AKI KUUSISTO
HARDWARE-IN-THE-LOOP TEST SETUP FOR BATTERY
MANAGEMENT SYSTEMS

Master's Thesis

Examiner: Lecturer Risto Mikkonen
Examiner and topic approved on
1.11.2017

ABSTRACT

AKI KUUSISTO: Hardware-In-the-Loop Test Setup For Battery Management Systems
Tampere University of Technology
Master of Science Thesis, 48 pages
November 2017
Master's Degree Programme in Electric Engineering
Major: Renewable Energy
Examiner: Lecturer Risto Mikkonen

Keywords: BMS, Battery Management System, Lithium-ion, Li-ion, $LiFePO_4$, HIL, Hardware-In-the-Loop, dSPACE, Simulink, Valmet Automotive, testing, simulation

Reduction of carbon dioxide emissions is one of the largest challenges of our society. Electric vehicles (EV) are one potential aspect of the solution to this challenge. EVs require energy storage, and lithium-ion (Li-ion) batteries are the best commercial solution at the moment. Li-ion batteries require management for both usability and safety. Battery management system (BMS) development and testing are demanding tasks, because EVs and other applicable systems are usually very complex products, and the successful management of the batteries is required for safe operation. Failure to keep the batteries within their safe operating parameters could result in broken equipment and even loss of life.

A Li-ion battery pack consists of battery cells and the BMS. Battery pack may consist of multiple modules of cells, and if the BMS is of a distributed construction, each module of cells has their own measuring card. Measuring card is in contact with the cells and supplies the voltage information to the main BMS via logic level signals. Both the cells and the measuring card may be mathematically modelled, which means that a real time simulation of them can be created. Using a programmable electronic control unit (ECU) to run the simulation and communicate with the BMS, a logic-level Hardware-In-the-Loop (HIL) system can be created, with the BMS as the hardware to be tested.

We created a model of battery modules on MathWorks Simulink software. The simulated modules run on a dSPACE MicroAutoBox II (MABX II) ECU. Simulation consists of individual cells and measuring cards, and it communicates with the BMS via Serial Peripheral Interface (SPI). Additional sensors and actuators are also simulated, and the simulation can be interacted with from the dSPACE ControlDesk software.

The BMS testing is safe and efficient with our HIL system. Going through the established test procedures is an order of magnitude faster than before, and some aspects of BMS functionality could be properly verified for the first time. Our system also found some new errors and potential faults in the tested BMS. HIL testing was found to be a useful solution for a BMS developer, and the tools offered by MathWorks and dSPACE to be most suitable for this type of a project.

PREFACE

Welcome to my thesis, glad to have you along. The first half of this work discusses the basics of electric vehicles, batteries and battery management systems, so for those familiar with the subjects, the interesting topics begin in chapter 5. There, the system about to be simulated is described, and starting from 5.4 the simulation itself is specified and described, along with pictures of Simulink code for those who, like me, find them entertaining. The actual testing and results may be found in chapter 6.

I have many people to thank for getting this far. Juuso Kelkka at Valmet Automotive gave me the opportunity and Stenka Keto, also at VA, offered the topic and guidance to complete it. Markku Rajamäki and Petri Nieminen helped with the battery science and proofreading, they along with all the colleagues at VA were supportive of my work. Lasse Kaitila and Mikael Bragge were especially helpful with the completion of the actual project. Thanks are also due to Raija Keni for the help with the language of the thesis. I may not have caught all the errors in my text, but without her I wouldn't have caught any.

Necessary skills for a project like this were gained at Tampere University of Technology. From there I owe the most to Risto Mikkonen and Aki Korpela, the lecturers responsible for most of my major studies. Without Risto as a demanding examiner and Aki as an exemplary explainer, I'm sure my thesis would have made quite dreadful reading. Thanks to them, I have some measure of confidence in my ability to explain the following concepts with sufficient clarity, but the result is of course for you to decide.

As always, all errors are mine.

Uusikaupunki, 24.11.2017

Aki Kuusisto

INDEX

ABBREVIATIONS AND NOTATION

BMS	Battery Management System
CAN	Controller Area Network
CO_2	Carbon Dioxide
CS	Chip Select
EV	Electric Vehicle
ECM	Equivalent Circuit Model
ECU	Electronic Control Unit
EOC	End Of Cycle
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
HIL	Hardware In the Loop
I/O	Input/Output
ICE	Internal Combustion Engine
Li-ion	Lithium-ion
$LiFePO_4$	Lithium Ferrite Polymer
MABX II	MicroAutoBox II
MISO	Master In, Slave Out
MOSI	Master Out, Slave In
NiCd	Nickel-Cadmium
NiMH	Nickel metal hydride
OCV	Open Circuit Voltage
PGI	Programmable Generic Interface
SCLK	Serial Clock
SEI	Solid Electrolyte Interface
SLI	Starting, Lighting and Ignition
SPI	Serial Peripheral Interface
SOC	State of Charge
TUT	Tampere University of Technology
VA	Valmet Automotive
V	Volt
A	Ampere
C	Coulomb
U	*voltage*
I	*current*
R	*resistance*

1. INTRODUCTION

Energy usage and environmental impact go hand in hand. Humans have converted available resources to energy for longer than there has been written history. Energy usage began with combustion. We have been burning fuel for warmth, cooking and eventually manufacturing, and combustion remains the main energy generation method to this day. The whole basis of our society, industrial revolution and all, was built on some form or other of combustion-powered machinery. Consequences of this approach to technological evolution are evident today.

Combustion generates emissions. The most widely discussed environmental issue is the carbon dioxide (CO_2), which acts as a greenhouse gas, contributing to global warming [1,2]. Small-scale burning of wood may be considered carbon neutral and renewable energy, because trees grow, and growing trees use and capture CO_2. Used in a larger scale, however, their capacity to capture CO_2 is quickly overcome. The more industrialized our society became, the more combustion was required, and plain wood became insufficient. The time we call the Industrial Revolution was mainly fueled by coal, and coal is still used in electric energy generation and industrial manufacturing [3,4]. Today the CO_2 level is higher than it has been in 800 000 years.[50] The causes and effects may be debatable, but that is the recorded fact, and it is most probable that we as a society have had something to do with it.

Oil was first used for lighting, and it was first produced from whale fat. The discovery of drilling for oil from the ground might have saved the whales from extinction, but for the ecology as a whole, it was not perhaps so lucky. Crude oil industry began in the 1800s, and veritably boomed. As oil in its various forms found its way from lighting and lubrication to powering power plants, ships and eventually automobiles, the whole global economy was taken over by it. Oil played a big part in world politics, and during the World Wars, the importance of controlling the oil supplies was realized by everyone.[5]

Despite oil crises and resulting developments in energy efficiency and renewable energy, emissions continue to rise. Improving living standards around the world and the indifference to environmental concerns in some developing countries have made it thus far impossible to reverse the trend, so more actions are required. Making energy production cleaner and more efficient is one aspect, but the one we will concentrate on is the energy consumption. Small internal combustion engines in mobile machinery are inherently inefficient compared to large power plants, yet somehow the energy needs to be brought to where it is used.

Chapter 2 presents the electric vehicle (EV). Electric car is not a new invention. In the turn of the 1900s, steam, electricity and internal combustion were in a relatively even competition as the source of motion for the masses[6]. Before the coming together of the internal combustion engine, naphtha, electric starter and the assembly line, any one of these three avenues of development could have been the winner. The eventual winner in this competition was so clear that, for a century, roads belonged to internal combustion with steam fading to obscurity and electricity hiding in the margins of mobility. Modern developments in battery technology have only recently made electricity competitive again, as electric cars finally compete with internal combustion engine (ICE) cars in both performance and range. Industrial and agricultural machinery have had electric alternatives nearly since their invention, and now they are also becoming more tempting in every respect compared to internal combustion versions.[7,8,9]

Chapter 3 discusses batteries and battery management systems (BMS). Modern batteries often utilize lithium-ion (Li-ion) technology, and are composed of multiple cells. These cells always have some manufacturing and material based variety, so cell voltage balancing is important. Li-ion cells are also potentially dangerous if their safe operating conditions are exceeded. For these reasons, a BMS is an important part of Li-ion battery packs.[10]

Different applications and batteries pose different demands for BMSs, so they are being constantly developed and improved. Developing requires testing, and testing with physical systems is expensive and time-consuming. Testing and development are discussed in **Chapter 4**. Modern solution to testing is a simulated system. Simulation of a battery running inside a purpose built electronic control unit (ECU) can communicate with a BMS as if it was the actual battery module. As the simulation can be controlled by the user, BMS functionality and fault handling can be tested safely and quickly, saving time and money for the developer. This method of testing is referred to as Hardware-In-the-Loop (HIL).

Creating a HIL test setup is the subject of my Master's Thesis. HIL test setup is to be used for BMS testing, and the purpose is to make the testing easier, safer, more thorough and more efficient. ECUs capable of running a simulation are readily available for industry, and many are made specifically for HIL-testing. The program to simulate a specific battery module to test a particular BMS cannot usually be purchased off-the-shelf. For a developer making both the battery pack and the BMS, it then makes sense to also develop the simulation for the new system.

The system to be simulated, the HIL test setup and the development process are described in **Chapter 5**. **Chapter 6** collects the experiences and test results with the finished setup, and **Chapter 7** introduces the conclusions.

2. ELECTRIC VEHICLE

The EV is an automotive (self-propelled) unit that relies on electricity for the source of motive force. Electric energy is created elsewhere and stored within the vehicle. The most common storage method is a battery, but capacitors and hydrogen are also being developed. Lithium-ion battery is the clear leader in electric cars, and it is finding its way to other machines as well. Some electric applications are depicted in a montage in Figure 2.1.

Figure 2.1: Examples of electric vehicles[11-17]

An electric vehicle is often a direct replacement for an internal combustion engine (ICE) vehicle. Electric cars are becoming common and working machines like forklifts have been available as electric since their invention.[7,8,9] With the improvements in battery technology, range (or operating time) of EVs has come closer to machines with primary fuel storage, and thus the number of EVs is rising ever faster.[9]

2.1   Foundation for the Electric Vehicle

The human imagination made the idea of a self-propelled vehicle almost inevitable. Prime movers such as windmills and water wheels were employed for stationary tasks,

and they inspired inventors to find other uses for the available energy. The designs of vehicles found in the first surviving sketches were usually impractical and impossible, but in 1599 Simon Stevin's sail wagon was capable of carrying 28 persons at a claimed speed of 12 km/h. Sail wagon has little practical use, but it was the first recorded truly automotive vehicle. Other inventors of this time experimented with clockworks and springs, but none of the innovations prior to the 18th century proved a viable power source for an automotive vehicle.[6]

The discovery of a practical source of motive force took nearly a century after the sail wagon, and another century for it to be employed in a vehicle. Steam engine was employed in a coach in 1802, and the development of the internal combustion powered vehicles was not far behind. For the next hundred years or so, steam, electricity and internal combustion were in a competition to be the prime mover of choice for automotive vehicles. For most of this time ICE was the definite underdog. Torque characteristics, numerous required adjustments during running and the need for specialized fuel made ICEs impractical and only suitable for mechanically-minded people. Steam engines also required driver input, to stoke the boiler and direct the steam, but since steam doesn't require revolutions to produce torque, they were generally easier and smoother to use. Electric motors were the simplest of all, and for a long time the vehicle chosen especially by women.[6]

At the end of the 19th century, electric cars seemed to be the most promising option. Invention of the rechargeable lead-acid battery made them practical, and the efforts of French car manufacturers Camille Jenatzy and Charles Jeantaud gave them another advantage – speed. Rail vehicles had already exceeded the barrier of 100 km/h, but Jeantaud's *Jamais Contente (Never Satisfied)* was the first land vehicle to do so, in 1899. This record stood for three years until a steam vehicle broke it in 1902.[6]

The beginning of the end for steam and electric vehicles can be attributed to the electric starter. In 1912 Charles F. Kettering developed a combined starter and charger, which made hand-cranking obsolete. This made ICE close enough in practicality to the other two options to become popular. In the 1920s steam cars were all but obsolete, and electric cars soon followed. There were no new inventions for either of them for decades to come, not until the lithium-ion battery in the turn of the millennium. Steam remains consigned to stationary plants, but electric car is back in the competition.[6]

Fossil fuels are the main cause of pollution. The efficiency of an internal combustion engine has a physical limit according to Carnot's cycle, and in actual use they never reach even that high. In large power plants the efficiency may be greatly improved, reducing the strain on resources per used energy unit. Stationary energy creation can also be accomplished with no emissions, or at least in a carbon neutral manner. Energy is rarely used close to the power plant however, so it needs to be transported to the user. In static installations like houses and factories this is relatively easy using the electric grid,

but mobile units like cars and machines cannot be constrained to cables. Electricity needs to be stored, and this has been a challenge for engineers ever since the 19th century.

2.2   Electric Vehicle Principles

Electric vehicles do not produce their own energy, so they do not produce any local emissions. The energy is stored in electric fields, in reversible chemical reactions or in liquid hydrogen. The most common method by far is the chemical one, storing energy in batteries.
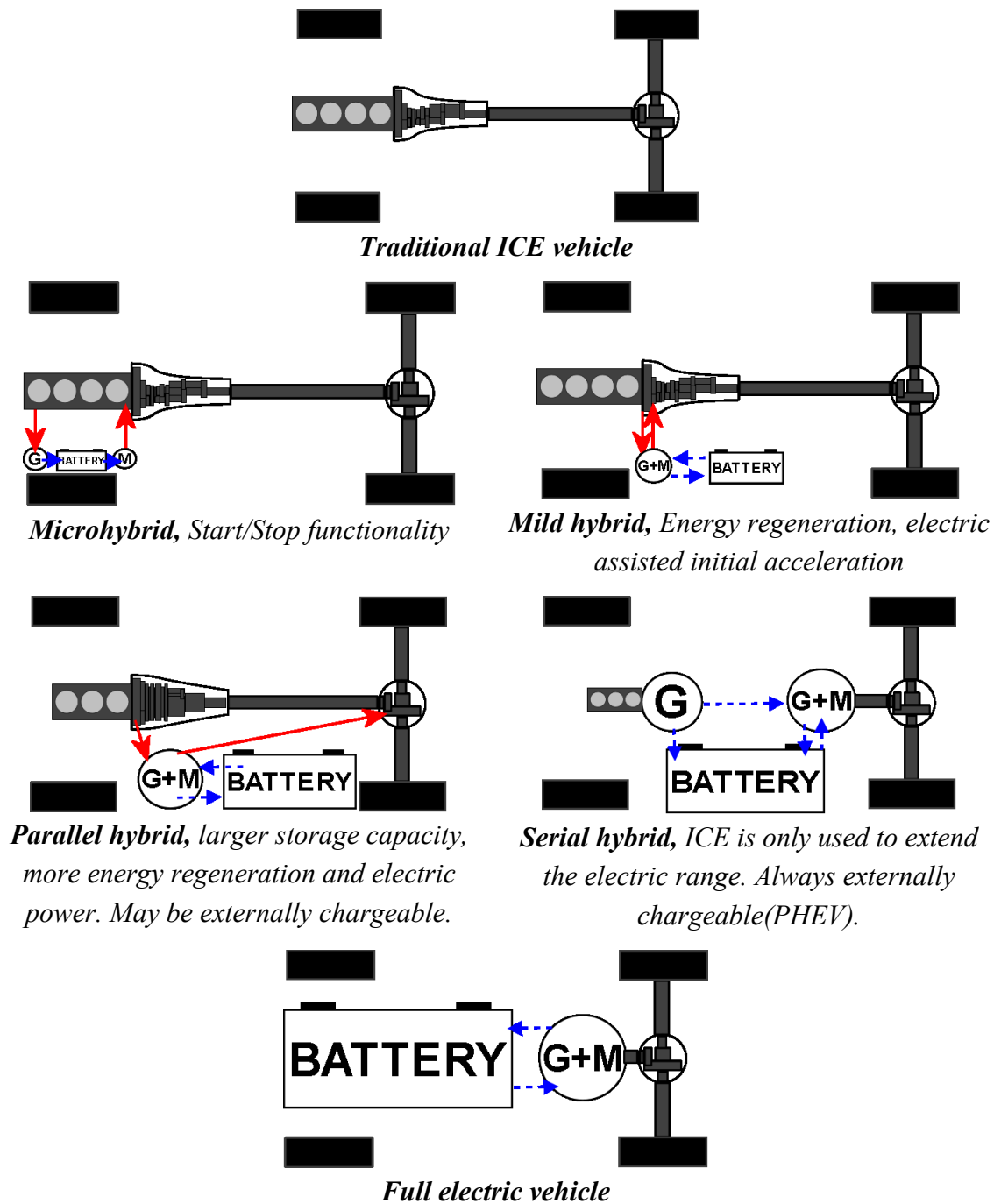
The power delivery of an electric motor differs from that of an ICE. While ICE's require revolutions to create useful torque and power, an electric motor generates maximum torque from the moment it begins to turn. This makes electric motors especially effective in tasks where intensive and intermittent power delivery is required. EVs are also easy to start and operate, as there is no need for a clutch or transmission.[6]

The energy consumption characteristics of ICEs and electric motors are also very different. Electric motor is always ready to give out power, but an ICE needs to be running to be in its ready state. An idling engine consumes fuel, and depending on the construction and fuel chosen, with a very low efficiency. This is not an issue on applications where the engine is required for constant power running, but for intermittent operation with idle times between power requirements, electric motors are much more energy efficient.

Not all EVs are purely electric. Especially in passenger cars, there's a wide variety of hybrid solutions. Hybrid electric vehicles have both an ICE and an electric motor, and more electricity storage capacity than required for normal starting-lighting-ignition (SLI) systems. Figure 2.2 collects the main topologies, but is by no means a comprehensive list. The first and last topologies are of a traditional ICE vehicle (SLI electricity omitted) and a pure electric vehicle.

Simplest form of a hybrid is the "microhybrid," a car with only slightly larger batteries that can handle start+stop-functionality. The engine can be turned off when the car is stationary and quickly started again when the car needs to move. The positive effects are only noticeable in extended urban driving or in emissions tests. Most ordinary cars produced today have this functionality. Enlarging the storage capacity and introducing regenerative braking and electric acceleration assistance creates a mild hybrid, like the first Honda Insight and Toyota Prius models. Due to electric assistance, the ICE can be downsized. The combustion cycle of the ICE is also often optimized, and continuously-variable transmissions utilized for economical constant-speed running. This is also known as a parallel hybrid, as the ICE and electric motor work in parallel, both operating on the drivetrain simultaneously. These variants use energy that is first created by the ICE. Hybrids that can be charged from an electrical socket are often referred to as

full hybrids, or plug-in hybrids (PHEV). Newer Prius models and most hybrids today, like the Mercedes Benz GLC 350e have a plug-in charge capability, and they have the ability to move under electric power alone, usually for 10 to 50 kilometers.

Traditional ICE vehicle

Microhybrid, *Start/Stop functionality*

Mild hybrid, *Energy regeneration, electric assisted initial acceleration*

Parallel hybrid, *larger storage capacity, more energy regeneration and electric power. May be externally chargeable.*

Serial hybrid, *ICE is only used to extend the electric range. Always externally chargeable(PHEV).*

Full electric vehicle

Figure 2.2 *Vehicle topologies*

PHEVs have a larger battery capacity than mild hybrids, but in the parallel configuration the main source of motion is still the ICE. In the serial configuration there is no physical link between ICE and drivetrain. This kind of vehicle is called a range-extender hybrid, or plug-in E-REV. In a range-extender hybrid the ICE is only used in parallel to the batteries to feed the electric motor, or after the batteries are empty. Pure electric

range is long enough for urban driving or moderate commuting. BMW i3 range extender model is one example and the Fisker Karma another.

One variant of EVs worth of mentioning still is the hydrogen fuel cell vehicle. These vehicles have an electric drivetrain, but the source of energy is a fuel cell instead of a battery. The fuel cell converts the energy of hydrogen into electricity, and produces water from hydrogen and air in the process. These vehicles need storage capacity for liquid hydrogen and hydrogen needs to be created by either electrolysis of water or from natural gas. Fuel cell vehicles, like other EVs, also suffer from the lack of a refueling infrastructure, and this combined with challenges in fuel cell development and clean hydrogen production have thus far prevented their widespread adoption into market.

2.3   Challenges of the Electric Vehicle

Energy density of fossil fuels is still unparalleled by electricity storage methods. Despite the advantage in energy efficiency and the developments in battery technology the range of electric vehicles remains shorter than that of an equivalent ICE vehicle.[9]

Charging times of batteries far exceed those of refueling times of fuel tanks. A semi-trailer truck can be refueled in less than 10 minutes, but charging even a small electric vehicle from an ordinary socket takes hours. Dedicated quick chargers can shorten the time for a practical amount of recharge to 30 minutes, but charger networks are much smaller than traditional gasoline station networks, and the charging time compared to refueling time is still an order of magnitude longer.

Combustion engines have been extensively developed for over a century and they are quite reliable in any environment. Even in the coldest winters a gasoline or diesel engine may be made to run, with external power and heat if necessary, and when it runs it will not stop due to varying weather. Electric motors are actually even more reliable, with less moving parts and a motive force that doesn't rely on thermodynamic reactions, but the source of electricity is another matter altogether. Most batteries rely on chemical reactions, and in extreme colds these reactions are not so readily realized. Lead-acid batteries suffer from capacity fade and restricted power output, and Li-ion batteries will be permanently damaged if an attempt is made to charge them in sub-zero cell temperatures.

The development of batteries with higher energy densities is ongoing. Large batteries require a large number of cells, which need to be kept at the same voltage. Charging and discharging currents need to be controlled depending on the temperature and the state of charge. Operation in a variety of environments requires heating and cooling to keep the batteries in their optimum operational temperature. These requirements mean that some form of battery management is usually needed. Electric motor is a simple and reliable component, but the inverter that drives it needs precise control. Control systems for bat-

teries, inverters and chargers are costly, and have had much less development time than internal combustion engines. Large battery packs also make electric machines generally heavier than their ICE competitors.

2.4 Future of Electric Vehicles

Replacing ICE vehicles with electric ones seems to be a growing trend. Environmental awareness and tax incentives make EVs attractive to consumers and practical for companies. Many people are willing to buy cars with electric motors based on the image alone, and as they begin to have actual useful advantages over traditional vehicles, EV market will increase even faster.[9]

The current development of EVs is fast and competitive. Tesla seems to have taken the industry by surprise and nobody likes to be left too far behind. At the moment the technological field is still wide and varied. Different manufacturers favor different types of batteries, motors and inverters, and different methods for overcoming obstacles and perceived deficiencies. There is still no global charging standard, for instance, and charging networks need to be developed further to facilitate a widespread EV use.[18]

Sometimes technological development race produces only one clear winner, such as happened for ICE. Electric starter, mass production and gasoline refinement made other options obsolete, even though steam and electricity had been serious contenders for over a century. Sometimes more than one competitor remains, like Otto-cycle and diesel engines, neither of which ever proved clearly better than the other. Now the challengers are different types of batteries, different materials of which these batteries are made of, and even fuel cells as possible alternatives. Motors and inverters are another difference between manufacturers, but the relevant variable for this thesis are the batteries.

In the present competition, manufacturers are not eager to commit to a single solution nor supplier.[18] The development and comparison for each application must be relatively fast. Machinery and especially automobiles are complex products, operating in an even more complex environment. The development and testing need to take into account numerous variables, while still being fast enough to stay in the competition. EVs are still considered new by the major public, which means that they are also considered potentially unreliable and unsafe. Every time an EV is implicated in an accident, it makes the headlines, and this makes it even more important to make sure that the product is safe before being released.

The market share of electric vehicles is expected to continue to grow. The number of electric cars globally reached one million in 2015, and in 2016 the stock reached 2 million. EV growth rate was slower in 2016 than in the previous two years, while the charging station growth rare was higher. The growth from 2010 to 2016 is shown in

Figure 2.3. Lower curve shows only the number of battery electric vehicles (BEV) and the upper curve includes plug-in hybrid vehicles (PHEV).[9]
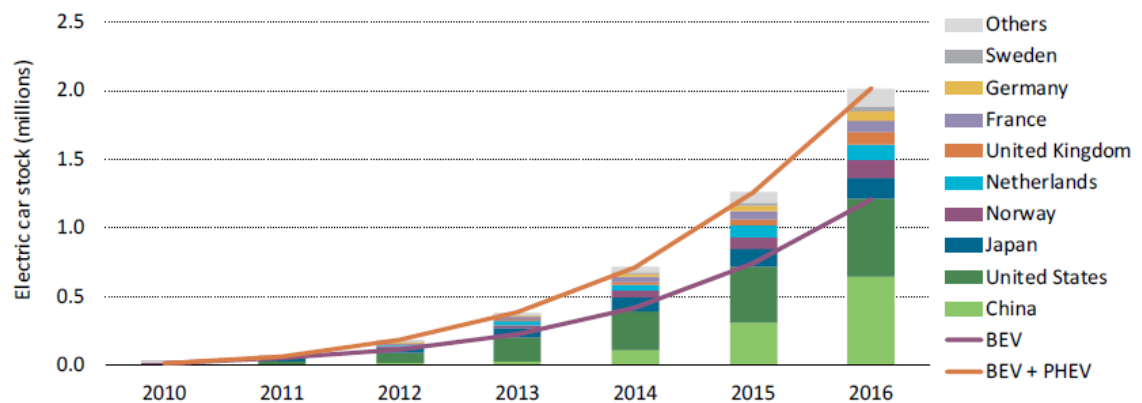
Figure 2.3 *Evolution of the global electric car stock, 2010-16[9]*

China has become the country with the largest electric car stock. Norway is the leader in market penetration, with a 29% market share. Even if the rate of growth seems to be slowing, the growth of EV stock does continue. The next few years will see the release of numerous compact- and family-class EVs, which will probably affect the growth rate positively.[9]

3. BATTERIES AND BATTERY MANAGEMENT SYSTEMS

Electricity is not easily conserved. Pure electricity can only be contained in electric conductors, and in reasonable quantities only in long loops of superconductive material. In superconductors there is no resistance that converts the energy into heat, but they require extreme cooling systems to stay in superconductive state. The superconductor materials and the cooling systems make them both expensive and heavy, so ill-suited for mobile applications. Electric potential is somewhat easier to store, either in the magnetic fields of coils or electric fields of capacitors, or in reversible chemical reactions within batteries.

For a mobile application, only the battery has proven a viable commercial option. The development of batteries began in 1700s, when it was discovered that certain metals connected by an electrolyte had enough electric potential between them to produce current. Luigi Galvani noticed in 1791 that a frog leg produced an electric discharge, which was in 1972 corrected by Alessandro Volta as being caused by the metals connected together via the muscle tissue, instead of the tissue itself. In 1800 Volta displayed his device based on this principle, the Volta pile. The Volta pile consisted of zinc and silver discs, separated by cardboards infused with acid solution. This invention enabled the rapid development of electrochemical studies, but Volta's rejection of the chemical aspect delayed the realization of a correct theory.[19,20]

The rechargeable, or secondary, lead-acid battery was invented in 1859 by Gaston Planté [21,22]. Planté studied the electrolytic reactions of different metals in diluted sulfuric acid, concentrating on the applications of the secondary current that the electrodes released after they were disconnected from an electrical source. This secondary current wasn't Planté's discovery, as it was first noticed by Johann Wilhelm Ritter in 1802, but Gaston Planté was the first to employ this discovery in an invention.[21] Eventually he found that for his application lead was the most suitable, as it had a high electric potential, and properly prepared didn't dissolve in the acid electrolyte. The electric charge was retained in the battery for days and even months, and the capacity faded very slowly. This was the first rechargeable battery with practical applications, such as lighting and brakes in trains, rechargeable lights at home and mines, detonation of explosives over long distances and numerous experiments with high voltage.[22] The lead-acid battery is still in use today.

The alkaline batteries were developed to improve upon the lead-acid battery performance. Numerous chemistries were researched, but the nickel-cadmium (NiCd) and nickel-metal-hydride (NiMH) chemistries have been the most successful. The NiMH is the most common alkaline battery today. The lithium-ion is the latest rechargeable battery development. Despite the need for a separate management system, Li-ion batteries are the most common solution for consumer electronics today, due to their superior power and energy per size and weight. Most electric cars, like Tesla, use lithium ion cells, and the development work on improving their specific energy and energy density is ongoing.

3.1 Chemistry and the lead-acid battery

Battery as a term used to refer to a group of smaller units stacked together. The original inventors of electric energy carrying cells had to group these together to create higher voltages and capacities, so naturally they began to be called batteries of cells, and nowadays battery is the given name for electrochemical storage units, regardless of the amount of cells they contain.[19,22]

The operation of a chemical battery is based on reduction and oxidation reactions (redox)[19,22,23,24]. A good example is the common lead-acid battery, where both electrodes are made of lead. The lead-acid battery is a secondary battery, which means that it can be recharged, as opposed to primary batteries which cannot. A charged lead acid battery has an anode, the negative electrode, of lead, and the cathode, or the positive electrode, of lead dioxide. The electrolyte is diluted sulfuric acid. When the battery is discharged, lead dioxide on the cathode is reduced

$$PbO_2 + 3H^+ + HSO_4^- + 2e^- \rightarrow PbSO_4 + H_2O, \tag{3.1}$$
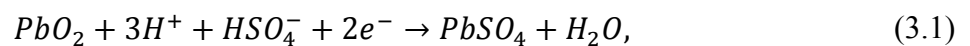
it gains an electron and consumes sulfuric acid to produce lead sulfate and water. On the anode the lead is oxidized upon discharge

$$Pb + HSO_4^- \rightarrow PbSO_4 + H^+ + 2e^-, \tag{3.2}$$
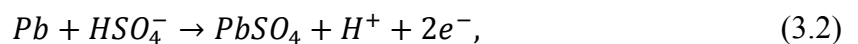
releasing an electron to the external circuit. Lead sulfate is also produced on the anode, in addition to hydrogen ions. Hydrogen ions transfer to the cathode via electrolyte, where they continue the chemical reaction between lead oxide and sulfuric acid.

The complete cell reaction is then

$$PbO_2 + Pb + 2H_2SO_4 \leftrightarrow 2PbSO_4 + 2H_2O, \tag{3.3}$$
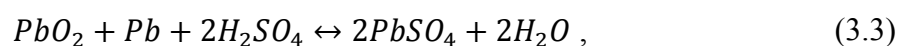
and we can see that the concentration of sulfuric acid drops when the battery is discharged. The difference in acid concentration is from approximately 30-40 % for full to 12-24 % for a discharged battery.[23,24]

Lead-acid batteries are safe and robust and they can be found in every ICE-powered automobile. They function in every weather, although in freezing temperatures their ability to release high currents is weaker, and a discharged battery with low concentration of sulfuric acid can even freeze, rendering it inoperable. Downsides of lead-acid batteries are their poor specific energy and energy density, meaning that for a given amount of storable energy and available power, lead-acid batteries are big and heavy. While this is not an issue in a conventional car, where only the energy for SLI needs to be stored, it makes these batteries impractical for a fully electric moving machine. They have been used in larger cargo-handling or working machines, and uninterruptible power supplies(UPS) but have been mostly replaced by other types of batteries in all but automotive SLI and industrial UPS applications.[23]

3.2   Alkaline batteries

Thomas Edison was not satisfied with the lead-acid battery. He was intrigued by the idea of an electric car, and deemed the lead-acid battery too heavy for this application. After three years of experiments with his team of scientists, he filed a patent in 1901 for an alkaline battery. This battery used iron as the anode, nickel oxide as the cathode and potassium hydroxide as the electrolyte. Although the alkaline battery didn't prove universally better than the lead-acid battery, it found its use in various applications, mostly through Edison's perseverance. Long development time meant that the original market he had envisioned had faded along with the popularity of the electric car, but Edison's thorough study of possible present and future applications resulted in widespread deployment of his nickel-iron(NiFe) alkaline battery.[25]

The NiCd battery was created in 1899 in Sweden by Waldemar Jungner. NiCd batteries had a much higher energy density than lead-acid batteries. They were robust and reliable, with a very stable voltage during discharge, and the internal resistance on NiCd batteries is smaller than that of lead-acid batteries. These attributes make NiCd batteries suitable for applications where high security or a high rate discharge is needed. The problem with NiCd is the high toxicity of cadmium.[26,29]

Edison's NiFe alkaline battery suffered from the instability of the iron anode. Nickel-zinc (NiZn) battery, another attempt at a better rechargeable cell, had a limited service life due to zinc dendrite growth on the anode. Manganese-zinc batteries, another development, suffers from stability issues. NiCd was clearly the most useful of these solutions, until the development of NiMH battery. NiMH offers a much higher energy density than NiCd, and replaces the toxic cadmium with a hydrogen-storing alloy. NiMH is the dominant rechargeable alkaline battery today.[27]
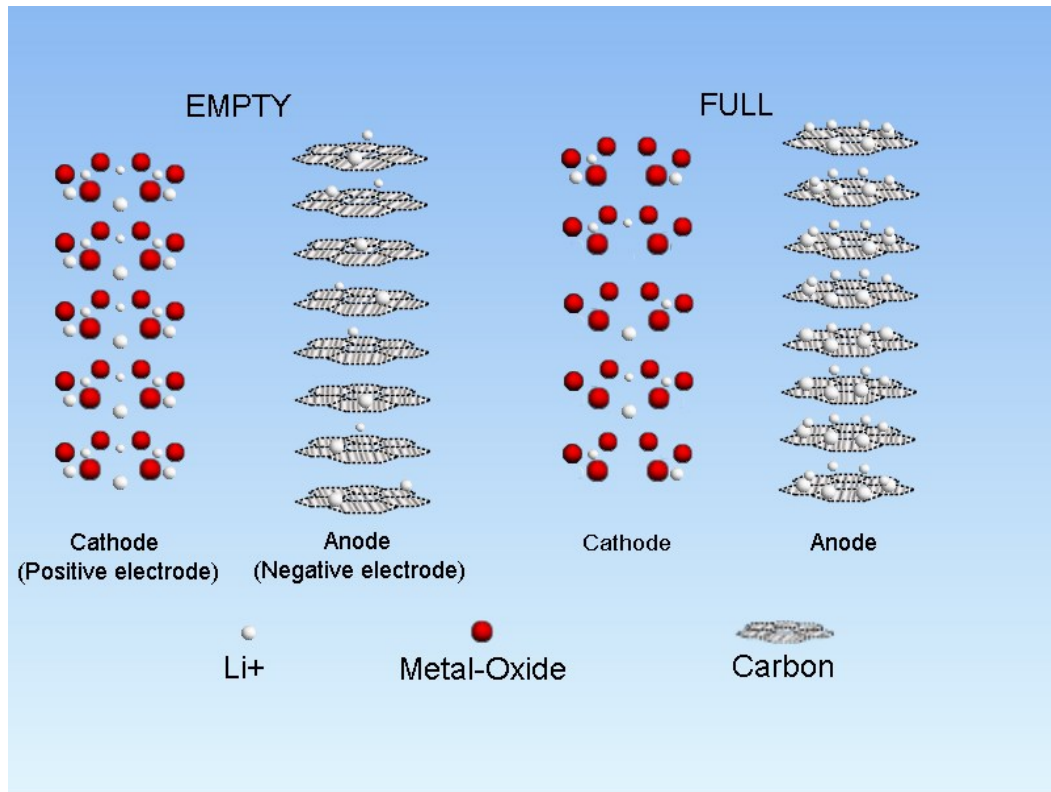
3.3 Li-ion batteries

Lithium batteries were originally developed as primary batteries. Implanted medical devices, such as pacemakers, used a zinc-mercury battery, which had to be changed every two years. Changing the battery on an implanted device requires surgery, so a longer-lasting battery was required. The development of the lithium-iodine primary battery increased the operational life between replacements to 6 and even 7 years, and showed the potential of lithium as a battery material.[28]

After the success of lithium primary batteries, development began on a lithium secondary battery. In 1978, intercalation materials capable of storing lithium ions on cathode were developed. Anode was lithium, where the ions were expected to return as plated lithium upon charging. In theory this seemed to work but lithium is a highly reactive material. The solid electrolyte interface (SEI) that forms on the surface of the lithium anode is capable of letting ions pass through, but in some cases it is irregular enough to slow and even block them. Ions that cannot pass easily enough form dendrites on the surface, which may eventually lead to short circuits. Short circuit in a cell containing material as highly reactive as lithium leads to spectacular failures – the battery may even explode. Replacing the liquid electrolyte with a polymer made the battery more stable, but due to the risks and gained bad reputation, rechargeable lithium battery never found mainstream success.[28]

Further development resulted in the Li-ion battery. The lithium of the anode was replaced with carbon capable of intercalating lithium ions, so now both the anode and cathode contained lithium only as ions instead of in metallic form. The ions would move from one storage to another upon charge and discharge, without reacting themselves. Reactive lithium metal does not normally exist in the battery and construction uses much less lithium, which brings material costs down.[23,28]

Li-ion battery is sometimes referred to as the "rocking-chair" battery. This name is due to the fact that when the battery is charged and discharged, the ions remain unchanged – they "rock" back and forth between the two poles. Both the anode and the cathode have a lattice structure where the ions fit. When the cathode material is formed, lithium is intercalated with a metal oxide, creating a lattice full of lithium-ions. When the battery is charged and the electrons move to the negative current collector, the ions travel to the carbon lattice on the anode. Discharging the battery lets the electrons flow to the positive current collector while the ions travel back to the cathode. Simplified lattices are depicted in figure 3.1. It is worth noticing that neither lattice can be completely empty of lithium, otherwise they risk collapse.

Figure 3.1 *Lithium-ion lattices*

The chemical reactions in a Li-ion battery are more straightforward than in the previous lead-acid example. When the battery is charged, the cathode is oxidized,

$$LiMO_2 \quad \rightarrow \quad Li_{1-x}MO_2 + x\,Li^+ + x\,e^-, \tag{3.4}$$
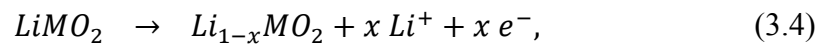
lithium molecules give up their electrons and deintercalate from the metal-oxide lattice. Metal here is not specified further, and can in fact have more than one phase as is the case with LiFePO4 (Li1-xFePO4 & FePO4), which we'll concentrate more on in chapter 5. The deintercalated Li-ions travel from the cathode through the electrolyte, separator and SEI and intercalate in to the anode lattice. Here we have a graphite anode, where in the reduction reaction

$$C + y\,Li^+ + y\,e^- \quad \rightarrow \quad Li_y C, \tag{3.5}$$

Li-ions gain electrons that have travelled to the negative current collector. These reactions are reversed upon discharge. The complete reaction in the cell is then

$$LiMO_2 + x/y\,C \quad \leftrightarrow \quad x/y\,Li_y C + Li_{1-x}MO_2. \tag{3.6}$$
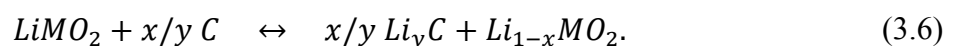
Typical values for x and y are in the range of x = 0.5 and y = 0.16, so about half of the lithium from the cathode is deintercalated. Again, important points are that lithium should only be present as ions instead of metal, and that the metal oxide lattice should never be too depleted of lithium or it may collapse.[23]

Sony unveiled the first commercial Li-ion secondary battery in 1991. These batteries use lithium cobalt oxide (LCO) as the cathode. The anode was originally coke carbon, but it has been superseded by graphitic carbon. This combination has been the most common, as LCO has good performance and safety characteristics, and is not very demanding for the manufacturing process. Cobalt, however, is an expensive raw material, so price and other desired improvements have led to the development of alternative cathode materials. A short comparison of different cathode materials is collected in Table 3.1. In the table the chemistries are compared with regards to their energy density and the average voltage versus lithium when discharged at a C-rate of 1/20 [23]. C-rate of 1 is the current which would charge the cell from 0 to 100 % of rated capacity in one hour. Voltage versus lithium correlates directly to the voltage that can be measured between the cathode and the anode.

Table 3.1 *Characteristics of Some Positive Electrode Materials[23]*

Material	Specific capacity mAh/g	Midpoint V vs. Li at C/20	Comments
$LiCoO_2$	155	3.9	Still the most common. Co is expensive.
$LiNi_{1-x-y}Mn_xCo_yO_2$ (NMC)	140-180	≈ 3.8	Safer and less expensive than $LiCoO_2$. Capacity depends on upper voltage cut-off.
$LiNi_{0.8}Co_{0.15}Al_{0.05}O_2$	200	3.73	About as safe as $LiCoO_2$, high capacity.
$LiMn_2O_4$	100-120	4.05	Inexpensive, safer than $LiCoO_2$, poor high temperature stability (but improving with R&D).
$LiFePO_4$	160	3.45	Synthesis in inert gas leads to process cost. Very safe. Low volumetric energy.
$Li[Li_{1/9}Ni_{1/3}Mn_{5/9}]O_2$	275	3.8	High specific capacity, R&D scale, low rate capability.
$LiNi_{0.5}Mn_{1.5}O_4$	130	4.6	Requires an electrolyte that is stable at a high voltage.

Li-ion batteries have become the standard in applications where size and weight of the battery are important. This includes everything from phones to passenger cars, and now even prototypes of semi-trailer trucks have been developed. Li-ion batteries are smaller and lighter than other batteries with similar power and energy, and they operate in a wide range of temperatures. The batteries can be stored for longer periods than alkaline batteries without losing capacity or charge, and they withstand more charge/discharge cycles until they fail. They can also be recharged before being completely empty without losing capacity. The single cell voltage is about 3 times higher than that of NiCd or NiMH cells, so fewer cells are needed for the same voltage. The higher voltage prevents the use of water-based electrolyte, as water starts to separate into hydrogen and oxygen in voltages over 2 V. the electrolytes for Li-ion cells usually contain dissolved lithium

salt, and unlike aqueous electrolytes, they are not acidic or corrosive. The Li-ion cell electrolytes are highly flammable, however, and the vapor pressure of these materials is high. These attributes contribute to a high risk of fire or explosion if they are handled incorrectly. The released vapor is also toxic, as the electrolytes commonly contain phosphor. Replacing liquid electrolytes with polymers makes the cells safer, but ion movement through polymers is slower than through liquid.[23]

A major disadvantage of Li-ion batteries is that they are not inherently regulated. There is no chemical process that prevents overcharge or overdischarge, and either scenario can lead to permanent degradation or even destruction. All charged Li-ion electrode materials react with the electrolyte in high temperatures, so cycling the cells above 60 degrees Celsius offers less capacity, and with enough generated heat can eventually lead to a thermal runaway event, where the heat itself increases the rate of detrimental reactions. Lithium metal is highly reactive, and while Li-ion batteries do not normally contain any metallic lithium, it can form on electrode surfaces in certain conditions. This is known as lithium plating.[10,23]

Thermal runaway occurs when new exothermic reactions are triggered by heat from previous exothermic reactions. When the cell temperature rises above 90 degrees Celsius, SEI layer will begin to decompose and release heat, exposing carbon anode to the electrolyte. Carbon reacts with the electrolyte, releasing gas and building pressure inside the cell. Separator between the anode and cathode starts to shrink, and melts at 120 degrees, and most cathode materials break down before 200 degrees. Above 200 degrees even the electrolyte breaks down and produces gas, which reacts with the oxygen released from decomposed electrodes. At this point if not earlier, it doesn't matter what a battery management system attempts to do, exothermic reactions feed each other and the battery will burn – and if gases cannot be vented fast enough, it will explode.[10,31]

Li-ion cells can be discharged in temperatures below 0 degrees Celsius, but charging is difficult. Lithium plating occurs normally during charging, but the plated lithium intercalates quickly in normal operating temperatures. Intercalation of lithium ions on the graphite anode slows down when temperature drops, and plating occurs faster than intercalation.[30] Plated lithium forms dendrites, and as the distances inside cells are small, this can quickly lead to short circuits. Overcharging in any temperature may also cause lithium plating, as well as overpotential on the cathode. Overpotential on the cathode means that the lattice has lost too many of its lithium molecules, making it unstable. Even when this does not result in a dangerous malfunction, the capacity of the cell is permanently reduced as the Li-ions cannot intercalate into collapsed parts of the lattice.[10,30]

3.4   Battery Management Systems

Li-ion batteries require management. The battery chemistry isn't self-stabilizing, there is no possibility of trickle-charging a full battery, and overcharge or overdischarge will result in reduced battery capacity and life. The battery state of charge (SOC) must be estimated from the open circuit voltage (OCV), and each different anode/cathode material combination has a different voltage curve in relation to state of charge. Charging or discharging outside of safe operating parameters may lead to failures and at worst, explosions. Imbalanced cells or insufficient measurements prevent the full utilization of battery capacity. A battery management system (BMS) is required for the safe and optimal use of a li-ion battery.[10]

The BMS is an integral part of a lithium-ion battery pack. It ensures that all the cells are balanced and the battery offers optimal performance to meet the demands placed upon it. Monitoring must be constant and if there is a danger of deviation from the safe operating conditions, BMS must react accordingly. In most cases this means limiting the current to or from the battery. The battery SOC needs to be estimated from the available data, such as the OCV and the usage statistics. Some cell chemistries make the SOC estimation from the OCV difficult, in which case Coulomb counting (measuring the passed current) can be employed. The cell manufacturer specifies the maximum and minimum operating voltages for a cell, from which full and empty states can be configured to the BMS. The amount of current that can be drawn between these states can be measured, and the measurement in Coulombs may be used to estimate remaining SOC.

The BMS needs to communicate the imposed current limits to the interconnected power electronics. In most cases these include the inverter that drives the motors and the charger that recharges the battery. In case these limitations are broken, or there is a clear fault in any part of the system, the BMS needs to disconnect the battery completely. The disconnection contactors will usually be operated directly by the BMS. The contactors must also be opened in case external damage occurs, for example when a vehicular collision is detected.

The cells of a Li-ion battery must be kept in balance. The BMS measures voltage data from each cell, and these voltages should be within tolerances compared to each other. The balancing can be done passively by dumping the excess energy through resistors, or actively by charging emptier cells with fuller ones. Often passive balancing is enough and it is much cheaper and easier to implement. If the cells are imbalanced, the full capacity cannot be used, and there is a danger of some cells being overcharged or overdischarged while others are still within safe voltages. To prevent temperature-related risks, BMS monitors the temperature of the battery. The BMS may directly actuate thermal control functions, such as heating elements or cooling pumps and fans. The battery must be cooled to prevent degradation and thermal runaway, and heated to enable charging

and discharging in cold weather. While temperatures are beyond safe limits, the charging and discharging currents must be limited or even prevented completely.[10,31]

The BMS is a safety-critical system, so a fault in the BMS may cause a dangerous failure that results in damage to material or personnel. The components of a mobile application are subjected to vibration and possible abuse due to human operator. EVs have numerous interconnected systems with their own failure probabilities, some of which the BMS also needs to react to. A BMS failure, either alone or in conjunction with other system failures, could expose the operator to risk of personal injury or death. The battery chargers and especially their physical connectors to the vehicle are subjected to physical wear and high currents, so a failure at some point is very likely. Although solid state power electronics are relatively robust, faults in inverters are not uncommon. Even normal operation could be dangerous if the operating environment is not suitable. The BMS must be able to operate in sub-zero temperatures, and it needs to be able to regulate the currents flowing to and from the battery, or if unable to do so, cut off power completely. Some of these situations are hard to detect and difficult to test for.[10,31]

Physical damage to a battery module can cause cells to rupture and electric contacts to either open or short circuit inadvertently. The BMS should detect such situations, and depending on the application, either ensure a safe shutdown or attempt limited operation. A situation like an EV ambulance in a car crash would be very challenging for a BMS to handle, as loss of life could result from either an explosive failure of the batteries or from a total loss of power.

The BMS can fail in numerous ways. For a lithium-ion battery pack, any failure can be dangerous. Overestimating the SOC might only result in reduced capacity and running out of power, but in some applications even the loss of power can be dangerous. Underestimating the SOC would result in unused capacity, but it could also result in an overcharge and all the dangers that an overcharged Li-ion cell poses. A failure to detect deviation from the safe limits regarding voltage and temperature may cause permanent damage, as described previously. The same situation could result from an erroneous current detection, if the current is higher than the BMS interprets it. Power cables are equipped with an interlock loop, which tells the battery management system that the cable is properly connected, and if this is not detected, charging or discharging could cause electric arcs and fire, or a high voltage electric shock to the user. The battery disconnection can be triggered by the detection of a failure, or by the user. If the disconnection fails for any reason, it can cause damage to the battery, equipment or people. A disconnect failure may be caused by BMS software faults, communication errors, failed actuators or a connector that's welded closed due to overcurrent.

4. DEVELOPMENT AND TESTING

Any new product needs to be verified before being released to the market. We will consider heavy equipment, such as cars and industrial and agricultural machines, that are meant to be used by consumers. The end user is rarely an expert, and most do not have, nor need to have, any idea of the inner workings of these products. For certain applications the users are not even required to have any certification or training, which increases the demands for safety considerations. The product needs to be robust, safe and functional, regardless of the use and abuse it is put to. Therefore, in addition to testing for the ability to function as planned, the developer needs to make sure that there is little to no risk to the user in case of misuse or a major malfunction.

Testing is a time consuming process. Traditional testing requires prototypes to be built and tested in various situations, often requiring the prototype to be irreversibly damaged. Some cases can be tested with only a partial product, but this still takes time and resources, as each piece needs to be manufactured, tested and recycled accordingly. Testing with prototype equipment is also not without risks, especially in fault simulations.

Modern machinery is controlled by multiple ECUs. An ECU has no physical actuators in itself, the only interfaces being electronic. Most often these interfaces operate on 5 Volt logic-level signals, even when the machinery in question may be powered by high voltage electricity. This opens up a new avenue for module testing, as the ECU itself can be interacted with using only 5 V signals. The ECU won't know the origin of the signals, as long as they tell the same story as they would, coming from the physical components. Logic-level equipment is also safer and more affordable than equipment capable of high voltages.

4.1 Hardware-In-the-Loop testing

Simulating a physical machine to test an ECU is common practice today. In a HIL test setup, component to be tested is connected to a virtual system that simulates the inputs and outputs of the actual system. Numerous tools are offered for industries, such as purpose-built programmable control units capable of emulating sensors and actuators. In the ECUs point of view it should make no difference if it is connected to the actual system or a simulation of one (Figure 4.1).
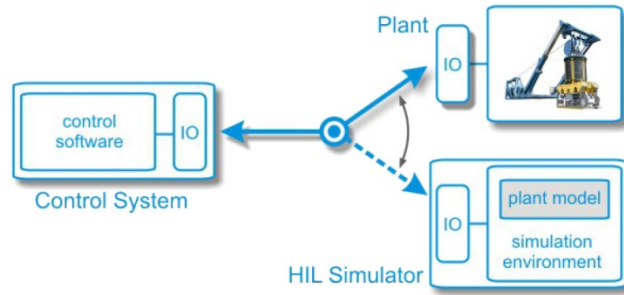
Figure 4.1 *Replacing a physical plant with an HIL simulator[32]*

A simulated system can be brought to states which would be unsafe for the actual system. Especially heavy equipment testing poses a high risk of injury to people and damage to equipment [6], while a simulated system can basically go up in flames just to see if the ECU handles the situation correctly. HIL testing can also be done simultaneously with the development process, for instance testing a control unit for a product that doesn't yet exist. This can bring the total development time down considerably, and help detect errors early. The cost of errors rises the further in the development cycle they are detected, so HIL testing can help save substantial amounts of money for the developer.[32]

HIL-setups come in different types. Some have physical actuators and can send and receive high voltage electricity, and there are also dedicated hardware setups purely for battery simulation. When the ECU is only communicated with using 5 V signals, the setup is referred to as Logic-level HIL. Using only low voltage reduces risk further, and can keep the cost of the HIL simulator down.

A simulated product, system or plant needs to react like the real thing. To accomplish a realistic simulation, the physics of the target must be known well enough to be modelled mathematically. Easily explainable example would be an electronic throttle body (Figure 4.2). When the ECU outputs a control signal, the throttle flap opens to a desired angle, and the position sensor sends back the position data. We can measure the data, and create a simulation that sends a similar response back to the ECU.

Figure 4.2 *Bosch electronic throttle body [41]*

Depending on the application, simulation can be superficial or in-depth. Superficial simulation could have a lookup table for each combination of input and output signals, or an empirical equation to calculate the right output. For an in-depth simulation, in the case of a throttle body we could measure the spring constant that pushes the flap against the control motor, and simulate the output with a physics-based spring equation, combined with the equation that transforms this into the right electrical output signal. With our simulation we could test the ECU with near-unlimited variety, with no fear of getting our fingers stuck between the flap and the housing. If we had the specifications from the manufacturer, we wouldn't even need the actual throttle body to develop a control program for it. Important benefit of HIL testing is that we can also test the ECUs ability to handle faults, in this case for instance a worn spring, broken position sensor or a sticking flap. Testing for these cases with a prototype would require for us to break the actual throttle body, or maybe separate bodies for each test.

HIL testing offers complete control over the testing procedure. The physical components are replaced with an ECU that interacts with the ECU under testing. In this case, the simulation ECU runs a program simulating a battery pack and interacts with a BMS. Testing with a virtual battery greatly reduces the risk of fire and explosions.

4.2 Planning an HIL setup

Before developing a test setup, it is important to properly specify what it is for. A perfect simulation would model its target completely, but it would also take a lot of time and effort to create such a program, and a lot of computing power to run it. Simplest simulation would present one output for one input with no variation in between, and this would be as quick to create as a table of values can be, and would require little power to run in real time.

HIL testing is useful in module and integration testing phases. Module testing concentrates on a single unit, such as one ECU, and integration testing checks that different modules work together as intended.[33] Tests may overlap, as an ECU may sometimes only be monitored and interacted with through the actual communication signals. Testing for proper responses of the module will then also test the inter-unit communication.

Testing an ECU can be done for many reasons and on many levels. Basic functionality testing would require little more than a preset output for input, to see for example that the lights turn on when a button is pressed. More complicated functionality, such as matching engine revolutions to target while the load changes, would require a mathematical model for varying load and torque, and correct sensory values fed back to the ECU. More complex the ECU, more complexity is usually required for the simulation.

An important aspect in many scenarios is the testing for faults and potentially dangerous situations. These tests can be planned with the help of original documentation for the

ECU software, which usually includes safety parameters. Not everything needs to be realistically simulated, nor would it be a productive use of working hours to do so. It is therefore important to realize just how much of the world the ECU can actually "see", and what would be the simplest way to describe the desired situation to the ECU, without compromising accuracy. For instance, it takes only a single equation to send back a rising temperature sensor value, instead of a complex model simulating a burning building.

4.3 Software development

Systematic approach is required in any larger project. One option for a systematic approach is to utilize the V-model. In a V-model, depicted in Figure 4.3, planning proceeds from the highest abstraction level to the lowest, with testing planned for each level before moving onto the next one. The testing is then done in the reverse order of abstraction. This means that the first test is done on the smallest possible subsystem in the program, followed by testing for the integration of subsystems, testing of the system functionality and so on, until we can be satisfied that the program fulfills the original purpose.[35]
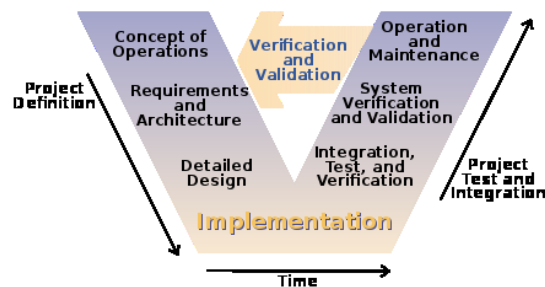
Figure 4.3 *The V-model of the Systems Engineering Process [42]*

The V-model consists of distinct stages. Design begins with a Requirement Analysis, where we determine what we want to accomplish with the software. In ordinary cases, a user requirements document would be made, but in our case the users are an integral part of the development process.

The second stage is the System Design. Here the options for achieving the required functionality are charted, and the feasibility of required functionality is addressed. System testing is planned at this point. In the Architecture Design phase, program is broken into modules and the modules' interactivity with each other is planned. Integration testing is planned, so we can verify that the modules work together as intended.

Low-level design of the program is done in Module Design phase. The separate pieces of the program are defined, and the coding of each module begins. The testing of separate modules is planned and the modules are tested as they reach a phase where this is

possible. Testing is done as early as possible, to minimize the effects of changes required. The validity and the form of the code are also verified here. Integration testing confirms that the modules function together, and when this is done, the whole system can be tested for intended functionality. The user acceptance test is the final phase, where we validate that the software meets customer expectations.

The software development V-model is not a universal template, so all the phases are not directly applicable for all projects. It does however require very few modifications for our case, so it proved a suitable guideline. In our kind of a project, with a very small project organization, the actual methodology ended up being closer to a W-model.[34] The W-model is a variation on the V-model, where the testing and coding are more integral. In a W-model, lower parts of the basic V-shape are repeated before the final rising edge. In a case like ours, when the application and development process is novel for the organization, even some architectural planning needed to be revised during coding, and program modules were tested and re-tested during development. As some hardware limitations were unknown at the beginning, some approaches had to be discarded and new ones implemented during development phase. As each new problem was resolved, the normal V-shape could resume.

5. SYSTEM DESCRIPTION AND TEST SETUP BUILD

The simulation parameters were mostly defined by the battery pack under development. The BMS to be tested had already seen extensive development and testing in actual conditions, but still required more of each. Easier testing would help reduce the development time and costs. In the early phase of our simulation design, the particulars of this BMS were largely ignored to avoid excessive biasing towards one BMS. The battery pack specifications were more relevant to the simulation.

The battery pack consists of modules, and each module consists of cells and measuring cards (Figure 5.1). The measuring card measures cell voltages and temperatures, and communicates with the BMS. A measuring card has many names, such as the cell control board, the cell-monitoring control board, the voltage temperature monitor(VTM) board or slave board.[43] We will refer to it as the measuring card from here on out. The measuring card contains one or more cell monitoring chips, and the chip manufacturer datasheet is a good basis for the simulation design. The measuring cards could also be integrated into the BMS, but our case uses a distributed system with separate cards for each module.
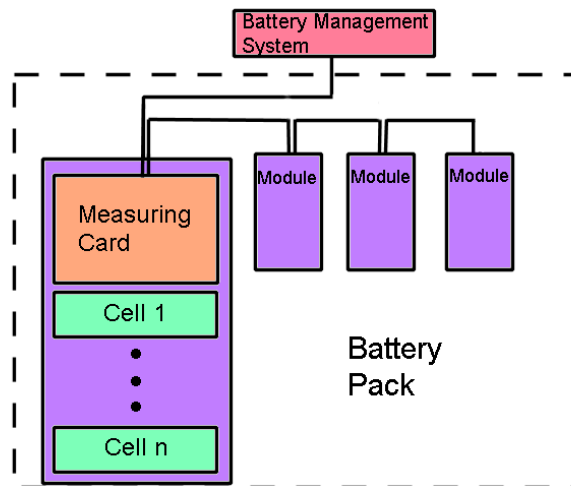
Figure 5.1 *Battery Module outline*

For the cell simulation the information supplied by the manufacturer may not be enough. In this case the manufacturer datasheet must be supplemented by performing cell measurements in different conditions. Our HIL project specifications require the

setup to be scalable for other battery types and battery management systems, so the model was developed in a modular fashion, the cells and the measuring card separately.

The measuring card functionality was given priority over cell simulation, with the accuracy of cell simulation being dependent on processing power – it would even have been enough for some cases to use constants or lookup tables instead of a real time equivalent circuit model. Realistic communication between the BMS and the simulated battery pack is what makes this testing possible, regardless of cell voltage response.

Some functionality beyond cell and monitor chip simulations is required for thorough BMS testing. The BMS has inputs for user interaction and charging/discharging currents, and outputs for physical contactors and power electronics that charge and discharge the battery. The BMS also communicates diagnostic information and instrumentation data via CAN bus. This additional functionality could be implemented if development progressed rapidly enough.

5.1   Lithium-ion cell

The cells to be simulated were of a Lithium Ferro Phosphate($LiFePO_4$) type. This refers to the cathode material, which here is based on iron, as opposed to for instance cobalt or manganese. $LiFePO_4$ cells are thermally very stable, which makes them somewhat safer than other Li-ion chemistries. They are environmentally friendly and have a high cycling performance, so they can be charged and discharged numerous times without wasting energy or losing capacity [10,24,35]. Iron is also cheap, but the manufacturing process increases the cost, as $LiFePO_4$ needs to be synthesized in inert atmosphere to prevent iron oxidization [24].

The $LiFePO_4$ anode is composed of two phases, both of which accept and release lithium ions. When only one phase is present, the cell voltage changes rapidly with the SOC, but when both phases are present, the voltage remains almost unchanged. This was one of the conclusions of Padhi et.al. in their study of $LiFePO_4$ as a potential cathode material [47]. When the cathode is undoped (without added impurities) $LiFePO_4$, as in our case, the two phases coexist for most of the capacity range, and the two-phase interface makes the voltage curve nearly flat. The voltage differential can't therefore be used as a reference except at the very ends of the capacity range, so SOC estimation is challenging.

The Li-ion cells used in laptops and electric cars are often cylindrical. Our system has prismatic cells, physically similar to the example in Figure 5.2. Prismatic cells can be stacked neatly, and because both poles are on the same end, they can be directly connected to the measuring card on one side of the stack. Li-ion cells can also come in soft pouches, but in electric vehicles they are usually either cylindrical or prismatic.

Figure 5.2*: Lithium-ion cell.[44]*

Relevant properties for the BMS, and our simulation, are the voltage and temperature of each cell. Both should change realistically depending on the SOC and the charge/discharge current. Neither value changes linearly, and Li-ion simulation has been the subject of numerous studies.[39,48,49] The methods used in the simulation of this setup are described in Chapter 5.6.2.

5.2   Measuring card

The core of the measuring card is a multicell battery monitoring chip. The chip, manufactured by Linear Technology Corporation, resides on a circuit board and communicates with the BMS via isolated serial protocol interface (IsoSPI). The battery modules to be simulated contain one measuring card and multiple cells in series, and the simulated battery pack has four of these modules connected in parallel.

The battery monitor chip used here is of a comparable type to the publicly available LTC6811. The LTC6811 monitors cell voltages directly and converts the analog voltage to a digital value, which is stored in an internal register. The analog-digital conversion is initiated by the BMS, as is the transfer of data between the BMS and the measuring cards. The LTC6811 has a total measurement error of less than 1.2 mV, and minimum measuring time for 12 cells is 290 microseconds. The chip also offers slower measuring speeds for higher accuracy. The monitor chip includes general purpose input and output ports for additional functions, such as cell temperature measurement. A cell balancing function is also included, so the chip can be used to activate balancing resistors to discharge cells, either automatically or by BMS commands. The power supply for the chip can be taken directly from the battery, and the current drain in sleep mode is 4 microamperes.[36]

The LTC6811 is offered in two versions. LTC6811-2 is made for parallel configurations, where each chip is addressed individually by the BMS, and LTC6811-1 is made for a serial configuration. This serial, or "daisy-chain" configuration, means that the BMS sees only one chip and communicates with it as if the chip had a multiple of the actual number of registers. Each series connected chip forwards the data to the next one, until the register of each is accessed. The application to be simulated has the chips in a daisy chain, as shown in Figure 5.3. Communication is timed with serial the clock signal (SCK), information is exchanged with the Master In, Slave Out (MISO) and Master

Out, Slave In (MOSI) signals, and the chip is addressed with the Slave Select (SS) signal. SS is sometimes referred to as the Chip Select (CS).
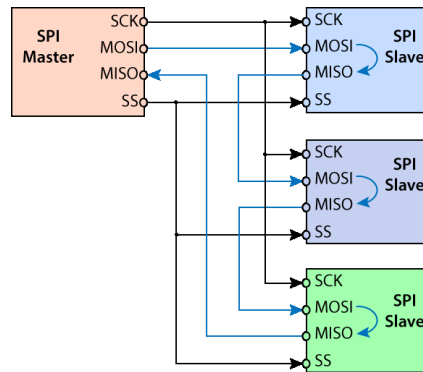
Figure 5.3: *Slaves in a daisy-chain configuration.[45]*

Serial Peripheral Interface(SPI) communication is synchronous, and it consists of one master and multiple slaves. There is no simultaneous two-way traffic, and the messages do not need a hierarchy, as the slaves only transmit upon request from the master. Linear Technology chips employ a proprietary IsoSPI interface, where the signal is isolated between transceivers. In IsoSPI the four-wire SPI signal is converted to a balanced two-wire signal and galvanically isolated with transformers or capacitors. The IsoSPI is less susceptible to distortion and noise, as both wires are subjected to the same conditions and only the potential difference between them matters for the signal. Common-mode noise is filtered out due to this construction, and communication is possible over longer distances than with regular SPI, even up to 100 meters. From a simulation standpoint this doesn't make a difference, as communication format is the same for both protocols.

5.3   Simulated variables

The simulation scope was defined by the rate of development of the project. That means simply that the faster the work progresses, the more functionality will be implemented. Planning and execution is begun with the most important functionality, which is the multicell monitor chip on the measuring card. The sell simulation is next, followed by the external sensors and actuators. Summary of the simulated signals is in table 5.1.

The workload is largest on the battery monitoring chip. The chip simulation is based on the LTC datasheet, where the handling of commands and the arrangement of memory registers is explained in detail. The cell simulation required more studying but less software development. The cells have no sensory outputs, so the only information is the voltage between the poles, and in this case external temperature measurement. The sensors and actuators are the last to be implemented, and they are also the easiest to simulate. They use either digital logic signals or logic-level analog signals. Digital signals

have a value of either 0 V or 5 V, and analog signals have a value in the range between 0 and 5 V. These signals are either conditional or linear, thus easily programmed.

Table 5.1 *List of simulation signals*

Variable	Source	Target	Connection
SPI communication	LTC simulation / BMS	BMS / LTC simulation	SPI
Cell voltage	Cell simulation	LTC simulation	Simulated functions
Cell temperature	Cell simulation	LTC simulation	Simulated functions
Contactor status	Sensor simulation	BMS	Digital Out
Current sensor	Sensor simulation	BMS	Analog Out
Battery link voltage	Sensor simulation	BMS	Analog Out
Battery charger	Actuator simulation	BMS	Digital Out
Power switch	Actuator simulation	BMS	Digital Out
Contactor control	BMS	Actuator simulation	Digital In
Inverter enabling	BMS	Actuator simulation	Digital In

5.4   Specifying the test setup

The simulation target is the battery pack except for the main BMS. Simulation should at least output and input the SPI communication, and to keep the data realistic, simulate sufficient number of cell voltages and temperatures. Simple cell models are therefore required. The communication can only succeed if the simulated measuring cards operate and communicate realistically, so after the communication functions, the monitor chip core logic has to be modelled.

The purpose of the project is to make testing fast, efficient and safe. Safety is all but guaranteed with a logic-level HIL setup, but efficiency requires the setup to be easily accessible. A graphical user interface (GUI) of some kind is required for this accessibility, and physical usability should also be considered. The rest of Chapter 5 will explain how these specifications can be met.

5.5 HIL-system Hardware

The ECU chosen for this project is the MicroAutoBox II (MABX II) by dSPACE GmbH. The MABX II (Figure 5.4) is a versatile ECU with enough computing power to run a real time simulation of a relatively complex system. A system like ours, where each cell and each measuring circuit is individually modelled, doesn't seem to stress the unit. There are some limitations, especially concerning SPI communication, but MABX II is in general a suitable tool for our application, and for HIL testing in general.[37]


Figure 5.4: *MicroAutoBox II .[46]*

Our simulated system consists of four measuring cards connected in a series to the BMS. Because the MABX II doesn't support SPI communication as it is, we need an extension ('piggyback'), a PGI SPI module [38]. The dSPACE PGI contains a field-programmable gate array (FPGA),which comes preprogrammed with the SPI communications protocol. The BMS and the MABX II are connected together in three ways (Figure 5.5). The SPI is used for the communication between the BMS and our simulated monitor chips, CAN is for the diagnostic information used to monitor the BMS operation, and the additional I/O's are for the sensors and actuators that the BMS requires. The MABX II is connected to a PC via ethernet. The PC is running the dSPACE ControlDesk program, where testing is monitored and controlled from a graphical user interface (GUI).
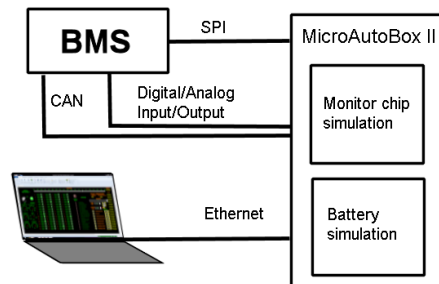
Figure 5.5: *Outline of the test setup*

The test setup needs to be easy to use. A wheeled aluminum rack was constructed to house all the setup components. A small setup like this is easy to put out of the way

when it is not required, and moved to where it is needed. The rack was designed with the CATIA V5, using the measured and modelled physical components as a reference to ensure fitment (Figure 5.6).
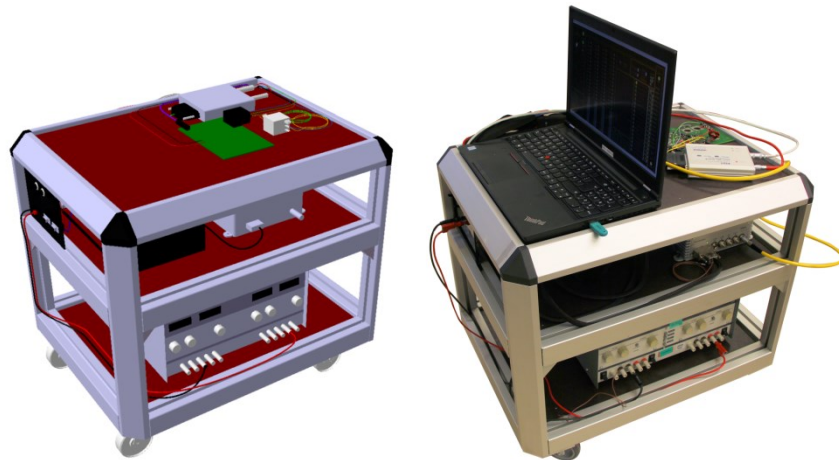

Figure 5.6 Designed and assembled HIL setup

The lower shelf houses a power supply capable of powering the BMS with a voltage of 50 V. The MABX II is located on the middle shelf with a 24 V power supply, and the SPI piggyback is on the top shelf with the BMS to be tested. The top shelf also has space for equipment, such as a laptop. One side of the rack houses an emergency stop switch, and the other side has a small physical control panel. The control panel has plugs for the BMS power, D-subs for external CAN communication, and power switches for the MABX II and the BMS separately. The wiring harness was made with dSPACE components and assorted standard parts.

5.6   HIL-system Software

The simulation software was made with Simulink by MathWorks. Programming in Simulink differs from traditional programming, as unlike in languages such as C, C++ or Python, the software is not written. The Simulink programming language is graphical, where words are replaced by different blocks which are connected together to create the desired logic. The figures in this chapter are of the developed software. Most of the figures hide the logic within subsystems, but Figure 5.9 shows the very basic blocks that create an equation which is also explained in Chapter 5.6.1. This is perhaps the best example of how the graphical programming language works, at least from a mathematical perspective. Rest of the language is intuitive for anyone who has studied logic in either mathematics or electronics. Programming in Simulink is easy to learn even with little or no previous programming experience, as I can testify. C-code may also be embedded in the Simulink model, but most operations can be achieved graphically.

dSPACE GmbH offers toolboxes for the latest Simulink releases, which makes it possible to create software that is ready to run on their hardware.[37,38] Most of the code is done using standard Simulink blocks, but the real time interface (RTI) requires dSPACE toolboxes. The toolboxes contain the protocols that the MABX II needs to interact with physical components and communicate with other ECUs. Simulink builds an executable program automatically from the graphical design. In addition to visual clarity, Simulink-created software is easy to debug, and the testing for each subsystem can be done directly in Simulink. A top level view of the program is shown in figure 5.7. The program is divided into 6 major subsystems; the CAN subsystem encompasses messages sent and received through CAN communication, and the SPI communication tasks are in the SPI_functions subsystem. The digital and analog signals are created and interpreted in the MasterPhysicalInterface subsystem, and the user inputs are collected within the UserInterface subsystem. The cell simulations are in the BatteryModel subsystem, and the monitoring chip logic is in the SlaveFunctions subsystem.

Information between the different subsystems is exchanged either via direct lines or through virtual memory blocks. Direct lines cannot be used between subsystems that are not synchronized, unless they convey static data such as constants. Any information that is dependent on a scheduled function in one block has to pass through a memory block and/or a rate transition to be reliably used in another block with a different timing.
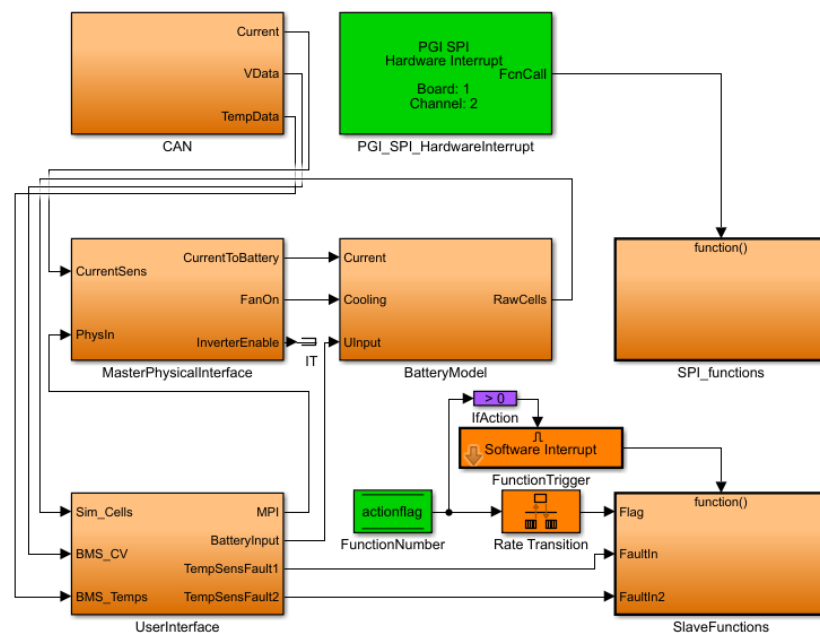
Figure 5.7: *Top level view of the program.*

The MABX II with the PGI SPI extension runs interrupt-triggered. Interrupts in software pause continuous tasks which have a lower priority, and trigger the tasks associated with that particular interrupt. The SPI-communication-related tasks (SPI_functions) are triggered by the PGI_SPI_HardwareInterrupt, and the core functions (SlaveFunc-

tions) are triggered by the software interrupt. The sell simulation (BatteryModel), CAN communication and the interface subsystems are the continuous tasks that get paused during these interrupts.
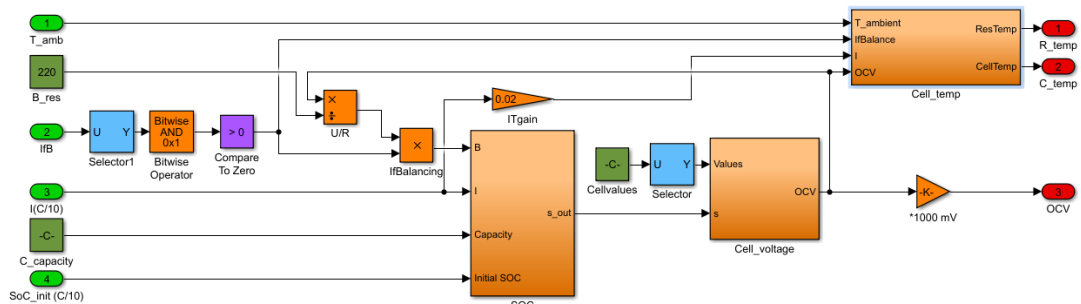
5.6.1 Cell simulation

The cell simulation was not the main focus of this project. Realistic testing requires functionality analogous with real world but real time simulation forces constraints on the complexity of the model. I chose an equation [39] that models the OCV/SOC curve of a LiFePO$_4$ cell with a good degree of accuracy, while being easy to implement in Simulink and relatively fast to process.

Parameterizing a cell is a demanding process. The cells would have to be charged and discharged at different C-rates in a wide range of temperatures to produce reliable OCV/SOC curves. Cell simulating algorithms model the internal voltage difference between cathode and anode, and OCV differs from this voltage due to the resistances and capacitances in different parts of the cell. To parameterize the resistances and the capacitances, the cells would need to undergo charge, discharge and frequency testing at different SOCs at different temperatures, as both values vary with current magnitude, SOC and temperature. Running all these tests would take up a lot of time, and the resulting simulation might be unsuitable for real time testing due to complexity. Models such as these have been studied and developed with promising results [48,49], but we decided on a simpler approach for this setup.

Our battery model simulates the cell rather superficially. While the curve is realistic, there is no voltage response delay or the effects of aging or current magnitude on cell behavior. Thermal modeling is purely empirical, based on the observed temperatures during battery testing. An empirical OCV/SOC model was justified in this case, as the BMS mostly relies on Coulomb counting for SOC calculations. The simplicity of the model prevents parametrization of the BMS for this particular cell, but the existing control algorithms can be tested with sufficient accuracy. This model was deemed sufficient for the current test specifications.

The cell simulation subsystem (Figure 5.8) in our Simulink model includes three subsystems. The SOC subsystem calculates the state of charge based on values for total capacity, initial SOC and current through the cell. The OCV subsystem calculates the voltage based on the SOC using an empirical equation with parameterized constant coefficients. The temperature subsystem simulates the thermal effects of charging, discharging and balancing of the cell, with separate temperature values for the cell and the balancing resistor.

Figure 5.8 *Cell simulation subsystem*

Open cell voltage was calculated using a generalized SOC-OCV model. The model was proposed in a paper by Zhang et.al[39], and was found suitable for this task. Exact modeling of a particular cell was not required for our test, so no real-time tuning takes place, and the SOC is a linear function. The Simulink model based on the proposed equation

$$V_{OCV} = a + b \cdot (-\ln s)^m + c \cdot s + d \cdot e^{n(s-1)} \qquad (7.1)$$

can be seen in Figure 5.9. Zhang et.al had parametrized different cell chemistries, and it was found that their parameters for LiFePO$_4$-cells matched our target cell closely. Further fine-tuning of the curve was done manually based on the manufacturer measurements. Each function in the equation corresponds to a different part of the curve, and thus to a certain chemical reaction. The coefficients for the functions (*a, b ,c, d, m* and *n*) are adjusted so that the correct function is dominant in the correct point of the curve. The coefficient *s* is the SOC, fed into this subsystem from the SOC subsystem. The logarithmic function $b \cdot (-\ln s)^m$ models the charge accumulation at low SOC, the linear function $c \cdot s$ models the redox reactions in the middle of the curve and the exponential function $d \cdot e^{n(s-1)}$ models both reactions at high SOC. The coefficient *a* affects the midpoint voltage, *b* and *m* affect the initial voltage hike at low SOC, while *c* changes the angle in the mostly linear midrange. The coefficients *d* and *n* change the angle of the curve at high SOC [39].
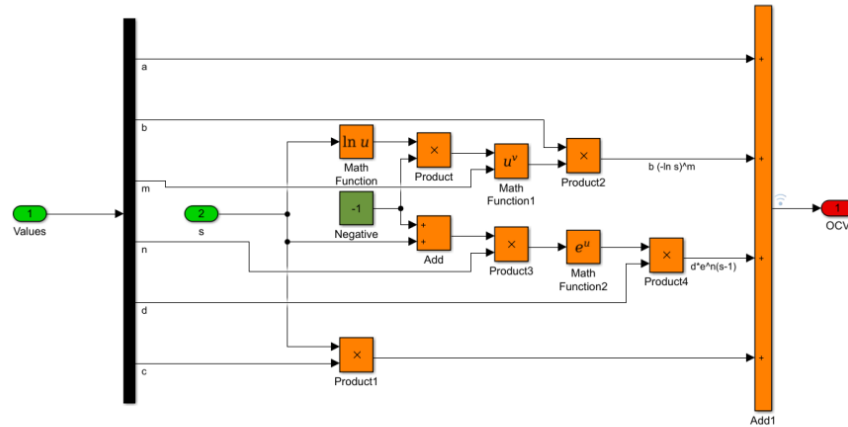
Figure 5.9 *Open-circuit voltage model*

This model is a good example of the Simulink programming language. Constant values and the *s* for SOC come from the left. In the middle these values combine with mathematical operators, and create the different functions of the equation 7.1. On the right, the functions are summed up, and the finished equation is output to the right from the model. The simulation runs tens of these cell models simultaneously so the models have to be basic, but the voltage curve is realistic, and cells can reach voltages low and high enough to trigger faults for the BMS. A realistic variation between the cells was achieved by choosing the equation coefficients for each cell with the Matlab random number generator function, based on the parameterized values. More realism could be achieved in future simulations by varying these values.

5.6.2   Measuring card simulation

The software development began with the multicell battery monitor from Linear Technology Corporation. The actual battery monitor communicates with the BMS using an isolated SPI interface. The BMS has an integrated IsoSPI transformer, and LTC offers a demonstration circuit that can be used to turn any SPI communication into an IsoSPI signal. Working with the dSPACE SPI piggyback however, we ran into so much communication issues that we decided to omit the IsoSPI-layer and use normal SPI communication between the BMS and the MABX II. This makes no functional difference, as the information and protocol are identical. Timing difference between the real and simulated systems is due to the MABX II limitations, not the lack of one isolation layer. Actual SPI communication was made possible by the dSPACE toolbox for Simulink which accompanied the PGI.

Once the communication between the MABX II and the BMS was established, the work began on proper message formats. As with most machine to machine communication, all messages are checked for errors. A packet error check (PEC) needs a cyclic redun-

dancy code (CRC), that adds a checksum to the data to be sent. This checksum is then verified by the receiver with the same CRC, to confirm that the message was transmitted correctly.

Creating software on Simulink is usually quick and straightforward. Creating a CRC graphically, however, would in this case have been both time-consuming and complicated, as the message format required the use of a large data table. The data table and the CRC were implemented as C-code instead, using Simulink's programmable function blocks. This method also ensured that the checksum is calculated in the exact same manner as it is done in the real system.

The successful communication between the MABX II and the BMS wasn't easily accomplished. Even after data was successfully transmitted, and the protocol correctly implemented, messages would not transmit accurately. Debugging required checking the signals with an oscilloscope and contacting the dSPACE support, until it became clear that their SPI piggyback is unable to keep up with the normal pace of communication. The BMS communication had to be slowed down, and the activation signal(Chip Select) had to be disabled between every transmitted data word. This is not standard SPI protocol, but the changes were possible without compromising the validity of testing.

In the SPI protocol, the communication is always started and timed by the master, in our case the BMS. The BMS needs to know exactly how many words it expects to receive, and send the timing pulses accordingly. In a real setup the slaves do not need to know the number of words beforehand, but for the MABX IIs SPI-extension, those need to be specified [38]. Further complications arose when the MABX II was unable to read more than two data words at a time. In practice this meant that there has to be a subinterrupt between each two words, which for four simulated slaves means that each communication cycle includes 9 interrupts. MABX II has enough computing power to run this software, but the SPI piggyback proved to be quite a bottleneck. dSPACE was made aware of found limitations and improvements are expected in a future release.

As the MABX IIs SPI communication can't handle communication cycles with a varying number of data words, the BMS code had to be further modified for this test. The highest number of words in one communication cycle occurs when the BMS sends or receives data to and from the measuring cards. The least number of words occurs when the BMS sends a single command to execute an action. The BMS program must send the difference as empty data words, otherwise the slaves' End-Of-Cycle (EOC) interrupt will not be triggered. Without the EOC trigger, BMS and MABX II would not be synchronized and the communication would fail after the first cycle.

The LTC monitor chip has distinct states for both the communication layer and the core. States in this case mean that the response time and the power consumption are different depending on whether the chip is in standby, sleep or active state. In some simulation

environments it would therefore make sense to simulate these states using the Simulink's Stateflow, but as the software on the MABX II is already interrupt-triggered, these different states can be considered inherent to the system. The communication layer "wakes up" on the hardware interrupt, the core "wakes up" on the software interrupt. This behavior is actually analogous to the actual circuit, with the exception of the chips wake-up times. The wake-up times are omitted for the aforementioned timing constraints in the communication. The assumption is that the BMS takes the slaves wake-up time into account, and that the actual BMS-slave communication is verified in another test with the actual components.

The BMS commands the slaves to either store data, execute actions or transmit data. In the simulation software the BMS communication triggers a hardware interrupt, and the received command triggers a software interrupt. As an example cycle, we have the BMS send a command to start the analog-digital conversion of the cell voltages (ADCV)(Figure 5.10.)
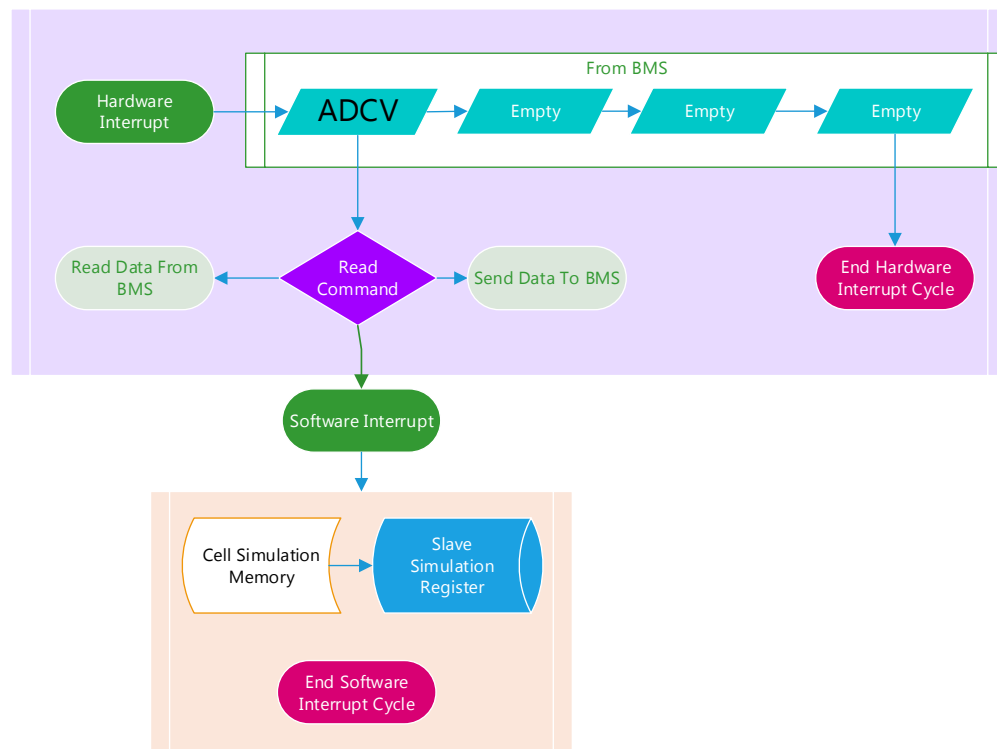


Figure 5.10: *ADCV cycle*

When the BMS initiates communication, it causes a hardware interrupt in the simulation. Every two data words cause a subinterrupt, which triggers a subsystem (Figure 5.11). First subsystem includes the Read Command block which captures the first words from the message buffer, and the subsystems for handling the command. The command is CRC-checked for packet errors and if cleared, the command is interpreted. A failed CRC check would mean that the command was corrupted and would have to be ignored. A command type is set in the simulation software to separate the commands requiring

actions from the commands requiring communication. The ADCV command is of a type requiring actions, so the "Action"-subsystem translating those commands is enabled.
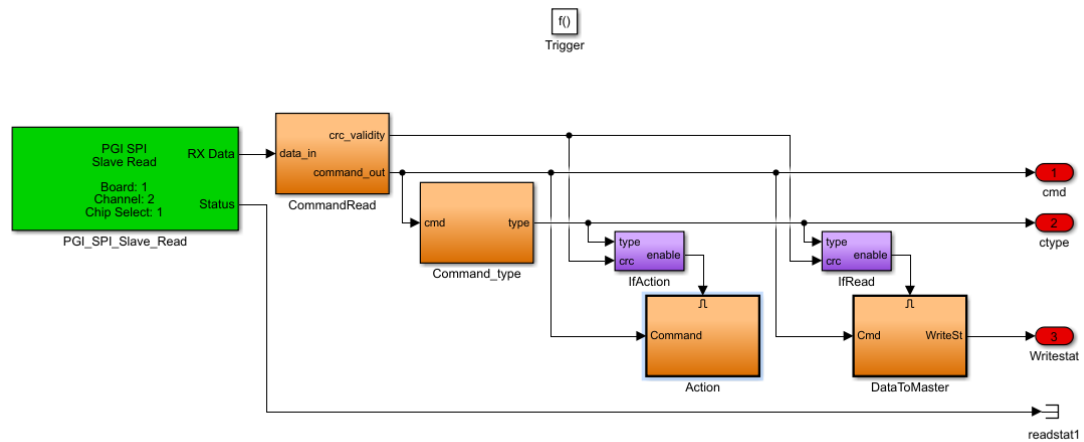
Figure 5.11: *First subinterrupt subsystem*

The commands are interpreted within the Action subsystem. A command flag for the ADCV is written in to a virtual memory block, which enables the software interrupt. The software interrupt triggers the Slave Functions –subsystem (Figure 5.12), where the core functions of the simulation are executed. The BMS continues sending data words to complete the communication cycle, but now the rest are empty, and do not cause any more triggers during this cycle.
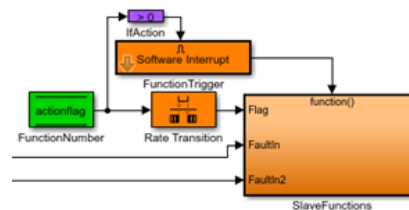
Figure 5.12: *Software interrupt for Slave Functions*

Inside the Slave Functions subsystem (Figure 5.13) the ADCV function block is enabled. The analog-digital conversion of the cell voltages is simulated simply by copying the simulated cell voltages to the simulated monitor chip registers. ADCV or any function in the Slave Functions subsystem clears the action flag, ending the core functions cycle.
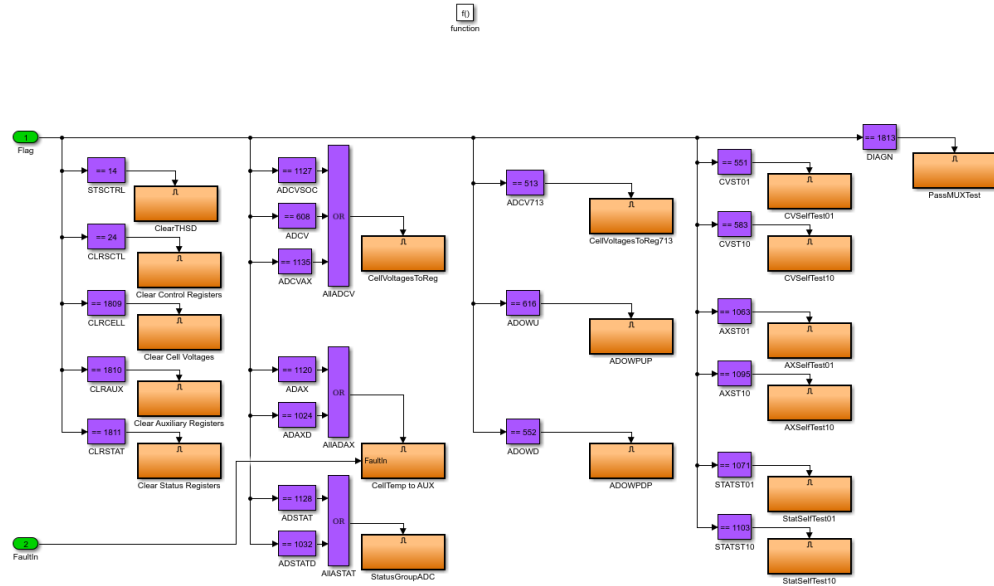
Figure 5.13*: Slave Functions subsystem*

After an ADCV command, the BMS requests the digital voltage information from the monitor chips. The command for data transfer doesn't require any core functions, so this time only communication subsystems are triggered and enabled. Outline of the cycle is depicted in Figure 5.14. The converted cell values are sent to the BMS, and after the set number of words have been transferred, the communication cycle ends.
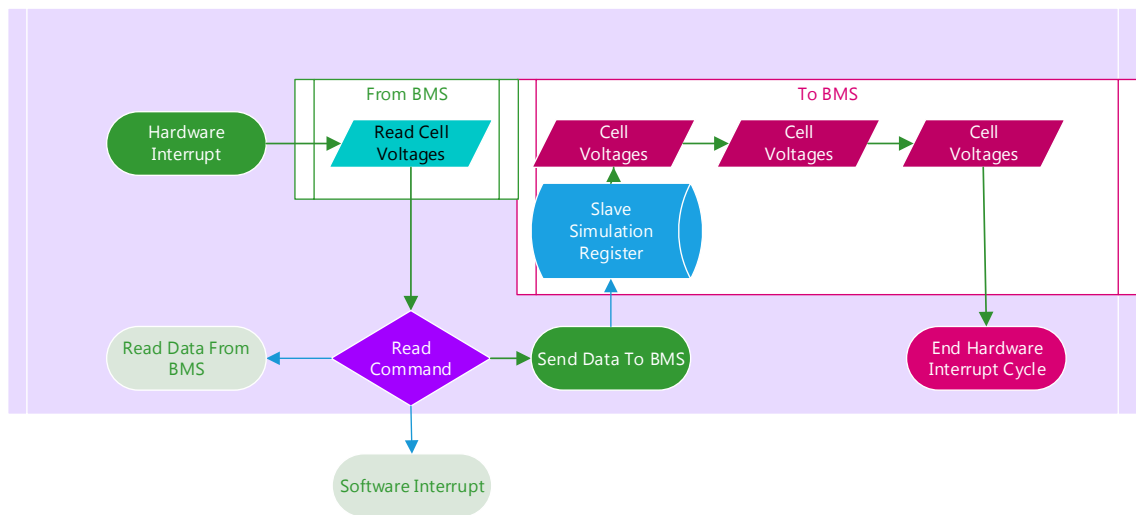

***Figure 5.14*** *Read Cell Voltages*

5.6.3 Physical interface simulation

The sensor and actuator simulation was implemented last. The minimum requirement was the ability to switch BMS on and off from the ControlDesk, but the sensors and ac-

tuators would eventually be required for some test cases. As the rest of the simulation development had progressed on schedule, the physical interface development came to include all of the planned functionality.

Outline of the MasterPhysicalInterface block is presented in Figure 5.15. The layout follows the same logic as the rest of the software, information enters from the left and leaves from the right side. The square blocks on the sides are the actual digital and analog inputs and outputs of the MABX II.
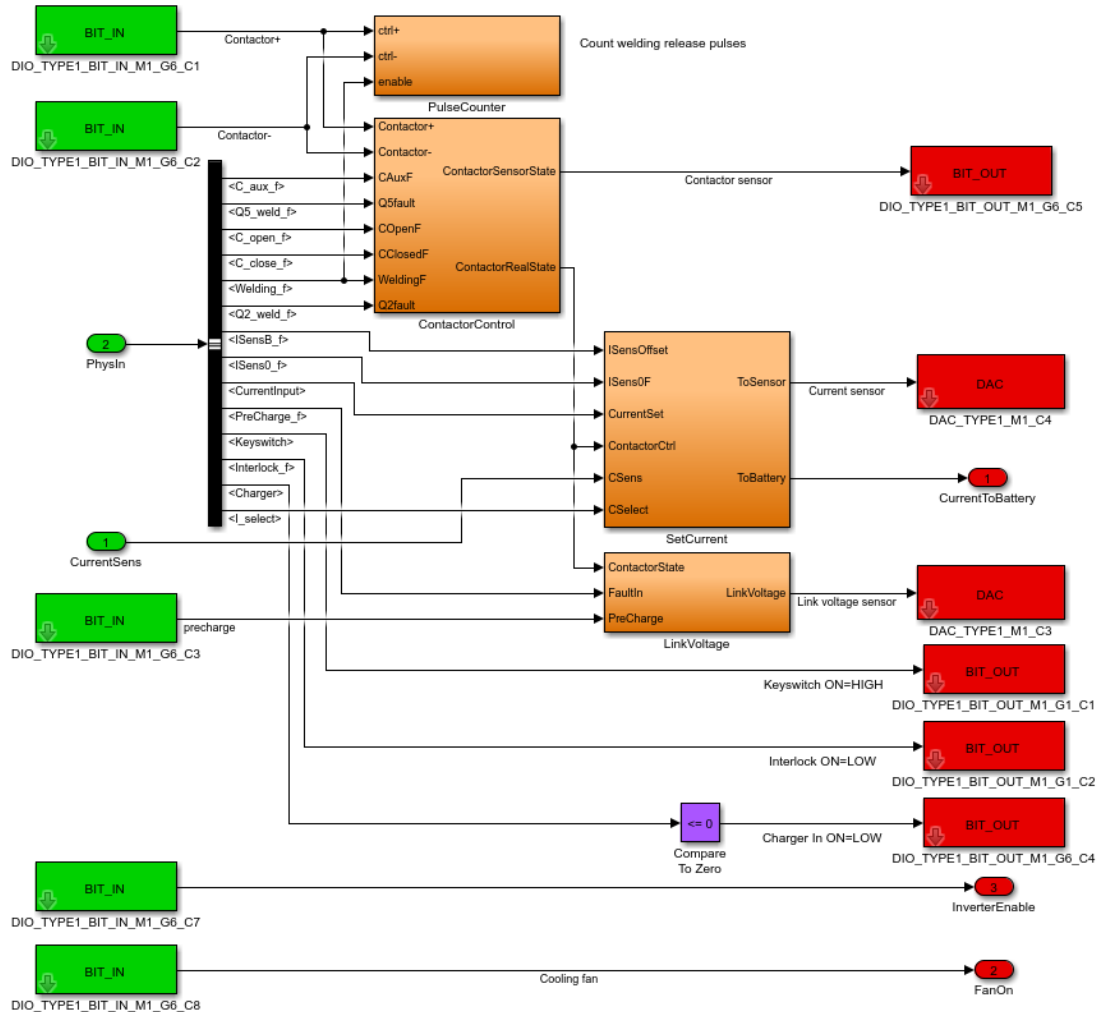
Figure 5.15 *Physical Interface subsystem*

The physical interface block includes two main subsystems and two secondary ones. The main subsystems simulate the battery pack contactor and charge/discharge current. Both subsystems have fault injection capabilities, and both output separate values for the simulation and for the BMS sensors. The values for the simulation are the "real" values, which affect the simulated battery, while the sensor values for the BMS are dependent on the fault injections. They may not match the real values if the sensors are "broken." The first secondary subsystem counts the contactor pulses for certain test results and the second, "LinkVoltage," feeds voltage information to the BMS.

5.7   Adjusting the model for fault injection

It is not enough for the HIL software to simulate the battery packs normal operation. BMSs ability to handle faults is especially important to test for safety, so we need to be able to inject faults into the simulation. A list of possible faults is for example:

- Faulty temperature sensor
- Increased internal cell resistance (faulty cell)
- Faulty contactor
- Faulty connector
- Temperatures beyond safe limits
- Cell voltages beyond safe limits

First goal of the software development was to have a simulation of a perfectly functioning battery pack. After this goal was reached, fault injection design and implementation could begin. Some faults can be injected directly to the BMS sensory inputs, such as current measurement and contactor status information. Some faults can be injected into the virtual actuators, and faults within the cells or measuring cards could be injected either into the equations or into the measured values before the data is sent to the BMS.

In practice the fault injection is a straightforward process. We can change any constant on a running program through the ControlDesk, so in the Simulink model we only need a group of constants for the user interface. Most of these constants can be simple boolean units with a true or false value, which are then connected to either logical operators or product and sum blocks. True, or 1, value in a block will change the output of some simulation process into a faulty one. Care had to be taken for the fault injection not to affect the healthy operation when not in use.

5.8   User Interface

A model running on the MABX II can be directly accessed with the dSPACE ControlDesk application [40]. The MABX II is connected to a computer via Ethernet cable, and once the program and variable descriptions have been uploaded through the ControlDesk, they can be viewed and interacted with. In our case, this means for example monitoring the measured cell voltages and the BMS status, adjusting the charge/discharge current, and injecting faults.

The ControlDesk has a user-configurable interface layout. There's a selection of instruments that can be connected to variables in the simulation. The variables were collected into their own subsystem which is depicted in figure 5.16. Each square depicts a value that can be adjusted. The four squares on the top right corner are the inputs for the battery simulation, where the user can change the temperature and the SOC, or inject faults into the cells. Below these are two fault injections for the measuring card. The middle

section has the physical interaction values, like the key switch and the fault injections for sensors and actuators. The subsystem on the left compares the simulated values with the measured ones. The results of these comparisons can be used in the ControlDesk user interface to quickly diagnose if the communication between the BMS and the simulation is successful.
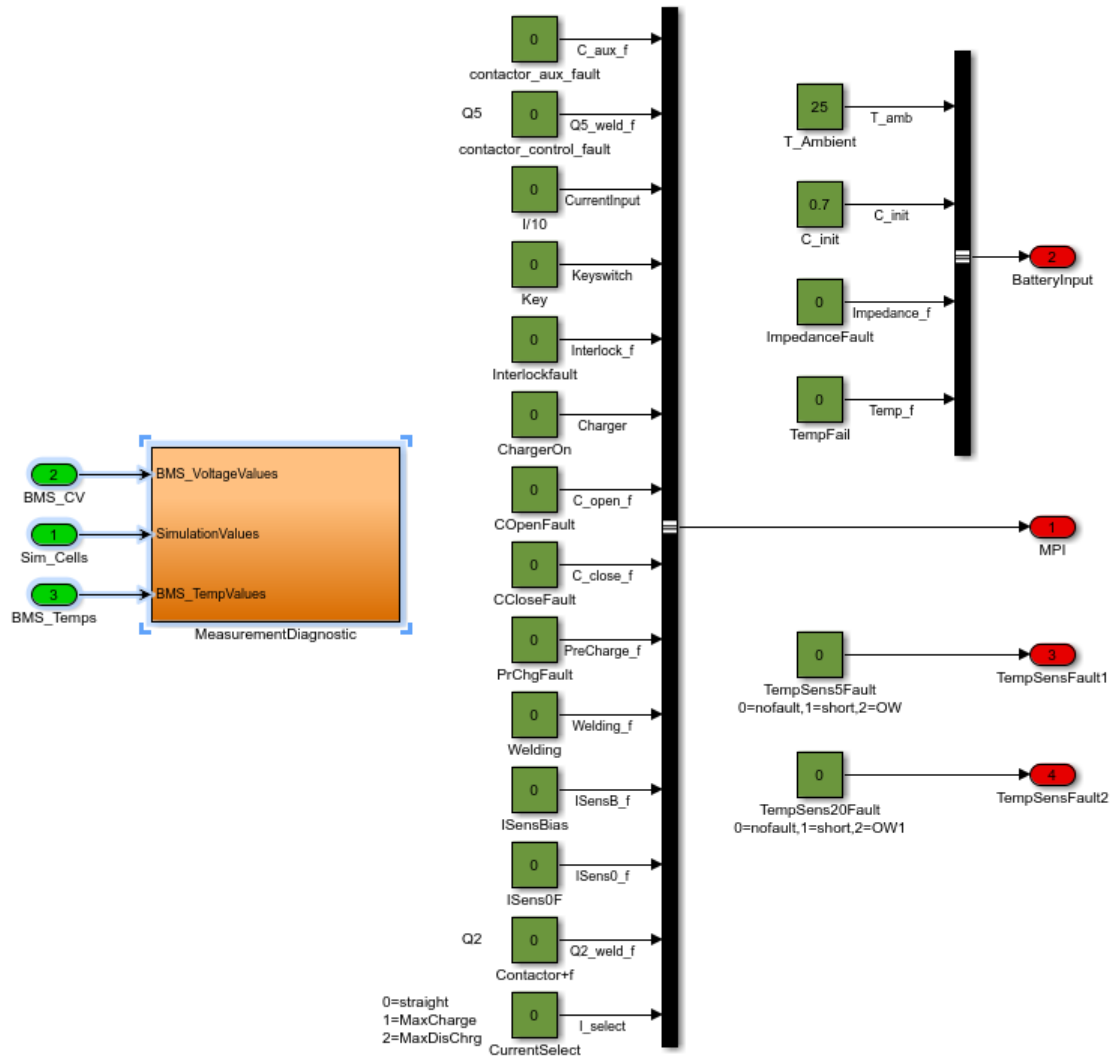

Figure 5.16 *User Interface subsystem*

Figure 5.17 has the default layout created for this project. The top left corner has a key switch and indicators for the BMS outputs for the contactor, the inverter and the precharge sequence. Below these are the indicators for BMSs communicated values represented as dials. These are the current limits for charging and discharging, estimated SOC, and the measured current, link voltage and pack voltage. Next section shows the cell voltages that the BMS has read, along with the actual state of the balancing resistors in the simulation. Current adjuster/indicator is at the top, with another, numeric input on the right. The current can also be automatically adjusted based on the limits issued by the BMS. The BMSs fault detection messages are next and these should light up when the BMS detects a problem in the battery pack.

Figure 5.17:*ControlDesk Layout*

The interface layout has a selection of fault inputs. The switches on the right change the values on the simulation which result in an erroneous output, such as a cell voltage beyond safe limits. Further on the right, the simulation values for state of charge and cell voltages for the first slave are shown. The voltages should match those on the BMS but the SOC may differ, as the virtual cells are not perfectly analogous to the real ones that the BMS is configured for. The ambient temperature and the initial state of charge can be changed from the numeric inputs on the layout. This layout makes each required BMS test possible, but the rest is a matter of taste. I like the color scheme and feel the lights make it easy to detect problems. Future layouts, however, might benefit from additional user input in terms of aesthetics and usability.

6. TESTING AND VALIDATION

Validating the HIL setup has two stages. First stage verifies that the battery pack is simulated realistically, and BMS communicates with and reacts to it as it would if it was the actual battery. Second stage validates the testing method. Here we run through a series of established and new tests for BMSs and continue until we find a problem. First 7 steps in table 6.1 are the first stage validation, and last 2 the second stage.

Table 6.1 *Project validation summary*

Aspect	Test
Communication functionality between simulation and BMS	Monitor communication at both BMS and simulation.
Message format validity	Interpret data sent by BMS, attempt to calculate checksum and interpret commands, try to send back data and read it in the BMS.
Cell simulation functionality	Plot SOC/OCV curve, compare to measured curves.
Simulation logic functionality	Read received commands and view actions executed in the simulation. Monitor communication, see if the right data is sent.
Simulation validity	Monitor BMS reaction to the simulation. Are all the values received in time, are there any faults or error messages.
User Interface functionality	Observe UI, ask for opinions from colleagues.
BMS testing functionality	Trigger faults and monitor results
BMS testing validity	Run through a list of established tests.
HIL-test concept validity	Continue testing until a bug is detected.

6.1   Test procedure

The tests were done in different stages of development. Communication functionality needed to be tested early on, before any further development could be done. Successful SPI communication with MABX II had not been attempted here before, and it was a critical component of the project. Communication testing was done with an oscilloscope connected to the SPI wiring, and by sending test data from MABX II and monitoring the received information on the BMS program. After basic testing, test data was replaced with actual data including checksums.

The cell simulation was verified on Simulink. Single cell model was tested with different currents, and the resulting voltage and temperature plots were compared to measured values. Suitability was further verified by observing the simulation turnaround time on the MABX II, to make sure that the execution didn't take too long for real time testing. The turnaround time tells how long it takes to complete the tasks of the program, and this should stay shorter than the preset time step length. Next time step begins the tasks again and these steps should occur fast enough, in our case, to simulate realistic measuring card behavior for the BMS. In practice the time step length was dictated by the SPI communication frequency.

The measuring card simulation testing consisted of attempts to execute functions and communicate correct data between the simulation and BMS. The software verification condition was the successful normal operation of the BMS, as monitored on the ControlDesk. BMS status information should indicate that the BMS is connected to a perfectly functional battery pack, with instantly updating and correct values for all the cells.

The cell measurement of the BMS could be verified by charging the simulated battery. Cell values should differ instantly, and after 3,3 V, cell balancing should begin. The balancing functionality can be verified by the indicators next to each cell voltage value in the ControlDesk. In normal operation all values are unique to each cell, and each measurement should still read correct. Individual voltage and temperature adjustment triggers are available for three separate cells to further verify functionality.

User interface should enable the simulation to be controlled and monitored. Interface should be comprehensive, yet simple enough that people unfamiliar with the software could also operate it. This functionality could be verified by running the BMS test procedures and asking others to do the same. Test functionality was verified by triggering faults and monitoring responses. Validation could be done by repeating established BMS tests and comparing the results. Tests should produce similar results and be quicker and easier to execute.

Validation of the concept was the final stage. Every complicated product has errors, especially one under development. It is highly likely that the tested BMS also has issues that either need to be repaired, modified or improved. Our HIL test setup had to be able to detect some of these to be proven useful.

6.2 BMS test examples

Testing for a specific fault handling functionality can be demonstrated with a current limit protection test. Testing is initialized by turning on the workstation connected to the HIL setup, and powering up the BMS and the MABX II. When the ControlDesk is open and connected to the setup, testing can be done form the GUI. Steps for the charging current limitation test are as follows:

- Turn BMS on, Set SOC to 50%, wait for Current limit to stabilize at 288 A
- Exceed the charge current limit by 2 A, no error should be present
- Exceed the limit by 10 A, contactor should open within 5 seconds and Overcharge Current Error should be indicated
- Turn Key Switch OFF, then ON, wait for the limit to reach 288 A again
- Exceed the limit by 100 A, contactor should open in less than 1 s, and Overcharge Current Error should be indicated.

Elapsed time for this test is less than two minutes. Without our HIL setup, test could only be done by changing the BMS limits in the source code to zero, or by charging the actual battery with a current over the limit.

The overvoltage protection is especially important for safety. In a battery pack made up of multiple cells, it is difficult to bring one cell to overvoltage while others remain within limits. Doing this is also very risky, as overcharged Li-ion cells degrade and may explode. In our test setup, however, this takes only a couple of seconds:

- Turn BMS on
- Set SOC to 75 %
- Trigger Impedance Fault
- Inspect the results, contactor should open, Cell Overvoltage error and Cell Unbalance error should be indicated.

Most of the test cases are executed in a similar fashion. Test case validation consisted of approximately 60 cases. Further test concept validation took place during and after development. BMS was under development so different software versions were tested and new testing methods discovered.

6.3 Validation test results

Project was successfully validated in each stage. Results from the performed tests are collected in table 6.2. Tests were done as previously described, and test results were as expected, though not always without further development.

Table 6.2 *Summary of validation results*

Aspect	Result
Communication functionality between simulation and BMS	Communication detected on the oscilloscope, and on both the BMS and HIL software.
Message format validity	Correct messages received, CRC check passed on both ends.
Cell simulation functionality	Voltage curve behavior matched that of the curves measured for $LiFePO_4$-cells.
Simulation logic functionality	Commands were read successfully, right actions were executed, correct values were stored and right data was sent to the BMS.
Simulation validity	BMS operated normally with no error messages, all values updated correctly.
User Interface functionality	User interface worked as intended and enabled easy test execution.
BMS testing functionality	Fault triggers resulted in expected error messages.
BMS testing validity	Established tests were fast to execute and results were similar to those performed before.
HIL-test concept validity	Some new issues were detected.

7. CONCLUSIONS AND FUTURE

Hardware-In-the-Loop –testing is a powerful tool for a developer. Testing an ECU can be made safe, fast and easy, and tests can be run simultaneously with the development process. Validation of our HIL test setup was successful, and we even detected new issues with our setup. Development progressed smoothly, and it can continue with the addition of new functionality, beyond the outline of this project.

Tools used in the creation of the setup were mostly suitable. Creating software with Simulink is easy and fast, and the logic is easily visualized. Code generation is automatic, and any errors in the software are detected and clearly communicated to the user. The dSPACE MABX II is directly compatible with Simulink and easy to use, both with software development and physical connectivity. The SPI extension didn't feel up to the standards of the rest of the dSPACE setup, and there was a surprising amount of technical limitations with the communication. For this reason I would have reservations about recommending MABX II for simulating modules that use the SPI communication. User interface creation with the dSPACE ControlDesk was simple, and creating necessary graphics for that with Corel Paint Shop Pro didn't take long. Paint Shop Pro was also used for all the original figures in the thesis. Designing the rack for the test setup with CATIA V5 was relatively straightforward, and Norcan aluminum profiles used in the construction are very easy to work with.

The simulation software is highly upgradeable. Battery models can be adjusted for different types of cell chemistries, and the equation coefficients can be adjusted in real time. Implementing transient response, thermal variation and aging for more realism can be achieved by changing the cell simulation library block, without changing anything else in the software.

Changing the functionality of the measuring card model can be done without changing the battery models. Linear Technology Corporation manufactures different chips with the same basic architecture and logic, so this model can be used to simulate those with minor modifications. Changing the number of measuring cards and battery cells is also possible.

The interaction between the HIL-simulation and the BMS can be improved further. At this point, there is no CAN communication between the BMS and the inverter and the charger included in the simulation. For proper integration testing, inverter and charger need to be simulated better in the future, with the actual communication simulation and realistic responses. Realistic load variance could also be implemented.

A BMS or any other ECU needs to tested each time when the software is modified. In this Regression Testing method each new software version goes through the same series of tens or even hundreds of tests. Even when most of them require only a few clicks, the cost in time justifies further development of the HIL setup. Automation is the next logical step and dSPACE offers an AutomationDesk in addition to the ControlDesk that was used here. The AutomationDesk is compatible with the same processes that were used to create the user interface, and can execute the performed test steps automatically. Automating the whole test series will make the regression testing an order of magnitude faster.

Safety certification is becoming important in every industry. There are numerous certificates for products where battery packs can be employed in, especially vehicular applications. Safety certificates encompass the whole development process, including testing and the tools used, so at some point certification will be necessary.

Validation results of our HIL setup were all positive. Test setup concept is good and the actual setup works as intended. Testing is easy and fast, and minor issues in the BMS were already detected during development and validation. Certain tests were only made possible with this test setup, so some BMS functionality was also verified for the first time. Initial project outline was conservative and all expectations were met, some even exceeded. Project is considered completely successful. From project start to final validation the work took about 4,5 months, and the HIL test setup is currently in active use.

REFERENCES

[1] L. Similä, I. Savolainen, Chapter 1: Introduction, In: Energy Visions 2050, 11-14, 2009, .

[2] I. Savolainen, A. Eerola, M. Hongisto, J. Honkatukia, T. Koljonen, S. Syri, N. Wessberg, Chapter 2: Global challenges and their implications for the energy sector, In: Energy Visions 2050, 15-49, 2009, .

[3] S. Helynen, M. Aho, L. Heikinheimo, J. Hämäläinen, T. Järvinen, J. Kiviluoma, T. Lindroos J., P. Lund, T. Mäkinen, H. Oravainen, E. Peltola, R. Rosenberg, S. Teir, S. Vuori, Chapter 4: Energy conversion technologies, In: Energy Visions 2050, 143-225, 2009, .

[4] S. Viinikainen, O. Aaltonen, J. Hakala, P. Lahti, A. Laitinen, J. Laurikko, J. Manninen, H. Pihala, K. Pingoud, L. Similä, E. Tetri, E. Tsupari, P. Tuomaala, P. Vastamäki, Chapter 3: Energy use, In: Energy Visions 2050, 51-141, 2009, .

[5] Nersesian, Roy. Energy for the 21st Century : A Comprehensive Guide to Conventional and Alternative Sources. Armonk: Taylor and Francis, 2010. Accessed September 11, 2017. ProQuest Ebook Central.

[6] L. Morello, F. Cavallino, G. Genta, Mechanical Engineering Series: The Motor Car : Past, Present and Future (1), Springer Netherlands, 2014.

[7] R. LeBlanc, History of the Forklift, The Palletizer, 2017, http://packagingrevolution.net/history-of-the-fork-truck/

[8] Forklift Trucks – The Backbone of the Industry, The MHEDA Journal

[9] Global EV Outlook 2017, IEA, Paris, 2017.

[10] P. Weicker, A Systems Approach to Lithium-Ion Battery Management, 1st ed. Artech House Inc, Norwood, 2014; 2013.

[11] https://www.tesla.com/tesla_theme/assets/img/models/v1.0/slideshow/Red_Bay-1440.jpg?20170907

[12] https://www.catlifttruck.com/sites/default/files/electric-landing_2.jpg

[13] https://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/T%C3%BBranor_Planet Solar_Rabat.JPG/1200px-T%C3%BBranor_PlanetSolar_Rabat.JPG

[14] http://i2.cdn.cnn.com/cnnnext/dam/assets/140606185514-efan-front-horizontal-large-gallery.jpg

[15] http://www.pngall.com/wp-content/uploads/2016/04/Satellite-PNG-File.png

[16] http://www.avanttecno.com/www/files/9414/0897/0023/Avant_760i_300x250.png

[17] http://www.plugincars.com/sites/default/files/think-blue-rear-620.jpg

[18] Competition heats up in electric vehicle race, in: The Japan News, The Yomiuri Shimbun, Tokyo, 2017.

[19] Bagotsky, Vladimir S., editor. Fundamentals of Electrochemistry [Internet]. Hoboken: John Wiley & Sons, Incorporated; 2005. [cited 2017 September 14]. Available from: ProQuest Ebook Central

[20] 1799–1999: Alessandro Volta's 'Electric Pile', in: Journal of Electroanalytical Chemistry, 1999, pp. 1-4.

[21] Gaston Planté and his invention of the lead–acid battery—The genesis of the first practical rechargeable battery, Lead–acid battery, 2010, pp. 4424-4434.

[22] G. Planté, The Storage of Electrical Energy and Researches in the Effects Created by Currents Combining Quantity with High Tension, Whittaker & Co., London, 1887, 268 p.

[23] D. Linden, T.B. Reddy, Linden's Handbook of batteries, 4th ed. McGraw-Hill, New York, 2011 .

[24] Fundamentals of Electrochemistry, John Wiley & Sons, Incorporated, Hoboken, 2005, .

[25] W. B. Carlson, Thomas Edison as a manager of R&D: the case of the alkaline storage battery, 1898-1915, IEEE Technology and Society Magazine, Vol. 7, Iss. 4, 1988, pp. 4-12.

[26] A. Green, Chapter 10 - Stationary applications. IV. the role of nickel-cadmium batteries, in: M. Broussely, G. Pistoia (ed.), Industrial Applications of Batteries, Elsevier, Amsterdam, 2007, pp. 547-571.

[27] Advances in alkaline batteries, in: Journal of Power Sources, 2004, pp. 45-52.

[28] B. Scrosati, History of lithium batteries, Journal of Solid State Electrochemistry, Vol. 15, No. 7, 2011, pp. 1623-1630.

[29] M.S. Whittingham, History, Evolution, and Future Status of Energy Storage, Proceedings of the IEEE, Vol. 100, No. Special Centennial Issue, 2012, pp. 1518-1534.

[30] S.S. Zhang, K. Xu, T.R. Jow, Study of the charging process of a LiCoO2-based Li-ion battery, Journal of Power Sources, Vol. 160, Iss. 2, 2006, pp. 1349-1354.

[31] A review on the key issues for lithium-ion battery management in electric vehicles, in: Journal of Power Sources, 2013, pp. 272-288.

[32] C. Kleijn, Introduction to Hardware-in-the-Loop Simulation, Controllab Products B.V., web page. Available (accessed 9/12/2017): http://www.hil-simulation.com/images/stories/Documents/Introduction%20to%20Hardware-in-the-Loop%20Simulation.pdf

[33] (2016). Butterfield, A., Ngondi, G.(Eds.), A Dictionary of Computer Science. : Oxford University Press. Retrieved 6 Oct. 2017, from http://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-5355.

[34] Ganesh B Regulwar, P R Deshmukh, R M Tugnayat, P M Jawandhiya, V S Gulhane, Variations in V Model for Software Development, International Journal of Advanced Research in Computer Science, Vol. 1, Iss. 2, 2010

[35] Y. Wang, P. He, H. Zhou, Olivine LiFePO4: development and future, Energy & Environmental Science, Vol. 4, Iss. 3, 2011, pp. 805-817.

[36] LTC6811-1/LTC6811-2 Multicell Battery Monitors Datasheet, LT0817 REV B ed., Linear Technology Corporation, USA, 2016, .

[37] dSPACE GmbH, MicroAutoBox II Hardware Installation and Configuration, 2012.

[38] dSPACE GmbH, PGI manual

[39] C. Zhang, J. Jiang, L. Zhang, S. Liu, L. Wang, P. Loh, A Generalized SOC-OCV Model for Lithium-Ion Batteries and the SOC Estimation for LNMCO Battery, Energies, Vol. 9, No. 11, 2016, pp. 900.

[40] dSPACE GmbH, ControlDesk manual

[41] *http://www.motorsport-systems.co.uk/products/alternators-starters-actuators/electronic-throttle-body/*

[42] *https://en.wikipedia.org/wiki/V-Model_(software_development)*

[43] J. Warner, The handbook of lithium-ion battery pack design, Elsevier, Amsterdam [u.a.], 2015, .

[44] https://www.ev-power.eu/LiFePO4-small-cells/LiFePO4-High-Power-Cell-3-2V-20Ah-Alu-case-CE.html

[45] http://dlnware.com/sites/dlnware.com/files/images/spi_daisy_chain.png

[46] *https://www.dspace.com/en/pub/home/products/hw/micautob.cfm*

[47]Padhi, A.K, Nanjundaswamy, K. S., Goodenough, J.B. Phoshpo-olivines as Positive-Elecrtode Materials for Rechargeable Lithium Batteries, Journal of The Electrochemical Society, 1997.

[48] Low Wen Yao, J.A. Aziz, Pui Yee Kong, N.R.N. Idris, Modeling of lithium-ion battery using MATLAB/simulink, IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, IEEE, pp. 1729-1734.

[49] K. Makinejad, R. Arunachala, S. Arnold, H. Ennifar, H. Zhou, A. Jossen, W. Changyun, A Lumped Electro-Thermal Model for Li-Ion Cells in Electric Vehicle Application, EVS28 International Electric Vehicle Symposium and Exhibition, KINTEX, Korea, .

[50] M. McGrath, Record surge in atmospheric CO2 seen in 2016, BBC News, 2017, http://www.bbc.com/news/science-environment-41778089, accessed on 16.11.2017