Also, different variances of the estimators running with SIR or APF need to be considered. Each type performs better in estimating for certain functions.

Reduced Particle impoverishment methods

It can be pointed out as a major drawback of re-sampling in the step of sampling for importance re-sampling algorithm, that the problem of degeneracy is followed by impoverishment. In the following improved re-sampling techniques shall be discussed which address the problem of importance weight values collapse. The most common applied methods are based on the regularized particle filter (RPF) with regularization re-sampling step (RRS) and the Markov Chain Monte Carlo (MCMC) algorithm.

- **Regularized Particle Filter** The problem of sample impoverishment originates from the fact that we have discrete samples forming a distribution which tries to approximate a continuous distribution. For the continuous case, a pure positive distribution is impossible, on the other side, given a discrete distribution, it is possible to draw e.g. two exact identical samples. Generally, in re-sampling a interval of N particles is generated by drawing samples from the discrete approximation to the full posterior or filtering distribution $p(x_k|Y_k)$ provided by the importance weighted particles $p(x_k|Y_k) \approx \sum_{i=1}^N w_k^{(i)} \delta_{x_k^{(i)}}$, where the delta function is centered at $x_k^{(i)}$. In the RPF, the filtering distribution $p(x_k|Y_k)$ is approximated using a kernel density estimate instead of the samples themselves. According to this, for the re-sampling step, the filtering distribution is replaced by

$$p(x_k|Y_k) \approx \sum_{i=1}^N w_k^{(i)} K(x_k - x_k^{(i)}) \quad (5.193)$$

with $K(x_k - x_k^{(i)})$ representing a kernel function around $x_k^{(i)}$. The optimal kernel density function corresponds to the function, which minimizes the distance between the kernel density estimate $\sum_{i=1}^N w_k^{(i)} K(x_k - x_k^{(i)})$ and the true function. Its bandwidth together with shape can be analytically be solved when setting the importance weights equal [419] and is called the Eqanechnikov kernel. The kernel is described by

$$K_\lambda = \frac{1}{N\lambda^n} K\left(\frac{x_k - x_k^{(i)}}{\lambda}\right) \quad (5.194)$$

where the kernel K represents a symmetrical probability density function with following properties:

$$K > 0 \quad \int K(x)dx = 1 \quad \int xK(x)dx = 0 \quad \int \|x\|^2 K(x)dx < \infty$$

The kernel bandwidth is chosen in such a way, that the Mean Integrated Squared Error (MISE) is calculated as

$$\text{MISE} = \int_{\mathbb{R}^n} \mathbb{E} \left[\left(\hat{p}(x_k | Y_k) - \frac{1}{N} \sum_{i=1}^N \phi_{1/h}(x_k - x_k^{(i)}) \right)^2 \right] dx \quad (5.195)$$

[420] where $p(x_k | Y_k)$ is $\frac{1}{N} \sum_{i=1}^N \phi_{1/h}(x_k - x_k^{(i)})$ and $\phi_{1/h}$ is defined as a family of rescaled kernels, representing the bandwidth of the kernel when $h > 0$. Another shape which will improve computation time is based on GAUSS normal distribution and covariance. Each sample $x_k^{(i)}$ is mapped $(\mathbf{S}^x)x_k^{(i)}$, with $\mathbf{S}(\mathbf{S}^x)^T = \mathbf{P}_{\text{RRS}^x}$ with $\mathbf{P}_{\text{RRS}^x}$ representing the covariance of the discrete set of particles. The bandwidth is calculated as

$$\lambda_{\min} = \left(\frac{4}{(n+2)N} \right)^{\frac{1}{n+4}} \quad \text{for multimodal distribution} \longrightarrow \lambda_{\min} = \frac{\lambda_{\min}}{2} \quad (5.196)$$

After the re-sampling step a new set of particles, $x_k^{(i)*} = x_k^{(i)} + \lambda_{\min} \mathbf{S}^x \epsilon^i$ is extracted, where $\epsilon^i \sim \mathcal{N}(\epsilon; 0, \mathbf{I}_n)$.

- Markov Chain Monte Carlo Methods** The difference to Ordinary Monte Carlo (OMC) lies in the fact that the sequence of random elements in OMCs are independent and identically distributed [421]. Monte Carlo methods based on Markov Chains (MCMC) are used to generate fair samples from complex distributions in high-dimensional space, with the help of random numbers which are gathered from uniform probability in a range of interests. In many cases, this method is the only effective [422] technique to approximate posteriori densities of highly combinatorial quantities. It is based on the idea, that a particle $x_k^{(i)}$ is only mapped to a new state $x_k^{(i)*}$ when $\vartheta \leq \alpha$, with $\vartheta \sim \mathcal{U}(\vartheta; 0, 1)$. α represents the acceptance ratio and indicates how probable the newly transformed sample is with $\alpha = \min\{1, \mathcal{R}\}$, with $\mathcal{R}$ representing the METROPOLIS-HASTINGS ratio. The MCMC step is followed after

the re-sampling procedure, based on the condition that

- N samples $x_k^{(i)}$ obtained from the re-sampling procedure at time step k and $x_{k-1}^{(i)}$ at $k-1$,
- N samples $x_k^{(i)*} \sim \pi(x_k|x_{k-1}^{(i)}, y_k)$ from the proposal distribution

are generated. The METROPOLIS-HASTINGS ratio is calculated as [380]

$$\mathcal{R} = \frac{p(y_k|x_k^{(i)*})p(x_k^{(i)*}|x_{k-1}^{(i)})\pi(x_k^{(i)}|x_{k-1}^{(i)}, y_k)}{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})\pi(x_k^{(i)*}|x_{k-1}^{(i)}, y_k)} \quad (5.197)$$

The METROPOLIS-HASTINGS ratio can be further simplified when replacing the proposal distribution by transition density $p(x_k|x_{k-1})$ leading to the evaluation of the likelihood function $\mathcal{R} = \frac{p(y_k|x_k^{(i)*})}{p(y_k|x_k^{(i)})}$. In general, the MCMC with Hastings ratio delivers improved estimation in comparison to RPF and GPF. The major improvement in terms of robustness is given for systems with marginal system and measurement noise affecting degeneracy.

Kernel Particle Filter

The particles are mapped into regions where the frequency distribution dominate, based on the gradient information, which is obtained from the kernel density estimation of the posteriori density. Based on the weighted particle swarm $\{x_k^{(i)}, \tilde{w}_k^{(i)}\}_{i=1}^N$, the re-sampling step is followed by a RRS, mapping each particle towards $x_k^{(i)*} = x_k^{(i)} + \lambda_{\min} \mathbf{S}^x \epsilon^i$, where the modes of the kernel density estimation correspond to the zero points of the gradients such as $\nabla p(x_k|Y_k) = 0$. Mean-shift mode seeking (MSMS) is a popular mode detection algorithm, or locating the maxima of a density, based on non-parametric method which is independent from a-priori information, where each sample points are moved towards the modes such as

$$m(x_k^{(i)}) = \frac{\sum_{j=1}^N K_\lambda(x_k^{(i)} - x_k^{(j)}) \tilde{w}_k^j x_k^{(j)}}{\sum_{j=1}^N K_\lambda(x_k^{(i)} - x_k^{(j)}) \tilde{w}_k^j} \quad (5.198)$$

The step is followed by re-sampling with

$$w_k^{(i),(j)} = \frac{p(x_k^{(i),(j)}|Y_k)}{\sum_{j=1}^N K_\lambda(x_k^{(i),(j)} - x_k^{(g),(j)})} \quad (5.199)$$

with the $x_k^{(\cdot),(j)}$ representing the particle distribution at time k and j -th mode detection algorithm iteration with

$$p(x_k^{(i),(j)}|Y_k) \propto p(y_k|x_k^{(i),(j)}) \sum_{g=1}^N p(x_k^{(i),(j)}|x_k^{(g),(j)}) w_k^{(g)} \quad (5.200)$$

newly weighted and mapped with MSMS. Other researcher further developed the estimation by combining advantages of ASIR and KPF to a hybrid estimator.

Adaptive Particle Filter

The earlier developed GPF profits especially during the initialization of a high value of particles. The initial density is in general fat and multi-modal. The shape changes over time with further observations. To improve the robustness and efficiency, later, the adaptive particle filter (APF) was introduced. This method allows on-line adjustment of the sample size thorough filtering processes. Mainly two methods are proposed. First criterion is based on the likelihoods, under the assumption that if the particles match the real posteriori with a certain accuracy. Second method uses statistic bounds on the approximation error of the true posterior density. The distance of two densities is described by the KULLBACK-LEIBLER (KL) divergence and approximated by [410]

$$\text{KL}(\pi||p) = \mathbb{E}\left[\log \frac{\pi(x)}{p(x)}\right] \approx \frac{1}{N} \sum_{i=1}^N \log \frac{\pi(x_k^{(i)})}{p(x_k^{(i)})} \quad (5.201)$$

$$= -\frac{1}{N} \sum_{i=1}^N \log w_k^{(i)} \quad (5.202)$$

under the premise that $\pi(\cdot) = p(\cdot)$ and $w_k^{(i)} = 1$. KL can still further simplified when assuming that $\pi(x_k^{(i)})$ is replaced with the transition prior, resulting in

$$-\sum_{i=1}^N \log w_k^{(i)} = -\sum_{i=1}^N \log w(x_k^{(i)}|x_{k-1}^{(i)}) - \sum_{i=1}^N \log p(Y_k|x_k^{(i)}) \quad (5.203)$$

$$\text{for normalized weights} = \text{KL}(\pi||p) \approx -\frac{1}{N} \sum_{i=1}^N \log \tilde{w}_k^{(i)} \equiv N_{KL} \quad (5.204)$$

Therefore, KL is tracking the goodness of the proposal distribution, which means that if KL reduces, the particle set can stay constant or also be reduced, if KL increases, the samples shall also increase. Therefore, the particles are controlled in such a way that the threshold is not violated. The KL determination increases the computational cost of the particle filter, hence, on-line application is doubtful.

Gaussian Particle Filter

The weight degeneracy problem is affected by the particle moving towards areas with higher likelihood, in general, a poor proposal distribution. Choosing the transition density as proposal distribution, fulfills the IS requirements, but under the premise that current measurements are not considered and favors degeneracy which may violate the Monte Carlo framework. It needs to be pointed out that robustness and accuracy of particles filter is defined by the proposal distribution, which considers current observations. Gaussian processes with inference algorithms approximate posteriori distributions over functions based on observations and is a method to reduce the problem of degeneracy. Gaussian filter frameworks, such as EKF, IEKF, UKF and more generate a distribution of the posteriori density of each particle

$$q(x_k|X_{k-1}, Y_k) \stackrel{\circ}{=} p(x_k|x_{k-1}, y_k) \quad (5.205)$$

$$\approx q_{\mathcal{N}}(x_k|Y_k) \quad (5.206)$$

$$= \mathcal{N}(x_k; \bar{x}_k^{(i)}, \mathbf{P}_{k|k}^{x, (i)}), \quad (5.207)$$

for $i = 1, \dots, N$, from which the proposal distribution is generated. One of the KALMAN filter frameworks is integrated which estimates for individual particles $x_{k-1}^{(i)}$ the new particles $x_k^{(i)}$ by estimating the mean and covariance. This method allows a variety of hybrid systems, which are more robust than the generic particle filter at the cost of higher computing power.

Gaussian Sum Particle Filter

For Gaussian Sum Particle filter the initial distribution $p(x_0)$ is represented as a Gaussian mixture. The aim is to obtain from noisy signals the corresponding filtering and predictive densities as a Gaussian mixture which is updated and approximated recursively [423][380].

$$p_{\text{GMM}}(x_{k-1}|Y_{k-1}) = \sum_{g=1}^G \alpha_{k-1}^{(g)} \mathcal{N}(x_{k-1}; \bar{x}_{k-1}^{(g)}, \mathbf{P}_{k-1}^{x, (g)}) \quad (5.208)$$

$$p_{\text{GMM}}(\kappa_{k-1}) = \sum_{m=1}^M \beta_{k-1}^{(m)} \mathcal{N}(\kappa_{k-1}; \bar{\kappa}_{k-1}^{(m)}, \mathbf{P}_{k-1}^{\kappa, (m)}) \quad (5.209)$$

The samples are passed through the non-linear system function $\mathbf{F}$ such as [424]

$$x_{k-1}^{(g),(i)} \sim \mathcal{N}(x_{k-1}; \bar{x}_{k-1}^{(g)}, \mathbf{P}_{k-1}^{x,(g)}) \quad (5.210)$$

$$\kappa_{k-1}^{(m),(i)} \sim \mathcal{N}(\kappa_{k-1}; \bar{\kappa}_{k-1}^{(m)}, \mathbf{P}_{k-1}^{\kappa,(m)}) \quad (5.211)$$

$$x_{k|k-1}^{(gm),(i)} = \mathbf{F}(x_{k-1}^{(g),(i)}, u_{k-1}, \kappa_{k-1}^{(m),(i)}) \quad (5.212)$$

$$\text{weighting } w^{(gm)} = \frac{\alpha_{k-1}^{(g)} \beta_{k-1}^{(m)}}{\sum_{gm=1}^{GM} \alpha_{k-1}^{(g)} \beta_{k-1}^{(m)}} \quad (5.213)$$

with the priori distribution

$$p_{\text{GMM}} = \sum_{gm=1}^{GM} w^{(gm)} \mathcal{N}(x_k; \bar{x}_{k|k-1}^{(gm)}, \mathbf{P}_{k|k-1}^{x,(gm)}) \quad (5.214)$$

with

$$\bar{x}_{k|k-1}^{(gm)} = \frac{1}{N} \sum_{g=1}^G x_{k|k-1}^{(gm),(i)} \quad (5.215)$$

$$\mathbf{P}_{k|k-1}^{x,(gm)} = \frac{1}{N} \sum_{g=1}^G (x_{k|k-1}^{(gm),(i)} - \bar{x}_{k|k-1}^{(gm)}) (x_{k|k-1}^{(gm),(i)} - \bar{x}_{k|k-1}^{(gm)})^T \quad (5.216)$$

resulting in a Gaussian mixture model (GMM). Originating from the Gaussian mixture approximation, the MMSE approach for state estimation can be applied. This is achieved by the fact that particles are drawn from the proposal distribution allowing point-mass approximation, the weighting step considers the latest observation with which one can extract the posteriori density $p(x_k|Y_k)$.

Gaussian Mixture Sigma-Point Particle Filter

Still, the aim of this work is to evaluate different recursive estimator in terms of accuracy, robustness and feasibility. Generally, particle filters are computationally expensive, therefore, further improvements are investigated. Particle filters may suffer from sample depletion, which may cause the posterior approximation to reduce to a small quantity. Previously we discussed techniques which address this problem by shifting particles to parts of the modes with higher likelihood. Applying Gaussian inference algorithms such as SPKF will generate the proposal distribution [390]. This burdens the need of high computing power, since for each sample in the posteriori state distribution the SPKF has to be executed multiple times. Further improvements are necessary, the Gaussian Mixture Sigma-Point Particle filter (GMSPPF) which shall deliver at least similar estimation performance in comparison to GPF, with the advantage of requiring less computing power.

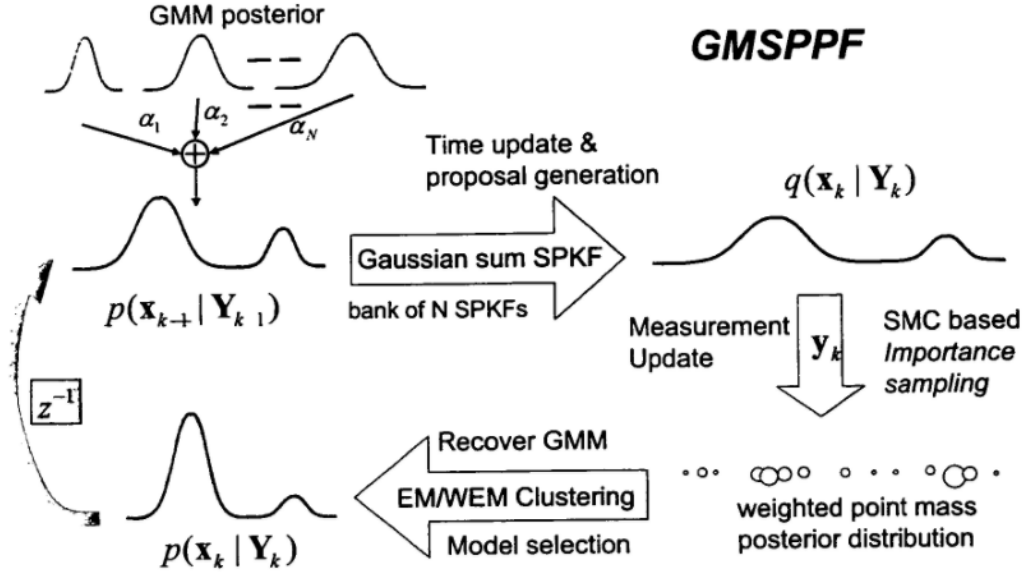


Figure 5.7: Illustration of the Gaussian mixture sigma-point particle filter [390].

It utilizes the framework of IS based measurement update in combination with SPKF based Gaussian sum filter for time update and defining π [390]. The idea is, that for the Gaussian mixture Sigma-Point Particle filter GMM approximation $p_{\text{GMM}}(x_k | Y_k)$ is generated by a finite set of Gaussian mixture models by e.g. a filterbank of extended KALMAN filter, to represent the posteriori filtering density. The proposal distribution is chosen in that manner

$$q(x_k | X_{k-1}, Y_k) \stackrel{\circ}{=} p_{\text{GMM}}(x_k | Y_k) \quad (5.217)$$

The posterior density is extracted by means of the weighted point-mass approximation from the posterior particle set during the IS. Since the setup of the GMSPPF is characterized by representing the posterior by GMM, which has to be recovered from the equally weighted, resampled particle set. Two methods, depending on the nature of the problem, might be applied:

- Resample-then-Expectation-Maximization (EM) algorithm:
- Weighted Expectation-Maximization (WEM) algorithm:

On the one hand, re-sampling is required to restrict the variance of the samples to increase too quickly, on the other hand re-sampling can play a part in particle depletion when the measurement likelihood shows clear peakedness. In this case the WEM may lead to better results. In applications, where the KL parameter between true posterior and GMM proposal is large, the pure EM algorithm is preferable [403].

(Adaptive) Rejection Particle Filter - Since rejection sampling is possible to generate

exact samples from the posterior distribution, some authors state that rejection sampling is preferable to importance sampling. The rejection particle filter deals with the problem that RS is not able to resolve the update step recursively. It finds a sequential proposal for solution and was developed by RANIZAKI [387]. The general idea is that there is no need not evaluate the integration when extracting the particles from the filtering density

$$p(x_k|Y_k) = \frac{1}{C_k} \int p(Y_k|x_k)p(x_k|x_{k-1})p(x_{k-1}|Y_{k-1})dx_{k-1} \quad (5.218)$$

$$= \frac{1}{N} \sum_{i=1}^N \frac{C_k^{(i)}}{C_k} \frac{p(Y_k|x_k)p(x_k|x_{k-1}^{(i)})}{C_k^{(i)}} \quad (5.219)$$

$$= \sum_{i=1}^N \lambda_k^{(i)} \frac{p(Y_k|x_k)p(x_k|x_{k-1}^{(i)})}{C_k^{(i)}} \quad (5.220)$$

with C_k representing the normalization constant given as $C_k \approx \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N p(Y_k|x_{k|k-1}^{(ji)}) \equiv \hat{C}_k$

and for the individual samples $\sum_{j=1}^N \sum_{i=1}^N p(Y_k|x_{k|k-1}^{(ji)}) \equiv \hat{C}_k^{(i)}$. Then the filtering density function will be approximated based on the weights $C_k^{(i)}/N_k \cdot C_n^{(i)} = \lambda_k^{(i)}$. For the rejection particle filter, the acceptance probability is defined as

$$\alpha(z) = \frac{\frac{p(Y_k|z)p(z|x_{k-1}^{(i)})\pi(z)}{\pi(z)}}{\sub_z \{p(Y_k|z)p(z|x_{k-1}^{(i)})\pi(z)\}} \quad (5.221)$$

The acceptance probability is a measure for the sample set generation, for higher values the required sample set is generated quicker. There are two choice of an appropriate proposal distribution π , either using the transition density function $p(x_k|x_{k-1})$ or a (Gaussian) mixture distribution, but in any event the variance of the proposal distribution need to overlap the posterior function. Since the computing time depends directly on the acceptance parameter, its real-time capability is unpredictable.

Rao-Blackwellization Particle Filter - An additional way to decrease computational cost is to reduce the complexity of individual sample, e.g. reducing the sampling variables e.g. from 5 to 3, which results in a reduction of system dimension. The Rao-Blackwellizaion Particle Filter (RBPF) [425] is based on that idea and identifies these variables, which are unnecessary to be sampled. The model will be evaluated partly analytical, then only the remaining part of the filtering has to be evaluated by particle, known as the *tractable substructure*. The more the system can be broken down into substructures, the greater is the performance increase in comparison to the GPF and its full model. A substructure of the state space vector can be defined as $x_k = \{x_{1,k}; x_{2,k}\}$ for which $p(x_{2,k}|X_{1,k}, Y_k)$ can be determined. This layout enables to apply a PF on $X_{1,k}$ while the particles $x_{2,k}$ are updated by $p(x_{2,k}|X_{1,k}, Y_k)$. Because RBPF is favorable for

systems which states evolve in a linear Gaussian manner, $p(x_{2,k}|X_{1,k}, Y_k)$ is Gaussian and a filter bank of N KALMAN filter to update particles of state $x_{2,k}$. This would lead to N KALMAN filter iterations at each step and N samples. Even though, there are more operations needed per particle in comparison to the GPF, one should take into account that the dimensions are reduced which nevertheless leads to an overall reduction of computing cost. As a positive side effect, the variance of the difference between estimate and true value is reduced.

5.4 Estimation Framework - Systems, Problem Statement and Implementation

5.4.1 Different Estimation Problems

In this section the different estimation formulations in question are discussed. Estimation problems depend on the investigated system and required specifications. The following estimation problems can be differentiated in (simultaneous) state and parameter estimation:

State Estimation [2]: A state of a system is defined by a (set of) variable(s) or a vector that gives the necessary and complete representation of the unknown, unobservable internal conditions or status of the discrete, dynamic system at any given time step, extracted from noise observations. In general, that is the most frequent task in estimation problems, because this information is interesting itself. The system is described by a (non-)linear, continuous/discrete state-space model. Normally, the physical systems are described in continuous time, but state estimation algorithms are executed on embedded controller and therefore implemented in discrete time representation. x_k represents the observed states of the system, y_k the input of measurements, and u_k the control input used of control systems. Since this work concentrates on estimation of non-controlled system, it is not further considered. The system noise κ influences the system dynamics, while the measurements are superimposed by the measurement noise γ . The (non-)linear functions $\mathcal{F}$ and $\mathcal{H}$ describe the system dynamics and measurement equation respectively and maybe parametrized by a known and constant set of parameters [426][380].

Parameter time-invariant estimation [2]: Parameter estimation defines and provides the solution of the process of using measurements to estimate the parameters of the selected distribution. For the parameter identification, which is referred as the parameter estimation of *time-invariant parameters*, describes the task of determine the non-linear, multi-dimensional static system defined as

$$y_k = \text{DSS}(\theta, [u_k, \tilde{x}_k]) \quad (5.222)$$

with the known state $\tilde{x}_k$. θ describes the discrete static system **DSS**. The identification finds the parameter in an optimum way, the introduced estimation algorithms may be applied. The corresponding state-space is defined as

$$\theta_k = \theta_{k-1} = \mathcal{F}(x_{k-1} = \theta_{k-1}, u_{k-1} = 0, \kappa_{k-1} = 0) \quad (5.223)$$

$$y(k) = \mathbf{DSS}(\theta, [u_k, \tilde{x}_k]) + \gamma(t_k) = \mathcal{H}(x_k = \theta_k, u_k, \gamma_k) \quad (5.224)$$

where the parameters are described by stationary processes. The predictive expectation is constant while the parameter is adapted during the filtering step. y_k may follow a non-linear observation parameterized by θ . The systems states are known and assigned to the control inputs.

Parameter time-variant estimation [376] In many systems, parameters varying during the life time, taking batteries as an example, due to aging effects during lifetime and diffusion processes, parameters cannot be described as time-invariant. To apply adaptive algorithms, the state space model has the form

$$\theta_k = \theta_{k-1} + \tilde{\kappa}_{k-1} = \mathcal{F}(x_{k-1} = \theta_{k-1}, u_{k-1} = 0, \kappa_{k-1} = \tilde{\kappa}_{k-1}) \quad (5.225)$$

$$y(k) = \mathbf{DSS}(\theta, [u_k, \tilde{x}_k]) + \gamma(t_k) = \mathcal{H}(x_k = \theta_k, u_k, \gamma_k) \quad (5.226)$$

The state transition is in comparison to the time-invariant system replaced by a random, stochastic process. By modifying the time-invariant equations, the change of parameters can even be tracked during the filter prediction step. The adaptive filter is therefore sensitive against changes in the system dynamics and its parameters. Regarding system noise, this area is still heavily discussed in the literature. So far, most promising methods (λ -Decay, Dobbins-Monro) are based on the idea, that initial system noise is set to a constant value, which will be updated on-line [426][380].

Dual estimation [427] Essentially, dual estimation uses two cooperating filters, e.g. extended KALMAN filter which are running concurrently. At every time step, the corresponding state estimator generates an estimation based on the current model estimate. The second filter estimates the weights/parameters using the current state estimate based on the scalar measurements y_k . The parameter estimator uses a model of dynamics and model output. Often, especially in battery systems engineering with model-based monitoring, it is compulsively necessary to couple both parameter and states estimation simultaneously. Depending on the mathematical innovation approach, the augmented state vector $x_{\text{AUG}} = [x \ \theta]^T$ is represented as a discrete state space model. Similar to the pure parameter estimation case, the model parameters are time-variant and defined as [380]

$$x_k = \theta_{k-1} x_{k-1} \quad (5.227)$$

$$x_{\text{AUG}} = [x \ \theta]^T \quad (5.228)$$

and the general state-space model

$$x_k = \mathcal{F}(x_{k-1}, u_{k-1}, w_{k-1}, \theta_{k-1}) \quad (5.229)$$

$$y_k = \mathcal{H}_k(x_k, u_k, v_k, \theta_{k-1}) \quad (5.230)$$

In this framework, due to decoupled state from parameter vector, any information regarding cross-correlations which are causing changes in state or parameters cannot be tracked. Time-invariant quantities are placed within $\mathcal{F}$ or $\mathcal{H}$ and excluded from the parameter vector [2]. This leads us to revised mathematical description for the system dynamics [426][380]

$$\theta_k = \theta_{k-1} + r_{k-1} \quad (5.231)$$

$$y_k = \mathcal{H}_k(f_{k-1}(x_{k-1}, u_{k-1}, \bar{w}_{k-1}, \theta_{k-1}), u_k, e_k, \theta_{k-1}) \quad (5.232)$$

It makes it possible to transform nearly all linear estimation problems to non-linear problems. Therefore, the introduced methods to approximate the RBET can be applied to the parallel parameter and state estimator. The corresponding cross-covariance between parameter and state variables defined by [2]

$$\mathbb{E}[(\hat{x}_{\text{AUG},k} - \mathbb{E}[\hat{x}_{\text{AUG},k}])(\hat{x}_{\text{AUG},k} - \mathbb{E}[\hat{x}_{\text{AUG},k}])^T] = \begin{bmatrix} \mathbf{P}_k^x & \mathbf{P}_k^{x,\theta} \\ \mathbf{P}_k^{\theta,x} & \mathbf{P}_k^\theta \end{bmatrix} \quad (5.233)$$

With the augmented description of the problem, it is possible to evaluate and investigate the uncertainties of both unknowns and their correlation and the EKF maybe applied. The augmented state vector is initialized with a valid guess for $\mathbb{E}[x_0]$, $\mathbb{E}[\theta_0]$ and error covariance and updated by the standard equations.

Joint estimation In joint estimation, the state and parameter vector are combined, which is estimated together at the same time. In comparison to the dual estimation framework, the system state/parameter matrix will become large eventually for more complex battery models (e.g. 2 RC Thevenin). Hence, it involves costly matrix calculations, additionally, different time scales of states and parameters may lead to poor results. Its implementation into augmented state matrix is given as

$$\begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \mathcal{F}_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}, \theta_{k-1}) \\ \theta_{k-1} + r_{k-1} \end{bmatrix} \quad (5.234)$$

$$y_k = \mathcal{H}(x_k, u_k, \theta_k) + v_k \quad (5.235)$$

After the augmented model of system state dynamics including parameter are derived, it can be applied to a nonlinear estimator. The properties with its most relevant questions around estimators for this application shall be answered for the battery. Therefore, before choosing a filter for a battery management system, the properties must be analyzed. General statements for filters are difficult to extend to its modified versions, however, it is required when discussing a great variety. Therefore, following questions need to be considered when assessing filter choices:

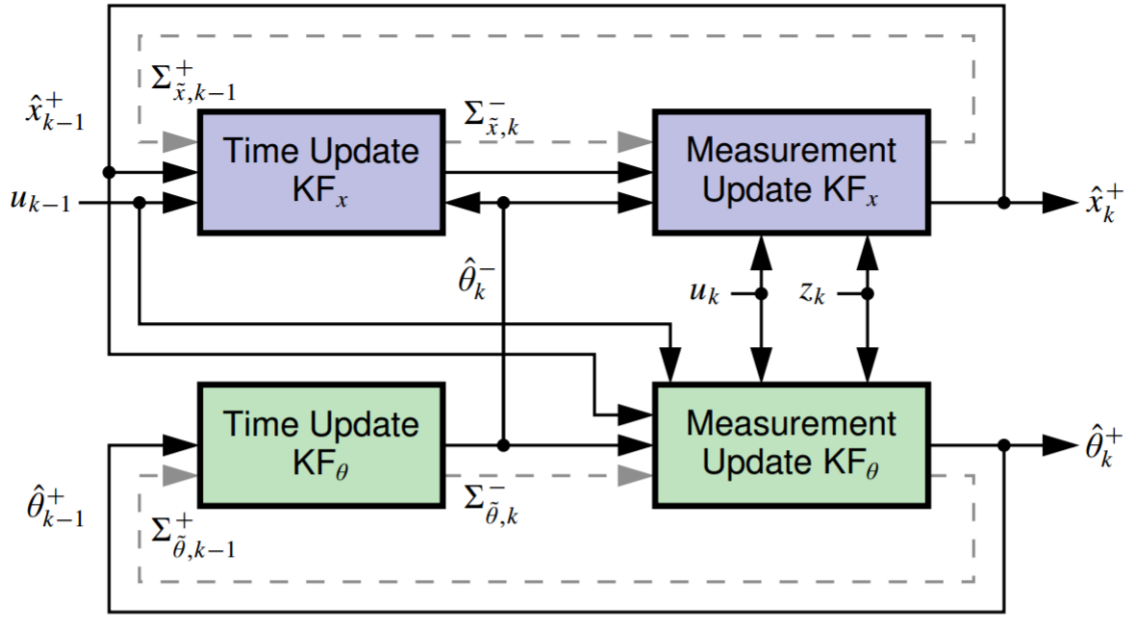


Figure 5.8: Dual estimation of state and parameter quantities using two KALMAN filters running in parallel including data sharing. [2]

- (a) understand the battery dynamics in terms of aging;
- (b) hidden dynamic states must be observed while considering measurement noise;
- (c) for good filter parameterization, knowledge of process and measurement noise covariances are one requirement;
- (d) on-line identification to counterattack model inaccuracies caused by model reduction and change of system behavior.
- (e) recursive or batched based problem
- (f) robustness and convergence towards meaningful values
- (g) equality and inequality constraints handling
- (h) low complexity on derivatives or need for Jacobians or Hessians
- (i) does system- and sensor noise to be considered; if so, what is the underlying noise process (Gaussian / non-Gaussian)
- (j) stochastic, deterministic or optimization problem statement
- (k) robust numerical algorithms available

- (1) feasibility of implementation, complexity handling and robust to uncertainties

5.4.2 Methods for Implementation

Choice of artificial system and measurement noise

The choice of system noise $\tilde{\kappa}$ is a critical parameter, since it cannot be universally applied to different inference methods and systems. There are methods such as the Probbins-Monro or λ -Decay which considers not just a constant system noise, rather a system noise which is adapted on-line. Under the assumption, that the noise for the measurement and system are strictly additive, the state space model for discrete systems can be simplified as following [426][380]

$$x_k = \mathcal{F}(x_{k-1}, u_{k-1}) + \kappa_{k-1} = \mathcal{F}(x_{k-1}, u_{k-1}, \kappa_{k-1}) \quad (5.236)$$

$$y_k = \mathcal{H}(x_k, u_k) + \gamma_k = \mathcal{H}(x_k, u_k, \kappa_k) \quad (5.237)$$

Square-root methods for Gaussian filters

To improve the basic KALMAN filter properties, besides ensuring symmetric, positive-definite covariance matrices or reduced computational complexity, the filter is sensitive to finite word length in regards to converging to optimal solution [2]. By using square roots, the required bits for representing numbers in a digital computer can be reduced by using square root of numbers. If we consider the CHOLESKY factorization, the covariance is defined as

$$\mathbf{P}^x = \mathcal{S}^x (\mathcal{S}^x)^T \quad (5.238)$$

$$\mathbf{P}^\kappa = \mathcal{S}^\kappa (\mathcal{S}^\kappa)^T \quad (5.239)$$

The aim is to compute the covariance time and measurement updates in terms of the CHOLESKY factorization, it secures the positive definite covariance matrix [426][380].

Local methods

The CHOLESKY factorization $\mathcal{S}_{k,k-1}^x$ is not quadratic, which is needed for the filtering step. The QR decomposition calculates two quantities, $Q \in \mathcal{R}^{n \times n}$ for a matrix $Z \in \mathcal{R}^{n \times m}$ in a way that $Z = QR$, Q is orthogonal, R is upper-triangular and $m \geq n$. [2] If there is a upper-triangular portion of R , then its transpose is the CHOLESKY factor of $\Sigma = M^T M$. If the upper-triangular portion of R is returned after QR decomposition, then lower-triangular

CHOLESKY factor is calculated by MM^T [2] resulting in

$$\mathcal{S}_{\tilde{x},k}^- = \text{qr} \left([A_{k-1}\mathcal{S}_{\tilde{x},k-1}^+, \mathcal{S}_{\tilde{w},k}]^T \right)^T \quad (5.240)$$

and transforms the $\mathcal{S}_{k,k-1}^x$ in quadratic form. Now it is also possible to calculate the covariance for the filtering step in square-root form and can be applied to almost all KF variants.

Quadratur methods

In the SR-UKF as well as CDFK SR-methods approaches were introduced in which the covariance matrix $\mathbf{P}^x$ is also positive definite. The CHOLESKY decomposition (CF) is computed in each iteration to extract the deterministic sigma-points. The requirements on computing power are very costly. However, roundoff errors can cause the computed value of $\mathbf{P}^x$ may have negative values and become unstable and effect the estimation error. A new approach is proposed in [372] where the computation is avoided. The improvement is based on the idea, that the covariance matrix $\mathbf{P}^x$ are passed through the non-linear function and updated rather than having to calculating it for each iteration, and also has no matrix inversions to calculate. The update of the CF is based on linear algebra algorithms, QR-decomposition, CF updating and least-square methods [380].

1. QR decomposition

The QR decomposition leading to the factorization of a matrix $\mathbf{A}$ is given by

$$\mathbf{A}^T = \mathbf{Q}\mathbf{R} \quad (5.241)$$

[428] where $\mathbf{Q} \in \mathbb{Q}^{N \times N}$ is orthogonal, $\mathbb{R}^{N \times L}$ is upper triangular with $N \geq L$. Looking at the upper triangular part $\tilde{\mathbf{R}}$, represents the transpose of the CF of $\mathbf{P} = \mathbf{A}\mathbf{A}^T$ respectively $\tilde{\mathbf{R}}^T \tilde{\mathbf{R}} = \mathbf{A}\mathbf{A}^T = \mathbf{P}$

2. CHOLESKY factor updating for state estimation

$\mathbf{S}$ represent the lower triangle CHOLESKY factor of $\mathbf{P} = \mathbf{A}\mathbf{A}^T$, this results into the CF of the rank-1 update to

$$\check{\mathbf{P}} = \mathbf{P} \pm \sqrt{\nu} \mathbf{u}\mathbf{u}^T \quad (5.242)$$

If $\mathbf{u}$ is a matrix, then M consecutive updates of CF are generated and M columns of $\mathbf{u}$.

3. Efficient least squares

The solution to the least-squares problem is given as

$$(\mathbf{A}\mathbf{A}^T)x = \mathbf{A}^Tb \quad (5.243)$$

$$\equiv \mathbf{A}x = b \quad (5.244)$$

$$\equiv x\mathbf{A} = b \quad (5.245)$$

This overdetermined system can be solved applying the QR-decomposition as described for the SR-SPKF state estimation, here during the linearization the CHOLESKY factorization (taking matrix square-root) of the state vector is determined, further passed and updated to generate the sigma-points [428][380]. Step $\mathbf{S}_{k|k-1}^x$ is determined applying the QR decomposition of the augmented matrix including the weighted propagated sigma-points $\mathcal{X}_{k|k-1}^x$ and the calculated square-root values of the process noise covariance $\mathbf{S}^\kappa$. This measure is not required for all SP family filter, exclusively for filters in which during the first iteration is a negative weight calculated, such as for the UKF. Hereby, the time-update of $\mathbf{P}_{k|k-1}^x$ was replaced by the mentioned improving steps, which are also applied to calculate the CHOLESKY factor of the error covariance $\mathbf{P}^{\tilde{y}}$, $\mathbf{S}_{k|k-1}^{\tilde{y}}$. The efficient least-squares method are used to derive the KALMAN gain factor [380].

$$\mathcal{K}(\mathbf{S}_k^{\tilde{y}}(\mathbf{S}_k^{\tilde{y}})^T) = \mathbf{P}_k^{xy} \quad (5.246)$$

$$\mathcal{K} = (\mathbf{P}_k^{xy}/(\mathbf{S}_k^{\tilde{y}})^T)/\mathbf{S}_k^{\tilde{y}} \quad (5.247)$$

Since $\mathbf{S}_k^{\tilde{y}}$ is triangular and quadratic, additional linear algebra simplifications may apply to further reduce the computing power, e.g. by transforming the equation of the KALMAN gain $\mathcal{K} = (\mathbf{P}_k^{xy}/(\mathbf{S}_k^{\tilde{y}})^T)/\mathbf{S}_k^{\tilde{y}}$ to a non-inversion problem. The CHOLESKY factor $\mathbf{S}_{k|k-1}^x$ of the posteriori state covariance matrix $\mathbf{P}_{k|k}^x$ is calculated in the end. SR-UKF provides improved numerical stability since it guarantees positive semi-definiteness of the underlying state covariance.

4. CHOLESKY factor updating for the parameter estimation

In general, the SR parameter estimation framework follows the state estimation implementation for the UKF. The knowledge of the linear state transition function can be utilized at the time-update step of the state covariance $\mathbf{P}_{k-1}^w = \mathbf{P}_k^{w_{k-1}} + \mathbf{R}_{k-1}^r$. Using the weighting factor of the least-squares method, the process noise covariance is $\mathbf{R}_{k-1}^r = (\lambda_{\text{RLS}}^{-1} - 1)\mathbf{P}_k^{w_{k-1}} = \lambda_{\text{RLS}}^{-1}\mathbf{P}_k^{w_{k-1}}$ [428][380]

Filter Concluding Remarks

The Bayesian framework for recursive inference is conceptually well understood, from observing measurements y_k from a sampling density $f(y|\Theta)$ where Θ may represent a vector of parameters or states with a prior $g(\Theta)$, then the posteriori density of Θ is proportional to $g(\Theta|y) \propto g(\Theta)f(y|\Theta)$. The posterior summaries are defined in terms of integrals

which need to be evaluated in a numerical way. Quadrature methods have limited use for Bayesian integration due to exponential increase in computation vs. increase of Θ dimension. Therefore, in this chapter, the focus lies on computational methods, used for computing high-dimensional Bayesian integrals, by approximation in Bayes recursive estimation theory in any way related to the corresponding probability density or the underlying structure [429] The approximation of the posterior densities requires complex algorithms because it is extremely difficult to calculate the true density function. Approximate Bayesian computation directly approximates the posterior distribution, which will be able to be implemented as an estimator. The accuracy, robustness and computational complexity of the corresponding approximation method is different of each inference, and it can be classified into application specific appropriate solutions and non-real-time functionality. The aim is, to determine appropriate filter for on-line monitoring estimation in battery management systems. As it turns out algorithms can be partly neglected beforehand, since practical implementation and computational overload prohibits the use. Therefore, the groups of following approximation methods shall be evaluated [380]:

1. GAUSS approximation methods
2. Gaussian Sum Approximation methods
3. Sequential MONTE-CARLO methods
4. Hybrid methods

This chapter started with the introduction of the theoretical solution to the stochastic and probabilistic theory. Its framework is determined by the recursive BAYES estimation theory (RBET). As shown, this framework only provides exact solution for linear Gaussian systems and can be solved by the linear KALMAN filter.

It is necessary to show, how RBET can be solved also for non-linear systems. At first, the solution to the Gaussian representation for which case the densities of RBET are normalized. In this case, the first order TAYLOR-series expansion is applied to linearize the non-linear system function in the operation point. Now it is possible to apply the KALMAN filter equations to the linearized system leading to the local model approximations in terms of the extended KALMAN filter (EKF) and its revealed potential improvements such as IEKF.

A further approximation method consider the probability density instead of approximating the systems equations. It uses discrete sums, called the Quadrature-methods and applied in the Sigma-point KALMAN filter family. In comparison to the TAYLOR-series expansion methods, Quadrature-methods does not require to continuously recalculate the operation point linearization while offering second order approximation.

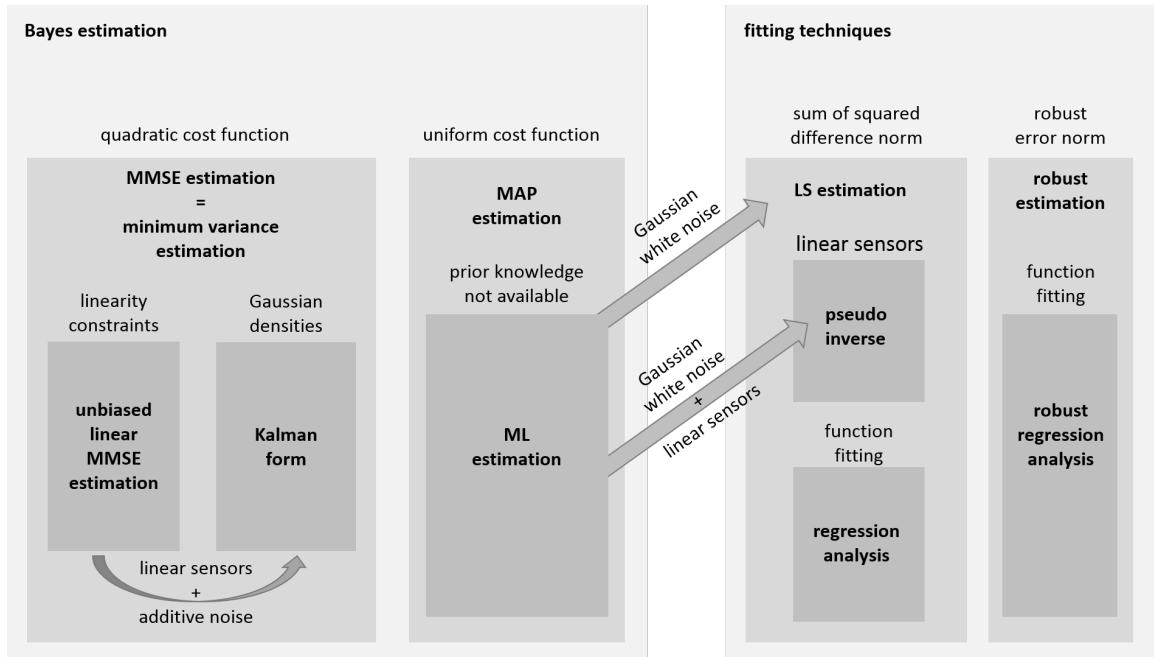
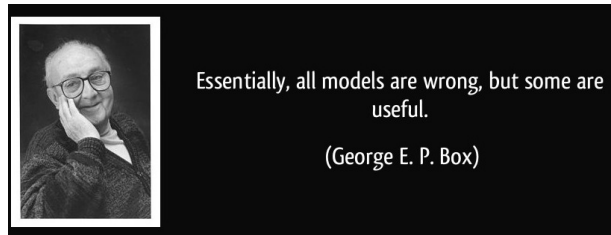


Figure 5.9: Summarizing Bayesian and classical methodology.

While the Gaussian systems assume Gaussian distribution for system and measurement noise, also non-linear dynamic systems with non-Gaussian system assumptions are discussed. Nonlinear Bayesian estimation using weighted Gaussian sum of probability density functions is able to closely approximate arbitrary density functions and define the group of Gaussian sum approximation. Later, sequential MONTE-CARLO methods are discussed. The basic idea to approximate the integrals of RBET, is to take discrete sums, over an adaptive grid-based map, where the sums are concentrated around the higher likelihoods of the state-space. SMC methods acting in a manner similar to SPKF, with the difference that the determination of the sample points, also referred as the particles, is carried out in a random way, not like the SPKF in a deterministic approach. Hybrid systems combine advantages of different methods to improve mainly the computational costs.

6 Evaluation Methodology for Bayesian Estimation and Machine Learning



It is critical for a safe operation of an electrified vehicle to monitor the battery's state-of-charge during operation. This quantity cannot be measured directly and has to be estimated by using different algorithms, which were introduced before (EKF, UKF, PF,...). For different applications, there are different types of lithium-ion batteries available, e.g. power cells (HEV) vs. energy cells (BEV) with different characteristics in electrical response behavior when excited with load current. It must be mentioned that different ECM are required for different cell technologies. There have been some comparative studies of estimators such as [430][431][432][433][434][435][436]. The authors focus on EKF, UKF, PF, Luenberger / sliding mode observers for specific battery model, while only in [431] the estimators are applied to different cell chemistries.

On the other hand, this work expands the library of non-linear filters beyond that ones compared in the literature so far. On top, machine learning algorithms are applied to health estimation, whose performance improves over time the more they are exposed to new data. The estimation approaches are verified using a Monte Carlo simulation as well as measurement data recorded in vehicle and lab to understand the problems around sensor and parameter uncertainty and estimator tuning parameters. The estimators are chosen based on the literature review discussed in Chapter 3.

In this chapter, the introduced estimators and its different battery reduced order equivalent circuit models shall be implemented. In the first steps, the models are generated and transformed from time-continuous system to discrete stochastic state-space models.

6.1 Battery State-Space Models

In general, systems can be represented as static, dynamic, (non)-linear, deterministic/stochastic, continuous/discrete time, time-(in-)variant parameters, lumped or distributed [362]. In this work, mathematical modeling for battery systems includes deterministic and stochastic processes. Since stochastic systems are difficult to implement, we apply the deterministic methods to describe the battery dynamic behavior imposed with additive system and measurement random noise. Also, it must be understood that no model will be able to fully describe all physical process within the battery cell, and imperfections, represented by uncertainties, therefore, results must be accepted. The system will also be affected by non-deterministic disturbances modeled as white Gaussian/non-Gaussian processes, and lastly, we have to deal with corrupted or missing data by the BMS sensor measurements (voltage, current, temperature...). In the most cases, also for battery systems, the dynamic system is described by differential equations. The target implementation is on the BMS, consisting of an ECU running the code. Hence, for implementing the algorithms and models on the target, the differential equations have to be converted into discrete time representation, which are described by linear coefficient difference equations in a sequential way. The differential equations are approximated by the difference equation to

$$x_k = f(x_{k-1}, u_{k-1}) \quad (6.1)$$

As described during the estimation theory chapter, the current state will only be derived from the previous state x_{k-1} , and the excitation of the battery as input (u_{k-1}) and the non-linear system model f . So, before estimators can be derived and implemented, we first discuss different state-space models (SSM) of various equivalent circuit battery models (ECBM).

6.1.1 Equivalent circuit/semi-empirical models

Batteries are highly non-linear electro-chemical devices with complex multi-time scale dynamics including temperature dependencies, electric and magnetic effects, double-layer formation, side reactions due to mass transport, SOC effects, and degradation effects (cycling & calendaric aging) etc [437][438], while these phenomena influence the dynamics at complete different time scales, starting at micro-seconds while others begin after years. This causes very mixed and superimposed interactions of chemical reactions and complex to model. As discussed earlier, the tasks of operating a traction battery in the most safe and optimal manner depends on its HW like measurement and controls electronics, but the key functions lays within the intelligent and adaptive BMS SW, hence, for the future next generation BMSs, as proposed and developed in this work, various battery models of varying complexity are needed for robust and precise battery performance prediction (SoC, SoH, SoF, FS...) considering all internal and external conditions accurate. In general, the different model architectures can be grouped as [438]:

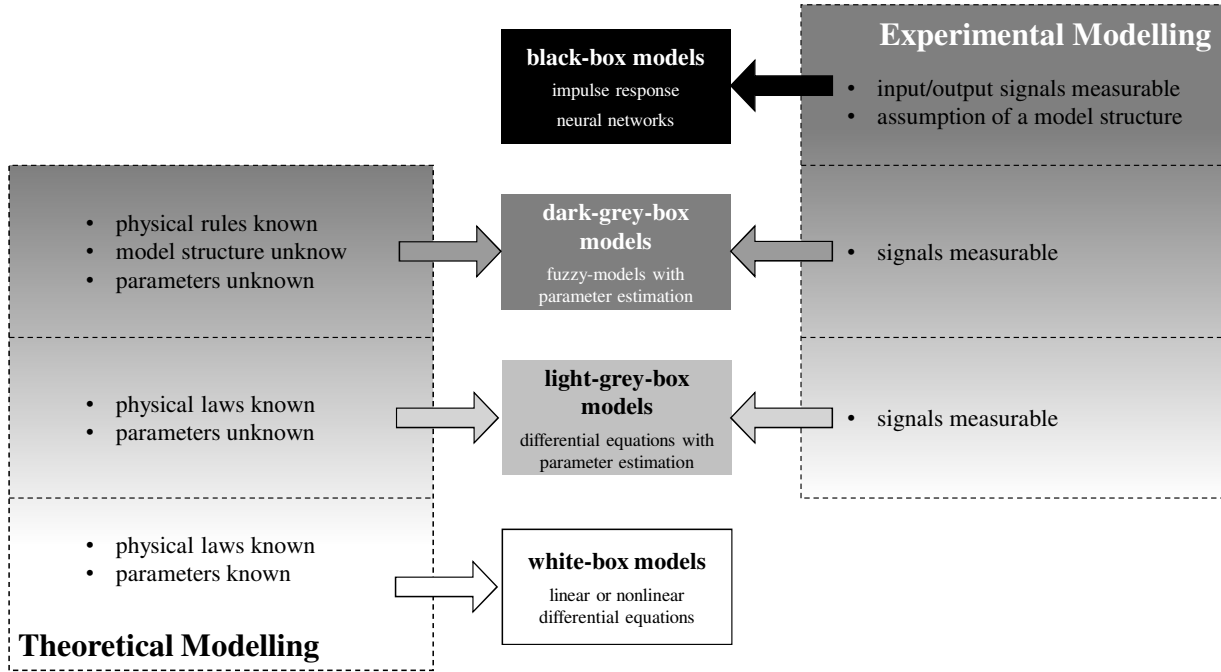


Figure 6.1: Comparison of different modeling approaches for battery systems

1. Ideal voltage source which offers unlimited power, e.g. used if dynamics and behavior are not of interest.
2. Full-Order ECMs i.e. electrochemically-inspired and
3. Reduced-Order ECMs i.e. approximate greybox/semi-empirical ones.
4. Black box models do not map underlying physical processes, e.g. Peukert's law.

Now, the mentioned models shall first briefly discussed and based on that the state-space model form is derived. Obviously, physico-chemical models give best results of the cell behavior in comparison to ECM models on the cost of very high computational demand, due to the system of coupled partial differential equations, which makes it very challenging for low-cost automotive ECUs, in terms of fulfilling real-time execution and insufficiently terms of controllability and observability. Such models may have up to hundred parameters, from which some can only be guessed. For on-line estimation of battery state, it can be assumed that it is sufficient to know the reaction and dependency of battery load current in regards to battery SoC, SoH, internal & external condition (temperatures...). Hence, ECMs are mostly used, different approaches have been presented in the literature with the idea to capture as much physical meaningful cell processes within the ECM. Even though the battery dynamics are multi-time scaled, nevertheless, the electrical I-V relationship can sufficiently be matched by using equivalent circuit models, namely impedance based models (full-order), and equivalent circuit models (reduced order).

Full-Order ECMs

Full-order equivalent circuit models are based on electro-chemical battery cell process considering polarization transients, electromagnetic interaction, mass and charge transport, migration and diffusion processes or formation of a passivating layer (SEI). These multi-scale and multiphysics processes taking place during charge/discharge influencing the battery impedance of the circuit in separate areas of the frequency response. The impedance models structure including Warburg Impedance (mass transport effects/diffusion) or ZARC elements to capture low/middle frequency processes [438]: These models are superior to the reduced order ECMs and are capable to better map the internal battery dynamics, at the cost that the models get also more complex in terms of fractional-order transfer functions due to ZARC and Warburg impedance elements and results in hurdles in estimation and control applications for next generation BMS SW. More accurate modeling and automatic control requires more robust modeling concepts and control strategies. Hence, full-order models are used on cell or module level system simulation, e.g. HIL, mostly coupled to electro-thermal models. With the possibility to investigate the off-line data from EIS measurements, tracking the change in the frequency domain by means of Nyquist graph changes in regrads to state of degradation and temperature, these models allow the estimation of State-of-Health or internal cell/pack temperature [438].

Reduced-Order ECMs

In terms of lower accuracy and computational demand, the reduced-order ECM models (n -order ECMs consisting of RC elements), are used and described by e.g. ODEs. Clearly, the motivation for such models lies within its on-line applicability due to their integration on HW efficiency for real-time task, but still re-mapping most of the physical processes (not supporting spacial resolution), such as adaptive on-line monitoring, estimation and control schemes, used in battery state and parameters estimation and among other thing safety related tasks (fault detection). Even though its increasing popularity for BMS application is noticeable, it also involves disadvantages. The model parameters have no physical meaningful representation i.e. they are not strictly mapping the parameters to a corresponding electro-chemical process occurring inside a cell rather following the mathematical optimization constraints (cost functions). In terms of estimating the cell temperature, lumped thermal models are used. Moreover, n -order ECMs are very versatile, the nature of parameterization makes it possible to adapt the model topology to new cell chemistries or devices with less effort.

Empirical Models

These models are based on simple semantic process, and can be of wide variety, such as based on expert intuitions knowledge, or derived from black-box relation (fitting measurement data to some model). Examples in battery technology are the Shepherd's model which describes I-V characteristics of a battery, Peukert's law approximates the change of capacity in SLI batteries for different C-rates. These models provide low accuracy especially for xEVs with varying operation conditions and customer usage. Additionally, since parameters also are not connected to physical processes their applicability to next generation BMS are not considered in this work. Nevertheless they are still good candidate for applications which require very low computational complexity. Non-realtime tasks, such as SoH tracking may still be a good application for their implementation.

6.1.2 State-Space Models for Battery State and Parameter Estimation

From a control engineering perspective, batteries ECMs are SISO, where the current is the input, and voltage is its response (e.g. temperature is treated as input variable which on which parameters depend). Deriving a transfer-function or state space model in the other side is very complex for a systems point of view, PDEs must be discretized to ODEs, thus system parameters need to be determined. From the infinite-dimensional equations, for observability, a finite-order model with only a few states must be derived. The extraction of many independent parameters is also non-trivial. A wide range of ECMs for Li-ion battery cell were developed and designed in the existing scientific paper or technical literature. In [2], it introduces several lumped models to reproduce the battery behavior under charging or discharging. For example the combined model, the simple model, the zero state hysteresis model, the one state hysteresis model and the enhanced Self-correcting (ESC) model. However the most commonly used ECMs in industry and also academic are first order RC model and second order RC model [439] [440] [441][442][443]. In [444], the author concluded that taking into account the accuracy, the complexity and the reliability of the models, first order RC model is preferred for LiNMC cells and if LiFePO₄ was used then first order RC model with one-state hysteresis will be the best choice instead. Here we will compare 4 different ECMs (RINT model, first order RC model, first order with current dependency and second order RC model with current dependency as well) performance with the combination of different adaptive filter technologies. The detailed model equations and corresponding circuit models are given below. Considering the targeting cell in our study is NMC cell in which hysteresis effect do not play a big role in the terminal voltage response, we will not take into account the hysteresis effect in all our presented models.

Equivalent Circuit Model 1: $OCV + R_I$

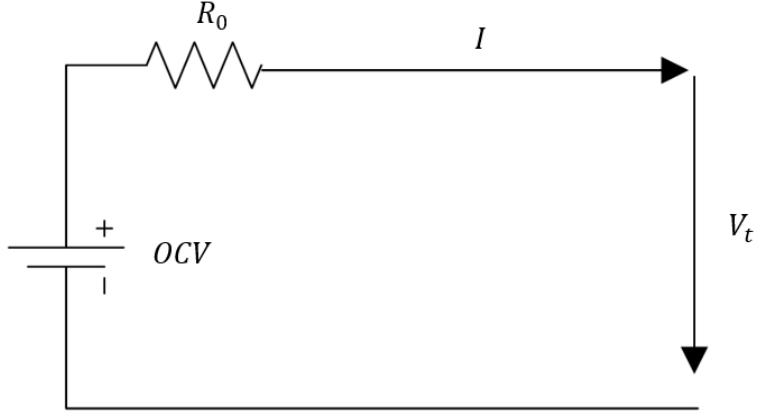


Figure 6.2: Simple ECM, considering only ohmic resistance.

States:

$$x(k) = \begin{pmatrix} SOC(k) \\ R_I(k) \end{pmatrix}$$

State equation:

$$SOC(k+1) = SOC(k) + \frac{\Delta t \cdot \eta}{C_N} \cdot I(k)$$

$$R_I(k+1) = R_I(k)$$

Output equation:

$$V(k) = OCV(SOC(k)) + R_I(k) \cdot I(k)$$

Matrices:

$$A_k = \left. \frac{\partial f}{\partial x} \right|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$C_k = \left. \frac{\partial g}{\partial x} \right|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} \frac{\partial OCV}{\partial SOC} & I \end{pmatrix}$$

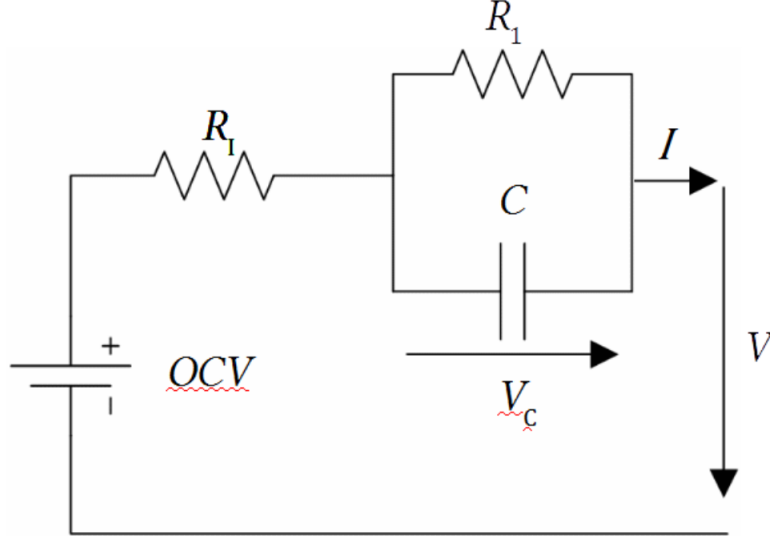
Equivalent Circuit Model 2: $OCV + R_0 + R_1C$ 

Figure 6.3: First-order ECM.

States:

$$x(k) = \begin{pmatrix} SOC(k) \\ V_C(k) \\ R_0(k) \\ R_1(k) \\ C(k) \end{pmatrix}$$

State equation:

$$\begin{aligned} SOC(k+1) &= SOC(k) + \frac{\Delta t \cdot \eta}{C_N} \cdot I(k) \\ V_C(k+1) &= V_C(k) \cdot \left(1 - \frac{\Delta t}{R_1 \cdot C}\right) + \frac{\Delta t}{C} \cdot I(k) \\ R_0(k+1) &= R_0(k) \\ R_1(k+1) &= R_1(k) \\ C(k+1) &= C(k) \end{aligned}$$

Output equation:

$$V(k) = OCV(SOC(k)) + V_C(k) + R_0(k) \cdot I(k)$$

Matrices:

$$A_k = \frac{\partial f}{\partial x} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 - \frac{\Delta t}{R_1 \cdot C} & 0 & \frac{V_C \cdot \Delta t}{R_1^2 \cdot C} & \frac{V_C \cdot \Delta t}{R_1 \cdot C^2} - \frac{\Delta t \cdot I}{C^2} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$C_k = \frac{\partial g}{\partial x} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} \frac{\partial OCV}{\partial SOC} & 1 & I & 0 & 0 \end{pmatrix}$$

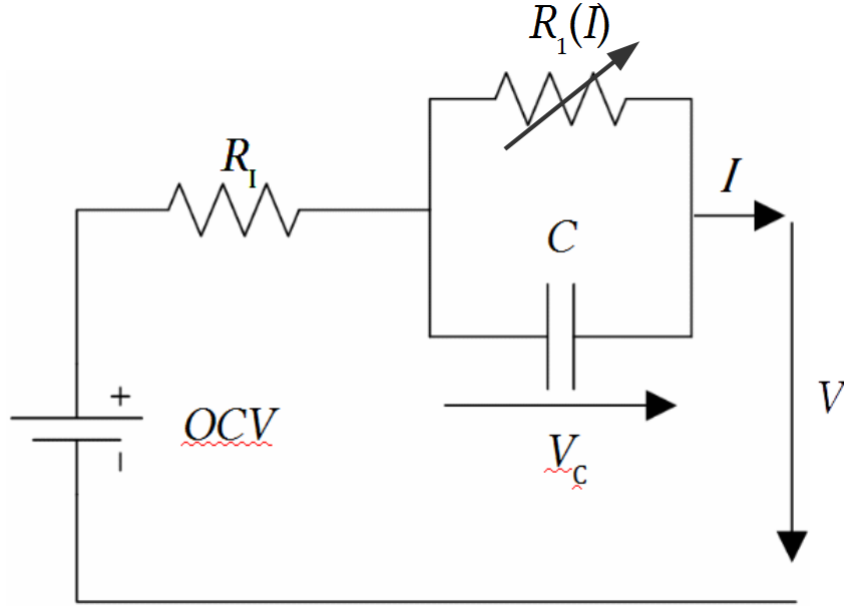
Equivalent Circuit Model 3: $OCV + R_0 + R_1(k_I)C$ 

Figure 6.4: Rate-dependent first-order ECM.

States:

$$x(k) = \begin{pmatrix} SOC(k) \\ V_C(k) \\ R_0(k) \\ R_1(k) \\ C(k) \\ k_I(k) \end{pmatrix}$$

State equation:

$$\begin{aligned} SOC(k+1) &= SOC(k) + \frac{\Delta t \cdot \eta}{C_N} \cdot I(k) \\ V_C(k+1) &= V_C(k) \cdot \left(1 - \frac{\Delta t}{R(I) \cdot C}\right) + \frac{\Delta t}{C} \cdot I(k) \\ R_0(k+1) &= R_0(k) \\ R_1(k+1) &= R_1(k) \\ C(k+1) &= C(k) \\ k_I(k+1) &= k_I(k) \end{aligned}$$

with:

$$R(I) = R_1 \cdot \frac{\ln \left(k_I \cdot I + \sqrt{(k_I \cdot I)^2 + 1} \right)}{k_I \cdot I}$$

Output equation:

$$V(k) = OCV(SOC(k)) + V_C(k) + R_0(k) \cdot I(k)$$

Matrices:

$$A_k = \frac{\partial f}{\partial x} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 - \frac{\Delta t}{R(I) \cdot C} & 0 & \frac{V_C \cdot \Delta t}{R(I)^2 \cdot C} \cdot g(I) & \frac{V_C \cdot \Delta t}{R(I) \cdot C^2} - \frac{\Delta t \cdot I}{C^2} & \frac{V_C \cdot \Delta t}{R(I)^2 \cdot C} \cdot h(I) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B_k = \frac{\partial f}{\partial u} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} \frac{\Delta t \cdot \eta}{C_N} \\ \frac{\Delta t}{C} + \frac{V_C \cdot \Delta t}{R(I)^2 \cdot C} \cdot \frac{k_I}{I} \cdot h(I) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

with:

$$g(I) = \frac{\ln \left(k_I \cdot I + \sqrt{(k_I \cdot I)^2 + 1} \right)}{k_I \cdot I}$$

$$h(I) = \frac{R_1}{k_I} \cdot \left(\frac{1}{\sqrt{(k_I \cdot I)^2 + 1}} - g(I) \right)$$

$$C_k = \frac{\partial g}{\partial x} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \left(\frac{\partial OCV}{\partial SOC} \quad 1 \quad I \quad 0 \quad 0 \quad 0 \right)$$

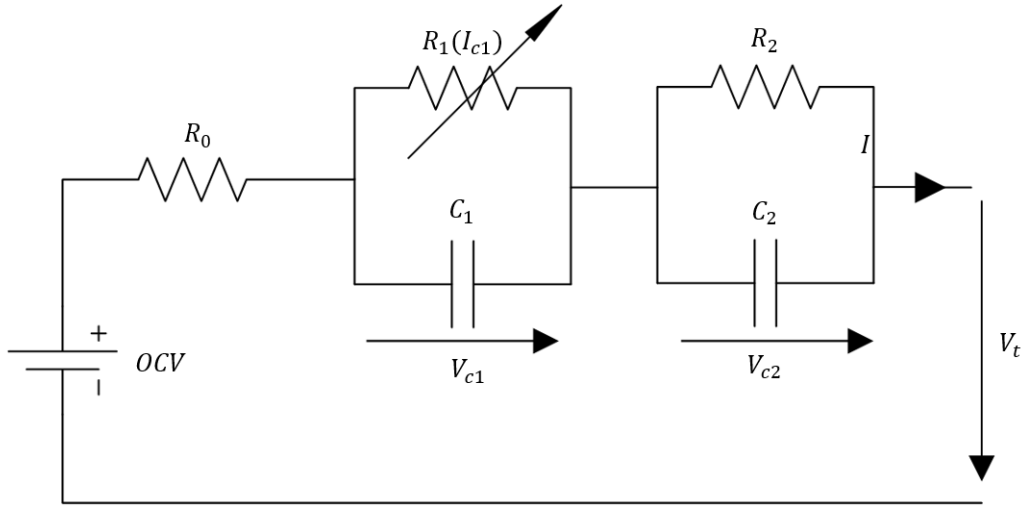
Equivalent Circuit Model 4: $OCV + R_0 + R_1(k_I)C_1 + R_2C_2$


Figure 6.5: Rate-dependent second-order ECM.

$$x(k) = \begin{pmatrix} SOC(k) \\ V_{C1}(k) \\ V_{C2}(k) \\ R_0(k) \\ R_1(I_{C1},k) \\ R_2(k) \\ C_1(k) \\ C_2(k) \\ k_I(k) \end{pmatrix}$$

State equation:

$$SOC(k+1) = SOC(k) + \frac{\Delta t \cdot \eta}{C_N} \cdot I(k)$$

$$V_{C1}(k+1) = V_{C1}(k) \cdot \left(1 - \frac{\Delta t}{R(I) \cdot C}\right) + \frac{\Delta t}{C} \cdot I(k)$$

$$V_{C2}(k+1) = V_{C1}(k) \cdot \left(1 - \frac{\Delta t}{R_2 \cdot C}\right) + \frac{\Delta t}{C} \cdot I(k)$$

with:

$$R(I) = R_1 \cdot \frac{\ln \left(k_I \cdot I + \sqrt{(k_I \cdot I)^2 + 1} \right)}{k_I \cdot I}$$

Output equation:

$$V(k) = OCV(SOC(k)) + V_{C1}(k) + V_{C2}(k) + R_0(k) \cdot I(k)$$

Matrices:

$$A_k = \frac{\partial f}{\partial x} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 - \frac{\Delta t}{R(I) \cdot C} & 0 & \frac{V_C \cdot \Delta t}{R(I)^2 \cdot C} \cdot g(I) & \frac{V_C \cdot \Delta t}{R(I) \cdot C^2} - \frac{\Delta t \cdot I}{C^2} & \frac{V_C \cdot \Delta t}{R(I)^2 \cdot C} \cdot h(I) \\ 0 & 1 - \frac{\Delta t}{R_2 \cdot C_2} & 0 & \frac{V_{C2} \cdot \Delta t}{R_2^2 \cdot C_2} & \frac{V_{C2} \cdot \Delta t}{R_2 C_2^2} - \frac{\Delta t \cdot I}{C_2^2} & \frac{V_{C2} \cdot \Delta t}{R_2^2 \cdot C_2} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B_k = \frac{\partial f}{\partial u} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \begin{pmatrix} \frac{\Delta t \cdot \eta}{C} + \frac{V_C \cdot \Delta t}{R(I)^2 \cdot C} \cdot \frac{k_I}{I} \cdot h(I) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

with:

$$g(I) = \frac{\ln \left(k_I \cdot I + \sqrt{(k_I \cdot I)^2 + 1} \right)}{k_I \cdot I}$$

$$h(I) = \frac{R_1}{k_I} \cdot \left(\frac{1}{\sqrt{(k_I \cdot I)^2 + 1}} - g(I) \right)$$

$$C_k = \frac{\partial g}{\partial x} \Big|_{x=\hat{x}^-} : \quad \hat{x}^- = \left(\frac{\partial OCV}{\partial SOC} \quad 1 \quad 1 \quad I \quad 0 \quad 0 \quad 0 \right)$$

6.2 Implementation of Non-Linear Filtering Concepts for BMS

Laboratory-Based Algorithm Verification

In recent years, many publications for BMS algorithm developed swamped the journals, based on KALMAN and particle filtering. Hence, in this section we are proposing to evaluate different methods to come to the conclusion, which of these stochastic inference methods based on Bayesian estimation show the best result. The literature shows controversial conclusion. Some would think, that using a battery system on the test bench, exciting it with some profiles to run the state and parameter algorithms for verification would be sufficient. Off-line, the results can be derived, rated and analyzed based on the filter quality.

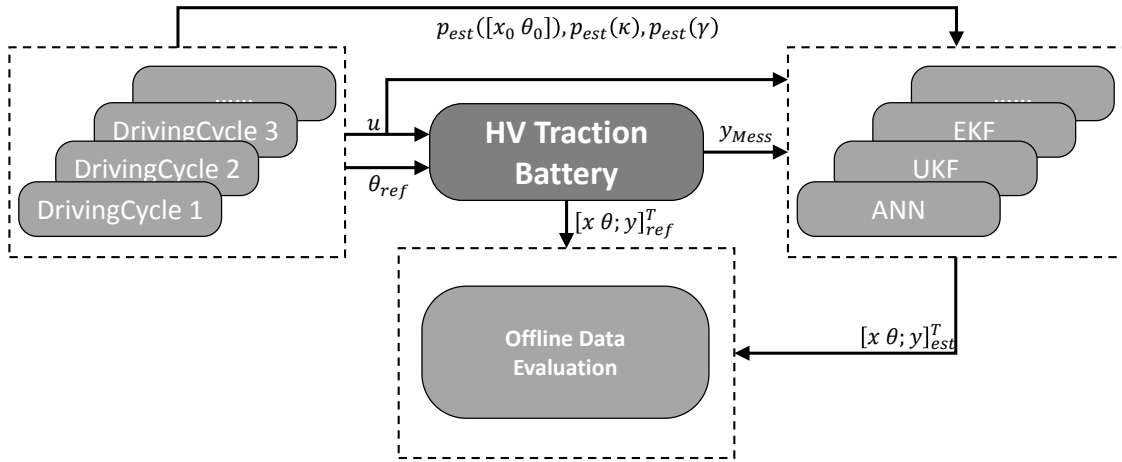


Figure 6.6: Validation of the algorithms on the actual HV traction battery system. (modified from [380])

However, this is not a practical solution, first of all, battery systems are very expensive, on material, design, safety and practically level. To be able to get an early prototype of a pack and its SW tested and verified is very unlikely, also, the safety of the operation has to be considered. It is also not very practical, it takes a long time to condition the battery in terms of temperature, SoC or even aging, also the same state cannot be reproduced in a real system. Therefore, in the first stage a pure off-line simulation framework was developed, to test various estimators.

Simulative-Based Algorithm Verification

When designing BMS and its algorithms, as key component of a battery pack, it is important to simulate the BMS and its SW functions in the most early stage, to guarantee that the system will meet all requirements. Even though battery packs may be integrated in a variety of applications, in this work area of interests for BMS verification/validation lies within the battery electric vehicle domain. In simulative-based algorithm verification procedure, the real battery pack is replaced by the simulation environment, consisting of an impedance-based derived model, while the filter model consisting of the derived non-linear, discrete stochastic system model (DSSM) to approximate the real battery in a constraint manner as good as possible (tracking all the specified complex, multi-time scale dynamics). Measurement data is collected with a electric vehicle which is equipped with measurement electronics. Additionally, test in the lab were conducted. Based on the reference model, it is possible to compare the "real" battery state with the estimation to conclude a quantitative assessment. Batteries have complex multi-time scale dynamics, hence in this work, the reference model was derived to approximate electric-, double-layer-, mass transport-, SOC-, temperature-, and aging effects to be able to account for all the testing requirements specified in technical safety and functional safety requirements. As discussed earlier, no model is capable to capture all effects of the real system, resulting in a divergence between filter model and real system. Hence, the tool includes different filter models to investigate their qualities for BMS application. The system description of the battery pack also includes all factors needed for dynamic simulation, drive-train characteristics, auxiliary components (e.g. charger). The vehicle task is characterized by the battery load profile, any kind of current input such as driving profile. As input, the user specifies the system input (current) and the initial probabilities, such as $p([x_0 \ \theta_0])$ as well for process and measurement noise $p(\kappa, \gamma)$. The reference model generates the "real" states and parameters. The reference values are generated such as measurement values for voltage, temperature and current, and used as input to the estimators, which can be selected out of the filter library. The filter library is builds up on [380] and extended with further filters. Based on that, and on the users choice of battery ECM, stored in the ECM library, the system state and parameter can be calculated [380].

$$x_k = \mathcal{F}_{\text{Ref,filter}}(x_{k-1}, u_{k-1}, \kappa_{k-1}) \quad (6.2)$$

$$y_k = \mathcal{H}_{\text{Ref,filter}}(x_k, u_k, \gamma_k) \quad (6.3)$$

This methods makes it possible, to reproduce the exact test and fault injection cases which leads to a quantitative assessment of the filter quality. In some cases, we use Monte-Carlo simulation, to fulfill a certain statistical coverage. Fig.6.7 shows the entire toolbox ReBAAI, consisting of stochastic inference methods for state- & parameter estimation as deep learning methods.

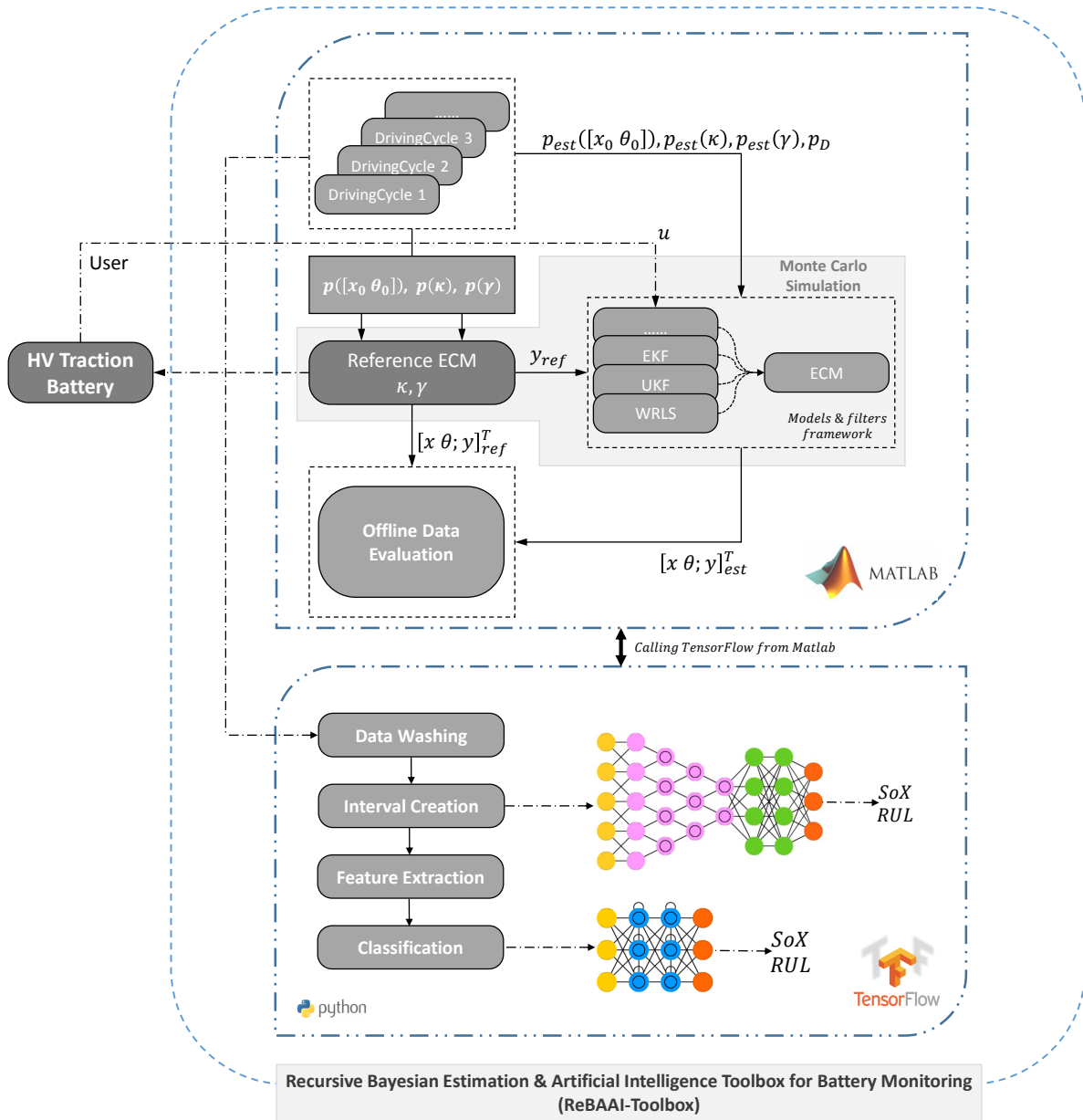


Figure 6.7: Validation of the algorithms based on simulative framework. (modified from [380])

6.3 Benchmarking Criteria

In this section, the benchmarking criteria, quantitative and qualitative methods are discussed to assess performance of different filter concepts with their filter models. This was conducted to provide scores regarding the performance validity of battery state and parameter inferences. The evaluation of the filter performances includes 10 filters, also based on machine learning.

Data set - experiment and measurement setup

The real drive test is conducted and implemented on a FEV eFIAT 500 battery electric vehicle as illustrated in Fig.6.8a(top). The measurement set up is illustrated in Fig.6.8a (bottom).

The test drives are anonymously and totally 130 groups of data were collected within more than 2 years of driving and at least 2 km route length of 51 different drivers. Overall 2547.7 km covered with an average speed of the test is around 23.37 km/h [445]. The highest ambient temperature of the test is 40°C and the lowest temperature of the test is 0°C. The different algorithms were spread over the two Microautoboxes 1401/1507 (MABX) from dSPACE (Rapid-Control-Prototyping).

As input to the MABX I & MABX II the digital signals from the propulsion CAN bus system are used containing all important information (current, system voltage, cell voltages, temperatures, speed,...). To localize the vehicles position a GPS receiver was installed, while other components are placed in the spare wheel well. To read/write the measurements results, FLEETlog from IPETRONIK are used with integrated 3G modem which allows to send the data to the back-end. A CAN-decoupler was used to prevent errors on the CAN-bus, it secures a pure "reading" of the signals and prevents "writing". The incline angle, as input to the energy manager is provided by the IMU sensor board including acceleration and yaw rate sensors [445]. These electronics are powered by a 5V-DC/DC converter from TRACO POWER. To not drain the 12V on-board power system while recording the charging event (HV system shut down to charge to 12V battery), an additional 12V battery was placed in the trunk, while during the driving mode, the components are powered/connected via KL 15, 30 and 31. On MABXII, all the state- and parameter estimation algorithms are running providing the necessary inputs to the vehicle energy manager (vehicle motion model, load model, battery pack electric and thermal model, route-based energy estimation, power prediction) implemented in MABXI [445]. During the collecting time, more than 80GB of data was collected. For simplicity of validation purposes, instead a Monte Carlo simulation was applied illustrated in Fig. 6.9a-6.9c. After the first initial parameters / states are set, followed by 100 Monte Carlo runs. In each Monte Carlo simulation, the estimated parameter values are selected within a pre-defined interval in the last 20 iterations. As it was rather difficult to process all the collected data off-line, the Monte-Carlo methods was used for validation purposes, which allows with different initializations to simulate battery aging behavior. The results in Fig. 6.9a-6.9c show the convergence

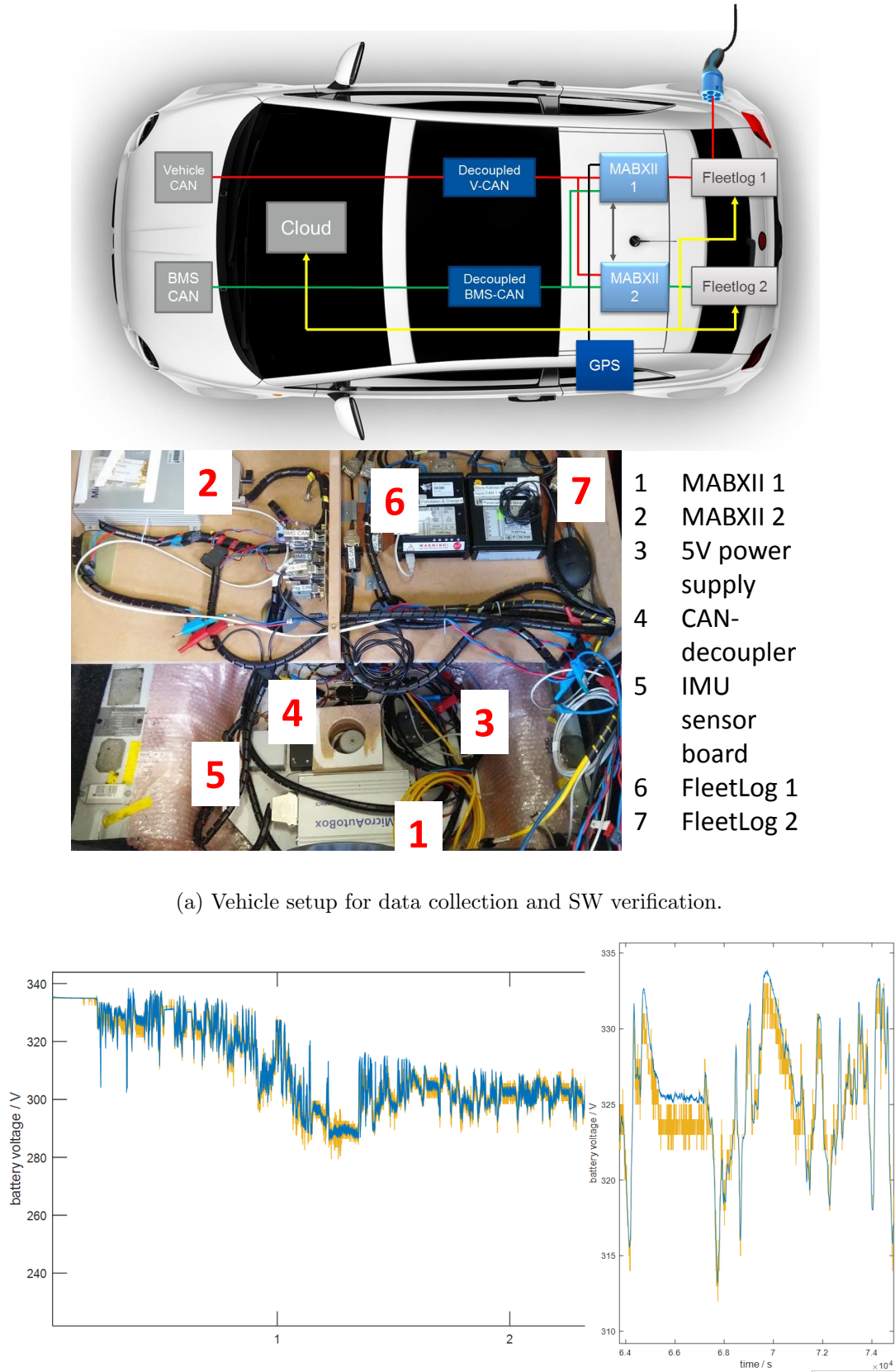
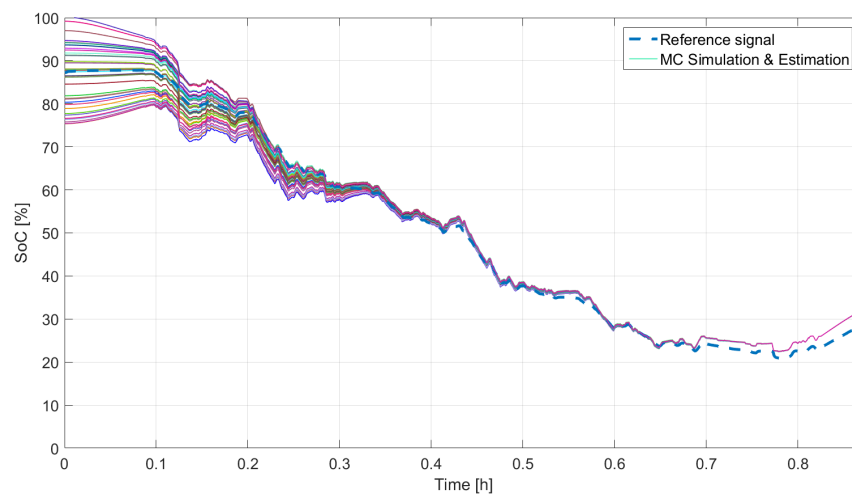
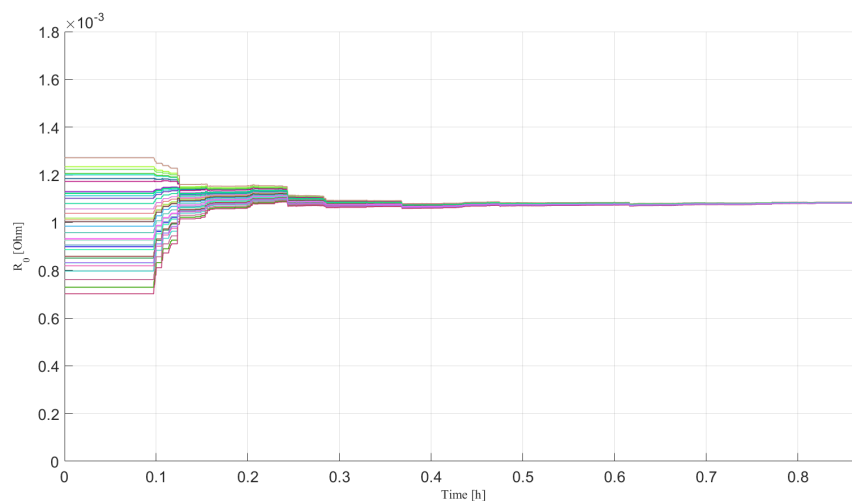


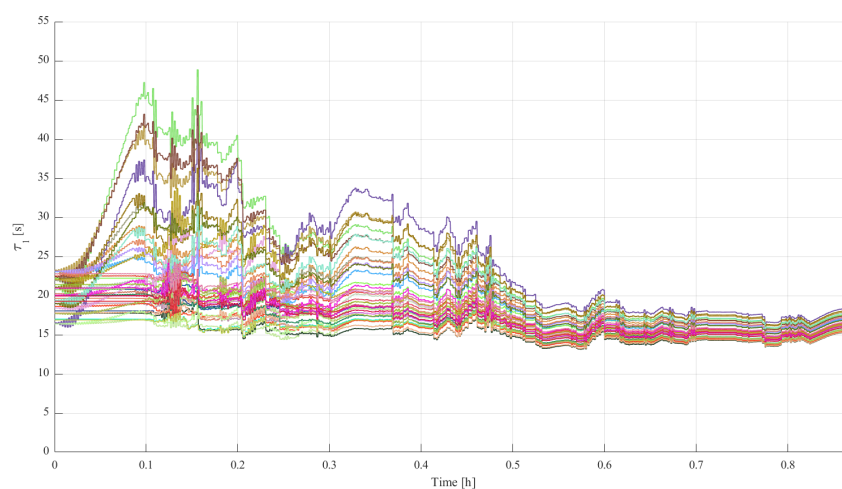
Figure 6.8: E-Fiat data collection and on-line estimation setup.



(a) Simulate aging by Monte-Carlo simulation - SoC.



(b) Simulate aging by Monte-Carlo simulation - internal resistance



(c) Simulate aging by Monte-Carlo simulation - time constant τ_1

Figure 6.9: Monte Carlo simulation to mimic aging behavior collected from eFiat data collection, here EKF dual estimation convergence properties of SoC , R_0 and τ_1

SWOT Analysis in regards to non-linear filtering requirements

This analysis shall give a personal judgement on pre-requisites for a successful preparation of the filtering frameworks including all necessary information, before the implementation can be started.

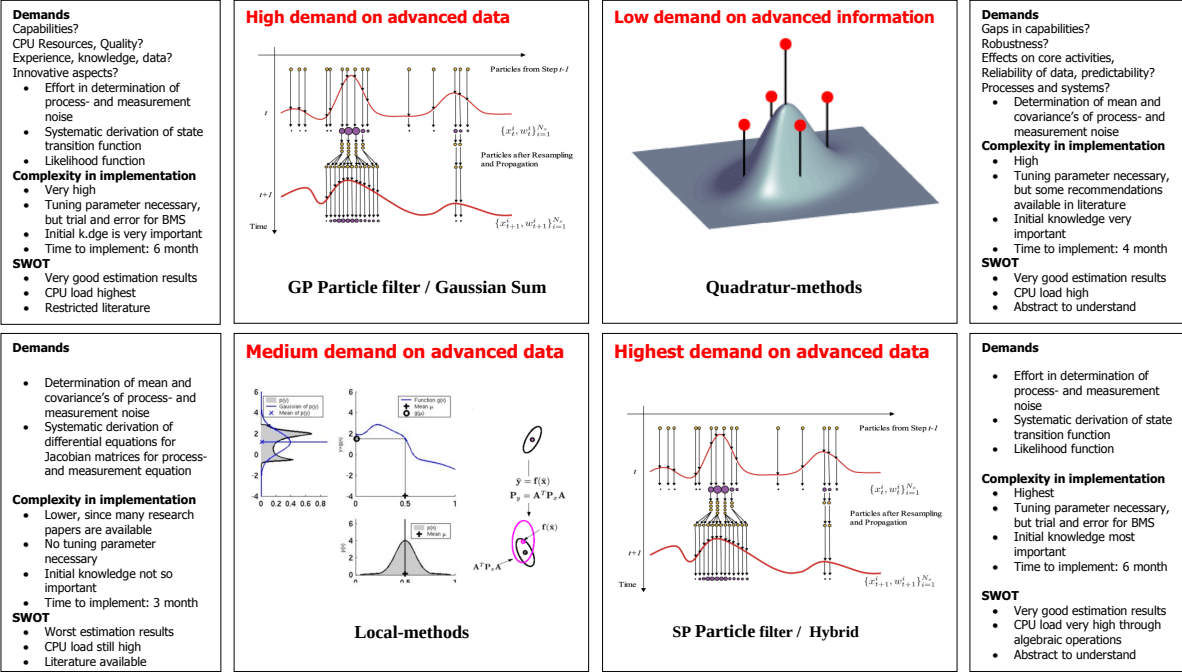


Figure 6.10: Expert judgment on necessary advanced data requirements and implementation challenges.

Preparation and implementation stringent requirements

A relatively complete library of non-linear filters were implemented, some of these filters are rather well known and easier to understand and implement, while other filters are relatively new and need lots of preparation before implementation. The Fig.6.10 shows subjective rating of advanced requirements the algorithm designer has to be aware of. The tunable parameters vary for different filters. While EKF is a non-tunable filter, but for hybrid Gaussian mixture filters, the components for GMM in terms of state, process- and measurement noise, particles and unscented transform parameters must be available. This is rather complex to tune and therefore makes the hybrid filters most difficult to implement. In regards to the implementation effort, this also varies, depending on the complexity of the algebraic operations. It is a relative rating based on own experience throughout the work, considering the availability on resources. What makes it so difficult is the fact, that some of these parameters are not relying on an entirely deterministic approach in which the rigorous implementation can be based on, rather recommendations from literature or even educated guess is the last step. Also, an important factor for implementation effort depends on the initial knowledge regarding these different families of filters. Someone starting at zero will struggle more than others with first experience, it is also in the nature of the concepts that they are different and each has to be understood

accordingly. Therefore, from the gained experience throughout the work, the subjective evaluation ranks as following, easiest to start with up to most difficult would be like:
Local methods (EKF) – > Quadrature methods (UKF) – > Particle filter (GPF) – > Gaussian Mixture Filter – > Particle/Quadrature hybrids.

6.4 Neural Networks and Deep Learning

Much has been written about AI's potential to reflect both the best and the worst of humanity. As leaders, it is incumbent on all of us to make sure we are building a world in which every individual has an opportunity to thrive. Understanding what AI can do and how it fits into your strategy is the beginning, not the end, of that process.

– Andrew Ng.

Many of the described techniques strongly depend on the reliability and high degree of accuracy of the dynamic equivalent circuit models. However, higher order models may be too complex as part of an overall BMS software in terms of real-time implementation on ordinary vehicle control units. A further challenge is to determine by on-line identification the model parameters. It may not always be possible to estimate changes in right direction by means of convergence; hence model robustness over time cannot be guaranteed. On the other side, a significant feature of solving problems in biological systems is the characteristic of adaptation towards unknown and unpredictable situations. E.g. the humans are able to execute special tasks because of the learning process a child goes through until adulthood. Artificial intelligence (AI) is the wide concept of electronic systems being able to automatically improve their performance without being explicitly programmed following the concept of the human nervous system as illustrated in Fig. 6.11. Machine learning is part of AI based around the concept that the systems are fed with (big) data from which they learn and adapt for itself. There are basically 3 types of machine learning problems [446]:

Supervised learning - Task driven here the system learns to predict a desired output (labeled responses) given an input object.

Unsupervised learning - Data driven is used to draw conclusions and find a good representation from data sets in the absence of labeled responses.

Semi-supervised learning contains both, labeled (minority of data) and unlabeled (majority of data) examples with the idea that the model will be more robust in comparison to the supervised learning method.

Reinforcement learning is close to how humans learn; in RL the methods used learn a policy on how the system shall act in an given environment. The systems action will

have consequences on the environment, as reaction, the environment will provide rewards that controls the learning.

Simplified speaking, AI algorithms consists of three steps:

- Input data sets
- Corresponding expected output
- evaluation whether the algorithm performs well

Hence, the challenge is to find a AI model that transforms the input data into meaningful outputs. Deep learning, or layered representation learning, is a sum of successive layers of representation. The depth of the model, meaning how many layers the network consists, may range up to hundreds/thousands of layers. On the other hand, other models focus on training only up to 1-5 layers, which is called shallow learning. To make a neural network complete the most common goals in machine learning, which are regression and classification, the right activation functions are quite important. Normally the output layer has only one unit and the activation function is linear for regression problem and logistic for classification problem [446].

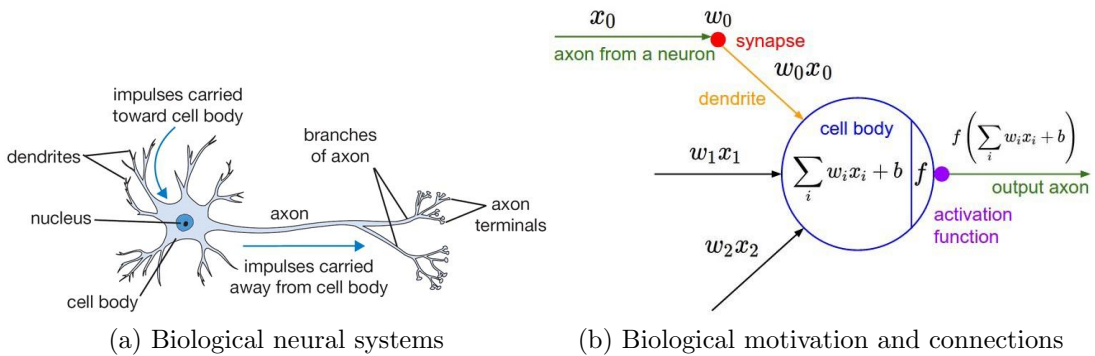


Figure 6.11: A cartoon drawing of a biological neuron (left) and its mathematical model (right). [447]

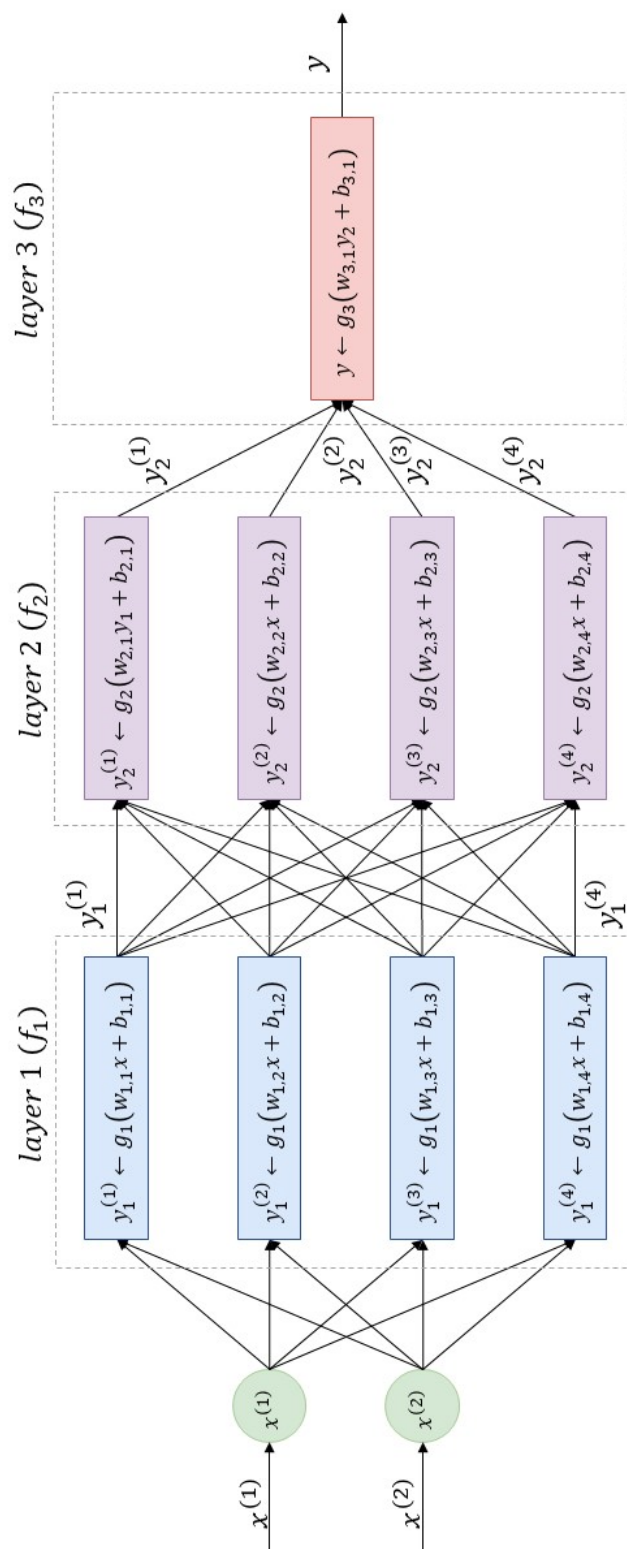


Figure 6.12: A feed-forward neural network with two hidden layers and modified from [446].

Activation

The activation function is unlimited in theory, as long as it is differentiable since the gradient descent method needs it to find all the parameters, i.e. weights and biases. However, to ensure the nonlinear performance, nonlinearity should be included in the activation functions. If all the functions are linear, then the neural network is not able to approximate any nonlinearity.

TanH (hyperbolic tangent function) and **ReLU** (rectified linear unit function) are two major choices besides logistic function [446].

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}},$$

$$\text{relu}(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise} \end{cases}$$

Matrix

In a regular neural, before execute activation, it calculates $\mathbf{W}_l \mathbf{z} + \mathbf{b}_l$ in layer l . The number of rows in $\mathbf{W}$ is the number of nodes in the current layer, while the number of columns is equal to the number of outputs from the previous layer. So, the final output of the layer l will be a vector whose dimension is the number of nodes [446].

6.4.1 Shallow learning

The methods on learning models of depth 1-3 layers of data representation are called shallow learning methods (like support vector machines (SVM), logistic regression, perceptron, restricted Boltzmann machines, de-noising autoencoder). These methods require extensive and difficult feature extraction/engineering (manually understanding the data and what is meaningful for the training) A new method using a well-studied technique from the field of soft-computing, which is called adaptive neuro-fuzzy inference system (ANFIS) is used for battery management systems. It combines the strength of Fuzzy Inference, namely the ability of human knowledge modeling, and Neural Networks, the capability of learning, into one framework to integrate a trainable system that adapts to the aging process of a battery pack.

Basics of Adaptive Neuro Fuzzy Inference Systems

Fuzzy inference systems and neural networks are well-known methods for modeling non-linear systems. In his dissertation, *Jyh-Shing Roger Jang* has proposed a method called

adaptive neuro fuzzy inference system (ANFIS), which combines both methods into one framework and thus possesses the advantages of both the learning capability and the representation of human knowledge [448]. Basically, ANFIS is a high effective tool to map input signals to output signals of a non-linear, uncertain or ill-defined system and, therefore, can be used as an universal approximator.

ANFIS Architecture

For *first-order Takagi-Sugeno* type of FIS, the output fuzzy sets are represented by linear functions of input variables. These function outputs are weighted using the *degrees of support* of every rule. The results are crisp output values. As a result, after aggregation by summing up all rule outputs, no defuzzification is needed, which is specific for FIS of *Takagi-Sugeno* type. The rule base of this ANFIS contains two if-then rules, whereas each leg leaving *layer 1* correspond to one rule:

- *Rule 1:* If $\mathbf{x}$ is $\mathbf{A}_1$ and $\mathbf{y}$ is $\mathbf{B}_1$, then $\mathbf{f}_1 = p_1x + q_1y + r_1$
- *Rule 2:* If $\mathbf{x}$ is $\mathbf{A}_2$ and $\mathbf{y}$ is $\mathbf{B}_2$, then $\mathbf{f}_2 = p_2x + q_2y + r_2$

ANFIS is a fuzzy inference system, but it is also a subclass of adaptive network systems. Consequently, the basic underlying structure is a neural network containing different kind of nodes and connection links. In case of ANFIS, the neural network is a five-layered feed-forward type (see Figure 6.13). The connection links within this network are non-weighted and the nodes can be either squared or circled depending on whether parameters pertain to them (squared nodes) or not (circled nodes). The parameters in *layer 1* define the input membership functions (here generalized bell functions) expressed in equation (6.4). They are referred to *premise parameters* as they are used in the *premise (antecedent)* part of each rule. The parameters in *layer 4* belong to the linear functions expressed in the consequent part of each rule and therefore they are called *consequent parameters*. In layer 1, each parameter set defines an input membership function expressed by the following equation:

$$\mu_{A_1}(x) = \frac{1}{1 + \left(\left(\frac{x - c_1}{a_1} \right)^2 \right)^{b_1}}. \quad (6.4)$$

x is the first input variable to the ANFIS. The membership function in equation (6.4) describes to which degree x belongs to fuzzy set A_1 . The parameter sets $\{a_i, b_i$ and $c_i\}$ determine the shapes of the membership functions. In layer 2, the output according to equation (6.5) is also called *firing strength* of the corresponding rule. Every T-norm operator performing an *AND*-operation can be used instead of multiplication. Equation (6.5) formulates the firing strengths for the ANFIS in Figure 6.13. Since generalized bell functions are non-zero at any time, it is guaranteed that every rule is fired.

$$w_i = \mu_{A_1}(x) \times \mu_{B_i}(y), \quad \text{for } i = 1, 2. \quad (6.5)$$

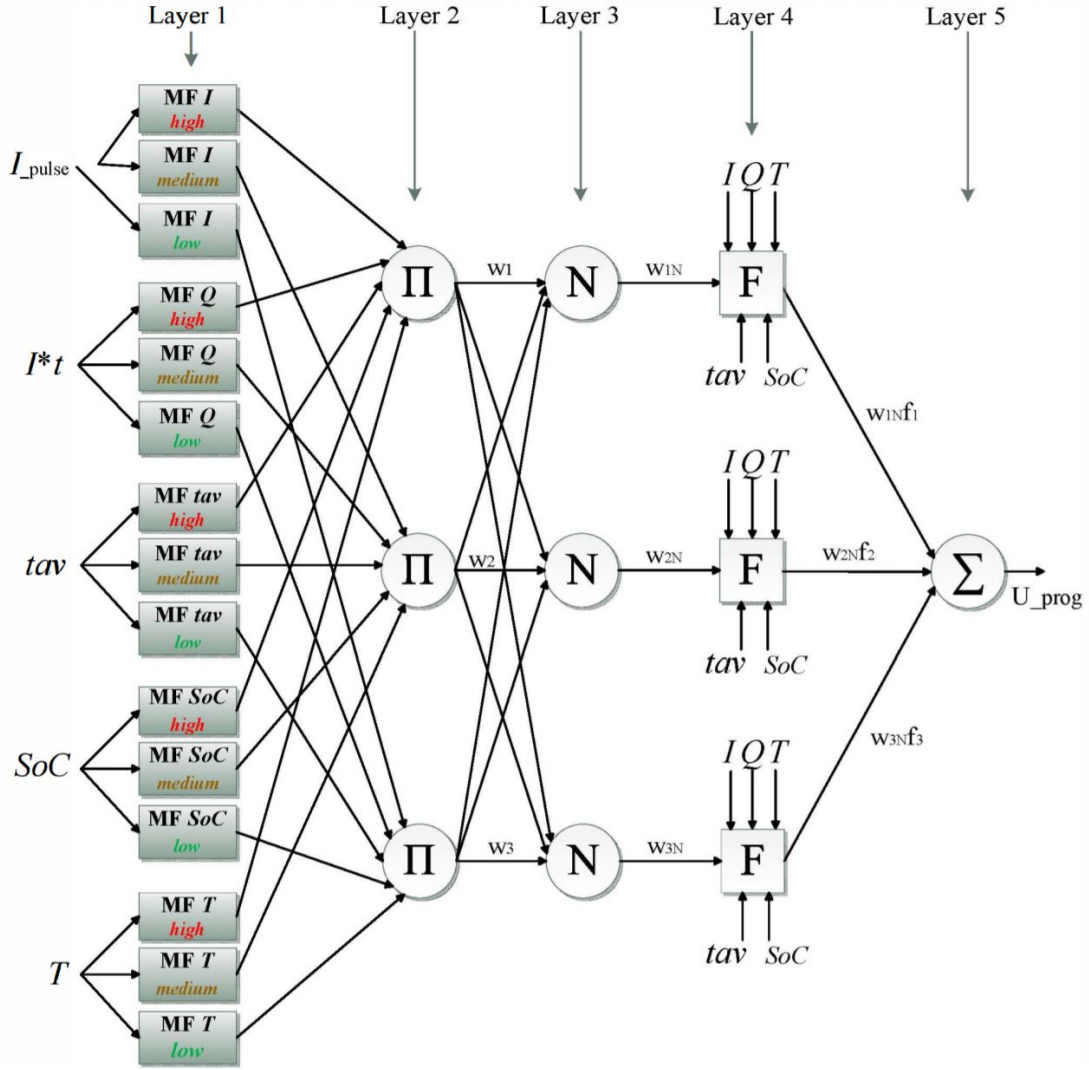


Figure 6.13: ANFIS structure.

In layer 3, *normalized firing strength* which is calculated in the respective nodes according to equation (6.6):

$$w_{iN} = \frac{w_i}{w_1 + w_2}, \quad \text{for } i = 1, 2. \quad (6.6)$$

Here, the degree of support of each rule is used to calculate the denominator of the normalization formula. In layer 4, The node function of node i is expressed by following equation (implication):

$$O_i^4 = w_{iN} f_i = w_{iN} (p_i x + q_i y + r_i). \quad (6.7)$$

O_i^4 is the output of rule i in layer 4. The parameter set $\{p_i, q_i, r_i\}$ describes the coefficients of a linear function of the input signals (here x and y). In layer 5 the *overall output* is

calculated by summing up every rule output from layer 4 (aggregation).

$$O_i^5 = \sum_i w_{iN} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \quad (6.8)$$

The basic ANFIS architecture for a *type-3* ANFIS introduced here serves as a basis for voltage prognosis.

6.4.2 Deep learning

On the other hand, deep learning simplifies the ML work-flow, through automatizing the extensive work of feature engineering as needed for shallow learning models into a simple end-to-end learning model [449][446]. Even though, the deep learning methods were understood early, e.g. long term memory concept in 1997, or convolutional neural network for computer vision even earlier, it took some time to become more applicable, that is due to the fact of fast improvements in hardware for these complex calculations, which are nowadays highly parallel (GPUs) (as deep neural network consist of many matrix multiplications). Big clusters with GPUs or even beyond, TPU (tensor processing unit) will increase the learning of deep neural networks. Secondly, only when appropriate data sets are available the findings are valuable. In terms of battery systems and management, with the start of monitoring the fleet and collect vehicle data, machine learning can contribute. Also, recent new training algorithms such as gradient propagation with its activation functions resulted in improvements. As in the first part discussed, SoH estimation is normally done by on-line parameter estimation techniques of reduced models or using probabilistic methods. As machine learning technology enables embedded systems to be self-learning, it was the motivation to understand the potential of these methods. Additionally, cloud computing becomes more and more important for OEMs and suppliers, as additional functions (e.g. early road side warnings) can be deployed and shared across OEMs. Already in China the regulation require a remote monitoring system (RMS) of electric vehicles sending data via the cellular network to the server where the data is analyzed, if the xEV is actually driven and used as intended, or the car was bought to avoid traffic ban for conventional vehicles. On the other hand, a huge amount of data will be collected, which enables the data driven approach, finding the overall fleet battery conditions based on the input from individual RMS clients.

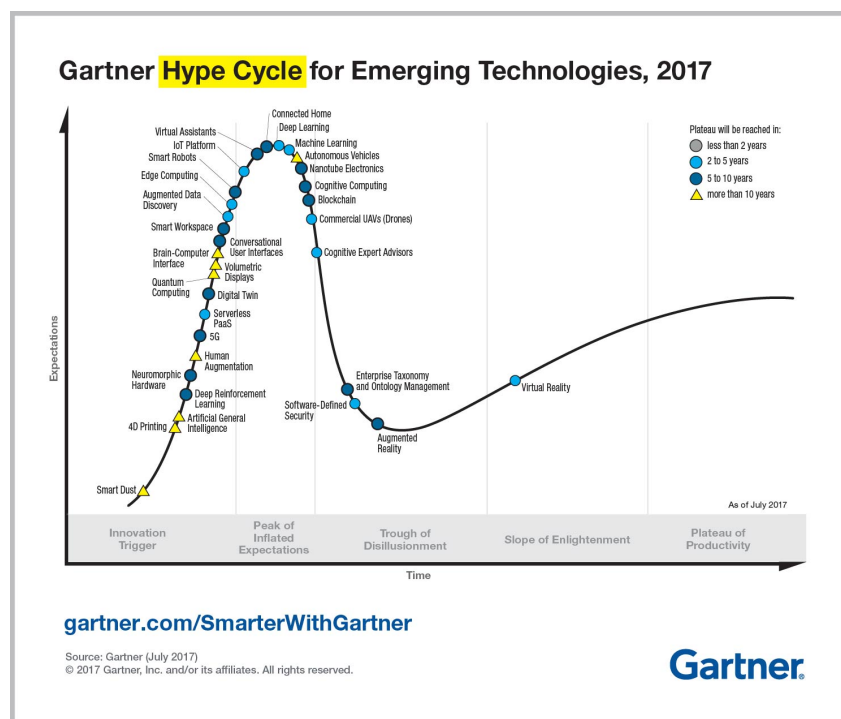


Figure 6.14: Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017 - Artificial Intelligence Everywhere

If the neural network has more than one hidden layer, then learning process can be referred as Deep Learning [446]. With the network becomes the deeper and deeper, training can be more difficult due to two fatal challenges, **exploding gradient** and **vanishing gradient**.

Some techniques as **gradient clipping** and **L1/L2 regularization** are developed to handle the gradient exploding problem.

As for the gradient vanishing problem, since backpropagation method uses the chain rule to update the parameters backwards, if the chain is too long the gradient can keep decreasing and eventually 'vanish'. Then, the parameters will stop updating and sometimes the neural network stops working. For instance, the TanH function has the gradient in the range from 0 to 1. These small gradients will be multiplied L times along the chain, where L is the number of layers. The gradient vanishes gradually because of the multiplications. Recently, many improvements, like ReLU in LSTM or skip connections in residual neural networks, are used to avoid the gradient vanishing problem.

With all the improvements above, nowadays, the term 'Deep Learning' means the combination of the neural network and those techniques.

6.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a kind of Neural Networks that have great performance in image recognition and classification. The basic principle are summarized in [450] is regional feature detection. By collecting subtle features like regional edges and shadows and aggregating them to form larger features like shapes and colors, CNNs can construct a feature pyramid while the top is the detected object's label or class.

The first CNN is LeNet proposed by Yann LeCun in 1990s. Recently CNNs have developed a lot but the main concepts are remaining. Amongst all the ideas, the core is the convolution.

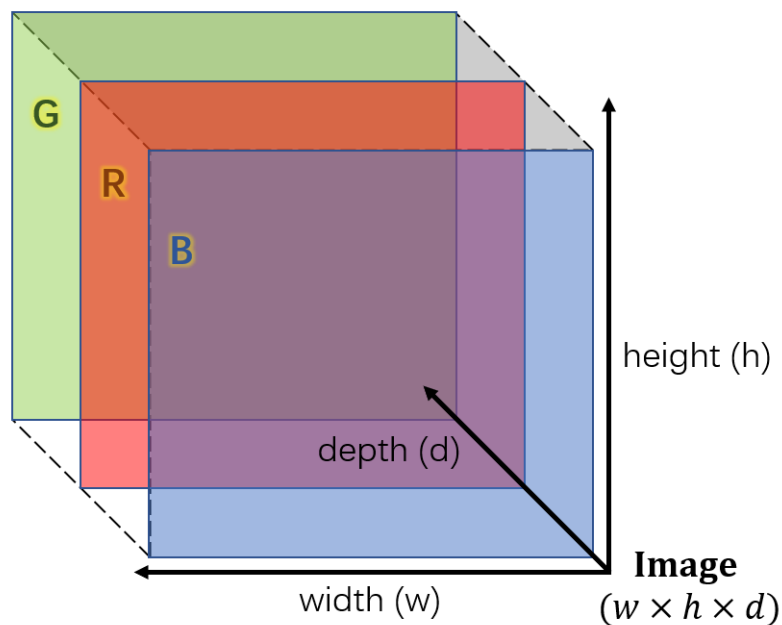


Figure 6.15: A classical RGB image concept [450].

6.5.1 Input Images

Before exploring more about CNNs, we need to look into the inputs first. The inputs are often images. Figure 6.15 shows a very common input image. However, the input doesn't have to be an exact figure. It can be in any similar form so the parameters, width (w), height (h) and depth (d), can always describe the input. Note that depth is also called channel sometimes.

6.5.2 Convolution

The "convolution" operation is executed by the convolutional layers, which are the key to extract features. A convolutional layer actually consists of several small three-dimensional filters which can also be described by width (w_f), height (h_f) and depth (d_f). This kind of filter slides over the whole input image and multiplies with the inputs elementwise and output the summation every time it slides. To show this in formula (6.9) [451],

$$o(i,j) = \sum_m \sum_n c(m,n)x(i-m,j-n) \quad (6.9)$$

where $o(i,j)$ is the output point at i -th row and j -th column, and c is the filter, x is the input. The values m and n mean every element in the filter.

An example is shown in figure 6.16. One filter can detect one feature, so more filters in the convolution layer mean more powerful ability to detect various kinds of features. The depth of output then depends on the number of filters in this layer.

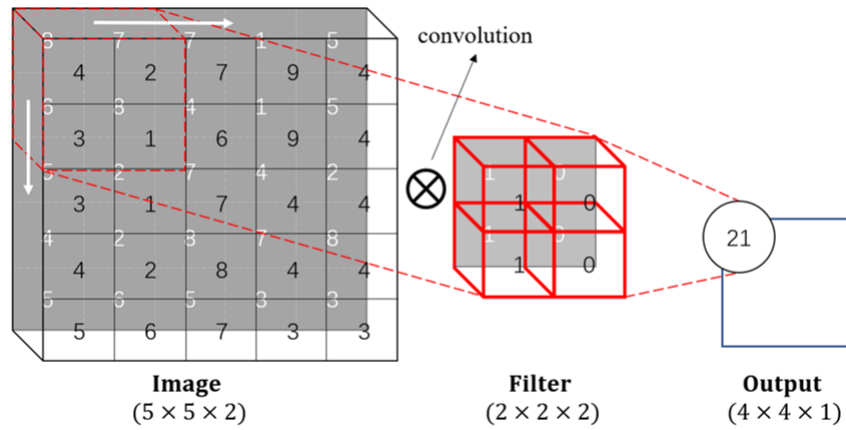


Figure 6.16: Convolution operation [450].

The output of every convolutional layer seems shrinking along width and height. In fact, there are two more common variables that control the output size besides the number of filters which controls the depth of output.

The first one is the **stride step** (s). The stride step is how far the filter strides. If $s = n$, the filter will move n points (pixels) to the right per stride until the end of current row, and then go back to the left but also go down n points then repeat.

The other one is **padding** (p). This is designed to save the information on the border. Padding means adding values along the border. The most popular one is zero-padding, which only adds zeros. For instance, in figure 6.16, adding one zero-padding means adding zeros at the periphery and the input image will be in size of $7 \times 7 \times 2$.

To calculate the size of the output, for a $(w \times h \times d)$ input and n ($w_f \times h_f \times d_f$) filters,

with stride step s and p padding, the output size is $(w_o \times h_o \times d_o)$ where [451]

$$w_o = 1 + \frac{w + 2p - w_f}{s} \quad (6.10)$$

$$h_o = 1 + \frac{h + 2p - h_f}{s} \quad (6.11)$$

$$d_o = n \quad (6.12)$$

6.5.3 Nonlinear Activation

After the convolution, we need to adjust the output and map it to more complicated relationship. Here comes the nonlinear activation function. As any ANNs, the nonlinearity is achieved by this function, while in CNNs the function is usually the Rectified Linear Unit (ReLU) [451],

$$ReLU(x) = \max(0, x) \quad (6.13)$$

The advantages of ReLU rather than other traditional activation functions as sigmoid and tanh are the avoidance of gradient vanishing and zero introducing.

6.5.4 Pooling

Although the using of convolutional layers reduces the number of parameters efficiently, a scan through the whole image still costs. Sometimes the features in a local region are similar and we only want the most representative one. Pooling is a down-sampling operation to achieve this idea. A pooling operation is using a window to scan the whole image like what a convolutional layer does, but the difference is that a pooling uses some strategy to just keep one value to represent this region. The most popular ones are max-pooling and mean-pooling. The max-pooling takes the maximal value inside its window, while the mean-pooling (or average-pooling) takes the mean value of all values in its window. There is normally no padding in the pooling but the stride also needs to be defined. Note that the pooling is not necessary in every layer.

6.5.5 Fully Connected Layer

Just like a vanilla forward neural network, to output a class or a value, the output of convolution is flattened and connected to the fully connected layer, in which every node connects to all the nodes in previous and next layers if exist [451].

6.5.6 Architecture

Overall, figure 6.17 shows the general architecture of a Convolutional Neural Network. A classic CNN stacks the convolutional layers and fully connected layers. The form of the output layer depends on the result we want.

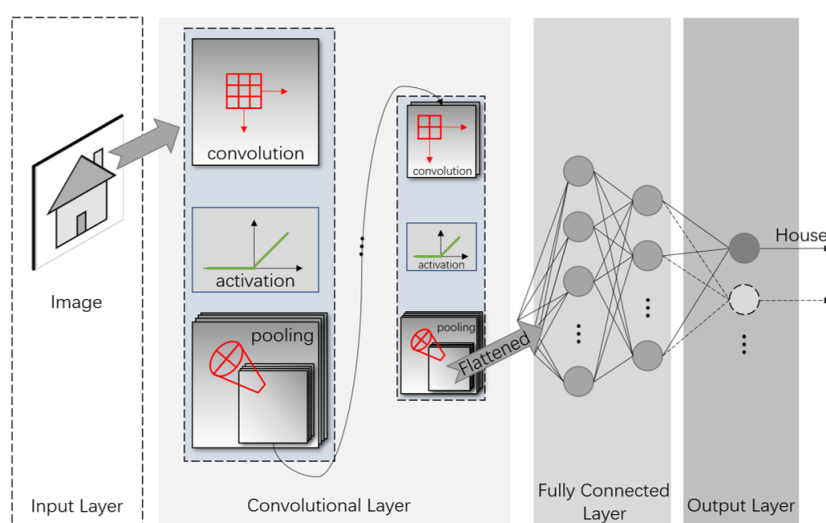
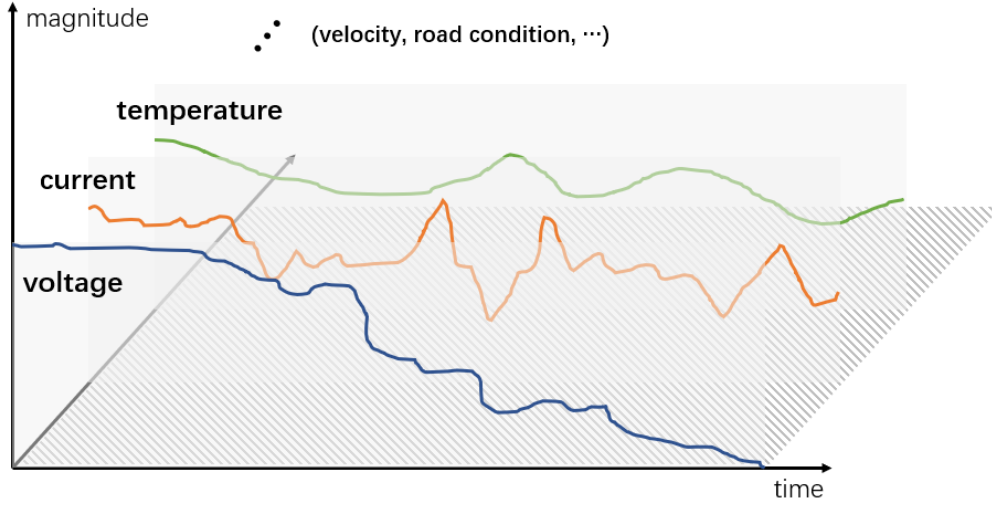


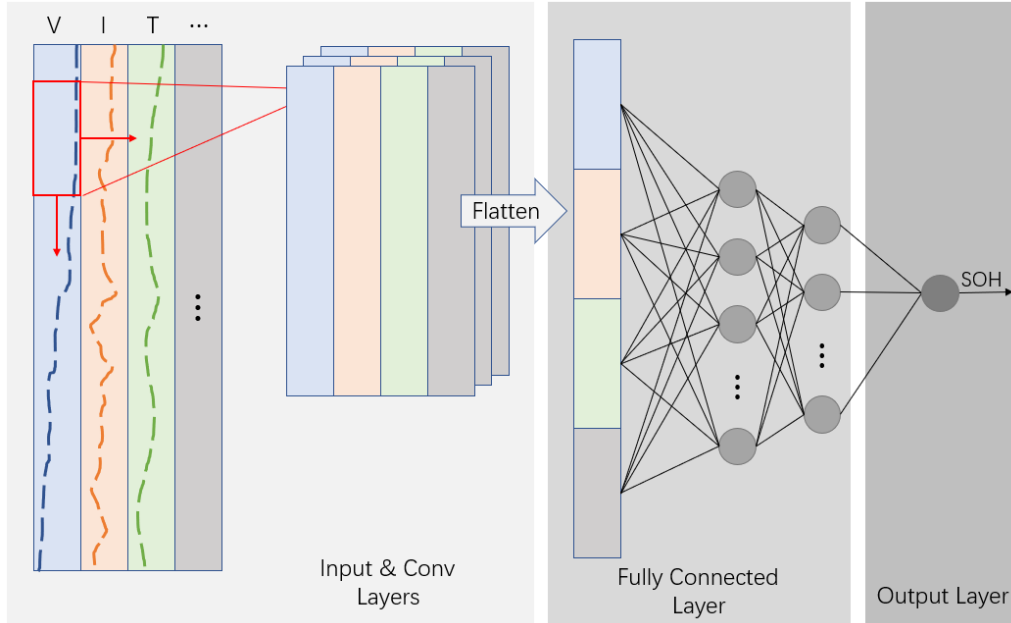
Figure 6.17: General CNN structure [450].

6.6 CNNs on Battery States Estimation

The lifetime of the battery is related to many factors while using, such as different charging and discharging behaviors, service environment, the influence of other components, etc. They are hard to be modeled and evaluated directly. However, the degradation of the battery is reflected on the signals of the battery. If the pattern of signals' changing over time can be discovered, we should be able to find the pattern of battery degradation.



(a) The way to generate a pseudo image consisting all the battery signals [450].



(b) The way to implement CNNs on the estimation of the battery's state of health [450].

Figure 6.18: CNN for State-of-Health determination

As mentioned above, CNNs can be used to detect spatial features inside an image and similar situations. The main idea is to capture the changes in successive values. As long as the battery signals are presented as an image, CNNs can be used to capture the hidden pattern.

As shown in Fig.6.18, the relevant signals including voltage, current, temperature, velocity and so on are concatenated together after resampling to the same length. To make sure they are of close magnitudes, the signals should be normalized first. Then, a pseudo

figure is obtained, whose size is determined by the resampling frequency and the number of features selected. With the pseudo figures as inputs and the state of health (SOH) as labels, the rest is nothing but a standard CNN working flow.

The amount of parameters of multi-layer perceptrons (MLP) grows very fast. More accurately, one additional layer means $(n_{l-1} + 1) \times n_l$ more parameters, where n_l is the number of nodes in the layer l .

Assuming images as the input, the high-dimensional input requires a complex architecture. Then a MLP can be too complex to run efficiently. The convolutional neural network (CNN) is designed to reduce the complexity of the original MLP.

In a picture, adjacent pixels usually represent a mutual object or clarify the edge of different objects. A square can be used to scan the image locally as a moving window. The window detects the area and edge of the target object and it needs to remember the pattern of certain object. Since one window can only detect one pattern, the number of windows depends on how many features needs to be found. To achieve this effect, the convolution between the window and the image part is calculated, which contains the relevant information. That is the basis of CNN.

The windows in a convolutional layer are called filters as they can filter the certain features. The filter will slide across the image, top to bottom and left to right. The stride step and the way of padding to the border need to be defined. The former property means how many pixels will be crossed in one slide. Padding is the method to deal with the border as to extend the border (normally with zero values) or not. What is more, generally all convolutional layers use ReLU as the activation function. Another term in CNN is **volume**. Because the existence of several filters leads to the same amount of output images, the input for the next convolutional layer (if there is) would be those images which are regarded as a collection called volume.

To speed up the training, a **pooling** layer following a convolutional layer is usually applied. Pooling is similar to convolutional layers but untrainable. It usually take the maximal or average value in its window.

Convolutional neural networks are developed for grid-like topology, such as time-series data taking measurements [452] at regular time intervals and employs discrete convolution instead of general matrix multiplication. The following CNN setup is as described in [453]. It convolves kernels over a two-dimensional measurement set for each layer to decrease the complexity in comparison to fully connected CNNs. The used data set for training and validating the network is represented by $D = \{(\Upsilon(1), SoH(1)), (\Upsilon(2), SoH(2)), (\Upsilon(3), SoH(3)), \dots\}$ where Υ represents the input data V, dV, I, T , and SoC , and SoH represents the ground-truth capacity value from the lab.

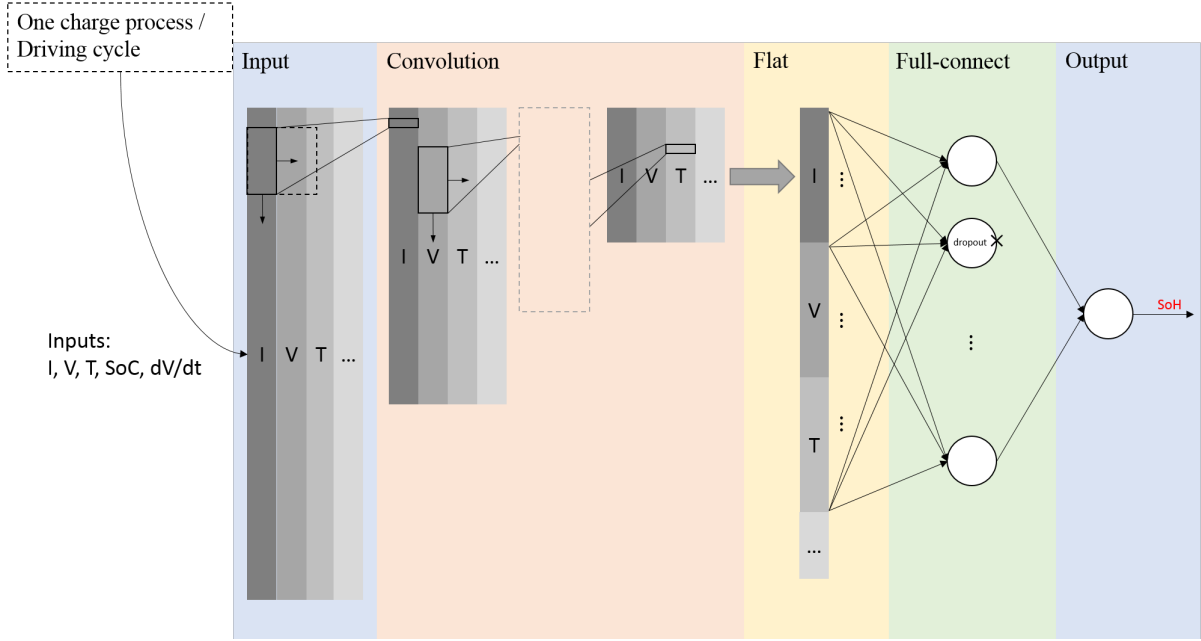


Figure 6.19: Architecture of a convolutional neural network (CNN) where each layer is designed of a convolution and pooling component followed by a fully connected layer.

Convolution

We assume measurement data to be defined by a function [454]

$$D : \{1, \dots, n_1\} \times \{1, \dots, n_2\} \rightarrow W \subseteq \mathbb{R}, (i, j) \mapsto D_{i,j} \quad (6.14)$$

such that the data D can be represented by an array $n_1 \times n_2$. Given the filter $K \in \mathbb{R}^{2h_1+1 \times 2h_2+1}$, the discrete convolution with filter K is given by [454]

$$(D * K)_{r,s} := \sum_{u=-h_1}^{h_1} \sum_{v=-h_2}^{h_2} K_{u,v} D_{r \mp u, s \mp v} \quad (6.15)$$

with (r, s) represents the position of the neuron, where K is represented as [454]

$$K = \begin{pmatrix} K_{-h_1, -h_2} & \cdots & K_{-h_1, h_2} \\ \vdots & K_{0,0} & \vdots \\ K_{h_1, -h_2} & \cdots & K_{h_1, h_2} \end{pmatrix} \quad (6.16)$$

with the Gaussian filter for smoothing

$$K_{G(\sigma)} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{r^2 + s^2}{2\sigma^2}\right) \quad (6.17)$$

with σ is defined as the standard deviation of the Gaussian distribution.

Bulding a deep convolutional network

The convolution layer l comprises $m_1^{(l-1)}$ features maps per previous input layer l , with the dimension of $m_2^{(l-1)} \times m_3^{(l-1)}$. The output of layer l comprises $m_1^{(l)}$ feature maps of the size $m_2^{(l)} \times m_3^{(l)}$. The i^{th} feature map in l is calculated as following [454]:

$$Y_i^l(\text{DVA}) = B_i^l + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)} \quad (6.18)$$

with B_i^l represents the bias matrix and $K_{i,j}^{(l)}$ the filter size. $2h_1^{(l)} + 1 \times 2h_2^{(l)} + 1$ connecting the j^{th} feature map in $(l-1)$ with the i^{th} feature map in l , with the output feature map $m_2^{(l)} = m_2^{(l-1)} - 2h_1^{(l)}$ and $m_3^{(l)} = m_3^{(l-1)} - 2h_2^{(l)}$. To apply this to a multi-layer perceptron at position (r,s) , the formula is further rewritten as following [454]:

$$\begin{aligned} \left(Y_i^{(l)}\right)_{r,s}(\text{DVA}) &= \left(B_i^{(l)}\right)_{r,s} + \sum_{j=1}^{m_1^{(l-1)}} \left(K_{i,j}^{(l)} * Y_j^{(l-1)}\right)_{r,s} \\ &= \left(B_i^{(l)}\right)_{r,s} + \sum_{j=1}^{m_1^{(l-1)}} \sum_{u=-h_1^{(l)}}^{h_1^{(l)}} \sum_{v=-h_2^{(l)}}^{h_2^{(l)}} \left(K_{i,j}^{(l)}\right)_{u,v} \left(Y_j^{(l-1)}\right)_{r+u,s+v} \end{aligned} \quad (6.19)$$

in which the filters and the bias matrices contain the network adaptive weights and the output feature maps [454]:

$$\begin{aligned} m_2^{(l)} &= \frac{m_2^{(l-1)} - 2h_1^{(l)}}{s_1^{(l)} + 1} \\ m_3^{(l)} &= \frac{m_3^{(l-1)} - 2h_2^{(l)}}{s_2^{(l)} + 1} \end{aligned} \quad (6.20)$$

Non-Linearity Layer

As the convolution layer is a linear operator, a non-linear layer after the convolution layer (ReLU, hyperbolic tangent function, Sigmoid) is added. The units at the position calculates the output as following [454]

$$Y_i^{(l)}(\text{DVA}) = g_i f \left(Y_i^{(l-1)}(\text{DVA}) \right) \quad (6.21)$$

where f is the activation function and g represents an additional gain coefficient.

$$Y_i^{(l)}(\text{DVA}) = \left| Y_i^{(l)}(\text{DVA}) \right| \quad (6.22)$$

Normalization Layer

The task of the normalization layer is to execute local competitiveness between adjacent units within a feature map and unit at same position but different feature map. The subtractive normalization operation computes [454]

$$Y_i^{(l)} = Y_i^{(l-1)} - \sum_{j=1}^{m_1^{(l-1)}} K_{G(\sigma)} * Y_j^{(l-1)} \quad (6.23)$$

with the Gaussian filter $K_{G(\sigma)}$. The local normalization in combination with the ReLu is given as [454]:

$$\left(Y_i^{(l)} \right)_{r,s} = \frac{\left(Y_i^{(l-1)} \right)_{r,s}}{\left(\kappa + \mu \sum_{j=1}^{m_1^{(l-1)}} \left(Y_j^{(l-1)} \right)_{r,s}^2 \right)^\mu} \quad (6.24)$$

where κ, μ are hyperparameters.

Pooling and Subsampling Layer

A pooling layer is placed in between the successive convolution layers to reduce [447] the amount of parameters and computational complexity and reduce the chance of over-fitting by downsampling the volume spatially. In the end we gain robustness to the precise location of the features. Here, the max pooling type is used, it divides the input signal into rectangular regions and computes the maximum of each.

Fully Connected Layer

The layer expects feature maps as input (e.g. $m_2^{(i-1)} \times m_3^{(l-1)}$), then the computation of the i^{th} unit in l is [454]

$$y_i^{(l)} = f(z_i^{(l)}) \text{ with} \quad (6.25)$$

$$z_i^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} \sum_{r=1}^{m_2^{(l-1)}} \sum_{s=1}^{m_3^{(l-1)}} \omega_{i,j,r,s}^{(l)} \left(Y_j^{(l-1)} \right)_{r,s}$$

where $\omega_{i,j,r,s}^{(l)}$ represents the weight connecting the unit at (r,s) in the j^{th} feature map of layer $(l-1)$ and the i^{th} unit in layer l [453]. Combining modules into a hierarchy, the overall CNN can be described that the convolutional layers learn about the features, while the fully connected layers are used to execute the classification based on computed features.

Results Convolutional Neural Networks

There are a lot of other extensions such as **attention**, **bidirection**, **seq2seq**, etc. for the recurrent neural networks [446]. As a reference case to evaluate the capability of deep learning based BMS, as such as being self-learning and adaptive to cell characteristic changes, one method is chosen from the SoH methods described earlier in the literature review. Exemplary of the long list of SoH algorithms, the methods based on Incremental Capacity Analysis (ICA) and Differential Voltage Analysis (DVA) or voltage plateau change estimation for e.g. during charging event shall be applied [455]. The ICA/DVA are quantitative approaches to investigate inner chemical reaction and capacity fade of lithium-ion batteries. The evolution of the IC curves is tracked to make a statement about the changes in the electrochemical characteristics. The plateaus, voltage gradients of the $Q - V$ relation are connected to the dQ/dV peaks in respect to the cell capacity.

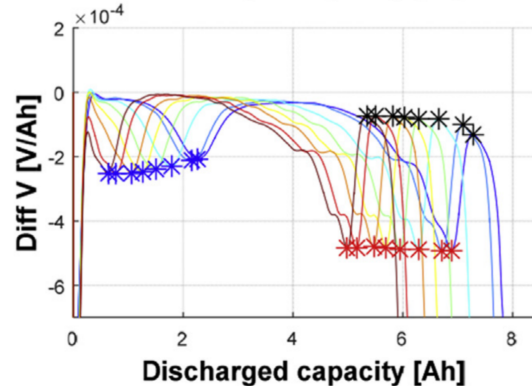
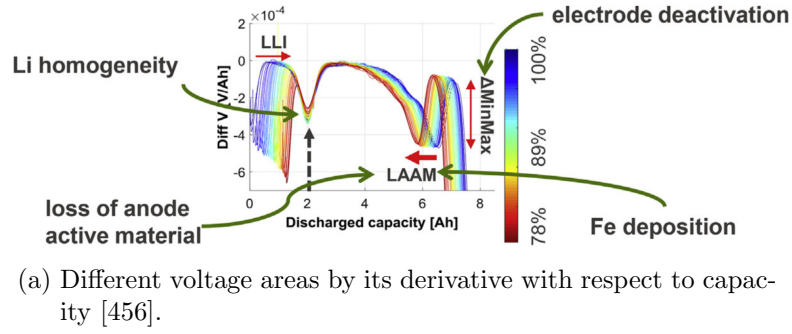


Figure 6.20: DVA as a tool for analyzing aging. [456]

This work does not emphasis on the deep learning methods and better deep neural networks can be found, rather in this work a general trend and opportunities shall be discussed towards possibilities for deep learning not only on vehicle level, but as a combination of individual monitoring as an input for large fleet level battery monitoring and if in the future these methods may replace classical methods. The long term idea is, to build a battery digital twin (global model), in terms of digital model of the operations of the electrical vehicle fleet, and continuously update the performance model with information and data from the real life usage. This extensively trained global model, divided into operation areas such as cold to hot zones, can then be deployed to update the models on the individual vehicles. The aim is to get a better understanding from the batteries in the field, as the customer usage and durability for new cells in cars are based on, bluntly said suppliers "gut feeling". For OEMs, its kind of a black box situation as battery data can only be gathered if customers have to come to read out logging data (which does not contain time series). With the help of the global model and big data analytics using deep learning methods, informed decisions are possible (fault detection and prediction, when to come to service, problems of certain components, deduce battery sizes...).

To evaluate the performance of the CNN, the ground-truth data is compared with the estimated SoH value resulting in the mean squared error. As an initial startup, the network weights were initialized by a random number generator with normal distribution. The CNN is capable of detecting the smallest changes and their correlations in the current

input profile in the DVA plateaus or charging curves as illustrated in 6.21.

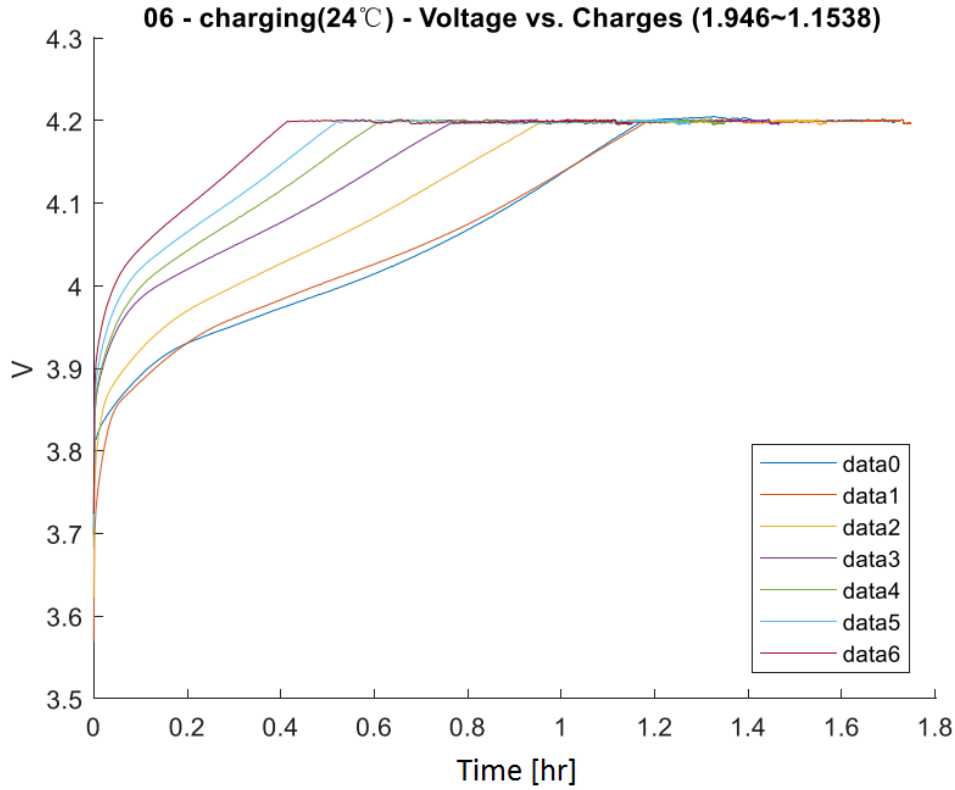
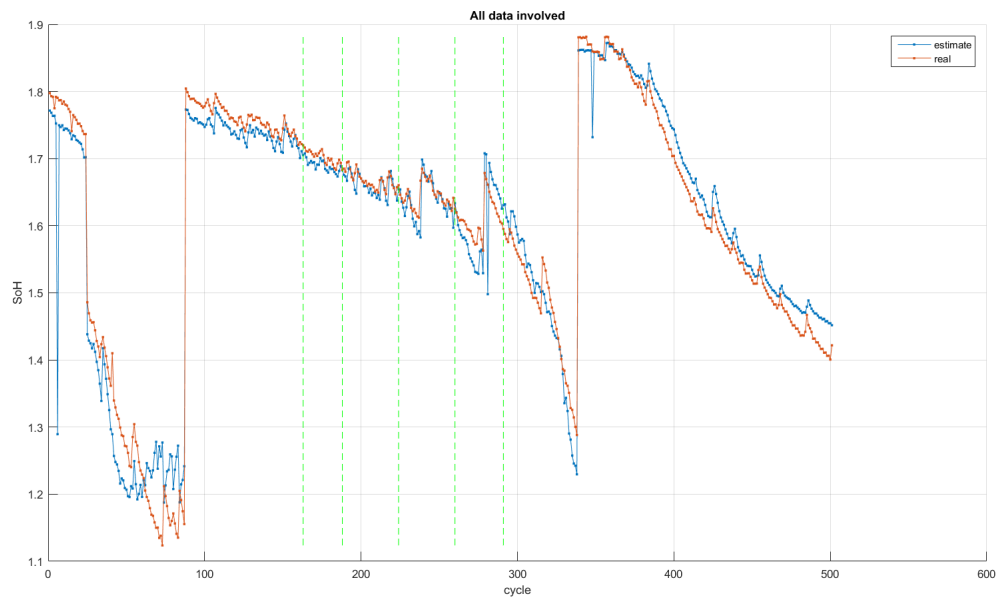


Figure 6.21: Figure 1. There is a fact that batteries with different SoH have different performance while charging [457].

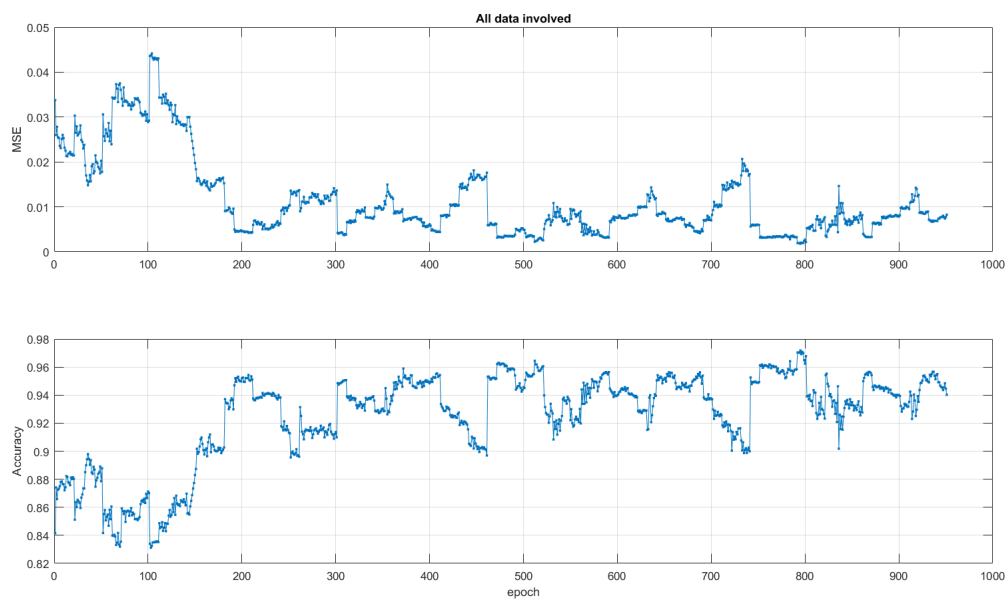
The network is designed in TensorFlow out of Matlab. The raw data contains of 34 datasets/batteries. Before the data can be fed to the CNN, it must be preprocessed, as the recorded data was very noise, jittered, or had missing data in need of data parsing and data purity. After pre-processing, only 28 sets were classified as valid. The dataset was divided into training set and test set, which contain 21 batteries for training, and 7 for validating accordingly for 4, 25, and 40°C. Any lost or nonphysical values were substituted by the mean of previous and next true values. The data set is available at NASA [457]. Each input data set is mapped towards SoH_C values, resulting in an average accuracy of 97,84%. The raw data includes 34 batteries tested under different conditions such as ambient temperature and discharging load. Each battery contains three processes which are charging, impedance and discharging. However, the charging processes are the same, which is constant current 1.5A until the voltage is 4.2V. The data needs to be trimmed firstly. Only charging processes are remained. In addition, some data have large error or don't have a reasonable behavior, e.g. the SoH is low at the beginning but so much higher then, so these data will be discarded. The trimmed data has three basic features: current (I), voltage (V) and temperature (T). To make sure all the data have the same length (for feeding CNN).

Estimation vs. Full Lab Data

A reasonable approach for DCA/ICA analysis is to estimate on-line the SoH is based on a Convolutional Neural Network (CNN), considering the direct measurements, V, dV, I, T , and takes into consideration user behavior, such as partial charging habits. We have chosen a CNN because it has been proven to be superior under the framework of ImageNet competition results. As can be seen in Figure 6.22, the SoH ground truth data oscillates rather strong, making it more difficult for the CNN to follow. As seen in the NASA data, the quality of the 28 tests have a range from good to poor quality. Therefore, processing techniques are used to identify and replace incomplete, inaccurate, irrelevant "dirty" data by aggregation, filtering, and standardizing.



(a) Full data set - CNN estimation vs. lab ground truth



(b) DVA curves for for aging [456].

Figure 6.22: SoH estimation based on CNN - (a) using all available test data from 28 cells with (b) best accuracy of 97.4% [450]

6.6.0.1 Recurrent Neural Network

Our signals are sequences and time-dependent, therefore, the investigation of RNN as SoH estimation on this data is applicable.

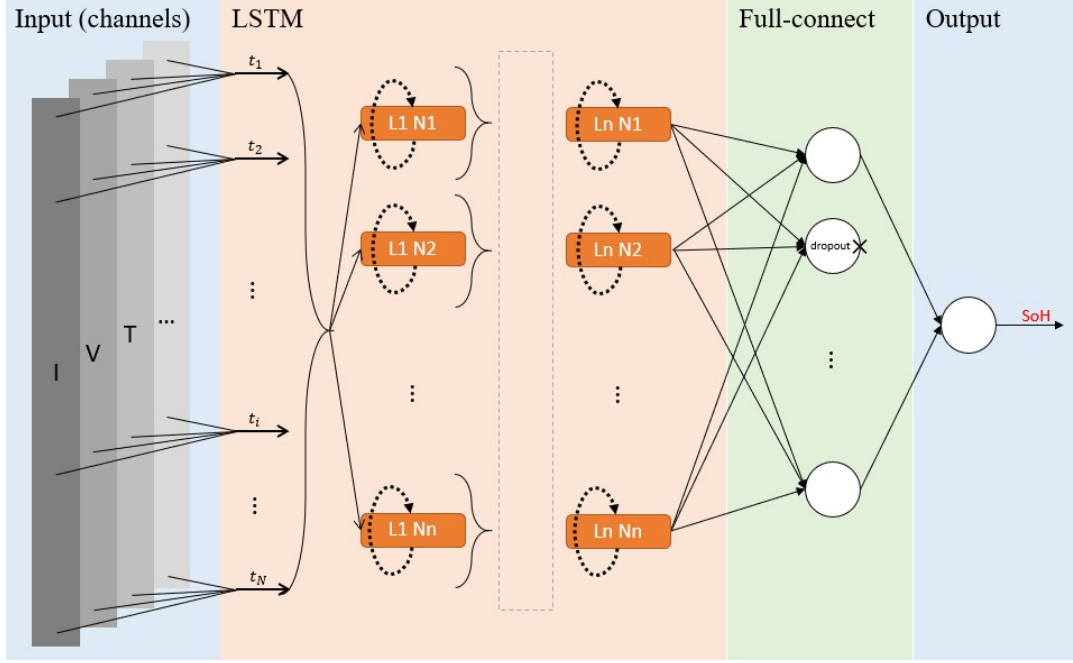


Figure 6.24: Structure of recurrent neural network used for SoH estimation [modified by [450]]

Since sequences such as texts, speeches and signals have a time line which contains some relationship amongst elements in the sequence, it can be assumed that one element can be determined by elements at other positions. **Recurrent neural networks** are invented to handle the sequences. Unlike the feed-forward neural networks, a RNN updates its inner **states** recurrently, which can be regarded as memorizing. In every loop, the units in a RNN will have two inputs, the input from previous layer and the state of itself from previous time instant. The outputs also consist of two parts, the output to the next layer and the updated state to itself for the next timestep. This process can be shown as [446]

$$h_t = f_h(W_h x_t + U_h h_{t-1} + b_h) \quad (6.26)$$

$$y_t = f_y(W_y h_t + b_y) \quad (6.27)$$

where h is the state and x, y are the input from the last layer and the output to the next layer. The output of each unit will have the same dimension as its input. The activation function f_h is applied to the linear combination of the input and state, and the TanH function is often the choice. For a classification problem, the output activation function

is often the softmax function:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^D e^{z_j}}, i = 1, 2, \dots, D \quad (6.28)$$

where D is the number of classes. The dimension of W_y depends on the dimensions of the state and output.

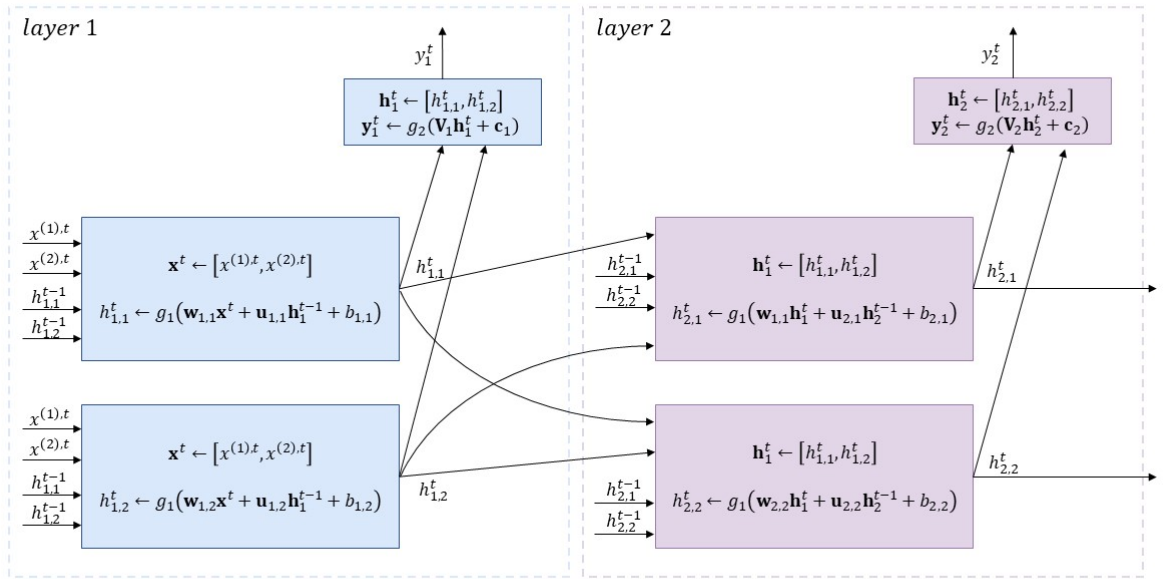


Figure 6.25: Variables flow in a RNN [446].

Because a RNN is trained along time, the back-propagation is also through time. It brings a obvious problem. For a long sequence, the gradient will be calculated through the folded network and tend to vanish (TanH and softmax functions suffer the vanishing gradient problem). Another major problem of RNN is the long-term dependency. For example, in a text fragment, a word may be related to the previous or next word, while it can also be related to a word in a different sentence.

The most practical RNNs are those with gates that can control to remember or forget the information at a certain timestep. **Long short-term memory** (LSTM) networks and networks with the **gated recurrent unit** (GRU) are known as gated RNNs. They use the gate state which is a value in the range of (0,1) to multiply with the information. As an example, a fully GRU has two gates, the update and reset gates while the minimal

GRU only has a forget gate:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (6.29)$$

$$h_t = f_t \circ h_{t-1} + (1 - f_t) \circ \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \quad (6.30)$$

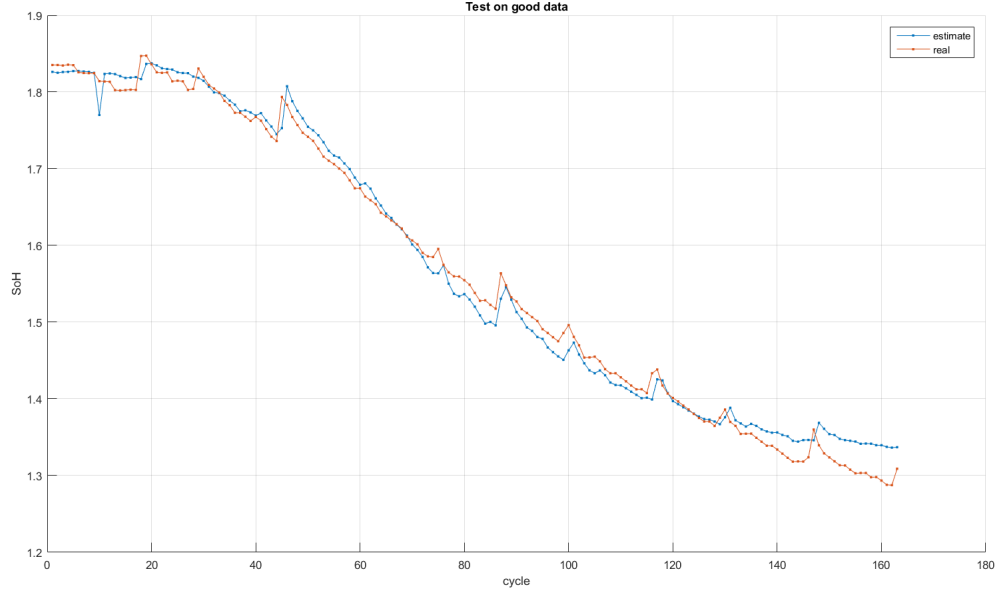
where $\circ$ is the elementwise product. In the formula, f_t is the forget gate. The activation function for the forget gate is sigmoid function. If the forget gate is close to one, the memory tends to keep its value from the last timestep, otherwise it will be substituted. The reason why the gate technique can solve the vanishing gradient is that sometimes it will keep the input, which means the derivative is constant, i.e. not vanishing.

Results Recurrent Neural Network

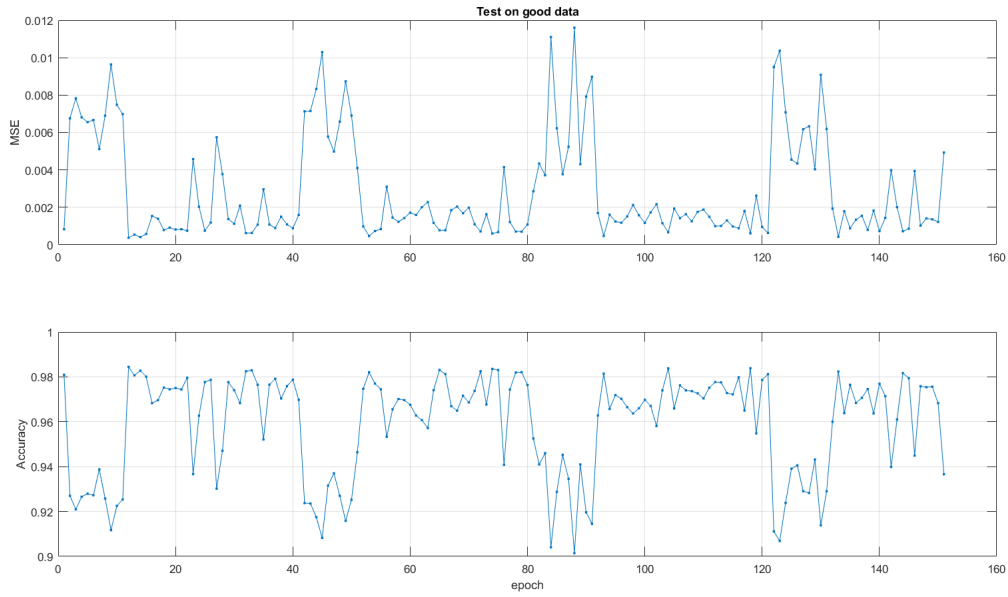
A standard LSTM neural network with one input layer, several LSTM layers (“Lx Ny” means Node y in Layer x), one full-connect layer and one output layer is applied in this work. Except that the last LSTM layer outputs only the value of the last time instant, other LSTM layers output the complete sequences. To evaluate the RNN feasibility, a basic test was conducted. Using RNN with three LSTM layers (128, 128, 64 nodes), two full-connect layers (64, 16 nodes), 0.001 learning rate in RMSprop Optimizer, recursively trained by four batteries’ data (05, 06, 07, 18), is leading to the following result.

For the whole dataset, which contains 28 batteries, the result is obtained as following. As the results for the "dirty" data is not very good, the following training will be conducted on 4 NASA batteries (#5, #6, #7, #18).

The problem for all batteries is the data of some of them is not very good, which means being highly corrupted or having strange changes in capacity change within one cycle. The RNN has more difficulties to follow this changes in comparison to the CNN.

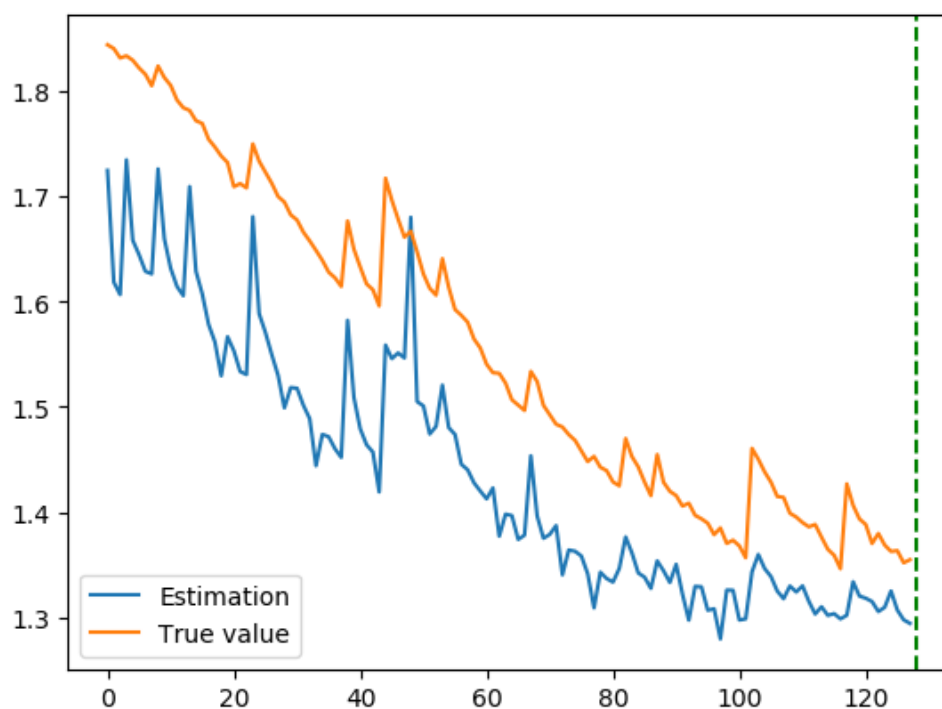


(a) Test on washed data / SoH estimation

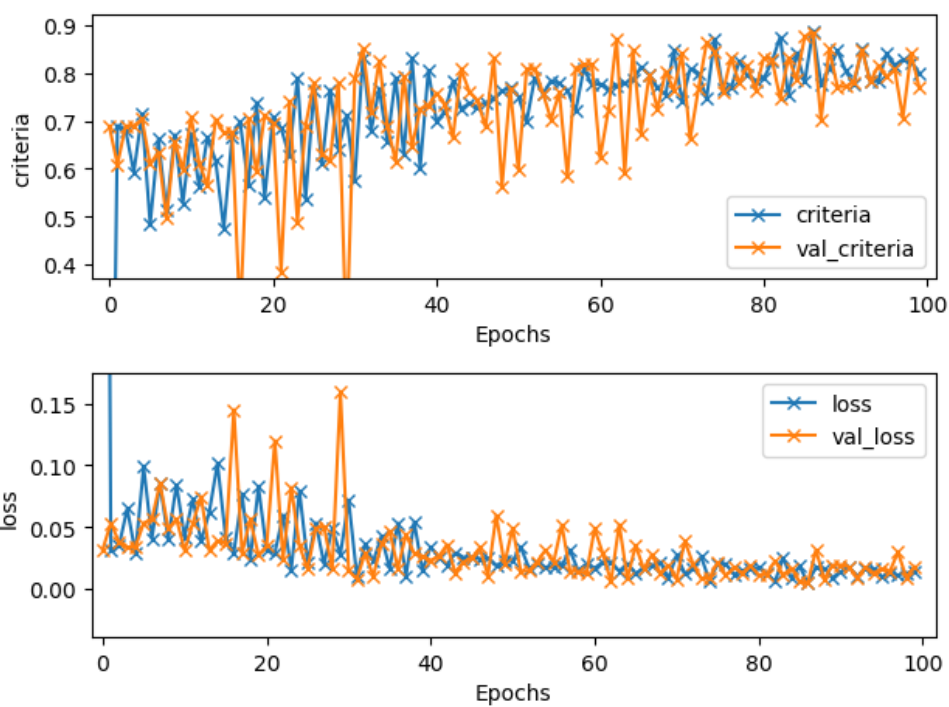


(b) Test on washed data / MSE and accuracy.

Figure 6.23: SoH estimation based on CNN - (a) using washed data from 28 cells (excluding corrupted data) with (b) better overall accuracy (max. 98.6%) [450]

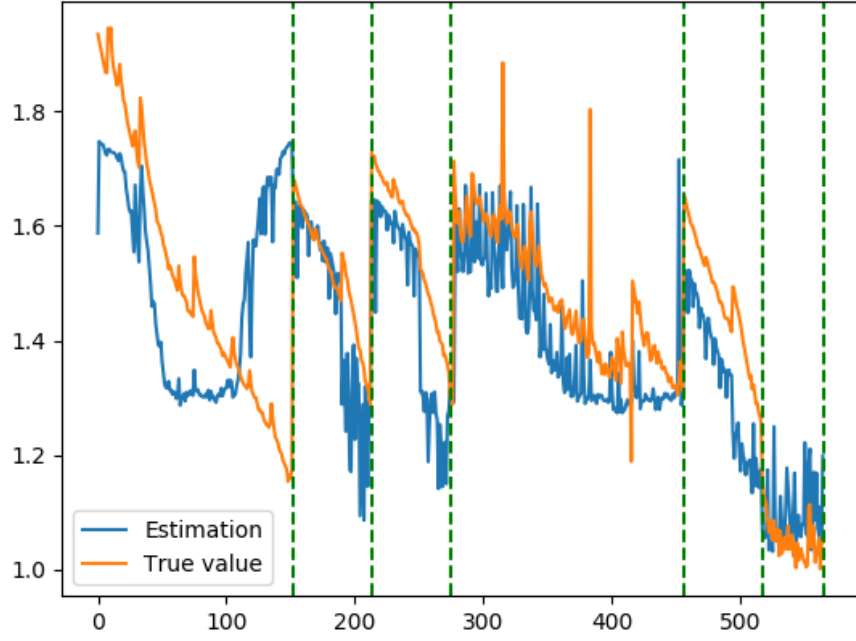


(a) Test on washed data / SoH estimation

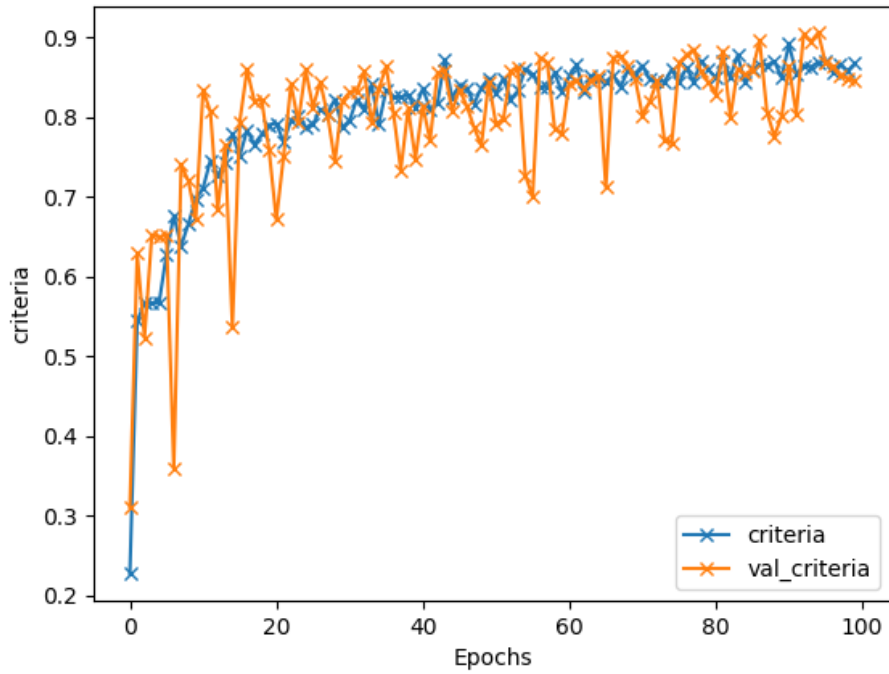


(b) est on washed data / MSE and accuracy

Figure 6.26: SoH estimation based on RNN - (a) estimation vs. lab data, (b) showing the loss in MSE [450]

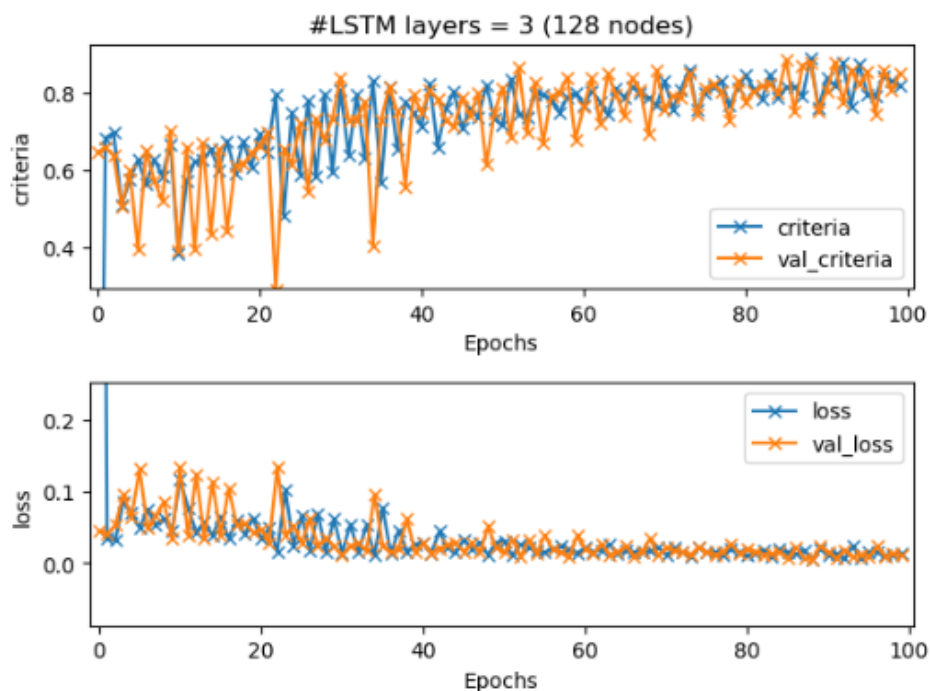


(a) Test on washed data / SoH estimation

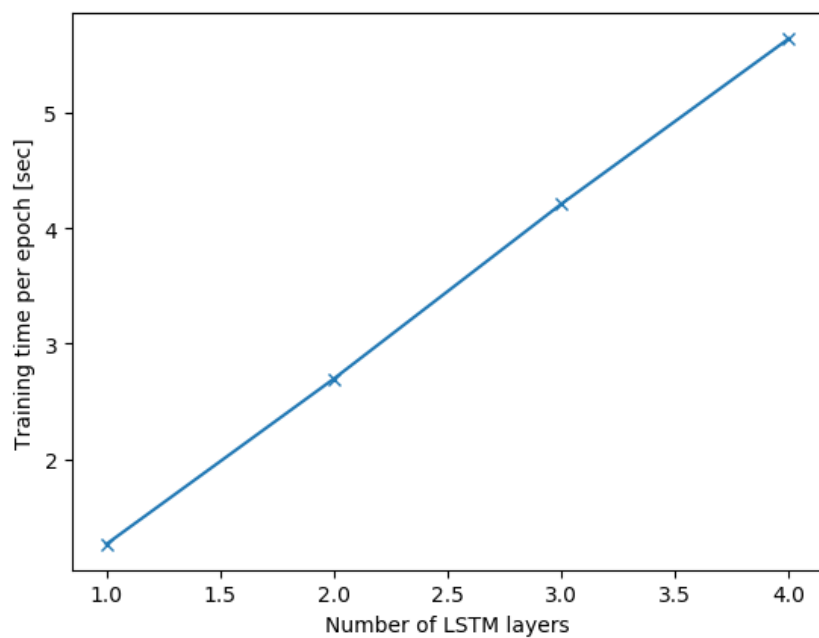


(b) Test on washed data showing the accuracy spread over the epochs

Figure 6.27: SoH estimation based on RNN - (a) including all the trimmed data, (b) the best accuracy results in 85% [450]



(a) a



(b) b

Figure 6.28: Variation of LSTM layers with limited improvements in MSE (trained on GPU Geforce 940MX)[450]

6.7 Numerical Benchmarking of Estimators

To give a conclusive evaluation of the different filter families, and which factors are important for the BMS application, following filter specifications are followed [380]:

1. **Robustness and quality** The task is to estimate the hidden states of the battery pack in an exact way while being always robust (numerical stable and robust, also the estimation error within threshold, divergence to meaningful value) in a timely manner and therefore has the highest priority. One quality measurement is based on the root-mean-squared error method which is a measure for the assessment of the estimation quality. It indicates in our case how much the estimation deviates on average over all Monte-Carlo simulations from the actual observation values. The RMSE gives indication about relative quality, which algorithms shows lowest estimation error, but not if the estimation is actually satisfactory. Therefore, an indicator is needed, which calculates a theoretical lower bound for the RMSE for any kind of unbiased estimator (UE) (for maximal second order). Since the posterior density function $p(x_k, Y_k)$ is in practice not following Gaussian shape, some statistical errors are involved. Nevertheless, the PCRB is widely considered for quality evaluation in non-linear estimation problems.

The Posterior Cramer-Rao Bound (PCRB) sets a lower bound on the variance of any unbiased estimators. Assume the PDF $p(\mathbf{x}; \theta)$ satisfies the regularity condition with expectation respect to $p(x[0]; \theta)$, then

$$\left[\frac{\partial \ln p(x[0]; \theta)}{\partial \theta} \right] = 0 \quad \forall \theta \quad (6.31)$$

As consequence, the variance of any unbiased estimator $\hat{\theta}$ must satisfy

$$\text{Var}(\hat{\theta}) \geq \left[\left(\frac{\partial \ln p(x[0]; \theta)}{\partial \theta} \right) \right]^{-1} \quad (6.32)$$

Additionally, an unbiased estimator that fulfills the CRLB for all θ for some functions g and $\mathcal{I}$ can be found, iff:

$$\frac{\partial \ln p(\mathbf{x}; \theta)}{\partial \theta} = \mathcal{I}(\theta)(g(\mathbf{x}) - \theta) \quad (6.33)$$

That estimator is the minimum variance unbiased (MVU) estimator $\hat{\theta} = g(\mathbf{x})$ and its minimum variance is $[\mathcal{I}(\theta)]^{-1}$, where $\mathcal{I}$ is referred as the Fisher information. It means the more information is available $\Rightarrow$ the lower the bound $\Rightarrow$ the lesser the variance. Besides the possibility of finding the MVU estimator using CRLB, the CRLB is applied to benchmark the performance of any unbiased estimator, the closer the estimator converges towards CRLB, the

better. Also, signals that change rapidly as θ changes result in more accurate estimators. Moreover, it is important for systematic evaluation of a set of different estimators to find and justify the estimators existence or to rule-out unfeasible estimators, that it is physically not applicable to find an unbiased estimator that beats CRLB. The unbiased estimator, which fulfills the CRLB is efficient in terms of MVU and it efficiently uses the available data. CRLB depends in general on the parameter θ and the data length. We further need to extend this theorem to a vector parameter, where the goal is to estimate a vector parameter, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_p]^T$. It is assumed that the assumptions for the scalar parameter are met. Then the variance of any unbiased estimator $\hat{\boldsymbol{\theta}}$ satisfies $\mathcal{C}_{\hat{\boldsymbol{\theta}}} - \mathcal{I}^{-1}(\boldsymbol{\theta}) \geq 0$ and $\mathcal{C}_{\hat{\boldsymbol{\theta}}}$ is positive semidefinite. $\mathcal{I}$ represents the Fisher information matrix, the unbiased estimator $\hat{\boldsymbol{\theta}} = \mathbf{g}(\mathbf{x})$ exists iff:

$$\mathcal{I}_{ij}(\boldsymbol{\theta}) = -\left(\frac{\partial^2 \ln p(\mathbf{x}; \boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\right) \implies \frac{\partial \ln p(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathcal{I}(\boldsymbol{\theta})(\mathbf{g}(\mathbf{x}) - \boldsymbol{\theta}) \quad (6.34)$$

In general, $\mathcal{I}$ is not diagonal and therefore, the CRLB will depend on other unknown parameters as the posterior $p(x_k|Y_k)$, normally not normal distributed, may lose information of higher moments. Another way to measure the filter quality is as following:

$$\sqrt{\frac{1}{MC} \sum_{i=1}^{MC} \left(\hat{x}_k^{(i)} - x_k^{(i)}\right)^2} \geq \sqrt{\text{diag}(\text{PCRB}_k)} \quad (6.35)$$

while PCRB represents the inverse of the filter information matrix J_k such that

$$J_k = D_{k-1}^{22} - D_{k-1}^{21} (J_{k-1} + D_{k-1}^{11})^{-1} D_{k-1}^{12} \quad (6.36)$$

where

$$D_{k-1}^{11} = \mathcal{E} \left[-\Delta_{x_{k-1}}^{x_{k-1}} \log p(x_k | x_{k-1}) \right] \quad (6.37)$$

$$D_{k-1}^{12} = \mathcal{E} \left[-\Delta_{x_{k-1}}^{x_k} \log p(x_k | x_{k-1}) \right] \quad (6.38)$$

$$D_{k-1}^{21} = \mathcal{E} \left[-\Delta_{x_k}^{x_{k-1}} \log p(x_k | x_{k-1}) \right] = [D_{k-1}^{12}]^T \quad (6.39)$$

$$D_{k-1}^{22} = \mathcal{E} \left[-\Delta_{x_k}^{x_k} \log p(x_k | x_{k-1}) \right] + \mathcal{E} \left[-\Delta_{x_k}^{x_k} \log p(y_k | x_k) \right] \quad (6.40)$$

Further, we assume that the initial state x_0 has known probability density function $p(x_0)$. ∇ and Δ are operators of the first and second-order-partial deriva-

tives,

$$\nabla_{x_k} = \left[\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n} \right] \quad (6.41)$$

$$\Delta_{x_k}^{x_k} = \nabla_{x_k} \Delta_{x_k}^T \quad (6.42)$$

The filter efficiency is defined as [380]

$$\eta_k = \frac{\sqrt{\text{PCRB}_k}}{\text{RMSE}_k} \times 100\% \quad (6.43)$$

between $0\% \leq \eta_k \leq 100\%$ and where $\eta_k = 0\%$ means that a poor filter was used. Another quality factor is the relative improvement, and calculated as [380]

$$\text{RTAMSE} = \sqrt{\frac{1}{\text{MC} \cdot K} \sum_{k=1}^K \sum_{i=1}^{\text{MC}} \left(\hat{x}_k^{(i)} - x_k^{(i)} \right)^2} \quad (6.44)$$

$$RI = \frac{\text{RTAMSE}_{ref} - \text{RTAMSE}_{filter}}{\text{RTAMSE}_{ref}} \times 100\% \quad (6.45)$$

where RTAMSE_{ref} represents the Root Time Averaged Mean Squared Error for the EKF, K carries the number of observation points. In case that the partial differential equations of the logarithmic probability density functions cannot be determined, alternatives to the PCRB have to be applied, such as WEISS-WEINSTEIN, BHATTACHARYA, Ziv-Zakai bound just to name a few [380].

Ziv-Zakai Bound was derived for time-delayed signal parameter estimation. In contrast, the CRLB gives in general a very good prediction of estimation error in applications where the signal-to-noise ratio is in expected range, the noise level small. In high-noise scenarios, the CRLB tends to be on the optimistic side and hence, for high-noise scenarios, a closer bound shall be used. The Ziv-Zakai bound (ZZB) shall be introduced, which was designed for this reasons. While for CRLB, only the measurement model is required, ZZB additionally need input from the prior $p(\Theta)$ and requires higher numerical calculations. On the other side, it allows the ZZB to include any a-priori information. Considering the estimation of a continuous vector random parameter $\theta \in \Theta$ through noise measurement $x \in \chi$. For any estimator $\hat{\theta}(x)$, the estimation error $\epsilon = \hat{\theta}(x) - \theta$, the error correlation function is defined as

$$R_\epsilon = E\epsilon\epsilon^T = E(\hat{\theta}(x) - \theta)(\hat{\theta}(x) - \theta)^T \quad (6.46)$$

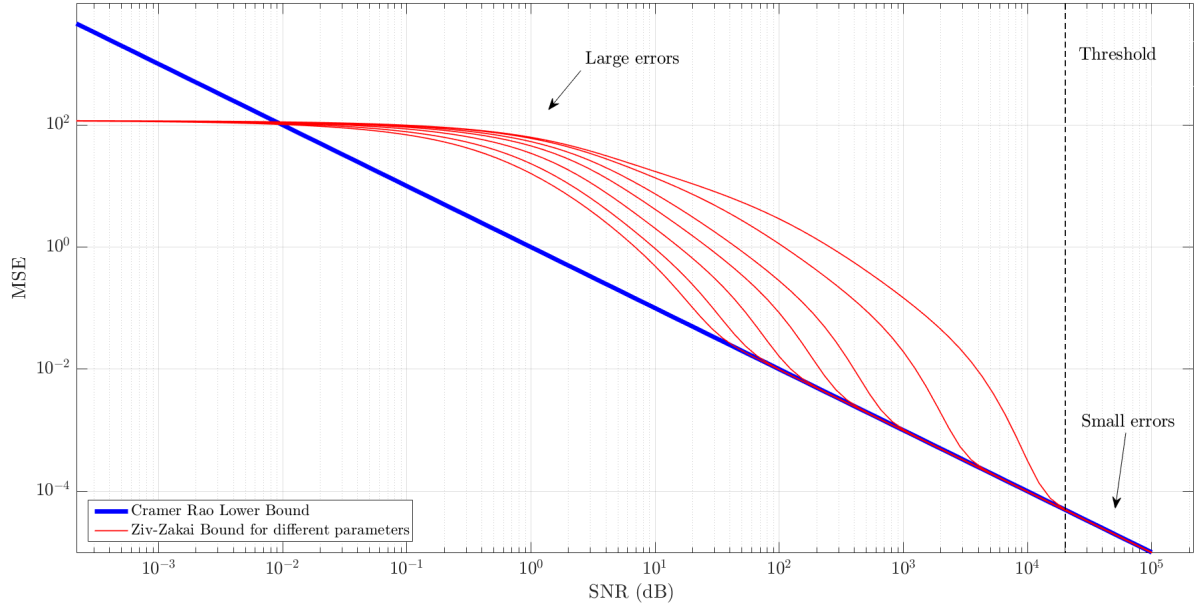


Figure 6.29: MSE bounded MLE performance vs. SNR for estimation for CRLB and Ziv-Zakai.

When θ is a D -dimensional vector, we want to estimate the lower-bounding $a^T R_\epsilon a$ for any D -dimensional vector a . By finding the unit vector a a bound on the MSE of the corresponding component of θ is found. Now, the vector bound is calculated by

$$a^T R(\hat{\theta})a \geq \frac{1}{2} \int_0^{\inf} \nu \left\{ \max_{\delta: a^T \delta = h} \int [p(\theta) + p(\theta + \delta)] P_{\min}(\theta, \theta + \delta) d\theta \right\} h dh \quad (6.47)$$

Lastly, the robustness of the filter shall be defined. Since little information regarding analytical description of non-linear filters can be found, a rather empirical method is used, in fact the Monte-Carlo simulation results are analyzed, by identifying the quantity of diverging MC simulation results per test case. By diverging we mean that for example the covariances will lose the positive definite property, that is after division by zero, or the results from reference model vs. estimation model are too far apart exceeding the defined threshold.

$$J_{\text{robust}} = \left(1 - \frac{\#_{\text{div}}}{MC} \times 100\% \right) \quad (6.48)$$

where an empirical conclusion regarding the robustness is defined, in fact $0\% \leq J_{\text{robust}} \leq 100\%$ with $J_{\text{robust}} = 0\%$ means that the filter is not robust.

2. **CPU load** The different level of complexity of algebraic operations will define the CPU load on the embedded target computer, which for battery management systems are generally low-cost controllers. Since the BMS SW does only consist partly of the algorithms for SoX determination, missing the safety functions, communication, balancing, fault diagnostics..., the maximal reserved CPU load for state and parameter estimators are limited and generally for the entire BMS software must not exceed 75% of available CPU performance. Since BMS is a safety-critical application, at all times the real-time execution must be fulfilled. However, it is very difficult to give exact measures about how much resources on an ECU may be used, it depends on the ECU architecture and clock frequency and hence makes it impossible to convert runtime from desktop simulation PC (DELL Core i7, 8GB RAM) to the multiple automotive ECUs on the market. Also, the simulation will be done in the Matlab/SCADE environment, the execution time also depends on how much Matlab parallelizes the algorithms, for which the user does not have any control. It is possible to analyze how much recursions/iterations (loops) are involved for each filter, however, that is not in the scope of this thesis and other rating will be used. Rather, a relative worst execution time tendency can be discussed. From the generated C-code for an given filter (reference filter EKF), the execution time on target can be compared against a second filter family which will also be run on the target ECU by repeatedly measuring the execution time. From there the difference in execution time can be used to estimate a general trend. This method can be used if no professional static SW analyzer is available (including ECU model of target etc). Knowing execution time on the used computer it is possible to make useful approximations of the running time on a different machine and therefore, a target platform independent conclusion can be provided. First, the simulations are run in Matlab/SCADE environment, first for the EKF reference case. The execution time for each function is measured and stored, this will be done for all corresponding algorithms. The absolute change in execution time between the filter is then calculated as following The relative performance time PT is defined as

$$PT = \frac{\sum_{i=1}^{MC} t_{\text{sim,ref}^{(i)}} - \sum_{i=1}^{MC} t_{\text{sim,filter}^{(i)}}}{\sum_{i=1}^{MC} t_{\text{sim,ref}^{(i)}}} \times 100\% \quad (6.49)$$

This gives an understanding in regards the execution time amongst the filters towards the reference case increases.

6.8 Results

In the context of this thesis, a new modular MATLAB/TensorFlow (deep learning) based toolbox was developed for testing and evaluating the combination of "best" model-filter selection in terms of *which method (filter & model) reaches a certain efficiency $\rho = X\%$ with a certain stability $\zeta = Y\%$ with the least implementation effort and lowest compu-*

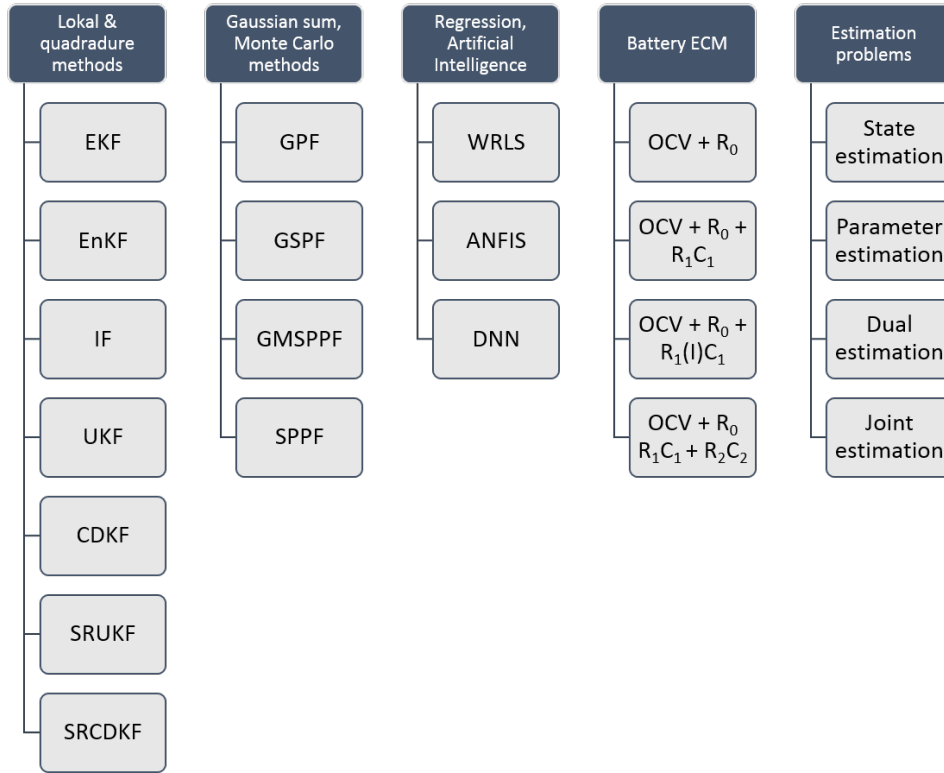


Figure 6.30: Implemented algorithm library in ReBAAI toolbox

tational cost. The toolbox was inspired by [458] and heavily extended to the **Recursive Bayesian Estimation & Artificial Intelligence Toolbox for Battery Monitoring (ReBAAI-Toolbox)** with the following attributes:

1. Evaluation of methods for classical and Bayesian estimation under consideration of various battery equivalent circuit models. The evaluation considers all the introduced quantitative metrics, such as different (re-)sampling methods, filter accuracy in comparison to reference filter (EKF), robustness, implementation effort and computational cost, and based on the results a recommendation of the model-filter combination can be provided.
2. Modular approach, ReBAAI is implemented so that the end-user only has to focus on implementing additional models without worrying about updating filter framework. The user defines the system model interface and parameters (likelihood, innovation, linearization, dimensions, noises,) of the generalized state-space model which will be used in the filter framework as input. The toolbox with its dynamic state space model framework allows to choose from a variety of probabilistic distributions (uni- and multivariate) for system and measurement noise.
3. Graphical user interface⁽ⁱ⁾ was developed for Matlab, in which the user can choose

⁽ⁱ⁾Due to the fact that during the implementation phase the decision was taken that the emphasis shall

the GSSM framework, model dimensions, initial parameters (e.g. SoC, impedances,...), input signals and filter.

4. ReBAAI filter and model library consists of classical approaches but mostly on Bayesian filters and methods from AI and machine learning techniques, as decisions are made based on data not on opinions.

6.8.1 Application Example - Non-linear Battery State Estimation

The toolbox can generate different noise source types, such as Gaussian, combination of independent Gaussian, Gaussian mixture and scalar gamma noises. For the first test, only the Gaussian noise source is added to the voltage and current signal. Additionally, the initial SoC level is set with a random value between real value and error bound of 20% resulting in a first indication of which estimator has the best performance.

One of the most important aspect to evaluate the filter's dynamic performance is the estimation accuracy. Normally the voltage error between the estimated value and the measured value are used as the indicator of the model accuracy. However, additionally, the estimated SoC value was used considering its the core value that will be considered in the real EVs application. The indicator function for the model accuracy was defined by root mean square error method. Experiments from different temperatures, battery degradation state and various road conditions (city road, high way and mixed road) are used from the FIAT 500 to get a comprehensive evaluation of the overall battery model performance. Aging is simulated via MONTE-CARLO simulation.

The improvement of first order and second order RC model compared to the RINT model is quite obvious. The more complicated models can achieve better the accuracy but will also cost more computation power instead. Apart from the model comparison, the algorithm comparison is also important to choose a reliable and simple SoC estimation method. The results represent the maximum, minimum and average RMS errors of different adaptive methods from the same simulation result as described above. And in most of the cases particle filters family show better result compare to KALMAN filters family. However the drawbacks are also quite obvious, which requires much more computation power and it's way above the limit of a cheap microprocessor. The safety and reliability performances of the electric vehicle is really important as it requires that the BMS can handle harsh environment conditions. The robustness of the different estimators need to be evaluated in order to test their dynamic performances. The robustness evaluation process consist of a poor initial state value, parameter uncertainties, different load profiles, different aging states and sensor noise. From the previous study conclusion, considering both the computation power and estimation accuracy using KALMAN filters with sec-

be on increasing the filter library, at the time of the publication of this work the consideration of all filters in the GUI was still ongoing (e.g. ANN).

ond order RC model and Particle filters with first order RC model makes more sense to compare the robustness performance of these methods. The filter tuning for the EKF and UKFs rely on the tuning parameter for noise covariances and the selection was not automatized in the evaluation.

6.8.2 Results - State Estimation

Nowadays the most common and widely used method is Ampere-hour counting method which is easy to implement and relatively cheap in terms of computation power compared to other method. But on the other hand it also have sever drawbacks. Poorly initial value and accumulated measurement error may cause a big deviation between the real and estimated value. This can be pretty annoying when a car suddenly stops in the middle of the road whereas sufficient battery capacity is indicated. The adaptive filter technology will not suffer from such deficits but the converge time will highly depend on the algorithm and also the tuning factors of the corresponded filters. The plots below show the result when the initial SOC value has 20% error from the real SOC value. The main observation from this test is that all filters can converge back to real SOC value in a very short period of time. In most of the cases Particle filters converge faster than the KALMAN Filters family.

MODEL 1							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	6,55	0,00	88,11	15,03	0,00	2,97	0,02
UKF	6,55	-0,01	88,11	6,37	61,18	2,97	0,02
CDKF	6,55	-0,01	88,11	6,76	58,79	2,97	0,02
SRUKF	6,55	-0,01	88,11	6,81	58,46	2,97	0,02
SRCDKF	6,55	-0,01	88,11	7,27	55,69	2,97	0,02
PF	7,35	20,48	89,40	9,02	44,99	2,65	0,03
GMSPPF	6,10	-18,79	87,11	90,00	-448,75	3,22	0,03
GSPF	5,49	-43,29	85,78	34,87	-112,58	3,55	0,02
SPPF	5,33	-51,30	85,38	809,38	-4834,73	3,65	0,02

Figure 6.31: Results of Monte-Carlo simulation for state estimation model 1, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

MODEL 2							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	11,76	0,00	89,60	16,40	0,00	2,60	0,07
UKF	11,77	0,13	89,60	7,23	55,90	2,60	0,07
CDKF	11,77	0,10	89,60	7,98	51,32	2,60	0,07
SRUKF	11,77	0,13	89,60	7,49	54,34	2,60	0,07
SRCDKF	11,77	0,10	89,60	8,37	48,98	2,60	0,07
PF	19,32	61,88	93,61	12,70	22,56	1,60	0,011
GMSPPF	13,82	21,35	90,89	128,46	-683,19	2,28	0,012
GSPF	23,12	58,90	93,80	60,87	-271,13	1,55	0,013
SPPF	10,57	-23,92	88,42	886,83	-5306,92	2,90	0,005

Figure 6.32: Results of Monte-Carlo simulation for state estimation model 2, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

MODEL 3							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	9,49	0,00	90,71	16,60	0,00	2,32	0,08
UKF	9,48	-0,02	90,70	7,51	54,77	2,32	0,08
CDKF	9,49	0,00	90,71	8,29	50,04	2,32	0,08
SRUKF	9,48	-0,02	90,70	7,77	53,22	2,32	0,08
SRCDKF	9,49	0,00	90,71	8,65	47,92	2,32	0,08
PF	13,26	47,74	93,30	14,13	14,90	1,67	0,010
GMSPPF	8,76	-27,27	89,65	132,14	-695,87	2,59	0,012
GSPF	17,92	54,98	94,16	63,46	-282,24	1,46	0,013
SPPF	7,61	-55,35	88,42	915,19	-5412,22	2,90	0,005

Figure 6.33: Results of Monte-Carlo simulation for state estimation model 3, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

MODEL4							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	100,00	0,00	99,85	0,00	0,00	0,04	0,003
UKF	100,00	-11,68	99,84	54,76	73,61	0,04	0,003
CDKF	100,00	-9,23	99,84	49,62	75,15	0,04	0,004
SRUKF	100,00	-11,68	99,84	53,16	76,98	0,04	0,003
SRCDKF	100,00	-9,23	99,84	46,71	74,41	0,04	0,004
PF	319,70	-88484,74	96,42	0,14	51,03	0,89	0,005
GMSPPF	366,72	-38472,20	97,51	-738,79	-288,17	0,62	0,008
GSPF	123,35	-214926,59	93,67	-303,70	-68,13	1,58	0,004
SPPF	100,00	88,95	99,95	-5393,14	-1489,42	0,01	0,001

Figure 6.34: Results of Monte-Carlo simulation for state estimation model 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

From the 100 Monte-Carlo simulation, it is obvious, that the errors and spread are rather small, that in general the more complex model the better accuracy for voltage estimation. For simple battery SSM, there is no improvement using complex estimators, this is different for complex models, then complex algorithms lead also to better voltage estimation. The robustness, RTAMSE is rather constant throughout models and estimators, while the computational costs gradually increase with model and filter complexity. For state estimation, the biggest improvement comes from using a second-order battery model, reducing SoC error from 2.4% to less than 1%.

6.8.3 Results - Joint Estimation

The performance characteristics of a lithium-ion battery depends on internal (degradation, temperature, SoC...) and external (driver behavior, ambient temperature...) conditions, hence the parameters will vary and these test cases will show how the different estimators can handle uncertainties in model parameters. Additionally, current sensor bias is evident for this kind of sensors, therefore, a Gaussian distributed random variable with corresponding variance is added with a maximum error of 0.1% maximum load current causing filters to drift away from the actual SoC. Regarding the voltage sensor variance, the standard deviation is set up to 5mV. Figure 6.37 summarizes the MC simulation results for joint estimation, namely state- and parameter estimation for 3 different SSMs.

MODEL 2							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	43,33	0,00	95,03	16,98	0,00	1,24	0,025
UKF	92,22	77,93	97,66	107,05	-530,58	0,58	0,016
CDKF	72,88	64,66	97,04	71,97	-323,94	0,74	0,012
SRUKF	70,36	62,08	96,94	109,18	-543,16	0,77	0,017
SRCDKF	73,22	64,98	97,06	71,11	-318,86	0,74	0,017
PF	68,52	46,32	96,67	545,10	-3111,02	0,83	0,244
GMSPPF	14,13	96,52	95,90	137,82	-4024,56	0,25	0,014
SPPF	7,43	95,51	94,57	390,24	-11578,36	0,36	0,008

Figure 6.35: Results of Monte-Carlo simulation for joint estimation for model 2, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

MODEL 3							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	17,70	0,00	95,28	17,38	0,00	1,18	0,013
UKF	14,12	-57,18	94,09	128,24	-638,01	1,48	0,026
CDKF	7,11	-519,26	88,26	85,13	-389,92	2,93	0,138
SRUKF	15,66	-27,81	94,67	130,12	-648,81	1,33	0,011
SRCDKF	15,84	-24,95	94,73	84,55	-386,59	1,32	0,011
PF	22,96	-5,20	95,58	557,62	-3109,00	1,10	0,098
GMSPPF	7,93	95,87	96,05	137,50	-3942,40	0,99	0,012
SPPF	3,87	88,63	92,99	402,32	-11727,80	0,75	0.0068

Figure 6.36: Results of Monte-Carlo simulation for joint estimation for model 3, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

MODEL4							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	96,09	0,00	95,93	17,07	0,00	1,02	0,011
UKF	98,48	4,79	96,02	176,51	-934,27	0,99	0,006
CDKF	96,06	-0,07	95,92	114,27	-569,57	1,02	0,009
SRUKF	99,72	7,14	96,07	178,68	-947,03	0,98	0,006
SRCDKF	98,92	5,63	96,04	113,44	-564,73	0,99	0,006
PF	90,00	-8,50	95,96	539,25	-3059,80	1,01	0,100
GMSPPF	6,93	81,17	35,15	145,47	-3626,41	0,16	0,008
SPPF	82,43	99,95	96,13	412,24	-10460,97	0,97	0,005

Figure 6.37: Results of Monte-Carlo simulation for joint estimation for model 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

From the result, it can be stated that already for basic SSM model, more complex estimators result in better accuracy, e.g. the UKF has an efficiency of 92% outperforming the EKF with 43%. For higher order SMM, the gap closes in, making the EKF competitive. In general, the non-Gaussian distribution filters outperform the Gaussian distribution filters (e.g. local methods), especially for voltage prediction, which also improves with higher order SSM. The efficiency between EKF and UKF differs only about 2%, while similar robustness can be reached. The best result is reached by the GMSPPF, with the highest accuracy for voltage (parameter estimation) and SoC.

6.8.4 Results - Dual Estimation

Figure 6.40 summarizes the MC simulation results for dual estimation, namely state- and parameter estimation for 3 different SSMs.

MODEL 2							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	28,88	0,00	80,66	17,59	-	4,83	0,180
UKF	79,70	86,87	92,99	11,05	37,17	1,75	0,006
CDKF	79,69	86,87	92,99	12,64	28,11	1,75	0,006
SRUKF	91,91	90,13	93,92	11,79	32,95	1,52	0,006
SRCDKF	62,60	78,72	91,08	11,44	34,96	2,23	0,004
PF	91,35	91,38	94,71	14,03	20,21	1,32	0,143
GMSPPF	29,75	91,68	95,00	81,90	-373,93	1,25	0,154

Figure 6.38: Results of Monte-Carlo simulation for dual estimation for model 2 and 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

MODEL 3							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	13,40	0,00	67,45	17,45	0,00	8,14	0,180
UKF	62,19	95,36	92,99	11,62	33,41	1,75	0,006
CDKF	62,18	95,36	92,99	13,18	24,43	1,75	0,006
SRUKF	71,13	96,45	93,87	12,23	29,92	1,53	0,006
SRCDKF	48,97	92,52	91,10	11,88	31,88	2,23	0,004
PF	70,04	95,84	93,54	13,90	20,34	1,61	0,143
GMSPPF	8,48	96,34	93,63	84,12	-392,38	1,59	0,147

Figure 6.39: Results of Monte-Carlo simulation for dual estimation for model 3, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

MODEL4							
Algorithms	Efficiency	Improvement	Robustness	Simulation	Calc.costs	SoC Error	Voltage Error
	[%]	[%]	[%]	[s]	[%]	[%]	[V]
EKF	61,14	0,00	92,95	17,30	0,00	1,76	0,125
UKF	100,00	69,17	96,08	11,95	30,94	0,98	0,001
CDKF	100,00	68,94	96,06	13,72	20,70	0,98	0,000
SRUKF	100,00	69,22	96,08	12,58	27,28	0,98	0,001
SRCDKF	100,00	71,45	96,23	12,35	28,63	0,94	0,003
PF	66,48	-5,99	93,01	13,80	20,20	1,75	0,023
GMSPPF	6,54	-41,57	92,15	87,08	-390,36	1,96	0,005

Figure 6.40: Results of Monte-Carlo simulation for dual estimation for model 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.

In direct comparison to joint estimation, here separate KALMAN filters are used for state estimation and parameter estimation. The joint estimation is computational more expensive, as the matrix operations for dual estimation is smaller. For the case of complex model and complex algorithm, the calculation time is almost 10x. Even though, by decoupling state- from parameter estimation and the loss of cross-correlation, the results for dual estimation is comparable, sometimes resulting in worse, but also better results. Especially for local approximation filter, the joint estimation has advantages. The superior estimators in terms of accuracy in estimation are the filters based on the quadrature methods. The SRCDKF shows very good parameter estimation, while the GSPF results with best SoC results. The robustness is the lowest for the EKF, as well as the SoC estimation accuracy.

6.8.5 Final Verdict

The safety and reliability performances of the electric vehicle is really important which requires the BMS can handle harsh environment conditions. The robustness of the different estimators need to be evaluated in order to test their dynamic performances. The robustness evaluation process consist of poor initial state value, parameter uncertainties, different load profiles, different aging states and sensor noise.

6.8.5.1 Poor Initial State Vector

Nowadays the most common and widely used method is Ampere-hour counting method which is easy to implement and relatively cheap compare to other methods. But on the

other hand it also have sever drawbacks. Poorly initial value and accumulated measurement error may cause a big deviation between the real and estimated value. This can be pretty annoying when a car suddenly stops in the middle of the road whereas sufficient battery capacity is indicated. The adaptive filter technology will not suffer from such deficits but the converge time will highly depending on the algorithm and also the tuning factors of the corresponded filters. In can be said, that the RMSE and the convergence rate increases with higher initial SoC error while the PF converges the fastest in comparison the EKF. That can be triggered by the fact that EKF applies a polynomial OCV-SoC representation. The result when the initial SOC value has 20% from the real SOC value show that all filters can converge back to real SOC value in a very short period of time. Especially particle filter which can track back the real SOC value in few seconds.

6.8.5.2 Parameter Error

Considering the battery itself is a electrochemical device. Not only the characteristics but also the internal parameters will be changed under different conditions eg. temperature, SOC and aging states. However the focus right now is not on how to update the parameters according to environment change. Instead the estimator or the algorithm itself should have the ability to handle the incorrect model parameters.

6.8.5.3 Sensor Drift

In the real vehicle application the major sensor error will come from the current sensors. It will not only affected by the sensors resolution but also the EMF problem within the vehicle. Here the sensor noise can be represent by a normal distribution with zero mean value and maximum deviation of 1A. In order to test the limits of the filters another set of test with maximum deviation of 2A is also given here as comparison. The voltage sensor drift will also be investigated here, the variance of the voltage sensor is also represented by a normal distribution with zero mean value and the standard deviation of 10mV.

The question of

Which method (filter & model) reaches a certain efficiency $\rho = X\%$ with a certain stability $\zeta = Y\%$ with the least implementation effort and lowest computational cost.

can not be easily answered, because requirements are so different, e.g. will the algorithm run on 16bit controller (as of today), or on a high-performance centralized vehicle CPU (e.g. automated driving computer). Looking at state estimation using model 1, the particle filter with a standard deviation of 1.5% shows the best results, for model 3 it drops to 1% while the best filter, the GSPF is around 0.5%. Looking at higher model 4, the SPPF shows the strongest performance in terms of accuracy with standard deviation of

0.2%, followed by UKF, CDKF, and EKF. Considering JE and model 2, the PF and EKF show similar performance (std. dev. 0.9%), while the SRUKF/SRCDKF have a std. dev. 0.2%. This trend follows for model 3, where the PF shows the best performance in terms of std.dev., while for model 4 all filters show similar std.dev. results. For DE, the PF and EKF for model 2/3 shows best results (std.dev. 0.2%), while the best results are gained with model 4, all filters perform similar, while the UKF is the best. The PF has problems caused by wrong tuning parameters. However, rather than having the most complex algorithms, improvements are easier by adding additional RC circuits to the model. The well known EKF can be seen as still competitive until up-to-day, however, to get the last % out, 1% accuracy increase for state estimation can save the OEM in long run money, however, with the constraint of much heavier calculations. It can also be concluded, that the PF family outperformed the EKF/UKF filters in terms of accuracy for higher order models / non-linearities. The tuning parameters have an significant impact on the results. A lot of time and effort can be spend on this task to find an optimal solution, unless an automated selection is in place. The UKF is free of local approximations (e.g. EKF) and no additional Jacobian's are necessities to be computed. Since it stores the 2nd order moments, UKF shows better results for higher order models. Here, the PF family may be a better substitute as it handles non-Gaussian noises.

7 Hardware-in-the-Loop Testing for Battery Management System

No amount of testing can prove a software right, a single test can prove a software wrong.

— Amir Ghahrai

The BMS is a part of a distributed automotive electrics/electronics (E/E) system whose design follows the automotive V-model breaking down into chronological cycles such as development, integration and test process [459]. To promote a decrease in software complexity, model-based development is an recognized method. Battery models may help to analyze functional problems. Functional safety depends on the correct functioning of the E/E system and is particular important. At all times and test levels safety components of E/E must ensure to be absolutely reliably for a later successful testing and validation of the battery storage system. The following test levels are established in the automotive industry [459]:

Model-in-the-Loop (MiL) comprises the co-simulation of the physical plant model (e.g. battery cell or pack model) with the control strategy on the algorithmic level (e.g. battery state estimation). Here, the response of the control logic is in focus and compared to the expected system response. If the results are correct, the control algorithm can be integrated on the target platform using e.g. automated code generation tools.

Software-in-the-Loop (SiL) is used after code generation where the C-code is simulated together with the battery cell/pack model from the MiL cycle. The C-code may compiled out of the toolbox using the host-PC periphery instead of the intended target ECU. Some differences in precision and performance may arise.

Processor-in-the-Loop (PiL) still focusing only on control algorithms testing but avoiding the disadvantages of SiL. In PiL the C-code is specifically generated for the target ECU platform and executed on a processor or evaluation hardware. Due to the synchronization between the processor board and the simulated platform the times and allocations can be investigated.

Hardware-in-the-Loop (HiL) is the last finale step before the integration into the overall vehicle system. Control software and hardware are combined into ECU network and tested in an experimental closed-loop environment. All relevant control, sensor and actuator signals are generated by the plant model and provided to the target hardware platform.

Virtual environment Hardware-in-the-Loop (vHiL) defines virtual prototyping for hardware dependent software development and makes the dependency on physical hardware unnecessary. It allows for remote usage or parallel execution using cloud-computing. The real ECU and plant are unnecessary and tests can be done before completion of the real system components.

Vehicle system integration is used in order to investigate the performance of control algorithms on the target platform during real conditions, therefore, it is moved into the test vehicle for further optimization and to secure the manufacturing cycle.

In comparison with tests conducted on real Li-ion batteries, BMS tests conducted on a Hardware-In-the-Loop (HiL) battery simulator have the following features:

- **Safety:** Critical scenarios and operating points can be simulated without causing damage to the hardware (real plant) or putting human life at risk.
- **Costs:** HiL tests are relatively cost-efficient in comparison to real-plant testing. Especially for battery storage systems, a virtual environment, e.g. change of battery state like SoC can be easier configured than in real leading to fewer test drives and running test bench experiments.
- **Controllability:** Virtual environments are easier to modify, e.g. the local measurement unit of master controller of a battery management system can be tested in different control loops. Moreover, real systems can be tested with different emulated controllers (e.g. algorithms). Different ambient conditions can be simulated by simply changing the parameters of the environment.
- **Usability:** Setting the operating point and parameters allows to repeat the tests directly. Therefore, critical scenarios and failure injections can be tested reproducibly for optimization purposes.
- **Scalability:** To meet rapidly changing needs in system designs and to allow for a fast and appropriate reproduction (e.g. different battery storage system topologies and sizes).
- **Hybrid testing:** Some systems are too complex to model or are provided by a supplier. HiL simulation enables to integrate parts of the system in the control loop.

-
- **Rapid prototyping:** The hardware and software of a battery storage system can be already tested during the development process.
 - **Automation:** Automated Hardware-in-the-Loop testing enables to execute tests at any time. The tests can be setup including failure insertion or calibration or different environment (e.g. hardware can be tested in parallel for different states and temperature) increases the test coverage.

These features may be more cost and time effective, more flexible and traceable, easier to reproduce and safer beyond the normal range of operation especially in comparison to vehicle testing. This is particularly the case of tests at early stages in the development process or during fault simulation. However, the use of a HIL battery simulator requires the development of SW and strategies for testing. While the possibilities are immense, it should be noted that the greater the level of complexity of the tests, the higher the demands for ad hoc development of SW and strategies. With regard to the latter, there is not a universal definition and there may be different points of view. Therefore different strategies may be followed, which may be classified into many different ways according to the testing level, methods or processes. We discuss the theoretical principles of BMS testing and to present a practical approach to develop ad hoc SW and strategies for BMS testing at system level on a commercial HIL simulator. A structured and comprehensive system test is an obligatory prerequisite for dependability and stability of a BMS in the difficult operation environment. Normally, developers and manufacturers of BMSs require extensive testing of controller HW and SW, such as analog front-end (AFE) and performance of generated control code. This is where the design process for intelligent BMS benefits with the contribution of Hardware-in-the-Loop systems. For testing safety-critical functions for BMS components and electric vehicle ECUs and networks, the ISO 26262-6 recommends HiL testing to ensure that safety requirements are met on the software level side. Subsection ISO 26262-4 considers HiL testing on system level (e.g. fault injection, functional testing) is extremely valuable. This work applies HIL for BMS algorithm development and firmware regression testing applying failure insertion unit (FIU).

After an extensive literature analysis given in [460][461][462], it can be stated that the current state-of-the-art HiL industry standards for testing battery management systems is still at an early stage and solutions present some HW or SW limitations, such as inaccuracy, functions, precision, resolution or iteration rate. These systems, for purposes of general high-level analysis of a BMS, simpler battery models are used in industry. However, more accurate models must be implemented with the aim of ensuring that complex functionalities of a BMS, such as diagnostic algorithms for SoC, SoF or SOH estimation as well as safety functions must be analyzed and validated using HIL simulation to detect bugs in an early stage.

Therefore, a multi-cell electro-thermal Li-ion battery (LIB) [460][461][462] model is integrated, coupled with an aging model, characterized and validated based on experimental data, and integrated in a system-level simulation. Then, the LIB system model is

converted to C code and run in real-time with the HIL simulator. Finally, in order to demonstrate the capabilities of the setup, experimental results of BMS tests over a certain set of exemplary cases are shown.

7.1 Hardware-in-the-Loop Simulator

The commercial HIL simulator test bench is provided by dSPACE GmbH and interfaces with the BMS under test and a host PC either via electrical, optical or CAN bus interfaces 7.1. The voltage emulation at cell level is performed with several EV1077 battery cell voltage emulation boards. Each of these controllable buffer amplifier boards can emulate four cells and be configured in order to match specific test requirements. It can provide a continuously controllable voltage in a range of 0-6 V 1.5 mV for each of the outputs. This relatively wide voltage range allows not only emulation of both over-voltage and under voltage scenarios, but also emulation of short circuit faults between cells. The EV1077 voltage outputs are galvanically isolated, allowing series connection. In the experiments presented 8 EV1077 boards are used to emulate 32 cells in series, but the setup could be extended to allow emulation of 128 cells (or group of cells in parallel) in series. Regarding the dynamic response, any voltage reference value step can be followed in less than 500ms. This feature combined with high computational power of the DSP and a fast data transmission enables a maximum update rate for all the cell's voltages of 1 kHz. Furthermore a bidirectional balancing current of maximum 1 A is allowed, which enables testing of both active and passive cell balancing systems. A number of other I/O cards are used to provide communication interfaces (e.g. CAN Bus) or emulation of temperature sensing or battery pack current sensing based continuously controllable voltage outputs. E.g. the implemented vehicle dynamics model generates the propulsion power demand which is converted to current profile and used as input to the underlying battery cell models. Additionally, the generated load current has to be accessible to the BUT. The calculated voltage drop is passed via a LiBMS-HiL digital/analog converter output to the shunt input reading of the BMS. The LiBMS-HiL is not capable of handling negative voltages, only [0]V - [10]V, an additional conversion board that scales this to $\pm[400]$ mV for the BMS was designed. The overall system setup is as following, LiBMS-HiL is connected to the host PC via a fiber optic cable to prevent ground loops and also to load the compiled C-code and data logging while running the simulation. The host PC also runs a fully automated programmed test sequence which requires no supervision (AutomationDesk). The signal conditioning is located above the expansion box and was adopted to the specifications discussed jointly with Aalborg University. The FIUs are used to generate electrical failures such as short circuit to the battery pack electrical system. The voltage supply for BUT is provided by controllable power unit. The voltage supply of the slaves is implemented via the individual battery cell voltages, normally 3-4 cells per slave are sufficient. The LIB system simulation model, illustrated in Fig.7.2, is developed off-line on MATLAB/Simulink SW from Mathworks. An implementation SW provided by dSPACE, so called Real-Time Interface (RTI), provides blocks that link the

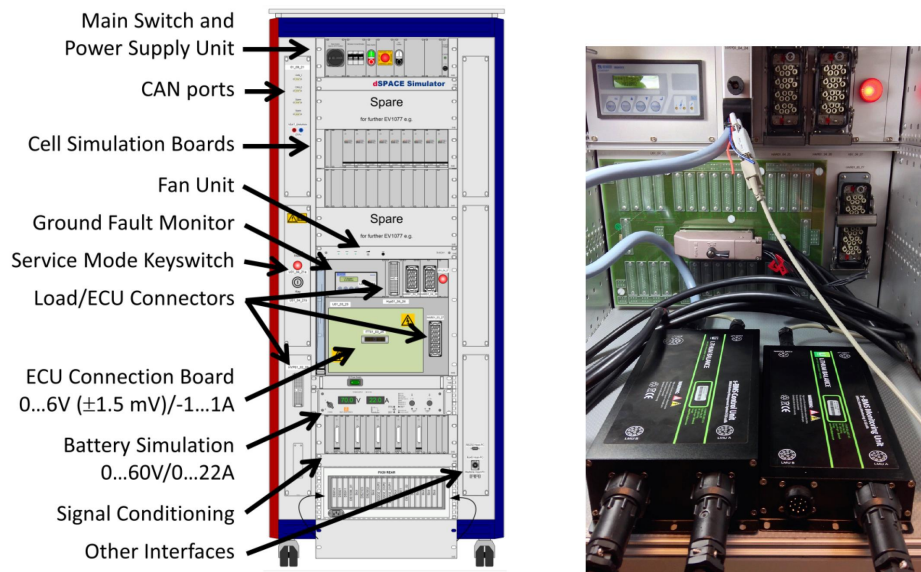


Figure 7.1: *Left:* Overview of the dSPACE HiL simulator; *right:* BMS under test: Lithium Balance sBMS. [460]

I/O capabilities of the dSPACE HIL simulator to the MATLAB/Simulink models.

7.2 HIL Battery System Simulation Framework

The HIL simulator is able to provide signals for testing a wide variety of BMS functions, such as cell protection; safety validation; measuring of current, voltage and temperature; balancing of cells; analysis and parameter prediction; logging, telemetry and external communications. Therefore, bypassed signals and sensors, which are normally connected to the BMS are emulated by the HiL. It should be pointed out that the HiL must plausibly reproduce the "real" system to guarantee realistic tests and analyses. The schematic structure of the battery storage system consist of interconnected battery cells, contactors for pre-charge, fuses, interlock line as well as the distributed battery management components. In this context, to validate the BMS and monitoring algorithms, at least one battery pack with CSC must be simulated. In contrast to low dynamic battery simulation, e.g. in system design for stationary applications, models for BMS monitoring testing require higher dynamics and accuracy for individual cell voltages. Each cell has its own dynamic characteristics, representing cell voltage variations also during charging, different behavior for sudden change in load or aging states. Foreverymore, the battery storage systems interacts with further drive-train components or electric comfort loads in the vehicle. Closed-loop operation enables to communicate with all other emulated components in the simulated HiL system. A multi-layered procedure is designed to meet requirements and demand for BMS HiL system, both at high (BMS monitoring) and low level (functional safety) software, as well as the electrical system level.

7.2.1 Lithium-Ion Electro-Thermal Multi-Cell Model

To apply the LiBMS-HiL simulator for BMS validation, the behavior of each individual cell has to be simulated. Therefore a real-time capable battery system model is developed simulating all physical effects necessary for HiL testing of BMS hardware and control algorithm verification. It consists on the one side of the LIB electro-thermal multi-cell ECM model, and on the other side of the interface model which conditions the signals for the simulator hardware. First, the ECM model is coupled with an aging model which is designed and characterized for a library of different commercial available lithium-ion cells including battery cells with different formats (cylindrical, pouch, prismatic), capacities (2Ah, 53Ah, 100Ah) and chemistries (NMC, LTO, LFP), including manufacturers such as Sanyo, Panasonic and SkyEnergy.

A suitable equivalent circuit model is used to represent the dynamic behavior, its parameters are extracted by means of electrochemical impedance spectroscopy following [438], where a method is described on how to transform of ECM frequency domain into time domain model equations. In this case, the low and high frequency processes include internal resistance and is modeled as ohmic resistor, solid electrolyte interface film (SEI), cell dynamics such as charge transfer and double layer (short time behavior) is modeled via ZARC element, while diffusion (long time behavior), which typically lies within BMS

sampling rates is modeled via Warburg element. The details how the model was implemented (approximation of ZARC element, parameter identification, the implementation of the ZARC model in the time domain, thermal model, aging model) can be found in [460][461][462], therefore, it is only mentioned on higher level.

7.2.2 Hardware-in-the-Loop System Model

The HIL simulation model consists not only of the electro-thermal multi-cell ECM coupled with an aging model, but rather it is an overall system model which was implemented considering thermal and electric properties on cell and system domain and its interaction considering all individual system components typical used in battery pack design, such as thermal system (cooling and heating), electrical system (battery disconnect unit), communication (CAN, isoSPI), connections to BMS and vehicle, charger, sensor (current, temperature), standardized driving profile generation and fault insertion unit (FIU).

Temperature sensor model A BMS may monitor temperature sensors throughout the whole battery system. Although, battery systems may differ in the number, location and type of devices used to measure the temperature. The commercial BMS under test uses Negative Temperature Coefficient (NTC) thermistors. One temperature sensor per cell, located in a central position, is assumed for emulation purposes. NTC thermistors are thermally sensitive semiconductor resistors that exhibit a non-linear decrease in effective resistance as temperature increases. This electro-thermal behavior can be modelled analytically, by means of the Steinhart-Hart equation.

$$\frac{1}{T_s} = \frac{1}{T_{s,0}} + \frac{1}{B} \cdot \ln \left(\frac{R_s}{R_{s,0}} \right) \quad (7.1)$$

Where all the temperatures are in Kelvin and all the resistances in Ohm, with T_s representing the current semiconductor sensing device absolute temperature, R_s the current resistance and $R_{s,0}$ the device resistance at temperature $T_{s,0} = 25 \text{ } ^\circ \text{C}$. Solving for R_s results in:

$$R_s = R_{s,0} e^{-B(\frac{1}{T_{s,0}} - \frac{1}{T_s})} \quad (7.2)$$

Knowing the characteristics of the sensor, represented by B , $R_{s,0}$ and $T_{s,0}$, and the parameters of the sensor excitation circuit, represented by $V_{s,aux}$ and $R_{s,aux}$, T_s may be estimated analytically in the BMS from a simple measure of the NTC thermistor voltage v_s .

Fuses and Battery Disconnect Unit Model The BMS is required to control the initial connection of the battery pack to a capacitive load such as a motor controller. A direct connection would lead to an in-rush current only limited by internal battery impedance. For instance, cells generally provide low internal impedance while the connected motor controller has a large capacitance, this forced in-rush current

may exceed couple of hundreds amps [9]. In high-voltage LIB packs, used e.g. in e-mobility applications, fuses and relays are commonly integrated in a so called battery disconnect unit (BDU).

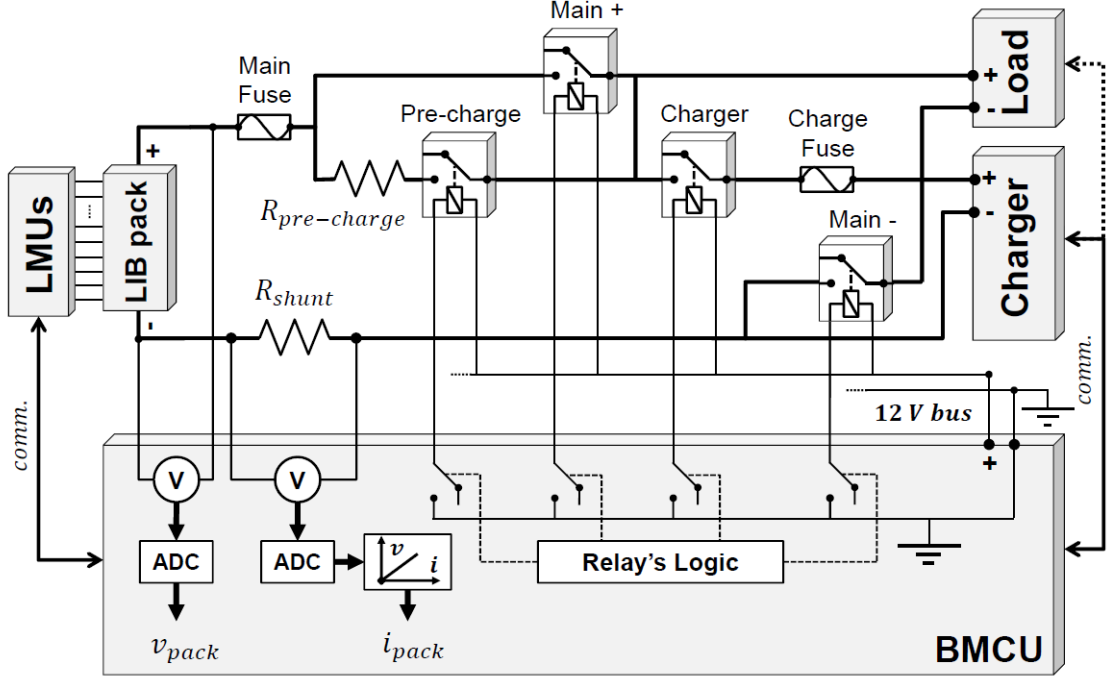


Figure 7.3: Schematic implementation of battery disconnect unit (BDU) topology and its interfaces to the battery system model.[460][461][462]

Moreover, the system requires a pre-charge circuit, for energizing the DC-link capacitor; a current measurement device, for instance a shunt resistor or a Hall Effect sensor; and external communication with a BMS, which may control the relays and measure the pack voltage and current. The design of the BDU, particularly the device selection and location, may differ widely depending on the specific characteristics of the LIB system and its application. For illustrative purposes, a most common topology of a BDU and its interfaces is implemented and shown in Fig.7.3. This BDU is made up of four relays and two fuses, a pre-charge resistor $R_{pre-charge}$ and a shunt resistor R_{shunt} . High currents outside the hardware and cell operation specifications have to be prevented. A common method uses passive over-current protection to break the circuit, e.g. fuses. These passive elements cannot be controlled by the BMS, however, the detection of a blown fuse belongs to the set of safety functions of an intelligent battery management system [9] and are implemented in the BDU model. The relays allow the system to operate in three modes: stand-by or idle, discharge and charge. Furthermore, it should be noted that, since the DC-link may be connected to a capacitive load at the load or the charger side, i.e. a DC-link capacitor, inrush currents have to be limited whenever the operating mode changes from stand-by to charge or discharge. For this purpose a pre-charge circuit, which consists of a pre-charge resistor in series with a pre-charge relay, is implemented. The BMS is responsible for all the relays logic and controllers. In the

HIL simulator presented, both the relays control signals and LIB pack voltage v_{pack} signals are transmitted from/to the BMS via CAN communication. Fuses are also considered. Regarding the relays, they are represented as ideal switches in series with a static resistor. The pre-charge resistor, $R_{\text{pre-charge}}$ Fig.7.3, is rated according to a desired pre-charging time of 1 s, taking into account the system nominal voltage, the DC link capacity and the power dissipation. On the other hand, the BMS is configured assuming that a shunt resistor R_{shunt} is used for pack current i_{pack} measurement using analog circuitry.

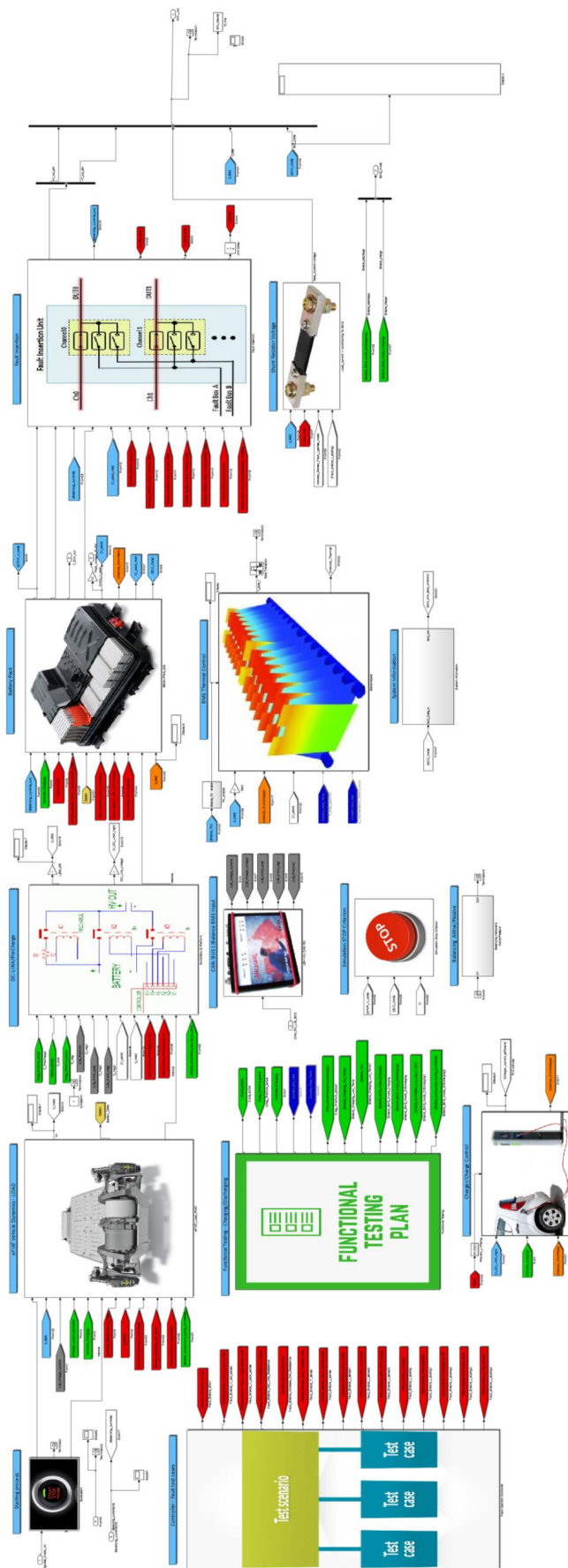
Isolation monitoring As mentioned earlier, HV systems in hybrid and electrical vehicles require versions for IT distribution networks, meaning the chassis of the components are grounded within the HV system, which is isolated from the vehicle chassis ground. The BMS has the assignment to support isolation- or ground fault detection evaluating the ground insulation resistance of the positive and negative DC bus [463]. According to the National Standard, 18384.3-2001, the battery insulation resistance is the relative resistance to maximum leakage current when the fault occurs somewhere between the two battery terminals and earth ground [463]. The DC current insulation detection measures the insulation resistance passively or actively. The passive system applies a isolation resistance between positive or negative pole vs. earth ground and the voltage across the resistor is calculated which is repeated for two measurements. For a continuous isolation supervision of this measurements are repeated constantly. The active system uses multiple sampling resistances and electronics to generate the DC signals via the transformer.

Charger For the BUT, the BMS is responsible for controlling the battery charger. Therefore, the battery charger is implemented into the system simulation model. The BUT uses charger-centric implementation, hence, the BMS enables the charger which is responsible for successful charging. BUT is responsible for the rate of charge to optimize battery life. Different charging methods are implemented, such as CC/CV or target voltage. The optimized PI controller controls the PWM signal to the MOSFETs.

High-voltage interlock status The function of the high-voltage interlock loop (HVIL) detects critical situations and will open switches to de-energize the HV circuit to prevent contact with hazardous electrical potentials. The BMS monitors the state of the HVIL at selected access points and takes action immediately the evaluational electronics detects the conductive path of the signal loop carrying the reference signal is interrupted. In the HV system of hybrid and electric vehicles exist active and passive components. Active systems are equipped with an signal detector, control electronics and communication interface and connected as all components of the HV system to the central control unit, such as the battery management system. Passive systems are only integrated into the loop in terms of connectors to the chassis or electrical contacts. The detector has to be adjusted to the reference signal of the generator and must not modify the signal. All detectors within the HVIL have to process the reference signal in order to make a precise decision weather the HV has to be opened or not.

Thermal Management Thermal Management can be seen as a support function to store, receive and provide electrical energy, for which thermal states are defined encapsulated into the different vehicle states. The thermal states must consider protection of maximum/minimum cell temperature and therefore follow the set target temperature to keep the battery within its optimal life span. The BUT takes over the thermal management control function, hence, an active (water-cooled) thermal system was implemented. Necessary inputs from the thermal cell models include the temperatures as well as sensing of coolant temperature in-/outlet, inputs from the SoX functions. A closed-loop system is required, which feeds back the signals of the pump and valves. The thermal manager requests soft/hard charge/discharge power and energy limits, while it needs feedback from the SoX estimators in terms of min/max cell temperatures. Hence, the controller requests/allows thermal energy or power from the traction battery and vice versa or minimum pump flow rate. All these functions require diagnostic checks. The SoX algorithms therefore have to estimate cooling or heating power, how much heat is generated resulting in change of cell temperatures, as well as how much energy is necessary for cooling/heating the system.

The overall HiL battery system simulation framework is provided in Fig.7.2, including the starting process, vehicle dynamics for dynamic driving profile generation, battery disconnect unit including fuses, battery pack, FIU, thermal system, high-voltage and current measurements, functional test plan (requirement based testing), charger and control, and simulation visualization.



- 1.) **Starting process:** KL 15,30,31, Watchdog.
- 2.) **Vehicle dynamics:** incl. environmental parameters, road profile, dynamic driving profile.
- 3.) **DC-link / Pre-charge:** Switch box with pre-charge control (Main+/- & pre-charge relay).
- 4.) **Battery pack:** 32 cells can be connected in series/parallel, electric-thermal model with individual aging factor.
- 5.) **Fault insertion:** Virtual faults to the systems (sensors, controller,...) are insert to simulate failures.
- 6.) **Thermal system:** active and passive cooling/heating system
- 7.) **Shunt/resistor voltage:** load current is conditioned for BMS HIL and converted to voltage.
- 8.) **Active/passive balancing mode**
- 9.) **CAN-bus read/write:** Signals are send/received to/from BMS DUT
- 10.) **Functional testing:** system is tested against functional requirements.
- 11.) **Fault test cases:** Controller for fault test cases.
- 12.) **Charger & control:** allows different charging profiles
- 13.) **STOP simulation**
- 14.) **System information:** Displays all relevant system parameters

Figure 7.2: Battery system model including all components.

7.3 Commercial BMSs Under Test

Besides other BMS under test, here we focus on a commercial BMS, sBMS v6 provided by Lithium Balance A/S. The BMS uses a master-slave configuration, where each slave unit can be configured to supervise 8 cells and 2 temperatures. The slave is capable of passive balancing with up to 1 A simultaneously on all 8 cells and will automatically reduce the balancing power if the heat cannot be dissipated at high enough rates. 4 slave units are used in the experiments presented in this paper. However, a total of 32 slave units can be used for each master, giving a total of 256 supervised cells and 64 temperature inputs. The master is capable of measuring LIB pack current, LIB pack voltage, isolation faults, and contactor faults. Communication to external systems is provided through a configurable CAN bus interface, where any parameter can be broadcasted. The control of contactors, switches and fans are all implemented in the BMS and can have a multitude of configurations.

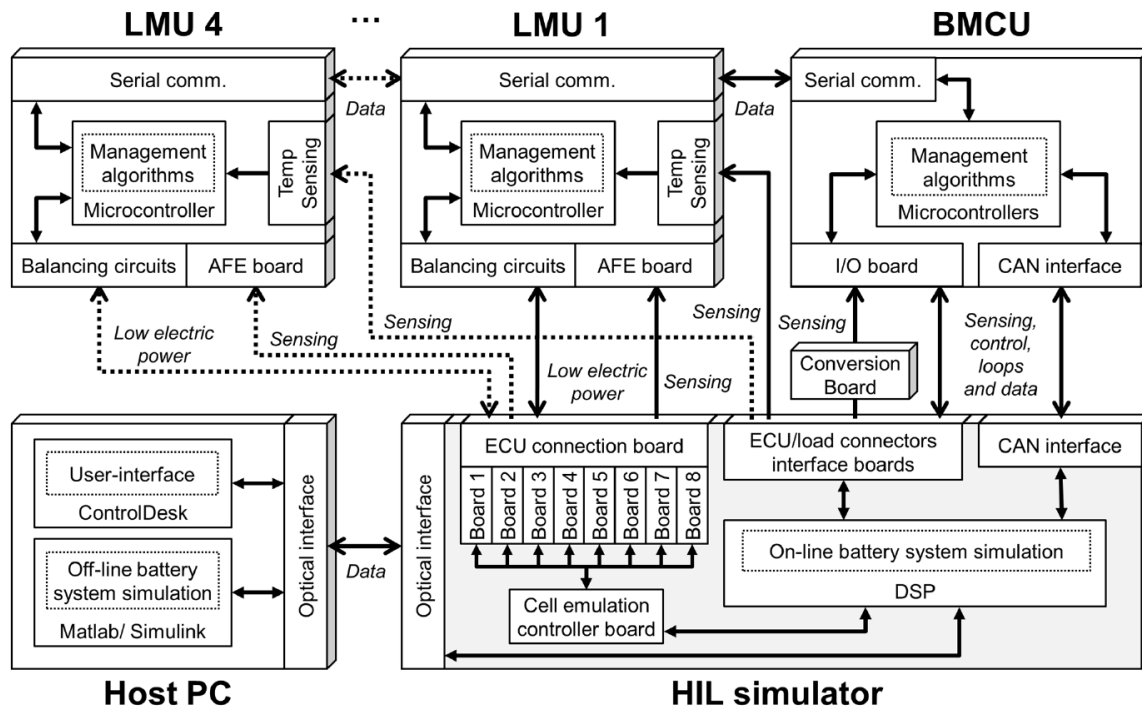


Figure 7.4: Overview of main functions and interfaces between the host PC, the HIL simulator and the BMS. [460][461][462]

7.4 Functional and Non-functional Target Testing

Failure is an option here. If things are not failing, you are not innovating enough.

– Elon Musk

To guarantee functional safety in a battery pack for hybrid and electrical vehicles, hazardous failures have to be identified during design process [464]. Since software has an influence on safety, car manufactures invest very heavily, in fact 75% of software development costs accounts to testing. With increasing system complexity it must be assumed that testing costs continue to increase even further. It would appear that by extending the testing phase the systems become less error-prone. However, to improve safety-critical systems, special testing methods have to apply corner cases that are normally not triggered during normal operation mode. Standards like ISO 26262 specify the required methods for validation of safety-critical systems including further demands on fault injection testing. Fault injection defines the process of dependability validation applying controlled experiments, where the systems responds to an explicitly induced fault, which then is observed and analyzed to determine the capability of complying with predefined specifications and used to measure the fault tolerance. According to its system application the techniques

inserts the fault to the specific domain and can be characterized as hardware or software fault insertion. This technique states back to the 1970s and essentially was developed to include errors at the hardware level. In this work the method is reviewed and adapted to for battery management system testing using innovative hardware-in-the-loop test bench. The fault injection techniques can be further grouped into [465]

- **Execution-based fault insertion:** The system itself is put to the test through fault insertion while its behavior is observed to measure the fault tolerance.
- **Simulation-based fault insertion:** A complex-varied plant model is developed and artificial faults are introduced. During the simulation the effects of the faults on the system stability and operation are examined.

Lastly, invasive injection methods introduce a latency which may lead to execution violation in real-time, time dependable systems. On the other side, non-invasive injection methods don't leave any footprint on the system other than the injected fault itself. Fault injection can be performed on a virtual environment (simulations and models) or prototypes in the early development stage. Therefore, this concept can speed up the understanding of faults on the target system behavior to enable suitable measures to reduce the number of faults for the final system. The idea is to test the system fault tolerance against known faults rather than randomly induce and debug the design faults.

7.4.1 Fault insertion environment

The implemented fault injection environment is illustrated in Figure 7.5. The system consists of following components:

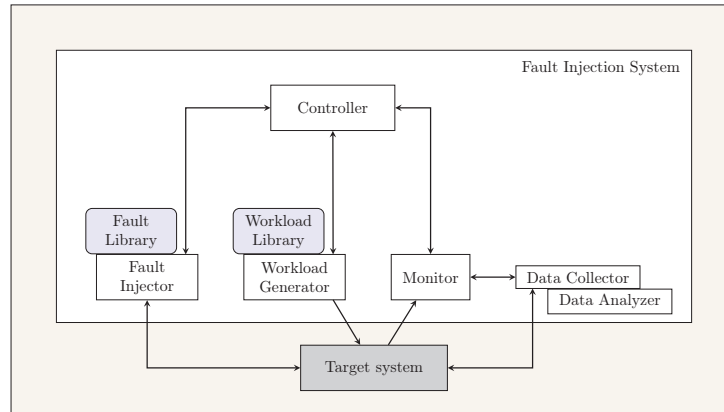


Figure 7.5: Components of fault injection environment. (modified from [465])

- Fault injector induces the defined faults into the target system during execution time.
- Fault library consists of different fault types, fault locations, and times.
- Profile generator generates the load profile for the target system as input based on the load profile library.
- Supervising unit controls the experiment and runs on a separate computer.
- Post-processing unit performs data processing, by collecting and analyzing.

The method of fault injection was first used in the 70's to induce faults on system hardware. This type is therefore called hardware based injection method to simulate hardware failures. It is performed at the physical level by modifying the value of the ECU pins [459] or causing electromagnetic interference. On the other side, unpredictable system response may introduce damage or being life-threatening. For distributed or high-level of integration system the portability cannot be always guaranteed as well as limited observability and controllability of I/O pins. This technique still involves a lot of effort and expense. Nowadays, software faults are the most frequently responsible for errors and consequently and therefore often emulated by software-based methods. The goal of software-based fault injection is to reproduce errors without physical intervention on the hardware. However, it is only possible to induce faults on those level of the system which are accessible by the software [459]. To cause the error, the software binary is modified by changing the source code with the possibility of different system behavior accessing the program states, communication and interactions [465]. The advantages of this technique is that it can be run in real-time allowing multiply error injection in parallel including any design fault available for the present hardware and software system. Software-based fault injection decrease testing and validation costs, the requirements for special-purpose hardware is not required. Simulation-based fault injection involves the design of a plant dynamic simulation model which is able to mimic the physical system under analysis. The failures are injected directly to the model, a useful tool for evaluating the systems fault-tolerant mechanism. This method provides a quick feedback to the system designers at the downside of complex model development, and its impossibility to cover all the possible faults. Hybrid-based methods apply a combination of the earlier described techniques. Often it is useful to link simulation-based fault injection with one of the others, since its controllability and observability increases.

7.4.2 The Threats: Faults, Errors, and Failures

The BMS of a hybrid and electrical vehicle must guarantee safe operation and therefore dependability is of central importance. However, the system may get into abnormal condition, characterized as an error, e.g. a subsequent failure due to overcharging. Not

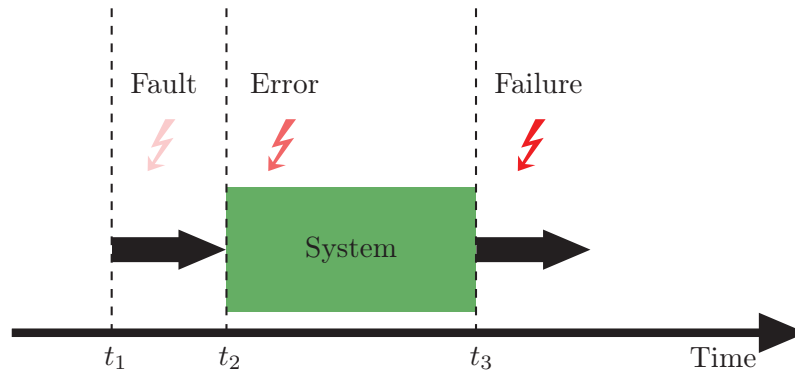


Figure 7.6: Connection between fault, error and failure [465]

only the direct safety measures of individual cells have to be taken into account, but also the safety of the entire system. As a consequence, characteristic threats have to be identified and considered in functional safety requirements of a BMS and battery pack system [466].

- **Fault** defines a condition which was in a measure not to be expected before and can negatively influence the capability to perform the required task. The systems failure is initiated by fault if activated, otherwise it is dormant.
- **Error** defines the contradiction between a measured, observed and computed signal value or state and the actual physical, specified or theoretical-numerical value or state and may or may not cause failures.
- **Failure** is a state, in which electrical, electronic and programmable system is unable to execute the required function or service. Thereby, a fault in the systems higher-layer can be insert by a failure in a sub-system.

Ideally, in case of failure, the battery pack with BMS is designed in such a way that a hazard to the system is avoided. But should a fault arise, besides detecting a failure, the system should continue to operate as fail-silent, fail-safe or fail-degraded. A system is considered fail-silent, when the output interfaces are terminated during failure. When a system supports fail-safe, it has the ability to fall back to a known safe state after fault detection.

For complex battery storage systems in different demanding applications it is nearly impossible to stay in a dependable state, meaning, dependability is defined as the trustworthiness of a electronic system executing the correct task at the right time such that total justification and trust can be placed on the expected service it delivers. Therefore, a higher degree of dependability has to be considered, ensuring security for individuals even if other parts of the system generate errors. Hence, further sub-categories have to be

defined, fail-degraded, where the system maintains within a degraded operation for software or hardware faults and fail-operational, thus it is possible for the system to continue its normal tasks and service despite the fact of active software or hardware faults.

The change to a defective state has to be detected by the BMS, when things go wrong the system does fault handling and recovers to a known safe state and should run a fail-operational, fail-safe or fail-degraded operational behavior in means to attain dependability [465].

7.4.3 Fault Classes

The cost and consequences of BMS failing can range from minor problems which are uncritical or even unnoticeable for the user to catastrophic, with serious injury occurring or lives lost. The BMS assumes more requirements to provide the most functionality to the battery storage system, therefore it becomes more significant to the overall systems safety. Faults are diverse in their origin, therefore, before carrying out extensive tests for fault insertion, it is necessary to identify and categorize generic sources of faults in a battery storage system. An appropriate battery storage system including a battery management can only be developed with knowledge of all kinds of faults in each individual component. It is therefore necessary to specify potential faults by analyzing the complete system. The result is classified in six major criteria [466].

- **System faults** have crept in during design process and are integrated in the system at the initial operation.
- **Run time faults** occur while the system is operating and may be caused by wearing faults or operating faults.
- **Other faults** define the last class and include production faults or maintenance faults.

Faults are further categorized into

- Faults in battery storage hardware (cell or pack level).
- Faults in BMS software (high or low level).
- Human-made faults (e.g. user operation).

The ultimate category in fault classes can further specified according to its persistence:

- **Permanent faults:** steady, persist after its occurrence and can only be recovered by means of fault handling methods.

- **Transient faults:** occurs intermittently but disappears again after a some time.

7.5 Battery Management System Testing Using Fault Insertion Techniques

In automotive industry, a formal process is needed that can help to move from design to implementation flawlessly, with always the possibility of testing and validating in a very rigorous approach. The industrial standard for SW development process (also applicable to embedded SW and HW) is the well-known V-model. By using a V-model, each phase of the development process during the project definition is related with its corresponding phase during test and integration. Following the V-model, during the development of a Battery Management System (BMS), appropriate testing is required at different phases. The developed on-line battery system model for HIL simulation takes into account electrical, thermal and aging aspects from cell to pack level. Furthermore, a HIL battery simulator may provide new testing features unfeasible with real batteries. For instance, using a HIL battery simulator it would be possible to provide on-line tracking of battery internal states. Turning to the model of SW development process, although the V-model it is broadly used and easy to understand and manage, it may be too simple to actually reflect the complexity of the BMS product development process. This is particularly true when a HIL battery simulator is used for high-level testing. The aim is to introduce the principles of BMS testing and to present a practical approach to develop ad hoc SW and strategies for BMS testing on a HIL battery simulator at system-level.

7.5.1 Background to BMS Testing

To understand the relevance of this discussion, it should be noted that there is not even a formal definition of testing and its goals. In fact, generic, qualitative or empirical approaches may be applied. From a generic perspective, the goal of the tests may be to ensure that the BMS does what it is supposed to do and that it does not do anything unintended. The challenges of BMS testing is illustrated in Fig.7.7. At the same time, according to ISO9000 quality standards, testing is required to provide Quality Assurance, which is defined as “part of quality management focused on providing confidence that quality requirements will be fulfilled”. The design of the tests focuses on critical aspects of the system with the aim of identifying errors. That said, it is easy to understand that different testing strategies may be followed. As explained in the following, these strategies may be classified into different ways, according to the testing level, method or process (Table 7.1).

Testing levels Tests may be classified according to their granularity level in three groups:

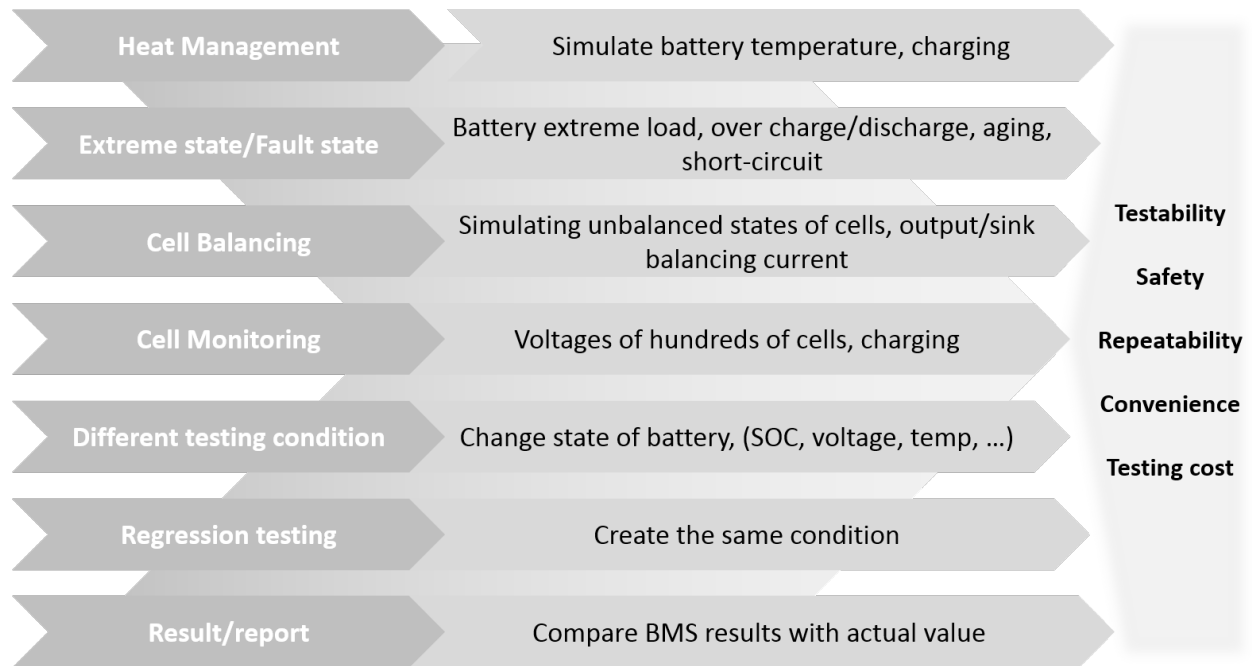


Figure 7.7: Challenges of BMS testing. Source: Shanghai Keliang Information Tech. and Eng. Co., Ltd.

Criteria	Types
Device under test	SW, embedded SW or HW
Development stage	Design, development, prototype, pre-release or release
Device state	Static (OFF) or dynamic (ON)
Box approach	Black-, white- or grey-box
Testing process	Functional or non-functional
Level	Unit or module (group of units), integration or system
Environment	(Near to) real, simulated or emulated (HIL or Power HIL)

Table 7.1: Classification of BMS Tests. [460][461][462]

module (or unit), integration or system testing. These categories correspond to tests at component, interface or whole-system level, respectively. When a BMS product is under development, module and integration testing may be more common. On the other hand, when a new BMS product is released or an existing BMS product is under revision, tests at whole-system level must be preferred. It is also important to realize that, if any changes are made to the product at low-level, e.g. at module or unit level, re-testing at integration or system level may be required. These tests are often referred as regression tests. Hence, not only test designing but also test planning is fundamental to reduce time and cost requirements of SW and HW testing.

Testing Methods Test are categorized according to the state of the SW or HW under test, when system is in off state, its static, when system is in on state, the tests are

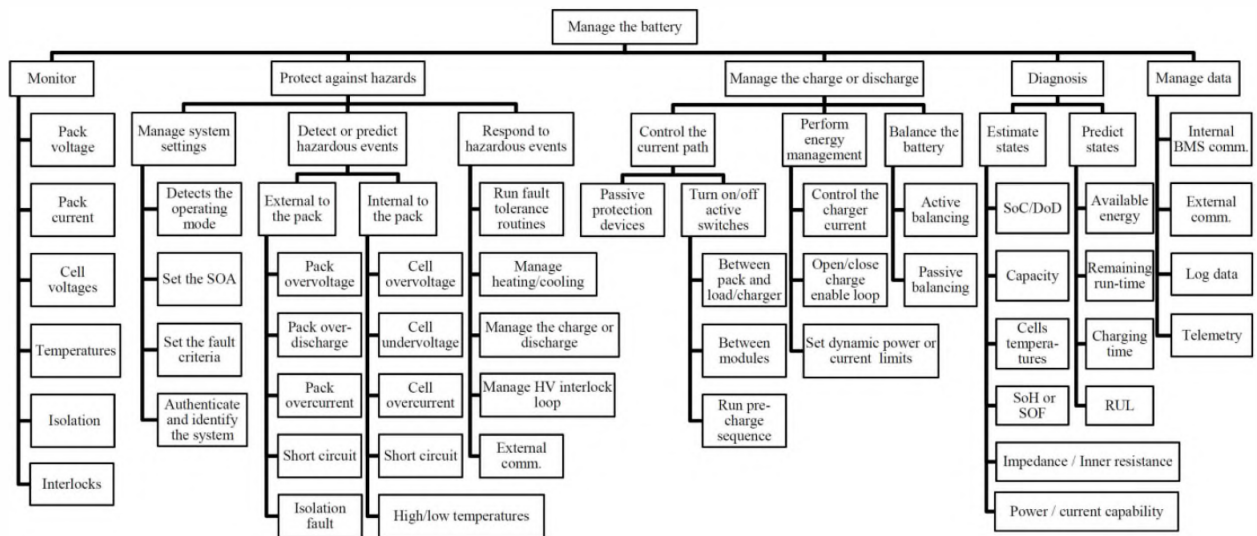


Figure 7.8: Functional tree of a state-of-the-art BMS for large Li-ion battery packs. [460][461][462]

referred as dynamic. Another typical way to classify the tests it is the so called box approach. White-box testing is based on the analysis of internal structure of the piece of SW and/or HW under test, while black-box testing is based on the analysis of its performance against certain requirements, the combination of both is so called grey-box testing. In the case of SW, white-box testing requires full-knowledge of and full-access to the source code. Regarding black-box testing, by definition, does not deal with logical internal structures. However, the knowledge about the system logical internal structures may ease the design of the test process, since it may help to identify the critical aspects of the system with the aim of identifying errors. Testing methods may also be categorized as destructive and non-destructive, whether or not damage is caused to the device under test.

Testing Processes With regard to testing processes, they may be classified into functional and non-functional. Functional testing may be defined as a process of testing a piece of SW and/or HW for its functional requirements (FRs) in order to provide QA or to seek for errors. Similarly, non-functional testing may be defined as a process of testing for its non-functional requirements (NFRs). According to systems engineering theory, FRs may define specific behavior or functions of a system, while NFRs may define specific criteria used to analyze the operation of a system. In a broad sense, FRs may define what a certain system should do and NFRs how a certain system should it and how a certain system should be. During functional testing, all the low-level functions of a BMS may be tested, either individually or globally, based in a black box approach. Additionally, each of these low-level functions may be related to a particular SW and/or HW of the BMS, which, in turn, may be related to a certain electric or electronic HW interface between the BMS and the rest of the system. A common problem during the development process is the poor description of the system requirements from the very beginning. NFRs

No.	Test Case	Description	End-of-test criteria
1	Idle or Stand-by	System settings are managed. The fault criteria are set in the BMS. Then next functions are checked in steady-state: data management, monitoring, detection of hazards or response to hazards	All functions tested in steady-state
2	Current, voltage, temperature sensing	All sensors are calibrated, if possible. Then full-range accuracy test are run keeping the BMS at different conditions, e.g. in a temperature chamber	All conditions tested
3	Dynamic discharge	Using a real dynamic discharge profile from an EV, the pack is fully discharged at room temperature	BMS stops the discharge to avoid over-discharge
4	Overvoltage during regenerative braking	Using a real dynamic discharge profile from an EV, the pack is fully discharged when a very high regenerative current is experienced	The BMS interrupts the regenerative charging current
5	Overtemperature during discharge	Using a real dynamic discharge profile from an EV, the pack is discharged at high temperatures	Pack reaches max. temp. and BMS stops the discharge
6	Short circuit	Shorts at different places in the pack. Case 1: internal or external short close to cell's tabs. Case 2: external short through fuses or shunt resistor. Case 3: External short through fuse and switch box (relays or fuses)	Short circuit current is zero
7	CCCV Charge	Conventional CCCV charge with active/passive balancing	End of charge
8	Charge test at low temperatures	Charge is enabled, but pack temperature is below limit for charging. Temperature increases progressively due to heating system.	Pack temperatures over threshold, charging starts
9	Diagnosis	Case 1. Emulated SoC vs BMS estimated SoC during real dynamic discharge profile from an EV. Case n. New cases according to the BMS diagnosis features	End of charge or discharge
10	Isolation Monitor	Single isolation fault on positive/negative terminal of a floating voltage pack	Isolation fault detected
11	Global power consumption	Case 1. Full discharge, considering a dynamic discharge profile from an EV, followed by CCCV charge. Case 2. Test on stand-by for a certain period of time	BMS power consumption is measured at all conditions

Figure 7.9: Test cases proposed to evaluate a BMS for its FRs and NFRs. [460][461][462]

for a BMS product may include a large list of combined HW and SW related requirements, for instance: EMI compliance, physical dimensions, physical properties, efficiency, safety, reliability, robustness, recoverability, integrity, compatibility, performance (response times, processing times...), capacity (throughput, storage...), scalability, security, documentation, etc..

7.5.2 Strategy on HiL BMS Testing

The proposed testing strategy can be classified as dynamic, black-box, non-destructive, functional and non-functional. In the following, the approach used to identify the FRs and NFRs to be evaluated the corresponding list of test cases is presented.

FR and NFR The definition of FRs deals with the identification of functions that a BMS may be required or expected to perform. The functional tree presented in Fig.7.8 is used for this purpose, facilitating the understanding of the system dependencies. With regard to the NFRs, it may be difficult to define all the NFRs in an universal way, since many of them would be user or application-defined. Moreover, it may be impossible to check them all due to time and space constraints. Hence, only certain generic and relevant NFRs are proposed to be included in the practical approach presented in this work, the overall test cases are summarized in the

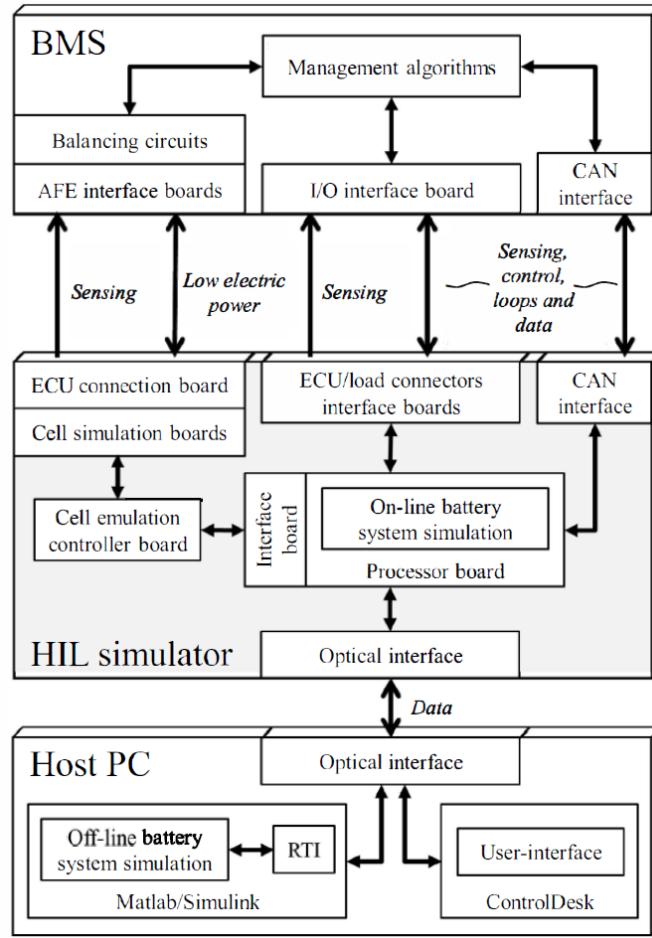


Figure 7.10: Experimental setup for HIL simulation: main functions and interfaces between host PC, HIL simulator and DUT. [460][461][462]

Appendix. Again, empirical evidence plays an important role in the identification of the most relevant features to be tested. Finally, three groups of NFRs related with power consumption, reliability and stress testing are included. With regard to reliability, NFRs related with temperature, current and voltage monitoring at different conditions are considered relevant be evaluated. On the other hand, with regard to stress testing, a systematic and extensive evaluation of Fault Insertion (FI) modes in the sensors, the actuators and the interface between a BMS and a battery is proposed. Accuracy and fault insertion testing are useful to evaluate the reliability and robustness of a BMS, key parameters for the industry nowadays.

Design of Test Cases Based on empirical evidence, a set of test cases is designed in order to check the aforementioned FRs and NFRs of the BMS using the HIL battery simulator, either individually or globally. The definition of the BMS's FRs and NFRs may be affected by the specific characteristics of the battery pack and its application, which may in turn result on a modification or extension of the tests cases proposed in this manuscript. All the test cases related with the evaluation

of the BMS's FRs and NFRs, excluding the stress testing, are listed and described briefly in Fig.7.9. Regarding stress testing, Table 7.2 lists the FI modes to be evaluated and their possible location. This include test cases that consider HW-based fault insertion in the HW interface, due to e.g. open circuit, short circuit, noise or increase of line resistance; or malfunction of sensors, actuators or any other system-level components, e.g. NTC thermistors or switch box relays. It should be noted that there may be many different interfaces between a BMS and a battery (or the HIL battery simulator), e.g. temperature, current and voltage sensing lines, communication data links, Analog or Digital I/Os, etc., which is beyond the test scope in this work (destructive, not of interest, to long time...).

FI modes	Location
Open circuit	Interface(s)
Short circuit	
Noise	
Increase of line resistance	
Malfunction	Sensor(s), actuator(s) or other system-level component(s)

Table 7.2: Summary of FI modes and their location.

Finally, it should be noted that this work focuses only on non-destructive FI modes off-board the BMS under test. FI modes on-board the BMS are typically part of tests at lower levels of integration.

7.5.3 Experimental Setup and Results

The commercial BMS product chosen for sample testing is the sBMS v6 provided by Lithium Balance A/S. Regarding the SoX functionality, a simple coulomb counting-based algorithm is provided for SoC estimation. Regarding the SW implementation, it has already been sketched in Fig. 3. The HIL simulation battery model is implemented off-line in MATLAB/Simulink. Then the real-time on-line features of the HIL simulator setup are linked with the off-line simulation model using Real Time Interface (RTI) SW from dSPACE, which enables automatic generation of C code for the HIL simulator from MATLAB. ControlDesk, another SW provided by dSPACE, is used to download the compiled code to the HIL simulator processor board. In addition, this SW enables the development of an ad hoc user-interface to control, to monitor and log and automate the experiments from the PC. Experimental results of the interaction of the BMS with the system-level HIL simulated battery are shown in the following. Fig.7.11 is just a snapshot of ad hoc SW for management of BMS settings visualizing the state of the system.

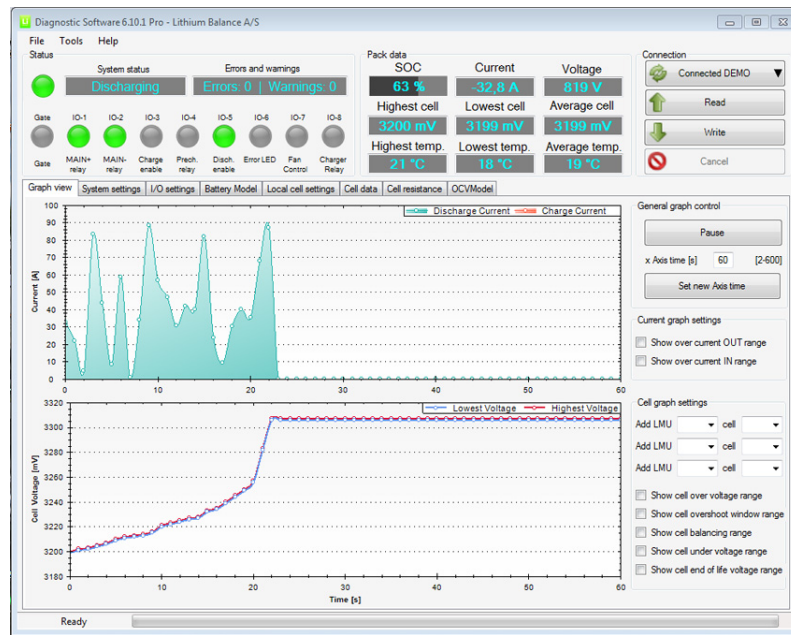
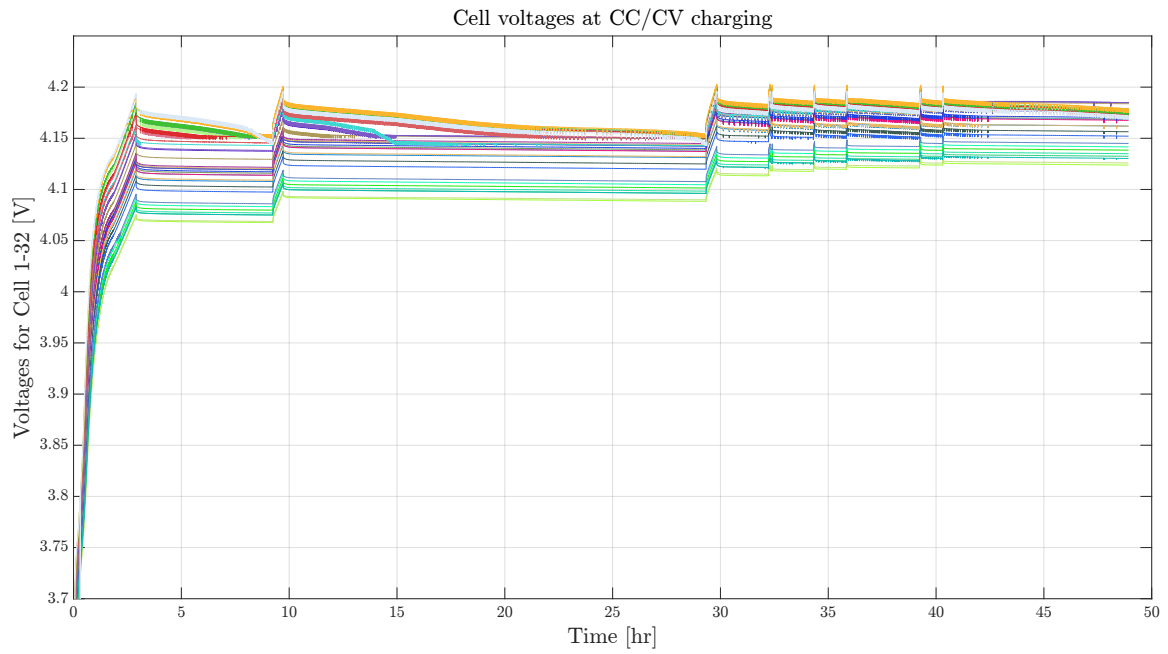


Figure 7.11: Snapshot of ad hoc SW for management of BMS settings provided by Lithium Balance.

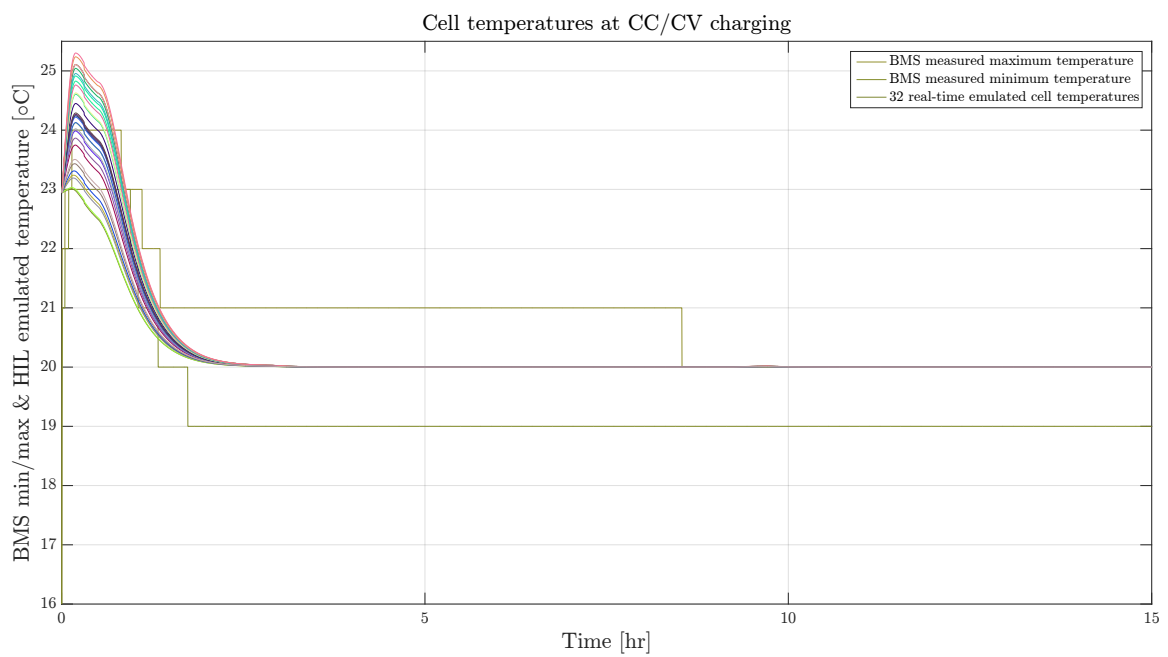
7.5.3.1 Functional Testing

CCCV Charging

The CCCV charging test results illustrate the dynamical behavior of the system during CC/CV charging. The initial pack state is SoC=0% but different SoH states, meaning, the pack is grossly unbalanced. It can be noted, that the total test case takes 50hrs until the pack is balanced (depending on user input of requirements). The results below illustrate the temperature distribution during charging. During charging state, the charger puts 40A output, which leads to a temperature rise, during CV/balancing state, the temperature reduces again. The DC-link (system) voltage for 32 cells connected in series is also illustrated below while the charging current is controlled by BUT. The min/max calculated SoC with initial 10% of SoC deviation before the charging event is applied, the BUT was able to reduce the imbalance up to 5% SoC. The test was stopped as there was no interest to wait for complete balanced system. Depending on the calibration of the BMS, balancing speed and threshold can be adjusted. The last image of this test illustrates the individual cell balancing currents and activation. Cells from 1-15 are new, cells from 16-32 are aged and hence need to be balanced more often as the cut-off voltage is reached earlier.

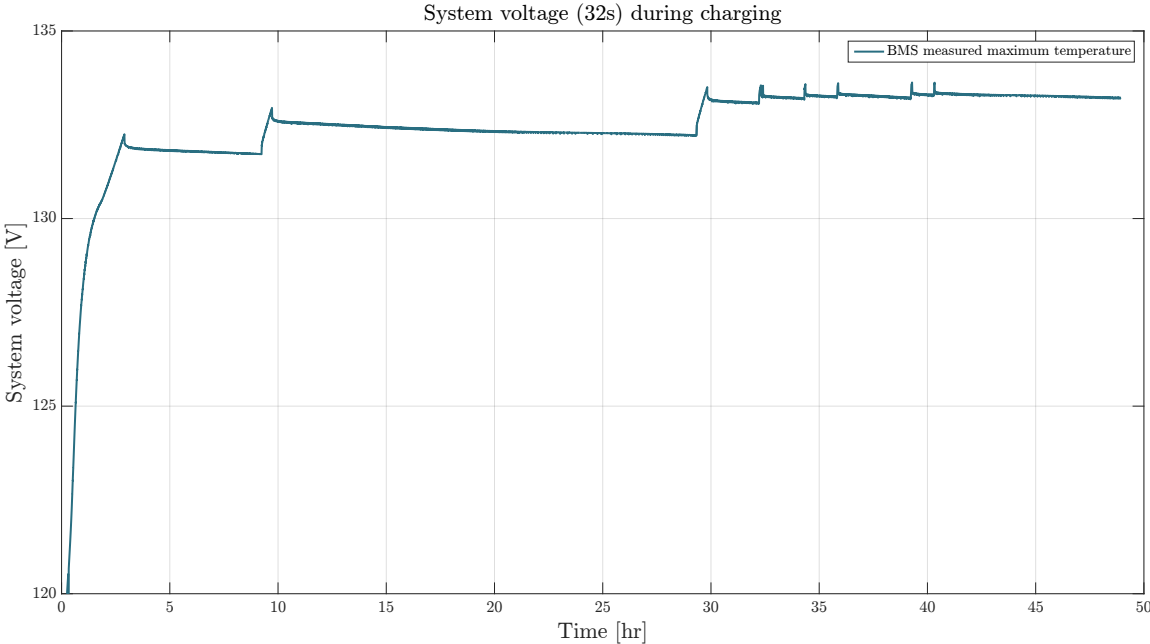


(a) Cell voltages during CCCV charging

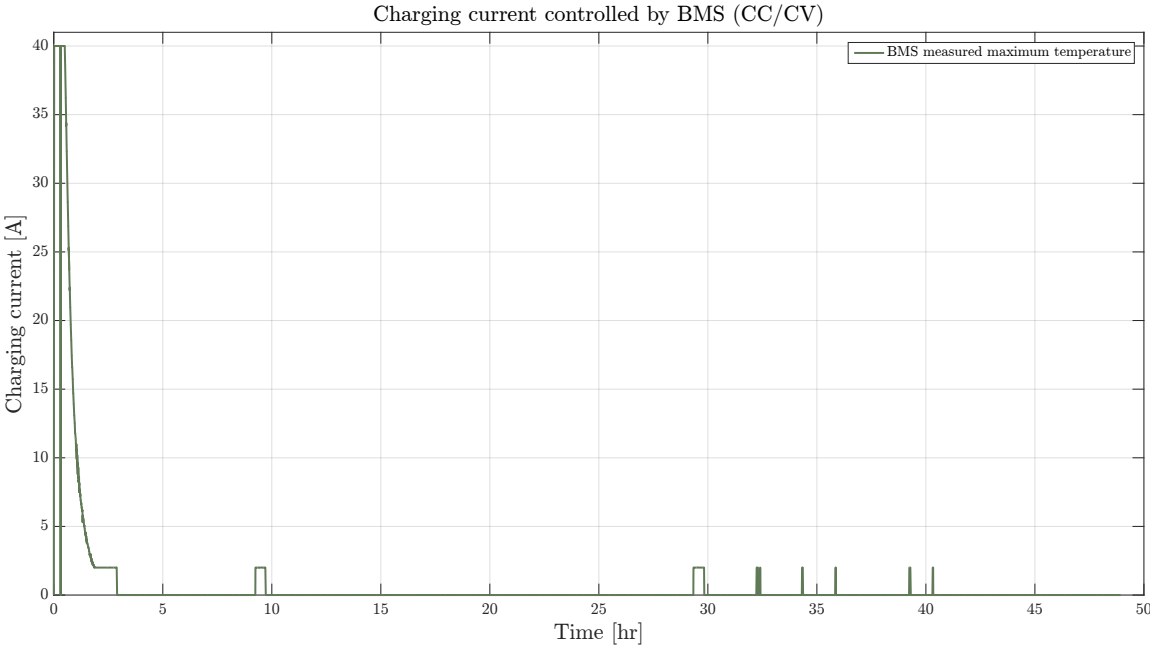


(b) Cell temperatures during CCCV charging

Figure 7.12: BMS functional testing: CCCV charging

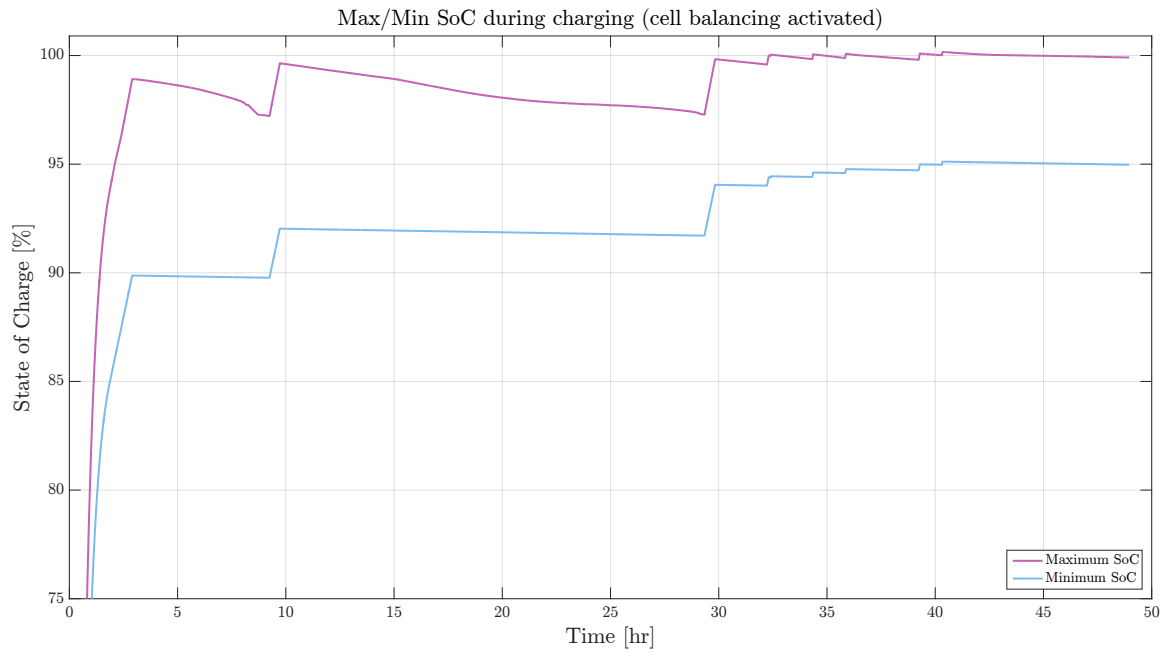


(a) System voltage (32s) during charging

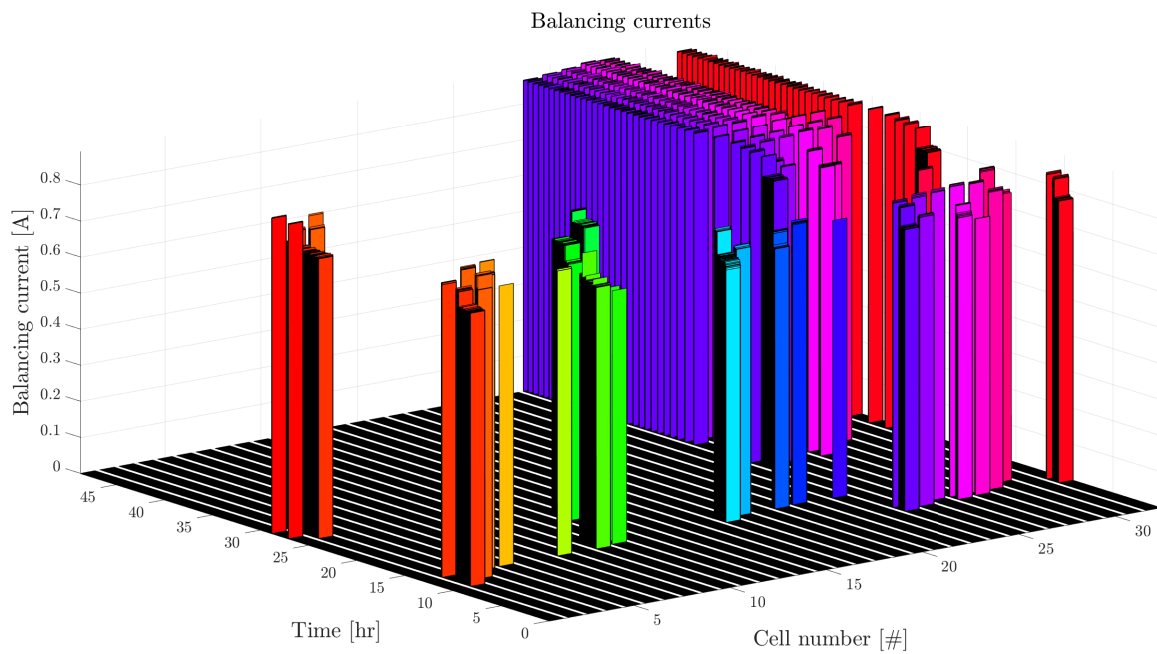


(b) Charging current controlled by LiB BMS

Figure 7.13: BMS functional testing: CCCV charging



(a) Min/max SoC during charging

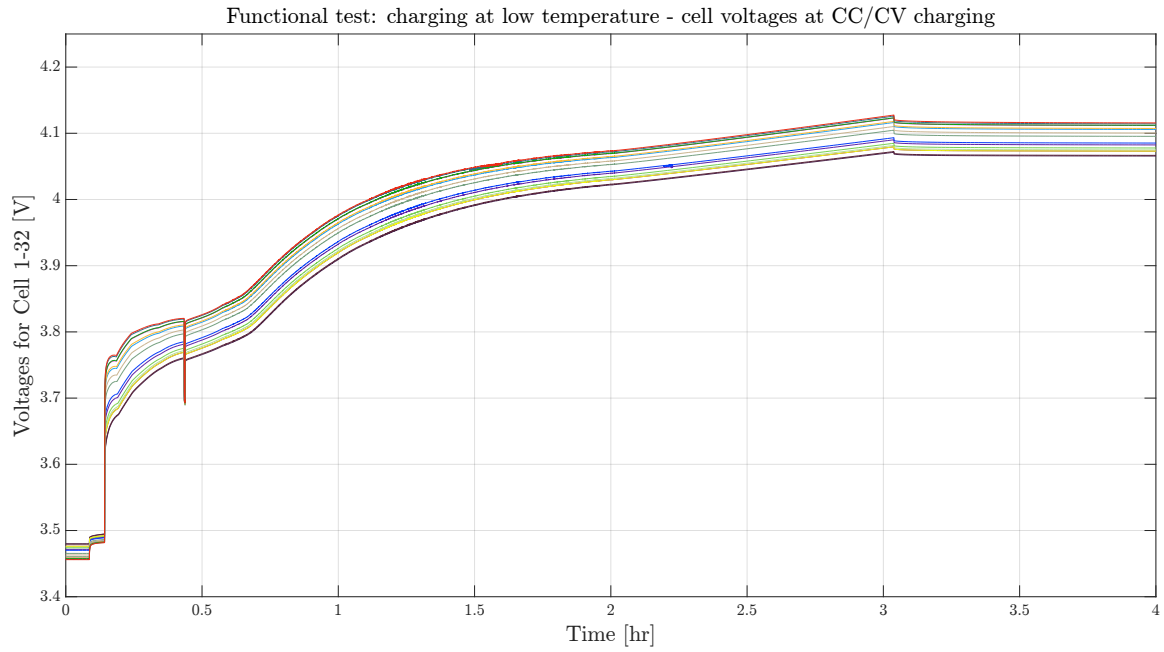


(b) Balancing currents during charging

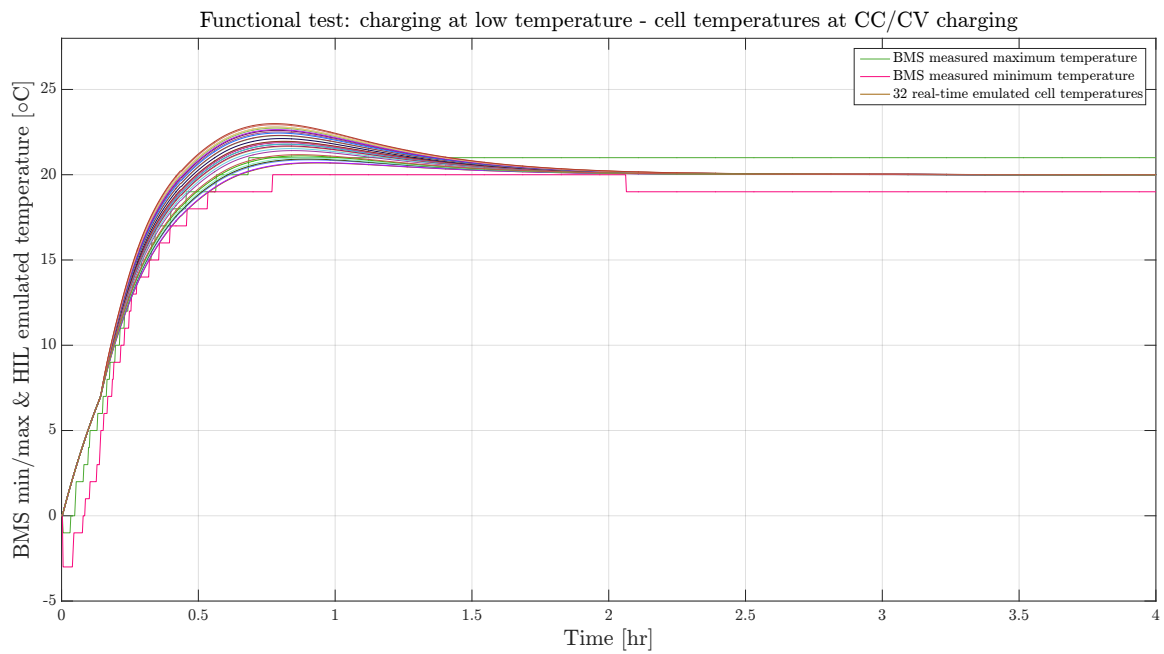
Figure 7.14: BMS functional testing: CCCV charging

Low Temperature Charging

This functional test shall illustrate the behavior of the BUT of charging the battery at low temperatures in Fig.7.16. As known, low temperature charging will cause lithium plating which may lead to severe faults. In this case, the battery pack starts at a temperature of 0°C, in which the BUT controls the charging current to 3A, resulting in an temperature increase. As soon as the temperature exceeds the threshold (here 6°C), the BUT steps up the charging current to 40A. The charging current step at 0.5hr is for diagnostic reasons to determine the internal resistance of the cells.

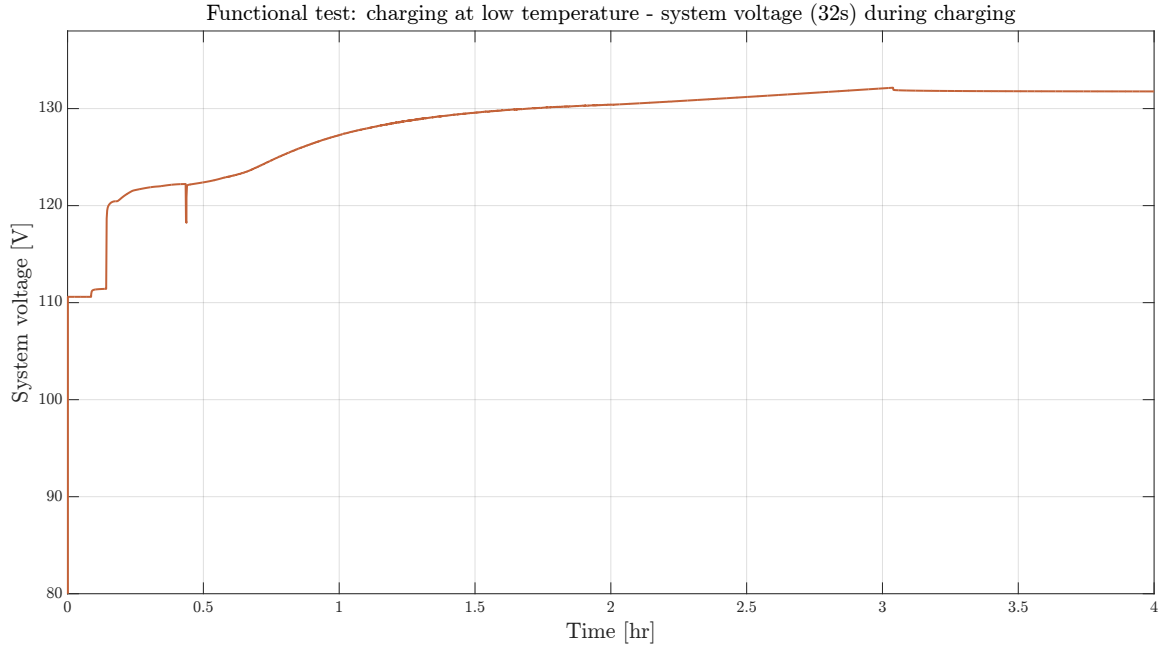


(a) Cell voltages during CCCV charging

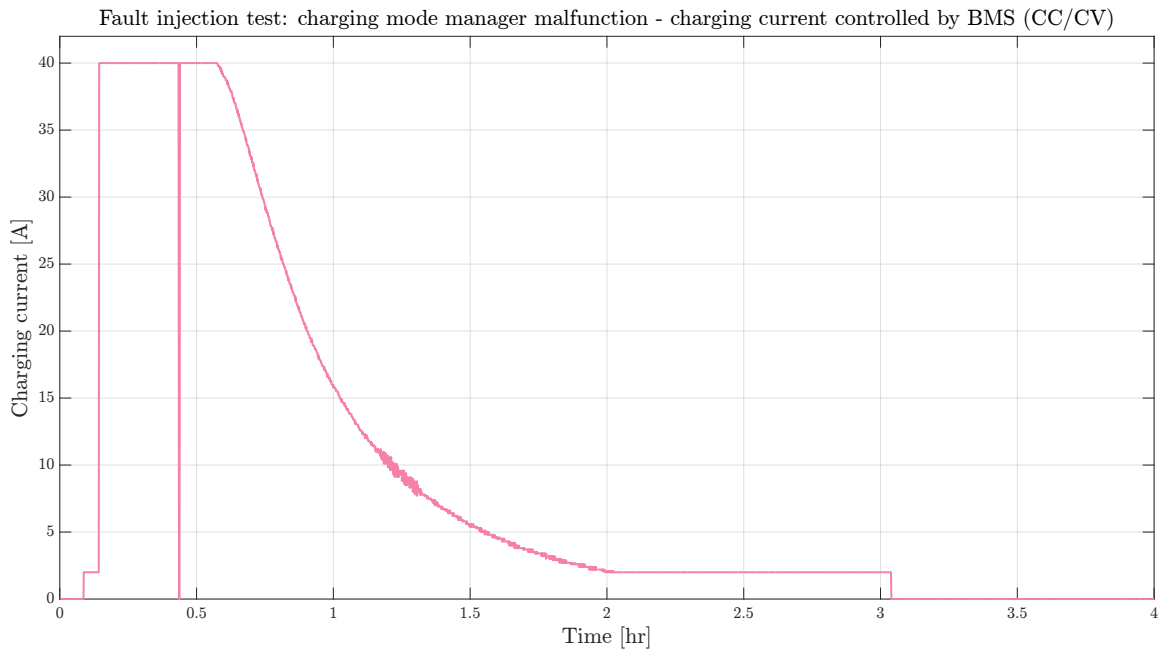


(b) Cell temperatures during CCCV charging

Figure 7.15: BMS functional testing: CCCV charging at low temperature



(a) System voltage (32s) during charging



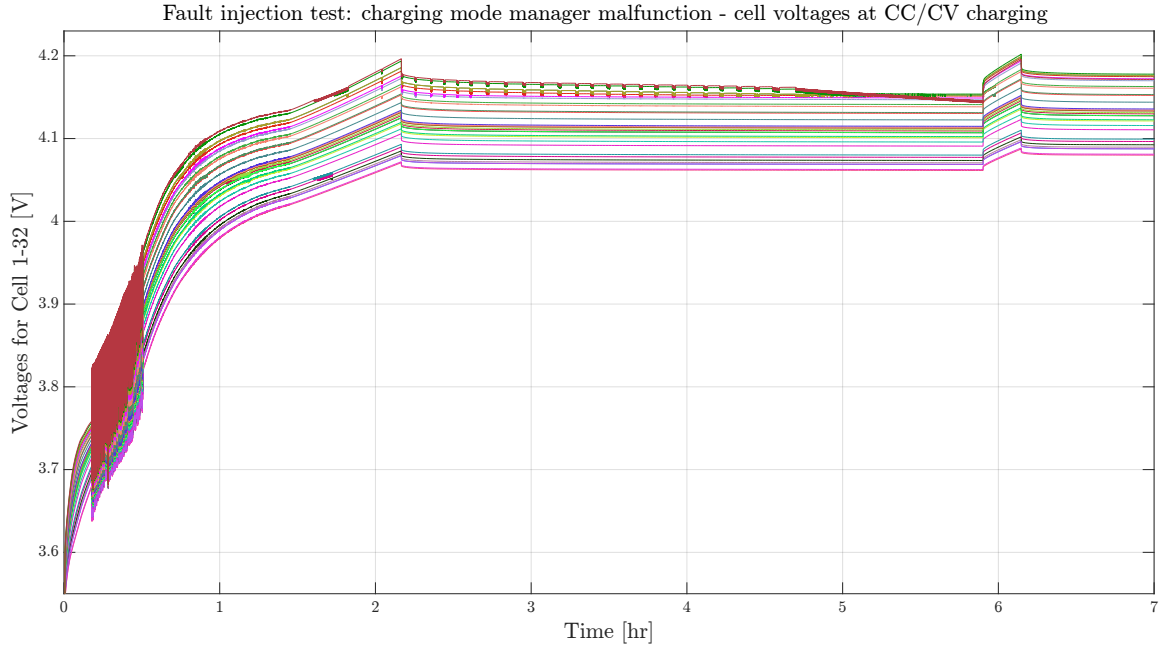
(b) Charging current controlled by LiB BMS

Figure 7.16: BMS functional testing: CCCV charging at low temperature

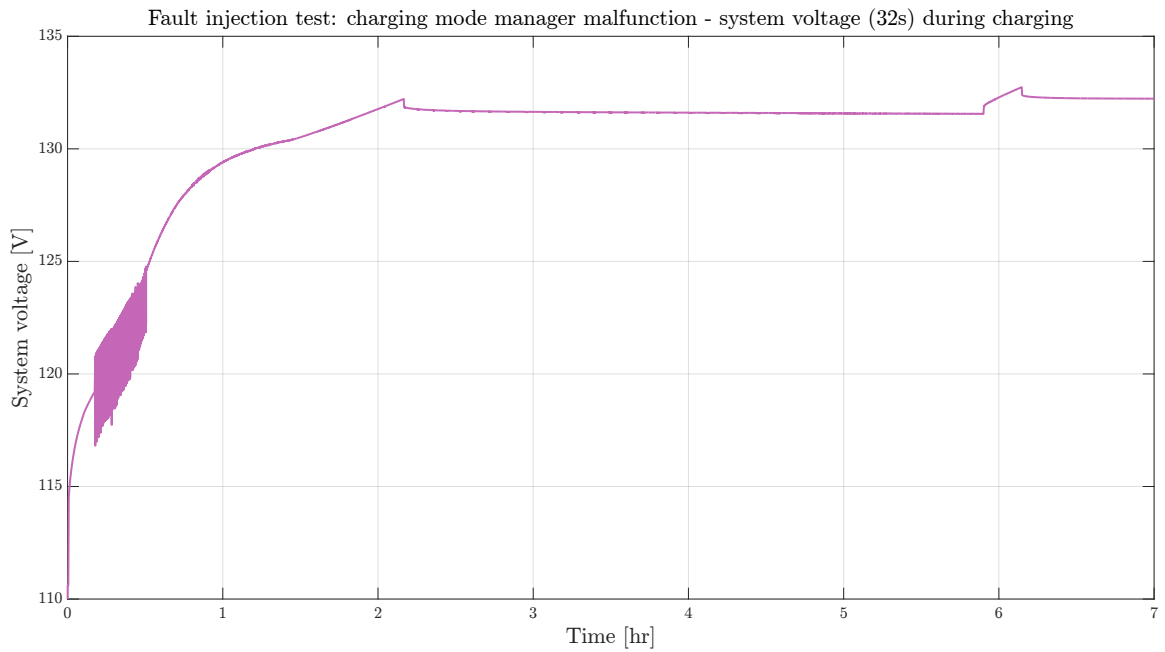
7.5.3.2 Non-Functional Testing

Charge Controller Malfunction - Oscillating Charging Current

This non-functional test shall illustrate an oscillating charging current, caused by a faulty charging mode manager. Fig.7.19 shows the charging current oscillating between 0A and 40A causing all other measurements to oscillate as well. Eventually the BUT is able to "heal" itself and continues with normal state behavior. Fig.7.19 shows a malfunction of the charger, consisting on a 50% increased charging current over the system target value. It can be observed that the BMS is interrupting and re-starting the charging process many times, until the current is kept below the pre-defined system's target current. The number of times and length of time periods between consecutive attempts can be pre-defined in the BMS settings.

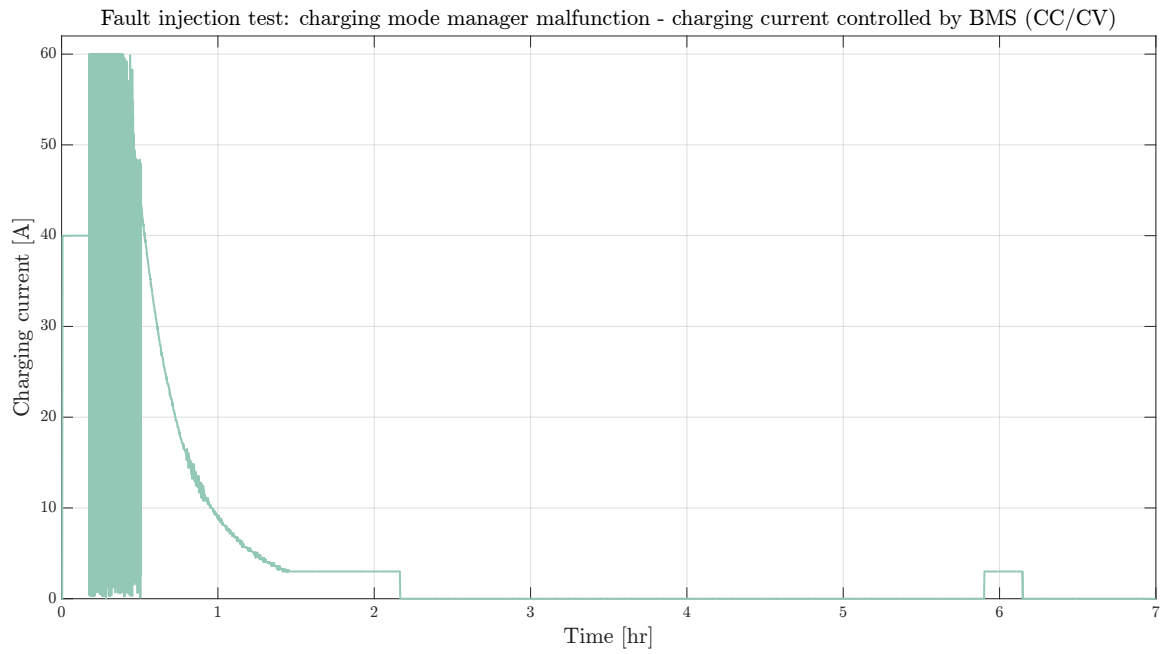


(a) Cell voltages during CCCV charging

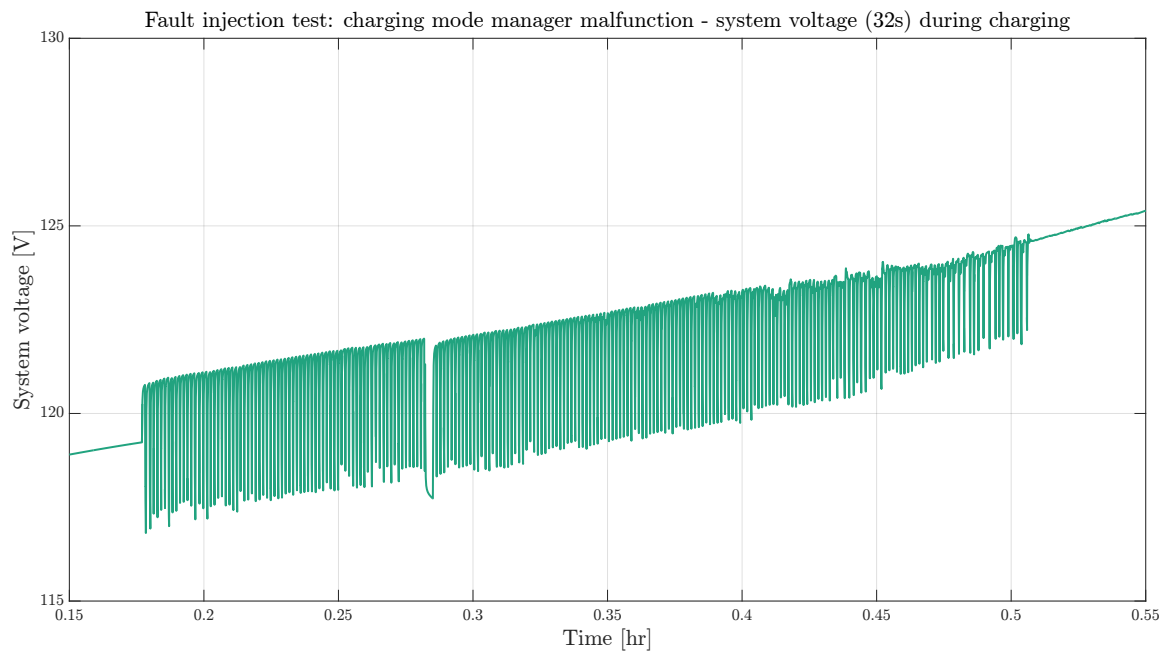


(b) System voltage (32s) during charging

Figure 7.17: BMS non-functional testing: charging mode manager malfunction

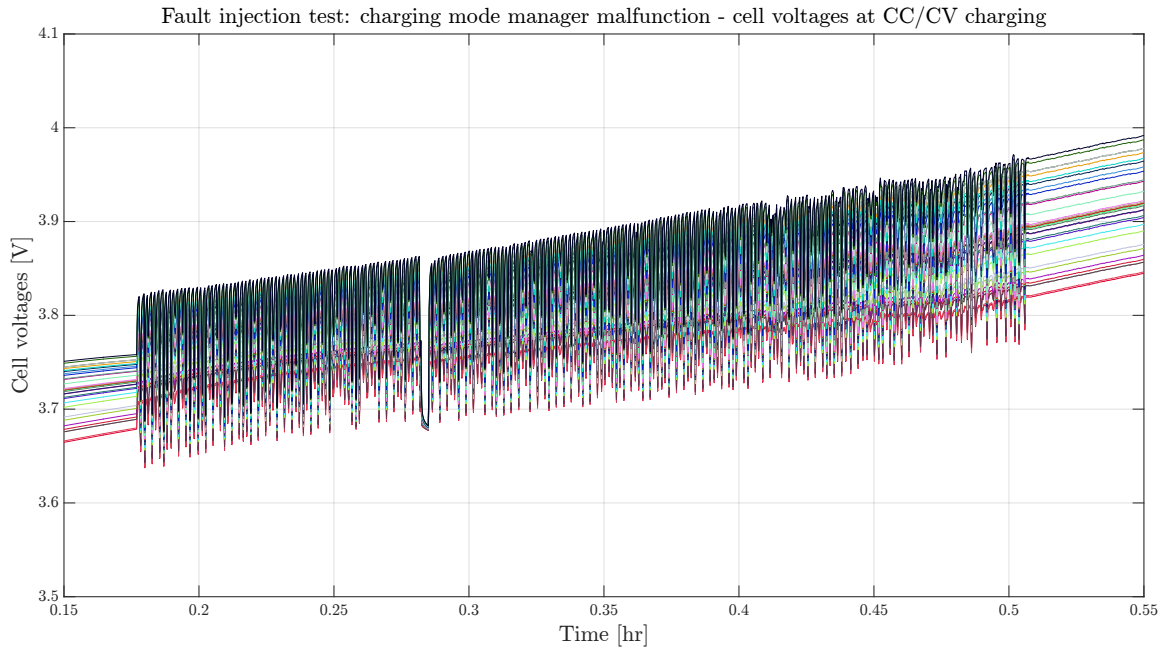


(a) Charging current controlled by LiB BMS

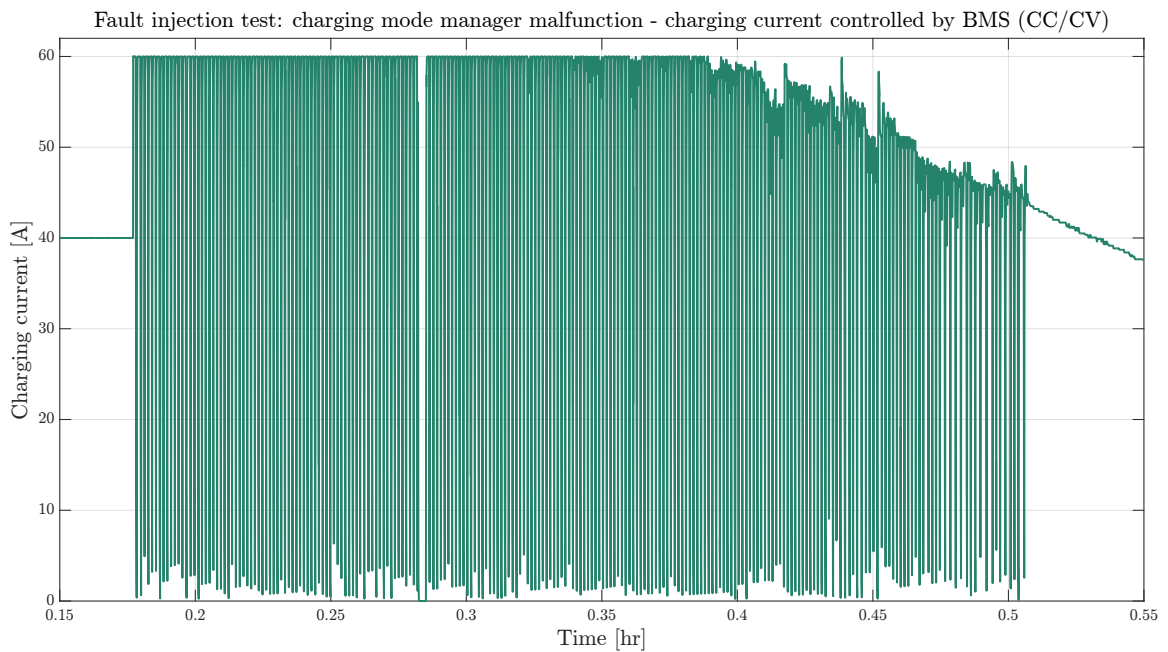


(b) System voltage (32s) during charging

Figure 7.18: BMS non-functional testing: charging mode manager malfunction



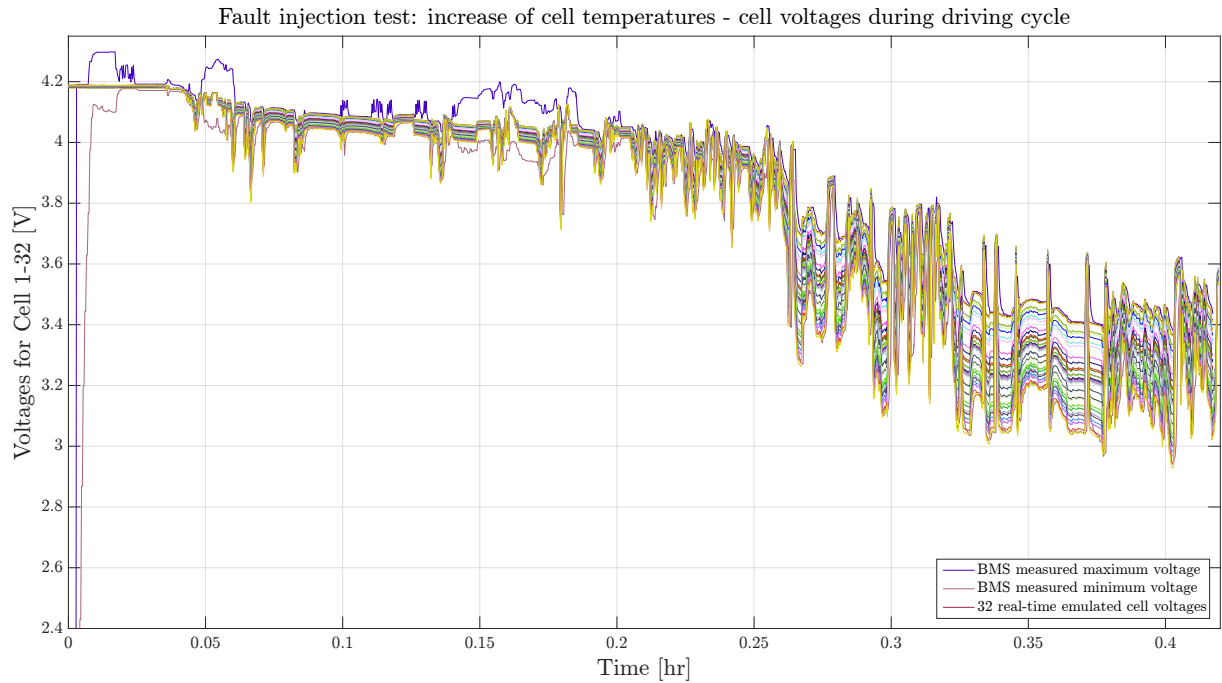
(a) Cell voltages during CCCV charging



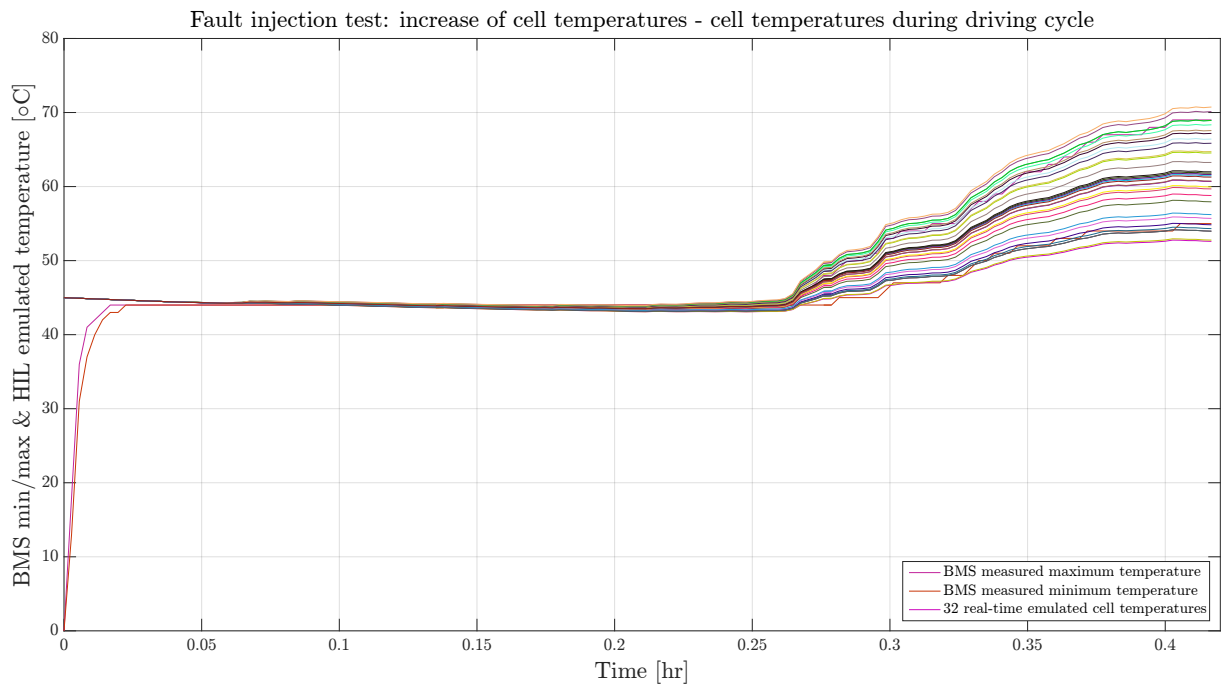
(b) Charging current

Figure 7.19: BMS non-functional testing: charging mode manager malfunction

Temperature Increase



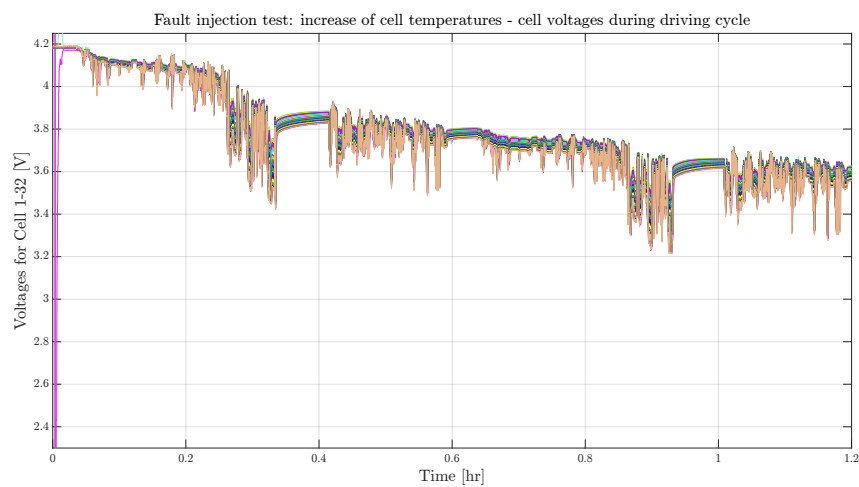
(a) Cell voltages during load



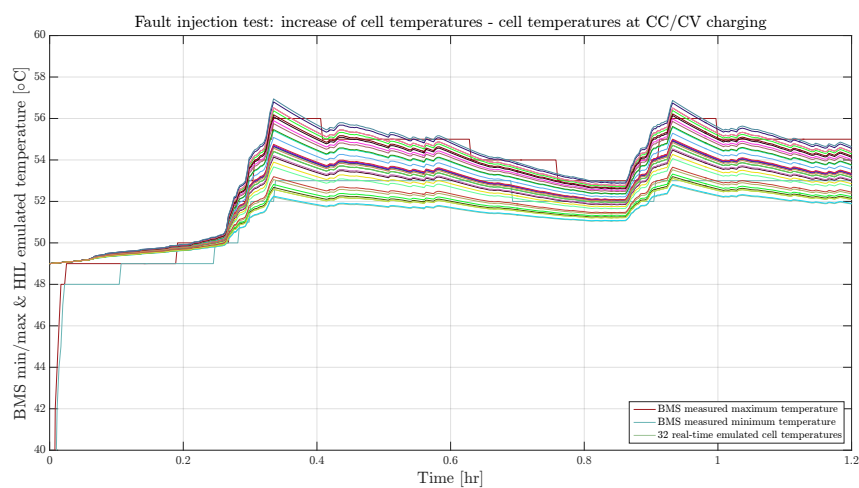
(b) Cell temperatures during load

Figure 7.20: BMS non-functional testing: cell temperature increase during load

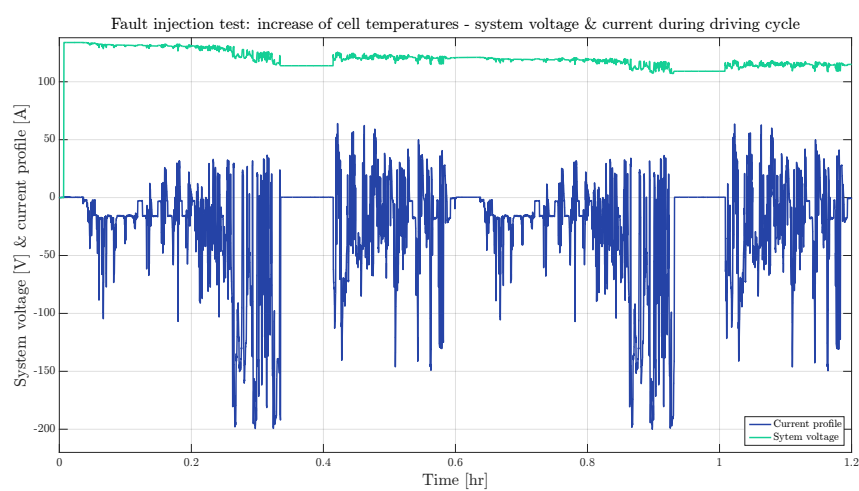
Cell Temperature Increase



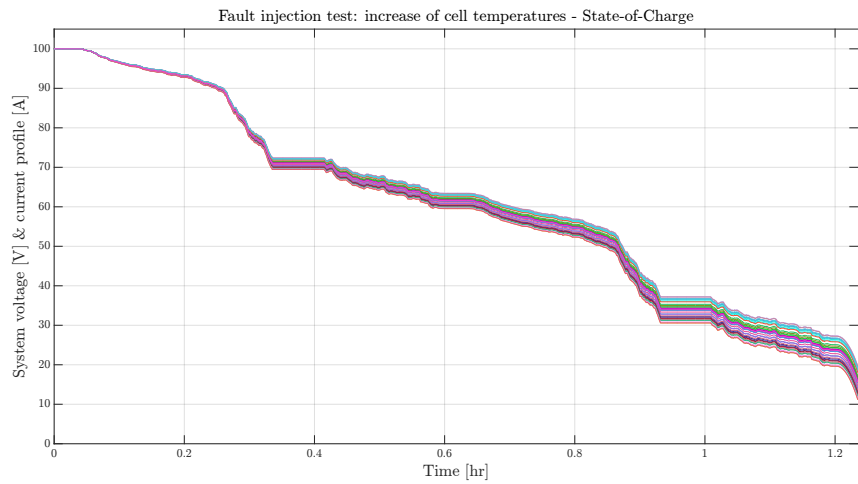
(a) Cell voltages during dynamic load cycle



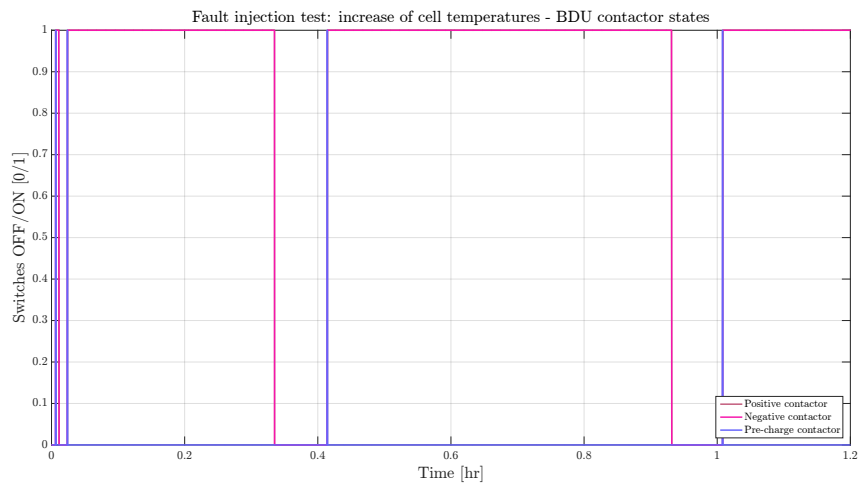
(b) Emulated cell voltages and BMS min/max measurements



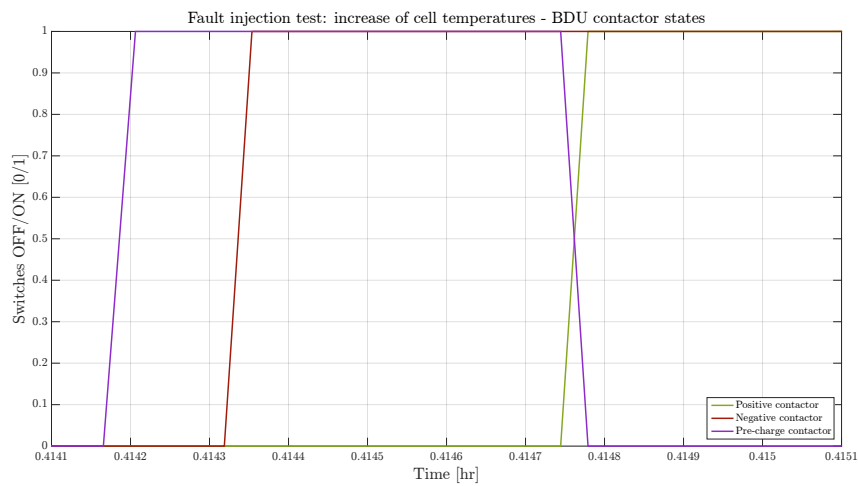
(c) System voltage (32s) & load current



(d) Individual SoC during driving cycle



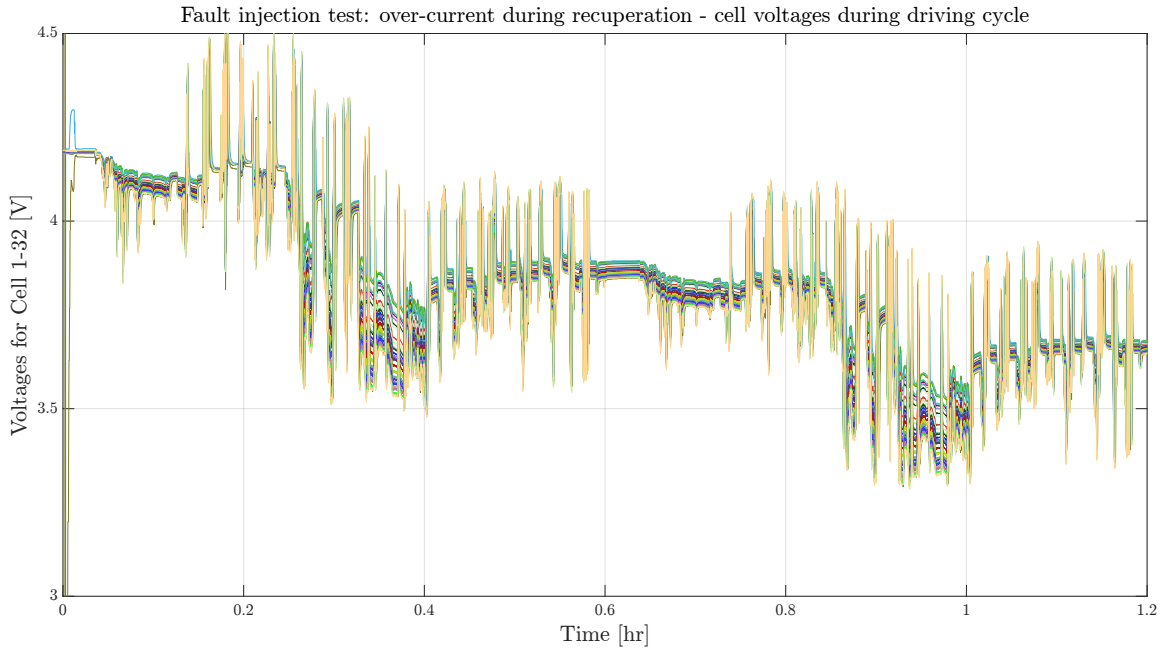
(e) Switches - BMS opens when $T > 55^{\circ}\text{C}$



(f) Pre-charge sequence

Figure 7.21: BMS non-functional testing: cell temperature increase during load

Recuperation

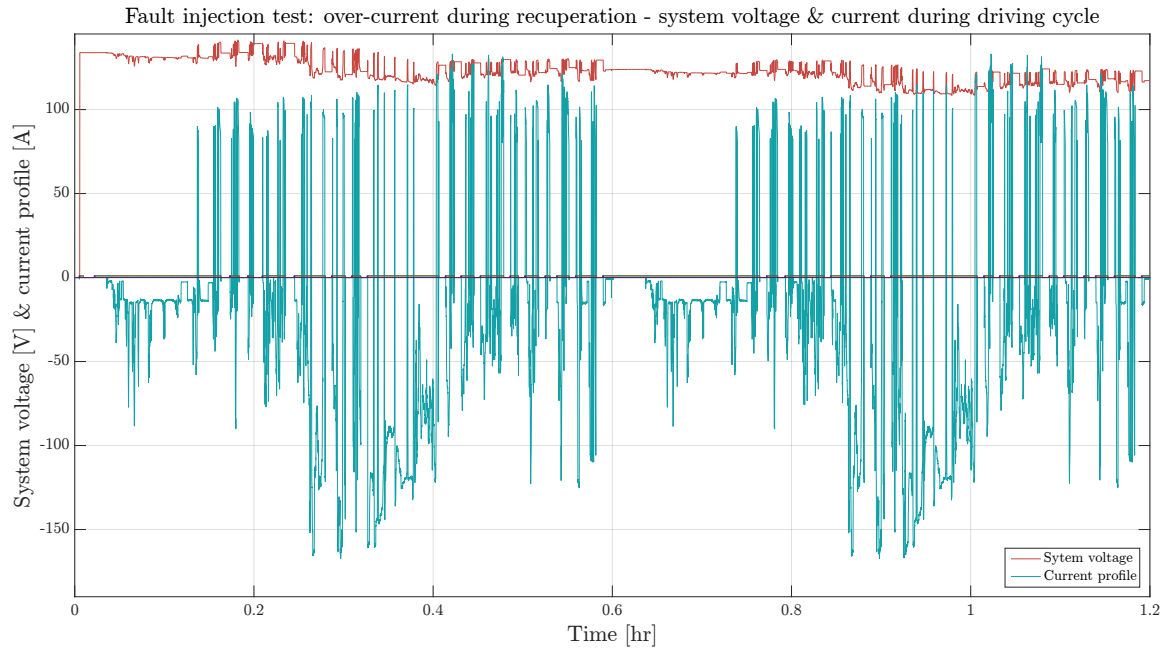


(a) Cell voltages during driving cycle and over-current.

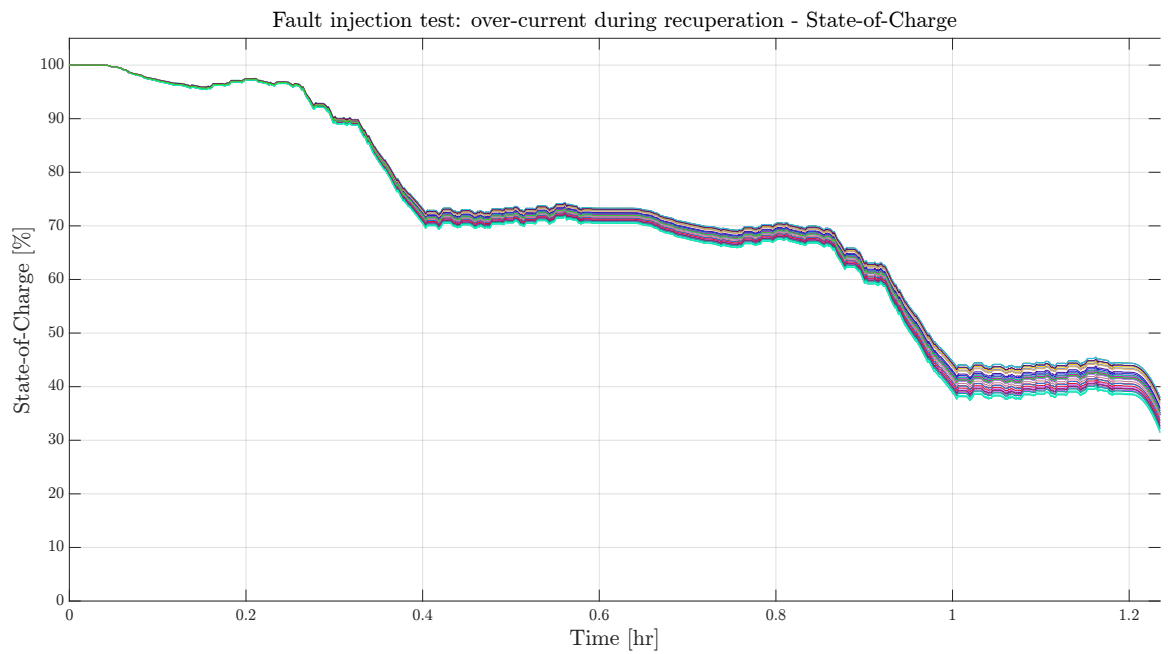
Figure 7.22: BMS non-functional testing: recuperation.

Cell Short-Circuit

Results from a dynamic discharge, i.e. test case 3, are shown in Fig.7.21(a). Results from real-time emulated cells voltages are compared with BMS measured maximum and minimum voltages. The detailed view highlights the practical limitations of the BMS under test to measure cells voltages, in terms of accuracy or limited sampling frequency (1 second). Results from an over-voltage during regenerative braking, corresponding to test case 4 (7.9), are plot in Fig.7.23. Some battery cells may be subjected to an over-voltage when a very high regenerative current is experienced at high SoC levels. The BMS detects the hazard and opens the relays of the switch box in order to stop the high inrush current. After a pre-defined time, the BMS may close the relays and operate again in discharge mode. Fig.7.21 shows test case 5, i.e. over-temperature during discharge. High initial temperatures and limited cooling capabilities are pre-defined in purpose for this test. Hence, the pack's cells experience over-temperature during discharge. Again, the BMS detects the hazard and opens the relays of the switch box in order to interrupt the discharge. After a pre-defined time, the BMS may close the relays and operate again in discharge mode. The response to an external short close to cell's tab is evaluated on sample test shown in Fig.7.24. Thermal run-away of the shorted cell cannot be avoided by the BMS by means of cooling management or interrupting the discharge. During this short circuit test it is also relevant to evaluate the max. temperature achieved by adjacent

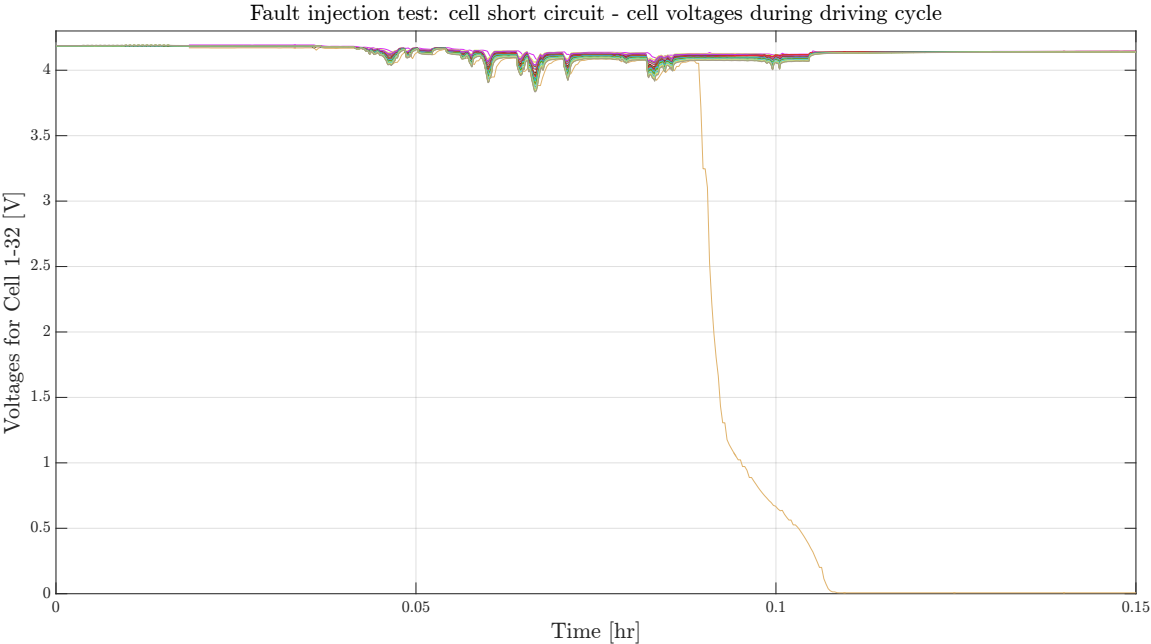


(a) System voltage (32s) & load current (over-current).

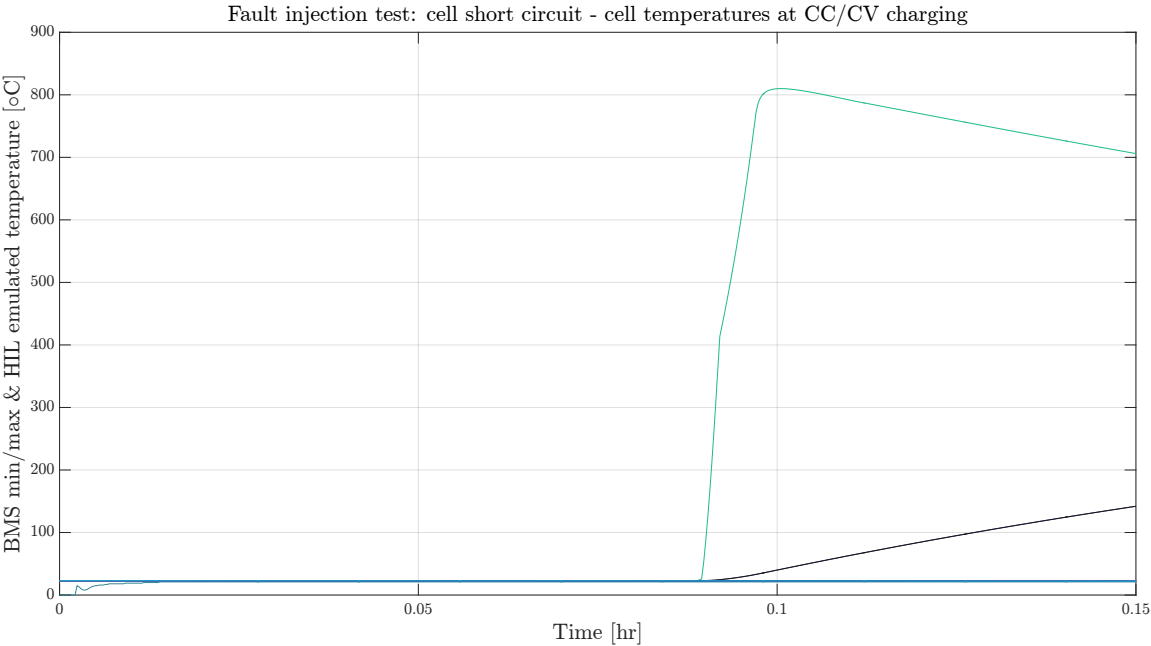


(b) Individual SoC during driving

Figure 7.23: BMS non-functional testing: recuperation.



(a) Cell voltages during cell short-circuit.



(b) Cell temperature during short circuit.

Figure 7.24: BMS non-functional testing: external short.

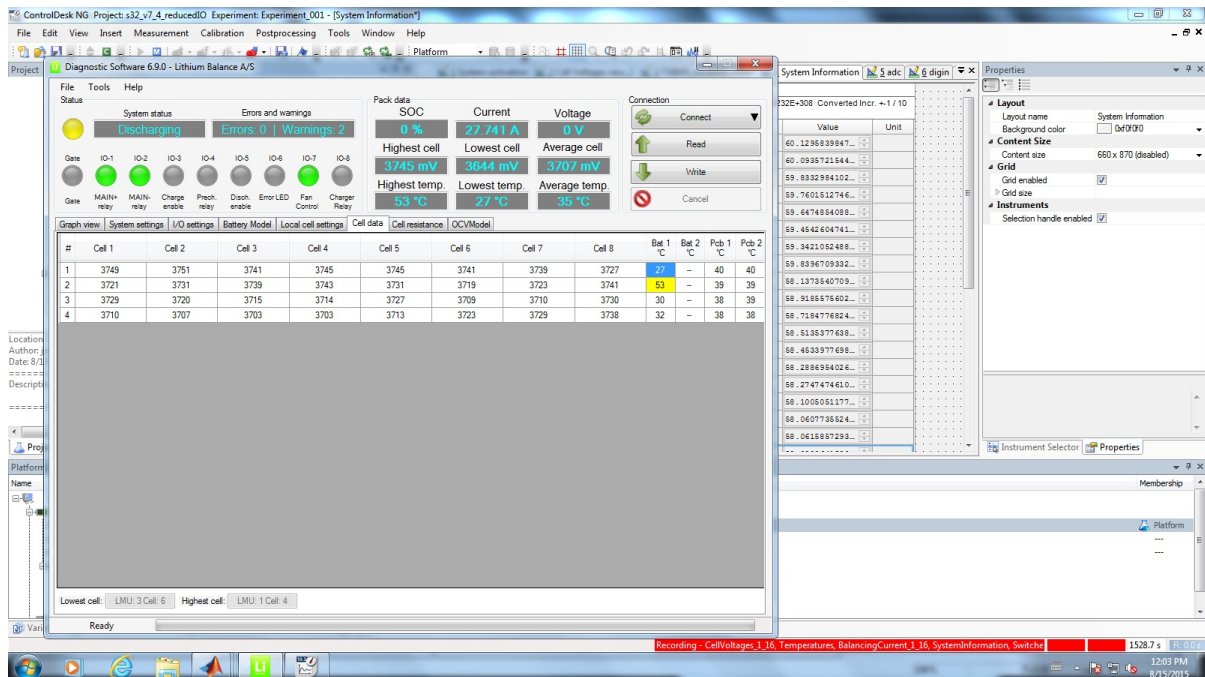


Figure 7.25: Fault insertion: Temperature scaling error

cells, since a chain reaction may happen if a certain threshold is exceeded. This limit may vary from one Li-ion cell chemistry to another.

7.5.4 Summary

Theoretical principles of BMS testing were introduced. A generic discussion about existing definitions for testing and its goals was given, highlighting the fact that many different point of views exist. Therefore different testing strategies may be proposed. An overall classification of these strategies was provided, according to the testing level, method or process. Pragmatic approaches, which may focus on critical aspects of the system, are often a solution preferred over theoretical approaches. One of the reasons for this is that it may be not always possible to perform all the possible test cases suggested by theoretical approaches. In fact, empirical evidence may give understanding of the scope and limitations of testing and suggests possible strategies to perform it. That said, a practical approach to develop ad hoc SW and strategies for BMS testing at system level on a commercial HIL simulator was given. The focus was on BMS products in late stages of development. A strategy to define the system's FRs and NFRs and to design suitable test cases was presented. A comprehensive list of test cases was proposed, including FI modes, followed by an extensive collection of experimental results. The interaction of the BMS with the system-level HIL simulated battery was shown under many different scenarios. The immense possibilities of the setup and the proposed approach were demonstrated, in terms of testing times, flexibility, traceability, ease to reproduce or safety beyond the

normal range of operation or under FI scenarios. On the other hand, it should be noted that the greater the level of complexity of the tests, the higher the demands for ad hoc development of SW and strategies. In fact, independent SW development processes for BMS testing should be considered.

8 Developing ISO 26262–Compliant Safety-Critical BMS Application

There are hundreds of ECUs spread through the car, 100 million or more lines of software code, security issues, infotainment, driver assistance... No matter how good you are, you are not going to write 100 million lines of code without bugs and doors for hackers.

– Wally Rhines

8.1 ISO 26262 - Functional Safety Concepts and Testing for BMS SW Development

Our modern automotive industry is going through radical changes, the car is transforming from computer-like car to a car-like computer on wheels. The software becomes a strategic in-house product and no longer in the responsibility of suppliers, the first on the market will gain technological advantage to its competitors as the amount of functions, especially for battery controls, is together with the overall vehicle functions dramatically increasing overtime together with algorithmic complexity and number of sensors with its information fusion. The used models to describe battery characteristics are, even though after years of studying, not fully adequate and require more efforts to make it fit, however, that will take more time. At the same time, the requirements for safety and reliability must be secured at all times. The objective to overcome these challenges of increased complexity handling can only be achieved by following new methodologies in development. As with every hard- and software component, systems and software products contain bugs which can cause faults and even result in damage claims, fatal injuries and lastly jeopardizing the company's reputation. Some rethinking had to be done, validation, validation, validation. As rare events happen rarely but still too often, SW testing must be carried out. Verification and validation (V&V) rely on target in vehicle testing, which is rather expensive and late in the process, new opportunities lie in virtually in any automated

testing and simulation to design verification as much as possible to find these rare cases. Automotive silo-thinking will not be sustainable, the way of working has to be changed towards cross-functional teams, fast iterations to learn quickly from the past mistakes as described in the Agile/SCRUM manifesto. The SW must adapt to imperfect sensing with uncertainties. This chapter will talk about safety-critical, model-based battery management system design which address the new challenges in automotive SW engineering. Simulation tests will reduce bugs in the early stage and improve safety. The solution of automation of ISO26262 work-flow will be effectively implemented using model-based design. International Electrotechnical Commission defines safety as following: "Freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment." Functional safety belongs to the overall safety concept and is connected to the system behavior in correct manner after system excitation. Hence, the goal is to ensure the safe execution of expected system behavior, regardless of internal and external environmental influences in the best possible way by removing unacceptable risks. Therefore, the ISO 26262 proposes a development procedure throughout the project (focusing on requirements specification, design, implementation, integration, verification, validation, and configuration). Potential risks are classified into integrity levels (SIL/ASIL). However, it can be noted that there exists misconceptions about safety standards requirements, such as that the classical V-cycle development process is required. It is also wrong to assume, that code can be written and only later, a layer of certification documents can be added to be ISO compliant. It can also be seen that many developers don't see the need and dread for requirements documents or structural coverage, and assume that just coding is sufficient. But that is just a wrong conclusion. If we consider project costs for high-integrity software solutions, coding accounts for less than 15%, most of the time is spent on project management, documentation, enforcing consistency, that the code is verifiable, as well as the maintenance of software integration processes. BMS developers and OEMs spend a lot of money on verification, a part of the overall quality assurance. Although, the classical V-cycle is prone to late detection of issues, efforts have to be made for costly rework cycle. Often programmers just start coding to prove a concept or just quick and dirty, and only later a layer of safety is added. This thinking will end up in huge costs and delayed time-to-market, and despite every effort, safety and quality issues are still present. In this work, we will focus on the ISO 26262 - 6 part, product development at the software level. It must be pointed out that not all functions need to be considered as safety-critical as long as there is a clear structure and separation, determination and understanding of safe states, degradation strategies and alternative measures. The summarized content below is collected through numerous discussions, workshops, and tool evaluations around this topic with companies like Timing-Architects [<https://www.timing-architects.com/>], Model Engineering Solutions [<https://model-engineers.com/de/academy/webinars/>], FENGCO Real-time control [<https://www.fengco.se/en/produkt/targetlink-2/>], BTC Embedded Systems [<https://www.btc-es.de/de/>], and ANSYS [<https://www.ansys.com/products/embedded-software/ansys-scade-suite>] which were very helpful in answering my questions, and providing data during this work.

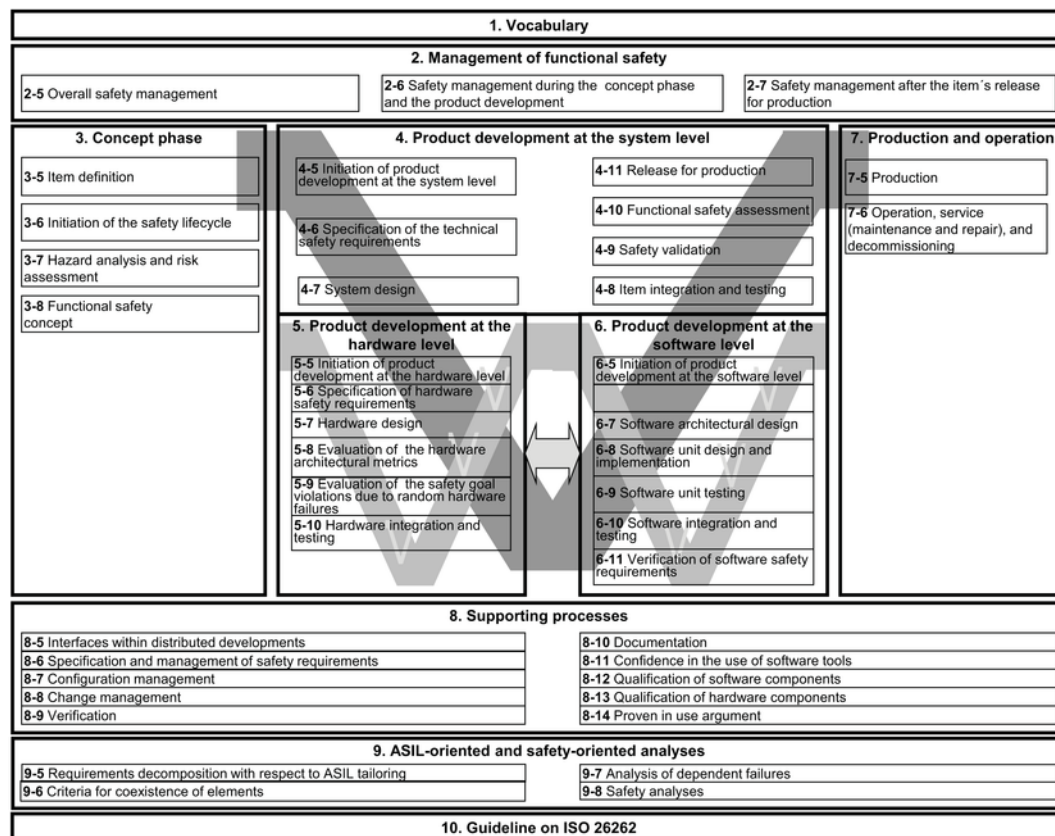


Figure 8.1: Overview of ISO 26262 [<https://www.iso.org/standard/43464.html>]

Specification of SW safety requirements The objective here is to plan the functional safety tasks for all sub-phases in the SW development following the reference plan in Fig. 8.1 but with first level of life-cycle tailoring (introduction of agility: SCRUM). This includes amongst others, safety plan, SW verification plan, design and coding guidelines. The prerequisites are that a safety plan and technical safety concept was derived (not scope of this work), system design specs and integration and testing plan are in place. As stated in Fig. 8.1, the model recommends to start with specification of software safety requirements (SSR). They are derived from the technical safety concept and system design specification. In here, the hardware-software interface requirements are defined and that they are consistent with the technical safety concept and system design specs. Safety-relevant SW features, functions, performances and interfaces may affect and cause system or SW faults [467]. It is important that the established SSR must consider all individual software-based functions whose fault could cause a violation of the derived SSR from technical safety requirement in regards to software. These are functions responsible for safe state, functions for fault detection and handling, or real-time critical functions. It must be pointed out that bad requirements generate bad software (garbage in, garbage out). This will result in non-verifiable code and non-consistency. The SSR verification criteria discusses accuracy, if the derived requirements are accurate; consistency, no

oppositeness within requirements; robustness, do the requirements consider out-of-range inputs/noise; verifiability, can it be tested that the software fully represents the requirement; and lastly, traceability towards SSR which helps detecting that all system requirements are supported or SSR are not related to system requirements.

Software architectural design During the software architectural design phase (SAD) the SW architecture is derived which realizes the SSR. It realizes all SW-C and their interaction in a hierarchical structure, such as interfaces and data paths between all SW-C as well as process sequences and timing behavior. The work products are SAD specification, SSR specification, dependent failures analysis and software verification report and can be expanded through out distributed ECUs. This implies that if code or SW-C with different ASIL levels, or non-safety related SW has to be integrated, all of the SW-C have to be treated accordingly to the highest ASIL level, if and only if the criteria for coexistence is met (freedom from interference). For SW partitioning (SWP), it must be considered that within the distributed resources no violation of interference will occur; that the SWP is supported by hardware features and that the SWP component is developed after at least same ASIL level; during SW integration process, SWP must be tested. A good architecture is known for hierarchical structure of software component (SW-C), with restricted number of SW-C/interfaces, restricted dependency between SW-C, appropriate scheduling and fair amount of interrupts. Errors may be found through range/plausibility checks, data errors, and control flow monitoring. Error handling in field can be addressed with static recovery mechanism, independent parallel redundancy (redundant computation using different methods). SAD can be verified by simulating the dynamic SW-C, walk-through, formal verification and control/data flow checking. A correct architecture is crucial, however, in many cases it is not sufficiently considered as a main task during the SW development phase, it is even built bottom up or taken from other previous projects without thoughtfully analyzing its re-use. In contrary, this work put a great focus in designing the SW architecture.

Software unit design and implementation The SW-C or units are implemented as specified in consideration of the SW architecture and SSR. During this sub-phase, static verification of the SW-C and implementation has to take place, both for non-/safety related requirements. This work uses model-based SW development rather than coding directly the source code, using modeling guidelines. Due to automatic code generation, production-ready code can be produced, so all the properties need to apply to the model and generated code. The design principles to follow are examples of the correct order of SW-C execution as defined during SAD. It needs to be sure, that the correctness of data and control flow between all SW-C are fulfilled which requires consistent interfaces between all units. Also, the objective is to have a clean model, simplicity is valued over complexity in terms of readability and later modification and additional SW units. Through a concept of testability, robustness is guaranteed (out of range values, deviation by 0, wrong data/process flow). What helps for robust SW unit design is a controlled entry and exit points, no dynamic variables or objects, that the model gets checked for correct initialization to pre-

vent multiple use of same variables. Also, a fine design rule does not allow global variables, limited use of pointers, implicit type conversions, jumps, and recursions since this may lead to data loss, or faults due to memory overwriting. At model level, semantic integrity check, traceability analysis, simulation to check algorithms and causality and to show that the model is correct mapped from the SSR, and formal verification will support to verify the SW design, if there is confidence in the SW tool chain. In regards to the safe mode managers, there can be only one transition firing in a cycle, this means that for each cycle, the model calculates the output vector from a stable input vector in a deterministic manner, so no interaction with external environment. A different tool chain, not the industry standard Matlab/Simulink is used to design the model and generate the code, rather a tool which is known from the aerospace industry, called SCADE. More details on this product will be given later. However, the tool generates C code which fits the requirements for safety-critical embedded software, such as OS independent and MISRA C complaint, structured, no pointers, bounded loops with no recursions, easy to read and static memory allocation.

Software unit testing In this step, it must be confirmed that the SW-C/units fulfill all the software unit design specifications and only map the desired functions. For model-based SW development, the testing has to consider all parts of the implemented model and the test object is the code derived from the model or the model itself (in case of SCADE), other tools require both to be tested. In traditional software development, MIL, SIL, PIL and HIL will be used from design phase up to implementation. Further details are discussed in Section 8.1. The question arise, against what should test cases be written. The first point is that test cases shall be written based on detailed design (low-level tests), to give confidence that the code complies with detailed design, and in case that if uncertified code generators are used. Secondly, test cases in regards to the SSR (high-level-tests) have to be executed to verify that the SW-C follow the SW requirements which is needed for all cases. The recommended test methods consists of requirement-based testing, interface testing, fault injection test, CPU load testing, back-to-back testing between model and code. Besides, the question if all requirements are implemented correctly must be addressed by requirements coverage analysis. In addition, the question if only the requirements were implemented must be addressed via structural coverage analysis, which includes statement coverage, branch coverage and MC/DC. It is advised the to carry out the coverage analysis at model domain (functional level) due to higher efficiency.

Software integration and testing In this step the code is integrated on HW and tested to demonstrate that the SW architecture is complete generated and executed as embedded software. Here, the entire SW comes together, all coupled SW-C will be tested on integration level with their interfaces even distributed over different ECUs. For the testing, it follows the same recommendation as for SW unit testing, however, the structural coverage differs, here function coverage and call coverage are with special attention. The test environment reaches from HIL, to entire E/E

architecture in a mule car and finally in the vehicle itself.

Verification of SW safety requirements It shall be demonstrated, that the generated C code fulfills SSR in the target environment.

Agility and teamwork SCRUM is an framework for teams to develop SW, with e.g. 2 week sprints. It is an iterative and incremental way by breaking down the SW-C into smaller units to built the product. The challenge in BMS SW design is the fact of having that concurrent thinking of control engineer (algorithm designer) and software engineers (implementer), the control engineer focuses on stability and performance following the control theory laws, while the software engineer thinks about resources (memory, wcet), robustness, testing, accuracy. Hence, only if both parties work together, the SW development process will be successful. In the end, it must be pointed out that the measurements for certification must be considered right from the start. The complex and risky SW units, functions, features shall be SCRUM'ed in the early stages and then enter the formal process. A general focus shall lay on the SW safety requirements.

Software Unit and Integration Testing Using Model-Based Testing

Among the most frequent reasons of bugs in large software projects are poorly defined requirements and their poor management. The requirements represents anything from an abstract to well structured implementation detail of a system and how it should function, which must fulfill a capability, standard, specifications, required by the designer to solve task or achieve objectives. They are defined and written in the beginning of the project and define costs estimation, timing, coding, testing and integration on target. Therefore, it is crucial in the early development stage to verify the model and if it meets its requirements. In the following, the methods are described how a model can be verified. The first step is by simulating the model (functional testing) followed by requirements testing, while all possible test cases must be meet. Formal verification is especially important for safety-critical functions. Commonly, in the V-cycle development process, the definition is integrated in a manual, protracted fashion, applying static testing methods which are prone to errors where stimulus input functions are compared with the output. The requirements are tested in a early simulative manner and by requirement testing.

Here, a combination of model-based software development (MBSD) and model-based testing (MBT) may come into place, which provides intensive support in developing a higher automation of manual writing test cases and checking of functional correctness in an early stage, meaning, that the requirements are represented by the model, which must guarantees the correctness of interpretation. Only then, the functions defined by the requirements can be validated and checked for properness. In MBSD profusion of techniques are used to convert the model into production code. [Source: TargetLink Addendum] Instead of manually reviewing the code, extensive testing, such as back-to-back testing is applied, to demonstrate that all software components fulfill strictly the

specifications and do not run further unintended functions. Appropriate, automated test vector generation ensures that the code review can be performed in higher degree of quality for total model- and code coverage metrics in comparison to manual review. On the other side, the integration tests shows that the software framework and architecture in functional and interaction between different domains is executed correctly. Following the functional safety standards, IEC 61508-3 and ISO 26262-6 recommend functional and black-box testing, combined with dynamic analysis, structural coverages testing, as well as expected demands on memory and resources for software components (unit) and integration. To ensure that the software is running correctly on the target ECU, dynamic testing (back-to-back) is vital to provide safety-related functions. During model-based software design we want to test our model and the generated code. The model represents the implemented requirements, so to say model-based specification and is our reference, while the code is integrated later into the BMS. The combination of a running model and its simulation capabilities of the plant model [Source: TL Reference work-flow] allow a systematic but quick performance check making possible to modify and adjust functions during development. In the end, the code which is running on the target has to behave exactly the same as the model. The testing and validation plan in the model-based development environment follows two layers:

Model verification - Does the model fulfill all the requirements?

Automated/hand code generation verification - Does the hand-coded software or automated generated production code preserves the behavior of the model on the target platform?

It must be ensured that the tests which are carried out on the model level, also have to be executed on the generated code. There are basically three main methods, simulative method on model level (model-in-the-loop (MiL)), method based on generated code but executed on host (software-in-the-loop (SiL)), and finally the method executing the automated generated code on the target platform (processor-in-the-loop (PiL)). Therefore, how much time has to be spend in writing and executing test cases which must cover all functional requirements? In general this is a very time consuming task. Developing appropriate test cases is based on a two-stage, manual and automatic approach.

8.2 ISO26262-Complaint Model-Driven Software Development

As software is increasingly replacing hardware and humans, and used in safety-critical applications such as the BMS, bugs in the SW can lead to catastrophic effects. However, these allows for more complex adaptive systems and cost reduced solutions, with the drawback of always providing quality assurance. This requires a shift beyond the

standard software developing processes to provide high-integrity software. This includes extensive effort like planning and project management, generating documentation, keeping work traceable, consistent, verifiable and maintainable under the highest agreed quality assurance.

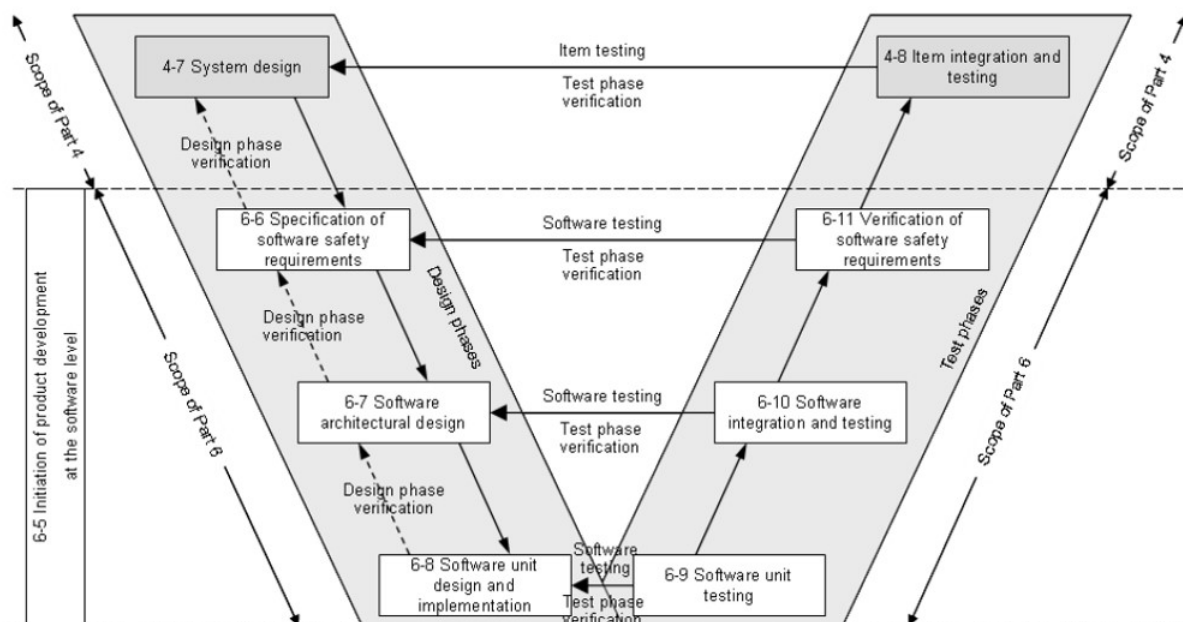


Figure 8.2: ISO 26262-6, Reference phase model for the software development <https://www.iso.org/standard/68383.html>

There are different conceptions about safety standard requirements for software development, e.g. the necessary SW development life-cycle is based on the V-cycle framework. There are converging views and work culture that coders need start implementing the functions and start testing and verification, and at the very end a layer of certification is added. There is also the assumption that requirements are only required for e.g. ISO 26262 certification process, and that for unit testing structural coverage is only specified but not recommended. Beforehand, the differences and definition for security and safety has to be mentioned when we talk about functional safety. Security, meaning free of care, while safety stands for uninjured and free of harm. When we talk about safe BMS, we are talking about a BMS which does not harm a person, e.g. if we lose the BMS information its function, the actor does not need to care as long as nobody gets harmed in the car. We have seen pictures of burning EVs at the roadside, and the question is, whether this care safe, it depends if all the driver or passengers are still in or could get out. If they had time to leave the car, and thats the philosophy, driver could still control and stop it, and leave safely, the FSM responsible for the vehicle and battery safety function did everything correct. If the car is lost afterwards, that is not of importance, the target is to protect the passengers, and personal interests of him to buy a new BMS is not the focus

and discussion within safety measurements in ISO 26262. In the following, we come to a reasoning on what to do in terms of applying functional safety to battery management systems and their software development.

A strategy was defined to how to address the tasks in developing a safety-related software model during the development process. Normally, the project does not start from zero, rather the software development process is defined and the tool chain is available from previous projects. In general, the budget is fixed during the project and extra costs for process enhancement might not be available and this makes it usually impossible to introduce necessary adoptions at once leading to prioritization. However, a prioritization for process improvements for ISO safety critical compliance has to be fulfilled at least in steps for the software development process. In the following the work shows an interpretation of the full ISO 26262 standard in terms of software development for battery management systems following the V-model. But first, the overview of requirements of ISO 26262 shall be discussed to know which really needs to be addressed for BMS SW development.

The V-model is given in the ISO 26262 for product development at the software level, starting at the left hand side, we define groups which are based on individual steps of requirements of the standard. In the first two steps, we address the development process which requires a tailoring of the BMS basis SW project toward ISO 26262 compliance and project specific requirements. In the next step, the team must systematically define the software development process. This means you have to tailor this ISO so it fits to your project specific needs. The second step addresses the adjustment of the tool chain, so it meets the requirements of the standard. These two steps are executed in the first process phase meaning during the initializations of the development. Followed by the phase called specification of software safety requirements, which is followed by software architectural design, as well as for unit design. In this step again, we create the design specifications. The fourth step discusses topics around checking and reviewing the generated models. The right hand side of the V-model deals with the ISO requirements in regards to systematic test process. It begins with the design of testing, how to do testing and what are the requirements, while on the other side the deepness of the testing is discussed and the test coverage matrix. This includes requirements testing, back to back testing and interface testing, while we also address more specific test methods, such as robustness and resource testing and error guessing based on expert knowledge. Last but not least the checking for unspecified functionality proves whether all required functionalities are available in the model, for the case of those function that are not available because they are deactivated, then they must not have any influence on the safety goals. There are important steps identified after own judgment towards ISO 26262 compliance. Important for quality assurance is the implementation of a higher and faster failures detection. It is vital to make sure that all the activities are implemented and checked for completeness. Followed by the task to avoid failures and lastly having the control of the systematical implementation and adjustments towards ISO complainant SW development processes.

Requirement-Based Methods

One way to reduce failures starts with requirements based checking. The ISO standard requires to execute requirements-based testing on unit and integration layer, where the interaction between units also has to be tested. Therefore, test management was implemented after first documented test cases and specifications were defined which got a unique ID for tracking, further all test cases are linked to the requirements. Finally, based on the test cases the appropriate test environment was defined, namely, MiL, SiL, HiL. ISO also requires to measure the coverage of requirement specifications, as well as test specifications on component level, therefore, at least one test case was designed for each requirement. Additionally, statement coverage and depending on the required ASIL level, branch coverage in SiL and PiL are necessary.

Design specifications

In this action task the software architectural design specification and components, for both the documentation is required by ISO. It defines which parts in the architecture are components or units with individual SW safety requirements allocation which in the end leads to ASIL level determination. The SW architecture is capable to be linked to the software requirements. Following subsets of requirement- and recommendation clauses are considered: Complex algorithms are executed on embedded system, therefore, it is important to have a good estimation of resources on the target, such as runtime behavior (real-time capability) and memory consumption. After the architecture is designed, the SW unit specification is created. If a model-based software development is used it enables to use the Matlab/Simulink model or at least partly. In this case, for unit specification ISO 26262 requires additional documentation for the aspects which are difficult to read out of the model itself to have a complete SW unit specification in the sense of ISO26262. This work identifies this action plan to be important because as the complexity of the BMS SW modules helping to prevent implicit/incorrect theories on the entire SW. By tracing the SW requirements into the architectural design or software units, an additional safety net is installed to increase the correctness and completeness of the requirements and vice versa. With the help of the SW architectural design and the tracing it can be checked that the SW is complete in respect to the implementation of the requirements. Also, the SW unit specification is important since it improves the maintainability of the model and it is crucial for the testing process. ISO recommends to distinguish and having different subjects working on developing and testing, in that configuration the quality of testing correlates with the quality of unit specification reducing the test efforts due to lower misinterpretations of requirements, resulting in higher quality of model and documentation, that, in the end, leads to less SW failures. It has to be prevented that requirements give room for misinterpretation to test engineers. This task is also manual step with potential in automation when you link the software model as part of the software design specifications.

Resolving failures in interfaces Since the BMS is connected to many components of the drive train (BDU,Charger,EPCM...), it is especially crucial to test the interfaces by checking the boundary values of each sending and receiving messages/signals while equivalence classes of these signal values are generated and analyzed in case that they are not completely covered. In the following, ISO26262 requires the measuring of structural coverage to understand the quality of testing.

Back-2-Back testing The code generator, who translates the models to C-code, may introduce faults (floating to fixed-point calculations), therefore, two or more functionally equivalent versions of a software are tested against each others, even on different domains (e.g. MiL vs. SiL). It is a rather timely test, as the evaluation of execution differences is difficult to automate. B2B is beneficial, because it ensures and gives confidence towards the designed test specifications during integration.

Static code analysis As an experienced battery specialist and tester, with the help of experience based testing (error-guessing) the test specifications are extended and expected parts of the model with possible error incidence or lead to errors in the past. Since BMS application is ASIL D rated, ISO 26262 requires checking the structural coverage using Modified condition/decision coverage (MC/DC). In the last step, the static code analysis is conducted (Polyspace, PC-Lint).

Team's model and requirement review The aim is to take the focus back to developing the BMS algorithms and control functions. In general, the cross-functional team shall provide a set of guidelines from which the developer can choose and adjust (MISRA C:2012, MISRA Autocode for Generic modeling design and style guidelines, Modelling design and style guidelines for the application of Simulink and Stateflow, Modelling style guidelines for the application of TargetLink in the context of automatic code generation, dSPACE Modeling Guidelines for Link, Control Algorithm Modeling Guidelines Using MATLAB) to name a few.

Unintended Features/functions As required by ISO26262, the SW shall only contain intended SW components derived from its requirements, the rest must be deactivated before safety goals are violated. This task takes place rather late in the project when all functions are implemented. With the help of measuring the structural coverage, it is possible to identify unwanted code.

Continuous process optimization ISO26262 with its functional safety extensions is applied to Automotive SPICE for process assessments. It includes the review of the current tool chain with its components are qualified. The tool chain automates the usage of modeling, modeling guidelines checking, metrics calculation, test case generation and testing. The result is that the development process is more efficient, detect errors in BMS SW, and spotting the remaining/hidden errors in early stage, such as doing applying first the guideline checker detects error-prone patterns, the readability and testability is improved beforehand generating a consistent layout makes formal reviewing and testing later easier.

Testing and reporting Fault injection, testing timing behavior of AUTOSAR multi-core ECUs, or resource usage will lead to detection of ineffective implementations, e.g. fault injection, derives relatively more benefits since it reinforces the error handling in the BMS software. It also supports to check if the model is experience missing error handling, memory corruption and so on. Besides the property of goals being reachable there are properties they cannot be detected within static analysis for reachability, something what was already identified as unreachable at initial startup, but in general, how many test goals for statements and decisions will be created no matter how these numbers will eventually increase. If there is an area in the code with is unreachable that means there is no combination of the input signals over time which can bring that system to this location. So if there is unreachable functions or code fragments, to put it that way, kind of dead code, depending on the project this may still lead to a production code. How to deal with that scenario, must be threatened individually in the project and analyzed correspondingly by the engineers. The term Covered represents that instance, in which at least 1 test case was covering these test goals, while unreachable means that there is a mathematical prove which shows that there will never get to that state, no combination of input signals can bring that system to this decision. Adding the results from both, covered and unreachable, will result in handled. Reaching 100% handled means that all structural goals have been analyzed and for all structural goals a result was obtained. The result can be either that its covered with a test case or it is proven to be unreachable. The goal with testing is first of all is to bring the handle to 100%, meaning that all has been analyzed. For unreachable results, each case has to be analyzed and checked if it is acceptable or not. In case of unreachable function call that must not happen because that can disrupt the functioning of the entire software. In case of a decision which consists of a multiple condition this should also be analyzed in detail (e.g. looking in the statements that are unreachable). For the reference case, there is a decision that can never become false, if the project can accept that, it can be moved on.

8.3 Model-Based Software Development

When we talk about BMS application, mainly two points have to be considered, deep understanding of battery technology itself is required which shall lie within the automotive company, on the other hand, when OEMs develop, design, test, validate and finally integrate BMS software, it requires tools and methods in order to be successful also in the sense of functional safety. Here, the experience is mainly on the side of the tool vendors. With the help of model-based software development both domains are merged. Active safety in HEV/BEV helps to increase battery and therefore road safety by decreasing also battery accidents. Model-based design is an instrumental tool to develop these technologies. When designing automotive software for H/BEV, following needs to be considered:

1. It is a valid question if a critical situation occurs, what is the right decision to take.
2. Extra complexity and challenges have to be addressed when controlling the battery dynamics/operation in order to guarantee that these systems really behave as desired in all driving situation, that requires many testing miles.
3. Sensor diversity and amount of sensors used to increase the accuracy and to fulfill the redundancy to compare sensor data against each other, also in the case of failure redundancy needs to address that.
4. Additionally, it is also very important to build a very precise description within the battery box including the different sensors which contributes to the overall picture of the battery.

The challenge of BMS software function development is to know how complex systems can be in a very early stage. Especially in EV development with battery pack design most topics are fairly new and mostly rely on real testing data, however, vehicle/BMS prototypes are often not available especially if a complete new product is developed. This is a disadvantage when various designs shall be tested. There is also no time available to wait until a prototype is available. Again, the developer have to make sure that all of the new features are working under all internal and external conditions and during the entire lifetime. Therefore, model-based design seeks to be a complementary approach which adds value to support the software developer to the idea and concept for new functions during the early design phase. Developers are able in a rapid prototyping environmental way to try out new functions and identify (non-)conceptional bugs in the algorithms to quickly fix them and re-verify and test them to ensure that the algorithms behave as expected. Especially in BEV safety-critical applications bugs may lead to failures with fatal consequences. Once the functions are developed, it is desired to quickly implement them on the target platform. Innovation cycles are becoming shorter, to set themselves apart from their competitors and being faster on the market is always of advantage. Therefore, in order to manage these challenges, model-based SW development with automated code generation is often used and state-of-the-art for many OEMs and suppliers. In general, model-based development refers to the well expected V-diagram, meaning that after defining the functional requirements, the design descriptions are derived from those. The design descriptions are grouped into executable specifications in terms of virtual prototype or simulation model, which allows to simulate the design decisions and continuously proof, improve, and verify it based on the defined requirements. Once the point is reached where all requirements are fulfilled the code can be generated and the software is then deployed on the target ECU. On the right side of the V-diagram you go up again which represents the testing. When we talk about MBD, for BMS functions it can actually include a wide spectra of different tasks. It can be applied in a way that MBD is used either for a single function unit but from the supplier perspective, for example the development of monitoring algorithms, or even going beyond that idea. At the same time, it means that with MBD a virtual prototype of the entire battery monitoring functions can be built just based on taking all sub-elements of single units and integrate them into one overall

virtual prototype, which allows complete system simulations in terms of real-life internal and external environment, while no lim. MBD allows to verify functions in the earliest development stage and long before any functional prototype is available. An important consideration when testing functions in early stage, assigning a test team bears some risks associated to test these kind of functionalities. The actual field test is carried out when the verification has taken place and developers can be sure that the functions behave as desired. Virtual prototypes have the benefit to completely test the functions in a risk-less manner for all possible scenarios and parameters, gaining confidence of the functionality leads to the final deployment on the prototype vehicle.

Generally, in safety-critical systems, failure can have severe impacts, can lead to significant damage to property or loss of life. These systems are very complex, millions of lines of code, highly integrated and connecting systems of systems. It must be ensured that these systems operate under any condition in a safe way while complying with up-to-date safety standards, therefore, software must not harm the world. To fulfill these requirements also during the development phase with traceability, validation and verification, model-based design methodologies including tool-chains are needed. This section examines the application of model-based SW design framework in context to complex battery management algorithms and battery safety critical applications. Hereby, the current automotive framework is introduced and compared with to aerospace industry. As in the early 2000s and Toyotas' unintended acceleration and software safety scandal, resulting in 89 deaths and 57 injuries [468]. The report from NASA and Carnegie Mellon states that bugs, not following the MISRA Guidelines completely (11 rules out of 144) have to led to 80.000 violations, such as the usage of 11528 global variables with different types, and can alter the value and uninitialized variables leading to spaghetti code. The complexity level exceeds 50 and up to 146, which implies that the application is difficult to test and that changes may result in bad fixes. Additionally, concurrency bugs and race conditions results in different timings and defects. This example clearly shows that conventional tool-chains reached the tipping point and new methods must be used. Already in the early 90s, model-based methods have been deployed for automotive control software, and during the years, this has been extensively be adopted for safety-critical embedded applications and where time to market is critical. Besides the behavioral aspects, developers also design the architecture and structure of the software. The classical model-based design comes from a mathematical method to solve problems when dealing with complex systems. However, nowadays, model-based design / software development is often used as visual tool (such as UML) to describe control flow or the systems architecture. Especially in aerospace and lately in automotive industry, the systems consists of safety-critical software, which puts higher constraints on the model-driven development, such as determinism and freedom from ambiguity [469]. However, the key challenges for safety-critical software development are better control over defect life-cycle, SW testing, legacy code merging, life-cycle models, automation, the need for certification and its requirements, coding standards which boils down to the tool effectiveness and support of determined and avoidance of ambiguity errors. As can be seen, testing alone is not the only answer to reliable systems. As example and mentioned before, during 2009 and 2011, Toyota had multiple related recalls, caused by unintended acceleration. US authorities investigated over 18 month and came to the conclusion that unsafe casts, ignoring return errors and

violation of the coding standard lead to fatal car accidents. Therefore, it is absolutely crucial to use an effective tool, which also has the ability to detect the maximum number of violations possible, while decreasing false-positive detections (MISRA C:2012).

The software development failures are indebted by the exclusive single core hardware usage resulting in a forced single-core specific software design. Also, this makes new projects dependent in the sense that a lot of legacy code has to be reused, and hence new SW component development cannot be stopped for re-engineering since the SW may have hundreds of millions of code lines. At the same time, when shifting to a new hardware platform (e.g. multi-core), not re-engineering won't cause any speedup, even less.

8.3.1 TargeLink Ecosystem

TargetLink is an production code generator which generates highly optimized C code, based on the graphical modeling tool MATLAB/Simulink/Stateflow which allows to convert the model into C code. However, model-based design for functional safety applications (ISO26262-6) requires to execute back-to-back testing. In this step the translation by the code generator from the model to the C code on a semantic level must be verified. It must be made sure, that the implementation on the target controller behaves the same way as the model and both produce the same output with the same input. In the first place, the model has to be verified against its requirements (MIL), only then the C code is tested in regards to rounding errors or stability. Then for both, model and C code a high degree of coverage shall be reached. As can be seen in the Targelink reference work-flow, testing becomes the most important and time-consuming (expensive) work for model-based design implementations. The fact, that the C code does not necessarily reflect the model behavior was one fact to think about a better way to develop a control design code. Secondly, Matlab/Simulink is designed to use graphical programming to develop control functions in a simulation environment, and not for embedded C programming. Also after having personal discussions with different automotive software developers it is clear that a new way to handle SW design must be considered.

8.3.2 SCADE - An Aerospace Solution for Highest Level of Safety in SW Development

There is a huge difference in quality consciousness between long time safety critical applications, such as the aerospace industry, and the traditional automotive industry with less focus on safety. Hence, it is reasonable to think that there can be a lesson learned to deliver safety-critical SW in the automotive industry. First, aerospace requirements are rather rigorous and changes are executed in a strict manner. Additionally, aerospace



Figure 8.3: Required tools in TargetLink Ecosystem - the industry's traditional model-based design toolset

projects take advantage of team-based tools, for traceability reasons for code changes, SW-issue tracking and reviews. Also following strictly the coding standards and code analysis to managing code complexity is a proven working method to improve SW quality. Regarding testing, advanced methods like MC/DC are used. A big difference also lies within the used toolchain, modern IDE are used for model-based design and verification using formal methods. In comparison to Matlab solution, SCADE was developed and defined for safety objectives, it does not allow for dangerous constructs (dynamic memory allocation, goto, state machines with intermediate states, global variables, unbounded loops,...) where inputs are read at the beginning of the execution cycle and delivered at the end, in between read/write is not possible, inputs and states are stable [470]. In comparison to automotive, avionics software engineering has over the years developed tools and methods to reduce the time required for testing and certification.

8.3.3 Comparison: Traditional vs Full Safety Critical Toolchain

To put it in exaggerated terms, software is consuming the world. In terms of cycle of innovation, a car platform lasts 10-15 years, a car development process 4-5 years, but in-car SW will be deployed every day/week which gives an innovation cycle with is by far much faster than any other technology. Considering that the new vehicles already (w/o AD or electrification SW) have 100 Mio lines of code, the traditional SW tools

must be questioned, as the automobile industry, recalled more vehicles than ever before due to SW bugs. Looking at the traditional toolchain ecosystem from Mathworks and its Matlab/Simulink tool, which is purported to be a simulation tool for control laws and not developed to generate production-ready code for safety critical applications, it shall be questioned if this way of working will fulfill the future needs in agile SW development. With Simulink, the code must be verified on model and code level, as the Simulink/TargetLink code does not behave exactly the same way nor is guaranteed to do so. This means, besides TargetLink for code generation, also back-to-back testing tool between MiL and SiL is required. Additionally, SW tools for Model guideline checking, model/code coverage, model-based testing, and others. On the other side, SCADE is a formally defined language for aerospace industry with functional safety in focus fulfilling all safety standards while keeping code efficient and portable, while the TargetLink Ecosystem is not. Additionally, as automotive embedded controller go towards multi-core platform, SCADE has been extended in order to meet multi-threaded systems using tread synchronization applying a specific communication protocol. The overall product platform includes synchronized functional and architectural modeling and testing. Therefore, SCADE toolchain is used for SW architecture and code development.

8.4 A Model-Based Approach for Multi-Core - Multiple Battery Cells Estimation

The aim of multi-core target platforms aims to provide additional computing power to fulfill the needs for advanced control strategies for new complex battery electric powertrains, advanced driver assistance systems. At the same time, legacy code must be migrated to the new hardware platform, which was designed for single-core systems and must guarantee the same performance and robustness. As introduced earlier, there are many algorithms to estimate the internal states of battery cell. While the literature focus on proving the estimation method on a single cell, is rarely discussed how to scale these algorithms in a way that they can be applied to all individual cells in a battery pack (up to several hundred cells). As OEMs using low cost embedded controller as cost savings potential for BMS systems, (Rothgang prices per system), and the fact that all the introduced algorithms demand high computational capability of the ECU, and hence it is impossible for currently on the market used systems to monitor all individual cells with high accuracy in the system. However, premium brand manufacturers looking more and more into a paradigm change resulting into a complete change rather than small steps of single core resource-constrained embedded systems. A survey shows, that 58% of companies consider multi-core processors due to the fact that an increase in performance is the reason. 92% claim, that their business application have high performance requirements, such as strict real-time property, highest reliability and safety, integration of multiple functional SW components on one ECU leading to reduced number of distributed ECUs in the vehicle (e.g. BMS application) [471]. for example, one attempt is based on the

assumption that the entire cells of the battery pack are considered as one big battery cell, and rather having individual state and parameter calculations, here the state of the entire pack is estimated. However, that is a rather extreme simplification with many drawbacks, such as capacity determination, or series connected cells reaching first full/empty state limiting the performance. Further improvements consider to group similar cells and run the state estimation on these groups. Some ideas propose techniques that estimate SoC for all the cells within the battery pack by estimating a reference cell and calculate for each other cells the deviation/mean change. So basically, the efforts focus on reduction of calculation cost by simplifying algorithms or reducing complexity or considering only a group of cells. In this work, we consider a different approach, which does not compromise on the accuracy and complexity of the algorithms, but rather utilizes the hardware. This requires a paradigm change for the software development for embedded systems, now considering parallel execution and timing conflicts. The challenges for the classical parallel code development and its implementation lies on the division of problem into different tasks which can be executed simultaneously, with least dependencies amongst the different parallel branches and reduced communication amongst the cores to guarantee for synchronized execution.

In [472] the work did not use the automotive industry's most widely used model-based development toolset (Matlab) because it includes a code generator that is not compliant with the safety standard governing the embedded software development process. Model-based development has greatly improved the quality and time-to-market of safety-critical automotive systems such as BMSs by enabling a graphical model to replace software architectural and unit design during the development process. The behavior of the model can be simulated and immediately viewed its results without the need for physical hardware. This makes it possible to gain critical insights early in the design process and rapidly improve the model's performance. Later, the model is used to automatically generate the embedded code, eliminating the need for manual coding and enabling the engineers to test their model instead of the code. This eliminates all code-based verification of the application. Using Simulink tools, each of these iterations would require back-to-back testing and functional testing phases to verify that the generated code implements the requirements correctly and matches the behavior of the model used in the design process. By creating, running and validating functional test cases in the SCADE Test environment for the SCADE model early in the design process developers can detect errors in their specification sooner, instead of during integration testing. When the cross-functional teams feel the model is ready, they use the SCADE Suite KCG code generator to generate C source code for the target environment. This code generator has been qualified for developing ISO 26262-compliant applications up to ASIL D, the highest safety requirement for automotive applications. More and more multi-core embedded controllers come into the market to support computational hungry applications by keeping the power consumption equal, however, no model-based multi-core generator is commercially available. On the other hand beta multi-core SCADE code generator is supporting multi-core architecture. In [472] the first approach in BMS SW development was carried out.

Battery Management System - Software Architecture

The detailed functional and architectural system modeling and verification was done with the help of SCADE Architect, a system design environment providing full ISO26262 support, following a set of requirements and proven practices and therefore the battery management must fulfill functional safety standard / goals to react accordingly to its inputs to keep the battery system safe for all conditions and operation. The derived safety concept applies redundant measurement and control to prevent failures and safety goal violations. The reference design for a BMS embedded system consists of 2 micro-controller, where the secondary controller has lower power, less complex firmware, precise and more robust power sources and used to execute the safety monitoring functionality [9]. The primary micro-controller, higher-power, more expensive in comparison to the secondary micro-controller, is used to execute the more complex algorithms for SoX estimation, which requires more computational power and requires no / reduced safety measures (QM / ASIL-A). Both micro-controllers are connected via I²C which enables mutual supervision and option to reset a suspended controller. Instead, here, a multicore and lockstep core architecture is implemented, with fast communication interfaces, Hardware Security Module and support of safety level up to ASIL-D. Fig. 8.4 describes the overall layered

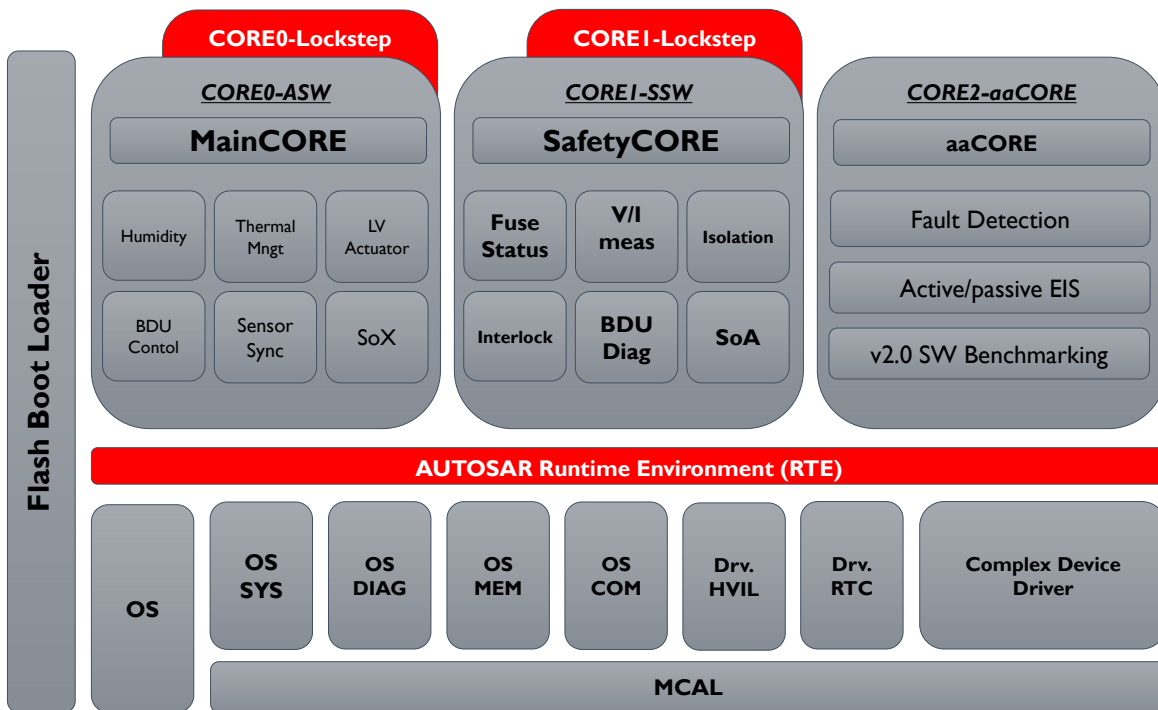


Figure 8.4: Layered Architecture of AUTOSAR with Application SW

SW including application layer (CORE0-ASW: Application SW; CORE1-SSW: Application Safety Software; CORE2-ABMA: Advanced algorithms) and basetech SW (Micro-controller Abstraction layer, ECU Abstraction Layer Complex Device Driver(CDD)) and Runtime Environment is the layer separating the Basic Software and Application Soft-

ware. The Core0-Core2 represent the TriCORE AURIX TC275TP from Infineon, with 3 processor cores from which 2 support lockstep. On CORE0-ASW, the main core, which executes the SoX functions, Core1-SSW (e.g. state- and parameter estimation), the safety core executing ASIL B /C/D functions (e.g. contactor control). The lockstep architecture controls and mitigates physical isolation, instruction-level execution diversity of 2 cycle delay [473]. The Cores0-1 with its lockstep core run the same software in parallel and compares the result and reacts if mismatches are found. CORE2-aaCORE is executing advanced algorithms, which needs extra computing power, such as model-based fault detection or further signal processing tasks for battery diagnostics. Also, CORE2-aaCORE is used for validation purpose, where new developed algorithms are flashed via over-the-air option and benchmarked against the current software. Also, this part is used for Big Data Analytics and machine-learning purposes. A local data-driven model is further trained and results are sent via GPRS to back-bone for off-line battery analysis.

8.4 A Model-Based Approach for Multi-Core - Multiple Battery Cells Estimation

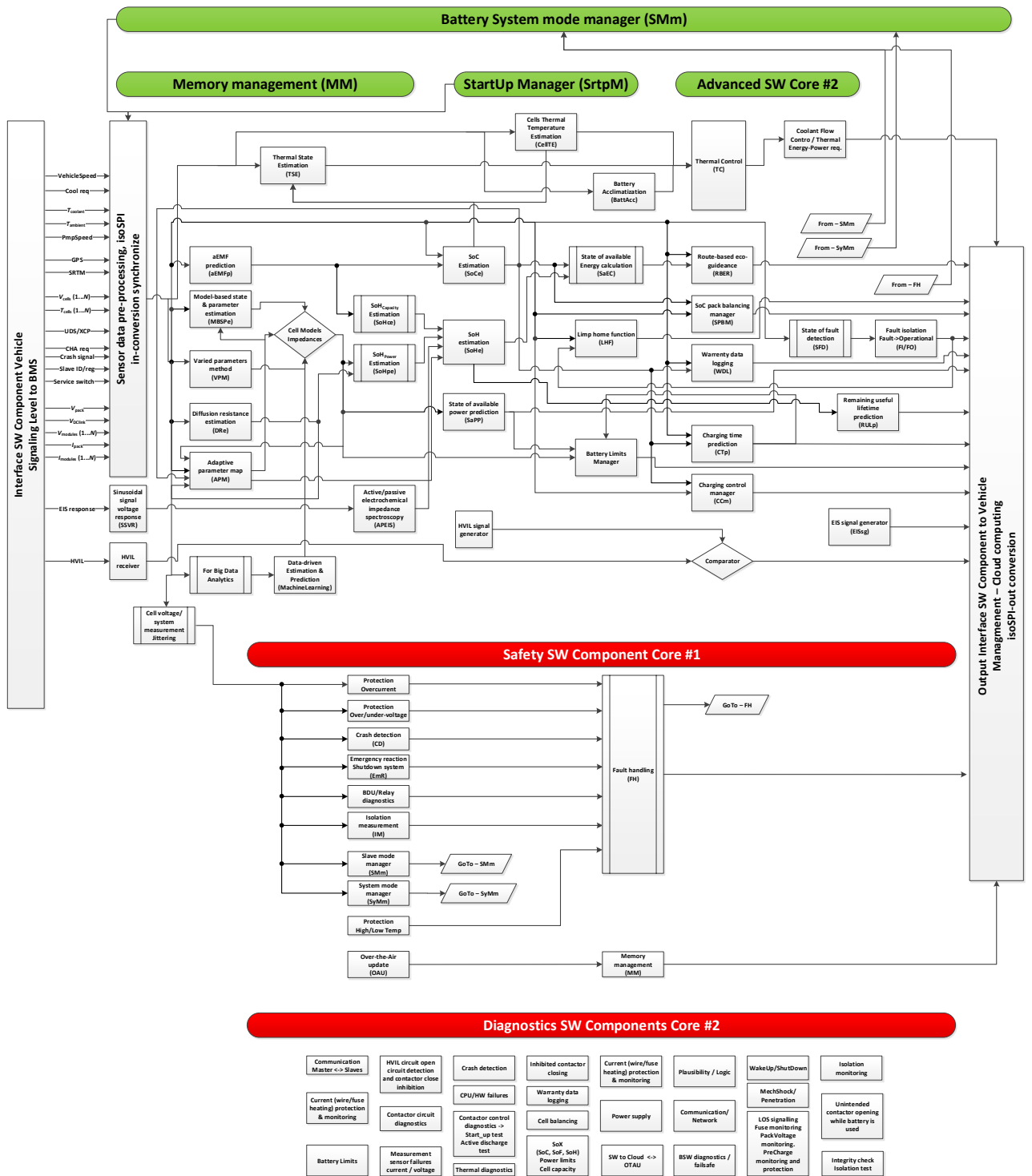


Figure 8.5: Proposed SW Functional Architecture for Battery Management System

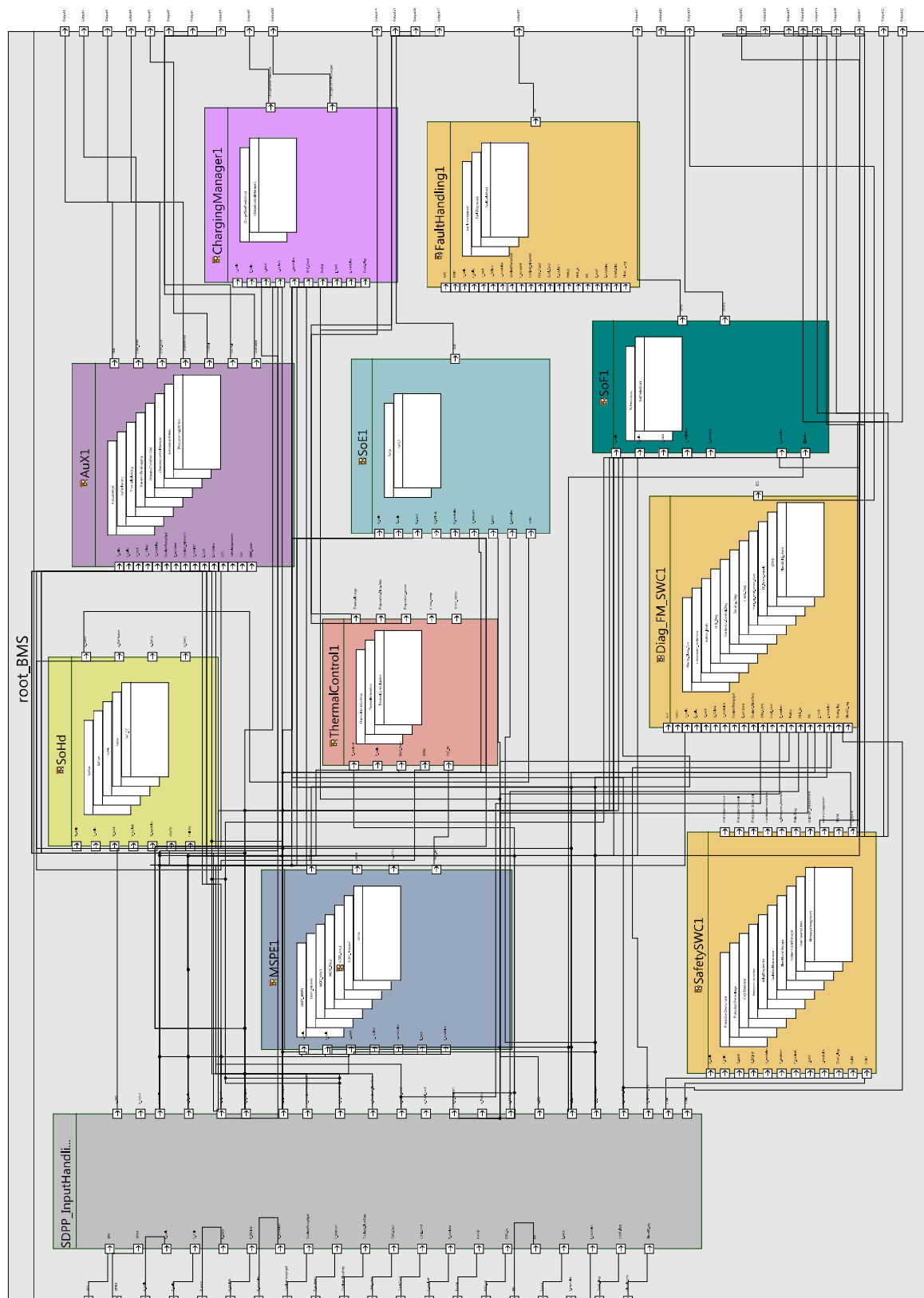


Figure 8.6: BMS functional and architectural system model in a SysML-based environment using SCADE Architect to deploy model-based system engineering

8.5 Parallel Optimization for Computational Efficiency

Parallel processing methods are described in [376], in which the unknown parameter set $\theta_1, \theta_2, \dots, \theta_{100}, \theta_n$ with assumed a priori probability for θ is provided. We distribute the processing power to allow individual cell monitoring using a conditional mean estimator, for which a filter bank of 100 instances of parallel running extended KALMAN filter are implemented with conditional state estimates $\hat{x}_{k|k-1, \theta_i}$, then the conditional mean state estimator is calculated as a weighted sum from all instances [376]. The a posteriori probability satisfying $\theta_i = \theta$ represents the weighting coefficient of the i -th state estimator. As discussed in the first chapters, Bayes' rule can be expressed for a signal model represented by an unknown parameter vector $\theta_1, \theta_2, \dots, \theta_{100}, \theta_N$ as

$$\begin{aligned} p(\theta_i|Y_k) &= \frac{p(Y_k, \theta_i)}{p(Y_k)} \\ &= \frac{p(Y_k|\theta_i)p(\theta_i)}{\sum_{i=1}^{\#instancesN} p(Y_k|\theta_i)p(\theta_i)} \end{aligned} \quad (8.1)$$

where Y_k represents the sequence of measurements, $p(\theta_t|Y_k)$ is the a posteriori probability, and $p(Y_k|\theta_i)$ the likelihood function. The bank of KALMAN filters are dedicated to recursively calculate these quantities.

$$\begin{aligned} p(\theta_i|Y_k) &= \frac{p(y_k, Y_{k-1}, \theta_i)}{p(y_k, Y_{k-1})} \\ &= \frac{p(y_k, \theta_i|Y_{k-1})p(Y_{k-1})}{p(y_k|Y_{k-1})p(Y_{k-1})} \\ &= \frac{p(y_k, \theta_i|Y_{k-1})}{p(y_k|Y_{k-1})} \\ &= \frac{p(y_k|Y_{k-1}, \theta_i)p(\theta_i|Y_{k-1})}{\sum_{i=1}^{\#instancesN} p(y_k|Y_{k-1}, \theta_i)p(\theta_i|Y_{k-1})} \end{aligned} \quad (8.2)$$

where $p(y_k|Y_{k-1}, \theta_i)$ is Gaussian distributed with mean $\hat{x}_{k|k-1, \theta_i}$ and covariance $P_{k|\theta_i}$. Hence, the parallel processing of state estimation can be split into multiple instances under the assumption that the parameter vector θ is constructed as $\theta_1, \theta_2, \dots, \theta_{100}, \theta_n$, then the conditional mean state estimate can be calculated as

$$\hat{x}_{k|k-1} = \sum_{i=1}^N \hat{x}_{k|k-1, \theta_i} p(\theta_i|Y_k) \quad (8.3)$$

where $\hat{x}_{k|k-1, \theta_i}$ represents the conditional estimate, and $p(\theta_i|Y_k)$ the conditional probabilities. As shown in 8.7, the conditional KALMAN filters with the set of parameters and $p(\theta_i|Y_k)$ are estimated using conditional innovation quantities $\tilde{y}_{k|\theta_i}$ and the covariances

$P_{k|\theta_i}$.

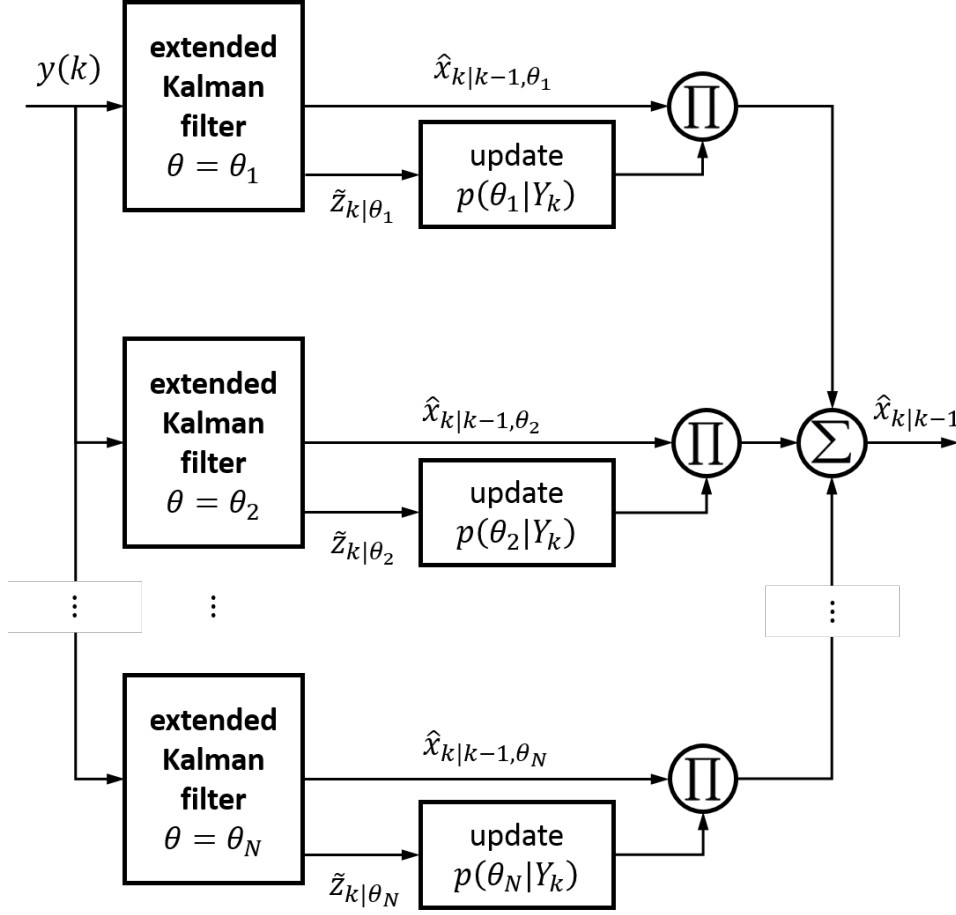


Figure 8.7: Extended KALMAN filter bank with 100 instances for parallel computing (modified from [376])

Multi-core processors can provide opportunities for embedded system developers, as many semiconductor manufacturers are now standardizing on multi-core processor technology. However, manufacturers maintaining their legacy code and put little effort in R&D for multi-core SW development. Secondly, traditional tools do not provide multi-core code generation and therefore is a challenge to efficiently utilize the new processor technology. SCADE on the other hand is a natively parallel language based on Kahn Process Networks and models, therefore, there is no control dependency only for data. It follows the formal methods, following the development flow by concise formalism for requirements, implementation of control SW, verification for increased SW reliability [474]. Also, SCADE programming language is not lacking with concurrent accesses to shared memory or aliasing [474]. Splitting the functions across the multi-core is very tricky when working on manual code development, however, SCADE offers the designer to mark operator instances to associate the threads, which can be a simple or iterated instance giving a predictable deterministic behavior and target independent. The SCADE code generator is able to check for causal independence amongst the SW units if the splitting in different threads may result in problems in terms of WECT. The automatic splitting can be replaced by

mixing explicit and implicit methods, meaning that we defined already which of the functions are dependent on each other and how to partition the model for parallel execution before they are going to be defined as parallel threads by grouping independent operators into parallel subsets. The KCG multi-core code generator prototype is based on a generic task based implementation OpenMP, in which each defined instance is mapped towards a "thread" for tasks and data parallelization [474]. KCG also checks statically if there is no dependency amongst the parallel subsets. In our case, the different KALMAN filters have no dependencies amongst each other, hence, they can be placed into the same parallel subset. The generated code structure guarantees isolation between tasks, as SCADE does not support global variables/memory access, rather uses communication channels [474]. This requires four macros as illustrated in Fig.8.8, in which the **SEND(ch)** sends on the channel (ch) the value stored in the structure fields, **RECV(ch)** receives the value from the channel (ch) and stores it within the structure fields, **DECL_SENDER(ch,t)** which declares a sender of channel (ch) of type t , and lastly, **DECL_RECEIVER(ch,t)** declares a receiver of channel (ch) of type t [474]. The communication channel (ch) is

```
KCG_DECL_RECEIVER_main_1_in_ch(main_1_in_ch_type, main_1_in_ch)
KCG_DECL_SENDER_main_1_out_ch(main_1_out_ch_type, main_1_out_ch)

{
  main_1_in_ch.Vt = V_t_EKF_MultiCoreRoot[99];
  main_1_in_ch.current = current_EKF_MultiCoreRoot[99];
  main_1_in_ch.reset = outC->init;
  KCG_CHANNEL_SEND_main_1_in_ch(main_1_in_ch);

  KCG_CHANNEL_RECV__111_main_1_out_ch(_111_main_1_out_ch);

  _493_main_1_in_ch.data.sem = CreateSemaphore(NULL, 0, 1, NULL);

  thr[0] = CreateThread(NULL, 0, &thread_0, NULL, 0, NULL);
```

Figure 8.8: SCADE multi-core code generated from v6.6 models [474]

implemented via a **struct** with data field to store semaphore or a lock. Fig.8.8 a sample shows the construct of the channel (ch) by using semaphores for synchronization purposes. The integration part is not carried out in this work, as only the idea of adaptive estimation via parallel processing is discussed. Since the multicore code generator is still in development status, the work on the bottom of the "SCADE model" block in Fig. 8.9 was executed by the ANSYS/Esterel engineers within the evaluation period (code generation). The SCADE model consists of the extended KALMAN filter for SoC estimation.

Results

To illustrate the implemented parallel optimization for an increased computational efficiency for multi-cell battery state estimation, 100 instances executing 100 extended KALMAN filter in parallel processing framework were used, as described in Section 8.5. Fig.8.10a shows in the left side the DOS command-line interpreter within Windows operating system. Here the mean step duration time for each instance is illustrated. In the

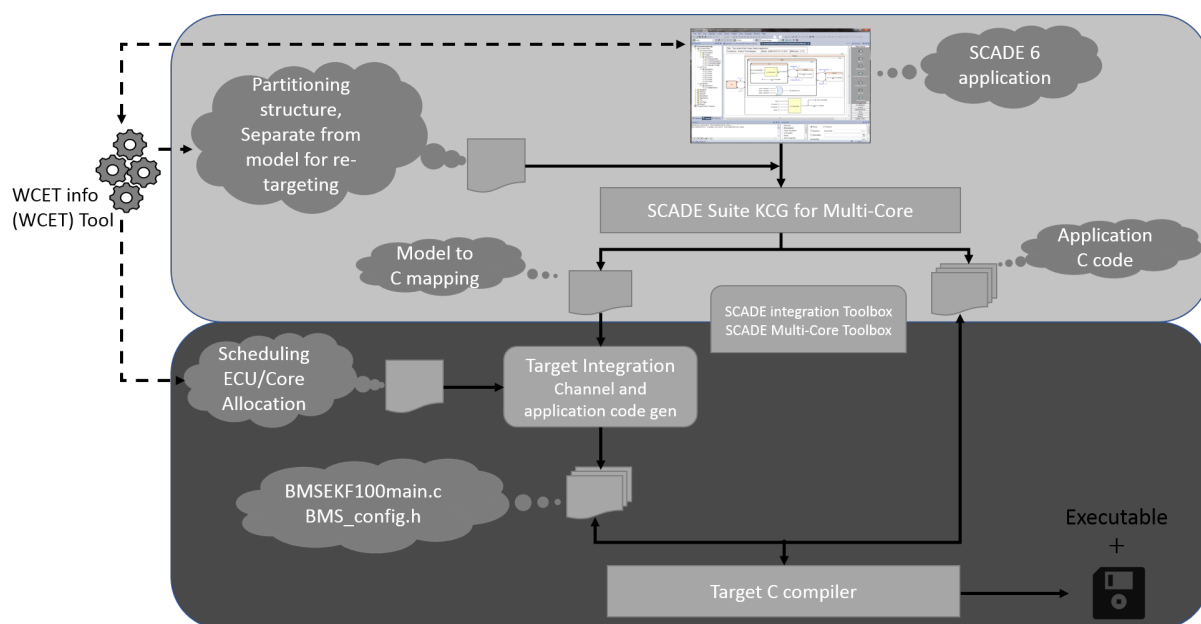
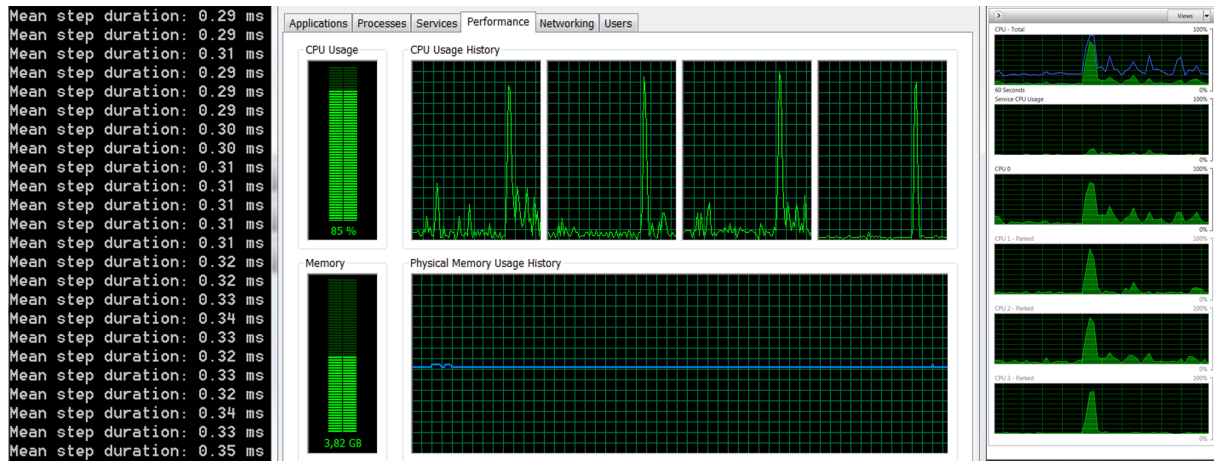


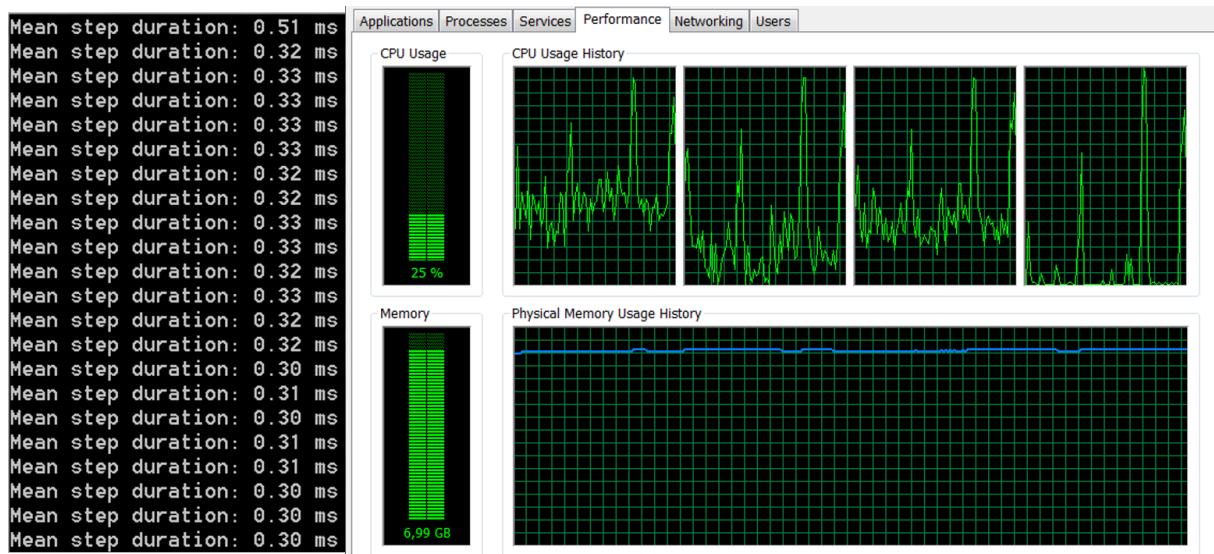
Figure 8.9: SCADe multi-core code generation for prototype v6.6 models (modified from [474])

left side of Fig.8.10a, the performance is recorded during the execution of the multicore-application (.exe). It can be noticed, that the individual Core (CPU0-CPU3) usage is utilized to a maximum, proving that the instances are executed in parallel.

On the other side, in Fig.8.10b is the sequential program is executed and its utilization shows a different result. Since the sequential program is executed not in parallel, the overall CPU usage is around 25%, and the multi-core at around 85%. The allocation on the Cores is handled by the operating system. As a conclusion, it is of great improvement to utilize the new multi-core automotive controllers, if and only if the model-based software design takes into account the parallelization into independent instances. This enables multi-cell monitoring, instead of tracking a subset of all cells in the pack, all the cells can be monitored enabling an even better understanding of the battery states and parameters.



(a) CPU load for multi-core optimized extended KALMAN filter bank



(b) CPU load for sequential executed extended KALMAN filter bank

Figure 8.10: Comparison of multi-core optimized BMS SW vs. sequential executed SW.

9 Conclusions and Outlook

To improve the filtering accuracy, recent publications show, more and more filters/machine learning algorithms appear in model-based or data-driven battery monitoring without giving quantitative or qualitative benchmarking. Therefore, this work evaluates most modern filter algorithms for state- and parameter estimation based on the defined benchmark criteria, such as filter performance or their computational comparison vs. the reference filter (EKF with Model 1). A toolbox including many stochastic filters and machine learning approaches for realistic assessment was developed for state- and parameter estimation. As industrial example of application, the statistical filters are used for on-line model-based monitoring of lithium-ion batteries, including 4 different ECM's. For this, the real-time capabilities for running on "low-cost-BMS hardware must be considered as calculations are very complex. The UKF for example, to calculate the new set of sigma points, the square root matrix of the posterior covariance for each sample. Therefore, the SRUKF takes advantage of the fact that updates are applied to Cholesky factors rather the full matrix given the possibility to immediately process the SR matrix, on the premise to improve calculation time without losing accuracy. As numerical methods are considered to be stable, for the case that their results for extended time steps convergences towards the correct value. The quantitative evaluation methodology additionally investigates the filters stability and efficiency, by means of sensitivity to noise, initial calibration, system state mutation or the capability to prevent divergence for non-stationary systems. The qualitative evaluation takes into account the implementation effort (tuning for optimization, complexity in integration,...) The recursive Bayesian estimation and Artificial Intelligence Toolbox for battery monitoring consists of the introduced filters. The machine learning algorithms were implemented in TENSORFLOW, an open source machine learning framework to support for machine learning and deep learning algorithms, which was connected and called from MATLAB.

It was shown during the BMS evaluation in Chapter 3, that BMS application algorithms alone don't make a functional BMS, a lot of efforts must take into account to fulfill ISO 26262-6:2011 product development at software level, including basetech. However, there are still many misconception about safety standards requirements in BMS development, not to least as its a fairly new domain and still today, first code is written and processes with their certification documents can be added later causing projects to fail. Therefore, the work discusses about safety-critical, model-based battery management system design which address the new challenges in automotive SW engineering. Simulation tests will reduce bugs in the early stage and improve safety. The solution of automation of ISO26262 work-flow will be effectively implemented using model-based design including

software safety requirements, software high-level architectural design, and integration and verification. As current state-of-the-art HIL BMS testing is still at an early stage this work promotes a novel HIL simulation battery system environment for BMS functional and non-functional testing on a commercial HIL simulator. A multi-cell electro-thermal Li-ion battery system model is implemented and coupled with an empirical aging model which was used to test commercial BMS.

Results

By means of the implemented toolbox, it was possible to provide the user a quantitative and qualitative benchmark criterion to evaluate the filter quality. The user needs to specify the system equations, while the filter library is already implement. The models from TENSORFLOW are called and executed within MATLAB.

State-of-Charge estimation

As state estimation we refer to quantities within a battery changing rather quickly, while parameters change slowly. The SoC is essential within the derived battery-models as SoC is used in many sub-functions of a BMS (e.g. power/energy prediction). As discussed during the literature review, there are poor voltage/current methods. In this work, the focus lies within model-based estimation taking into account the current/voltage behavior. We compared not only the ordinarily known EKF and UKF approaches commonly discussed in the literature, rather an extensive filter library with more than 10 filters were implemented and benchmarked using simple to more complex ECMs. It can be seen that the simple Gaussian approaches have difficulties following non-gaussian system noise, reducing the overall filter quality. It can be seen that the particle filter family has the highest accuracy also within all ECMs. Hybrid systems show similar performance.

Joint and Dual estimation

So far, we assumed that ECM model of battery cell is constant and identified. However, the system model must change during battery lifetime as degradation will cause the system model to deviate. Hence, the parameter values which change slower must be tracked. The two approaches discussed here is the joint and dual estimation. In joint estimation [2], the state and parameter vector are merged into one vector. This leads to complex, high dimensional, on-line matrix operations and the possibility of numeric conditioning [2] since battery have multi-time scale dynamics. In dual estimation, dedicated filter for state and parameter estimation are used. The overall requirements on computation can be reduced

in comparison to joint estimation, also proven in our results. The joint estimation outperforms the dual estimation in terms of accuracy. One reason is that interconnection and cross-correlation between parameter and state is lost.

Remaining useful life

The prediction of the remaining useful life is one of the most challenging task in SW development for BMS until this date. The concepts in the literature are deploying lifetime model or SoH prediction model. While the lifetime model methods completely rely on the correctness of the model itself, the data-driven SoH prediction scheme estimates the parameters (capacity, impedance...) which are used as input features to the prediction model. In this work, different deep structured learning methods were investigated, such as RNN, LSTM or CNN to understand the usage for prognostics in predictive health management. The difficulty lies within the fact that predictions shall happen constantly and direct, however, degradation of a battery takes years and aging phenomena are not seen/learned directly. However, as can be seen, the LSTM shows better performance than the conventional RNN.

Outlook

This work presents a comprehensive comparison of non-linear filtering methods. Nevertheless, there is always ongoing work in developing new algorithms, and cannot be considered within this work, so future work shall consider new breakthroughs within classical and Bayesian estimation. On the other hand, the toolbox also shows the potential of artificial intelligence within battery management systems, e.g. for predicting the remaining life of the battery. Also, as futurists proclaim, the future will be automated, digitized, and simulated, possible also in BMS. As Big Data becomes the new oil, connected vehicles and cloud computing enable remote battery telematics. As reported by BloombergNEF: “Our latest forecast shows sales of electric vehicles (EVs) increasing from a record 1.1 million worldwide in 2017, to 11 million in 2025 and then surging to 30 million in 2030 as they become cheaper to make than internal combustion engine (ICE) cars. China will lead this transition, with sales there accounting for almost 50% of the global EV market in 2025.”, it gets obvious that the expected rapid increase of electric vehicles requires battery telematics. Second enabler, batteries provide huge amount of data, which requires storage and data analytics. Only cloud services can handle the big data, as it becomes automated and affordable, which also allows more complex BMS functionality. Thirdly, data transfer from vehicles to the cloud via telecommunication channels is very expansive, but will drop in prices, as low cost CAN-to-IoT devices becomes affordable. Lastly but most important, data analytics and AI for EV batteries could not be used before, models, tools, computational power etc were not available, therefore, Big Data with new

algorithms, even optimized for online applications, will lift battery systems to an optimum and enable predictive functions. The benefits are obvious for OEMs, fleet manager or cell manufacturer as:

- Down time of fleet is reduced by optimized charging strategy.
- Fault prediction and isolation to prolong battery lifetime due to predictive maintenance.
- Fleet data accessible on one platform allows predictive future system behavior and trend analysis.
- Extends battery lifetime, as defective patterns (abnormal self-discharge, spikes,...) are detected to initiate early actions (isolation).
- Reduce battery break down by monitoring different parameters, as data analysis can take place automated on remote server sending out direct warnings
- Allows custom battery operation window for personalized battery/driving configuration -> battery size can be reduced
- Product development and time-to-market can be increased, as battery data can be used for early lifetime prediction simulations / system behavior and to build virtual models to test and validate batteries earlier.
- No other classical algorithms offer this flexibility and insight on system level allowing whitebox decision making

All this will lead to an overall cloud-based SW platform. The individual vehicles are providing the data (time-series or model parameters). After data ingestion, data is stored and processed (from dealers, service & maintenance departments, live traffic and geo-location) to the data hub, executing big data analytics and machine learning for automated battery model learning, classification of batteries (based on user behavior, aging...), generating updated or even new SW features in real-time (power & energy) or multi-time scales (aging), detecting anomalies and more. The corresponding newly features or calculated parameters are send back to the vehicles resulting in a better driving and user experience, optimized battery power and energy characteristic. Also another trend to be foreseen is, that OEMs move from distributed electrical architecture towards centralized, computing platform. Hence, future BMS function will be integrated onto the more powerful central processor taking care of the propulsion system management, such as machine learning methods for BMS.

A Appendix

A.1 BMS HW evaluation

A.1.1 System #1

	Available	Not Available	Comments
System			
wake up time		x	unknown
After run time		x	unknown
Active balancing	x		not acceptable
passive balancing	x		
Continous balancing	x		
Balancing capacity	x		100mA
Time to balance 1% per cell	x		5h
Cell Level			
Cell Voltage measurement	x		
Voltage resolution	x		2mV (acceptable)
Voltage measurement error	x		$\pm 3,2\text{mV}$ (acceptable)
Sampling rate (Voltage)	x		40ms (acceptable)
Transmission rate	x		20ms
Cell Current measurement		x	still acceptable
Cell internal resistance measurement	x		
Cell temperature measurement	x		1 sensor (per 12 cells? clarify)
Module Level			
module voltage measurement		x	acceptable
module current measurement		x	acceptable
module temperature measurement		x	
Temperature sensor supported	x		NTC
Pack Level			
Pack voltage measurement	x		
Voltage range	x		500V
Voltage resolution		x	unknown
Voltage measurement error		x	unknown
Pack Current measurement	x		
Current Sensor type	x		dual(shunt/Hall)
Dual current sensor	x		
Current measurement range	x		-1523A to 2076A
Current measurement resolution	x		30mA
Current measurement error	x		0.5%/0.8%
Error at 160 KW	x		1.280 W
Sampling rate	x		1ms
Transmission rate	x		20ms(acceptable)
Communication types	x		CAN -LIN - Diff SPI (1Mbps)
Default speed	x		500kbps
Speed modifiable	x		
numbers of CPUs	x		1 acceptable
CPU 1 type	x		Dual core 200MHz
Default CPU load	x		70-75% (acceptable)
Customizable	x		
Redundant	x		Current S main controller safely functions. Temp no V, no relay
Saftey seperated		x	Have separeate core for saftey
BDU control	x		CAN + Hard wired
EMC filter	x		
Leak detection Active/Passive	x		passive (acceptable)

Figure A.1: HW evaluation of System #1

A.1.2 System #2

	Available	Not Available	Comments
System			
wake up time	x		1000ms
After run time	x		100ms
Active balancing	x		not acceptable
passive balancing	x		
Continous balancing	x		
Balancing capacity	x		acceptable
Time to balance 1% per cell	x		10h (acceptable)
Cell Level			
Cell Voltage measurement	x		
Voltage resolution	x		1mV (acceptable)
Voltage measurement error	x		±4mV (acceptable)
Sampling rate (Voltage)	x		30ms (Acceptable)
Transmission rate		x	unknown
Cell Current measurement		x	still accpetable
Cell internal resistance measurement		x	still accpetable
Cell temperature measurement		x	
Module Level			
module voltage measurement		x	acceptable
module current measurement		x	acceptable
module temperature measurement		x	
Temperature sensor supported	x		NTC
Pack Level			
Pack voltage measurement	x		
Voltage range		x	unknown
Voltage resolution		x	unknown
Voltage measurement error		x	unknown
Pack Current measurement	x		
Current Sensor type	x		shunt (acceptable)
Dual current sensor	x		Optional(HALL)
Current measurement range	x		±1500 A
Current measurement resolution		x	unknown
Current measurement error	x		1% from 0-300A 5% from 300 to 1500A (unacceptable)
Error at 160 KW	x		8.000W(unacceptable)
Sampling rate	x		2ms
Transmission rate		x	unknown
Communication types	x		CAN 4 channels (Vehicle, BJB, BMM, Debug) acceptable
Default speed	x		500kbps
Speed modifiable		x	can be studied if can be modifiable
numbers of CPUs	x		1 acceptable
CPU 1 type	x		Infineon MCU MCAL
Default CPU load	x		75%(acceptable)
Customizable	x		No Samsung, third party (acceptable)
Redundant		x	
Saftey seperated		x	Still acceptable
BDU control	x		Via CAN + hard wired
EMC filter		x	unknown
Leak detection Active/Passive		x	unknown

Figure A.2: HW evaluation of System #2

A.1.3 System #3

	Available	Not Available	Comments
System			
wake up time	x		70-100ms
After run time	x		<10ms
Active balancing	x		not acceptable
passive balancing	x		
Continous balancing		x	
Balancing capacity	x		300mA
Time to balance 1% per cell	x		time not specified
Cell Level			
Cell Voltage measurement	x		
Voltage resolution		x	1mV
Voltage measurement error	x		3mV (Acceptable)
Sampling rate (Voltage)	x		not certain can be <10ms
Transmission rate	x		up to 10ms
Cell Current measurement		x	still acceptable
Cell internal resistance measurement	x		still acceptable
Cell temperature measurement	x		2 NTC per 12 cells
Module Level			
module voltage measurement	x		
module current measurement		x	acceptable
module temperature measurement	x		
Temperature sensor supported	x		NTC
Pack Level			
Pack voltage measurement	x		
Voltage range	x		540V
Voltage resolution	x		0.5V
Voltage measurement error	x		1%
Pack Current measurement	x		
Current Sensor type	x		shunt + hall (acceptable?)
Dual current sensor	x		unacceptable solution
Current measurement range	x		-1500A - +1500A
Current measurement resolution	x		16bits
Current measurement error	x		<0,5%
Error at 160 KW	x		800W
Sampling rate	x		1ms
Transmission rate	x		10ms
Communication types	x		acceptable
Default speed	x		
Speed modifiable	x		
numbers of CPUs	x		
CPU 1 type	x		
Default CPU load	x		
Customizable	x		
Redundant	x		
Saftey seperated	x		
BDU control	x		
EMC filter		x	still acceptable
Leak detection Active/Passive	x		acceptable

Figure A.3: HW evaluation of System #3

A.1.4 System #4

	Available	Not Available	Comments
System			
wake up time	x		<1000ms
After run time	x		<100ms
Active balancing	x		not acceptable
passive balancing	x		
Continuous balancing	x		
Balancing capacity	x		100mA
Time to balance 1% per cell	x		5h
Cell Level			
Cell Voltage measurement	x		
Voltage resolution	x		0,1mV
Voltage measurement error	x		2mV
Sampling rate (Voltage)	x		500us
Transmission rate	x		100ms
Cell Current measurement		x	still acceptable
Cell internal resistance measurement	x		
Cell temperature measurement	x		3 to 5 sensor
Module Level			
module voltage measurement		x	can be done
module current measurement		x	can be done
module temperature measurement	x		3 to 5 sensors
Temperature sensor supported	x		NTC (open to other types)
Pack Level			
Pack voltage measurement	x		
Voltage range	x		0 - 500V
Voltage resolution	x		<1V
Voltage measurement error	x		0.5 to 1%
Pack Current measurement	x		
Current Sensor type	x		Dual(Hall and Flux)
Dual current sensor	x		
Current measurement range	x		±600 A (High accuracy) ±1200 A Total range
Current measurement resolution	x		32 bits (100uA)
Current measurement error	x		0.5%
Error at 160 KW	x		800 W
Sampling rate	x		1 ms(1kHz)
Transmission rate	x		10ms
Communication types	x		CAN – Diff 2 wires
Default speed	x		500kbps
Speed modifiable	x		
numbers of CPUs	x		2 safety + main dual core
CPU 1 type	x		dual core
Default CPU load	x		TBD, at least 50% available
Customizable	x		
Redundant	x		current controllers, ADC, many more
Safety separated	x		
BDU control	x		Hard wired
EMC filter		x	unknown
Leak detection Active/Passive		x	unknown

Figure A.4: HW evaluation of System #4

A.1.5 System #5

	Available	Not Available	Comments
System			
wake up time	x		1000ms
After run time	x		<10ms
Active balancing	x		not acceptable
passive balancing	x		
Continous balancing	x		
Balancing capacity	x		100mA
Time to balance 1% per cell	x		5h
Cell Level			
Cell Voltage measurement	x		
Voltage resolution	x		2mV(acceptable)
Voltage measurement error	x		±4mV (Acceptabel)
Sampling rate (Voltage)		x	unknown
Transmission rate	x		100ms
Cell Current measurement		x	still acceptable
Cell internal resistance measurement		x	still acceptable
Cell temperature measurement	x		4 NTC per 16 cells
Module Level			
module voltage measurement		x	accpetable
module current measurement		x	acceptable
module temperature measurement	x		
Temperature sensor supported	x		NTC
Pack Level			
Pack voltage measurement	x		
Voltage range	x		450V
Voltage resolution	x		1V
Voltage measurement error	x		1.5% (typ 0.1 V)
Pack Current measurement	x		
Current Sensor type	x		shunt (non dual, acceptable)
Dual current sensor		x	redundant measurement ONE shunt
Current measurement range	x		-500A to +500A (maybe acceptable)
Current measurement resolution	x		exact value unknown
Current measurement error	x		1%(opt. 0,35%, initial 0,1%)
Error at 160 KW	x		1.600W
Sampling rate	x		1ms
Transmission rate	x		20ms
Communication types	x		CAN (acceptable)
Default speed	x		500kbps
Speed modifiable	x		
numbers of CPUs	x		2 CPUs (1 for saftey)
CPU 1 type	x		Freescale(150MHz)
Default CPU load	x		40% (30% reserved for H level SW)
Customizable	x		
Redundant	x		2nd controller
Saftey seperated	x		
BDU control	x		hard wired
EMC filter		x	still acceptable
Leak detection Active/Passive	x		passive (acceptable)

Figure A.5: HW evaluation of System #5

A.1.6 System #6

	Available	Not Available	Comments
System			
wake up time	x		200ms (without pre-charging)
After run time	x		Can be calibrated
Active balancing	x		not acceptable
passive balancing	x		
Continuous balancing	x		
Balancing capacity	x		balancing current= 200mA per cell
Time to balance 1% per cell	x		2.5h
Cell Level			
Cell Voltage measurement	x		
Voltage resolution	x		12bit($\approx 0.3\text{mV}$)
Voltage measurement error	x		5mV(w/o EOL calibration); +2mV with EOL calibration
Sampling rate (Voltage)	x		100ms (Acceptable, can be made faster)
Transmission rate		x	might be acceptable?
Cell Current measurement		x	still acceptable
Cell internal resistance measurement	x		By core function
Cell temperature measurement	x		up to 6 NTC's per module
Module Level			
module voltage measurement		x	acceptable
module current measurement		x	acceptable
module temperature measurement	x		up to 6 NTCs per module (there is no difference between cell and module temperature)
Temperature sensor supported	x		NTC
Pack Level			
Pack voltage measurement	x		done on BCU
Voltage range	x		$\leq 1000\text{V}$
Voltage resolution	x		$\approx 0.1\text{V}$
Voltage measurement error	x		$< \approx 3\%$ (Acceptable)
Pack Current measurement	x		
Current Sensor type	x		supported sensor types : CAN based shunt and Hall sensors; Hall sensors read by ADC
Dual current sensor	x		there can be more than one sensor in the system. Also, Hall sensors that provide two different ranges in one sensor are supported (e.g. LEM)
Current measurement range	x		Depending on battery system, the matching sensor is selected
Current measurement resolution	x		Depending on current sensor, typically $< 10\text{mA}$ (acceptable)
Current measurement error		x	unknown
Error at 160 KW		x	unknown
Sampling rate	x		typically 10ms
Transmission rate		x	unknown
Communication types	x		To modules: -CAN (proprietary bus based on battery monitoring chip) To vehicle : CAN -Flexray (optional), options acceptable
Default speed		x	? Unknown
Speed modifiable		x	? Unknown
numbers of CPUs	x		1 (single core; multi core in the future)

Figure A.6: HW evaluation of System #6

A.1.7 System #7 (Lithium Balance)

	Available	Not Available	Comments
System			
wake up time	x		250ms+ precharge
After run time	x		<10ms
Active balancing (XXXXXXXXXXXXXXXXXXXX-XXXXXX)	x		not acceptable
passive balancing	x		
Continuous balancing	x		
Balancing capacity	x		200mA
Time to balance 1% per cell	x		2,5h
Cell Level			
Cell Voltage measurement	x		
Voltage resolution	x		0,1mV
Voltage measurement error	x		±1,5mV
Sampling rate (Voltage)		x	unknown
Transmission rate	x		2ms
Cell Current measurement		x	still acceptable
Cell internal resistance measurement	x		
Cell temperature measurement	x		1 sensor per cell
Module Level			
module voltage measurement	x		not perfect but acceptable
module current measurement		x	acceptable
module temperature measurement	x		
Temperature sensor supported	x		NTC
Pack Level			
Pack voltage measurement	x		
Voltage range	x		0-1000V
Voltage resolution	x		1V
Voltage measurement error	x		± 1V
Pack Current measurement	x		
Current Sensor type	x		Hall(LEM) Dual
Dual current sensor	x		
Current measurement range	x		1000A
Current measurement resolution	x		12 bit
Current measurement error	x		1%
Error at 160 KW	x		1.600W
Sampling rate	x		10ms
Transmission rate	x		10ms
Communication types	x		CAN Diff-isoSPI-12c
Default speed	x		250kbps
Speed modifiable	x		up to 1Mbps
numbers of CPUs	x		1 CPU dual core 90 degrees 160MHz
CPU 1 type	x		Dual core Arm based
Default CPU load	x		60-70%
Customizable	x		
Redundant	x		Other systems
Safety separated		x	still acceptable
BDU control	x		Hard wired
EMC filter		x	unknown
Leak detection Active/Passive		x	unknown

Figure A.7: HW evaluation of System #7, off-the-shelf product from Lithium Balance

System A

Rating	Explanation
+	Exceeds automotive standard
/	Automotive standard
-	Does not fulfill automotive standard

	Available	N/A	Notes	Rating
State of Charge¹				
Ampere-hour counting	X			/
OCV-based estimation	X			/
- Estimation of the battery OCV after current interruption	X			/
Model-based estimation	X			/
- Electrical models	X			/
- Electrochemical models		X		-
Impedance-based estimation (dependence of some battery impedance parameters on the SoC)		X		-
Estimation based on static battery characteristics		X		-
Estimation using fuzzy logic and machine learning		X		-
- Fuzzy Logic		X		-
- Artificial neural networks (ANN)		X		-
- Fuzzy based neural networks (ANFIS)		X		-
- Support Vector Machines		X		-
Estimation using special measurement techniques		X		-
Monitoring of the SOC of each individual cell in series connection	X		Only static SoC determination, used for balancing	/
Resolution			floating point, 0.1%	/
Other methods	TBD	TBD	Adaptive SoC algorithm currently under development	-

¹The battery SoC can be employed in the figurative sense as a replacement for a fuel gauge used in conventional vehicles. The SoC is basically the relationship between the residual battery capacity in its present state (Cr) and total capacity Cbat after completely charging the battery, expressed in a percentage: $SoC = Cr/Cbat * 100\%$.

	Available	N/A	Notes	Rating
Capacity²				
Methods based on the relationship between the ampere-hours charged or discharged from the battery and voltage difference before and after the respective CHA/DIS				
- Group 1: Methods considering the change in the measured battery OCV before and after CHA/DIS	X			/
- Group 2: Methods estimating OCV change from battery terminal voltage measured under load		X		-
- Group 3: Electrical model based joint estimation of SOC and capacity	TBD	TBD	Adaptive SoC/Capacity algorithm currently under development,	-
- Group 3: Electrical model based dual estimation of SOC and capacity		X		-
Methods based on incremental capacity analysis (ICA) and different voltage analysis techniques		X		-
Other methods	X			+

²The battery capacity is a figure of merit determining the energy that is stored in the battery and is available for usage when the battery is fully charged. The capacity of the particular battery or cell in a new state is defined by the battery or cell design and varies only slightly for individual batteries or cells of a given type because of the production tolerances. Over the battery lifetime, however, the capacity changes considerably due to aging processes. Therefore, the present capacity of the battery must be determined by the monitoring algorithms.

	Available	N/A	Notes	Rating
Impedance parameters				
- Group 1: Methods based on impedance spectroscopy (active (A)/passive (P))		X		-
- Group 2: Methods based electrical models (How many RC elements)		X	2 RC Elements	+
Kalman-Filter based method		X		-
Least-squares based method		X		/
Non-recursive estimation (based on Least-squares)		X		-
Search algorithm		X		-
Other adaptive filters and observers		X		-
- Group 3: Methods based on electrochemical model				-
Least-squares based method		X		-
Gradient-based recursive methods		X		-
Multi-rate particle filter		X		-
- Group 4: Other methods	X		Impedance parameters estimation method as described in "SoH Group 3"	+

	Available	N/A	Notes	Rating
State of Health³				
- Group 1: Methods estimate capacity as stated above	X			+
- Group 2: Methods estimate impedance as stated above		X		-
- Group 3: Methods based on the measurement of the absolute voltage or voltage drop when a certain load is applied.	X			+
- Group 4: Methods based fuzzy logic method is employed. As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used.		X		-
- Group 5: Instead of estimating battery characteristics, conditions such as temperature, number of load cycles, accumulated ampere-hours) are observed over the battery lifetime	X		"Combination of - prediction based on off-line determined aging data and load cycles monitoring - on-line feedback correction based on capacity measurement and impedance measurement is used."	/

³The understanding is that the battery SoH is 100% when the battery is new and 0% when the capability of the battery to store energy or provide power decreases to a certain minimum level. In general, as indicators of the energy and power capability of the battery, the battery capacity (SOHc) and impedance (SOHp), respectively, are evaluated.

Group 3: The approaches are very similar to the estimation of the battery resistance; therefore, the result reflects the power capability of the battery (SoHp).

Group 4: As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used. The resulting SoH does not represent the change in the particular battery characteristic or capability, but the estimation of the general state (eg. NEW, AGED, DEAD).

	Available	N/A	Notes	Rating
Remaining useful life⁴				
- Group 1: Methods based on lifetime model used for SOH estimation	X			/
- Group 1: Methods based on the SOH prediction model (part 1: estimation of battery characteristics; part 2: prediction model)	X			/

	Available	N/A	Notes	rating
Available power⁵				
Methods based on a characteristic map use the static interdependence that exists among the available power of the battery, battery states and power pulse parameters		X		-
Methods based on a dynamic battery model using KF/observer	X		dynamic cell model is used	+

	Available	N/A	Notes	Rating
Auxiliary algorithm				
Data preprocessing (thermal model to identify temperatures inside cells, plausibility check of sensor data, detecting low/high performing cells)	X			/
OCV estimation/adaptation during lifetime	X			-
Adaptive characteristic maps (store parameters for algorithm initialization after wakeup)	X			/
Available Energy calculation		X		-
Cell equalization control (only during charging / discharging or combined)	X		combined, during drive, charge and sleep if required	/
Method 1: Terminal voltage based approach		X		-
Method 2: OCV based approach		X		-
Method 3: SOC based approach	X			+
Method 4: PWM (early balancing before reaching max voltage)		X		-
Data gathering for off-line diagnosis (warranty analysis)	X			/
Charge time prediction	X			/

⁴Remaining time or number of load cycles until the battery reaches a SoH of 0%.

⁵Batteries must not only deliver a certain amount of energy to the drive train during operation but also provide a certain power in various situations. The power of the battery can be limited by the voltage, current, SoC, and temperature ranges allowed for safe operation.

	Available	N/A	Notes	Rating
Basic software (safety systems that comply with ISO 26262 and IEC 61508 standards)				
AUTOSAR is considered in the developing process for defining a software architecture, software interfaces and interchange formats, which AUTOSAR version is applied?	X		AUTOSAR 4.x	/
Model-based design?	X			/
Production-ready MCAL & OS, which OS is used? Compliant to AUTOSAR and OSEK?	X		Vector AUTOSAR OS (Microsar OS)	/
OS: Are you compliant with a full pre-emptive scheduling?	X			/
OS: deterministic RTOS baseline (tasks (timing and protected), events, counters, alarms, messages)	X		Alive monitoring & program flow monitoring using Vector "SafeWatchdog" concept	/
OS: timing based task determinism (low-latency, precise timing for periodic tasks)	X			/
OS: protected memory / EEPROM/Flash memory management in compliant to AUTOSAR?	X			/
OS: Diagnostic protocols (ISO) and fault management (AUTOSAR) compliant?	X			/
OS: Multi-core features	X		"Redundant lock step used, multicore general available	+
Interpolations-/CRC,Crypto- and Watchdog routines compliant to AUTOSAR?	X			/
Inputs/Outputs (PWM, ADC, DIO, GPT...) driver compliant to AUTOSAR?	X			/
Bootloader?	X			/
Bootloader able to update itself?	X			/
Flashing over CAN? If not, which interface?	X			/
JTAG port available on pins on external connector?		X		-
Measurement- and calibration (UDS/CCP/XCP)?	X		XCP	/

Unit-testing: Interface-/Fault injection-/Resource usage/ Model vs. C-Code tests	X			/
Fixed point implementation?	X			/
Floating point implementation?	X			/
Documentation of sourcecode?		X	Available for review, but not handed out	-
Available updates / SW release (if, how often can we expect software updates?)	X			/
Application software - auto generated code? If so, how is multi-core CPU utilized	X			/
Various software components implemented with different ASIL in APP Layer (please state), which levels?	X		"QM, ASIL-A, ASIL-B, ASIL-C	+
Supply of source code		X		-
Application software: Are individual cells considered for battery monitoring, all, if not, how will be decided?	X		for balancing and cell impedance	+
Will be able to integrate own intelligent algorithms which will be run in parallel to supplier? (Benchmarking)		X		/

System B

Rating	Explanation
+	Exceeds automotive standard
/	Automotive standard
-	Does not fulfill automotive standard

	Available	N/A	Notes	Rating
State of Charge⁶				
Ampere-hour counting	X			/
OCV-based estimation				/
- Estimation of the battery OCV after current interruption	X			/
Model-based estimation				/
- Electrical models	X		Electrical model was only used with Kalman Filter based SOC estimation. As this estimation technique was not stable enough under certain conditions we decided to use a coulomb counting algorithm again.	/
- Electrochemical models		X		-
Impedance-based estimation (dependence of some battery impedance parameters on the SoC)		X		-
Estimation based on static battery characteristics	X			+
Estimation using fuzzy logic and machine learning		X		-
- Fuzzy Logic		X		-
- Artificial neural networks (ANN)		X		-
- Fuzzy based neural networks (ANFIS)		X		-
- Support Vector Machines		X		-
Estimation using special measurement techniques		X		-
Monitoring of the SOC of each individual cell in series connection	X			+
Resolution	X		Internal resolution for computation is mAs for the Ah SOC and 0,00003% for %-SOC	+
Other methods		X		-

⁶The battery SoC can be employed in the figurative sense as a replacement for a fuel gauge used in conventional vehicles. The SoC is basically the relationship between the residual battery capacity in its present state (Cr) and total capacity Cbat after completely charging the battery, expressed in a percentage: $SoC = Cr/Cbat * 100\%$.

	Available	N/A	Notes	Rating
Capacity⁷				
Methods based on the relationship between the ampere-hours charged or discharged from the battery and voltage difference before and after the respective CHA/DIS				
- Group 1: Methods considering the change in the measured battery OCV before and after CHA/DIS		X		-
- Group 2: Methods estimating OCV change from battery terminal voltage measured under load	X		This method is used to calculate the impedance of the complete battery pack and each cell.	+
- Group 3: Electrical model based joint estimation of SOC and capacity		X		-
- Group 3: Electrical model based dual estimation of SOC and capacity		X		-
Methods based on incremental capacity analysis (ICA) and different voltage analysis techniques		X		-
Other methods	X		The complete charge/discharge throughput of the cells is measured over lifetime and compared with a table for aging, which is given by the cell manufacturer.	+

⁷The battery capacity is a figure of merit determining the energy that is stored in the battery and is available for usage when the battery is fully charged. The capacity of the particular battery or cell in a new state is defined by the battery or cell design and varies only slightly for individual batteries or cells of a given type because of the production tolerances. Over the battery lifetime, however, the capacity changes considerably due to aging processes. Therefore, the present capacity of the battery must be determined by the monitoring algorithms.

	Available	N/A	Notes	Rating
Impedance parameters				
- Group 1: Methods based on impedance spectroscopy (active (A)/passive (P))	X		Passive. The ECU waits for special workloads of the battery to calculate the Ri of each cell and the pack.	/
- Group 2: Methods based electrical models (How many RC elements)	X			+
Kalman-Filter based method	X		Electrical model with 2 RC elements was used with Kalman Filter based SOC estimation. As this estimation technique was not stable enough under certain conditions it was decided to use a coulomb counting algorithm again.	+
Least-squares based method		X		-
Non-recursive estimation (based on Least-squares)		X		-
Search algortihm		X		-
Other adaptive filters and observers		X		-
- Group 3: Methods based on electrochemical model				-
Least-squares based method		X		-
Gradient-based recursive methods		X		-
Multi-rate particle filter		X		-
- Group 4: Other methods		X		-

	Available	N/A	Notes	Rating
State of Health⁸				
- Group 1: Methods estimate capacity as stated above	X			+
- Group 2: Methods estimate impedance as stated above	X		The impedance estimation is used for calculating the power of the battery pack. The calculated impedance is compared with the manufacturer tables for a new cell (dependent on temperature and SOC)	+
- Group 3: Methods based on the measurement of the absolute voltage or voltage drop when a certain load is applied.		X		+
- Group 4: Methods based fuzzy logic method is employed. As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used.		X		-
- Group 5: Instead of estimating battery characteristics, conditions such as temperature, number of load cycles, accumulated ampere-hours) are observed over the battery lifetime	X			/

	Available	N/A	Notes	Rating
Remaining useful life⁹				
Remaining time or number of load cycles until the battery reaches a SoH of 0%				
- Group 1: Methods based on lifetime model used for SOH estimation	X			/
- Group 1: Methods based on the SOH prediction model (part 1: estimation of battery characteristics; part 2: prediction model)		X		-

⁸The understanding is that the battery SoH is 100% when the battery is new and 0% when the capability of the battery to store energy or provide power decreases to a certain minimum level. In general, as indicators of the energy and power capability of the battery, the battery capacity (SOHc) and impedance (SOHp), respectively, are evaluated.

Group 3: The approaches are very similar to the estimation of the battery resistance; therefore, the result reflects the power capability of the battery (SoHp).

Group 4: As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used. The resulting SoH does not represent the change in the particular battery characteristic or capability, but the estimation of the general state (eg. NEW, AGED, DEAD).

⁹Remaining time or number of load cycles until the battery reaches a SoH of 0%.

	Available	N/A	Notes	rating
Available power¹⁰				
Methods based on a characteristic map use the static interdependence that exists among the available power of the battery, battery states (for example, SoC, temperature, and voltage), and power pulse parameters (for example, the duration of the power pulse)	X			+
Methods based on a dynamic battery model using KF/observer		X		-

	Available	N/A	Notes	Rating
Auxiliary algorithm				
Data preprocessing (thermal model to identify temperatures inside cells, plausibility check of sensor data, detecting low/high performing cells)		X		-
OCV estimation/adaptation during lifetime		X		-
Adaptive characteristic maps (store parameters for algorithm initialization after wakeup)	X			/
Available Energy calculation	X			/
Cell equalization control (only during charging / discharging or combined)	X			/
Method 1: Terminal voltage based approach	X			/
Method 2: OCV based approach		X		-
Method 3: SOC based approach	X			+
Method 4: PWM (early balancing before reaching max voltage)		X		-
Data gathering for off-line diagnosis (warranty analysis)	X		Some data of cell history is collected.	/
Charge time prediction		X		-

¹⁰Batteries must not only deliver a certain amount of energy to the drive train during operation but also provide a certain power in various situations. The power of the battery can be limited by the voltage, current, SoC, and temperature ranges allowed for safe operation.

	Available	N/A	Notes	Rating
Basic software (safety systems that comply with ISO 26262 and IEC 61508 standards)				
AUTOSAR is considered in the developing process for defining a software architecture, software interfaces and interchange formats, which AUTOSAR version is applied?	X		No Autosar in BMS implementation	-
Model-based design?	X		Model based design using MATLAB/Simulink	/
Production-ready MCAL & OS, which OS is used? Compliant to AUTOSAR and OSEK?	X		Vector OSEK	/
OS: Are you compliant with a full pre-emptive scheduling?	X			/
OS: deterministic RTOS baseline (tasks (timing and protected), events, counters, alarms, messages)	X		Tasks are using events and alarms for generating periodic events.	/
OS: timing based task determinism (low-latency, precise timing for periodic tasks)	X		A high priority task is used to provide cyclic calls.	/
OS: protected memory / EEPROM/Flash memory management in compliant to AUTOSAR?	X		"The EEPROM memory is protected with several methods: Important data is stored three times in the EEPROM Every data block is protected with a checksum Every data block can be encrypted for security reasons Every flash cell is protected by an ECC (2 bit) on controller level."	/
OS: Diagnostic protocols (ISO) and fault management (AUTOSAR) compliant?	X			/
OS: Multi-core features		X		/
Interpolations-/CRC,Crypto- and Watchdog routines compliant to AUTOSAR?		X	A proprietary interface is used.	-
Inputs/Outputs (PWM, ADC, DIO, GPT...) driver compliant to AUTOSAR?		X	A proprietary interface is used.	-

Bootloader?	X			/
Bootloader able to update itself?	X			/
Flashing over CAN? If not, which interface?	X			/
JTAG port available on pins on external connector?		X		-
Measurement- and calibration (UDS/CCP/XCP)?	X		UDS and proprietary interface using extended CAN identifiers which can be activated/deactivated	/
Unit-testing: Interface-/Fault injection-/Resource usage/ Model vs. C-Code tests	X		full development / test process according automotive spice	/
Fixed point implementation?		X		-
Floating point implementation?		X		-
Documentation of sourcecode?	X		Specifications are done using DOORS	/
Available updates / SW release (if, how often can we expect software updates?)	X			/
Application software - auto generated code? If so, how is multi-core CPU utilized	X		No multicore	/
Various software components implemented with different ASIL in APP Layer (please state), which levels?	X		Safety goals of the BMS are up to level C	+
Supply of source code		X	Source code and models can only be reviewed on site at MQ	-
Application software: Are individual cells considered for battery monitoring, all, if not, how will be decided?	X			-
Will be able to integrate own intelligent algorithms which will be run in parallel to supplier? (Benchmarking)	X		It is possible to integrate 3rd party libraries	/

System C

Rating	Explanation
+	Exceeds automotive standard
/	Automotive standard
-	Does not fulfill automotive standard

	Available	N/A	Notes	Rating
State of Charge¹¹				
Ampere-hour counting	X			/
OCV-based estimation	X			/
- Estimation of the battery OCV after current interruption	X			/
Model-based estimation	X			/
- Electrical models	X			/
- Electrochemical models	X			+
Impedance-based estimation	X			+
Estimation based on static battery characteristics	X			+
Estimation using fuzzy logic and machine learning		X		-
- Fuzzy Logic		X		-
- Artificial neural networks (ANN)		X		-
- Fuzzy based neural networks (ANFIS)		X		-
- Support Vector Machines		X		-
Estimation using special measurement techniques	X			+
Monitoring of the SOC of each individual cell in series connection	X			+
Resolution	< 0,01%		SoC of all cells is estimated in 32 Bit Float variables, allowing 1/2 ²⁸ % resolution.	/
Other methods	X		self-discharge rates into account and also considers residual current consumers in the system, such as cell monitoring electronics and/or balancing currents.	+

¹¹The battery SoC can be employed in the figurative sense as a replacement for a fuel gauge used in conventional vehicles. The SoC is basically the relationship between the residual battery capacity in its present state (Cr) and total capacity Cbat after completely charging the battery, expressed in a percentage: $SoC = Cr/Cbat * 100\%$.

	Available	N/A	Notes	Rating
Capacity¹²				
Methods based on the relationship between the ampere-hours charged or discharged from the battery and voltage difference before and after the respective CHA/DIS				
- Group 1: Methods considering the change in the measured battery OCV before and after CHA/DIS	X		Main technique for capacity estimation, in conjunction with OCV-curve adaption technique.	+
- Group 2: Methods estimating OCV change from battery terminal voltage measured under load		X		-
- Group 3: Electrical model based joint estimation of SOC and capacity		X		-
- Group 3: Electrical model based dual estimation of SOC and capacity	X		Electrical model can be used to estimate both SoC and Capacity at the same time. May not be necessary, if Group 1 methods are sufficiently fast and accurate.	+
Methods based on incremental capacity analysis (ICA) and different voltage analysis techniques		X		-
Other methods	X		Capacity estimation during charging using cell-specific characteristics. OCV-Curve aging detection.	+

¹²The battery capacity is a figure of merit determining the energy that is stored in the battery and is available for usage when the battery is fully charged. The capacity of the particular battery or cell in a new state is defined by the battery or cell design and varies only slightly for individual batteries or cells of a given type because of the production tolerances. Over the battery lifetime, however, the capacity changes considerably due to aging processes. Therefore, the present capacity of the battery must be determined by the monitoring algorithms.

	Available	N/A	Notes	Rating
Impedance parameters				
- Group 1: Methods based on impedance spectroscopy (active (A)/passive (P))		X	Currently not done in mass-production electronics.	-
- Group 2: Methods based electrical models (How many RC elements)	X		Number of RC elements is a parameter and can be adopted to used battery chemistry.	+
Kalman-Filter based method		X		-
Least-squares based method		X		-
Non-recursive estimation (based on Least-squares)	X		Regression techniques	/
Search algorithm		X		-
Other adaptive filters and observers	X			+
- Group 3: Methods based on electrochemical model		x		-
Least-squares based method		X		-
Gradient-based recursive methods		X		-
Multi-rate particle filter		X		-
- Group 4: Other methods	X		Analysis of specific test pulses that can be triggered during certain operating conditions of the battery pack.	+

	Available	N/A	Notes	Rating
State of Health¹³				
- Group 1: Methods estimate capacity as stated above	X			+
- Group 2: Methods estimate impedance as stated above	X			+
- Group 3: Methods based on the measurement of the absolute voltage or voltage drop when a certain load is applied.	X		No specific different method, same method as battery resistance estimation is used in this case	/
- Group 4: Methods based fuzzy logic method is employed.		X		-
- Group 5: Instead of estimating battery characteristics, conditions such as temperature, number of load cycles, accumulated ampere-hours) are observed over the battery lifetime		X		-

¹³The understanding is that the battery SoH is 100% when the battery is new and 0% when the capability of the battery to store energy or provide power decreases to a certain minimum level. In general, as indicators of the energy and power capability of the battery, the battery capacity (SOHc) and impedance (SOHp), respectively, are evaluated.

Group 3: The approaches are very similar to the estimation of the battery resistance; therefore, the result reflects the power capability of the battery (SOHp).

Group 4: As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used. The resulting SoH does not represent the change in the particular battery characteristic or capability, but the estimation of the general state (eg. NEW, AGED, DEAD).

	Available	N/A	Notes	Rating
Remaining useful life¹⁴				
- Group 1: Methods based on lifetime model used for SOH estimation		X	No online life model is implemented at the moment.	-
- Group 1: Methods based on the SOH prediction model (part 1: estimation of battery characteristics; part 2: prediction model)	X			+

	Available	N/A	Notes	rating
Available power¹⁵				
Methods based on a characteristic map use the static interdependence that exists among the available power of the battery, battery states and power pulse parameters	X			/
Methods based on a dynamic battery model using KF/observer	X		Observer is required for acceptable accuracy	+

¹⁴Remaining time or number of load cycles until the battery reaches a SoH of 0%.

¹⁵Batteries must not only deliver a certain amount of energy to the drive train during operation but also provide a certain power in various situations. The power of the battery can be limited by the voltage, current, SoC, and temperature ranges allowed for safe operation.

	Available	N/A	Notes	Rating
Auxiliary algorithm				
Data preprocessing (thermal model to identify temperatures inside cells, plausibility check of sensor data, detecting low/high performing cells)	X			/
OCV estimation/adaptation during lifetime	X		OCV adaption during lifetime is important for our capacity estimation	+
Adaptive characteristic maps (store parameters for algorithm initialization after wakeup)	X		maps are altered with aging information	/
Available Energy calculation	X		Basis for range estimation. Range estimation algorithm in km is also implemented on the BMS.	+
Cell equalization control (only during charging / discharging or combined)	X			/
Method 1: Terminal voltage based approach	X			/
Method 2: OCV based approach		X		-
Method 3: SOC based approach	X		cell soc when first cell is fully charged is taken into account. This needs cell capacity as an input.	+
Method 4: PWM (early balancing before reaching max voltage)		X		-
Data gathering for off-line diagnosis (warranty analysis)	X			/
Charge time prediction	X			/

	Available	N/A	Notes	Rating
Basic software (safety systems that comply with ISO 26262 and IEC 61508 standards)				
AUTOSAR is considered in the developing process for defining a software architecture, software interfaces and interchange formats, which AUTOSAR version is applied?		X		-
Model-based design?	X		For control algorithms	/
Production-ready MCAL & OS, which OS is used? Compliant to AUTOSAR and OSEK?	X		ISO26262 compliant OS is used, OSEK and AUTOSAR NM are supported.	/
OS: Are you compliant with a full pre-emptive scheduling?	X			/
OS: deterministic RTOS baseline (tasks (timing and protected), events, counters, alarms, messages)	X			/
OS: timing based task determinism (low-latency, precise timing for periodic tasks)	X			/
OS: protected memory / EEPROM/Flash memory management in compliant to AUTOSAR?	X			/
OS: Diagnostic protocols (ISO) and fault management (AUTOSAR) compliant?	X			/
OS: Multi-core features	X		Separate cores for safety and application	+
Interpolations-/CRC,Crypto- and Watchdog routines compliant to AUTOSAR?	X			/
Inputs/Outputs (PWM, ADC, DIO, GPT...) driver compliant to AUTOSAR?		X	Presently, drivers are not AUTOSAR compliant, but this could easily be changed.	-
Bootloader?	X		USD Bootloader	/
Bootloader able to update itself?	X		USD Bootloader	/
Flashing over CAN? If not, which interface?	X			/
JTAG port available on pins on external connector?		X		-
Measurement- and calibration (UDS/CCP/XCP)?	X		UDS and XCP	/

Unit-testing: Interface-/Fault injection-/Resource usage/ Model vs. C-Code tests	X		100% code coverage in unit testing	/
Fixed point implementation?		X		-
Floating point implementation?	X			/
Documentation of sourcecode?	X			/
Available updates / SW release (if, how often can we expect software updates?)	X		Weekly builds will be provided and tested	+
Application software - auto generated code? If so, how is multi-core CPU utilized	X		~ 30% CPU load	+
Various software components implemented with different ASIL in APP Layer (please state), which levels?	X		Safety software runs on a different CPU. ASIL C compliance.	+
Supply of source code		X		-
Application software: Are individual cells considered for battery monitoring, all, if not, how will be decided?	X		Generally, all cells are considered.	+
Will be able to integrate own intelligent algorithms which will be run in parallel to supplier? (Benchmarking)	X			+

System D

Rating	Explanation
+	Exceeds automotive standard
/	Automotive standard
-	Does not fulfill automotive standard

	Available	N/A	Notes	Rating
State of Charge¹⁶				
Ampere-hour counting	X			/
OCV-based estimation	X			/
- Estimation of the battery OCV after current interruption	X			/
Model-based estimation	X			/
- Electrical models	X			/
- Electrochemical models		X	Usually electrochemical models are not used in microcontroller (processing power)	-
Impedance-based estimation		X		-
Estimation based on static battery characteristics	X			/
Estimation using fuzzy logic and machine learning		(X)	Not included	-
- Fuzzy Logic		X		-
- Artificial neural networks (ANN)		X		-
- Fuzzy based neural networks (ANFIS)		X		-
- Support Vector Machines		X		-
Estimation using special measurement techniques		X		-
Monitoring of the SOC of each individual cell in series connection		(X)	Not included in our current module, but is possible to implement.	-
Resolution				/
Other methods	X		Ampere-hour counting and OCV-based estimation are combined to estimate SoC.	/

¹⁶The battery SoC can be employed in the figurative sense as a replacement for a fuel gauge used in conventional vehicles. The SoC is basically the relationship between the residual battery capacity in its present state (Cr) and total capacity Cbat after completely charging the battery, expressed in a percentage: $SoC = Cr/Cbat * 100\%$.

	Available	N/A	Notes	Rating
Capacity¹⁷				
Methods based on the relationship between the ampere-hours charged or discharged from the battery and voltage difference before and after the respective CHA/DIS.				
- Group 1: Methods considering the change in the measured battery OCV before and after CHA/DIS		X	<i>Currently not part</i>	-
- Group 2: Methods estimating OCV change from battery terminal voltage measured under load		X		-
- Group 3: Electrical model based joint estimation of SOC and capacity		X		-
- Group 3: Electrical model based dual estimation of SOC and capacity		X		-
Methods based on incremental capacity analysis (ICA) and different voltage analysis techniques		X		-
Other methods		X		-

¹⁷The battery capacity is a figure of merit determining the energy that is stored in the battery and is available for usage when the battery is fully charged. The capacity of the particular battery or cell in a new state is defined by the battery or cell design and varies only slightly for individual batteries or cells of a given type because of the production tolerances. Over the battery lifetime, however, the capacity changes considerably due to aging processes. Therefore, the present capacity of the battery must be determined by the monitoring algorithms.

	Available	N/A	Notes	Rating
Impedance parameters				
- Group 1: Methods based on impedance spectroscopy (active (A)/passive (P))		X		-
- Group 2: Methods based electrical models (How many RC elements)	X		Currently 0 RC-elements, because of customer specification.	-
Kalman-Filter based method		X		-
Least-squares based method		X	Linear interpolation is implemented.	-
Non-recursive estimation (based on Least-squares)		X	Linear interpolation is implemented.	-
Search algorithm				-
Other adaptive filters and observers			-	-
- Group 3: Methods based on electrochemical model				-
Least-squares based method		X	<i>Usually electrochemical models are not used in microcontroller (processing power)</i>	-
Gradient-based recursive methods		X		-
Multi-rate particle filter		X		-
- Group 4: Other methods				-

	Available	N/A	Notes	Rating
State of Health¹⁸				
- Group 1: Methods estimate capacity as stated above	X			+
- Group 2: Methods estimate impedance as stated above		X		-
- Group 3: Methods based on the measurement of the absolute voltage or voltage drop when a certain load is applied.		X		-
- Group 4: Methods based fuzzy logic method is employed. As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used.		(X)		-
- Group 5: Instead of estimating battery characteristics, conditions such as temperature, number of load cycles, accumulated ampere-hours) are observed over the battery lifetime	X			-

¹⁸The understanding is that the battery SoH is 100% when the battery is new and 0% when the capability of the battery to store energy or provide power decreases to a certain minimum level. In general, as indicators of the energy and power capability of the battery, the battery capacity (SOHc) and impedance (SOHp), respectively, are evaluated.

Group 3: The approaches are very similar to the estimation of the battery resistance; therefore, the result reflects the power capability of the battery (SoHp).

Group 4: As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used. The resulting SoH does not represent the change in the particular battery characteristic or capability, but the estimation of the general state (eg. NEW, AGED, DEAD).

	Available	N/A	Notes	Rating
Remaining useful life¹⁹				
- Group 1: Methods based on lifetime model used for SOH estimation	X			+
- Group 1: Methods based on the SOH prediction model (part 1: estimation of battery characteristics; part 2: prediction model)		X		-

Available power²⁰				
Methods based on a characteristic map use the static interdependence that exists among the available power of the battery, battery states and power pulse parameters	X			/
Methods based on a dynamic battery model using KF/observer	X			-

	Available	N/A	Notes	Rating
Auxiliary algorithm				
Data preprocessing (thermal model to identify temperatures inside cells, plausibility check of sensor data, detecting low/high performing cells)		X		-
OCV estimation/adaptation during lifetime	X		(Static table, no adaptation)	-
Adaptive characteristic maps (store parameters for algorithm initialization after wakeup)		?		-
Available Energy calculation	X			+
Cell equalization control (only during charging / discharging or combined)	X		Passive balancing	/
Method 1: Terminal voltage based approach	X		Cell voltage will be measured while balancing break.	/
Method 2: OCV based approach	X			/
Method 3: SOC based approach		X		-
Method 4: PWM (early balancing before reaching max voltage)		X		-
Data gathering for off-line diagnosis (warranty analysis)	X			/
Charge time prediction		X	Can be implemented.	-

¹⁹Remaining time or number of load cycles until the battery reaches a SoH of 0%.

²⁰Batteries must not only deliver a certain amount of energy to the drive train during operation but also provide a certain power in various situations. The power of the battery can be limited by the voltage, current, SoC, and temperature ranges allowed for safe operation.

	Available	N/A	Notes	Rating
Basic software (safety systems that comply with ISO 26262 and IEC 61508 standards)				
AUTOSAR is considered in the developing process for defining a software architecture, software interfaces and interchange formats, which AUTOSAR version is applied?			Adaption based on ASR3.1	/
Model-based design?		X		-
Production-ready MCAL & OS, which OS is used? Compliant to AUTOSAR and OSEK?	X		"EB tresos AutoCore OS 4.2.2 AUTOSAR and OSEK compliant"	/
OS: Are you compliant with a full pre-emptive scheduling?	X			/
OS: deterministic RTOS baseline (tasks (timing and protected), events, counters, alarms, messages)	X			/
OS: timing based task determinism (low-latency, precise timing for periodic tasks)	X			/
OS: protected memory / EEPROM/Flash memory management in compliant to AUTOSAR?	X		memory protection for safety relevant software by MMU / MPU	/
OS: Diagnostic protocols (ISO) and fault management (AUTOSAR) compliant?	X		DCM, DEM accordings ASR3.1	/
OS: Multi-core features		X	used microcontroller is a single core	/
Interpolations-/CRC,Crypto- and Watchdog routines compliant to AUTOSAR?		X	Watchdog yes, rest is not used	-
Inputs/Outputs (PWM, ADC, DIO, GPT...) driver compliant to AUTOSAR?	X			/
Bootloader?	X		ASR3.1 based	/
Bootloader able to update itself?	X		Yes, Bootloader can be updated (via CAN).	/
Flashing over CAN? If not, which interface?	X		All components (also CSSU/CVS) and bootloader can be flashed via vehicle can.	/
JTAG port available on pins on external connector?		X	JTAG interface is only available for development with special connectors on pcb.	-
Measurement- and calibration (UDS/CCP/XCP)?	X		via UDS and (CCP/XCP)	/

Unit-testing: Interface-/Fault injection-/Resource usage/ Model vs. C-Code tests	X		full development / test process according automotive spice	/
Fixed point implementation?	X			/
Floating point implementation?	X			/
Documentation of sourcecode?	X		internally available	-
Available updates / SW release (if, how often can we expect software updates?)	X		Timing and changes have to be individually planned.	/
Application software - auto generated code? If so, how is multi-core CPU utilized		X	handwritten C-Code on a single-core CPU	/
Various software components implemented with different ASIL in APP Layer (please state), which levels?		X	In APP-Layer only a stuck detection for the conductors is safety relevant (ASIL-A). All other safety funtions are implemented in the basic software (ASIL-C).	/
Supply of source code		X	No, the code is only a black box.	-
Application software: Are individual cells considered for battery monitoring, all, if not, how will be decided?	X		Every single cell is supervised. For the SOC, SOP calculation the min/max cells are taken into account.	+
Will be able to integrate own intelligent algorithms which will be run in parallel to supplier? (Benchmarking)	X		It is possible to integrate a highlevel SW or only parts of it to the basic software. A parallel use of 2 different algorithms with same function at the same time is not prepared. Can be designed and implemented.	+

System E

Rating	Explanation
+	Exceeds automotive standard
/	Automotive standard
-	Does not fulfill automotive standard

	Available	N/A	Notes	Rating
State of Charge²¹				
Ampere-hour counting	X			/
OCV-based estimation	X			/
- Estimation of the battery OCV after current interruption	X			/
Model-based estimation	X			/
- Electrical models	X			/
- Electrochemical models		X		-
Impedance-based estimation (dependence of some battery impedance parameters on the SoC)	X			+
Estimation based on static battery characteristics		X		-
Estimation using fuzzy logic and machine learning		X		-
- Fuzzy Logic		X		-
- Artificial neural networks (ANN)		X		-
- Fuzzy based neural networks (ANFIS)		X		-
- Support Vector Machines		X		-
Estimation using special measurement techniques		X		-
Monitoring of the SOC of each individual cell in series connection	X			+
Resolution				NA
Other methods		X		-

²¹The battery SoC can be employed in the figurative sense as a replacement for a fuel gauge used in conventional vehicles. The SoC is basically the relationship between the residual battery capacity in its present state (Cr) and total capacity Cbat after completely charging the battery, expressed in a percentage: $SoC = Cr/Cbat * 100\%$.

	Available	N/A	Notes	Rating
Capacity²²				
Methods based on the relationship between the ampere-hours charged or discharged from the battery and voltage difference before and after the respective CHA/DIS				
- Group 1: Methods considering the change in the measured battery OCV before and after CHA/DIS		X		-
- Group 2: Methods estimating OCV change from battery terminal voltage measured under load		X		-
- Group 3: Electrical model based joint estimation of SOC and capacity		X		-
- Group 3: Electrical model based dual estimation of SOC and capacity		X		-
Methods based on incremental capacity analysis (ICA) and different voltage analysis techniques		X		-
Other methods	X			+

	Available	N/A	Notes	Rating
Impedance parameters				
- Group 1: Methods based on impedance spectroscopy (active (A)/passive (P))		X		-
- Group 2: Methods based electrical models (How many RC elements)				-
Kalman-Filter based method		X		-
Least-squares based method	X			/
Non-recursive estimation (based on Least-squares)		X		-
Search algorithm		X		-
Other adaptive filters and observers		X		-
- Group 3: Methods based on electrochemical model				-
Least-squares based method		X		-
Gradient-based recursive methods		X		-
Multi-rate particle filter		X		-
- Group 4: Other methods		X		-

²²The battery capacity is a figure of merit determining the energy that is stored in the battery and is available for usage when the battery is fully charged. The capacity of the particular battery or cell in a new state is defined by the battery or cell design and varies only slightly for individual batteries or cells of a given type because of the production tolerances. Over the battery lifetime, however, the capacity changes considerably due to aging processes. Therefore, the present capacity of the battery must be determined by the monitoring algorithms.

	Available	N/A	Notes	Rating
State of Health²³				
- Group 1: Methods estimate capacity as stated above	X			+
- Group 2: Methods estimate impedance as stated above	X			+
- Group 3: Methods based on the measurement of the absolute voltage or voltage drop when a certain load is applied.		X		-
- Group 4: Methods based fuzzy logic method is employed. As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used.		X		-
- Group 5: Instead of estimating battery characteristics, conditions such as temperature, number of load cycles, accumulated ampere-hours) are observed over the battery lifetime		X		-

	Available	N/A	Notes	Rating
Remaining useful life²⁴				
- Group 1: Methods based on lifetime model used for SOH estimation		X		-
- Group 1: Methods based on the SOH prediction model (part 1: estimation of battery characteristics; part 2: prediction model)		X		-

²³The understanding is that the battery SoH is 100% when the battery is new and 0% when the capability of the battery to store energy or provide power decreases to a certain minimum level. In general, as indicators of the energy and power capability of the battery, the battery capacity (SOHc) and impedance (SOHp), respectively, are evaluated.

Group 3: The approaches are very similar to the estimation of the battery resistance; therefore, the result reflects the power capability of the battery (SoHp).

Group 4: As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used. The resulting SoH does not represent the change in the particular battery characteristic or capability, but the estimation of the general state (eg. NEW, AGED, DEAD).

²⁴Remaining time or number of load cycles until the battery reaches a SoH of 0%.

	Available	N/A	Notes	rating
Available power²⁵				
Methods based on a characteristic map use the static interdependence that exists among the available power of the battery, battery states (for example, SoC, temperature, and voltage), and power pulse parameters (for example, the duration of the power pulse)	X			/
Methods based on a dynamic battery model using KF/observer	X			+

	Available	N/A	Notes	Rating
Auxiliary algorithm				
Data preprocessing (thermal model to identify temperatures inside cells, plausibility check of sensor data, detecting low/high performing cells)	X			/
OCV estimation/adaptation during lifetime	X			-
Adaptive characteristic maps (store parameters for algorithm initialization after wakeup)	X			/
Available Energy calculation	X			+
Cell equalization control (only during charging / discharging or combined)	X			/
Method 1: Terminal voltage based approach		X		-
Method 2: OCV based approach		X		-
Method 3: SOC based approach	X			+
Method 4: PWM (early balancing before reaching max voltage)		X		-
Data gathering for off-line diagnosis (warranty analysis)	X			/
Charge time prediction		X		-

²⁵Batteries must not only deliver a certain amount of energy to the drive train during operation but also provide a certain power in various situations. The power of the battery can be limited by the voltage, current, SoC, and temperature ranges allowed for safe operation.

	Available	N/A	Notes	Rating
Basic software (safety systems that comply with ISO 26262 and IEC 61508 standards)				
AUTOSAR is considered in the developing process for defining a software architecture, software interfaces and interchange formats, which AUTOSAR version is applied?	X			/
Model-based design?	X			/
Production-ready MCAL & OS, which OS is used? Compliant to AUTOSAR and OSEK?	X			/
OS: Are you compliant with a full pre-emptive scheduling?	X			/
OS: deterministic RTOS baseline (tasks (timing and protected), events, counters, alarms, messages)	X			/
OS: timing based task determinism (low-latency, precise timing for periodic tasks)	X			/
OS: protected memory / EEPROM/Flash memory management in compliant to AUTOSAR?	X			/
OS: Diagnostic protocols (ISO) and fault management (AUTOSAR) compliant?	X			/
OS: Multi-core features	X			+
Interpolations-/CRC,Crypto- and Watchdog routines compliant to AUTOSAR?	X			/
Inputs/Outputs (PWM, ADC, DIO, GPT...) driver compliant to AUTOSAR?	X			/
Bootloader?	X			/
Bootloader able to update itself?	X			/
Flashing over CAN? If not, which interface?	X			/
JTAG port available on pins on external connector?		X		-
Measurement- and calibration (UDS/CCP/XCP)?	X			/

Unit-testing: Interface-/Fault injection-/Resource usage/ Model vs. C-Code tests	X		full development / test process according automotive spice	/
Fixed point implementation?		X		-
Floating point implementation?	X			/
Documentation of sourcecode?	X			/
Available updates / SW release (if, how often can we expect software updates?)	X			/
Application software - auto generated code? If so, how is multi-core CPU utilized	X			/
Various software components implemented with different ASIL in APP Layer (please state), which levels?	X			/
Supply of source code		X		-
Application software: Are individual cells considered for battery monitoring, all, if not, how will be decided?	X			+
Will be able to integrate own intelligent algorithms which will be run in parallel to supplier? (Benchmarking)	X			+

System P (proposed)

Rating	Explanation
+	Exceeds automotive standard
/	Automotive standard
-	Does not fulfill automotive standard

	Available	N/A	Notes	Rating
State of Charge²⁶				
Ampere-hour counting	X			+
OCV-based estimation	X			+
- Estimation of the battery OCV after current interruption	X			+
Model-based estimation	X			+
- Electrical models	X			+
- Electrochemical models		X		-
Impedance-based estimation (dependence of some battery impedance parameters on the SoC)	X			-
Estimation based on static battery characteristics		X		-
Estimation using fuzzy logic and machine learning			Available	+
- Fuzzy Logic	X			+
- Artificial neural networks (ANN)		X		-
- Fuzzy based neural networks (ANFIS)	X			+
- Support Vector Machines		X		-
Estimation using special measurement techniques		X		-
Monitoring of the SOC of each individual cell in series connection	X			+
Resolution	X		Floating	+
Other methods				-

²⁶The battery SoC can be employed in the figurative sense as a replacement for a fuel gauge used in conventional vehicles. The SoC is basically the relationship between the residual battery capacity in its present state (Cr) and total capacity Cbat after completely charging the battery, expressed in a percentage: $SoC = Cr/Cbat * 100\%$.

	Available	N/A	Notes	Rating
Capacity²⁷				
Methods based on the relationship between the ampere-hours charged or discharged from the battery and voltage difference before and after the respective CHA/DIS				
- Group 1: Methods considering the change in the measured battery OCV before and after CHA/DIS	X			+
- Group 2: Methods estimating OCV change from battery terminal voltage measured under load	X			+
- Group 3: Electrical model based joint estimation of SOC and capacity	X			+
- Group 3: Electrical model based dual estimation of SOC and capacity	X			+
Methods based on incremental capacity analysis (ICA) and different voltage analysis techniques		X		-
Other methods	X			+

	Available	N/A	Notes	Rating
Impedance parameters				
- Group 1: Methods based on impedance spectroscopy (active (A)/passive (P))	X			/
- Group 2: Methods based electrical models (How many RC elements)	X			+
Kalman-Filter based method	X			+
Least-squares based method	X			+
Non-recursive estimation (based on Least-squares)		X		-
Search algorithm	X			+
Other adaptive filters and observers	X			+
- Group 3: Methods based on electrochemical model				-
Least-squares based method		X		-
Gradient-based recursive methods		X		-
Multi-rate particle filter	X			+
- Group 4: Other methods		X		-

²⁷The battery capacity is a figure of merit determining the energy that is stored in the battery and is available for usage when the battery is fully charged. The capacity of the particular battery or cell in a new state is defined by the battery or cell design and varies only slightly for individual batteries or cells of a given type because of the production tolerances. Over the battery lifetime, however, the capacity changes considerably due to aging processes. Therefore, the present capacity of the battery must be determined by the monitoring algorithms.

	Available	N/A	Notes	Rating
State of Health²⁸				
- Group 1: Methods estimate capacity as stated above	X			+
- Group 2: Methods estimate impedance as stated above	X			+
- Group 3: Methods based on the measurement of the absolute voltage or voltage drop when a certain load is applied.		X		-
- Group 4: Methods based fuzzy logic method is employed. As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used.		X		-
- Group 5: Instead of estimating battery characteristics, conditions such as temperature, number of load cycles, accumulated ampere-hours) are observed over the battery lifetime		X		-

	Available	N/A	Notes	Rating
Remaining useful life²⁹				
Remaining time or number of load cycles until the battery reaches a SoH of 0%		x		
- Group 1: Methods based on lifetime model used for SOH estimation		X		-
- Group 1: Methods based on the SOH prediction model (part 1: estimation of battery characteristics; part 2: prediction model)		X		-

²⁸The understanding is that the battery SoH is 100% when the battery is new and 0% when the capability of the battery to store energy or provide power decreases to a certain minimum level. In general, as indicators of the energy and power capability of the battery, the battery capacity (SOHc) and impedance (SOHp), respectively, are evaluated.

Group 3: The approaches are very similar to the estimation of the battery resistance; therefore, the result reflects the power capability of the battery (SoHp).

Group 4: As inputs, the measured impedance parameters and/ or other measured or calculated battery characteristics are used. The resulting SoH does not represent the change in the particular battery characteristic or capability, but the estimation of the general state (eg. NEW, AGED, DEAD).

²⁹Remaining time or number of load cycles until the battery reaches a SoH of 0%.

	Available	N/A	Notes	rating
Available power³⁰				
Methods based on a characteristic map use the static interdependence that exists among the available power of the battery, battery states (for example, SoC, temperature, and voltage), and power pulse parameters (for example, the duration of the power pulse)	X			+
Methods based on a dynamic battery model using KF/observer	X			+

	Available	N/A	Notes	Rating
Auxiliary algorithm				
Data preprocessing (thermal model to identify temperatures inside cells, plausibility check of sensor data, detecting low/high performing cells)	X			+
OCV estimation/adaptation during lifetime	X			+
Adaptive characteristic maps (store parameters for algorithm initialization after wakeup)	X			+
Available Energy calculation		X		-
Cell equalization control (only during charging / discharging or combined)	X			+
Method 1: Terminal voltage based approach	X			+
Method 2: OCV based approach	X			+
Method 3: SOC based approach	X			+
Method 4: PWM (early balancing before reaching max voltage)		X		-
Data gathering for off-line diagnosis (warranty analysis)		X		-
Charge time prediction	X			+

³⁰Batteries must not only deliver a certain amount of energy to the drive train during operation but also provide a certain power in various situations. The power of the battery can be limited by the voltage, current, SoC, and temperature ranges allowed for safe operation.

	Available	N/A	Notes	Rating
Basic software (safety systems that comply with ISO 26262 and IEC 61508 standards)				
AUTOSAR is considered in the developing process for defining a software architecture, software interfaces and interchange formats, which AUTOSAR version is applied?	X			/
Model-based design?	X			/
Production-ready MCAL & OS, which OS is used? Compliant to AUTOSAR and OSEK?	X			/
OS: Are you compliant with a full pre-emptive scheduling?	X			/
OS: deterministic RTOS baseline (tasks (timing and protected), events, counters, alarms, messages)	X			/
OS: timing based task determinism (low-latency, precise timing for periodic tasks)	X			/
OS: protected memory / EEPROM/Flash memory management in compliant to AUTOSAR?	X			/
OS: Diagnostic protocols (ISO) and fault management (AUTOSAR) compliant?	X			/
OS: Multi-core features	X			+
Interpolations-/CRC,Crypto- and Watchdog routines compliant to AUTOSAR?	X			+
Inputs/Outputs (PWM, ADC, DIO, GPT...) driver compliant to AUTOSAR?	X			+
Bootloader?	X			+
Bootloader able to update itself?	X			/
Flashing over CAN? If not, which interface?	X			/
JTAG port available on pins on external connector?		X		-
Measurement- and calibration (UDS/CCP/XCP)?	X			/

Unit-testing: Interface-/Fault injection-/Resource usage/ Model vs. C-Code tests	X			/
Fixed point implementation?		X		-
Floating point implementation?	X			/
Documentation of sourcecode?		X		-
Available updates / SW release (if, how often can we expect software updates?)	X			/
Application software - auto generated code? If so, how is multi-core CPU utilized	X			/
Various software components implemented with different ASIL in APP Layer (please state), which levels?		X		-
Supply of source code		X		-
Application software: Are individual cells considered for battery monitoring, all, if not, how will be decided?	X			+
Will it be possible to integrate own intelligent algorithms which will be run in parallel to supplier? (Benchmarking)	X			/

List of Figures

2.1	From battery monitoring to battery management - an illustration (modified from [5])	6
2.2	Complexity of automotive software functions in terms of tasks execution (Trends in Embedded Systems and Vehicle Networks — Gunnarsson, BMW AG — 2016-05-11)	9
2.3	Content of work.	13
2.4	Conducted test drives of different xEV's for evaluating user experience and SW quality.	17
2.5	Conducted test drives of different xEV's for evaluating user experience and SW quality.	18
3.1	Schematic illustration of a battery pack and its main components. (modified from [3])	20
3.2	Schematic illustration of a battery module and its components. (modified from [3])	20
3.3	Schematic illustration of battery management system. (modified from [3]) .	21
3.6	Proposed and developed SW architecture following AUTOSAR standard. .	29
	(a) AUTOSAR layered architecture provided by AUTOSAR.	29
	(b) Developed SW architecture for advanced, safety-critical BMS.	29
3.4	Schematic illustration of BMS functions (modified from [4][11]).	32
3.5	Schematic illustration of BMS functions.	33
3.7	Supplier A: BMS E-architecture	43
3.8	Supplier B: BMS E-architecture	44
3.9	System C: BMS E-architecture	45
3.10	System D: System architecture	46
3.11	System E: System architecture	47
3.12	System F: Tested EE architectures	48
3.13	System G: Tested EE architectures	49
3.14	BMS HW evaluation diagram	50
3.15	BMS SW evaluation diagram	52
4.1	Block diagram of adaptive filtering [362].	54
4.2	Hidden Markov model for battery state and measurement evolution	75
5.1	Approximation solution for recursive Bayesian estimation [380]	81
5.2	State-of-Charge estimation using EnKF. Data provided by [2]	86
5.3	Battery cell voltage estimation using EnKF. Data provided by [2]	86

5.4	Parameter estimation of 1-RC ECM using EnKF. Data provided by [2]	86
5.5	Performing joint estimation for first order Li-ion battery ECM using ensemble KALMAN filter.	86
5.6	A pictorial description of the particle filter. [415]	113
5.7	Illustration of the Gaussian mixture sigma-point particle filter [390].	121
5.8	Dual estimation of state and parameter quantities using two KALMAN filters running in parallel including data sharing. [2]	126
5.9	Summarizing Bayesian and classical methodology.	131
6.1	Comparison of different modeling approaches for battery systems	135
6.2	Simple ECM, considering only ohmic resistance.	138
6.3	First-order ECM.	139
6.4	Rate-dependent first-order ECM.	141
6.5	Rate-dependent second-order ECM.	143
6.6	Validation of the algorithms on the actual HV traction battery system. (modified from [380])	145
6.7	Validation of the algorithms based on simulative framework. (modified from [380])	147
6.8	E-Fiat data collection and on-line estimation setup.	149
	(a) Vehicle setup for data collection and SW verification.	149
	(b) E-Fiat system voltage CAN-bus reading (in yellow) and real-time estimation (in blue) running on MABXII2 while driving. HW 1,3,4,5 was installed in [445] and extended in this work with HW 2,6,7.	149
6.9	Monte Carlo simulation to mimic aging behavior collected from eFiat data collection, here EKF dual estimation convergence properties of SoC , R_0 and τ_1	150
	(a) Simulate aging by Monte-Carlo simulation - SoC .	150
	(b) Simulate aging by Monte-Carlo simulation - internal resistance	150
	(c) Simulate aging by Monte-Carlo simulation - time constant τ_1	150
6.10	Expert judgment on necessary advanced data requirements and implementation challenges.	151
6.11	A cartoon drawing of a biological neuron (left) and its mathematical model (right).[447]	154
	(a) Biological neural systems	154
	(b) Biological motivation and connections	154
6.12	A feed-forward neural network with two hidden layers and modified from [446].	155
6.13	ANFIS structure.	158
6.14	Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017 - Artificial Intelligence Everywhere	160
6.15	A classical RGB image concept [450].	161
6.16	Convolution operation [450].	162
6.17	General CNN structure [450].	164

6.18	CNN for State-of-Health determination	165
(a)	The way to generate a pseudo image consisting all the battery signals [450].	165
(b)	The way to implement CNNs on the estimation of the battery's state of health [450].	165
6.19	Architecture of a convolutional neural network (CNN) where each layer is designed of a convolution and pooling component followed by a fully connected layer.	167
6.20	DVA as a tool for analyzing aging. [456]	171
(a)	Different voltage areas by its derivative with respect to capacity [456].	171
(b)	DVA curves for for aging [456].	171
6.21	Figure 1. There is a fact that batteries with different SoH have different performance while charging [457].	172
6.22	SoH estimation based on CNN - (a) using all available test data from 28 cells with (b) best accuracy of 97.4% [450]	174
(a)	Full data set - CNN estimation vs. lab ground truth	174
(b)	DVA curves for for aging [456].	174
6.24	Structure of recurrent neural network used for SoH estimation [modified by [450]]	175
6.25	Variables flow in a RNN [446].	176
6.23	SoH estimation based on CNN - (a) using washed data from 28 cells (excluding corrupted data) with (b) better overall accuracy (max. 98.6%) [450]	179
(a)	Test on washed data / SoH estimation	179
(b)	Test on washed data / MSE and accuracy.	179
6.26	SoH estimation based on RNN - (a) estimation vs. lab data, (b) showing the loss in MSE [450]	180
(a)	Test on washed data / SoH estimation	180
(b)	est on washed data / MSE and accuracy	180
6.27	SoH estimation based on RNN - (a) including all the trimmed data, (b) the best accuracy results in 85% [450]	181
(a)	Test on washed data / SoH estimation	181
(b)	Test on washed data showing the accuracy spread over the epochs . .	181
6.28	Variation of LSTM layers with limited improvements in MSE (trained on GPU Geforce 940MX)[450]	182
(a)	Test on washed data / SoH estimation using 3 LSTM layers, 128 nodes and 100 epochs.	182
(b)	Linear training time per epochs vs. number of LSTM layers	182
6.29	MSE bounded MLE performance vs. SNR for estimation for CRLB and Ziv-Zakai.	186
6.30	Implemented algorithm library in ReBAAI toolbox	188
6.31	Results of Monte-Carlo simulation for state estimation model 1, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.	190
6.32	Results of Monte-Carlo simulation for state estimation model 2, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.	191

6.33	Results of Monte-Carlo simulation for state estimation model 3, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.	191
6.34	Results of Monte-Carlo simulation for state estimation model 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.	192
6.35	Results of Monte-Carlo simulation for joint estimation for model 2, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100. . .	193
6.36	Results of Monte-Carlo simulation for joint estimation for model 3, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100. . .	193
6.37	Results of Monte-Carlo simulation for joint estimation for model 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100. . .	194
6.38	Results of Monte-Carlo simulation for dual estimation for model 2 and 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100.	195
6.39	Results of Monte-Carlo simulation for dual estimation for model 3, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100. . .	195
6.40	Results of Monte-Carlo simulation for dual estimation for model 4, used driving cycle: E-Fiat-10Grad-City-Highway, number of iterations - 100. . .	196
7.1	<i>Left:</i> Overview of the dSPACE HiL simulator; <i>right:</i> BMS under test: Lithium Balance sBMS. [460]	203
7.3	Schematic implementation of battery disconnect unit (BDU) topology and its interfaces to the battery system model.[460][461][462]	206
7.2	Battery system model including all components.	209
7.4	Overview of main functions and interfaces between the host PC, the HiL simulator and the BMS. [460][461][462]	211
7.5	Components of fault injection environment. (modified from [465])	212
7.6	Connection between fault, error and failure [465]	214
7.7	Challenges of BMS testing. Source: Shanghai Keliang Information Tech. and Eng. Co., Ltd.	217
7.8	Functional tree of a state-of-the-art BMS for large Li-ion battery packs. [460][461][462]	218
7.9	Test cases proposed to evaluate a BMS for its FRs and NFRs. [460][461][462]	219
7.10	Experimental setup for HiL simulation: main functions and interfaces between host PC, HiL simulator and DUT. [460][461][462]	220
7.11	Snapshot of ad hoc SW for management of BMS settings provided by Lithium Balance.	222
7.12	BMS functional testing: CCCV charging	223
	(a) Cell voltages during CCCV charging	223
	(b) Cell temperatures during CCCV charging	223
7.13	BMS functional testing: CCCV charging	224
	(a) System voltage (32s) during charging	224
	(b) Charging current controlled by LiB BMS	224
7.14	BMS functional testing: CCCV charging	225
	(a) Min/max SoC during charging	225
	(b) Balancing currents during charging	225
	(a) Cell voltages during CCCV charging	227

(b)	Cell temperatures during CCCV charging	227
7.16	BMS functional testing: CCCV charging at low temperature	228
(a)	System voltage (32s) during charging	228
(b)	Charging current controlled by LiB BMS	228
(a)	Cell voltages during CCCV charging	230
(b)	System voltage (32s) during charging	230
(a)	Charging current controlled by LiB BMS	231
(b)	System voltage (32s) during charging	231
(a)	Cell voltages during CCCV charging	232
(b)	Charging current	232
7.20	BMS non-functional testing: cell temperature increase during load	233
(a)	Cell voltages during load	233
(b)	Cell temperatures during load	233
(a)	Cell voltages during dynamic load cycle	234
(b)	Emulated cell voltages and BMS min/max measurements	234
(c)	System voltage (32s) & load current	234
7.21	BMS non-functional testing: cell temperature increase during load	235
(d)	Individual SoC during driving cycle	235
(e)	Switches - BMS opens when $T > 55^{\circ}C$	235
(f)	Pre-charge sequence	235
(a)	Cell voltages during driving cycle and over-current.	236
7.23	BMS non-functional testing: recuperation.	237
(a)	System voltage (32s) & load current (over-current).	237
(b)	Individual SoC during driving	237
(a)	Cell voltages during cell short-circuit.	238
(b)	Cell temperature during short circuit.	238
7.25	Fault insertion: Temperature scaling error	239
8.1	Overview of ISO 26262 [https://www.iso.org/standard/43464.html]	243
8.2	ISO 26262-6, Reference phase model for the software development https://www.iso.org/standard/68383.html	248
8.3	Required tools in TargetLink Ecosystem - the industry's traditional model-based design toolset	256
8.4	Layered Architecture of AUTOSAR with Application SW	259
8.5	Proposed SW Functional Architecture for Battery Managment System	261
8.6	BMS functional and architectural system model in a SysML-based environment using SCADE Architect to deploy model-based system engineering	262
8.7	Extended KALMAN filter bank with 100 instances for parallel computing (modified from [376])	264
8.8	SCADE multi-core code generated from v6.6 models [474]	265
8.9	SCADE multi-core code generation for prototype v6.6 models (modified from [474])	266
(a)	CPU load for multi-core optimized extended KALMAN filter bank	267
(b)	CPU load for sequential executed extended KALMAN filter bank	267

A.1	HW evaluation of System #1	274
A.2	HW evaluation of System #2	276
A.3	HW evaluation of System #3	277
A.4	HW evaluation of System #4	278
A.5	HW evaluation of System #5	279
A.6	HW evaluation of System #6	281
A.7	HW evaluation of System #7, off-the-shelve product from Lithium Balance	282

List of Tables

3.1 Summary of SWC for ASW, BSW, FSSW 31

7.1 Classification of BMS Tests. [460][461][462] 217

7.2 Summary of FI modes and their location. 221

Bibliography

- [1] H. Chaoui, N. Golbon, I. Hmouz, R. Souissi, and S. Tahar, “Lyapunov-Based Adaptive State of Charge and State of Health Estimation for Lithium-Ion Batteries,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1610–1618, Mar. 2015.
- [2] G. Plett, *Battery Management Systems, Volume II: Equivalent-Circuit Methods*, ser. Artech House power engineering series v. 2. Artech House, 2015.
- [3] “Advances in Battery Technologies for Electric Vehicles,” in *Advances in Battery Technologies for Electric Vehicles*, ser. Woodhead Publishing Series in Energy, B. Scrosati, J. Garche, and W. Tillmetz, Eds., Woodhead Publishing, 2015, pp. 517–526.
- [4] D. Andrea, *Battery Management Systems for Large Lithium Ion Battery Packs*, ser. EBL-Schweitzer. Artech House, 2010.
- [5] B. Fridholm. (2018). “The Role of Battery Management in Large Scale Electrification,” (visited on 04/13/2018).
- [6] A. G. Security. (2015). “Hackers Remotely Kill a Jeep on the Highway — With Me in It,” (visited on 07/21/2015).
- [7] G. Pistoia, *Lithium-Ion Batteries: Advances and Applications*. Jan. 2014, pp. 1–612.
- [8] J. Warner, *The Handbook of Lithium-Ion Battery Pack Design: Chemistry, Components, Types and Terminology*. Elsevier Science, 2015.
- [9] P. Weicker, *A Systems Approach to Lithium-Ion Battery Management*, ser. Power Engineering. Artech House, 2013.
- [10] S. Santhanagopalan, K. Smith, J. Neubauer, G. Kim, A. Pesaran, and M. Keyser, *Design and Analysis of Large Lithium-Ion Battery Systems*, ser. Artech House power engineering series. Artech House Publishers, 2014.
- [11] J. Barreras, “Practical Methods in Li-ion Batteries: for Simplified Modeling, Battery Electric Vehicle Design, Battery Management System Testing and Balancing System Control,” English, Ph.D. dissertation, 2017.
- [12] I. to AUTOSAR. (2019). “AUTOSAR CLASSIC PLATFORM 4.3.1,” [Online]. Available: www.autosar.org.
- [13] W. Waag, C. Fleischer, and D. U. Sauer, “Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles,” *Journal of Power Sources*, vol. 258, pp. 321–339, Jul. 2014.
- [14] A. Farmann, W. Waag, A. Marongiu, and D. U. Sauer, “Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles,” *Journal of Power Sources*, vol. 281, pp. 114–130, 2015.
- [15] A. Farmann and D. U. Sauer, “A comprehensive review of on-board State-of-Available-Power prediction techniques for lithium-ion batteries in electric vehicles,” *Journal of Power Sources*, vol. 329, pp. 123–137, 2016.
- [16] B. Ritter, A. Schild, M. Feldt, and U. Konigorski, “The design of nonlinear observers for wind turbine dynamic state and parameter estimation,” *Journal of Physics: Conference Series*, vol. 753, p. 052029, Sep. 2016.

- [17] Z. Deng, L. Yang, Y. Cai, H. Deng, and L. Sun, "Online available capacity prediction and state of charge estimation based on advanced data-driven algorithms for lithium iron phosphate battery," *Energy*, vol. 112, pp. 469–480, Oct. 2016.
- [18] Z. He, C. Dong, C. Pan, C. Long, and S. Wang, "State of charge estimation of power Li-ion batteries using a hybrid estimation algorithm based on UKF," *Electrochimica Acta*, vol. 211, pp. 101–109, Sep. 2016.
- [19] Y.-S. Lee, M. Liu, C.-C. Sun, and M.-W. Cheng, "State-of-charge estimation with aging effect and correction for lithium-ion battery," *IET Electrical Systems in Transportation*, vol. 5, no. 2, pp. 70–76, Jun. 2015.
- [20] S. Sepasi, R. Ghorbani, and B. Y. Liaw, "Inline state of health estimation of lithium-ion batteries using state of charge calculation," *Journal of Power Sources*, vol. 299, pp. 246–254, Dec. 2015.
- [21] N. Yang, X. Zhang, and G. Li, "State of charge estimation for pulse discharge of a LiFePO₄ battery by a revised Ah counting," *Electrochimica Acta*, vol. 151, pp. 63–71, Jan. 2015.
- [22] C. Zhang, K. Li, L. Pei, and C. Zhu, "An integrated approach for real-time model-based state-of-charge estimation of lithium-ion batteries," *Journal of Power Sources*, vol. 283, pp. 24–36, Jun. 2015.
- [23] Y. Zhang, W. Song, S. Lin, and Z. Feng, "A novel model of the initial state of charge estimation for LiFePO₄ batteries," *Journal of Power Sources*, vol. 248, pp. 1028–1033, Feb. 2014.
- [24] G. V. Avvari, B. Balasingam, K. R. Pattipati, and Y. Bar-Shalom, "A battery chemistry-adaptive fuel gauge using probabilistic data association," *Journal of Power Sources*, vol. 273, pp. 185–195, Jan. 2015.
- [25] C. Y. Chun, J. Baek, G. S. Seo, B. H. Cho, J. Kim, I. K. Chang, and S. Lee, "Current sensor-less state-of-charge estimation algorithm for lithium-ion batteries utilizing filtered terminal voltage," *Journal of Power Sources*, vol. 273, pp. 255–263, Jan. 2015.
- [26] C. Y. Chun, B. H. Cho, and J. Kim, "Covariance controlled state-of-charge estimator of LiFePO₄ cells using a simplified hysteresis model," *Electrochimica Acta*, vol. 265, pp. 629–637, Mar. 2018.
- [27] X. Dang, L. Yan, K. Xu, X. Wu, H. Jiang, and H. Sun, "Open-Circuit Voltage-Based State of Charge Estimation of Lithium-ion Battery Using Dual Neural Network Fusion Battery Model," *Electrochimica Acta*, vol. 188, pp. 356–366, Jan. 2016.
- [28] G. Dong, J. Wei, C. Zhang, and Z. Chen, "Online state of charge estimation and open circuit voltage hysteresis modeling of LiFePO₄ battery using invariant imbedding method," *Applied Energy*, vol. 162, pp. 163–171, Jan. 2016.
- [29] C. Huang and L. Wang, "Gaussian process regression-based modelling of lithium-ion battery temperature-dependent open-circuit-voltage," *Electronics Letters*, vol. 53, no. 17, pp. 1214–1216, Jul. 2017.
- [30] Z. Li, R. Xiong, and H. He, "An Improved Battery On-line Parameter Identification and State-of-charge Determining Method," in *Energy Procedia*, vol. 103, Dec. 2016, pp. 381–386.
- [31] K. C. Lim, H. A. Bastawrous, V. H. Duong, K. W. See, P. Zhang, and S. X. Dou, "Fading Kalman filter-based real-time state of charge estimation in LiFePO₄ battery-powered electric vehicles," *Applied Energy*, vol. 169, pp. 40–48, May 2016.
- [32] H. Pan, Z. Lü, W. Lin, J. Li, and L. Chen, "State of charge estimation of lithium-ion batteries using a grey extended Kalman filter and a novel open-circuit voltage model," *Energy*, vol. 138, pp. 764–775, Nov. 2017.
- [33] G. Pérez, M. Garmendia, J. F. Reynaud, J. Crego, and U. Viscarret, "Enhanced closed loop State of Charge estimator for lithium-ion batteries based on Extended Kalman Filter," *Applied Energy*, vol. 155, pp. 834–845, Oct. 2015.

-
- [34] H. Rahimi-Eichi, F. Baronti, and M.-Y. Chow, "Online Adaptive Parameter Identification and State-of-Charge Coestimation for Lithium-Polymer Battery Cells," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 2053–2061, Apr. 2014.
 - [35] S. Tong, M. P. Klein, and J. W. Park, "On-line optimization of battery open circuit voltage for improved state-of-charge and state-of-health estimation," *Journal of Power Sources*, vol. 293, pp. 416–428, Oct. 2015.
 - [36] C. Truchot, M. Dubarry, and B. Y. Liaw, "State-of-charge estimation and uncertainty for lithium-ion battery strings," *Applied Energy*, vol. 119, pp. 218–227, Apr. 2014.
 - [37] Z. Wei, B. Xiong, D. Ji, and K. J. Tseng, "Online State of Charge and Capacity Dual Estimation with a Multi-timescale Estimator for Lithium-ion Battery," in *Energy Procedia*, vol. 105, May 2017, pp. 2953–2958.
 - [38] C. Weng, J. Sun, and H. Peng, "A unified open-circuit-voltage model of lithium-ion batteries for state-of-charge estimation and state-of-health monitoring," *Journal of Power Sources*, vol. 258, pp. 228–237, Jul. 2014.
 - [39] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, Jan. 2014.
 - [40] R. Xiong, Q. Yu, L. Y. Wang, and C. Lin, "A novel method to obtain the open circuit voltage for the state of charge of lithium ion batteries in electric vehicles by using H infinity filter," *Applied Energy*, vol. 207, pp. 346–353, Dec. 2017.
 - [41] R. Yang, R. Xiong, H. He, H. Mu, and C. Wang, "A novel method on estimating the degradation and state of charge of lithium-ion batteries used for electrical vehicles," *Applied Energy*, vol. 207, pp. 336–345, Dec. 2017.
 - [42] L. Zhu, Z. Sun, H. Dai, and X. Wei, "A novel modeling methodology of open circuit voltage hysteresis for LiFePO₄batteries based on an adaptive discrete Preisach model," *Applied Energy*, vol. 155, pp. 91–109, Oct. 2015.
 - [43] F. Leng, C. M. Tan, R. Yazami, and M. D. Le, "A practical framework of electrical based online state-of-charge estimation of lithium ion batteries," *Journal of Power Sources*, vol. 255, pp. 423–430, Jun. 2014.
 - [44] H. Mu, R. Xiong, and F. Sun, "A novel multi-model probability based battery stateofcharge fusion estimation approach," in *Energy Procedia*, vol. 88, Jun. 2016, pp. 840–846.
 - [45] S. Muhammad, M. U. Rafique, S. Li, Z. Shao, Q. Wang, and N. Guan, "A Robust Algorithm for State-of-Charge Estimation With Gain Optimization," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2983–2994, Dec. 2017.
 - [46] Y. Tian, D. Li, J. Tian, and B. Xia, "State of charge estimation of lithium-ion batteries using an optimal adaptive gain nonlinear observer," *Electrochimica Acta*, vol. 225, pp. 225–234, Jan. 2017.
 - [47] W. Wang, H. Shu-Hung Chung, and J. Zhang, "Near-Real-Time Parameter Estimation of an Electrical Battery Model With Multiple Time Constants and SOC-Dependent Capacitance," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 5905–5920, Nov. 2014.
 - [48] Y. Wang, D. Yang, X. Zhang, and Z. Chen, "Probability based remaining capacity estimation using data-driven and neural network model," *Journal of Power Sources*, vol. 315, pp. 199–208, May 2016.
 - [49] G. Yang, "Battery parameterisation based on differential evolution via a boundary evolution strategy," *Journal of Power Sources*, vol. 245, pp. 583–593, Jan. 2014.
 - [50] H. Aung, K. Soon Low, and S. Ting Goh, "State-of-Charge Estimation of Lithium-Ion Battery Using Square Root Spherical Unscented Kalman Filter (Sqrt-UKFST) in Nanosatellite," *IEEE Transactions on Power Electronics*, vol. 30, no. 9, pp. 4774–4783, Sep. 2015.

- [51] G. Bai, P. Wang, C. Hu, and M. Pecht, "A generic model-free approach for lithium-ion battery health management," *Applied Energy*, vol. 135, pp. 247–260, Dec. 2014.
- [52] H. Chaoui, A. El Mejdoubi, and H. Gualous, "Online Parameter Identification of Lithium-Ion Batteries With Surface Temperature Variations," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2000–2009, Mar. 2017.
- [53] M. Charkhgard and M. H. Zarif, "Design of adaptive H_∞ filter for implementing on state-of-charge estimation based on battery state-of-charge-varying modelling," *IET Power Electronics*, vol. 8, no. 10, pp. 1825–1833, Oct. 2015.
- [54] C. Chen, F. Sun, R. Xiong, and H. He, "A Novel Dual H Infinity Filters Based Battery Parameter and State Estimation Approach for Electric Vehicles Application," in *Energy Procedia*, vol. 103, Dec. 2016, pp. 375–380.
- [55] C. Chen, R. Xiong, and W. Shen, "A Lithium-Ion Battery-in-the-Loop Approach to Test and Validate Multiscale Dual H Infinity Filters for State-of-Charge and Capacity Estimation," *IEEE Transactions on Power Electronics*, vol. 33, no. 1, pp. 332–342, Jan. 2018.
- [56] X. Cheng, S. Wang, and L. Yao, "Li-ion Battery Pack State-of-Charge Estimation Disturbed by Colored Noises," in *Energy Procedia*, vol. 105, May 2017, pp. 4104–4109.
- [57] H. Dai, P. Guo, X. Wei, Z. Sun, and J. Wang, "ANFIS (adaptive neuro-fuzzy inference system) based online SOC (State of Charge) correction considering cell divergence for the EV (electric vehicle) traction batteries," *Energy*, vol. 80, pp. 350–360, Feb. 2015.
- [58] G. Dong, Z. Chen, J. Wei, C. Zhang, and P. Wang, "An online model-based method for state of energy estimation of lithium-ion batteries using dual filters," *Journal of Power Sources*, vol. 301, pp. 277–286, Jan. 2016.
- [59] G. Dong, J. Wei, and Z. Chen, "Kalman filter for onboard state of charge estimation and peak power capability analysis of lithium-ion batteries," *Journal of Power Sources*, vol. 328, pp. 615–626, Oct. 2016.
- [60] G. Dong, J. Wei, Z. Chen, H. Sun, and X. Yu, "Remaining dischargeable time prediction for lithium-ion batteries using unscented Kalman filter," *Journal of Power Sources*, vol. 364, pp. 316–327, Oct. 2017.
- [61] G. Dong, X. Zhang, C. Zhang, and Z. Chen, "A method for state of energy estimation of lithium-ion batteries based on neural network model," *Energy*, vol. 90, pp. 879–888, Oct. 2015.
- [62] H. Fang, X. Zhao, Y. Wang, Z. Sahinoglu, T. Wada, S. Hara, and R. A. De Callafon, "Improved adaptive state-of-charge estimation for batteries using a multi-model approach," *Journal of Power Sources*, vol. 254, pp. 258–267, May 2014.
- [63] B. Fridholm, M. Nilsson, and T. Wik, "Robustness comparison of battery state of charge observers for automotive applications," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, 2014, pp. 2138–2146.
- [64] W. Gao, Y. Zou, F. Sun, X. Hu, Y. Yu, and S. Feng, "Data pieces-based parameter identification for lithium-ion battery," *Journal of Power Sources*, vol. 328, pp. 174–184, Oct. 2016.
- [65] Y. Guo, Z. Zhao, and L. Huang, "SoC Estimation of Lithium Battery Based on AEKF Algorithm," in *Energy Procedia*, vol. 105, May 2017, pp. 4146–4152.
- [66] M. Haddad Zarif, M. Charkhgard, and A. Alfi, "Hybrid state of charge estimation for lithium-ion batteries: design and implementation," *IET Power Electronics*, vol. 7, no. 11, pp. 2758–2764, Nov. 2014.
- [67] H. He, H. Qin, Y. Shui, and K. Oleksandr, "Lithium-ion battery SoC estimation with UKF and RTOS μ COS-II platform," in *Energy Procedia*, vol. 61, 2014, pp. 468–471.
- [68] H. W. He, Y. Z. Zhang, R. Xiong, and C. Wang, "A novel Gaussian model based battery state estimation approach: State-of-Energy," *Applied Energy*, vol. 151, pp. 41–48, Aug. 2015.

-
- [69] T. Huria, G. Ludovici, and G. Lutzemberger, "State of charge estimation of high power lithium iron phosphate cells," *Journal of Power Sources*, vol. 249, pp. 92–102, Mar. 2014.
 - [70] M. R. Khan, G. Mulder, and J. Van Mierlo, "An online framework for state of charge determination of battery systems using combined system identification approach," *Journal of Power Sources*, vol. 246, pp. 629–641, Jan. 2014.
 - [71] S. Lee and J. Kim, "Discrete wavelet transform-based denoising technique for advanced state-of-charge estimator of a lithium-ion battery in electric vehicles," *Energy*, vol. 83, pp. 462–473, Apr. 2015.
 - [72] D. Li, J. Ouyang, H. Li, and J. Wan, "State of charge estimation for LiMn₂O₄ power battery based on strong tracking sigma point Kalman filter," *Journal of Power Sources*, vol. 279, pp. 439–449, Apr. 2015.
 - [73] J. Li, J. Klee Barillas, C. Guenther, and M. A. Danzer, "Sequential Monte Carlo filter for state estimation of LiFePO₄ batteries based on an online updated model," *Journal of Power Sources*, vol. 247, pp. 156–162, Feb. 2014.
 - [74] Y. Li, P. Chattopadhyay, S. Xiong, A. Ray, and C. D. Rahn, "Dynamic data-driven and model-based recursive analysis for estimation of battery state-of-charge," *Applied Energy*, vol. 184, pp. 266–275, Dec. 2016.
 - [75] Y. Li, C. Wang, and J. Gong, "A combination Kalman filter approach for State of Charge estimation of lithium-ion battery considering model uncertainty," *Energy*, vol. 109, pp. 933–946, Aug. 2016.
 - [76] —, "A multi-model probability SOC fusion estimation approach using an improved adaptive unscented Kalman filter technique," *Energy*, vol. 141, pp. 1402–1415, Dec. 2017.
 - [77] C. Lin, H. Mu, R. Xiong, and J. Cao, "Multi-model probabilities based state fusion estimation method of lithium-ion battery for electric vehicles: State-of-energy," *Applied Energy*, vol. 194, pp. 560–568, May 2017.
 - [78] C. Lin, H. Mu, R. Xiong, and W. Shen, "A novel multi-model probability battery state of charge estimation approach for electric vehicles using H-infinity algorithm," *Applied Energy*, vol. 166, pp. 76–83, Mar. 2016.
 - [79] C. Liu, W. Liu, L. Wang, G. Hu, L. Ma, and B. Ren, "A new method of modeling and state of charge estimation of the battery," *Journal of Power Sources*, vol. 320, pp. 1–12, Jul. 2016.
 - [80] C.-z. Liu, Q. Zhu, L. Li, W.-q. Liu, L.-Y. Wang, N. Xiong, and X.-y. Wang, "A State of Charge Estimation Method Based on H_∞ Observer for Switched Systems of Lithium-Ion Nickel–Manganese–Cobalt Batteries," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8128–8137, Oct. 2017.
 - [81] X. Liu, Z. Chen, C. Zhang, and J. Wu, "A novel temperature-compensated model for power Li-ion batteries with dual-particle-filter state of charge estimation," *Applied Energy*, vol. 123, pp. 263–272, Jun. 2014.
 - [82] J. Meng, G. Luo, and F. Gao, "Lithium Polymer Battery State-of-Charge Estimation Based on Adaptive Unscented Kalman Filter and Support Vector Machine," *IEEE Transactions on Power Electronics*, vol. 31, no. 3, pp. 2226–2238, Mar. 2016.
 - [83] M. Ouyang, G. Liu, L. Lu, J. Li, and X. Han, "Enhancing the estimation accuracy in low state-of-charge area: A novel onboard battery model through surface state of charge determination," *Journal of Power Sources*, vol. 270, pp. 221–237, Dec. 2014.
 - [84] M. 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*, vol. 23, no. 1, pp. 357–363, Jan. 2015.

- [85] M. Paschero, G. L. Storti, A. Rizzi, F. M. F. Mascioli, and G. Rizzoni, "A Novel Mechanical Analogy-Based Battery Model for SoC Estimation Using a Multicell EKF," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 4, pp. 1695–1702, Oct. 2016.
- [86] S. Peng, C. Chen, H. Shi, and Z. Yao, "State of Charge Estimation of Battery Energy Storage Systems Based on Adaptive Unscented Kalman Filter With a Noise Statistics Estimator," *IEEE Access*, vol. 5, pp. 13 202–13 212, 2017.
- [87] D. A. Pola, H. F. Navarrete, M. E. Orchard, R. S. Rabie, M. A. Cerda, B. E. Olivares, J. F. Silva, P. A. Espinoza, and A. Perez, "Particle-Filtering-Based Discharge Time Prognosis for Lithium-Ion Batteries With a Statistical Characterization of Use Profiles," *IEEE Transactions on Reliability*, vol. 64, no. 2, pp. 710–720, Jun. 2015.
- [88] K. Propp, D. J. Auger, A. Fotouhi, S. Longo, and V. Knap, "Kalman-variant estimators for state of charge in lithium-sulfur batteries," *Journal of Power Sources*, vol. 343, pp. 254–267, Mar. 2017.
- [89] S. Sepasi, R. Ghorbani, and B. Y. Liaw, "Improved extended Kalman filter for state of charge estimation of battery pack," *Journal of Power Sources*, vol. 255, pp. 368–376, Jun. 2014.
- [90] —, "A novel on-board state-of-charge estimation method for aged Li-ion batteries based on model adaptive extended Kalman filter," *Journal of Power Sources*, vol. 245, pp. 337–344, Jan. 2014.
- [91] M. Shehab El Din, M. F. Abdel-Hafez, and A. A. Hussein, "Enhancement in Li-Ion Battery Cell State-of-Charge Estimation Under Uncertain Model Statistics," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4608–4618, Jun. 2016.
- [92] P. Shen, M. Ouyang, L. Lu, J. Li, and X. Feng, "The Co-estimation of State of Charge, State of Health, and State of Function for Lithium-Ion Batteries in Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 92–103, Jan. 2018.
- [93] Y. Shen, "A chaos genetic algorithm based extended Kalman filter for the available capacity evaluation of lithium-ion batteries," *Electrochimica Acta*, vol. 264, pp. 400–409, Feb. 2018.
- [94] —, "Hybrid unscented particle filter based state-of-charge determination for lead-acid batteries," *Energy*, vol. 74, no. C, pp. 795–803, Sep. 2014.
- [95] F. Sun, R. Xiong, and H. He, "Estimation of state-of-charge and state-of-power capability of lithium-ion battery considering varying health conditions," *Journal of Power Sources*, vol. 259, pp. 166–176, Aug. 2014.
- [96] Y. Tian, B. Xia, W. Sun, Z. Xu, and W. Zheng, "A modified model based state of charge estimation of power lithium-ion batteries using unscented Kalman filter," *Journal of Power Sources*, vol. 270, pp. 619–626, Dec. 2014.
- [97] Y. Wang, H. Fang, Z. Sahinoglu, T. Wada, and S. Hara, "Adaptive Estimation of the State of Charge for Lithium-Ion Batteries: Nonlinear Geometric Observer Approach," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 948–962, May 2015.
- [98] Y. Wang, C. Liu, R. Pan, and Z. Chen, "Modeling and state-of-charge prediction of lithium-ion battery and ultracapacitor hybrids with a co-estimator," *Energy*, vol. 121, pp. 739–750, Feb. 2017.
- [99] Y. Wang, C. Zhang, and Z. Chen, "A method for state-of-charge estimation of Li-ion batteries based on multi-model switching strategy," *Applied Energy*, vol. 137, pp. 427–434, Jan. 2015.
- [100] —, "Model-based state-of-energy estimation of lithium-ion batteries in electric vehicles," in *Energy Procedia*, vol. 88, Jun. 2016, pp. 998–1004.
- [101] —, "A method for joint estimation of state-of-charge and available energy of LiFePO₄ batteries," *Applied Energy*, vol. 135, pp. 81–87, Dec. 2014.
- [102] —, "State-of-charge Estimation of Lithium-ion Batteries Based on Multiple Filters Method," in *Energy Procedia*, vol. 75, Aug. 2015, pp. 2635–2640.

-
- [103] —, “A method for state-of-charge estimation of LiFePO₄batteries at dynamic currents and temperatures using particle filter,” *Journal of Power Sources*, vol. 279, pp. 306–311, Apr. 2015.
 - [104] J. Wei, G. Dong, and Z. Chen, “On-board adaptive model for state of charge estimation of lithium-ion batteries based on Kalman filter with proportional integral-based error adjustment,” *Journal of Power Sources*, vol. 365, pp. 308–319, Oct. 2017.
 - [105] Z. Wei, J. Zhao, D. Ji, and K. J. Tseng, “A multi-timescale estimator for battery state of charge and capacity dual estimation based on an online identified model,” *Applied Energy*, vol. 204, pp. 1264–1274, Oct. 2017.
 - [106] S. Wijewardana, R. Vepa, and M. H. Shaheed, “Dynamic battery cell model and state of charge estimation,” *Journal of Power Sources*, vol. 308, pp. 109–120, Mar. 2016.
 - [107] S. Xie, R. Xiong, Y. Zhang, and H. He, “The Estimation of State of Charge for Power Battery Packs used in Hybrid Electric Vehicle,” in *Energy Procedia*, vol. 105, May 2017, pp. 2678–2683.
 - [108] R. Xiong, F. Sun, Z. Chen, and H. He, “A data-driven multi-scale extended Kalman filtering based parameter and state estimation approach of lithium-ion polymer battery in electric vehicles,” *Applied Energy*, vol. 113, pp. 463–476, Jan. 2014.
 - [109] R. Xiong, Y. Zhang, H. He, X. Zhou, and M. G. Pecht, “A Double-Scale, Particle-Filtering, Energy State Prediction Algorithm for Lithium-Ion Batteries,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1526–1538, Feb. 2018.
 - [110] F. Yang, Y. Xing, D. Wang, and K. L. Tsui, “A comparative study of three model-based algorithms for estimating state-of-charge of lithium-ion batteries under a new combined dynamic loading profile,” *Applied Energy*, vol. 164, pp. 387–399, Feb. 2016.
 - [111] J. Yang, B. Xia, Y. Shang, W. Huang, and C. C. Mi, “Adaptive State-of-Charge Estimation Based on a Split Battery Model for Electric Vehicle Applications,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 889–10 898, Dec. 2017.
 - [112] J. Ye, A. Emadi, R. Gu, P. Malysz, and H. Yang, “State-of-charge and state-of-health estimation with state constraints and current sensor bias correction for electrified powertrain vehicle batteries,” *IET Electrical Systems in Transportation*, vol. 6, no. 2, pp. 136–144, Jun. 2016.
 - [113] M. Ye, H. Guo, and B. Cao, “A model-based adaptive state of charge estimator for a lithium-ion battery using an improved adaptive particle filter,” *Applied Energy*, vol. 190, pp. 740–748, Mar. 2017.
 - [114] M. Ye, H. Guo, R. Xiong, and H. Mu, “An Online Model-based Battery Parameter and State Estimation Method Using Multi-scale Dual Adaptive Particle Filters,” in *Energy Procedia*, vol. 105, May 2017, pp. 4549–4554.
 - [115] M. Ye, H. Guo, R. Xiong, and R. Yang, “Model-based State-of-charge Estimation Approach of the Lithium-ion Battery Using an Improved Adaptive Particle Filter,” in *Energy Procedia*, vol. 103, Dec. 2016, pp. 394–399.
 - [116] M. Ye, H. Guo, R. Xiong, and Q. Yu, “A double-scale and adaptive particle filter-based online parameter and state of charge estimation method for lithium-ion batteries,” *Energy*, vol. 144, pp. 789–799, Feb. 2018.
 - [117] Q. Yu, R. Xiong, and C. Lin, “Online Estimation of State-of-charge Based on the H infinity and Unscented Kalman Filters for Lithium Ion Batteries,” in *Energy Procedia*, vol. 105, May 2017, pp. 2791–2796.
 - [118] Q. Yu, R. Xiong, C. Lin, W. Shen, and J. Deng, “Lithium-Ion Battery Parameters and State-of-Charge Joint Estimation Based on H-Infinity and Unscented Kalman Filters,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8693–8701, Oct. 2017.
 - [119] C. Zhang, K. Li, J. Deng, and S. Song, “Improved Realtime State-of-Charge Estimation of LiFePO₄ Battery Based on a Novel Thermoelectric Model,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 654–663, Jan. 2017.

- [120] C. Zhang, Y. Zhang, and X. Zhang, "State-of-charge estimation of the lithium-ion battery system with time-varying parameter for hybrid electric vehicles," *IET Control Theory & Applications*, vol. 8, no. 3, pp. 160–167, Feb. 2014.
- [121] W. Zhang, W. Shi, and Z. Ma, "Adaptive unscented Kalman filter based state of energy and power capability estimation approach for lithium-ion battery," *Journal of Power Sources*, vol. 289, pp. 50–62, Sep. 2015.
- [122] X. Zhang, Y. Wang, C. Liu, and Z. Chen, "A novel approach of remaining discharge energy prediction for large format lithium-ion battery pack," *Journal of Power Sources*, vol. 343, pp. 216–225, Mar. 2017.
- [123] X. Zhang, Y. Wang, J. Wu, and Z. Chen, "A novel method for lithium-ion battery state of energy and state of power estimation based on multi-time-scale filter," *Applied Energy*, vol. 216, pp. 442–451, Apr. 2018.
- [124] X. Zhang, Y. Wang, D. Yang, and Z. Chen, "An on-line estimation of battery pack parameters and state-of-charge using dual filters based on pack model," *Energy*, vol. 115, pp. 219–229, Nov. 2016.
- [125] Y. Zhang, R. Xiong, H. He, and W. 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*, vol. 32, no. 6, pp. 4421–4431, Jun. 2017.
- [126] Y. Z. Zhang, H. W. He, and R. Xiong, "A Data-Driven Based State of Energy Estimator of Lithium-ion Batteries Used to Supply Electric Vehicles," in *Energy Procedia*, vol. 75, Aug. 2015, pp. 1944–1949.
- [127] Z. Zhang, X. Cheng, Z.-Y. Lu, and D.-J. Gu, "SOC Estimation of Lithium-Ion Battery Pack Considering Balancing Current," *IEEE Transactions on Power Electronics*, vol. 33, no. 3, pp. 2216–2226, Mar. 2018.
- [128] Z.-L. Zhang, X. Cheng, Z.-Y. Lu, and D.-J. Gu, "SOC Estimation of Lithium-Ion Batteries With AEKF and Wavelet Transform Matrix," *IEEE Transactions on Power Electronics*, vol. 32, no. 10, pp. 7626–7634, Oct. 2017.
- [129] S. Zhao, S. R. Duncan, and D. A. Howey, "Observability Analysis and State Estimation of Lithium-Ion Batteries in the Presence of Sensor Biases," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 326–333, Jan. 2017.
- [130] X. Zhao, Y. Cai, L. Yang, Z. Deng, and J. Qiang, "State of charge estimation based on a new dual-polarization-resistance model for electric vehicles," *Energy*, vol. 135, pp. 40–52, Sep. 2017.
- [131] Y. Zheng, W. Gao, M. Ouyang, L. Lu, L. Zhou, and X. Han, "State-of-charge inconsistency estimation of lithium-ion battery pack using mean-difference model and extended Kalman filter," *Journal of Power Sources*, vol. 383, pp. 50–58, Apr. 2018.
- [132] L. Zhong, C. Zhang, Y. He, and Z. Chen, "A method for the estimation of the battery pack state of charge based on in-pack cells uniformity analysis," *Applied Energy*, vol. 113, pp. 558–564, Jan. 2014.
- [133] D. Zhou, K. Zhang, A. Ravey, F. Gao, and A. Miraoui, "Online Estimation of Lithium Polymer Batteries State-of-Charge Using Particle Filter-Based Data Fusion With Multimodels Approach," *IEEE Transactions on Industry Applications*, vol. 52, no. 3, pp. 2582–2595, May 2016.
- [134] Q. Zhu, L. Li, X. Hu, N. Xiong, and G.-D. Hu, "H ∞ -Based Nonlinear Observer Design for State of Charge Estimation of Lithium-Ion Battery With Polynomial Parameters," *IEEE Transactions on Vehicular Technology*, no. 12, pp. 10 853–10 865, Dec. 2017.
- [135] Y. Zou, X. Hu, H. Ma, and S. E. Li, "Combined State of Charge and State of Health estimation over lithium-ion battery cell cycle lifespan for electric vehicles," *Journal of Power Sources*, vol. 273, pp. 793–803, Jan. 2015.

-
- [136] M. Cacciato, G. Nobile, G. Scarcella, and G. Scelba, "Real-Time Model-Based Estimation of SOC and SOH for Energy Storage Systems," *IEEE Transactions on Power Electronics*, vol. 32, no. 1, pp. 794–803, Jan. 2017.
 - [137] H. Chaoui and H. Gualous, "Adaptive State of Charge Estimation of Lithium-Ion Batteries With Parameter and Thermal Uncertainties," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 752–759, Mar. 2017.
 - [138] Q. Chen, J. Jiang, H. Ruan, and C. Zhang, "Simply designed and universal sliding mode observer for the SOC estimation of lithium-ion batteries," *IET Power Electronics*, vol. 10, no. 6, pp. 697–705, May 2017.
 - [139] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "A novel approach for state of charge estimation based on adaptive switching gain sliding mode observer in electric vehicles," *Journal of Power Sources*, vol. 246, pp. 667–678, Jan. 2014.
 - [140] X. Chen, W. Shen, M. Dai, Z. Cao, J. Jin, and A. Kapoor, "Robust Adaptive Sliding-Mode Observer Using RBF Neural Network for Lithium-Ion Battery State of Charge Estimation in Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 1936–1947, Apr. 2016.
 - [141] M. Gholizadeh and F. R. Salmasi, "Estimation of State of Charge, Unknown Nonlinearities, and State of Health of a Lithium-Ion Battery Based on a Comprehensive Unobservable Model," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1335–1344, Mar. 2014.
 - [142] W. Li, L. Liang, W. Liu, and X. Wu, "State of Charge Estimation of Lithium-Ion Batteries Using a Discrete-Time Nonlinear Observer," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 11, pp. 8557–8565, Nov. 2017.
 - [143] Y. Ma, B. Li, G. Li, J. Zhang, and H. Chen, "A nonlinear observer approach of SOC estimation based on hysteresis model for lithium-ion battery," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 2, pp. 195–204, Apr. 2017.
 - [144] B. Ning, J. Xu, B. Cao, B. Wang, and G. Xu, "A sliding mode observer SOC estimation method based on parameter adaptive battery model," in *Energy Procedia*, vol. 88, Jun. 2016, pp. 619–626.
 - [145] X. Tang, B. Liu, and F. Gao, "State of Charge Estimation of LiFePO₄ Battery Based on a Gain-classifier Observer," in *Energy Procedia*, vol. 105, May 2017, pp. 2071–2076.
 - [146] X. Tang, B. Liu, Z. Lv, and F. Gao, "Observer based battery SOC estimation: Using multi-gain-switching approach," *Applied Energy*, vol. 204, pp. 1275–1283, Oct. 2017.
 - [147] X. Tang, Y. Wang, and Z. Chen, "A method for state-of-charge estimation of LiFePO₄ batteries based on a dual-circuit state observer," *Journal of Power Sources*, vol. 296, pp. 23–29, Nov. 2015.
 - [148] B. Xia, C. Chen, Y. Tian, W. Sun, Z. Xu, and W. Zheng, "A novel method for state of charge estimation of lithium-ion batteries using a nonlinear observer," *Journal of Power Sources*, vol. 270, pp. 359–366, Dec. 2014.
 - [149] Jun Xu, C. C. Mi, Binggang Cao, Junjun Deng, Zheng Chen, and Siqi Li, "The State of Charge Estimation of Lithium-Ion Batteries Based on a Proportional-Integral Observer," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1614–1621, May 2014.
 - [150] W. Allafi, K. Uddin, C. Zhang, R. Mazuir Raja Ahsan Sha, and J. Marco, "On-line scheme for parameter estimation of nonlinear lithium ion battery equivalent circuit models using the simplified refined instrumental variable method for a modified Wiener continuous-time model," *Applied Energy*, vol. 204, pp. 497–508, Oct. 2017.
 - [151] B. Balasingam, G. V. Avvari, B. Pattipati, K. R. Pattipati, and Y. Bar-Shalom, "A robust approach to battery fuel gauging, part I: Real time model identification," *Journal of Power Sources*, vol. 272, pp. 1142–1153, Dec. 2014.

- [152] V. H. Duong, H. A. Bastawrous, K. C. Lim, K. W. See, P. Zhang, and S. X. 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*, vol. 296, pp. 215–224, Nov. 2015.
- [153] V. H. Duong, H. A. Bastawrous, and K. W. See, "Accurate approach to the temperature effect on state of charge estimation in the LiFePO₄ battery under dynamic load operation," *Applied Energy*, vol. 204, pp. 560–571, Oct. 2017.
- [154] Y. Hu and Y.-Y. Wang, "Two Time-Scaled Battery Model Identification With Application to Battery State Estimation," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1180–1188, May 2015.
- [155] J. Li, J. Klee Barillas, C. Guenther, and M. A. Danzer, "Multicell state estimation using variation based sequential Monte Carlo filter for automotive battery packs," *Journal of Power Sources*, vol. 277, pp. 95–103, Mar. 2015.
- [156] Z. Li, R. Xiong, H. Mu, H. He, and C. Wang, "A novel parameter and state-of-charge determining method of lithium-ion battery for electric vehicles," *Applied Energy*, vol. 207, pp. 363–371, Dec. 2017.
- [157] C. Unterrieder, C. Zhang, M. Lunglmayr, R. Priewasser, S. Marsili, and M. Huemer, "Battery state-of-charge estimation using approximate least squares," *Journal of Power Sources*, vol. 278, pp. 274–286, Mar. 2015.
- [158] Y. Wang, C. Zhang, and Z. Chen, "An adaptive remaining energy prediction approach for lithium-ion batteries in electric vehicles," *Journal of Power Sources*, vol. 305, pp. 80–88, Feb. 2016.
- [159] Z. Wei, S. Meng, B. Xiong, D. Ji, and K. J. Tseng, "Enhanced online model identification and state of charge estimation for lithium-ion battery with a FBCRLS based observer," *Applied Energy*, vol. 181, pp. 332–341, Nov. 2016.
- [160] Z. Wei, C. Zou, F. Leng, B. H. Soong, and K.-J. Tseng, "Online Model Identification and State-of-Charge Estimate for Lithium-Ion Battery With a Recursive Total Least Squares-Based Observer," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1336–1346, Feb. 2018.
- [161] B. Xia, C. Chen, Y. Tian, M. Wang, W. Sun, and Z. Xu, "State of charge estimation of lithium-ion batteries based on an improved parameter identification method," *Energy*, vol. 90, pp. 1426–1434, Oct. 2015.
- [162] C. Zhang, W. Allafi, Q. Dinh, P. Ascencio, and J. Marco, "Online estimation of battery equivalent circuit model parameters and state of charge using decoupled least squares technique," *Energy*, vol. 142, pp. 678–688, Jan. 2018.
- [163] R. Ahmed, M. El Sayed, I. Arasaratnam, J. Tjong, and S. Habibi, "Reduced-Order Electrochemical Model Parameters Identification and State of Charge Estimation for Healthy and Aged Li-Ion Batteries—Part II: Aged Battery Model and State of Charge Estimation," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 678–690, Sep. 2014.
- [164] R. Ahmed, M. E. Sayed, I. Arasaratnam, J. Tjong, and S. Habibi, "Reduced-Order Electrochemical Model Parameters Identification and SOC Estimation for Healthy and Aged Li-Ion Batteries Part I: Parameterization Model Development for Healthy Batteries," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 659–677, Sep. 2014.
- [165] A. Bartlett, J. Marcicki, S. Onori, G. Rizzoni, X. G. Yang, and T. Miller, "Electrochemical Model-Based State of Charge and Capacity Estimation for a Composite Electrode Lithium-Ion Battery," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 1–1, 2015.
- [166] M. Corno, N. Bhatt, S. M. Savaresi, and M. Verhaegen, "Electrochemical Model-Based State of Charge Estimation for Li-Ion Cells," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 117–127, Jan. 2015.

-
- [167] S. Dey, B. Ayalew, and P. Pisu, "Nonlinear Robust Observers for State-of-Charge Estimation of Lithium-Ion Cells Based on a Reduced Electrochemical Model," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1935–1942, Sep. 2015.
 - [168] R. Gu, P. Malysz, H. Yang, and A. Emadi, "On the Suitability of Electrochemical-Based Modeling for Lithium-Ion Batteries," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 4, pp. 417–431, Dec. 2016.
 - [169] X. Han, M. Ouyang, L. Lu, and J. Li, "Simplification of physics-based electrochemical model for lithium ion battery on electric vehicle. Part II: Pseudo-two-dimensional model simplification and state of charge estimation," *Journal of Power Sources*, vol. 278, pp. 814–825, Mar. 2015.
 - [170] —, "Simplification of physics-based electrochemical model for lithium ion battery on electric vehicle. Part I: Diffusion simplification and single particle model," *Journal of Power Sources*, vol. 278, pp. 802–813, Mar. 2015. arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).
 - [171] A. Hasan and J. Jouffroy, "Infinite-Dimensional Boundary Observer for Lithium-Ion Battery State Estimation," in *Energy Procedia*, vol. 141, Dec. 2017, pp. 494–501.
 - [172] X. Hu, D. Cao, and B. Egardt, "Condition Monitoring in Advanced Battery Management Systems: Moving Horizon Estimation Using a Reduced Electrochemical Model," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 167–178, Feb. 2018.
 - [173] X. Li, G. Fan, K. Pan, G. Wei, C. Zhu, G. Rizzoni, and M. Canova, *A physics-based fractional order model and state of energy estimation for lithium ion batteries. Part I: Model development and observability analysis*, Nov. 2017.
 - [174] X. Li, K. Pan, G. Fan, R. Lu, C. Zhu, G. Rizzoni, and M. Canova, *A physics-based fractional order model and state of energy estimation for lithium ion batteries. Part II: Parameter identification and state of energy estimation for LiFePO₄ battery*, Nov. 2017.
 - [175] C. Lin, A. Tang, and J. Xing, "Evaluation of electrochemical models based battery state-of-charge estimation approaches for electric vehicles," *Applied Energy*, vol. 207, pp. 394–404, Dec. 2017.
 - [176] C. Lin, J. Xing, and A. Tang, "Lithium-ion Battery State of Charge/State of Health Estimation Using SMO for EVs," in *Energy Procedia*, vol. 105, May 2017, pp. 4383–4388.
 - [177] N. Lotfi, R. G. Landers, J. Li, and J. Park, "Reduced-Order Electrochemical Model-Based SOC Observer With Output Model Uncertainty Estimation," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1217–1230, Jul. 2017.
 - [178] S. J. Moura, F. B. Argomodo, R. Klein, A. Mirtabatabaei, and M. Krstic, "Battery State Estimation for a Single Particle Model With Electrolyte Dynamics," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 453–468, Mar. 2017.
 - [179] P. Tagade, K. S. Hariharan, P. Gambhire, S. M. Kolake, T. Song, D. Oh, T. Yeo, and S. Doo, "Recursive Bayesian filtering framework for lithium-ion cell state estimation," *Journal of Power Sources*, vol. 306, pp. 274–288, Feb. 2016.
 - [180] T. R. Tanim, C. D. Rahn, and C. Y. Wang, "State of charge estimation of a lithium ion cell based on a temperature dependent and electrolyte enhanced single particle model," *Energy*, vol. 80, pp. 731–739, Feb. 2015.
 - [181] A. Tulsyan, Y. Tsai, R. B. Gopaluni, and R. D. Braatz, "State-of-charge estimation in lithium-ion batteries: A particle filter approach," *Journal of Power Sources*, vol. 331, pp. 208–223, Nov. 2016.
 - [182] B. Wang, Z. Liu, S. E. Li, S. J. Moura, and H. Peng, "State-of-Charge Estimation for Lithium-Ion Batteries Based on a Nonlinear Fractional Model," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 3–11, Jan. 2017.
 - [183] Y. Wang, C. Zhang, and Z. Chen, "On-line battery state-of-charge estimation based on an integrated estimator," *Applied Energy*, vol. 185, pp. 2026–2032, Jan. 2017.

- [184] N. Yang, X. Zhang, and G. Li, "State-of-charge estimation for lithium ion batteries via the simulation of lithium distribution in the electrode particles," *Journal of Power Sources*, vol. 272, pp. 68–78, Dec. 2014.
- [185] X. Zhang, J. Lu, S. Yuan, J. Yang, and X. Zhou, "A novel method for identification of lithium-ion battery equivalent circuit model parameters considering electrochemical properties," *Journal of Power Sources*, vol. 345, pp. 21–29, Mar. 2017.
- [186] X. Zhao and R. A. de Callafon, "Modeling of battery dynamics and hysteresis for power delivery prediction and SOC estimation," *Applied Energy*, vol. 180, pp. 823–833, Oct. 2016.
- [187] L. Zheng, L. Zhang, J. Zhu, G. Wang, and J. Jiang, "Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model," *Applied Energy*, vol. 180, pp. 424–434, Oct. 2016.
- [188] C. Zou, X. Hu, S. Dey, L. Zhang, and X. Tang, "Nonlinear Fractional-Order Estimator with Guaranteed Robustness and Stability for Lithium-Ion Batteries," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 1–1, 2017.
- [189] C. Zou, C. Manzie, D. Nešić, and A. G. Kallapur, "Multi-time-scale observer design for state-of-charge and state-of-health of a lithium-ion battery," *Journal of Power Sources*, vol. 335, pp. 121–130, Dec. 2016.
- [190] J. Li, Q. Lai, L. Wang, C. Lyu, and H. Wang, "A method for SOC estimation based on simplified mechanistic model for LiFePO₄battery," *Energy*, vol. 114, pp. 1266–1276, Nov. 2016.
- [191] Y. Li, P. Chattopadhyay, and A. Ray, "Dynamic data-driven identification of battery state-of-charge via symbolic analysis of input-output pairs," *Applied Energy*, vol. 155, pp. 778–790, Oct. 2015.
- [192] M. H. Hung, C. H. Lin, L. C. Lee, and C. M. Wang, "State-of-charge and state-of-health estimation for lithium-ion batteries based on dynamic impedance technique," *Journal of Power Sources*, vol. 268, pp. 861–873, Dec. 2014.
- [193] Y. Ma, X. Zhou, B. Li, and H. Chen, "Fractional modeling and SOC estimation of lithium-ion battery," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 281–287, Jul. 2016.
- [194] H. Mu, R. Xiong, H. Zheng, Y. Chang, and Z. Chen, "A novel fractional order model based state-of-charge estimation method for lithium-ion battery," *Applied Energy*, vol. 207, pp. 384–393, Dec. 2017.
- [195] T. Stockley, K. Thanapalan, M. Bowkett, J. Williams, and M. Hathway, "Advanced EIS techniques for performance evaluation of Li-ion cells," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, 2014, pp. 8610–8615.
- [196] X. Hu, S. E. Li, and Y. Yang, "Advanced Machine Learning Approach for Lithium-Ion Battery State Estimation in Electric Vehicles," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 2, pp. 140–149, Jun. 2016.
- [197] H. Chaoui and C. C. Ibe-Ekeocha, "State of Charge and State of Health Estimation for Lithium Batteries Using Recurrent Neural Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8773–8783, Oct. 2017.
- [198] J. Chen, Q. Ouyang, C. Xu, and H. Su, "Neural Network-Based State of Charge Observer Design for Lithium-Ion Batteries," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 313–320, Jan. 2018.
- [199] M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad, and A. Ayob, "Neural Network Approach for Estimating State of Charge of Lithium-Ion Battery Using Backtracking Search Algorithm," *IEEE Access*, vol. 6, pp. 10 069–10 079, 2018.
- [200] A. A. Hussein, "Derivation and Comparison of Open-loop and Closed-loop Neural Network Battery State-of-Charge Estimators," in *Energy Procedia*, vol. 75, Aug. 2015, pp. 1856–1861.

-
- [201] L. W. Kang, X. Zhao, and J. Ma, "A new neural network model for the state-of-charge estimation in the battery degradation process," *Applied Energy*, vol. 121, pp. 20–27, May 2014.
 - [202] X. Liu, J. Wu, C. Zhang, and Z. Chen, "A method for state of energy estimation of lithium-ion batteries at dynamic currents and temperatures," *Journal of Power Sources*, vol. 270, pp. 151–157, Dec. 2014.
 - [203] F. Sun, R. Xiong, and H. He, "A systematic state-of-charge estimation framework for multi-cell battery pack in electric vehicles using bias correction technique," *Applied Energy*, vol. 162, pp. 1399–1409, Jan. 2016.
 - [204] Z. Deng, L. Yang, Y. Cai, and H. Deng, "Maximum Available Capacity and Energy Estimation Based on Support Vector Machine Regression for Lithium-ion Battery," in *Energy Procedia*, vol. 107, Feb. 2017, pp. 68–75.
 - [205] J. N. Hu, J. J. Hu, H. B. Lin, X. P. Li, C. L. Jiang, X. H. Qiu, and W. S. Li, "State-of-charge estimation for battery management system using optimized support vector machine for regression," *Journal of Power Sources*, vol. 269, pp. 682–693, Dec. 2014.
 - [206] H. Sheng and J. Xiao, "Electric vehicle state of charge estimation: Nonlinear correlation and fuzzy support vector machine," *Journal of Power Sources*, vol. 281, pp. 131–137, May 2015.
 - [207] J. C. Alvarez Anton, P. J. Garcia Nieto, E. Garcia Gonzalo, J. C. Viera Perez, M. Gonzalez Vega, and C. Blanco Viejo, "A New Predictive Model for the State-of-Charge of a High-Power Lithium-Ion Cell Based on a PSO-Optimized Multivariate Adaptive Regression Spline Approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4197–4208, Jun. 2016.
 - [208] G. O. Sahinoglu, M. Pajovic, Z. Sahinoglu, Y. Wang, P. V. Orlik, and T. Wada, "Battery State-of-Charge Estimation Based on Regular/Recurrent Gaussian Process Regression," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4311–4321, May 2018.
 - [209] H. Sheng, J. Xiao, and P. Wang, "Lithium Iron Phosphate Battery Electric Vehicle State-of-Charge Estimation Based on Evolutionary Gaussian Mixture Regression," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 544–551, Jan. 2017.
 - [210] J. Cannarella and C. B. Arnold, "State of health and charge measurements in lithium-ion batteries using mechanical stress," *Journal of Power Sources*, vol. 269, pp. 7–14, Dec. 2014.
 - [211] H. Dai, C. Yu, X. Wei, and Z. Sun, "State of charge estimation for lithium-ion pouch batteries based on stress measurement," *Energy*, vol. 129, pp. 16–27, Jun. 2017.
 - [212] A. El Mejdoubi, A. Oukaour, H. Chaoui, H. Gualous, J. Sabor, and Y. Slamani, "State-of-Charge and State-of-Health Lithium-Ion Batteries' Diagnosis According to Surface Temperature Variation," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2391–2402, Apr. 2016.
 - [213] A. Ganguli, B. Saha, A. Raghavan, P. Kiesel, K. Arakaki, A. Schuh, J. Schwartz, A. Hegyi, L. W. Sommer, A. Lochbaum, S. Sahu, and M. Alamgir, "Embedded fiber-optic sensing for accurate internal monitoring of cell state in advanced battery management systems part 2: Internal cell signals and utility for state estimation," *Journal of Power Sources*, vol. 341, pp. 474–482, Feb. 2017.
 - [214] L. Gold, T. Bach, W. Virsik, A. Schmitt, J. Müller, T. E. Staab, and G. Sextl, "Probing lithium-ion batteries' state-of-charge using ultrasonic transmission – Concept and laboratory testing," *Journal of Power Sources*, vol. 343, pp. 536–544, Mar. 2017.
 - [215] Xinfan Lin, A. G. Stefanopoulou, Yonghua Li, and R. D. Anderson, "State of Charge Imbalance Estimation for Battery Strings Under Reduced Voltage Sensing," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1052–1062, May 2015.
 - [216] A. Raghavan, P. Kiesel, L. W. Sommer, J. Schwartz, A. Lochbaum, A. Hegyi, A. Schuh, K. Arakaki, B. Saha, A. Ganguli, K. H. Kim, C. A. Kim, H. J. Hah, S. K. Kim, G. O. Hwang, G. C. Chung, B. Choi, and M. Alamgir, "Embedded fiber-optic sensing for accurate internal monitoring of cell state in advanced battery management systems part 1: Cell embedding method and performance," *Journal of Power Sources*, vol. 341, pp. 466–473, Feb. 2017.

- [217] Y. Hua, A. Cordoba-Arenas, N. Warner, and G. Rizzoni, "A multi time-scale state-of-charge and state-of-health estimation framework using nonlinear predictive filter for lithium-ion battery pack with passive balance control," *Journal of Power Sources*, vol. 280, pp. 293–312, Apr. 2015.
- [218] F. Sun and R. Xiong, "A novel dual-scale cell state-of-charge estimation approach for series-connected battery pack used in electric vehicles," *Journal of Power Sources*, vol. 274, pp. 582–594, Jan. 2015.
- [219] R. Xiong and H. He, "Cell state-of-charge estimation for the multi-cell seriesconnected battery pack with model bias correction approach," in *Energy Procedia*, vol. 61, 2014, pp. 172–175.
- [220] H. Wu, S. Yuan, X. Zhang, C. Yin, and X. Ma, "Model parameter estimation approach based on incremental analysis for lithium-ion batteries without using open circuit voltage," *Journal of Power Sources*, vol. 287, pp. 108–118, Aug. 2015.
- [221] L. Zheng, J. Zhu, D. D. C. Lu, G. Wang, and T. He, "Incremental capacity analysis and differential voltage analysis based state of charge and capacity estimation for lithium-ion batteries," *Energy*, vol. 150, pp. 759–769, Mar. 2018.
- [222] C. Fleischer, W. Waag, H. M. Heyn, and D. U. Sauer, "On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models part 2. Parameter and state estimation," *Journal of Power Sources*, vol. 262, pp. 457–482, Sep. 2014.
- [223] C. Fleischer, W. Waag, H.-M. Heyn, and D. U. Sauer, "On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models: Part 1. Requirements, critical review of methods and modeling," *Journal of Power Sources*, vol. 260, no. 2014, pp. 276–291, Aug. 2014.
- [224] J. Kalawoun, K. Biletska, F. Suard, and M. Montaru, *From a novel classification of the battery state of charge estimators toward a conception of an ideal one*, Apr. 2015.
- [225] J. Klee Barillas, J. Li, C. Günther, and M. A. Danzer, "A comparative study and validation of state estimation algorithms for Li-ion batteries in battery management systems," *Applied Energy*, vol. 155, pp. 455–462, Oct. 2015.
- [226] X. Lai, Y. Zheng, and T. Sun, "A comparative study of different equivalent circuit models for estimating state-of-charge of lithium-ion batteries," *Electrochimica Acta*, vol. 259, pp. 566–577, Jan. 2018.
- [227] Z. Li, J. Huang, B. Y. Liaw, and J. Zhang, *On state-of-charge determination for lithium-ion batteries*, Apr. 2017.
- [228] 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*, vol. 316, pp. 183–196, Jun. 2016.
- [229] R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun, "Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles," *IEEE Access*, vol. 6, pp. 1832–1843, 2018.
- [230] C. Zhang, L. Y. Wang, X. Li, W. Chen, G. G. Yin, and J. Jiang, "Robust and Adaptive Estimation of State of Charge for Lithium-Ion Batteries," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 4948–4957, Aug. 2015.
- [231] Y. Zheng, M. Ouyang, X. Han, L. Lu, and J. Li, *Investigating the error sources of the online state of charge estimation methods for lithium-ion batteries in electric vehicles*, Feb. 2018.
- [232] R. Relan, K. Tiels, J.-M. Timmermans, and J. Schoukens, "A Local Polynomial Approach to Non-parametric Estimation of the Best Linear Approximation of Lithium-Ion Battery From Multiple Datasets," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 182–187, Jul. 2017.
- [233] W. Huang and J. A. Qahouq, "An Online Battery Impedance Measurement Method Using DC–DC Power Converter Control," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 5987–5995, Nov. 2014.

-
- [234] W. D. Widanage, A. Barai, G. H. Chouchelamane, K. Uddin, A. McGordon, J. Marco, and P. Jennings, "Design and use of multisine signals for Li-ion battery equivalent circuit modelling. Part 2: Model estimation," *Journal of Power Sources*, vol. 324, pp. 61–69, Aug. 2016.
 - [235] J. Huang, Z. Li, and J. Zhang, "Dynamic electrochemical impedance spectroscopy reconstructed from continuous impedance measurement of single frequency during charging/discharging," *Journal of Power Sources*, vol. 273, pp. 1098–1102, Jan. 2015.
 - [236] J. Huang, Z. Li, B. Y. Liaw, and J. Zhang, "Graphical analysis of electrochemical impedance spectroscopy data in Bode and Nyquist representations," *Journal of Power Sources*, vol. 309, pp. 82–98, Mar. 2016.
 - [237] N. Wolff, N. Harting, M. Heinrich, F. Röder, and U. Krewer, "Nonlinear Frequency Response Analysis on Lithium-Ion Batteries: A Model-Based Assessment," *Electrochimica Acta*, vol. 260, pp. 614–622, Jan. 2018.
 - [238] D. A. Howey, P. D. Mitcheson, V. Yufit, G. J. Offer, and N. P. Brandon, "Online Measurement of Battery Impedance Using Motor Controller Excitation," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2557–2566, 2014.
 - [239] H. Piret, P. Granjon, N. Guillet, and V. Cattin, "Tracking of electrochemical impedance of batteries," *Journal of Power Sources*, vol. 312, pp. 60–69, Apr. 2016.
 - [240] Y. Jiang, B. Xia, X. Zhao, T. Nguyen, C. Mi, and R. A. de Callafon, "Data-based fractional differential models for non-linear dynamic modeling of a lithium-ion battery," *Energy*, vol. 135, pp. 171–181, Sep. 2017.
 - [241] Y. Zou, S. E. Li, B. Shao, and B. Wang, "State-space model with non-integer order derivatives for lithium-ion battery," *Applied Energy*, vol. 161, pp. 330–336, Jan. 2016.
 - [242] H. Dai, T. Xu, L. Zhu, X. Wei, and Z. Sun, "Adaptive model parameter identification for large capacity Li-ion batteries on separated time scales," *Applied Energy*, vol. 184, pp. 119–131, Dec. 2016.
 - [243] B. Fridholm, T. Wik, and M. Nilsson, "Robust recursive impedance estimation for automotive lithium-ion batteries," *Journal of Power Sources*, vol. 304, pp. 33–41, Feb. 2016.
 - [244] W.-J. Shen and H.-X. Li, "A Sensitivity-Based Group-Wise Parameter Identification Algorithm for the Electric Model of Li-Ion Battery," *IEEE Access*, vol. 5, pp. 4377–4387, 2017.
 - [245] A. Farmann, W. Waag, and D. U. Sauer, "Adaptive approach for on-board impedance parameters and voltage estimation of lithium-ion batteries in electric vehicles," *Journal of Power Sources*, vol. 299, pp. 176–188, Dec. 2015.
 - [246] J. Brand, Z. Zhang, and R. K. Agarwal, "Extraction of battery parameters of the equivalent circuit model using a multi-objective genetic algorithm," *Journal of Power Sources*, vol. 247, pp. 729–737, Feb. 2014.
 - [247] B. Wang, S. E. Li, H. Peng, and Z. Liu, "Fractional-order modeling and parameter identification for lithium-ion batteries," *Journal of Power Sources*, vol. 293, pp. 151–161, Oct. 2015.
 - [248] Z. Yu, L. Xiao, H. Li, X. Zhu, and R. Huai, "Model Parameter Identification for Lithium Batteries Using the Coevolutionary Particle Swarm Optimization Method," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5690–5700, Jul. 2017.
 - [249] A. Lievre, A. Sari, P. Venet, A. Hijazi, M. Ouattara-Brigaudet, and S. Pelissier, "Practical Online Estimation of Lithium-Ion Battery Apparent Series Resistance for Mild Hybrid Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4505–4511, Jun. 2016.
 - [250] S. M. Alavi, C. R. Birkel, and D. A. Howey, "Time-domain fitting of battery electrochemical impedance models," *Journal of Power Sources*, vol. 288, pp. 345–352, Aug. 2015.

- [251] S. Yuan, L. Jiang, C. Yin, H. Wu, and X. Zhang, “A transfer function type of simplified electrochemical model with modified boundary conditions and Padé approximation for Li-ion battery: Part 1. lithium concentration estimation,” *Journal of Power Sources*, vol. 352, pp. 245–257, Jun. 2017.
- [252] —, “A transfer function type of simplified electrochemical model with modified boundary conditions and Padé approximation for Li-ion battery: Part 2. Modeling and parameter estimation,” *Journal of Power Sources*, vol. 352, pp. 258–271, Jun. 2017.
- [253] M. Wang and H. X. Li, “Spatiotemporal modeling of internal states distribution for lithium-ion battery,” *Journal of Power Sources*, vol. 301, pp. 261–270, Jan. 2016.
- [254] T. R. Ashwin, A. McGordon, W. D. Widanage, and P. A. Jennings, “Modified electrochemical parameter estimation of NCR18650BD battery using implicit finite volume method,” *Journal of Power Sources*, vol. 341, pp. 387–395, Feb. 2017.
- [255] M. A. Rahman, S. Anwar, and A. Izadian, “Electrochemical model parameter identification of a lithium-ion battery using particle swarm optimization method,” *Journal of Power Sources*, vol. 307, pp. 86–97, Mar. 2016. arXiv: arXiv:1011.1669v3.
- [256] L. Zhang, L. Wang, G. Hinds, C. Lyu, J. Zheng, and J. Li, “Multi-objective optimization of lithium-ion battery model using genetic algorithm approach,” *Journal of Power Sources*, vol. 270, pp. 367–378, Dec. 2014.
- [257] X. Gong, R. Xiong, and C. Mi, “A data-driven bias correction method based lithiumion battery modeling approach for electric vehicle applications,” *IEEE Transactions on Industry Applications*, vol. 52, no. 2, pp. 1–1, 2015.
- [258] Y. Zhao and S. Y. Choe, “A highly efficient reduced order electrochemical model for a large format LiMn2O4/Carbon polymer battery for real time applications,” *Electrochimica Acta*, vol. 164, pp. 97–107, May 2015.
- [259] K. Y. Oh, N. A. Samad, Y. Kim, J. B. Siegel, A. G. Stefanopoulou, and B. I. Epureanu, “A novel phenomenological multi-physics model of Li-ion battery cells,” *Journal of Power Sources*, vol. 326, pp. 447–458, Sep. 2016.
- [260] M. T. Von Srbik, M. Marinescu, R. F. Martinez-Botas, and G. J. Offer, “A physically meaningful equivalent circuit network model of a lithium-ion battery accounting for local electrochemical and thermal behaviour, variable double layer capacitance and degradation,” *Journal of Power Sources*, vol. 325, pp. 171–184, Sep. 2016.
- [261] G. Fan, X. Li, and M. Canova, “A Reduced-Order Electrochemical Model of Li-Ion Batteries for Control and Estimation Applications,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 76–91, Jan. 2018.
- [262] M. Farag, M. Fleckenstein, and S. Habibi, “Continuous piecewise-linear, reduced-order electrochemical model for lithium-ion batteries in real-time applications,” *Journal of Power Sources*, vol. 342, pp. 351–362, Feb. 2017.
- [263] G. Piłatowicz, A. Marongiu, J. Drillkens, P. Sinhuber, and D. U. Sauer, “A critical overview of definitions and determination techniques of the internal resistance using lithium-ion, lead-acid, nickel metal-hydride batteries and electrochemical double-layer capacitors as examples,” *Journal of Power Sources*, vol. 296, pp. 365–376, Nov. 2015.
- [264] A. Seaman, T. S. Dao, and J. McPhee, *A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation*, Jun. 2014.
- [265] A. Fotouhi, D. J. Auger, K. Propp, and S. Longo, “Accuracy Versus Simplicity in Online Battery Model Identification,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 2, pp. 195–206, Feb. 2018.
- [266] A. Jokar, B. Rajabloo, M. Désilets, and M. Lacroix, *Review of simplified Pseudo-two-Dimensional models of lithium-ion batteries*, Sep. 2016.

-
- [267] A. Cordoba-Arenas, S. Onori, Y. Guezennec, and G. Rizzoni, "Capacity and power fade cycle-life model for plug-in hybrid electric vehicle lithium-ion battery cells containing blended spinel and layered-oxide positive electrodes," *Journal of Power Sources*, vol. 278, pp. 473–483, Mar. 2015. arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).
 - [268] A. Marongiu, N. Nlandi, Y. Rong, and D. U. Sauer, "On-board capacity estimation of lithium iron phosphate batteries by means of half-cell curves," *Journal of Power Sources*, vol. 324, pp. 158–169, Aug. 2016.
 - [269] L. Tao, C. Lu, and A. Noktehdan, "Similarity recognition of online data curves based on dynamic spatial time warping for the estimation of lithium-ion battery capacity," *Journal of Power Sources*, vol. 293, pp. 751–759, Oct. 2015.
 - [270] M. Ouyang, S. Gao, L. Lu, X. Feng, D. Ren, J. Li, Y. Zheng, and P. Shen, "Determination of the battery pack capacity considering the estimation error using a Capacity-Quantity diagram," *Applied Energy*, vol. 177, pp. 384–392, Sep. 2016.
 - [271] B. Balasingam, G. V. Avvari, B. Pattipati, K. R. Pattipati, and Y. Bar-Shalom, "A robust approach to battery fuel gauging, part II: Real time capacity estimation," *Journal of Power Sources*, vol. 269, pp. 949–961, Dec. 2014.
 - [272] T. Kim, Y. Wang, H. Fang, Z. Sahinoglu, T. Wada, S. Hara, and W. Qiao, "Model-based condition monitoring for lithium-ion batteries," *Journal of Power Sources*, vol. 295, pp. 16–27, Nov. 2015.
 - [273] T. Kim, Y. Wang, Z. Sahinoglu, T. Wada, S. Hara, and W. Qiao, "A Rayleigh Quotient-Based Recursive Total-Least-Squares Online Maximum Capacity Estimation for Lithium-Ion Batteries," *IEEE Transactions on Energy Conversion*, vol. 30, no. 3, pp. 842–851, Sep. 2015.
 - [274] S. Li, S. Pischinger, C. He, L. Liang, and M. Stapelbroek, "A comparative study of model-based capacity estimation algorithms in dual estimation frameworks for lithium-ion batteries under an accelerated aging test," *Applied Energy*, vol. 212, pp. 1522–1536, Feb. 2018.
 - [275] J. Li, C. Lyu, L. Wang, L. Zhang, and C. Li, "Remaining capacity estimation of Li-ion batteries based on temperature sample entropy and particle filter," *Journal of Power Sources*, vol. 268, pp. 895–903, Dec. 2014.
 - [276] X. Zhang, Y. Wang, C. Liu, and Z. Chen, "A novel approach of battery pack state of health estimation using artificial intelligence optimization algorithm," *Journal of Power Sources*, vol. 376, pp. 191–199, Feb. 2018.
 - [277] X. Li, J. Jiang, L. Y. Wang, D. Chen, Y. Zhang, and C. Zhang, "A capacity model based on charging process for state of health estimation of lithium ion batteries," *Applied Energy*, vol. 177, pp. 537–543, Sep. 2016.
 - [278] Y. Li, M. Abdel-Monem, R. Gopalakrishnan, M. Berecibar, E. Nanini-Maury, N. Omar, P. van den Bossche, and J. Van Mierlo, "A quick on-line state of health estimation method for Li-ion battery with incremental capacity curves processed by Gaussian filter," *Journal of Power Sources*, vol. 373, pp. 40–53, Jan. 2018.
 - [279] G. Liu, M. Ouyang, L. Lu, J. Li, and X. Han, "Online estimation of lithium-ion battery remaining discharge capacity through differential voltage analysis," *Journal of Power Sources*, vol. 274, pp. 971–989, Jan. 2015.
 - [280] T. Shibagaki, Y. Merla, and G. J. Offer, "Tracking degradation in lithium iron phosphate batteries using differential thermal voltammetry," *Journal of Power Sources*, vol. 374, pp. 188–195, Jan. 2018.
 - [281] A. A. Hussein, "Capacity Fade Estimation in Electric Vehicle Li-Ion Batteries Using Artificial Neural Networks," *IEEE Transactions on Industry Applications*, vol. 51, no. 3, pp. 2321–2330, May 2015.
 - [282] J. Li, L. Wang, C. Lyu, L. Zhang, and H. Wang, "Discharge capacity estimation for Li-ion batteries based on particle filter under multi-operating conditions," *Energy*, vol. 86, pp. 638–648, Jun. 2015.

- [283] C. Hu, G. Jain, C. Schmidt, C. Strief, and M. Sullivan, "Online estimation of lithium-ion battery capacity using sparse Bayesian learning," *Journal of Power Sources*, vol. 289, pp. 105–113, Sep. 2015.
- [284] L. Chen, W. Lin, J. Li, B. Tian, and H. Pan, "Prediction of lithium-ion battery capacity with metabolic grey model," *Energy*, vol. 106, pp. 662–672, Jul. 2016.
- [285] C. Hu, G. Jain, P. Zhang, C. Schmidt, P. Gomadam, and T. Gorka, "Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery," *Applied Energy*, vol. 129, pp. 49–55, Sep. 2014.
- [286] T. R. Ashwin, Y. M. Chung, and J. Wang, "Capacity fade modelling of lithium-ion battery under cyclic loading conditions," *Journal of Power Sources*, vol. 328, pp. 586–598, Oct. 2016.
- [287] R. H. Milocco, J. E. Thomas, and B. E. Castro, "Generic dynamic model of rechargeable batteries," *Journal of Power Sources*, vol. 246, pp. 609–620, Jan. 2014.
- [288] R. Xiong, J. Tian, H. Mu, and C. Wang, "A systematic model-based degradation behavior recognition and health monitoring method for lithium-ion batteries," *Applied Energy*, vol. 207, pp. 372–383, Dec. 2017.
- [289] C. Lu, L. Tao, and H. Fan, "Li-ion battery capacity estimation: A geometrical approach," *Journal of Power Sources*, vol. 261, pp. 141–147, Sep. 2014.
- [290] Z. Yang, D. Patil, and B. Fahimi, "Online Estimation of Capacity Fade and Power Fade of Lithium-Ion Batteries Based on Input–Output Response Technique," *IEEE Transactions on Transportation Electrification*, vol. 4, no. 1, pp. 147–156, Mar. 2018.
- [291] A. Farmann, W. Waag, A. Marongiu, and D. U. Sauer, "Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles," *Journal of Power Sources*, vol. 281, pp. 114–130, May 2015.
- [292] X. Han, M. Ouyang, L. Lu, and J. Li, "A comparative study of commercial lithium ion battery cycle life in electric vehicle: Capacity loss estimation," *Journal of Power Sources*, vol. 268, pp. 658–669, Dec. 2014.
- [293] L. Pei, C. Zhu, T. Wang, R. Lu, and C. C. Chan, "Online peak power prediction based on a parameter and state estimator for lithium-ion batteries in electric vehicles," *Energy*, vol. 66, pp. 766–778, Mar. 2014.
- [294] Y. Wang, R. Pan, C. Liu, Z. Chen, and Q. Ling, "Power capability evaluation for lithium iron phosphate batteries based on multi-parameter constraints estimation," *Journal of Power Sources*, vol. 374, pp. 12–23, Jan. 2018.
- [295] C. Burgos-Mellado, M. E. Orchard, M. Kazerani, R. Cárdenas, and D. Sáez, "Particle-filtering-based estimation of maximum available power state in Lithium-Ion batteries," *Applied Energy*, vol. 161, pp. 349–363, Jan. 2016.
- [296] D.-I. Stroe, M. Swierczynski, A.-I. Stroe, S. K. Kaer, and R. Teodorescu, "Lithium-ion battery power degradation modelling by electrochemical impedance spectroscopy," *IET Renewable Power Generation*, vol. 11, no. 9, pp. 1136–1141, Jul. 2017.
- [297] L. Wang, Y. Cheng, and J. Zou, "Battery available power prediction of hybrid electric vehicle based on improved Dynamic Matrix Control algorithms," *Journal of Power Sources*, vol. 261, pp. 337–347, Sep. 2014.
- [298] B. Fridholm, T. Wik, H. Kuusisto, and A. Klintberg, "Estimating power capability of aged lithium-ion batteries in presence of communication delays," *Journal of Power Sources*, vol. 383, pp. 24–33, Apr. 2018.
- [299] A. Farmann and D. U. Sauer, *A comprehensive review of on-board State-of-Available-Power prediction techniques for lithium-ion batteries in electric vehicles*, Oct. 2016.

-
- [300] M. Galeotti, L. Cinà, C. Giammanco, S. Cordiner, and A. Di Carlo, "Performance analysis and SOH (state of health) evaluation of lithium polymer batteries through electrochemical impedance spectroscopy," *Energy*, vol. 89, pp. 678–686, Sep. 2015.
 - [301] R. Mingant, J. Bernard, and V. Sauvant-Moynot, "Novel state-of-health diagnostic method for Li-ion battery in service," *Applied Energy*, vol. 183, pp. 390–398, Dec. 2016.
 - [302] C. Pastor-Fernández, K. Uddin, G. H. Chouchelamane, W. D. Widanage, and J. Marco, "A Comparison between Electrochemical Impedance Spectroscopy and Incremental Capacity-Differential Voltage as Li-ion Diagnostic Techniques to Identify and Quantify the Effects of Degradation Modes within Battery Management Systems," *Journal of Power Sources*, vol. 360, pp. 301–318, Aug. 2017.
 - [303] J. Kim and B. H. Cho, "An innovative approach for characteristic analysis and state-of-health diagnosis for a Li-ion cell based on the discrete wavelet transform," *Journal of Power Sources*, vol. 260, pp. 115–130, Aug. 2014.
 - [304] M. Berecibar, M. Garmendia, I. Gandiaga, J. Crego, and I. Villarreal, "State of health estimation algorithm of LiFePO₄ battery packs based on differential voltage curves for battery management system application," *Energy*, vol. 103, pp. 784–796, May 2016.
 - [305] M. Dubarry, M. Berecibar, A. Devie, D. Anseán, N. Omar, and I. Villarreal, "State of health battery estimator enabling degradation diagnosis: Model and algorithm description," *Journal of Power Sources*, vol. 360, pp. 59–69, Aug. 2017.
 - [306] Z. Guo, X. Qiu, G. Hou, B. Y. Liaw, and C. Zhang, "State of health estimation for lithium ion batteries based on charging curves," *Journal of Power Sources*, vol. 249, pp. 457–462, Mar. 2014.
 - [307] S. Torai, M. Nakagomi, S. Yoshitake, S. Yamaguchi, and N. Oyama, "State-of-health estimation of LiFePO₄/graphite batteries based on a model using differential capacity," *Journal of Power Sources*, vol. 306, pp. 62–69, Feb. 2016.
 - [308] L. Wang, C. Pan, L. Liu, Y. Cheng, and X. Zhao, "On-board state of health estimation of LiFePO₄ battery pack through differential voltage analysis," *Applied Energy*, vol. 168, pp. 465–472, Apr. 2016.
 - [309] L. Wang, X. Zhao, L. Liu, and C. Pan, "State of health estimation of battery modules via differential voltage analysis with local data symmetry method," *Electrochimica Acta*, vol. 256, pp. 81–89, Dec. 2017.
 - [310] Z. Wang, J. Ma, and L. Zhang, "State-of-Health Estimation for Lithium-Ion Batteries Based on the Multi-Island Genetic Algorithm and the Gaussian Process Regression," *IEEE Access*, vol. 5, pp. 21 286–21 295, 2017.
 - [311] C. Weng, X. Feng, J. Sun, and H. Peng, "State-of-health monitoring of lithium-ion battery modules and packs via incremental capacity peak tracking," *Applied Energy*, vol. 180, pp. 360–368, Oct. 2016.
 - [312] J. Yang, B. Xia, W. Huang, Y. Fu, and C. Mi, "Online state-of-health estimation for lithium-ion batteries using constant-voltage charging current analysis," *Applied Energy*, vol. 212, pp. 1589–1600, Feb. 2018.
 - [313] Q. Yang, J. Xu, B. Cao, D. Xu, X. Li, and B. Wang, "State-of-health Estimation of Lithium-ion Battery Based on Interval Capacity," in *Energy Procedia*, vol. 105, May 2017, pp. 2342–2347.
 - [314] J. Bi, T. Zhang, H. Yu, and Y. Kang, "State-of-health estimation of lithium-ion battery packs in electric vehicles based on genetic resampling particle filter," *Applied Energy*, vol. 182, pp. 558–568, Nov. 2016.
 - [315] J. Wu, Y. Wang, X. Zhang, and Z. Chen, "A novel state of health estimation method of Li-ion battery using group method of data handling," *Journal of Power Sources*, vol. 327, pp. 457–464, Sep. 2016.

- [316] D. Yang, Y. Wang, R. Pan, R. Chen, and Z. Chen, "A Neural Network Based State-of-Health Estimation of Lithium-ion Battery in Electric Vehicles," in *Energy Procedia*, vol. 105, May 2017, pp. 2059–2064.
- [317] G. won You, S. Park, and D. Oh, "Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach," *Applied Energy*, vol. 176, pp. 92–103, Aug. 2016.
- [318] G.-W. You, S. Park, and D. Oh, "Diagnosis of Electric Vehicle Batteries Using Recurrent Neural Networks," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4885–4893, Jun. 2017.
- [319] G. Bai, P. Wang, and C. Hu, "A self-cognizant dynamic system approach for prognostics and health management," *Journal of Power Sources*, vol. 278, pp. 163–174, Mar. 2015.
- [320] V. Klass, M. Behm, and G. Lindbergh, "A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation," *Journal of Power Sources*, vol. 270, pp. 262–272, Dec. 2014.
- [321] D. Yang, Y. Wang, R. Pan, R. Chen, and Z. Chen, *State-of-health estimation for the lithium-ion battery based on support vector regression*, Aug. 2017.
- [322] H. Dong, X. Jin, Y. Lou, and C. Wang, "Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter," *Journal of Power Sources*, vol. 271, pp. 114–123, Dec. 2014.
- [323] J. Wei, G. Dong, and Z. Chen, "Remaining Useful Life Prediction and State of Health Diagnosis for Lithium-Ion Batteries Using Particle Filter and Support Vector Regression," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5634–5643, 2018.
- [324] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Gaussian process regression for forecasting battery state of health," *Journal of Power Sources*, vol. 357, pp. 209–219, Jul. 2017. arXiv: 1703.05687.
- [325] A. Bartlett, J. Marcicki, K. Rhodes, and G. Rizzoni, "State of health estimation in composite electrode lithium-ion cells," *Journal of Power Sources*, vol. 284, pp. 642–649, Jun. 2015.
- [326] Y. Cai, L. Yang, Z. Deng, X. Zhao, and H. Deng, "Online identification of lithium-ion battery state-of-health based on fast wavelet transform and cross D-Markov machine," *Energy*, vol. 147, pp. 621–635, Mar. 2018.
- [327] Y. Li, P. Chattopadhyay, A. Ray, and C. D. Rahn, "Identification of the battery state-of-health parameter from input-output pairs of time series data," *Journal of Power Sources*, vol. 285, pp. 235–246, Jul. 2015.
- [328] S. Peikun and W. Zhenpo, "Research of the relationship between Li-ion battery charge performance and SOH based on MIGA-GPR method," in *Energy Procedia*, vol. 88, Jun. 2016, pp. 608–613.
- [329] B. Sun, J. Jiang, F. Zheng, W. Zhao, B. Y. Liaw, H. Ruan, Z. Han, and W. Zhang, "Practical state of health estimation of power batteries based on Delphi method and grey relational grade analysis," *Journal of Power Sources*, vol. 282, pp. 146–157, May 2015.
- [330] Y. Cui, P. Zuo, C. Du, Y. Gao, J. Yang, X. Cheng, Y. Ma, and G. Yin, "State of health diagnosis model for lithium ion batteries based on real-time impedance and open circuit voltage parameters identification method," *Energy*, vol. 144, pp. 647–656, Feb. 2018.
- [331] F. Camci, C. Ozkurt, O. Toker, and V. Atamuradov, "Sampling based State of Health estimation methodology for Li-ion batteries," *Journal of Power Sources*, vol. 278, pp. 668–674, Mar. 2015.
- [332] C. Ozkurt, F. Camci, V. Atamuradov, and C. Odorry, "Integration of sampling based battery state of health estimation method in electric vehicles," *Applied Energy*, vol. 175, pp. 356–367, Aug. 2016.
- [333] Y. Merla, B. Wu, V. Yufit, N. P. Brandon, R. F. Martinez-Botas, and G. J. Offer, "Novel application of differential thermal voltammetry as an in-depth state-of-health diagnosis method for lithium-ion batteries," *Journal of Power Sources*, vol. 307, pp. 308–319, Mar. 2016.

-
- [334] C. Lin, A. Tang, and W. Wang, "A Review of SOH Estimation Methods in Lithium-ion Batteries for Electric Vehicle Applications," in *Energy Procedia*, vol. 75, Aug. 2015, pp. 1920–1925.
 - [335] S. M. Rezvanizani, Z. Liu, Y. Chen, and J. Lee, *Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (EV) safety and mobility*, Jun. 2014.
 - [336] Y. Chang, H. Fang, and Y. Zhang, "A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery," *Applied Energy*, vol. 206, pp. 1564–1578, Nov. 2017.
 - [337] X.-S. Si, "An Adaptive Prognostic Approach via Nonlinear Degradation Modeling: Application to Battery Data," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5082–5096, Aug. 2015.
 - [338] Q. Zhai and Z.-S. Ye, "RUL Prediction of Deteriorating Products Using an Adaptive Wiener Process Model," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2911–2921, Dec. 2017.
 - [339] D. Z. Li, W. Wang, and F. Ismail, "A Mutated Particle Filter Technique for System State Estimation and Battery Life Prediction," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 8, pp. 2034–2043, Aug. 2014.
 - [340] Z. Liu, G. Sun, S. Bu, J. Han, X. Tang, and M. Pecht, "Particle Learning Framework for Estimating the Remaining Useful Life of Lithium-Ion Batteries," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 2, pp. 280–293, Feb. 2017.
 - [341] C. Sbarufatti, M. Corbetta, M. Giglio, and F. Cadini, "Adaptive prognosis of lithium-ion batteries based on the combination of particle filters and radial basis function neural networks," *Journal of Power Sources*, vol. 344, pp. 128–140, Mar. 2017.
 - [342] Datong Liu, Wei Xie, Haitao Liao, and Yu Peng, "An Integrated Probabilistic Approach to Lithium-Ion Battery Remaining Useful Life Estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 660–670, Mar. 2015.
 - [343] J. Wu, C. Zhang, and Z. Chen, "An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks," *Applied Energy*, vol. 173, pp. 134–140, Jul. 2016.
 - [344] Datong Liu, Jianbao Zhou, Haitao Liao, Yu Peng, and Xiyuan Peng, "A Health Indicator Extraction and Optimization Framework for Lithium-Ion Battery Degradation Modeling and Prognostics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 6, pp. 915–928, Jun. 2015.
 - [345] M. A. Patil, P. Tagade, K. S. Hariharan, S. M. Kolake, T. Song, T. Yeo, and S. Doo, "A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation," *Applied Energy*, vol. 159, pp. 285–297, Dec. 2015.
 - [346] Y. Zhou, M. Huang, Y. Chen, and Y. Tao, "A novel health indicator for on-line lithium-ion batteries remaining useful life prediction," *Journal of Power Sources*, vol. 321, pp. 1–10, Jul. 2016.
 - [347] S. S. Ng, Y. Xing, and K. L. Tsui, "A naive bayes model for robust remaining useful life prediction of lithium-ion battery," *Applied Energy*, vol. 118, no. 4, pp. 114–123, Apr. 2014.
 - [348] Z.-X. Zhang, X.-S. Si, C.-H. Hu, and M. G. Pecht, "A Prognostic Model for Stochastic Degrading Systems With State Recovery: Application to Li-Ion Batteries," *IEEE Transactions on Reliability*, vol. 66, no. 4, pp. 1293–1308, Dec. 2017.
 - [349] E. Walker, S. Rayman, and R. E. White, "Comparison of a particle filter and other state estimation methods for prognostics of lithium-ion batteries," *Journal of Power Sources*, vol. 287, pp. 1–12, Aug. 2015.
 - [350] T. A. S. Corporation and A. Gelb, *Applied Optimal Estimation*. The MIT Press, 1974.

- [351] B. Lei, G. Xu, M. Feng, Y. Zou, F. van der Heijden, D. d. Ridder, and D. M. J. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*, 2nd. Wiley Publishing, 2017.
- [352] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [353] B. Lei, G. Xu, M. Feng, Y. Zou, F. van der Heijden, D. de Ridder, and D. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*. Wiley, 2017.
- [354] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*, ser. Chapman & Hall/CRC Applied Mathematics & Nonlinear Science. CRC Press, 2004.
- [355] R. G. Brown, "Introduction to Random Signal Analysis and Kalman Filtering," 1983.
- [356] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems (2Nd Ed.)* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [357] S. S. Haykin and B. V. Veen, *Signals and Systems*, 1st. New York, NY, USA: John Wiley & Sons, Inc., 1998.
- [358] D. P. Mandic. (2018). "Advanced Signal Processing - Introduction to Estimation Theory," [Online]. Available: https://www.commsp.ee.ic.ac.uk/~mandic/ASP_Slides/ASP_Lecture_3_Estimation_Intro_2018.pdf (visited on 07/21/2018).
- [359] A. Karimi. (2013). "Estimation Theory," [Online]. Available: <https://la.epfl.ch/files/content/sites/la/files/users/139973/public/Estimation%20theory/EstimationTheory.pdf> (visited on 07/21/2018).
- [360] S. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*, ser. Fundamentals of Statistical Signal Processing. PTR Prentice-Hall, 1993.
- [361] T. Schön, "On computational methods for nonlinear estimation," Ph.D. dissertation, Linköping University, The Institute of Technology, 2003, p. 62.
- [362] J. Raol, G. Gopalratnam, and B. Twala, *Nonlinear Filtering: Concepts and Engineering Applications*. CRC Press, 2017.
- [363] A. Haug, *Bayesian Estimation and Tracking: A Practical Guide*. Wiley, 2012.
- [364] D. o. P. Saroja Polavarapu. (2004). "Introduction to Estimation Theory," [Online]. Available: http://www.atmosp.physics.utoronto.ca/PHY2509/on9901_ch4.pdf (visited on 06/26/2019).
- [365] D. Mandic. (2018). "Advanced Signal Processing - Adaptive Estimation and Filtering," [Online]. Available: https://www.commsp.ee.ic.ac.uk/~mandic/ASP_Slides/ASP_Lecture_7_Adaptive_Filters_2018.pdf.
- [366] A. Karimi. (2013). "Estimation Theory," [Online]. Available: https://la.epfl.ch/page-53049-en-html/estimation_theory/.
- [367] T. B. Schön, "Estimation of Nonlinear Dynamic Systems : Theory and Applications," Ph.D. dissertation, Linköping University, The Institute of Technology, 2006.
- [368] *Fundamentals of Statistical Signal Processing: WITH Estimation Theory, Volume 1 AND Detection Theory*, v. 2. Pearson Education (US), 2001.
- [369] G. Lebanon. (2006). "The Rao-Blackwell Theorem and the UMVUE," [Online]. Available: <http://theanalysisofdata.com/notes/RaoBlackwell.pdf>.
- [370] E. Zivot. (2009). "Maximum Likelihood Estimation," [Online]. Available: <http://faculty.washington.edu/ezivot/econ583/mleLectures.pdf> (visited on 07/21/2018).
- [371] M. G. Marasinghe. (2018). "Statistics 580, Maximum Likelihood Estimation," [Online]. Available: <http://www.public.iastate.edu/~mervyn/stat580/Notes/s09mle.pdf> (visited on 07/21/2018).

-
- [372] A. Gelb, A. S. C. T. Staff, and A. S. Corporation, *Applied Optimal Estimation*, ser. Applied Optimal Estimation. MIT Press, 1974.
 - [373] H. Van Trees, *Detection, Estimation, and Modulation Theory: Detection, estimation, and linear modulation theory*, ser. Detection, Estimation, and Modulation Theory. Wiley, 1968.
 - [374] —, *Detection, Estimation, and Modulation Theory*, ser. Detection, Estimation, and Modulation Theory v. 1. John Wiley & Sons, 2001.
 - [375] D. Manolakis, V. Ingle, and S. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering, and Array Processing*, ser. Artech House signal processing library. Artech House, 2005.
 - [376] B. Anderson and J. Moore, *Optimal Filtering*, ser. Dover Books on Electrical Engineering. Dover Publications, 2012.
 - [377] F. van der Heijden, R. Duin, D. de Ridder, and D. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MatLab*, Undefined. United States: Wiley, 2004.
 - [378] J. P. N. Bishwal, *Parametric Stochastic Differential Equations*, ser. Mathematics and Statistics. Springer-Verlag Berlin Heidelberg, 2008.
 - [379] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, ser. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2011.
 - [380] M.-A. Beyer, “Methodische Evaluierung von wahrscheinlichkeitstheoretischen Inferenzverfahren fuer dynamische Systeme,” Ph.D. dissertation, Fakultät fuer Maschinenbau der Ruhr-Universität Bochum, The address of the publisher, 2008.
 - [381] B. Koehler, *Konzepte der statistischen Signalverarbeitung*. Springer Berlin Heidelberg, 2005.
 - [382] T. Sauer, “Computational Solution of Stochastic Differential Equations,” *WIREs Comput. Stat.*, vol. 5, no. 5, pp. 362–371, Sep. 2013.
 - [383] J. Bishwal, *Parameter Estimation in Stochastic Differential Equations*, ser. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2007.
 - [384] K. Matsumori, Y. Koike, and K. Matsumoto, “A Biased Bayesian Inference for Decision-Making and Cognitive Control,” *Frontiers in Neuroscience*, vol. 12, p. 734, 2018.
 - [385] S. M. Lynch, *Introduction to Applied Bayesian Statistics and Estimation for Social Scientists*. New York, NY: Springer Science & Business Media, LLC., Apr. 2007.
 - [386] J. V. Candy, *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods*. New York, NY, USA: Wiley-Interscience, 2009.
 - [387] Z. Chen, “Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond,” McMaster University, Tech. Rep., 2003.
 - [388] J. J. K. Ó Ruanaidh and W. (J. Fitzgerald, *Numerical Bayesian methods applied to signal processing*, English. New York : Springer, 1996.
 - [389] P. Zarchan, H. Musoff, F. Lu, A. I. of Aeronautics, and Astronautics, *Fundamentals of Kalman Filtering: A Practical Approach*, ser. Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, 2009.
 - [390] R. Van Der Merwe, “Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-space Models,” Ph.D. dissertation, 2004.
 - [391] F. Gustafsson, *Adaptive Filtering and Change Detection*, 2000.
 - [392] N. Bergman, “Recursive Bayesian estimation: Navigation and tracking applications,” Division of Automatic Control and Communication Systems Department of Electrical Engineering Linköping University, Sweden, Tech. Rep., 1999.

- [393] A. Doucet, S. Godsill, and C. Andrieu, “On Sequential Monte Carlo Sampling Methods for Bayesian Filtering,” *STATISTICS AND COMPUTING*, vol. 10, no. 3, pp. 197–208, 2000.
- [394] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, “A novel Gaussian Sum Filter Method for accurate solution to the nonlinear filtering problem,” pp. 1–8, Jun. 2008.
- [395] N. Wiebe, C. E. Granade, A. Kapoor, and K. M. Svore, “Bayesian inference via rejection filtering,” *CoRR*, vol. abs/1511.06458, 2015. arXiv: 1511.06458.
- [396] S. S. Haykin, *Kalman Filtering and Neural Networks*. New York, NY, USA: John Wiley & Sons, Inc., 2001.
- [397] S. Haykin, *Kalman Filtering and Neural Networks*, ser. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley, 2004.
- [398] R. H. Reichle, J. P. Walker, R. D. Koster, and P. R. Houser, “Extended versus Ensemble Kalman Filtering for Land Data Assimilation,” *Journal of Hydrometeorology*, vol. 3, no. 6, pp. 728–740, 2002. eprint: [https://doi.org/10.1175/1525-7541\(2002\)003<0728:EVEKFF>2.0.CO;2](https://doi.org/10.1175/1525-7541(2002)003<0728:EVEKFF>2.0.CO;2).
- [399] T. D. Laet, “Estimation,” Ph.D. dissertation, 2013.
- [400] K. Ito and K. Xiong, “Gaussian Filters for Nonlinear Filtering Problems,” *IEEE Transactions on Automatic Control*, vol. 45, pp. 910–927, 1999.
- [401] S. J. Julier, “The scaled unscented transformation,” in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 6, May 2002, 4555–4559 vol.6.
- [402] K. Ito and K. Xiong, “Gaussian filters for nonlinear filtering problems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.
- [403] R. V. D. Merwe and E. Wan, “Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models,” in *In Proceedings of the Workshop on Advances in Machine Learning*, 2003.
- [404] —, *Sigma-Point Kalman Filters For Integrated Navigation*, 2004.
- [405] I. I. Arrese, *Implementation and Testing of Sigma-Point Kalman Filters in Simulink for Nonlinear Estimation*. TU Darmstadt, 2016.
- [406] V. Šmídl and M. Gašperin, “Rao-Blackwellized point mass filter for reliable state estimation,” in *Proceedings of the 16th International Conference on Information Fusion*, Jul. 2013, pp. 312–318.
- [407] R. S. Bucy and K. Senne, “Digital Synthesis of Nonlinear Filters,” Jul. 1970.
- [408] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, “Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation,” *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2151–2156, Sep. 2011.
- [409] W. I. Tam, K. Plataniotis, and D. Hatzinakos, “An adaptive Gaussian sum algorithm for radar tracking,” *Signal Processing*, vol. 77, no. 1, pp. 85–104, 1999.
- [410] Z. Chen, “Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond,” McMaster University, Tech. Rep., 2003.
- [411] J. D. Hol, “Resampling in particle filters,” Linköpings University, Sweden, Tech. Rep., 2004.
- [412] M. I. Jordan. (2010). “Stat260: Bayesian Modeling and Inference,” [Online]. Available: <https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture17.pdf> (visited on 06/12/2018).
- [413] Z. CHEN. (2019). “Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond,” [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.7415&rep=rep1&type=pdf>.
- [414] J. D. Hol, T. B. Schon, and F. Gustafsson, “On Resampling Algorithms for Particle Filters,” in *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Sep. 2006, pp. 79–82.

-
- [415] J. B. Lee, M. J. Kim, S. Yoon, and E. Y. Chung, "Application-Support Particle Filter for Dynamic Voltage Scaling of Multimedia Applications," *IEEE Transactions on Computers*, vol. 61, no. 9, pp. 1256–1269, Sep. 2012.
 - [416] T. Li, S. Sun, T. P. Sattar, and J. M. Corchado, "Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches," *Expert Syst. Appl.*, vol. 41, pp. 3944–3954, 2014.
 - [417] N. Whiteley and A. M. Johansen, "Auxiliary particle filtering: recent developments," in *Bayesian Time Series Models*, D. Barber, A. T. Cemgil, and S. Chiappa, Eds. Cambridge University Press, 2011, pp. 52–81.
 - [418] —, (2012). "Recent Developments in Auxiliary Particle Filtering," [Online]. Available: https://people.maths.bris.ac.uk/~manpw/apf_chapter.pdf (visited on 06/12/2018).
 - [419] E. Orhan. (2012). "Particle Filtering," [Online]. Available: <http://www.cns.nyu.edu/~eorhan/notes/particle-filtering.pdf> (visited on 06/12/2018).
 - [420] D. CRISAN and J. MIGUEZ, "Particle-kernel estimation of the filter density in state-space models," 2013.
 - [421] P. Diaconis. (2012). "Particle Filtering," [Online]. Available: <http://statweb.stanford.edu/~cgates/PERSI/papers/MCMCRev.pdf> (visited on 06/12/2018).
 - [422] C. Graham, *Markov Chains: Analytic and Monte Carlo Computations*, ser. Wiley Series in Probability and Statistics. Wiley, 2014.
 - [423] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.
 - [424] —, "Gaussian Sum Particle Filtering," *Trans. Sig. Proc.*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.
 - [425] F. Mustiere, M. Bolic, and M. Bouchard, "A Modified Rao-Blackwellised Particle Filter," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 3, May 2006, pp. III–III.
 - [426] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.
 - [427] G. L. Plett, "Dual and Joint EKF for Simultaneous SOC and SOH Estimation, in," in *CD-ROM Proceedings of the 21st Electric Vehicle Symposium (EVS21)*, 2005.
 - [428] R. V. der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 6, 2001, 3461–3464 vol.6.
 - [429] J. Albert, *Bayesian Computation with R*, ser. Use R! Springer New York, 2009.
 - [430] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "A comparative study of observer design techniques for state of charge estimation in electric vehicles," in *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Jul. 2012, pp. 102–107.
 - [431] B. Fridholm, M. Nilsson, and T. Wik, "Robustness Comparison of Battery State of Charge Observers for Automotive Applications," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 2138–2146, 2014.
 - [432] J. Li, J. K. Barillas, C. Guenther, and M. A. Danzer, "A comparative study of state of charge estimation algorithms for LiFePO4 batteries used in electric vehicles," *Journal of Power Sources*, vol. 230, pp. 244–250, 2013.
 - [433] S. Konatowski, P. Kaniewski, and J. Matuszewski, "Comparison of Estimation Accuracy of EKF, UKF and PF Filters," *Annual of Navigation*, vol. 23, pp. 69–87, 2016.
 - [434] C. Huang, Z. Wang, Z. Zhao, L. Wang, C. Lai, and D. Wang, "Robustness Evaluation of Extended and Unscented Kalman Filter for Battery State of Charge Estimation," *IEEE Access*, May 2018.

- [435] R.-E. Tudoroiu, M. Zaheeruddin, S.-M. Radu, and N. Tudoroiu, "Estimation Techniques for State of Charge in Battery Management Systems on Board of Hybrid Electric Vehicles Implemented in a Real-Time MATLAB/SIMULINK Environment," in Jan. 2019.
- [436] A. Papazoglou, S. Longo, D. Auger, and F. Assadian, "Nonlinear filtering techniques comparison for battery state estimation," *Journal of Sustainable Development of Energy, Water and Environment Systems*, vol. 2, pp. 259–269, Sep. 2014.
- [437] A. Jossen, "Fundamentals of battery dynamics," *Journal of Power Sources*, vol. 154, no. 2, pp. 530–538, 2006.
- [438] S. Buller, "Impedance based simulation models for energy storage devices in advanced automotive power systems," Ph.D. dissertation, Aachen, 2003, 138 S. : Ill., graph. Darst.
- [439] W. Widanage, A. Barai, G. Chouchelamane, K. Uddin, A. McGordon, J. Marco, and P. Jennings, "Design and use of multisine signals for Li-ion battery equivalent circuit modelling. Part 1: Signal design," *Journal of Power Sources*, vol. 324, pp. 70–78, 2016.
- [440] —, "Design and use of multisine signals for Li-ion battery equivalent circuit modelling. Part 2: Model estimation," *Journal of Power Sources*, vol. 324, pp. 61–69, 2016.
- [441] J.-C. K. Yi-Hsien Chiang Wu-Yang Sean, "Online estimation of internal resistance and open-circuit voltage of lithium-ion batteries in electric vehicles," *Journal of Power Sources*, vol. 196, pp. 3921–3932, 2011.
- [442] B. Y. Liaw, G. Nagasubramanian, R. G. Jungst, and D. H. Doughty, "Modeling of lithium ion cells—A simple equivalent-circuit model approach," *Solid State Ionics*, vol. 175, no. 1, pp. 835–839, 2004.
- [443] B. Y. L. Matthieu Dubarry, "Development of a universal modeling tool for rechargeable lithium batteries," *Journal of Power Sources*, vol. 174, pp. 856–860, 2007.
- [444] H. P. Xiaosong Hu Shengbo Li, "A comparative study of equivalent circuit models for Li-ion batteries," *Journal of Power Sources*, vol. 198, pp. 359–367, 2012.
- [445] J. Sieg, "Entwicklung und Erprobung eines durch Routeninformationen gestuetztes Energiemanagementsystems fuer Elektrofahrzeuge," Ph.D. dissertation, ISEA, RWTH Aachen, Jul. 2014.
- [446] A. Burkov, *The Hundred-Page Machine Learning Book*, 1st ed. Kindle Direct Publishing, 2019.
- [447] A. Karpathy. (2018). "CS231n: Convolutional Neural Networks for Visual Recognition," [Online]. Available: <http://cs231n.github.io/neural-networks-1/> (visited on 07/01/2018).
- [448] J.-S. R. Jang, "Neuro-Fuzzy Modeling: Architectures, Analyses and Applications," Ph.D. dissertation, University of California, Berkeley, 1992.
- [449] F. Chollet, *Deep Learning with Python*, 1st. Greenwich, CT, USA: Manning Publications Co., 2017.
- [450] Z. Zhang, *Student assistant, Supervisor: Christian Fleischer*. Chalmers University of Technology: Christian Fleischer, 2018.
- [451] S. Albawi, T. Abed Mohammed, and S. ALZAWI, "Understanding of a Convolutional Neural Network," Aug. 2017.
- [452] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1, pp. 305–307, Jun. 2018.
- [453] D. Stutz. (2014). "Understanding Convolutional Neural Networks," [Online]. Available: <https://davidstutz.de/wordpress/wp-content/uploads/2014/07/seminar.pdf> (visited on 07/01/2018).
- [454] —, (2014). "Understanding Convolutional Neural Networks."
- [455] A. Marongiu, "Performance and Aging Diagnostic on Lithium Iron Phosphate Batteries for Electric Vehicles and Vehicle-to-Grid Strategies," Ph.D. dissertation, Aachen, 2017.

-
- [456] M. Lewerenz, A. Marongiu, A. Warnecke, and D. Sauer, "Differential voltage analysis as a tool for analyzing inhomogeneous aging: A case study for LiFePO₄—Graphite cylindrical cells," vol. 368, pp. 57–67, Sep. 2017.
 - [457] NASA. (2010). "Li-ion battery aging datasets,," [Online]. Available: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#battery> (visited on 04/01/2018).
 - [458] R. van der Merwe. (2004). "ReBEL: Recursive Bayesian Estimation Library toolkit," [Online]. Available: <http://www.csee.ogi.edu/~rudmerwe/#software> (visited on 06/12/2018).
 - [459] T. De Schutter, *Better Software. Faster!: Best Practices in Virtual Prototyping*. Synopsys Press, 2014.
 - [460] J. V. Barreras, C. Fleischer, A. E. Christensen, M. Swierczynski, E. Schaltz, S. J. Andreasen, and D. U. Sauer, "An Advanced HIL Simulation Battery Model for Battery Management System Testing," *IEEE Transactions on Industry Applications*, vol. 52, no. 6, pp. 5086–5099, Nov. 2016.
 - [461] C. Fleischer, D. U. Sauer, J. V. Barreras, E. Schaltz, and A. E. Christensen, "Development of software and strategies for Battery Management System testing on HIL simulator," in *2016 Eleventh International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Apr. 2016, pp. 1–12.
 - [462] J. V. Barreras, M. Swierczynski, E. Schaltz, S. J. Andreasen, C. Fleischer, D. U. Sauer, and A. E. Christensen, "Functional analysis of Battery Management Systems using multi-cell HIL simulator," in *2015 Tenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Mar. 2015, pp. 1–10.
 - [463] S. Jiang, J. Jiang, and C. Zhang, *Fundamentals and Applications of Lithium-ion Batteries in Electric Drive Vehicles*, ser. Wiley Online Library. Wiley, 2015.
 - [464] D. Smith and K. Simpson, *Functional Safety*. Taylor & Francis, 2004.
 - [465] A. Benso and P. Prinetto, *Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation*, 1st. Springer Publishing Company, Incorporated, 2010.
 - [466] —, *Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation*, 1st. Springer Publishing Company, Incorporated, 2010.
 - [467] S. Li and S. Duo, "Safety Analysis of Software Requirements: Model and Process," *Procedia Engineering*, vol. 80, pp. 153–164, 2014.
 - [468] P. P. Koopman. (2014). "A Case Study of Toyota Unintended Acceleration and Software Safety," [Online]. Available: https://users.ece.cmu.edu/~koopman/toyota/koopman-09-18-2014_toyota_slides.pdf.
 - [469] P. Raistrick. (2009). "Moving model-based development into safety-critical embedded applications," [Online]. Available: <https://www.embedded.com/print/4008789>.
 - [470] M. Unemyr. (2015), [Online]. Available: <http://blog.atollic.com/how-embedded-developers-can-learn-from-the-aerospace-industry>.
 - [471] C. Diggins. (2009). "Three Reasons for Moving to Multicore," [Online]. Available: <http://www.drdobbs.com/parallel/three-reasons-for-moving-to-multicore/216200386>.
 - [472] C. Fleischer, *Leading the Electric Vehicle Charge*, Issue 2. ANSYS Advantage - Volume XI, 2017.
 - [473] Infineon. (2018), [Online]. Available: https://www.infineon.com/dgdl/Infineon-TriCore_Family_BR-2018-BC-v03_00-EN.pdf?fileId=5546d4625d5945ed015dc81f47b436c7.
 - [474] G. S. (S. B. U. F. R. K. (E.-S. G. Bruno PAGANO Cédric PASTEUR. (2018). "A Model Based Safety Critical Flow for the AURIX™ Multi-core Platform," [Online]. Available: <https://www.erts2018.org/programme-thursday.html>.

ABISEA Band 1

Eßer, Albert

Berührungslose, kombinierte
Energie- und Informations-
übertragung für bewegliche
Systeme

1. Aufl. 1992, 129 S.

ISBN 3-86073-046-0

ABISEA Band 2

Vogel, Ulrich

Entwurf und Beurteilung von
Verfahren zur
Hochausnutzung des Rad-
Schiene-Kraftschlusses
durch Triebfahrzeuge

1. Aufl. 1992, 131 S.

ISBN 3-86073-060-6

ABISEA Band 3

Reckhorn, Thomas

Stromeinprägendes
Antriebssystem mit
fremderregter Synchron-
maschine

1. Aufl. 1992, 128 S.

ISBN 3-86073-061-4

ABISEA Band 4

Ackva, Ansgar

Spannungseinprägendes
Antriebssystem mit
Synchronmaschine und
direkter Stromregelung

1. Aufl. 1992, 137 S.

ISBN 3-86073-062-2

ABISEA Band 5

Mertens, Axel

Analyse des
Oberschwingungsverhaltens
von taktsynchronen Delta -
Modulationsverfahren zur
Steuerung von Pulsstrom-
richtern bei hoher Taktzahl

1. Aufl. 1992, 178 S.

ISBN 3-86073-069-X

ABISEA Band 6

Geuer, Wolfgang

Untersuchungen über das
Alterungsverhalten von Blei-
Akkumulatoren

1. Aufl. 1993, 97 S.

ISBN 3-86073-097-5

ABISEA Band 7

Langheim, Jochen

Einzelradantrieb für
Elektrostraßenfahrzeuge

1. Aufl. 1993, 213 S.

ISBN 3-86073-123-8

(vergriffen)

ABISEA Band 8

Fetz, Joachim

Fehlertolerante Regelung
eines Asynchron-
Doppelantriebes für ein
Elektrospeicherfahrzeug

1. Aufl. 1993, 136 S.

ISBN 3-86073-124-6

(vergriffen)

ABISEA Band 9

Schülting, Ludger

Optimierte Auslegung
induktiver Bauelemente für
den Mittelfrequenzbereich

1. Aufl. 1993, 126 S.

ISBN 3-86073-174-2

(vergriffen)

ABISEA Band 10

Skudelny, H.-Ch.

Stromrichtertechnik

4. Aufl. 1997, 259 S.

ISBN 3-86073-189-0

ABISEA Band 11

Skudelny, H.-Ch.

Elektrische Antriebe

3. Aufl. 1997, 124 S.

ISBN 3-86073-231-5

ABISEA Band 12

Schöpe, Friedhelm

Batterie-Management für
Nickel-Cadmium

Akkumulatoren

1. Aufl. 1994, 148 S.

ISBN 3-86073-232-3

(vergriffen)

ABISEA Band 13

v. d. Weem, Jürgen

Schmalbandige aktive Filter
für Schienentriebfahrzeuge
am Gleichspannungs-
fahrdraht

1. Aufl. 1995, 126 S.

ISBN 3-86073-233-1

ABISEA Band 14

Backhaus, Klaus

Spannungseinprägendes
Direktantriebssystem mit
schnelllaufender
geschalteter

Reluktanzmaschine

1. Aufl. 1995, 146 S.

ISBN 3-86073-234-X

(vergriffen)

ABISEA Band 15

Reinold, Harry

Optimierung dreiphasiger
Pulsdauermodulations-
verfahren

1. Aufl. 1996, 107 S.

ISBN 3-86073-235-8

ABISEA Band 16

Köpken, Hans-Georg

Regelverfahren für
Parallelschwingkreis-
umrichter

1. Aufl. 1996, 125 S.

ISBN 3-86073-236-6

ABISEA Band 17

Mauracher, Peter

Modellbildung und
Verbundoptimierung bei
Elektrostraßenfahrzeugen

1. Aufl. 1996, 192 S.

ISBN 3-86073-237-4

ABISEA Band 18

Protiwa, Franz-Ferdinand

Vergleich dreiphasiger
Resonanz-Wechselrichter in
Simulation und Messung

1. Aufl. 1997, 178 S.

ISBN 3-86073-238-2

ABISEA Band 19

Brockmeyer, Ansgar

Dimensionierungswerkzeug
für magnetische Bau-
elemente in Stromrichter-
anwendungen

1. Aufl. 1997, 175 S.

ISBN 3-86073-239-0

ABISEA Band 20

Apeldoorn, Oscar

Simulationsgestützte Bewertung von Steuerverfahren für netzgeführte Stromrichter mit verringerter Netzrückwirkung

1. Aufl. 1997, 134 S.
ISBN 3-86073-680-9

ABISEA Band 21

Lohner, Andreas

Batteriemanagement für verschlossene Blei-Batterien am Beispiel von Unterbrechungsfreien Stromversorgungen

1. Aufl. 1998, 126 S.
ISBN 3-86073-681-7

ABISEA Band 22

Reinert, Jürgen

Optimierung der Betriebseigenschaften von Antrieben mit geschalteter Reluktanzmaschine

1. Aufl. 1998, 153 S.
ISBN 3-86073-682-5

ABISEA Band 23

Nagel, Andreas

Leitungsgebundene Störungen in der Leistungselektronik: Entstehung, Ausbreitung und Filterung

1. Aufl. 1999, 140 S.
ISBN 3-86073-683-3

ABISEA Band 24

Menne, Marcus

Drehschwingungen im Antriebsstrang von Elektrostraßenfahrzeugen - Analyse und aktive Dämpfung

1. Aufl. 2001, 169 S.
ISBN 3-86073-684-1

ABISEA Band 25

von Bloh, Jochen

Multilevel-Umrichter zum Einsatz in Mittelspannungsgleichspannungs-Übertragungen

1. Aufl. 2001, 137 S.
ISBN 3-86073-685-X

ABISEA Band 26

Karden, Eckhard

Using low-frequency impedance spectroscopy for characterization, monitoring, and modeling of industrial batteries

1. Aufl. 2002, 137 S.
ISBN 3-8265-9766-4

ABISEA Band 27

Karipidis, Claus-Ulrich

A Versatile DSP/ FPGA Structure optimized for Rapid Prototyping and Digital Real-Time Simulation of Power Electronic and Electrical Drive Systems

1. Aufl. 2001, 164 S.
ISBN 3-8265-9738-9

ABISEA Band 28

Kahlen, Klemens

Regelungsstrategien für permanentmagnetische Direktantriebe mit mehreren Freiheitsgraden

1. Aufl. 2002, 154 S.
ISBN 3-8322-1222-1

ABISEA Band 29

Inderka, Robert B.

Direkte Drehmomentregelung Geschalteter Reluktanzantriebe

1. Aufl. 2003, 182 S.
ISBN 3-8322-1175-6

ABISEA Band 30

Schröder, Stefan

Circuit-Simulation Models of High-Power Devices Based on Semiconductor Physics

1. Aufl. 2003, 123 S.
ISBN 3-8322-1250-7

ABISEA Band 31

Buller, Stephan

Impedance-Based Simulation Models for Energy Storage Devices in Advanced Automotive Power Systems

1. Aufl. 2003, 138 S.
ISBN 3-8322-1225-6

ABISEA Band 32

Schönknecht, Andreas

Topologien und Regelungsstrategien für das induktive Erwärmen mit hohen Frequenz-Leistungsprodukten

1. Aufl. 2004, 157 S.
ISBN 3-8322-2408-4

ABISEA Band 33

Tolle, Tobias

Konvertertopologien für ein aufwandsarmes, zweistufiges Schaltnetzteil zum Laden von Batterien aus dem Netz

1. Aufl. 2004, 148 S.
ISBN 3-8322-2676-1

ABISEA Band 34

Götting, Gunther

Dynamische Antriebsregelung von Elektrostraßenfahrzeugen unter Berücksichtigung eines schwingungsfähigen Antriebsstrangs

1. Aufl. 2004, 157 S.
ISBN 3-8322-2804-7

ABISEA Band 35

Dieckerhoff, Sibylle

Transformatorlose Stromrichterschaltungen für Bahnfahrzeuge am 16 2/3Hz Netz

1. Aufl. 2004, 147 S.
ISBN 3-8322-3094-7

ABISEA Band 36

Hu, Jing

Bewertung von DC-DC-Topologien und Optimierung eines DC-DC-Leistungsmoduls für das 42-V-Kfz-Bordnetz

1. Aufl. 2004, 148 S.
ISBN 3-8322-3201-X

ABISEA Band 37

Detjen, Dirk-Oliver

Characterization and Modeling of Si-Si Bonded Hydrophobic Interfaces for Novel High-Power BIMOS Devices

1. Aufl. 2004, 135 S.
ISBN 3-8322-2963-9

ABISEA Band 38

Walter, Jörg

Simulationsbasierte Zuverlässigkeitsanalyse in der modernen Leistungselektronik

1. Aufl. 2004, 121 S.
ISBN 3-8322-3481-0

ABISEA Band 39

Schwarzer, Ulrich

IGBT versus GCT in der Mittelspannungsanwendung - ein experimenteller und simulativer Vergleich

1. Aufl. 2005, 170 S.
ISBN 3-8322-4489-1

ABISEA Band 40

Bartram, Markus

IGBT-Umrichtersysteme für Windkraftanlagen: Analyse der Zyklenbelastung, Modellbildung, Optimierung und Lebensdauervorhersage

1. Aufl. 2006, 185 S.
ISBN 3-8322-5039-5

ABISEA Band 41

Ponnaluri, Srinivas

Generalized Design, Analysis and Control of Grid side converters with integrated UPS or Islanding functionality

1. Aufl. 2006, 163 S.
ISBN 3-8322-5281-9

ABISEA Band 42

Jacobs, Joseph

Multi-Phase Series Resonant DC-to-DC Converters

1. Aufl. 2006, 185 S.
ISBN 3-8322-5532-X

ABISEA Band 43

Linzen, Dirk

Impedance-Based Loss Calculation and Thermal Modeling of Electrochemical Energy Storage Devices for Design Considerations of Automotive Power Systems

1. Aufl. 2006, 185 S.
ISBN 3-8322-5706-3

ABISEA Band 44

Fiedler, Jens

Design of Low-Noise Switched Reluctance Drives

1. Aufl. 2007, 176 S.

ISBN 978-3-8322-5864-1

ABISEA Band 45

Fuengwarodsakul, Nisai

Predictive PWM-based Direct Instantaneous Torque Control for Switched Reluctance Machines

1. Aufl. 2007, 141 S.
ISBN 978-3-8322-6210-5

ABISEA Band 46

Meyer, Christoph

Key Components for Future Offshore DC Grids

1. Aufl. 2007, 187 S.
ISBN 978-3-8322-6571-7

ABISEA Band 47

Fujii, Kansuke

Characterization and Optimization of Soft-Switched Multi-Level Converters for STATCOMs

1. Aufl. 2008, 199 S.
ISBN 978-3-8322-6981-4

ABISEA Band 48

Carstensen, Christian

Eddy Currents in Windings of Switched Reluctance Machines

1. Aufl. 2008, 179 S.
ISBN 978-3-8322-7118-3

ABISEA Band 49

Bohlen, Oliver

Impedance-based battery monitoring

1. Aufl. 2008, 190 S.
ISBN 978-3-8322-7606-5

ABISEA Band 50

Thele, Marc

A contribution to the modelling of the charge acceptance of lead-acid batteries - using frequency and time domain based concepts

1. Aufl. 2008, 165 S.
ISBN 978-3-8322-7659-1

ABISEA Band 51

König, Andreas

High Temperature DC-to-DC Converters for Downhole Applications

1. Aufl. 2009, 154 S.
ISBN 978-3-8322-8489-3

ABISEA Band 52

Dick, Christian Peter

Multi-Resonant Converters as Photovoltaic Module-Integrated Maximum Power Point Tracker

1. Aufl. 2010, 182 S.
ISBN 978-3-8322-9199-0

ABISEA Band 53

Kowal, Julia

Spatially resolved impedance of nonlinear inhomogeneous devices: using the example of lead-acid batteries

1. Aufl. 2010, 203 S.
ISBN 978-3-8322-9483-0

ABISEA Band 54

Roscher, Michael Andreas

Zustandserkennung von LiFeP04-Batterien für Hybrid- und Elektrofahrzeuge

1. Aufl. 2011, 186 S.
ISBN 978-3-8322-9738-1

ABISEA Band 55

Hirschmann, Dirk

Highly Dynamic Piezoelectric Positioning

1. Aufl. 2011, 146 S.
ISBN 978-3-8322-9746-6

ABISEA Band 56

Rigbers, Klaus

Highly Efficient Inverter Architectures for Three-Phase Grid Connection of Photovoltaic Generators

1. Aufl. 2011, 244 S.
ISBN 978-3-8322-9816-9

ABISEA Band 57

Kasper, Knut

Analysis and Control of the Acoustic Behavior of Switched Reluctance Drives

1. Aufl. 2011, 205 S.
ISBN 978-3-8322-9869-2

ABISEA Band 58

Köllensperger, Peter

The Internally Commutated Thyristor - Concept, Design and Application

1. Aufl. 2011, 214 S.

ISBN 978-3-8322-9909-5

ABISEA Band 59

Schoenen, Timo

Einsatz eines DC/DC-Wandlers zur Spannungsanpassung zwischen Antrieb und Energiespeicher in Elektro- und Hybridfahrzeugen

1. Aufl. 2011, 128 S.

ISBN 978-3-8440-0622-3

ABISEA Band 60

Hennen, Martin

Switched Reluctance Direct Drive with Integrated Distributed Inverter

1. Aufl. 2012, 141 S.

ISBN 978-3-8440-0731-2

ABISEA Band 61

van Treek, Daniel

Position Sensorless Torque Control of Switched Reluctance Machines

1. Aufl. 2012, 144 S.

ISBN 978-3-8440-1014-5

ABISEA Band 62

Bragard, Michael

The Integrated Emitter Turn-Off Thyristor. An Innovative MOS-Gated High-Power Device

1. Aufl. 2012, 164 S.

ISBN 978-3-8440-1152-4

ABISEA Band 63

Gerschler, Jochen B.

Ortsaufgelöste Modellbildung von Lithium-Ionen-Systemen unter spezieller Berücksichtigung der Batteriealterung

1. Aufl. 2012, 334 S.

ISBN 978-3-8440-1307-8

ABISEA Band 64

Neuhaus, Christoph R.

Schaltstrategien für Geschaltete Reluktanzantriebe mit kleinem Zwischenkreis

1. Aufl. 2012, 133 S.

ISBN 978-3-8440-1487-7

ABISEA Band 65

Butschen, Thomas

Dual-ICT- A Clever Way to Unite Conduction and Switching Optimized Properties in a Single Wafer

1. Aufl. 2012, 168 S.

ISBN 978-3-8440-1771-7

ABISEA Band 66

Plum, Thomas

Design and Realization of High-Power MOS Turn-Off Thyristors

1. Aufl. 2013, 113 S.

ISBN 978-3-8440-1884-4

ABISEA Band 67

Kiel, Martin

Impedanzspektroskopie an Batterien unter besonderer Berücksichtigung von Batteriesensoren für den Feldeinsatz

1. Aufl. 2013, 226 S.

ISBN 978-3-8440-1973-5

ABISEA Band 68

Brauer, Helge

Schnelldrehender Geschalteter Reluktanzantrieb mit extremem Längendurchmesser-verhältnis

1. Aufl. 2013, 192 S.

ISBN 978-3-8440-2345-9

ABISEA Band 69

Thomas, Stephan

A Medium-Voltage Multi-Level DC/DC Converter with High Voltage Transformation Ratio

1. Aufl. 2014, 226 S.

ISBN 978-3-8440-2605-4

ABISEA Band 70

Richter, Sebastian

Digitale Regelung von PWM Wechselrichtern mit niedrigen Trägerfrequenzen

1. Aufl. 2014, 126 S.

ISBN 978-3-8440-2641-2

ABISEA Band 71

Bösing, Matthias

Acoustic Modeling of Electrical Drives - Noise and Vibration Synthesis based on Force Response Superposition

1. Aufl. 2014, 188 S.

ISBN 978-3-8440-2752-5

ABISEA Band 72

Waag, Wladislaw

Adaptive algorithms for monitoring of lithium-ion batteries in electric vehicles

1. Aufl. 2014, 232 S.

ISBN 978-3-8440-2976-5

ABISEA Band 73

Sanders, Tilman

Spatially Resolved Electrical In-Situ Measurement Techniques for Fuel Cells

1. Aufl. 2014, 126 S.

ISBN 978-3-8440-3121-8

ABISEA Band 74

Baumhöfer, Thorsten

Statistische Betrachtung experimenteller Alterungsuntersuchungen an Lithium-Ionen Batterien

1. Aufl. 2015, 157 S.

ISBN 978-3-8440-3423-3

ABISEA Band 75

Andre, Dave

Systematic Characterization of Ageing Factors for High-Energy Lithium-Ion Cells and Approaches for Lifetime Modelling Regarding an Optimized Operating Strategy in Automotive Applications

1. Aufl. 2015, 196 S.

ISBN 978-3-8440-3587-2

ABISEA Band 76

Merei, Ghada

Optimization of off-grid hybrid PV-wind-diesel power supplies with multi-technology battery systems taking into account battery aging

1. Aufl. 2015, 184 S.

ISBN 978-3-8440-4148-4

ABISEA Band 77

Schulte, Dominik

Modellierung und experimentelle Validierung der Alterung von Blei-Säure Batterien durch inhomogene Stromverteilung und Säureschichtung

1. Aufl. 2016, 165 S.

ISBN 978-3-8440-4216-0

ABISEA Band 78

Schenk, Mareike

Simulative Untersuchung der Wicklungsverluste in Geschalteten Reluktanzmaschinen

1. Aufl. 2016, 126 S.

ISBN 978-3-8440-4282-5

ABISEA Band 79

Wang, Yu

Development of Dynamic Models with Spatial Resolution for Electrochemical Energy Converters as Basis for Control and Management Strategies

1. Aufl. 2016, 188 S.

ISBN 978-3-8440-4303-7

ABISEA Band 80

Ecker, Madeleine

Lithium Plating in Lithium-Ion Batteries:

An Experimental and Simulation Approach

1. Aufl. 2016, 154 S.

ISBN 978-3-8440-4525-3

ABISEA Band 81

Zhou, Wei

Modellbasierte Auslegungsmethode von Temperierungssystemen für Hochvolt-Batterien in Personenkraftfahrzeugen

1. Aufl. 2016, 175 S.

ISBN 978-3-8440-4589-5

ABISEA Band 82

Lunz, Benedikt

Deutschlands Stromversorgung im Jahr 2050

Ein szenariobasiertes Verfahren zur vergleichenden Bewertung von Systemvarianten und Flexibilitätsoptionen

1. Aufl. 2016, 187 S.

ISBN 978-3-8440-4627-4

ABISEA Band 83

Hofmann, Andreas G.

Direct Instantaneous Force Control: Key to Low-Noise Switched Reluctance Traction Drives

1. Aufl. 2016, 228 S.

ISBN 978-3-8440-4715-8

ABISEA Band 84

Budde-Meiwes, Heide

Dynamic Charge Acceptance of Lead-Acid Batteries for Micro-Hybrid Automotive Applications

1. Aufl. 2016, 157 S.

ISBN 978-3-8440-4733-2

ABISEA Band 85

Engel, Stefan P.

Thyristor-Based High-Power On-Load Tap Changers Control under Harsh Load Conditions

1. Aufl. 2016, 156 S.

ISBN 978-3-8440-4986-2

ABISEA Band 86

Van Hoek, Hauke

Design and Operation Considerations of Three-Phase Dual Active Bridge Converters for Low-Power Applications with Wide Voltage Ranges

1. Aufl. 2017, 231 S.

ISBN 978-3-8440-5011-0

ABISEA Band 87

Diekhans, Tobias

Wireless Charging of Electric Vehicles - a Pareto-Based Comparison of Power Electronic Topologies

1. Aufl. 2017, 151 S.

ISBN 978-3-8440-5048-6

ABISEA Band 88

Lehner, Susanne

Reliability Assessment of Lithium-Ion Battery Systems with Special Emphasis on Cell Performance Distribution

1. Aufl. 2017, 184 S.

ISBN 978-3-8440-5090-5

ABISEA Band 89

Käbitz, Stefan

Untersuchung der Alterung von Lithium-Ionen-Batterien mittels Elektroanalytik und elektrochemischer Impedanzspektroskopie

1. Aufl. 2016, 258 S.

DOI: 10.18154/RWTH-2016-12094

ABISEA Band 90

Witzenhausen, Heiko

Elektrische Batteriespeichermodelle: Modellbildung, Parameteridentifikation und Modellreduktion

1. Aufl. 2017, 266 S.

DOI: 10.18154/RWTH-2017-03437

ABISEA Band 91

Münnix, Jens

Einfluss von Stromstärke und Zyklentiefe auf graphitische Anoden

1. Aufl. 2017, 171 S.

DOI: 10.18154/RWTH-2017-01915

ABISEA Band 92

Pilatowicz, Grzegorz

Failure Detection and Battery Management Systems of Lead-Acid Batteries for Micro-Hybrid Vehicles

1. Aufl. 2017, 212 S.

DOI: 10.18154/RWTH-2017-09156

ABISEA Band 93

Drillkens, Julia

Aging in Electrochemical Double Layer Capacitors: An Experimental and Modeling Approach

1. Aufl. 2017, 179 S.

DOI: 10.18154/RWTH-2018-223434

ABISEA Band 94

Magnor, Dirk

Globale Optimierung netzgekoppelter PV-Batteriesysteme unter besonderer Berücksichtigung der Batteriealterung
1. Aufl. 2017, 210 S.
DOI: 10.18154/RWTH-2017-06592

ABISEA Band 95

Ilikso, Merve

Elucidation and Comparison of the Effects of Lithium Salts on Discharge Chemistry of Nonaqueous Li-O₂ Batteries
1. Aufl. 2018, 160 S.
DOI: 10.18154/RWTH-2018-223782

ABISEA Band 96

Schmalstieg, Johannes

Physikalisch-elektrochemische Simulation von Lithium-Ionen-Batterien: Implementierung, Parametrierung und Anwendung
1. Aufl. 2017, 168 S.
DOI: 10.18154/RWTH-2017-04693

ABISEA Band 97

Soltau, Nils

High-Power Medium-Voltage DC-DC Converters: Design, Control and Demonstration
1. Aufl. 2017, 176 S.
DOI: 10.18154/RWTH-2017-04084

ABISEA Band 98

Stieneker, Marco

Analysis of Medium-Voltage Direct-Current Collector Grids in Offshore Wind Parks
1. Aufl. 2017, 144 S.
DOI: 10.18154/RWTH-2017-04667

ABISEA Band 99

Masomtob, Manop

A New Conceptual Design of Battery Cell with an Internal Cooling Channel
1. Aufl. 2017, 167 S.
DOI: 10.18154/RWTH-2018-223281

ABISEA Band 100

Marongiu, Andrea

Performance and Aging Diagnostic on Lithium Iron Phosphate Batteries for Electric Vehicles and Vehicle-to-Grid Strategies
1. Aufl. 2017, 222 S.
DOI: 10.18154/RWTH-2017-09944

ABISEA Band 101

Gitis, Alexander

Flaw detection in the coating process of lithium-ion battery electrodes with acoustic guided waves
1. Aufl. 2017, 109 S.
DOI: 10.18154/RWTH-2017-099519

ABISEA Band 102

Neeb, Christoph

Packaging Technologies for Power Electronics in Automotive Applications
1. Aufl. 2017, 132 S.
DOI: 10.18154/RWTH-2018-224569

ABISEA Band 103

Adler, Felix

A Digital Hardware Platform for Distributed Real-Time Simulation of Power Electronic Systems
1. Aufl. 2017, 156 S.
DOI: 10.18154/RWTH-2017-10761

ABISEA Band 104

Becker, Jan

Flexible Dimensionierung und Optimierung hybrider Lithium-Ionenbatteriespeichersysteme mit verschiedenen Auslegungszielen
1. Aufl., 2017, 157 S.
DOI: 10.18154/RWTH-2017-09278

ABISEA Band 105

Warnecke, Alexander J.

Degradation Mechanisms in NMC Based Lithium-Ion Batteries
1. Aufl. 2017, 158 S.
DOI: 10.18154/RWTH-2017-09646

ABISEA Band 106

Taraborrelli, Silvano

Bidirectional Dual Active Bridge Converter using a Tap Changer for Extended Voltage Ranges
1. Aufl. 2017, 94 S.
DOI: 10.18154/RWTH-2018-228242

ABISEA Band 107

Sarriegi, Garikoitz

SiC and GaN Semiconductors: The Future Enablers of Compact and Efficient Converters for Electromobility
1. Aufl. 2017, 106 S.
DOI: 10.18154/RWTH-2018-227548

ABISEA Band 108

Senol, Murat

Drivetrain Integrated Dc-Dc Converters utilizing Zero Sequence Currents
1. Aufl. 2017, 134 S.
DOI: 10.18154/RWTH-2018-226170

ABISEA Band 109

Kojima, Tetsuya

Efficiency Optimized Control of Switched Reluctance Machines
1. Aufl. 2017, 142 S.
DOI: 10.18154/RWTH-2018-226697

ABISEA Band 110

Lewerenz, Meinert

Dissection and Quantitative Description of Aging of Lithium-Ion Batteries Using Non-Destructive Methods Validated by Post-Mortem-Analyses
1. Aufl. 2018, 139 S.
DOI: 10.18154/RWTH-2018-228663

ABISEA Band 111

Büngeler, Johannes

Optimierung der Verfügbarkeit und der Lebensdauer von Traktionsbatterien für den Einsatz in Flurförderfahrzeugen

1. Aufl. 2018, 171 S.

DOI: 10.18154/RWTH-2018-226569

ABISEA Band 112

Wegmann, Raphael

Betriebsstrategien und Potentialbewertung hybrider Batteriespeichersysteme in Elektrofahrzeugen

1. Auflage 2018, 184 S.

DOI: 10.18154/RWTH-2018-228833

ABISEA Band 113

Nordmann, Hannes

Batteriemanagementsysteme unter besonderer Berücksichtigung von Fehlererkennung und Peripherieanalyse

1. Aufl. 2018, 222 S.

DOI: 10.18154/RWTH-2018-228763

ABISEA Band 114

Engelmann, Georges

Reducing Device Stress and Switching Losses Using Active Gate Drivers and Improved Switching Cell Design

1. Aufl. 2018, 195 S.

DOI: 10.18154/RWTH-2018-228973

ABISEA Band 115

Klein-Heßling, Annegret

Active DC-Power Filters for Switched Reluctance Drives during Single-Pulse Operation

1. Aufl. 2018, 166 S.

DOI: 10.18154/RWTH-2018-231030

ABISEA Band 116

Burkhart, Bernhard

Switched Reluctance Generator for Range Extender Applications - Design, Control and Evaluation

1. Aufl. 2018, 194 S.

DOI: 10.18154/RWTH-2019-00025

ABISEA Band 117

Biskoping, Matthias

Discrete Modeling and Control of a versatile Power Electronic Test Bench with Special Focus on Central Photovoltaic Inverter Testing

1. Aufl. 2018, 236 S.

DOI: 10.18154/RWTH-2019-03346

ABISEA Band 118

Schubert, Michael

High-Precision Torque Control of Inverter-Fed Induction Machines with Instantaneous Phase Voltage Sensing

1. Aufl. 2019, 221 S.

DOI: 10.18154/RWTH-2018-231364

ABISEA Band 119

Van der Broeck, Christoph

Methodology for Thermal Modeling, Monitoring and Control of Power Electronic Modules

1. Aufl. 2019, 290 S.

DOI: 10.18154/RWTH-2019-01370

ABISEA Band 120

Hust, Friedrich Emanuel

Physico-chemically motivated parameterization and modelling of real-time capable lithium-ion battery models – a case study on the Tesla Model S battery

1. Aufl. 2019, 203 S.

DOI: 10.18154/RWTH-2019-00249

ABISEA Band 121

Ralev, Iliya

Accurate Torque Control of Position Sensorless Switched Reluctance Drives

1. Aufl. 2019, 154 S.

DOI: 10.18154/RWTH-2019-03071

ABISEA Band 122

Ayeng'o, Sarah Paul

Optimization of number of PV cells connected in series for a direct-coupled PV system with lead-acid and lithium-ion batteries

1. Aufl. 2019, 114 S.

DOI: 10.18154/RWTH-2019-01843

ABISEA Band 123

Koschik, Stefan Andreas

Permanenterregte Synchronmaschinen mit verteilter Einzelzahnsteuerung - Regelkonzepte und Betriebsstrategien für hochintegrierte Antriebssysteme

1. Aufl. 2019, 158 S.

DOI: 10.18154/RWTH-2019-03446

ABISEA Band 124

Farmann, Alexander

A comparative study of reduced-order equivalent circuit models for state-of-available-power prediction of lithium-ion batteries in electric vehicles

1. Aufl. 2019, 214 S.

DOI: 10.18154/RWTH-2019-04700

ABISEA Band 125

Mareev, Ivan

Analyse und Bewertung von batteriegetriebenen, oberleitungsversorgten und brennstoffzellengetriebenen Lastkraftwagen für den Einsatz im Güterfernverkehr in Deutschland

1. Aufl. 2019, 158 S.

DOI: 10.18154/RWTH-2019-04698

ABISEA Band 126

Qi, Fang

Online Model-predictive Thermal Management of Inverter-fed Electrical Machines

1. Aufl. 2019, 154 S.

DOI: 10.18154/RWTH-2019-08304

ABISEA Band 127

Kairies, Kai-Philipp

Auswirkungen dezentraler Solarstromspeicher auf Netzbetreiber und Energieversorger
1. Aufl. 2019, 140 S.
DOI: 10.18154/RWTH-2019-06706

ABISEA Band 128

Fleischer, Michael

Traction control for Railway Vehicles
1. Aufl. 2019, 162 S.
DOI: 10.18154/RWTH-2019-10570

ABISEA Band 129

Teuber, Moritz

Lifetime Assessment and Degradation Mechanisms in Electric Double-Layer Capacitors
1. Aufl. 2019, 150 S.
DOI: 10.18154/RWTH-2019-10071

ABISEA Band 130

Bußer, Christian

Investigation of Optimal Transformation Pathways towards 2050 for the Successful Implementation of a Sustainable Reduction of Carbon Emissions from Power Generation
1. Aufl. 2019, 204 S.
DOI: 10.18154/RWTH-2019-09975

ABISEA Band 131

Wienhausen, Arne Hendrik

High Integration of Power Electronic Converters enabled by 3D Printing
1. Aufl. 2019, 146 S.
DOI: 10.18154/RWTH-2019-08746

ABISEA Band 132

Kwiecien, Monika

Electrochemical Impedance Spectroscopy on Lead-Acid Cells during Aging
1. Aufl. 2019, 138 S.
DOI: 10.18154/RWTH-2019-09480

ABISEA Band 133

Titiz, Furkan Kaan

A Three-phase Low-voltage Grid-connected Current Source Inverter
1. Aufl. 2019, 128 S.
DOI: 10.18154/RWTH-2020-00458

ABISEA Band 134

Wünsch, Martin

Separation der Kathodenalterung in Lithium-Ionen-Batteriezellen mittels elektrochemischer Impedanzspektroskopie
1. Aufl. 2019, 177 S.
DOI: 10.18154/RWTH-2019-11017

ABISEA Band 135

Badede, Julia

Modeling and Steering of Multi-Use Operation with Uninterruptible Power Supply Systems - utilizing the example of lead-acid batteries
1. Aufl. 2020, 282 S.
DOI: 10.18154/RWTH-2020-05456

ABISEA Band 136

Kleinsteiberg, Björn

Energy Efficiency Increase of a Vanadium Redox Flow Battery with a Power-Based Model
1. Aufl. 2020, 163 S.
DOI: 10.18154/RWTH-2020-06092

ABISEA Band 137

Cai, Zhuang

Optimization of dimension and operation strategy for a wind-battery energy system in German electricity market under consideration of battery ageing process
1. Aufl. 2020, 144 S.
DOI: 10.18154/RWTH-2020-06525

ABISEA Band 138

Sabet, Pouyan Shafiei

Analysis of Predominant Processes in Electrochemical Impedance Spectra and Investigation of Aging Processes of Lithium-Ion Batteries with Layered Oxide Cathodes and Graphitic Anodes
1. Aufl. 2020, 136 S.
DOI: 10.18154/RWTH-2020-07683

ABISEA Band 139

Angenendt, Georg

Operation, Optimization and Additional Market Participation of Households with PV Battery Storage System and Power-to-Heat Application
1. Aufl. 2020, 221 S.
DOI: 10.18154/RWTH-2020-05200

ABISEA Band 140

Oberdieck, Karl Friedrich

Measurement and Mitigation of Electromagnetic Emissions of Propulsion Inverters for Electric Vehicles
1. Aufl. 2020, S.
DOI: 10.18154/RWTH-2020-

ABISEA Band 141

Bubert, Andreas Martin

Optimierung des elektrischen Antriebsstrangs von Elektrofahrzeugen mit Betrachtung parasitärer Ströme innerhalb der elektrischen Maschine
1. Aufl. 2020, S.
DOI: 10.18154/RWTH-2020-09556