

# **POLITECNICO DI TORINO**

Master's Degree in Mechatronic Engineering



**Politecnico  
di Torino**

Master's Degree Thesis

## **Design of a battery pack model to test a Battery Management System in Hardware-In-The-Loop environment**

Supervisors

Prof. Giovanni GRIVA  
Ing. Antonio VITALE

Candidate

Andrea Gaetano NEVIERA

## ABSTRACT

With the increasingly common use of high-voltage traction systems, whether for hybrid or full-electric powertrains, the automotive industry needed useful tools to study batteries and the electronic control units (ECU) associated with them. The aim of this thesis work, carried out at Kineton S.r.l., is to test a Battery Management System (BMS) in a Hardware-In-The-Loop (HIL) environment so that its behaviour and connections with a simulated battery pack can be analysed. After the BMS was selected, the electrical schematic and datasheet were studied to create a configuration between the device under test (DUT) and the HIL simulator to identify the inputs and outputs expected from the control unit. Once the inputs and outputs were identified, a Matlab/Simulink model was designed to interface with the real-time environment of the simulator. In fact, the simulator must include a physical and mathematical model in the Simulink environment that replicates a reliable scenario so that real control units and actuators can be connected to it. Thus, starting from the Thevenin equivalent circuit of a single battery cell, a parameterizable physical model of a module with a total of twelve cells was developed using the Matlab/Simulink base library. The modelling decisions made enable testing the BMS considered with the simulated battery, regardless of the type of battery cells used, thanks to the possibility to easily change the data depending on their architecture or composition. The connection between the ECU and the Simulink model, ensured by the HIL simulator, allows, with the help of the dSpace toolchain, the study of all parameters of interest to the BMS, such as state of charge (SOC), state of health (SOH), terminal voltage, current and temperature of each cell. Hardware-In-The-Loop simulation has brought several advantages to the automotive industry, mainly because of its high repeatability, high flexibility and low cost compared to in-vehicle testing. Thanks to this simulation, the ECU can be made to pretend that it's inside the vehicle, so that the characteristics of the BMS can be deepened through real-time testing.

# SUMMARY

1	INTRODUCTION.....	5
2	STATE OF THE ART.....	6
2.1	Battery basics.....	6
2.1.1	Primary cell chemistry.....	7
2.1.2	Secondary cell chemistry.....	8
2.1.3	Fuel cell.....	9
2.1.4	Ultracapacitors.....	10
2.2	Battery design and architecture.....	12
2.2.1	Cylindrical cells.....	15
2.2.2	Prismatic cells.....	16
2.2.3	Pouch cells.....	17
2.3	Battery types used in EVs.....	21
2.3.1	Lead Acid.....	22
2.3.2	Nickel Cadmium.....	23
2.3.3	Nickel Metal Hydride.....	23
2.3.4	ZEBRA Battery.....	24
2.3.5	Lithium Ions.....	24
3	HARDWARE IN THE LOOP (HIL) APPLICATION.....	28
3.1	Real time simulation bases.....	28
3.1.1	HIL simulation purpose.....	29
3.1.2	Sensors and Actuators.....	30
3.2	dSpace SCALEXIO System.....	33
3.2.1	SCALEXIO Processing Unit.....	35
3.2.2	Power Supply Unit.....	36
3.2.3	DS2680 I/O Unit.....	36
3.3	ECU-Simulator connection.....	45
3.3.1	Battery Management System (BMS).....	45
4	BATTERY DESIGN ON SIMULINK.....	49
4.1	Electrical model.....	49
4.1.1	Common equivalent circuit models.....	50
4.1.2	Thevenin equivalent circuit on Simulink.....	52

4.2	State of Charge (SOC) computation.....	64
4.2.1	SOC estimation with Coulomb counting method.....	66
4.2.2	SOC estimation with Voltage method.....	69
4.2.3	SOC estimation with Extended Kalman Filter.....	70
4.3	State of Health (SOH) computation.....	75
4.3.1	SOH estimation with maximum capacity.....	77
4.3.2	SOH estimation with energy loss.....	78
4.3.3	SOH estimation with internal resistance.....	79
4.4	Thermal model.....	80
4.4.1	Temperature variation within a single battery cell.....	82
4.4.2	Temperature variation between multiple cells.....	85
5	BMS CONFIGURATION ON HIL.....	88
5.1	Specifications of the ECU under test.....	88
5.1.1	TLE9015DQU transceiver board.....	89
5.1.2	TLE9012DQU cell management circuit.....	90
5.1.3	Cell Management Unit.....	91
5.1.4	Power management units.....	97
5.1.5	Temperature measurement unit.....	98
5.1.6	Communication interface.....	100
5.2	Configuration between ECU and simulator.....	102
5.2.1	Wiring Harness.....	102
5.3	dSpace Toolchain.....	107
5.3.1	ConfigurationDesk.....	108
5.3.2	ConfigurationDesk interface.....	110
5.3.3	Channels configuration on ConfigurationDesk.....	115
5.3.4	ControlDesk.....	119
5.3.5	ControlDesk interface.....	120
5.4	Final project.....	123
5.4.1	Core 1 – Battery module model.....	124
5.4.2	Core 2 – Soft ECU.....	127
5.4.3	New application in ConfigurationDesk.....	136
6	CONCLUSIONS AND FUTURE DEVELOPMENTS.....	140
7	BIBLIOGRAPHY.....	150
8	LIST OF FIGURES.....	153

# 1. INTRODUCTION

Hardware-In-The-Loop (HIL) simulation enables real-time testing of a variety of components before they reach the final stages of their production. A HIL simulator then connects the controller developed in Simulink to the selected DUT and creates a model that can fool the DUT into feeling like it is in the vehicle. This type of testing is mainly used for validating electronic control units. For the simulator, everything that represents an output for the ECU, such as the actuators, becomes an input and, conversely, its inputs, the sensors, become an output for the simulator and are provided through the electrical channels integrated in it. Taking advantage of the potential of this technology, it is therefore possible to simulate high-voltage devices such as battery packs integrated in hybrid or fully electric vehicles and observe their behaviour with the real ECUs. Based on these assumptions and the in-depth study of a battery management system (BMS), the objective of this work is to develop, in the Simulink environment, the model of a battery pack that can be related to a real BMS through a dSpace simulator. Starting from the study of the composition of a battery cell and its operation, a Simulink model has been developed to reproduce the behaviour of an entire battery pack, capable of interfacing with a real ECU through the simulator channels to provide the required information. The paper follows the logical process that led to the final results, starting from the general study of batteries and of the HIL simulator used, through the creation of the Simulink model, the study of the chosen control unit and the configuration of the simulator through the dSpace toolchain. By replacing a real plant with a digitized one, HIL simulation significantly improves production quality, reduces costs and increases safety compared to tests performed directly on the vehicle. Indeed, the simulator channels can be used to generate different types of signals, with the advantage that certain tests can be performed over and over again without having to use real components such as battery cells.

## 2. STATE OF THE ART

In electric vehicle (EV) and hybrid electric vehicle (HEV), a battery is an electrochemical device used to storage the chemical energy and convert it into electrical energy.

### 2.1 Battery basics

First, we need to consider the difference between a battery and a cell. A cell is the basic unit of a battery. A simple cell consists of two electrodes located in a container that holds the electrolyte [1]. The electrodes are a fundamental part of the battery cell, conductors that allow the exchange of current within the electrolyte. There are two types of electrodes, the cathode and the anode. The cathode is the positive electrode, and the anode is the negative electrode. The cathode must have a high redox potential and a high specific capacity. On the contrary, the anode must have a low redox potential, while the specific capacity must be high. Both electrodes are immersed in the electrolyte, a solution that allows the movement of positive ions (cations) and negative ions (anions). It allows a path for electron flow and can be a salt, an acid, or an alkaline solution [1]. For example, the dry cell has a paste as electrolyte, while the galvanic cell has a liquid form. Thus, the electrolyte can be liquid, solid, a polymer, or a hybrid. The cell also has a separator through which ions can flow and which prevents a short circuit between the electrodes [2].

There are two main types of battery cells: Primary and Secondary cells. In the primary cell, the chemical reaction occurs only once. In the secondary cell, the electrodes and electrolyte change due to the chemical reaction that occurs during this process when the cell supplies current. In contrast to the primary cell, the chemical reaction in the secondary cell is reversible, i.e. this cell can

be recharged by electric current [3]. These two cells are not the only electrochemical storage that exist. There are also fuel cells and ultracapacitors.

### 2.1.1 Primary cell chemistry

Primary cells are batteries that can only be used once and cannot be recharged. A basic primary cell is the dry cell, a zinc-carbon battery in which the zinc can be used as both the container and the negative electrode. The positive electrode is a rod of carbon surrounded by a paste of manganese oxide, zinc chloride, ammonium chloride, carbon powder, and a small amount of water. A disadvantage of the zinc-carbon battery, aside from the fact that it cannot be recharged, is that the zinc container oxidizes and its contents eventually leak, so this type of battery should not be left in an electrical device for extended periods.

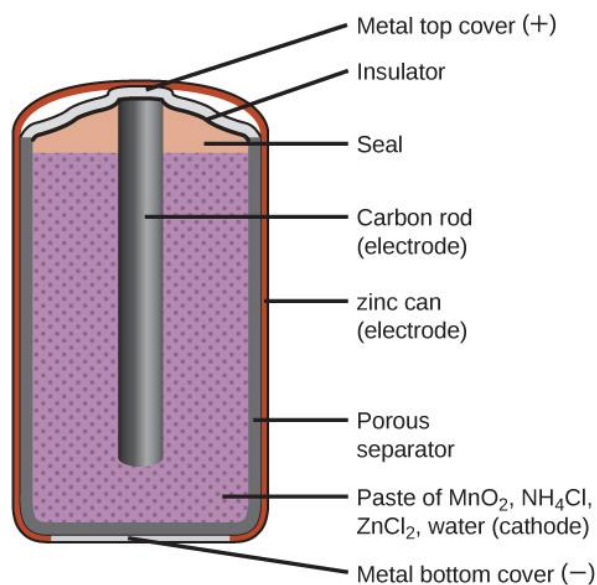
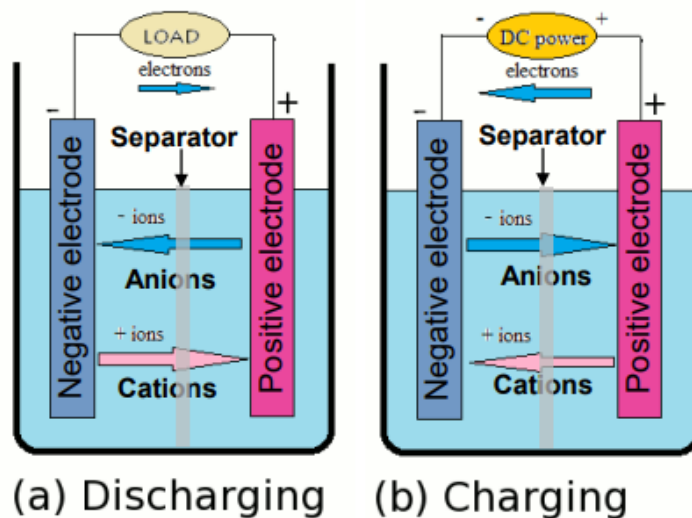


Fig. 1. Cross section of a zinc-carbon dry cell (<https://chem.libretexts.org>)

Another type of primary cell is the alkaline battery. An alkaline battery can provide about three to five times as much energy as a zinc-carbon cell of the same size. Some of them are rechargeable, but most alkaline batteries are not. Attempting to recharge a non-rechargeable alkaline battery will often cause the battery to rupture and leak potassium hydroxide electrolyte. Leakage of the potassium hydroxide electrolyte also occurs when the battery is used for an extended period of time [4].

### 2.1.2 Secondary cell chemistry

Secondary cells are often used in electric and hybrid vehicles because of their rechargeability. There are two different modes for this type of cell: discharge mode and charge mode.



*Fig. 2. Discharge and charge mode of a secondary cell (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022)*

A battery cell is in discharge mode when it is connected to a load that allows current to flow from the cathode to the anode while electrons flow from the



anode to the cathode. This behaviour of the electrons creates a charge imbalance that is balanced by the ions in the cell. Anions flow to the anode to perform an oxidation reaction and donate electrons to the anode. Meanwhile, cations flow to the cathode to perform a reduction reaction and allow the cathode to accept electrons. In charge mode, the current flow is reversed by an external voltage source and the chemical reaction is reversed [2].

### 2.1.3 Fuel cell

The fuel cell, unlike the other types of electrochemical storage, is an energy source that uses a fuel and oxygen from the air. As long as they are powered by fuel, they provide heat and electricity. Like the other cells mentioned above, they have two electrodes, an anode and a cathode, immersed in an electrolyte. This type of cell often uses hydrogen to feed the anode, while the cathode is powered by air. The fuel cell is similar to a battery only when hydrogen and oxygen are continuously supplied.

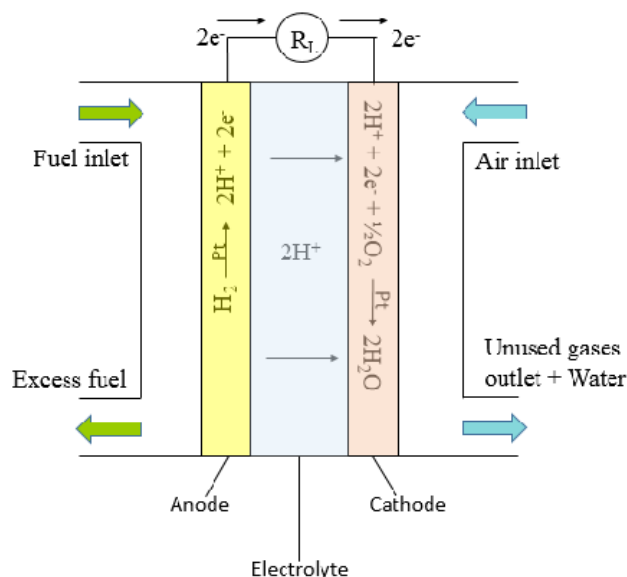


Fig. 3. Schematic illustration of a hydrogen fuel cell (Slaughter, Gymama, *Enzymatic Glucose Biofuel Cell and its Application*, Baltimore, 2015)

The hydrogen and oxygen cause a reaction at the electrodes, producing electrons, water and heat. The process is exactly the opposite of electrolysis; when water is split into ions, electricity must be supplied, but instead the reverse process produces electricity and water [2].

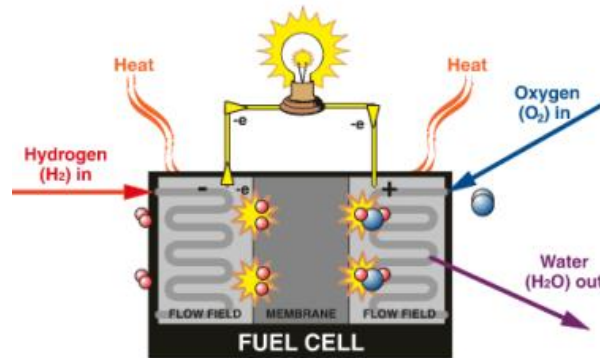


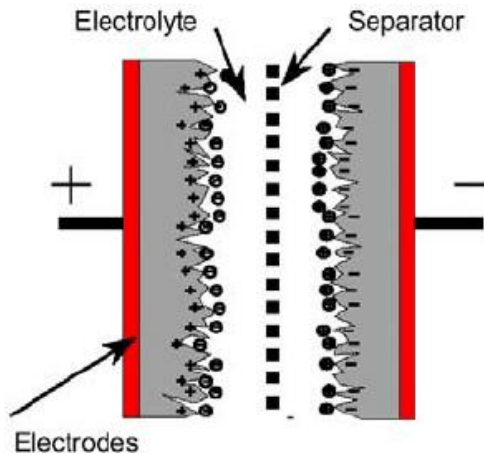
Fig. 4. Electrons, water and heat production in a fuel cell (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Turin, 2022)

There are different types of fuel cells. Although the basic processes are the same, different types have been developed to use them for different applications as the fuel and electrolytes vary.

#### 2.1.4 Ultracapacitors

Just like a battery, an ultracapacitor is a device used to store electrical energy. An ultracapacitor has two electrodes inside, one positive and one negative, separated by a dielectric insulator immersed in an electrolyte solution. The separator is non-conductive, allowing the positive and negative ions to move to the interfaces between the electrolyte and the two electrodes. In effect, ultracapacitors store energy electrostatically like normal capacitors, rather than chemically like batteries. Unlike batteries, where the internal materials change

chemically during charging and discharging due to redox reactions, the materials of supercapacitors do not change.



*Fig. 5. Schematic view of an ultracapacitor (Wasterlain, Sébastien, Hybrid power source with batteries and supercapacitor for vehicle applications, Paris, 2006)*

The separator supplies a higher stored energy density than a normal capacitor. Nevertheless, an ultracapacitor stores less energy than a battery for the same size, but releases it faster regardless of chemical reactions [5]. When individual capacitors are connected in series, a high voltage ultracapacitor pack is formed. When ultracapacitors are connected in series, voltage equalization is required (active or passive).



*Fig. 6. Example of an ultracapacitor pack (<https://maxwell.com>)*

Ultracapacitors have high specific power, which allows them to withstand very high charge/discharge currents over short periods of time. They also have low degradation values over thousands of charge/discharge cycles. The only drawback is the low specific energy.

They can be combined with batteries to provide a high energy density, high power and high density solution. Ultracapacitor stacks are usually integrated with a DC-DC converter. Its voltage is lower compared to the battery voltage [2].

## **2.2 Battery design and architecture**

Before delving into the design of the various types of batteries most commonly used in automotive applications, it is important to understand how they are defined, from the individual unit to the entire battery used for each vehicle. When it comes to automotive batteries, we need to distinguish between the following definitions:

- Battery cell: it is the single unit that stores energy. It can have different shapes and sizes, but the most common ones used for vehicles are cylindrical, prismatic and pouch.
- Battery module: The individual battery cells are grouped together in modules. The cells in the modules are connected in series and parallel with each other.
- Battery pack: The combination of modules, on the other hand, forms the battery pack. The series and parallel connections between the cells enable the entire battery pack to achieve the desired voltage and energy capacity. It is the final system that supplies power to the vehicle. It is usually equipped with a BMS (battery management system) and a

cooling system in addition to the modules. On average, a battery pack reaches a voltage in the hundreds (200-800 V).

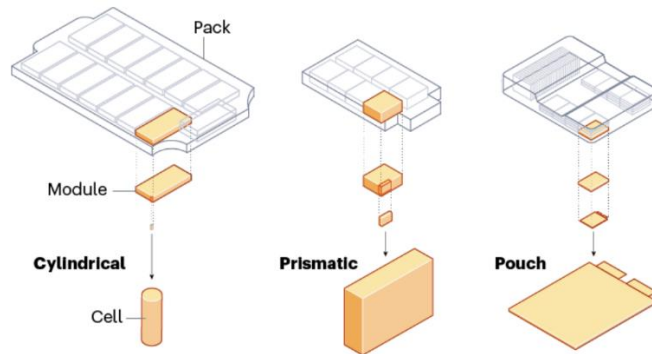


Fig. 7. Illustration of how battery cells are arranged into modules and then assembled into battery packs (<https://www.evenergi.com/electric-battery-management>)

The battery cells, as already mentioned, are designed in different shapes and sizes. In fact, depending on the user's needs, these are selected since they influence the entire design of the battery pack.

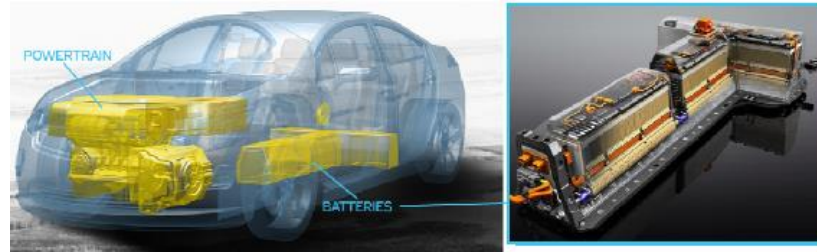
Battery packs can be installed in different ways within EVs. Among the main ones that are used are:

- Floor integration: the battery pack is located below the floor, which offers advantages in terms of cooling and packaging. It's one of the most widely used configurations on the market [6].



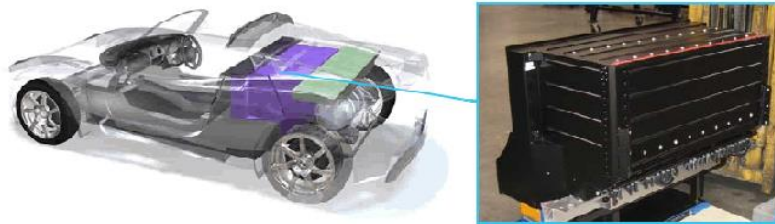
Fig. 8. Floor integration configuration in the Bolloré Bluecar (Wisch, Marcus, Crash compatibility and battery safety, EVERSAFE, 2014)

- Tunnel mounting: in this case the battery pack is installed below the passenger seats in a T-shaped tunnel [6].



*Fig. 9. Tunnel mounting configuration in the Chevrolet Volt (Wisch, Marcus, Crash compatibility and battery safety, EVERSAFE, 2014)*

- Rear mounting: this configuration places the battery pack in the rear of the vehicle. Although space is gained in the cabin, the battery pack is vulnerable in the event of a collision in the rear of the car [6].



*Fig. 10. Rear mounting configuration in the Tesla Roadster (Wisch, Marcus, Crash compatibility and battery safety, EVERSAFE, 2014)*

- Platform: is a configuration that guarantees more flexibility and space in the installation of the battery pack. Unlike floor integration the battery is not removable, so the platform should be considered as the floor itself and not integrated to it [6].



Fig. 11. Platform configuration in the Tesla model S (Wisch, Marcus, Crash compatibility and battery safety, EVERSAFE, 2014)

### 2.2.1 Cylindrical cells

The cylindrical cell is produced in automated processes by numerous manufacturers. This makes it easy to find several cells that are suitable for the final product and have the same or similar dimensions and performance as the one the car manufacturer wants to use.

The interior of the cell consists of an anode, a cathode and a separator arranged in a cylindrical container. These components are in the form of sheets that are connected to each other and rolled up in a spiral. Normally, these cells have the negative pole at the bottom and the positive pole at the top, but they can also be designed so that both poles are at the top.

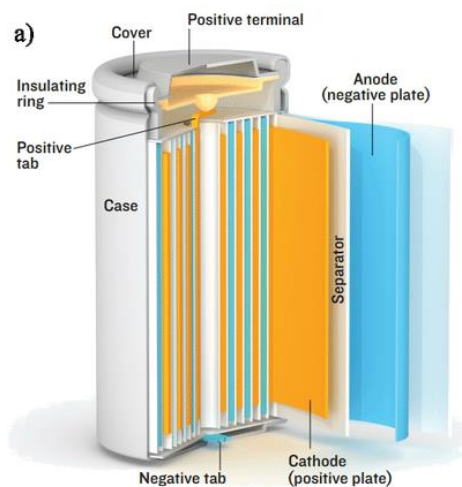
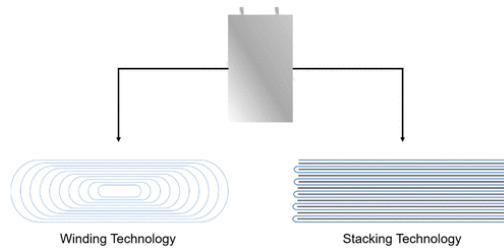


Fig. 12. Composition of a cylindrical cell (Lebel, Félix-Antoine, Méthode de dimensionnement et modélisation de batteries lithium-ion, Sherbrooke, 2017)

Unfortunately, their shape entails a low packing density, so it is not possible to fully utilize the available space inside the battery pack [7].

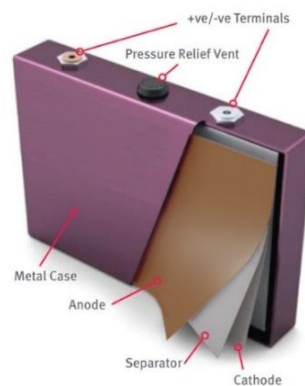
### 2.2.2 Prismatic cells

In prismatic cells, similar to cylindrical ones, the anode, cathode and separator consist of large sheets. These are either rolled up or assembled with layer stacking to be then placed inside a cubic plastic or metal container.



*Fig. 13. Rolled up and stacked layers in prismatic cells (Li, Jianlin, From Materials to Cell: State-of-the-Art and Prospective Technologies for Lithium-Ion Battery Electrode Processing, Oak Ridge, 2021)*

Rolled prismatic cells, unlike stacked cells, store more energy and have a longer life span. Instead, those stacked are able to release more energy [8].



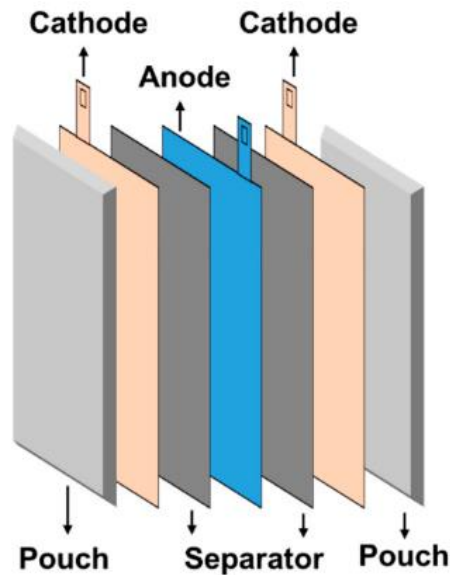
*Fig. 14. Example of prismatic cell structure (Zschornak, Matthias, Fundamental principles of battery design, Freiberg, 2017)*



This type of cell is often used inside EVs, being suitable for applications that require large amounts of energy. The packaging, thanks to their layers, is superior to that of cylindrical cells.

### 2.2.3 Pouch cells

In pouch cells, all battery components are enclosed in an aluminium-coated plastic film from which only two terminals protrude. The two terminals, also called tabs, are highly conductive and, as in the cells already seen, are differentiated into a positive and a negative pole. The electrical energy generated by the pouch cell is taken from them.



*Fig. 15. Structure of a pouch cell (Yoo, Sunggoo, Analysis of Pouch Performance to Ensure Impact Safety of Lithium-Ion Battery, Gunsan, 2019)*

The components, already seen in the other cell types, are arranged in repeated stacks of multiple layers inside the pouch. Each stack contains a liquid electrolyte, three layers of solid plates, and two layers of active material. The active material is applied to each current collector and causes the electrode to

react. Also, as mentioned earlier, there is the cathode and anode from which the ions travel through the electrolyte.

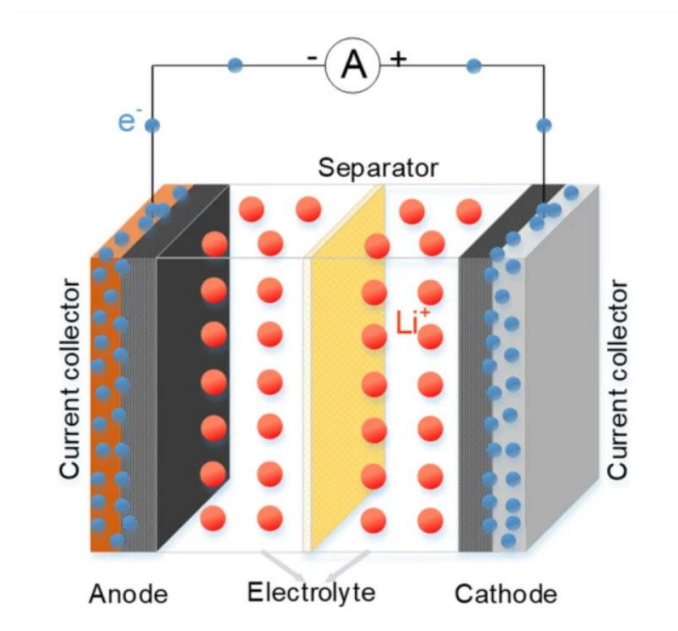


Fig. 16. Schematic view of a pouch cell (Zhu, Pengcheng, A review of current collectors for lithium-ion batteries, Coventry, 2020)

The electrode sheets contain various components such as a binding agent or solvent, an active material, and a conductive material. The first allows you to incorporate all the components and create a homogeneous paste. The second component, in addition to what has already been said, generates electrical energy during discharge. On the other hand, the conductive material increases the electrical conductivity. To combine these elements and provide batteries with excellent quality and safety, manufacturers follow precise steps. Although the basic procedures do not change, they may vary from manufacturer to manufacturer. The above ingredients (binder/solvent, active and conductive material) are mixed under vacuum to prevent bubbles from forming in the mixture and to avoid contaminating the electrolyte. When the paste is homogeneous, it is transported to the coating machine, which compresses it to produce the current collectors in the form of a highly conductive metal sheet.

The coating machine removes the excess mixture and dries the paste obtained so that it can be compressed with a cylinder press for high pressures. This process allows to obtain the right porosity and the best thickness to allow the electrons to flow better inside the cell, improving the performance.

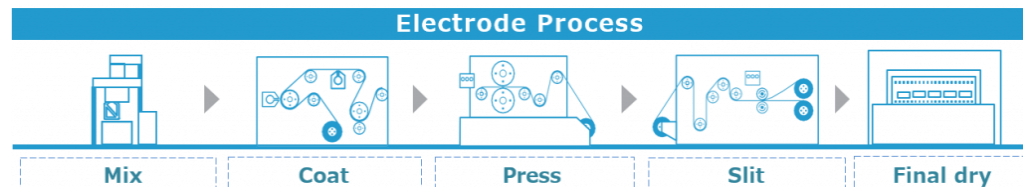


Fig. 17. Electrode production for a pouch cell (<https://djausa.com/djausa/automotive-assembly-equipment/lib/>)

After the foil electrodes are made, the current collectors are connected by a metal strip and placed in a stacking machine with the separator in the middle. Once the stack is ready, it is placed in the bag, the sides of which are sealed by heat so that the liquid electrolyte can be poured in later without leaking. After vacuum sealing again, the cell pouch is ready and the terminals can be wrapped with the same metal strip used before. The final step to complete the product is called formation. This is when the cell is charged for the first time so that it is ready for use.

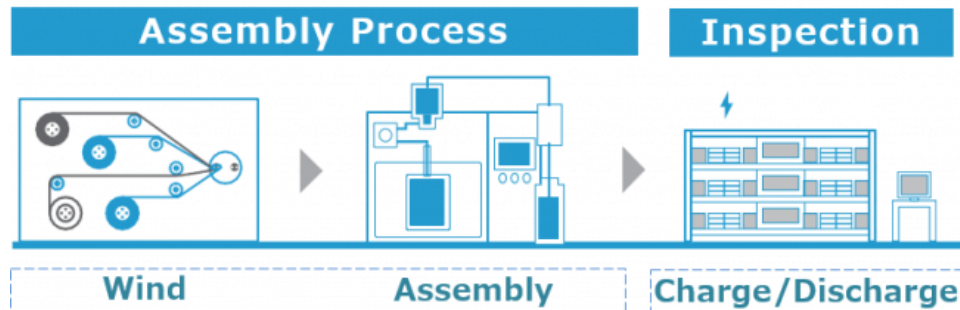
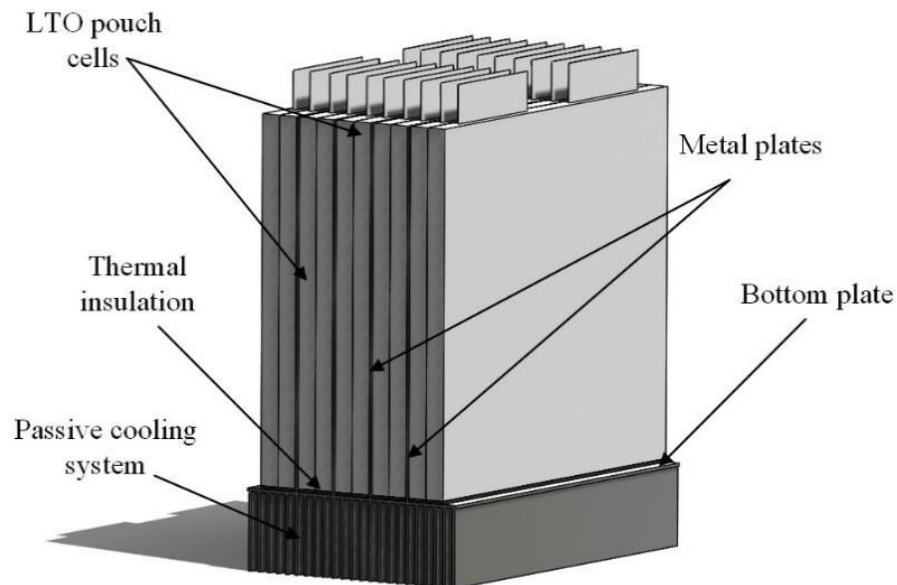


Fig. 18. Final steps to produce a pouch cell (<https://djausa.com/djausa/automotive-assembly-equipment/lib/>)

Of all existing cell types, pouch cells have the highest packaging efficiency, achieving values between 90 and 95 percent. These cells offer lightweight, flexible and simple solutions for battery pack design, but high temperatures and humidity can shorten their life. In addition, cell volume can increase by up to 10 percent due to a phenomenon called swelling, which produces gases during charge and discharge cycles. The more cells affected by this phenomenon, the greater the pressure that can damage the cells. To avoid this, an opposite pressure can also be applied to prevent the cells from becoming too large. On the other hand, as regards high temperatures, unlike cylindrical and prismatic cells from which heat can be extracted from the sides of the battery pack, an aluminium plate or a cooling plate must be inserted between each pouch cell, so that the heat can be dissipated [9].



*Fig. 19. Battery module consisting of pouch cells and metal plates (Murashko, Kirill, Thermal modelling of commercial lithium-ion batteries, Kuopio, 2016)*

## 2.3 Battery types used in EVs

Specifications	Lead Acid	NiCd	NiMH	ZEBRA	Lithium Polymer	Lithium Ions
Specific energy density (Wh/kg)	30-50	45 - 80	60 - 120	90 - 120	100 - 160	90 - 190
Internal resistance (mΩ)	<100	100 - 200	200 - 300	180	2 - 6 per cell	25 - 75 per cell
Cycle life	200-300	1000	300 - 500	400	>1000	500 - 2000
Cell voltage (nominal, V)	2.00	1.20	1.20	2.50	3.70	3.30 - 3.80
Charge cutoff voltage (V/cell)	2.40	Full charge detection by voltage signature	Full charge detection by voltage signature	2.70	4.20	3.60 - 4.20
Discharge cutoff voltage (V/cell)	1.75	1.00	1.00	1.15	3.00	2.50 - 3.00
Charge temperature	-20 to 50 °C	0 to 45 °C	0 to 45 °C	250 to 350°C	0 to 45 °C	0 to 45 °C
Discharge temperature	-20 to 50 °C	-20 to 65 °C	-20 to 65 °C	250 to 350°C	-20 to 60 °C	-20 to 60 °C

Table 1 Battery types and typical parameters

There are different types of batteries that have been and are being used for electric vehicles. As mentioned earlier, the cell generates a direct current (DC) due to a chemical reaction. This chemical reaction can be repeated, so in rechargeable batteries, the cell can restore its initial state by reversing the current. This has made batteries the most important storage devices for electric vehicles, and over time other technologies have been developed to achieve the desired performance [6]. Batteries have been developed over the years mainly with a view to improving energy efficiency, because the higher the efficiency of a battery, the higher the percentage of energy stored and usable in it. The charging time of a high-efficiency battery is shorter because a high-efficiency battery is equivalent to a high-capacity battery. This allows a greater depth of discharge (DOD) over the same period of time than less efficient batteries [10].

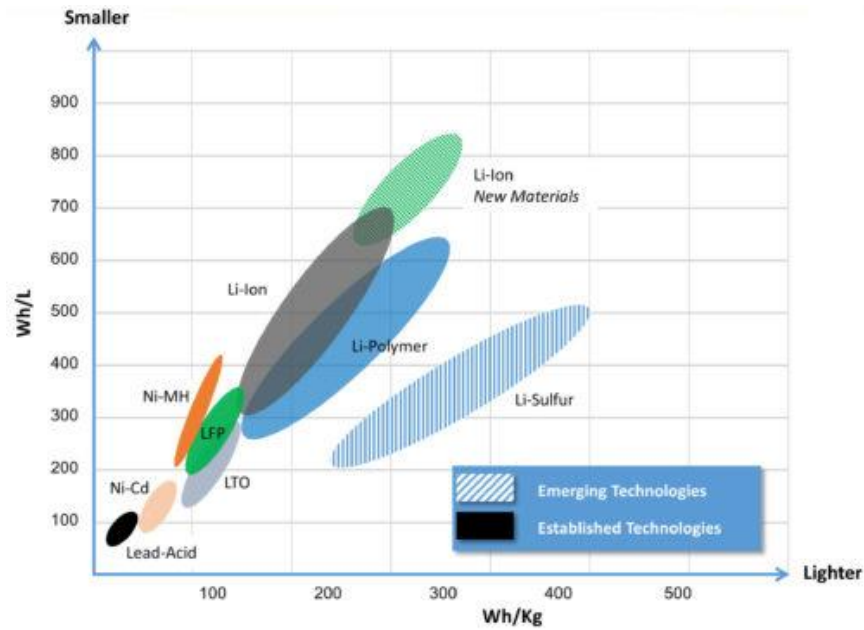
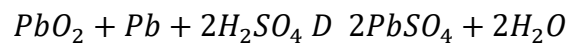


Fig. 20. Energy density comparison of different batteries (<https://inventuspower.com/lithium-ion-power-for-data-storage-and-servers/>)

### 2.3.1 Lead Acid

Lead-acid batteries are among the most commonly used batteries in the automotive industry. It is used in vehicles for starting, ignition and other electrical functions. The active ingredient is a lead oxide, which, together with the electrodes and electrolyte, is involved in the electrochemical reaction that provides charging or discharging of the battery [10]. During these processes the following reaction occurs:



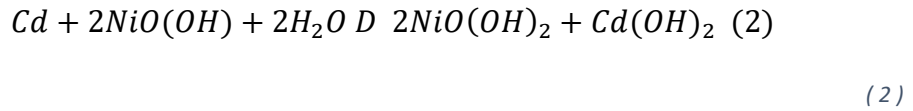
( 1 )

During discharge, lead sulphate and water are formed, while during charging the reverse reaction takes place. The lead-acid battery is one of the cheapest,

but compared to others, it is too heavy and cannot provide the energy required for EV applications [11].

### 2.3.2 Nickel Cadmium

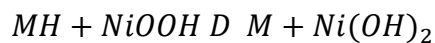
Nickel-cadmium batteries offer more advantages than lead-acid batteries, such as good lifetime, high internal resistance, and lighter weight [10]. Most of these positive aspects are their ability to absorb the gases produced during discharge. The following electrochemical reaction takes place:



This type of battery is affected by the memory problem. This problem is a disadvantage and occurs when the battery is recharged before it is completely discharged [11].

### 2.3.3 Nickel Metal Hydride

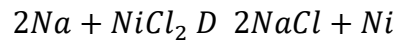
Nickel-metal hydride batteries have rapidly displaced nickel-cadmium batteries for use in electric vehicles, although they have a shorter discharge cycle than their predecessors. The presence of hydrogen instead of cadmium in the negative electrode improves energy density. They also do not have the same memory problem as nickel-cadmium batteries. The only drawback is that battery life is drastically shortened during rapid and repeated discharge cycles when high loads are applied. The electrochemical reaction can be represented by the following equation:



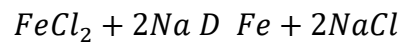
This type of battery is used more for HEVs than EVs [11].

#### 2.3.4 ZEBRA Battery

The cell of a ZEBRA battery consists of two different electrodes. There is liquid sodium at the negative electrode, and the positive electrode is nickel immersed in a solution of nickel, nickel chloride, iron sulphate and a liquid electrolyte. A solid electrolyte, usually consisting of beta alumina, separates the two electrodes. The following electrochemical reactions occur in the cell:



( 4 )



( 5 )

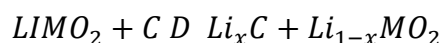
ZEBRA batteries have many advantages, including energy density comparable to lithium-ion batteries. However, compared to these, the ZEBRA are cheaper and have a higher coulombic efficiency. The only drawback is that because the electrolyte liquefies at 157 °C, a ZEBRA battery must operate in a temperature range of 250-350 °C, so it must be equipped with a heater. This unfortunately affects many applications [12].

#### 2.3.5 Lithium Ions

Lithium-ion batteries are currently most commonly used in the automotive industry for the realization of HEVs and EVs. Inside the cell, as in the batteries seen before, there is a negative and a positive electrode. The negative electrode is made of graphite, while the positive electrode can be made of different



lithium solutions. The electrolyte separating the two electrodes is liquid and consists of lithium salts in an ethylene carbonate solvent [2]. The following electrochemical reaction takes place:



(6)

However, the chemical composition of the cell can also be changed by the electrolyte and not only by the positive electrode. In fact, if you replace the liquid electrolyte with a lithium polymer, you get what is known as a lithium polymer (LiPo) battery. They are more expensive and have slightly less power than lithium-ion batteries but are more robust and can be used for multiple purposes due to their versatility [13]. Depending on the active material, we get different types of lithium-ion batteries. Their name depends on that of the active material of which they are composed and each of them has its own advantages, disadvantages and applications.

- Lithium Cobalt Oxide (LCO): This type of battery is made of lithium carbonate and cobalt, with the cathode made of cobalt oxide, while the anode is made of graphite, as in most lithium-ion batteries. Advantages include high specific energy, which is why they are mainly used for cell phones, laptops and cameras. However, disadvantages include short life, high cobalt cost and low specific power [14].
- Lithium Iron Phosphate (LFP): In this type, the positive electrode is made of lithium iron phosphate instead. Its long life and thermal stability make it one of the most widely used lithium-ion batteries. Stability and safety depend on the low strength. It becomes problematic when they are used at low temperatures, as their performance can be

affected. In addition, they have a low specific energy compared to other types [15].

- Lithium Titanate (LTO): Titanate lithium batteries are the only lithium-ion batteries in which the composition of the anode changes. Thus, the negative electrode is lithium titanate and the positive electrode is either nickel, manganese and cobalt or lithium manganese oxide. Their advantages include fast charging, safety, stability, and high density compared to non-lithium-ion batteries. Nevertheless, these batteries have low energy density compared to other lithium-ion batteries and are very expensive [14].
- Lithium Nickel Manganese Cobalt Oxide (NMC): In NMC batteries, the cathode consists of a combination of nickel, manganese and cobalt. This mixture can be made with different percentages of the material. A low concentration of cobalt, for example, results in cheaper batteries. The anode, on the other hand, makes the cell mechanically unstable during charging and discharging, as it tends to expand and contract as the silicon is added to the graphite. This type of lithium-ion battery is widely used because it can compensate for the shortcomings of the three different materials by combining them. Manganese is stable but has a low specific energy, while nickel is more unstable and has a high specific energy. Unlike pure cobalt batteries, NMCs have a slightly lower voltage [14].
- Lithium Manganese Oxide (LMO): In this type of battery, lithium manganese oxide is used in the cathode to improve ion flow within the cell. The three-dimensional structure that results also improves stability and safety and reduces internal resistance. Because of these characteristics, the LMO battery is used primarily in the medical field,

although it can also be found in HEVs, EVs, power tools and other fields. These batteries have a short life, but offer high specific power and can be charged quickly [15].

- Lithium Nickel Cobalt Aluminium Oxide (NCA): The cathode of these batteries is made of a mixture of metal oxides containing lithium, nickel, cobalt and aluminium. Because of their long life cycle and high specific energy, they can provide a high amount of current for long periods of time. They are not the safest among lithium-ion batteries and have a high price, although lower than that of LTO. Currently, they are mainly used for electric vehicle manufacturing [14].

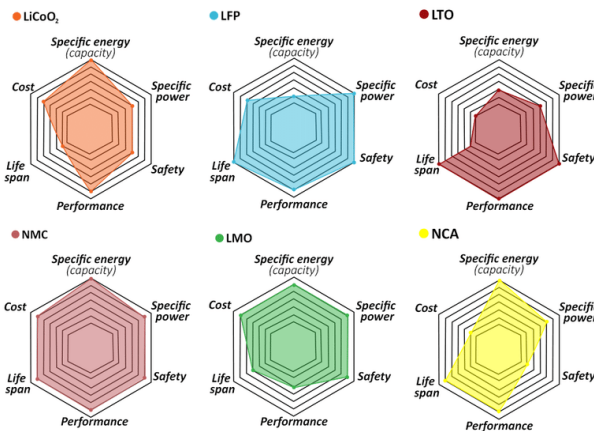


Fig. 21. Comparison between different types of cathodes in Li-ion batteries (Salgado, Rui, *The Latest Trends in Electric Vehicles Batteries*, Porto, 2021)

### 3. HARDWARE IN THE LOOP (HIL) APPLICATION

The following information, from 3.1 to 3.2 included, was provided by the company where I did my thesis (Kineton s.r.l.) and was collected from the dSpace systems user manual.

### 3.1 Real time simulation bases

Hardware-in-the-loop simulation (HIL) is a type of real-time simulation, used to develop and verify complex embedded systems. HIL simulation is mainly used to test the design of a controller developed through Model-Based Design. Within the V-shaped workflow, this simulation is one of the most important steps since it precedes the tests in the automotive industry.

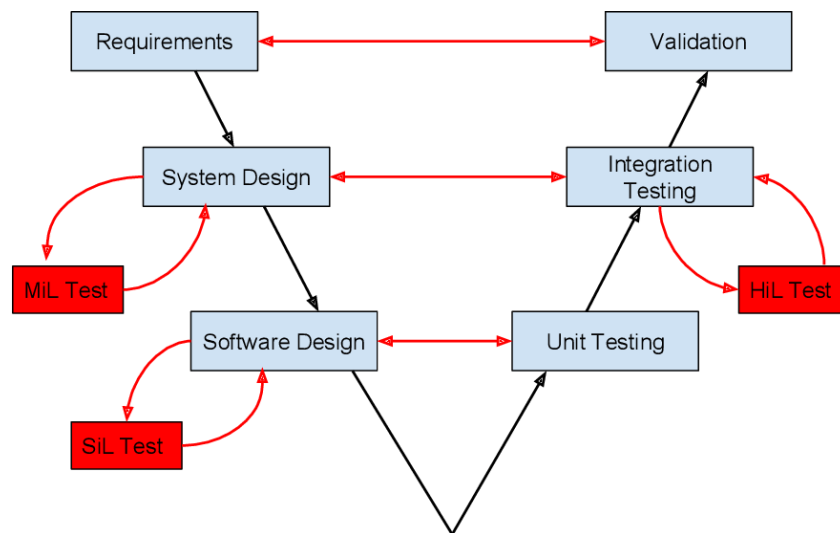


Fig. 22. Overview of the V-shaped workflow in Model-Based Design (Bergström, David, Model- and Hardware-in-the-Loop Testing in a Model-Based Design Workflow, Lund, 2016)

HIL is responsible for the simulation and development of Device-under-tests (DUT), which are mostly electronic control units (ECU). The simulation proceeds

in real time using systems and processors. It's important that every single task assigned by the system happens in exactly one millisecond, whether it's synchronous or asynchronous.

Simulators are designed to test ECUs in a HIL simulation. Among the different simulators that can be used one of the most famous, in addition to being the one on which the tests performed for the thesis project are based, is the SCALEXIO of the company dSpace.

### 3.1.1 HIL simulation purpose

A HIL was created with the aim of developing and testing complex systems. In general, a hardware in the Loop is a form of real time simulation comprising one or more components, those commonly called control units. In case it has only one control unit, then the system is called Component HIL, because it uses the least number of nodes to get a first operation. Through a HIL it is possible to stimulate an ECU from an electrical and electronic point of view in such a way that it can be felt in an environment as close as possible to a real environment. To this end, the real time simulation includes a physical and mathematical model replicating a scenario that is reliable. Therefore, in the case of powertrains, all the control units that deal with the traction of the vehicle will be considered.

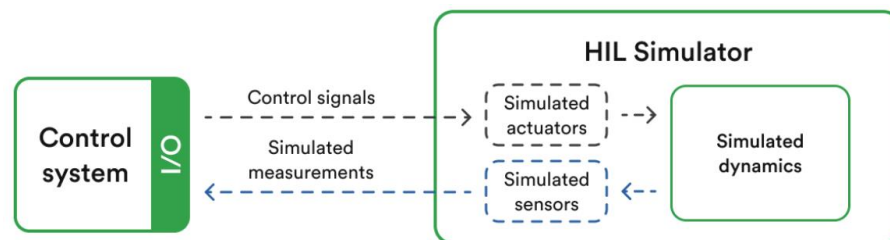
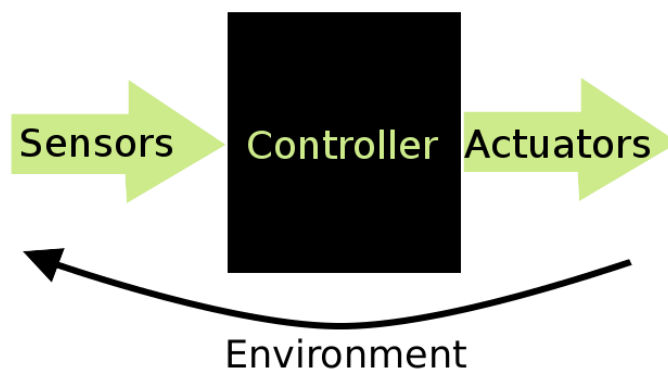


Fig. 23 HIL testing concept (<https://www.rinf.tech/why-is-hardware-in-the-loop-hil-testing-important-for-automotive-solutions/>)

Inside the HIL systems the (real) electronic control units are interfaced with everything that is simulated through a Simulink model. In the model we can find Soft ECU, (not real) control units that replicate the behaviour of an ECU compared to the under-test control unit. In addition, it is also possible to simulate sensors or, if you do not want to use real ones, actuators. The whole is based on a physical and mathematical model to simulate the vehicular dynamics.

### 3.1.2 Sensors and Actuators

The task of the HIL simulator is therefore to trick the control unit or the control units, making them believe they are inside the vehicle. What are the outputs for the control unit will then be considered input for the simulator and consequently the outputs of the simulator will be input for the connected control unit. In this case, through the dedicated channels that are inside the simulator, you can set the acquisitions provided and received by the control unit. In this perspective, what are the actuators will become the inputs for our simulator, while the sensors will be simulated and become the outputs that provide the necessary data to the control unit to behave as software.



*Fig. 24 Diagram of the relationship between a control unit and its sensors and actuators (<https://www.manufakturet.create.aau.dk/resources-test/senors/>)*

The main difference between sensor and actuator is that the sensor converts a physical action into electrical signals, while the actuator is responsible for converting the electrical signal into mechanical work. The sensors convert the input variable into a signal that can be used for measurements. The sensor aims to convert the physical amount that is measured into another easier to handle.

Classification of the sensors:

- Active/Passive: The Active is if the physical effect on which it is based guarantees the transformation of the energy of the measurand into electrical energy (thermal, mechanical, irradiation, ...). In these the conditioning circuit has the task of adapting the parameters of the electricity generated by the transducer to the characteristics of the input of the measured system. An example is thermoelectric, pyroelectric and so on. Passive if the effect of the measurand results in an impedance change of the sensory element, as for the strain gauge. Here the conditioning circuit is essential for the generation of the electrical signal.
- According to the measured quantity: temperature, humidity, lighting, speed, pressure, etc.
- Based on the quantity that provides the output: resistive, inductive, capacitive, etc.
- Voltage, Current, etc.
- Analog/Digital.

Actuators are devices that, in an automated system, transform an automatic control, processed by an electronic control board, into a physical action in the process that is regulated. An actuator allows you to act on a system to change its behaviour and get the desired output. More generally, the actuator can be defined as a transducer capable of transforming a signal from a physical (electrical) domain into an equivalent amount of another domain (mechanical, hydraulic, thermal, etc.).

Classification according to the type of electric actuation control:

- Digital
- Analog (PWM)

Therefore, for each real control unit used it is necessary to provide a different configuration between it and the simulator. These are designed starting from the electrical schematic of the control unit provided, so that it is possible to devise a Wiring Harness.

All sensors are simulated, as written above, through the Simulink model. For the actuators, however, it is possible to proceed both through simulation and through real components. The choice of using real or simulated components falls on the configurator and is made taking as a starting point the function of the actuator itself. The first question a configurator should ask is whether that real behaviour can easily be simulated. For example, a switch is easily simulated through a Simulink model. On the contrary, if we wanted to consider a throttle valve (TVA), it would be too complicated to simulate its opening. Moreover, considering the possibility of real time simulators to use real loads, it becomes easier to understand the faults that can come from them. In fact, one of the advantages of HIL simulators is that they allow us to connect complex actuators directly to the simulator, such as the hydraulic braking system, turbochargers and other valves besides the TVA.



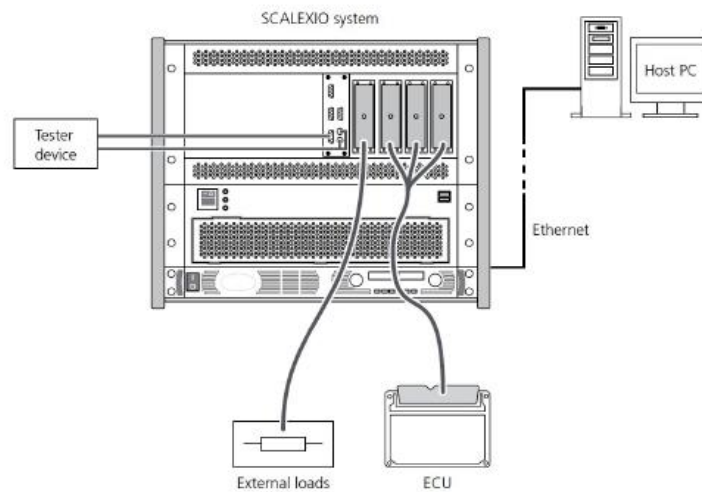
*Fig. 25 Example of a throttle valve for automotive applications (Bischoff, Bastian, Learning Throttle Valve Control Using Policy Search, Munich, 2013)*



### 3.2 dSpace SCALEXIO System

As previously written, one of the HIL simulators used in the automotive industry is the simulator SCALEXIO of the company dSpace. The SCALEXIO, simulating the ECU system, behaves like a real vehicle, being able, for example, to generate, measure, connect, receive, and send the signals of the sensors and the control unit, as well as providing the battery voltage.

The system is controlled by real-time application implemented in a host PC connected to SCALEXIO via Ethernet connection.



*Fig. 26. Connections to a SCALEXIO system (Help Desk, SCALEXIO dSpace)*

Within the hardware of the SCALEXIO we find different types of boards or units, listed below:

- SCALEXIO Processing Unit: is the unit that provides the necessary power to complete applications in real-time.

- SCALEXIO I/O boards: are boards that allow a large number of input and output channels, including channels dedicated to specific functions.
- MultiCompact I/O units and boards: the most used unit, in this case, is the DS2680 I/O Unit. This card allows you to manage a large number of channels for specific applications.
- Highflex I/O boards: the typical unit, in this case, is the DS2601 for signal measurement, but the DS2621 for generation, the DS2642 for failure simulation and power switches and the DS2671 for bus board are also used. They are used for different input and output functions due to their versatility.
- Slot Units: allow the installation of different boards in the SCALEXIO rack.

Depending on the boards installed, the SCALEXIO simulator used can obtain different configurations.

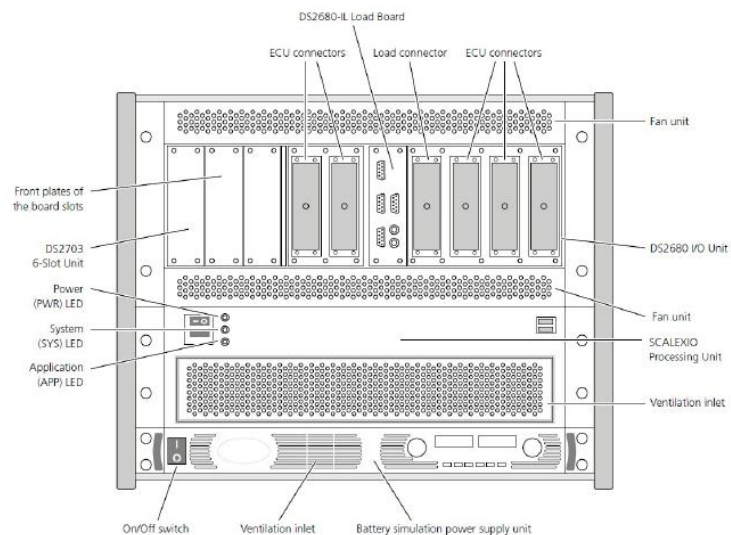


Fig. 27. Detailed view of the SCALEXIO simulator (Help Desk, SCALEXIO dSpace)

At the back of the SCALEXIO system we find several connectors, such as main connectors, Ethernet connectors or IOCNET connectors.

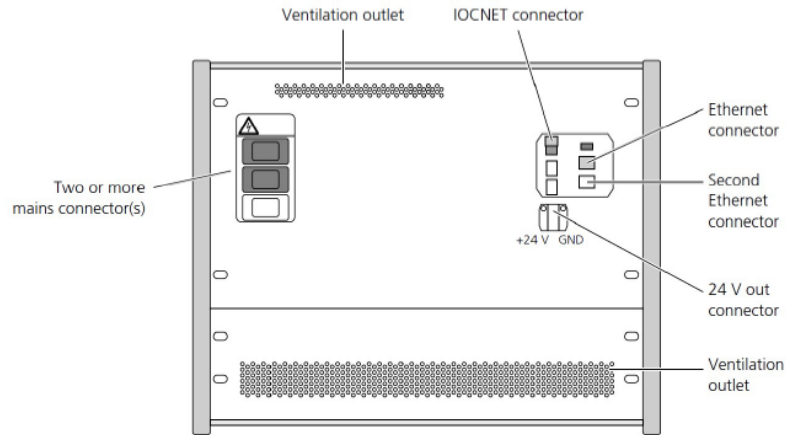


Fig. 28. Back of a SCALEXIO system (Help Desk, SCALEXIO dSpace)

### 3.2.1 SCALEXIO Processing Unit

To provide the computing power to calculate the real time applications is the Processing Unit. Among its components we can find:

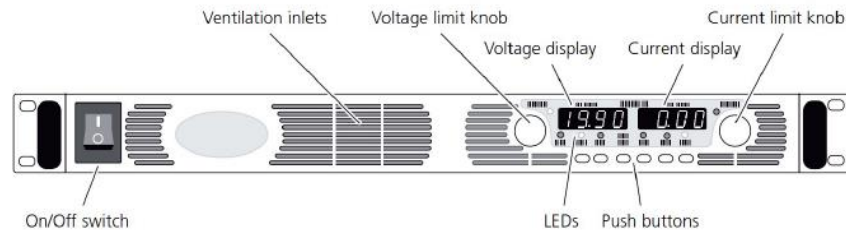
- SCALEXIO Real-Time PC: consisting of an ATX motherboard and a multicore main processor.
- Angular processing units: are six in total and each of them has a resolution of 16 bits.
- DS2502 IOCNET Link Board: provides, through IOCNET connections, communication with the HighFlex Board and the MultiCompact I/O Unit.



Fig. 29 SCALEXIO Processing Unit (Help Desk, SCALEXIO dSpace)

### 3.2.2 Power Supply Unit

As its name suggests, it is a unit used to generate voltages or currents useful for simulating the battery of a vehicle and is controlled by software. The knobs allow to set the maximum values of voltage and current, so as to be able to deliver what is supplied to the control unit so as to remain in safety conditions. In addition, during the simulation you can see directly from the displays the voltage and current values delivered at that precise instant.



*Fig. 30 Power supply unit (Help Desk, SCALEXIO dSpace)*

Some I/O card channels require this power supply because the control unit does not generate the necessary voltage. Among these are, for example, the Power Switches that will be deepened later.

### 3.2.3 DS2680 I/O Unit

The DS2680 I/O Unit is a board that allows you to manage many channels. It features a DS2671 module with channels that can be configured as transceivers, CAN, or LIN. The simulator wiring is connected to the connectors or, rarely, to the Breakout Box (BOB). In the BOB the dSpace side is in red, while in black is that of the control unit. The connectors are Hypertac type and connect the control unit to the simulator, generating a mapping between the pins. In total there are 6 Hypertac connectors, since on the SCALEXIO we have two MultiCompact boards.

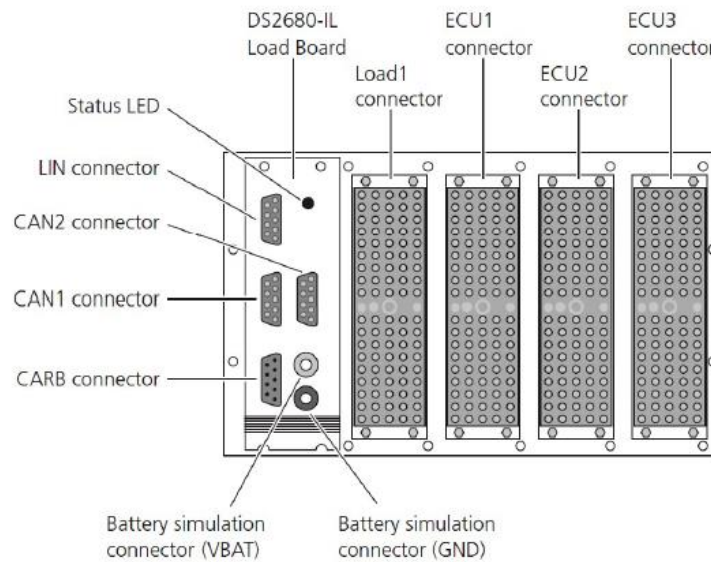


Fig. 31. Connectors in the DS2680 I/O Unit (Help Desk, SCALEXIO dSpace)

The Load Board is used to introduce internal loads to channels, such as inflexions such as resistors, through jumpers.

The Load Plate, however, is different from the Load Board, because it allows you to insert loads that can be both real and emulated. As difficult as the dynamics are, the real ones between the two are always preferred. Among the insertable components are butterfly valves and injectors, studied in their electromechanical behaviour. The connections to these actual components allows you to know parameters such as current consumption, valve position or injector control. There are two other Hypertacs that connect to Load Plates, but the jumpers are not connected to all channels.

In general, the connections between the DS2680 and the motor control unit (ECM) are set by analysing the electrical diagram provided by the carmaker. This allows you to understand which channel the ECM pin should be connected to. The list of these connections is called wiring harness.

The existing channels are as follows:

- Power Control (1): takes care of the control of the power supply of the Battery inside the simulator.
- Power Switch (6): the DS2680 I/O provides six channels of Power Switch to change the supply voltage of the battery simulation to the ECM or other connected control units. These can be set on or off via software and can also be used by the Failure Insertion Unit (FIU) to simulate short circuits. Each pin can be used in a range from 0 to +60 V and 0 to +6 Arms. Vbat is the battery voltage generated by the external voltage generator, which goes to close the switch in case I want to plug battery. In this case in the upper ECU pins we will have Vbat. Then close the switches the control unit will have a battery voltage equal to that provided by the voltage generator. The maximum current is 6A per pin, as represented by the fuses in the diagram below. The GND of the Power Switches are all short-circuited and their reference is the GND of the simulator. The load does not necessarily need to be connected to the GND, but it can also be connected to the power supply.

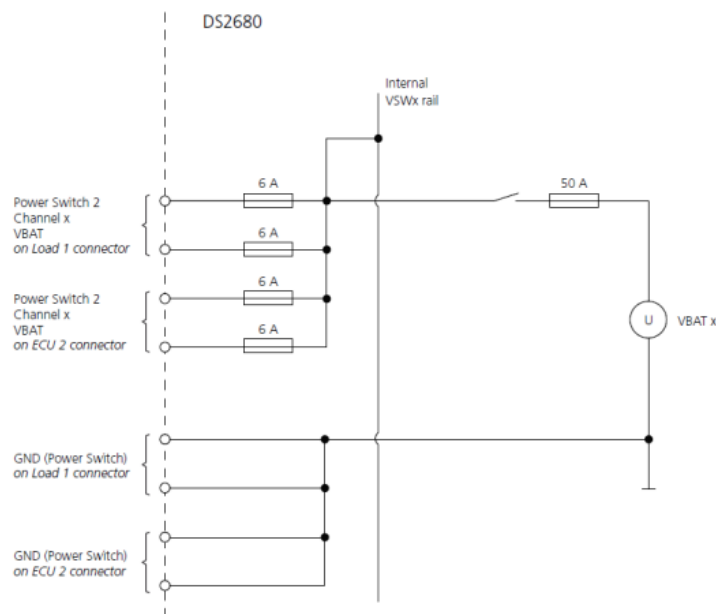


Fig. 32. Power Switch channel circuit (Help Desk, SCALEXIO dSpace)

- Analog In 1 (20): is used to measure the signal of an analog voltage in the range from 0 V up to +60 V. Each channel provides three different pins: one for measuring the signal, one for providing the mass and the other for changing the analog voltage in the plate loads. Through the FRU we can do all the configurations we want. J1 and J2 (jumper) can be configured via the Load Board. I have two possibilities, either I use the Load Plate by inserting a resistor on it, because the signal pin is connected to the Load, or I insert a resistor directly into the Load Board. The first option is preferable, because the resistance remains visible and is more manageable.

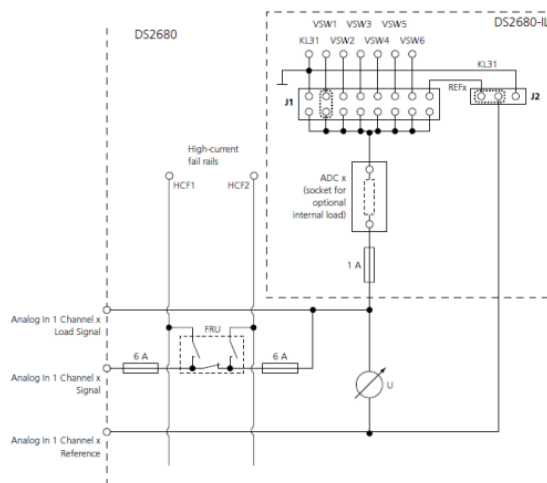


Fig. 33. Analog In 1 channel circuit (Help Desk, SCALEXIO dSpace)

- Analog In 2 (2): they are more or less like Analog In 1, but they are more differential. The current passing through them is smaller and there is a higher sampling rate. They are used for faster signals, mainly for lambda probes. There are only 2 inside the DS2680 because in a combustion engine vehicle you can find up to two lambda probes. The first post upstream of the catalyst to regulate the stoichiometric ratio of the mixture, the second post after the catalyst so that it can make a diagnosis.

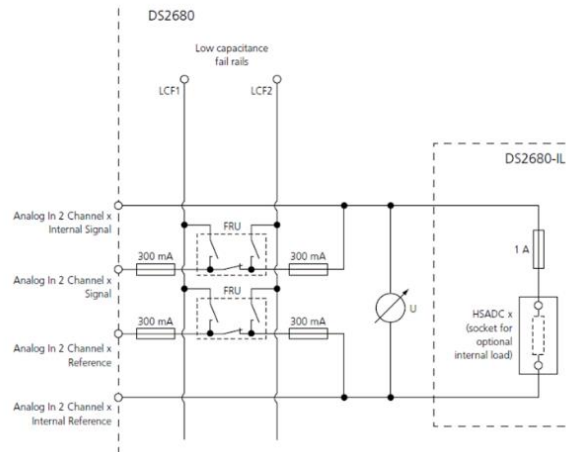


Fig. 34 Analog In 2 channel circuit (Help Desk, SCALEXIO dSpace)

- Digital In (30): is used to measure the digital voltage signal changing in a range from 0 to +60 V with a Threshold voltage ranging from 0 to 23.8 V. Each channel provides three pins, just like the Analog In 1. Digital Ins always have the reference on the GND simulator and cannot be changed. Usually the digital signals that we have to acquire from the control unit are either in high side or low side, so we will insert either a pull-down resistor or a pull-up based on the electrical schematic.

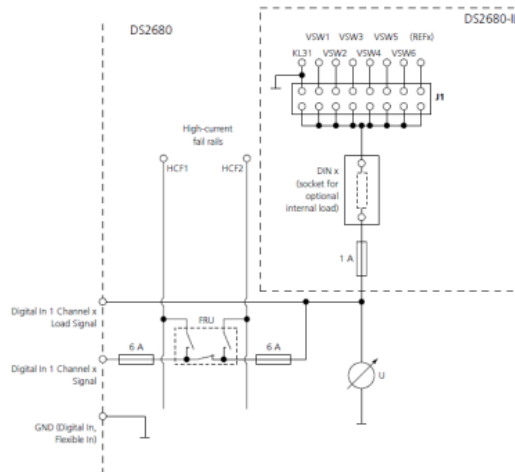


Fig. 35. Digital In 1 channel circuit (Help Desk, SCALEXIO dSpace)



- Flexible In (18): they are channels that allow to measure the current in addition to the voltage. To be exact, digital voltage, digital current or analog current. Of this type there are no Out channels and they are particularly useful for injections. Each channel provides three pins: one for measuring the signal, one for providing ground and the other for changing the analog voltage in the charged plate.

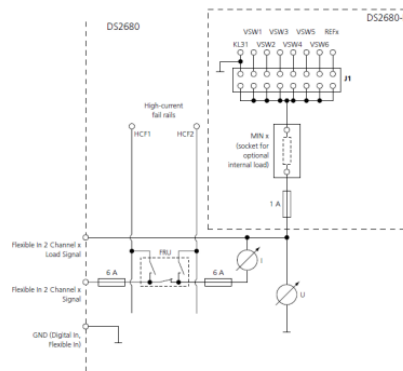


Fig. 36. Flexible In channel circuit (Help Desk, SCALEXIO dSpace)

- Analog Out 1 (15): they are used to provide output voltages for sensor simulation. It has two pins, one providing the voltage signal and the other the ground signal. They can also be generated without the help of the external generator. Each Analog Out has its own reference.

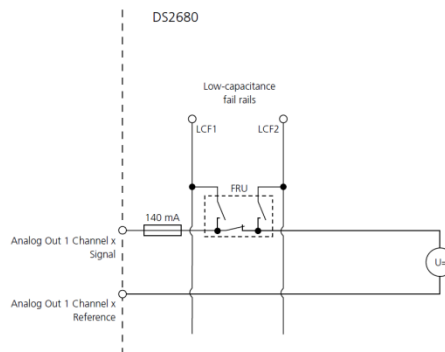


Fig. 37. Analog Out 1 channel circuit (Help Desk, SCALEXIO dSpace)

- Analog Out 2 (2): they're only for lambda probes. It's an output that also gives us the ability to go and modify the current. It's basically the same as Analog Out 1, but it has a faster response.

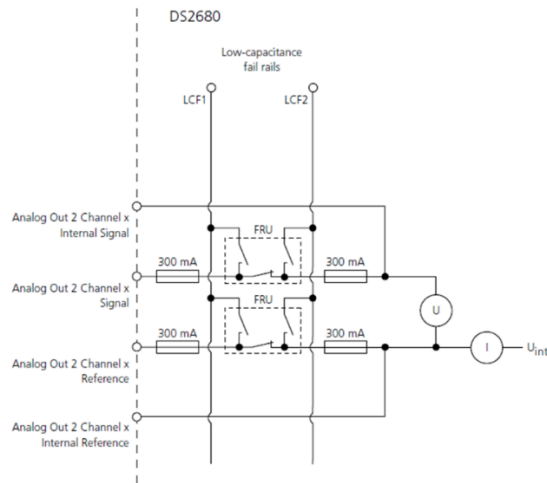


Fig. 38 Analog Out 2 channel circuit (Help Desk, SCALEXIO dSpace)

- Analog Out 3 (7): they are useful if I have to use sinusoidal signals at a higher frequency. A sine wave can also be generated via Analog Out 1, but the 3 allows me to have a reference to a lower potential. They are used for the simulation of knock sensors.

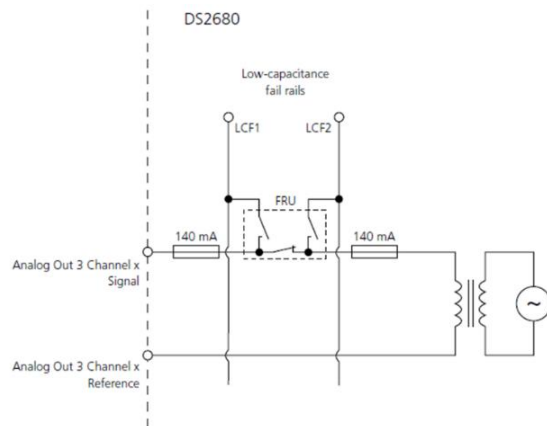


Fig. 39 Analog Out 3 channel circuit (Help Desk, SCALEXIO dSpace)

- Analog Out 4 (8): they allow to have both a voltage generator and a current generator. They have almost the same operation as the Flexible Ins, but in this case they are used as output from the simulator.

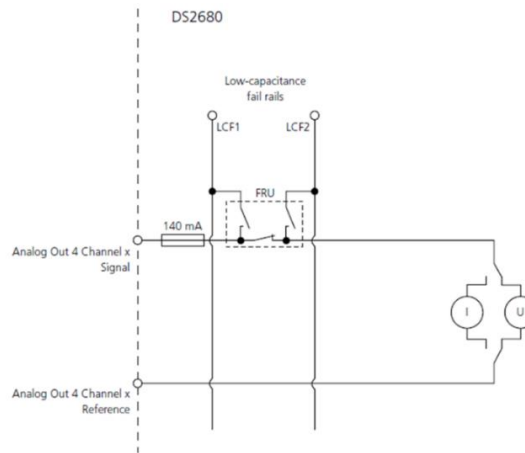


Fig. 40 Analog Out 4 channel circuit (Help Desk, SCALEXIO dSpace)

- Digital Out 1 (28): they're used to simulate a 0 V digital output or in a range between 5 V and 60 V. Each channel has a common pin to connect an external reference voltage. You can configure it in various ways via ConfigurationDesk. So, based on the digital value we want to get the transistor will close or not. Can be set as Low Side, High Side or a Push/Pull. In low-side can only carry to ground. The high-side can be set to an internal reference voltage indicated as Vbat or an external voltage greater than +5 V. In Push/Pull logic, if I have a 1 here I will touch the 12 V, while if I have a 0 will go to the GND. It's possible to put a Push/Pull with the enable, so if the enable will be active it will also be the Push/Pull. This way I can also leave it floating by opening the two switches.

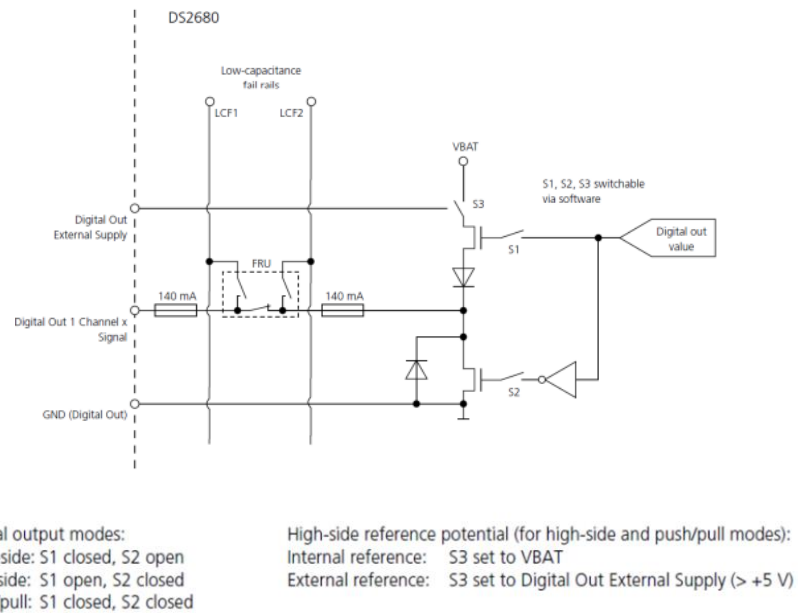


Fig. 41. Digital Out 1 channel circuit (Help Desk, SCALEXIO dSpace)

- Resistance Out (12): they're in charge of simulating resistance. Each channel has two pins: one for the output signal and the other for the ground signal. It emulates a variable resistor like a digital potentiometer, so we can decide the resistance value. It is used for example with temperature sensors.

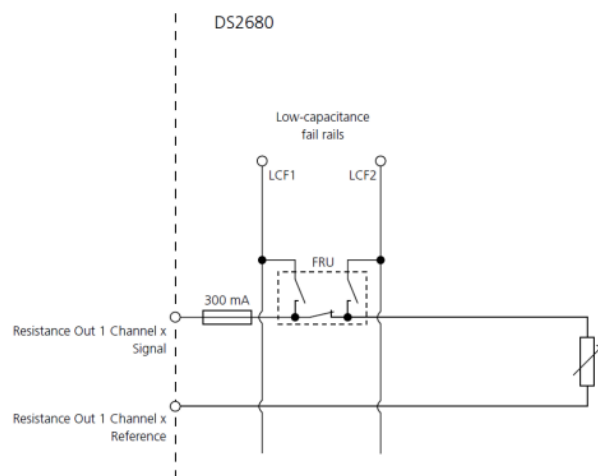


Fig. 42. Resistance Out channel circuit (Help Desk, SCALEXIO dSpace)

### **3.3 ECU – Simulator connection**

To proceed with the connection between a control unit and the HIL simulator, you need to know first what the characteristics of the simulator are and what is the board I/ O that uses. The most commonly used MultiCompact I/O board is usually the DS2680. In particular, in the case of dspace's ESIL simulators, we can find up to two DS2680 cards mounted inside the SCALEXIO rack.

The knowledge of the board used, and its channels allows us to take the first step to build the connection that will join the control unit used and the simulator. Knowing the channels used it is possible to understand what will be the inputs and outputs of the model that will be built on Simulink, always bearing in mind that the simulator inputs will be the outputs of the control unit and vice versa. The second step is to know what the pinouts of the control unit are, an operation possible thanks to the electrical schematic provided by the eventual supplier.

#### **3.3.1 Battery Management System (BMS)**

The larger the number of cells contained in a battery pack, the more difficult it becomes to manage it. Battery monitoring and management is possible through a battery management system (BMS). BMSs are electronic control units that deal with different tasks, the most important among them is the balance between the cells contained within the battery pack so that it is possible to prevent degradation and operations that do not ensure safety [16]. The task of the BMS is therefore to improve battery performance and its durability. It protects, balances, and monitors cells, communicating battery status to high-end systems.

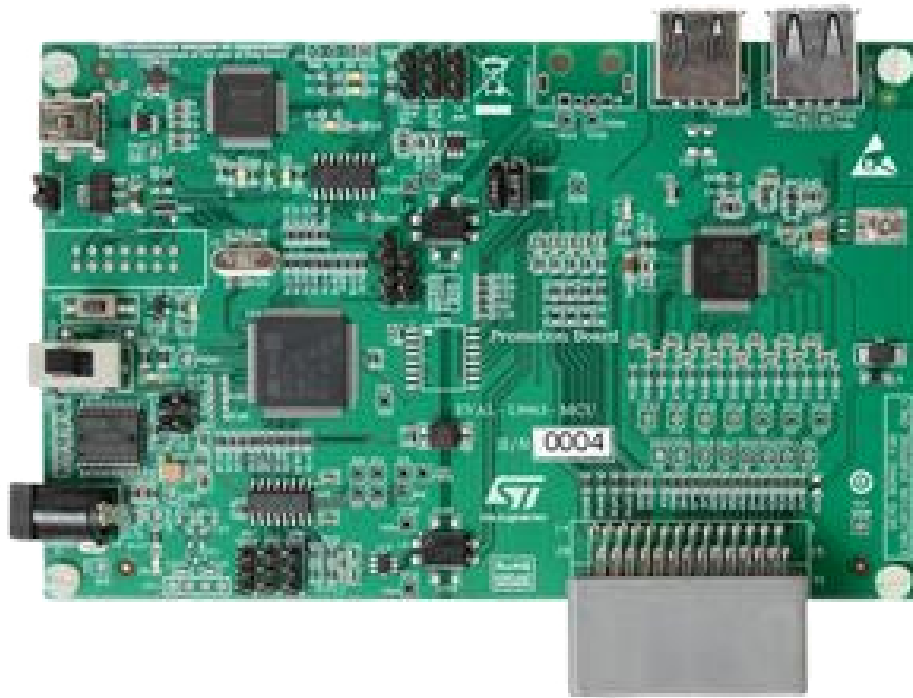


Fig. 43 BMS architecture (<https://www.mouser.it/new/stmicroelectronics/stm-battery-management-system-solution/>)

In figure 43 you can see the typical architecture of a BMS for a high voltage battery. As written above, the individual cells are combined, via series and parallel connections, in modules, each of them monitored by a balancing circuit. It's possible to monitor the individual cells through integrated circuits, but obviously requires more complex systems from the point of view of the electronic control unit. Different low-level cards communicate with a host controller [16].

Not having a real BMS control unit, for the purposes of the thesis were studied electrical schematics of different control units to find the inputs and outputs to be used for the realization of the model. Among those of which I can disclose information there are the control unit TLE9012DQU Infineon and the Orion

BMS 1 and 2. In particular, it was decided to order the first one in order to use it for the thesis experiment.

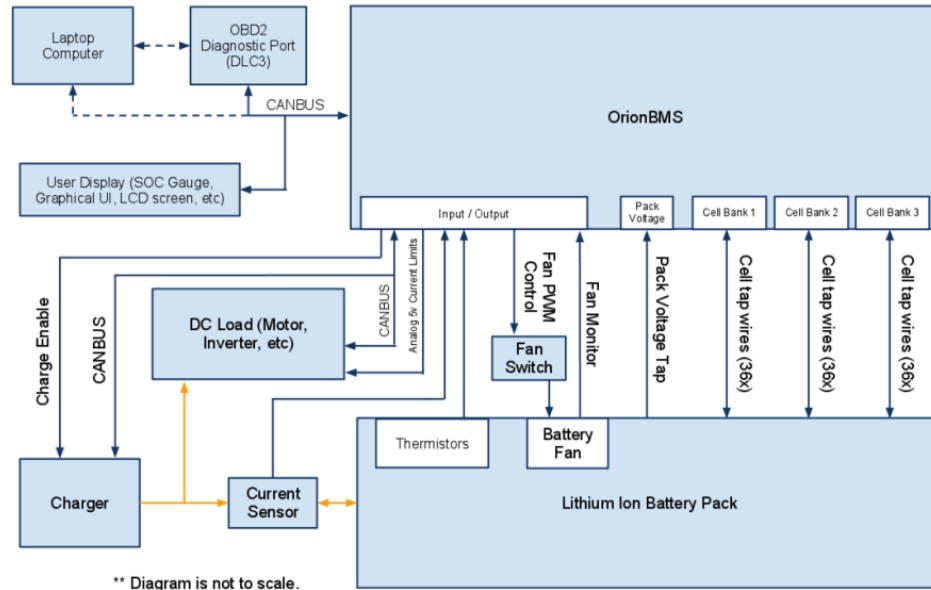


Fig. 44 Wiring overview of Orion BMS (Ewert Energy Systems, Orion BMS wiring and Installation Manual, 2018)

From the simplified scheme provided by the two Orion BMS, you can guess which are the main pinouts that we find in any battery management system. Among these, the most important, for obvious reasons, are the pins that are connected directly to the cells of the battery pack that allow their monitoring and balancing. The balancing, in this case, is controlled by the most important BMS actuator; the cell management controller (CMC). This unit can be mounted directly on the modules or individual cells that make up the battery pack and is able to measure the voltage and temperature of the cell. Moreover, through a 2-wire bus, it can communicate these parameters with the other CMSs. Its design makes it possible to install directly on the cell so as not to use additional cables, but it is also integrated in the BMS in some electronic control units. The CMC is therefore classifiable as an additional control unit, which

deals with the individual element since the communication with the remaining ECUs of the vehicle is entrusted directly to the BMS [17]. As for the balancing managed by the CMC, there are two different types: passive and active.

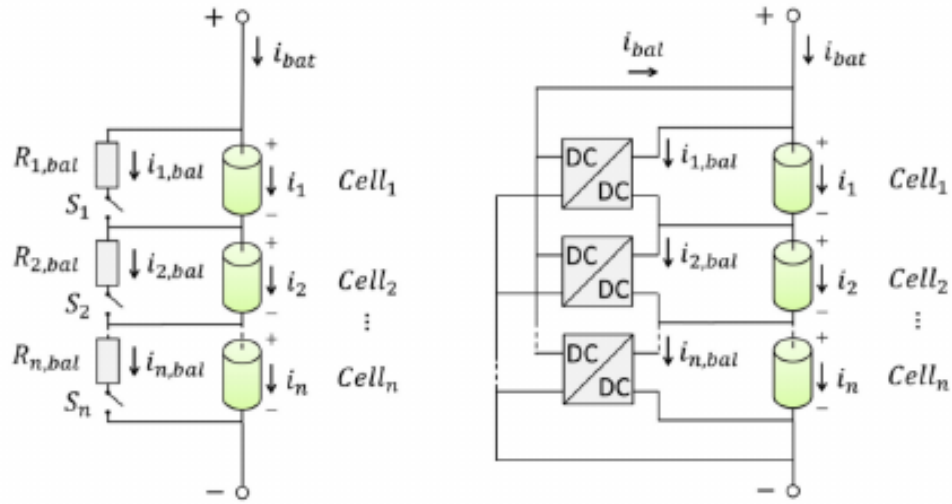


Fig. 45 Passive (left) and active (right) balancing circuit (Barreras, Jorge, Multi-Objective Control of Balancing Systems for Li-Ion Battery Packs: A Paradigm Shift?, Aalborg, 2014)

Passive cell balancing is simpler than active one and is based on the idea of cell discharge through a bypass road. This can be both external and integrated, but it must be considered that the life of the battery can be strongly affected during the discharge cycle due to the dissipated energy [18]. Active balancing uses capacitive or inductive charge shuttling to transfer the charge between cells. Unlike the passive one, this technique does not waste excess charge, going rather to distribute energy between cells. The main problem is that the components that require it are expensive and that it is more difficult to implement [18].



## 4. BATTERY DESIGN ON SIMULINK

Once you have identified the inputs and outputs from the chosen electronic control unit, it becomes possible to design a model on Simulink that can meet the set objectives. When creating the model, it is therefore necessary to take into account the data expected by the BMS and the input parameters that, whether or not generated by the actuators connected to the control unit, will allow the battery to respect the initial conditions and the physical behaviour it should have in reality.

### 4.1 Electrical Model

The purposes of the electric model are as follows [2]:

- Estimate the voltage provided to the power converter.
- Estimate the State of Charge (SOC).
- Estimate the losses.

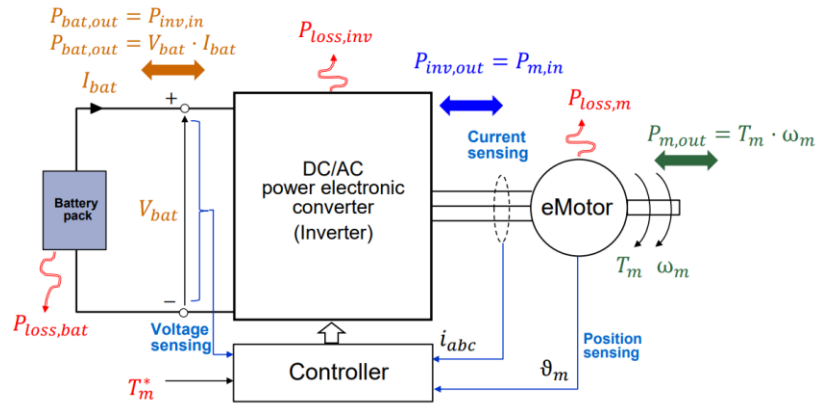


Fig. 46 Battery pack in an electrical powertrain (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Torino, 2022)

The modelling of a battery pack is particularly difficult because of the electrochemical processes with a high level of non-linearity that take place

inside it. Therefore, in order to take into account the behaviour of the battery, different circuit models have been designed to take into account the functioning and reactions that take place inside the cell. The structure of these circuits is extremely linear and concise, so that it is feasible to analyse the variables and to identify the parameters necessary for the estimation of SOC.

#### 4.1.1 Common equivalent circuit models

The most common equivalent circuit models for simulating a battery cell are four; the Rint model, Thevenin model, PNGV model, and GNL model. The Rint model is the simplest of the four and is represented with an ideal  $E$  voltage source placed in series with an internal resistance  $R_o$ . Unfortunately, because of its simplicity, it has low accuracy and cannot represent the influence of the electrochemical reaction that takes place inside the cell [19].

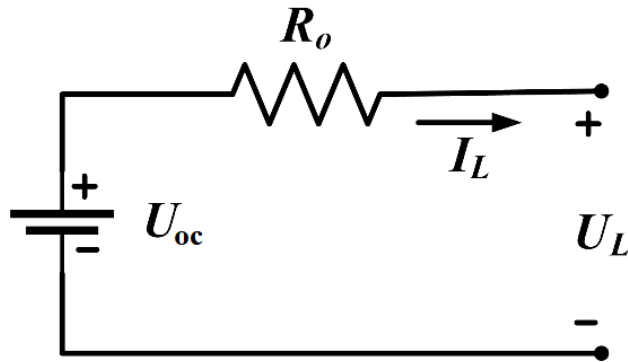


Fig. 47 Equivalent circuit of battery according to the Rint model (Hossain, Eklas, *A Comprehensive Review on Second-Life Batteries: Current State, Manufacturing Considerations, Applications, Impacts, Barriers & Potential Solutions, Business Strategies, and Policies*, Khulna, 2019)

The equivalent circuit of the Thevenin model, on the other hand, adds to Rint's basic model  $n$ -RC branches to simulate the effect of polarization in charge and discharge processes. Unlike the Rint model, it manages to represent the

dynamics that occur inside a battery cell, but we must consider that due to the complexity of the operations that take place inside a cell it is not possible to perfectly simulate the performance with only RC branches [19].

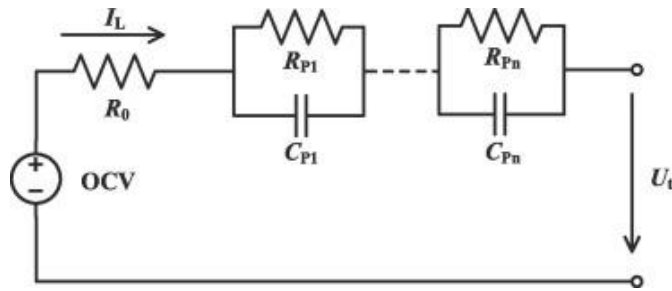


Fig. 48 Equivalent circuit of battery according to the Thevenin model (Ding, Xiaofeng, An improved Thevenin model of lithium-ion battery with high accuracy for electric vehicles, 2019)

The PNGV model, on the other hand, adds a capacitance plate  $C_0$  on the Thevenin model. This capacity has the task of simulating the open circuit voltage (OCV) that changes with the accumulation of current. This makes its accuracy higher than the Thevenin model but is not adequate for estimating SOC in practical applications since it has a high computational complexity to establish its own parameters [19].

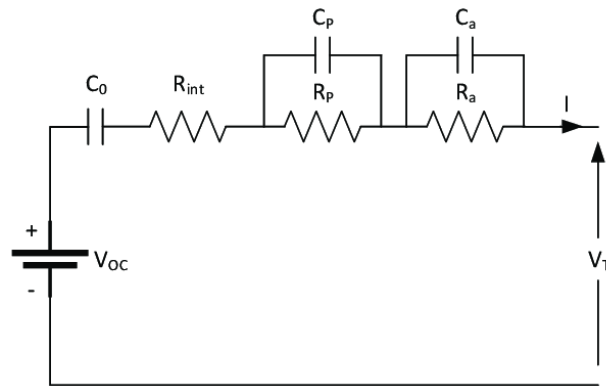


Fig. 49 Equivalent circuit of battery according to the PNGV model (Saldana, Gaizka, Analysis of the Current Electric Battery Models for Electric Vehicle Simulation, Bilbao, 2019)

The latest, GNL-model, combines all previous circuits seen. Inside the model there are so many components between resistors and capacitors that make the dynamic order too high, so the data that needs to be processed are too many and it is difficult to compute them. Thus, this is also not suitable for real-time estimation of SOC [19].

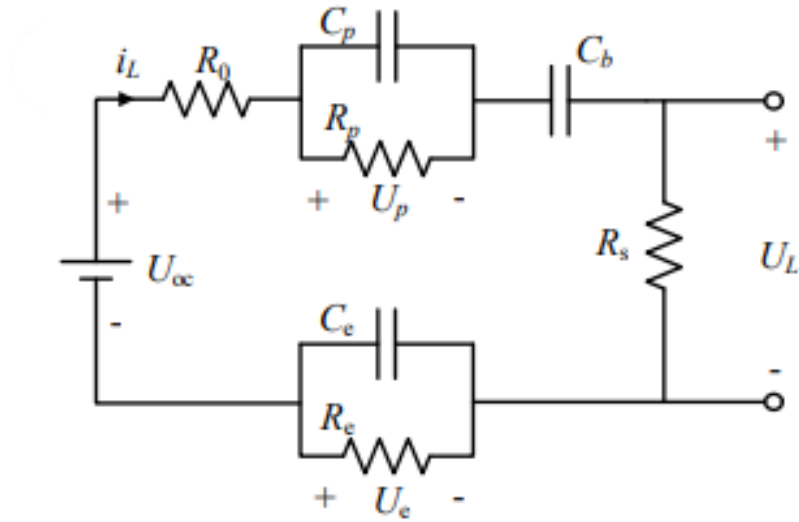


Fig. 50 Equivalent circuit of battery according to the GNL-model (Kai, Wang, State of Charge (SOC) Estimation of Lithium-ion Battery Based on Adaptive Square Root Unscented Kalman Filter, Foshan, 2019)

#### 4.1.2 Thevenin equivalent circuit on Simulink

Among the models presented in the previous paragraph, the Thevenin equivalent circuit model proves to be the most suitable for the real-time estimation of the SOC. To study the dynamics of the electrochemical reactions that take place within the cell you can use n-order RC-branches. However, the more branches there are, the more complex it is to calculate the dynamics of the system, so a good compromise is to use first-order or second-order RC-branches. Despite the small number of orders, they still manage to properly reflect the characteristics of battery cells, proving also easy to implement for engineering applications [20].

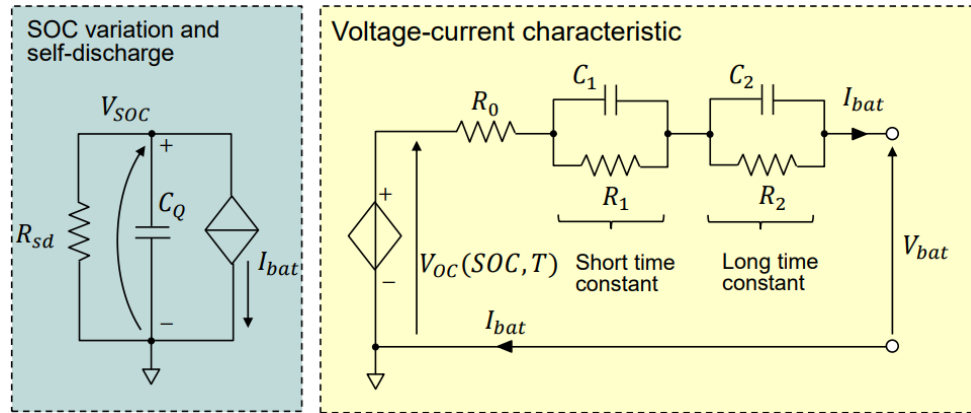


Fig. 51 Simplified battery Thevenin circuital model (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Torino, 2022)

In the circuit model above, there are two RC-branches with different time constants.

$$\tau = R \cdot C$$

(7)

I utilized this Thevenin model as a starting point for the modelling of the electric model, so I looked for a battery that could be used as a basis for the construction of the battery pack.

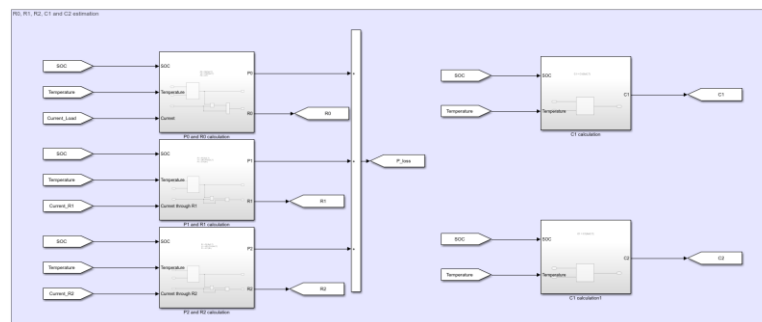


Fig. 52  $R_0$ ,  $R_1$ ,  $R_2$ ,  $C_1$ ,  $C_2$  blocks in Simulink (From Simulink model)

I selected the Panasonic NCR18650PF as the battery cell, of which I was able to find the datasheet and experimental values to use for the RC-branches. Obviously, the model has been developed to be able to test various types of batteries, for which it is possible to replace the data with those of any other cell of which the datasheet is possessed and the necessary values. The RC-branches can be removed if there are no useful parameters to identify the electrochemical dynamics, keeping in mind that this would exclude internal battery reactions from the model.

Rated capacity <sup>(1)</sup>	Min. 2700mAh
Capacity <sup>(2)</sup>	Min. 2750mAh Typ. 2900mAh
Nominal voltage	3.6V
Charging	CC-CV, Std. 1375mA, 4.20V, 4.0 hrs
Weight (max.)	48.0 g
Temperature	Charge*: 0 to +45°C Discharge: -20 to +60°C Storage: -20 to +50°C
Energy density <sup>(3)</sup>	Volumetric: 577 Wh/l Gravimetric: 207 Wh/kg

<sup>(1)</sup> At 20°C <sup>(2)</sup> At 25°C <sup>(3)</sup> Energy density based on bare cell dimensions

*Fig. 53 NCR18650PF Lithium-ion battery specifications (Panasonic, Datasheet NCR18650PF, 2016)*

The values of capacities and resistances, whether those of the RC-branches or the internal one, depend directly on the value of SOC and temperature. Therefore, the most suitable method to estimate their values is to use Look-Up-Table (LUT).

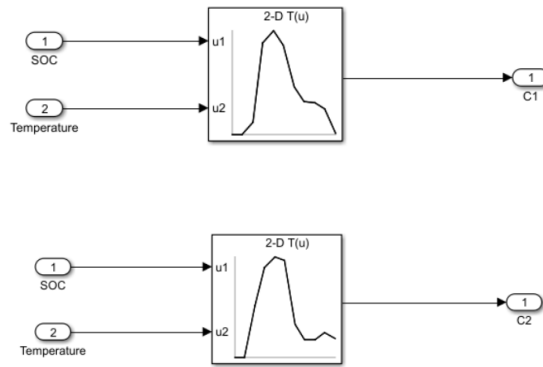


Fig. 54 Computation of C1 and C2 in Simulink (From Simulink model)

The smaller the electrical resistance is the less restrictions are affecting the battery in providing power. As a result, the cell's runtime decreases as this resistance increases since the voltage edges approach the cut-off voltage value more quickly.

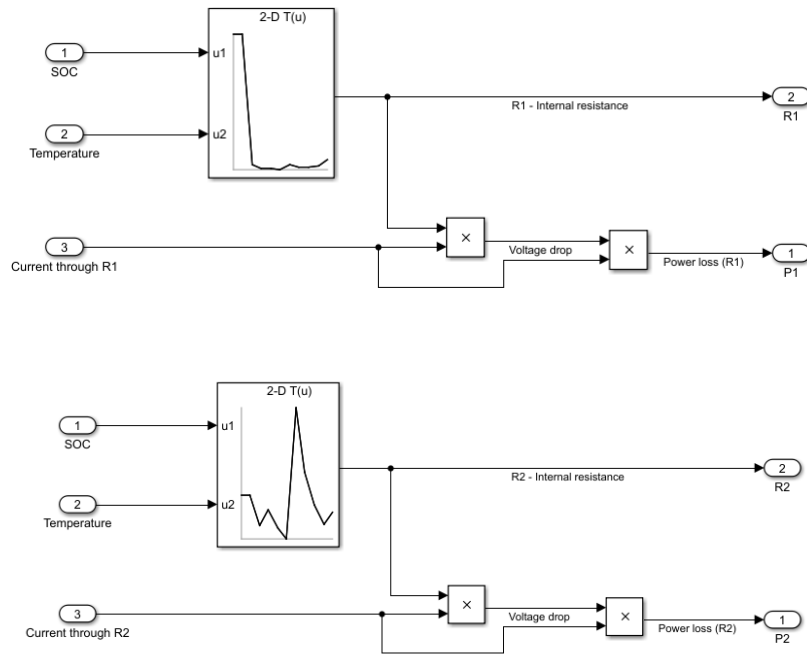


Fig. 55 Computation of R1 and R2 in Simulink (From Simulink model)

Depending on the type of battery cell, we will have several changes with the variation of the SOC. For example, resistance levels in Nickel Metal-Hydride are greater when the SOC is low and right after charging. Instead, during the discharge, the internal resistance falls at the mid-charge and then begins to rise once more. With Lithium-ion, the charge trend from empty to full is very flat. Between 0% and 30% of the SOC, during charging and discharge, is typically when the largest changes take place. Lastly, the internal resistance of the lead-acid battery increases with discharge. This is due to the decrease in specific gravity, so much so that the trend between increased resistance and decrease in specific gravity is linear [21]. Despite the type of battery cell, it has been demonstrated that when the temperature rises, the resistance of the cell lowers. However, high temperatures still cause the battery to degrade faster [22].

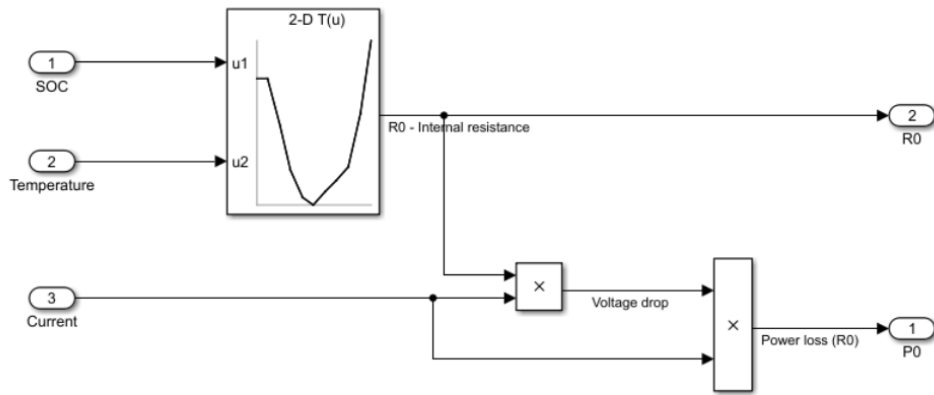


Fig. 56 Computation of  $R_0$  in Simulink

The first part of the Thevenin circuit deals with the variation of the SOC and is defined as a self-discharge circuit. This circuit contains a capacitor, a resistor and a current generator with current equal to that absorbed or released by the battery [2].



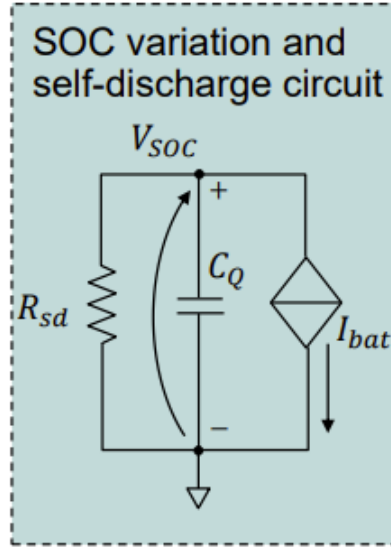


Fig. 57 SOC variation and self-discharge circuit (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Torino, 2022)

The following formula is used to calculate the capacity  $C_Q$  value:

$$C_Q = 3600 \cdot Q_{rated} \cdot k_{cycle} \cdot k_{temp}$$

( 8 )

$Q_{rated}$  is the rated capacitance, also called nominal capacitance. It's the capacitance that the battery cell has at the beginning of its life cycle before it's used. As mentioned above, the battery is destined to degrade over time due to various factors, so its capacitance will change after several cycles. As a rated capacitance, however, it is always necessary to use the initial one. Instead,  $k_{cycle}$  and  $k_{temp}$  are corrective gains that depend on the number of battery life cycles and temperature. In case they are unknown, it's possible to set them both equal to one [2].

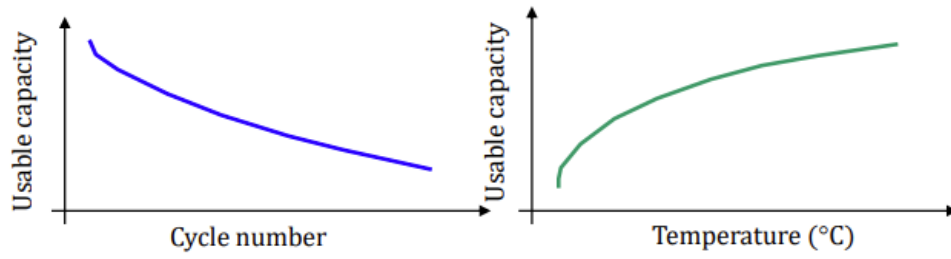


Fig. 58 Usable capacity considering the cycle number and temperature of the battery cell (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Torino, 2022)

$R_{sd}$ , on the other hand, is the self-discharge resistance. Self-discharge is a phenomenon that occurs within battery cells, whether primary or secondary. It is, to be precise, a chemical reaction that reduces the charge stored inside the battery without it being connected between its own electrodes or with any external circuit. Like normal discharge, self-discharge occurs faster at high temperatures and its rate is higher in secondary cells than in primary cells. The self-discharge rate depended mainly on SOC, charging current, temperature and battery type. Returning to the parameter used in the Thevenin circuit, if you do not know this resistance you can set to tens of  $k\Omega$  or eliminate it completely.  $V_{SOC}$  will be deepened in the paragraph concerning the calculation of the SOC [2].

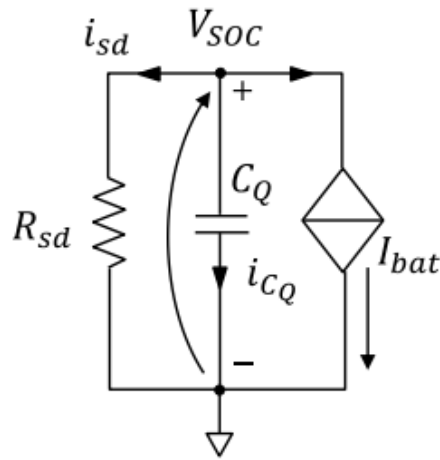


Fig. 59 Behaviour of the self-discharge circuit (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Torino, 2022)

The second part of the Thevenin equivalent circuit includes a voltage source, the internal resistance and, in our case, two RC branches. The open circuit voltage  $V_{oc}$  depends directly on the temperature and the SOC. In fact, typically  $V_{oc}$  is generated through a LUT or through a polynomial interpolation considering the SOC.  $V_{oc}$  is generally provided by the company that produces the battery [2].

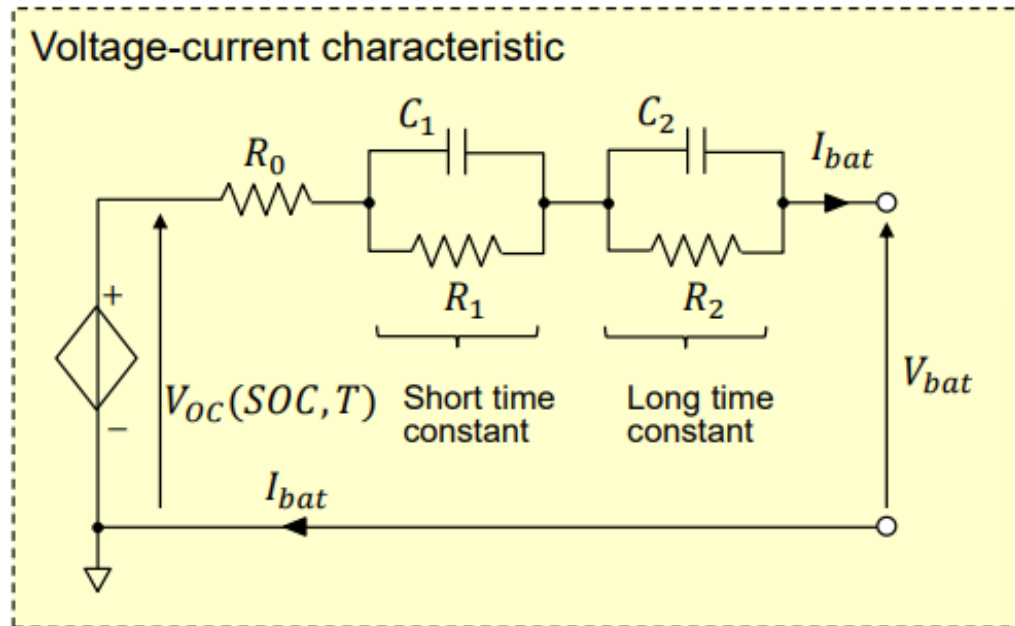


Fig. 60 Voltage-current characteristic in Thevenin equivalent circuit (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Torino, 2022)

The battery load  $I_{bat}$  is modelled as a source of current. It, in the case of the discharge process, represents the current required by all those components within the vehicle that need the power supplied by the high-voltage battery. These include the Inverter and the APM. As for the charging process, the current comes directly, if any, from the charge port or from the charging mechanisms inside the hybrid vehicle.

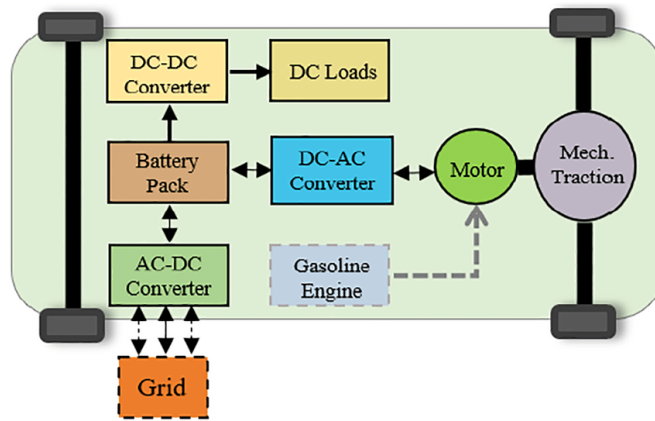


Fig. 61 Block diagram of the current through a hybrid vehicle (Kumar, Siddhant, Battery charging topology, infrastructure, and standards for electric vehicle applications: A comprehensive review, Mandi, 2021)

In the model the load current has been integrated as follows:

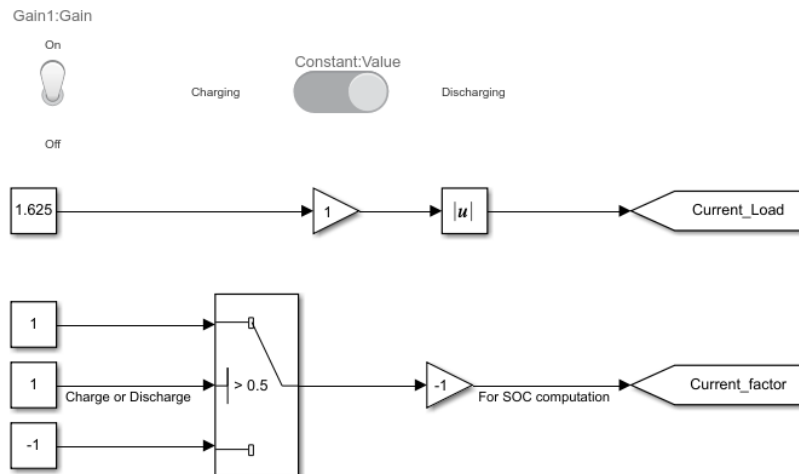


Fig. 62 Input current and Current Factor for the computation of SOC (From Simulink model)

Instead of entering a current pulse I chose a constant followed by a gain controlled by a toggle switch. In this way you are able to simulate the switching on and off of the battery according to your wishes and not for a pre-set and always equal time interval. Also, I inserted a switch whose condition is

controlled by the button placed next to the toggle. This allows you to change modes, switching from charging to discharging in the section dedicated to the SOC.

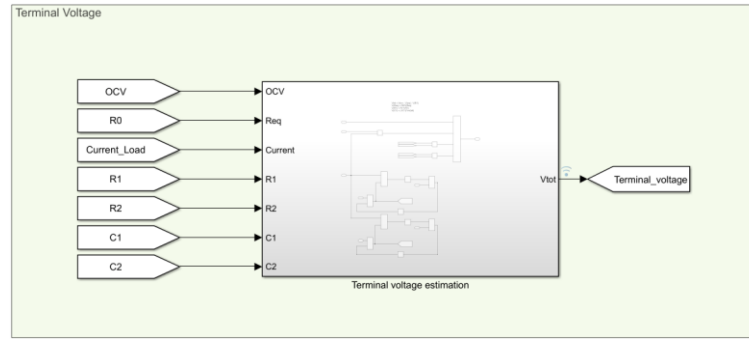


Fig. 63 Terminal voltage block in Simulink (From Simulink model)

Considering the battery current as an input variable, the following formula will be used to calculate the terminal voltage:

$$V_{bat}(t) = V_{oc}(t) - R_0 \cdot I_{bat}(t) - v_1(t) - v_2(t) \quad (9)$$

The last two voltages are the tensions that occur between the RC branches, calculated as follows:

$$v_1(t) = I_{batt} \cdot R_1 \cdot (1 - e^{-\frac{t}{\tau_1}})$$

$$v_2(t) = I_{batt} \cdot R_2 \cdot (1 - e^{-\frac{t}{\tau_2}}) \quad (10)$$

Using the formulas discussed above and the elements within the Thevenin equivalent circuit, the Simulink model needed to calculate the voltage at the cell terminals was modelled.

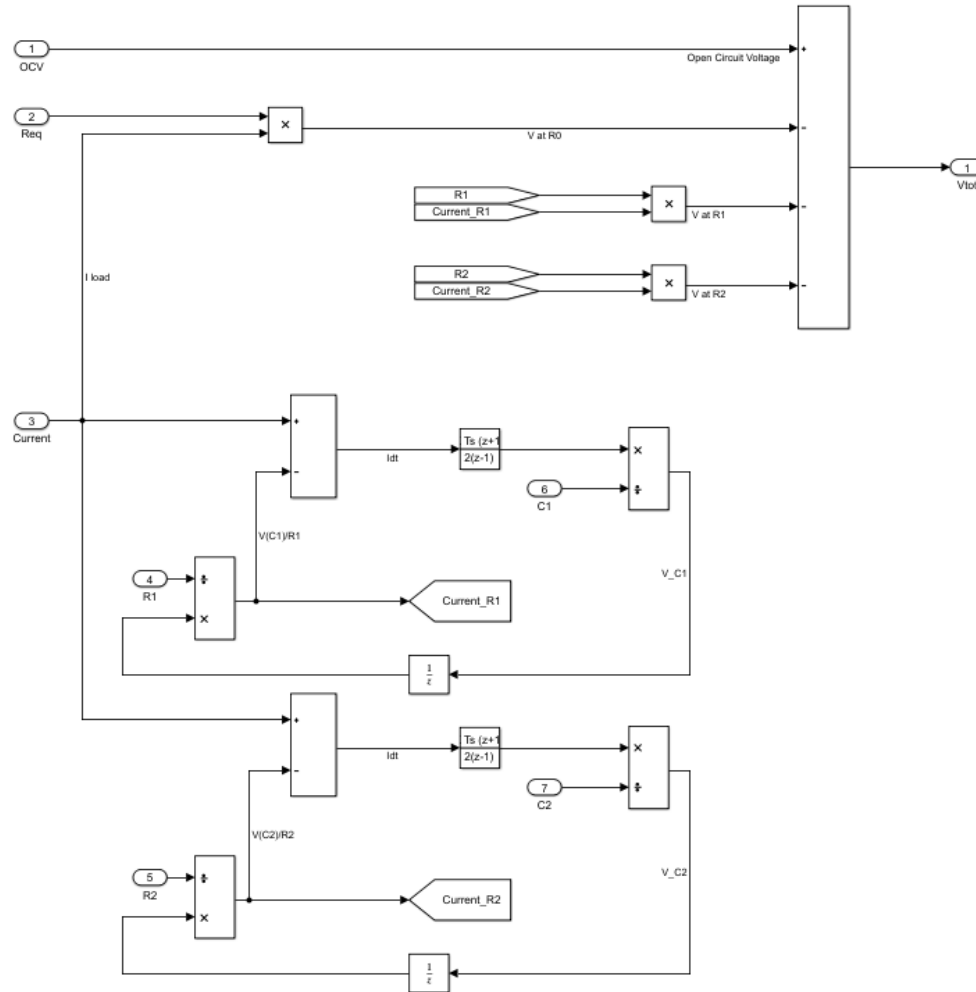


Fig. 64 Computation of  $V_{bot}$  in Simulink (From Simulink model)

To experimentally identify the parameters and values of the components within the battery cell this is discharged through a current pulse. Then, measuring the battery voltage, you can identify the voltages related to the internal resistance and resistances of the RC branches. The first step is due to the tension generated

on the internal resistance  $R_0$ . After that the tensions due, in our case, to the two RC branches come into play. The open circuit voltage is recorded after a sufficient relaxation time [2].

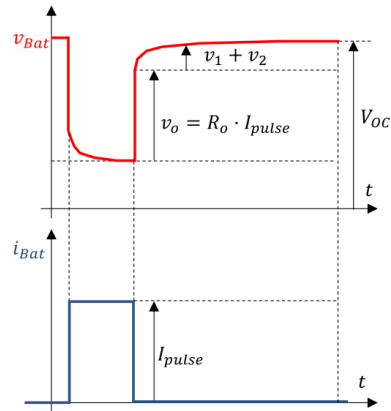


Fig. 65 Voltage trend with respect to current pulses (Musumeci, Salvatore, *Energy sources in HEVs and EVs*, Torino, 2022)

The result obtained with the model implemented on Simulink is shown below:

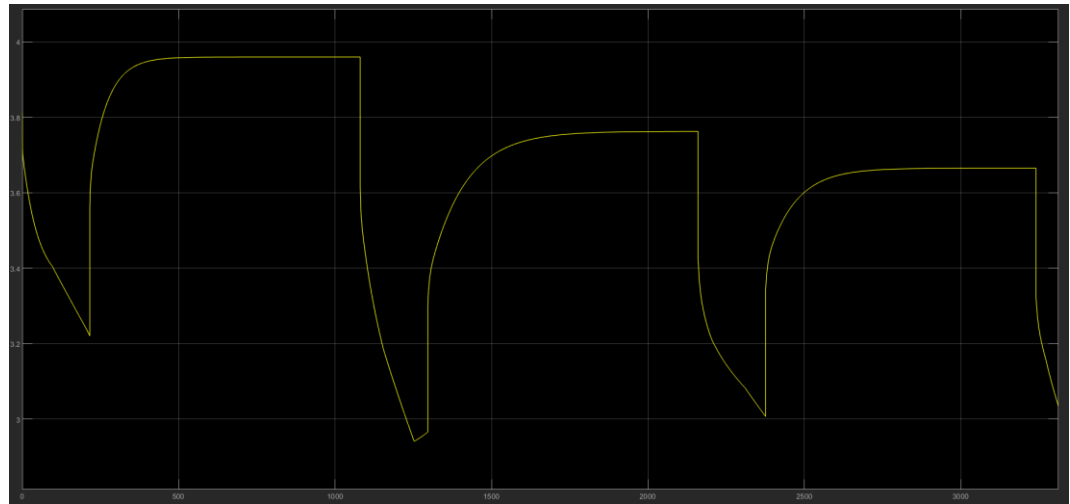
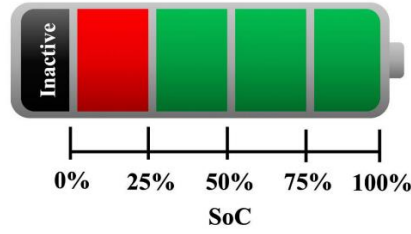


Fig. 66 Terminal voltage graph of a single battery cell (From Simulink model)

## 4.2 State of Charge (SOC) computation

The remaining capacity of a battery, stated in Ah and expressed as a percentage of its rated capacity, is known as the state of charge [23]. This is the most important parameter to consider when using a battery pack. As written above, the battery management system is responsible for reading the state of charge of the battery pack, interfacing with the CMCs that identify that of each single cell used in the various modules. Understanding the state of charge of each cell allows us, if necessary, to intervene with a balance, active or passive, so that all cells are brought to the same value of state of charge. Moreover, through it we can also understand how much the state of health of the cell is and whether it is damaged. Because it is dependent on the battery type and its intended use, estimating the state of charge is difficult.



*Fig. 67 Stored energy status of a battery (Espedal, Ingvald, Current trends for State-of-Charge (SoC) Estimation in Lithium-Ion Battery Electric Vehicles, Trondheim, 2021)*

SOC is commonly defined as the capacity that can be released by the battery in relation to its maximum capacity, indicated by the manufacturer, in percentage.

$$SOC = \frac{C_{releasable}}{C_{rated}} \cdot 100\%$$

( 11 )



If a discharge process is occurring, then we could consider Depth of Discharge (DOD) with the same formula. This indicates the amount of capacity of the battery cell that has been discharged.

$$DOD = \frac{C_{releasable}}{C_{rated}} \cdot 100\%$$

( 12 )

To improve the accuracy of the SOC estimation, which is a BMS's primary responsibility, we have worked to do so throughout the years. Actually, accurate assessment avoids the battery cell from being overloaded or over-discharged, which might result in long-term harm to the system. By precisely measuring the coulombs and the current circulating within the cell under different conditions and voltage, it is possible to determine the SOC. This data is then used together with the temperature, age of the cell and other relevant parameters obtained by the battery manufacturer to increase accuracy [23]. The main methods for estimating the charge state are three: the coulomb counting method, the voltage method and the Kalman filter method.

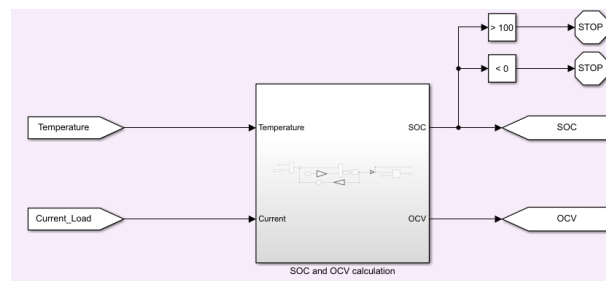


Fig. 68 SOC block in Simulink (From Simulink model)

Before going deeper into the individual methods for estimating SOC, it is necessary to specify that the Simulink model was developed to stop the simulation if the SOC exceeded its limits.

#### 4.2.1 SOC estimation with Coulomb counting method

The Coulomb counting method is, among the three, the most used, because it is less complicated than the Extended Kalman Filter (EKF) and more precise than the voltage method. This technique calculates the remaining capacity inside the battery considering the accumulated charge that flows or is absorbed by it. The formula used is this:

$$SOC = SOC(t_0) + \frac{1}{C_{rated}} \int_{t_0}^{t_0+\tau} (I_b - I_{loss}) dt$$

( 13 )

Starting from an accurate analysis of the current of the battery cell and its initial SOC, having a capacity already known, the SOC can be estimated by integrating the charge and discharge currents. However, we must take into account the losses that occur during the charging and discharging cycles, as they lead to errors that could compromise the accuracy of the estimation. Therefore, to avoid that the result is too far from reality, we must consider these factors [23].

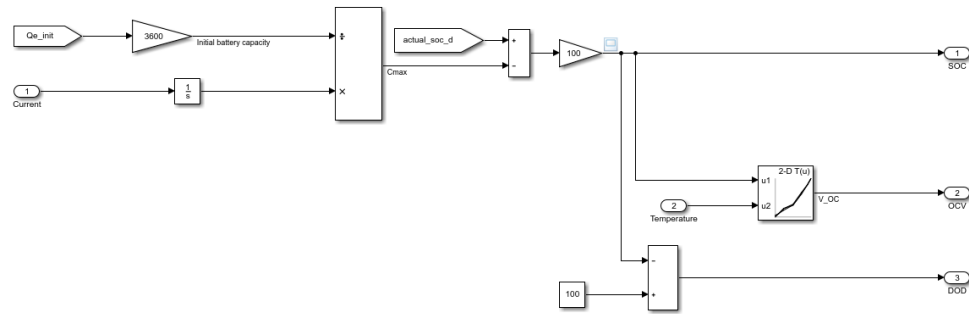
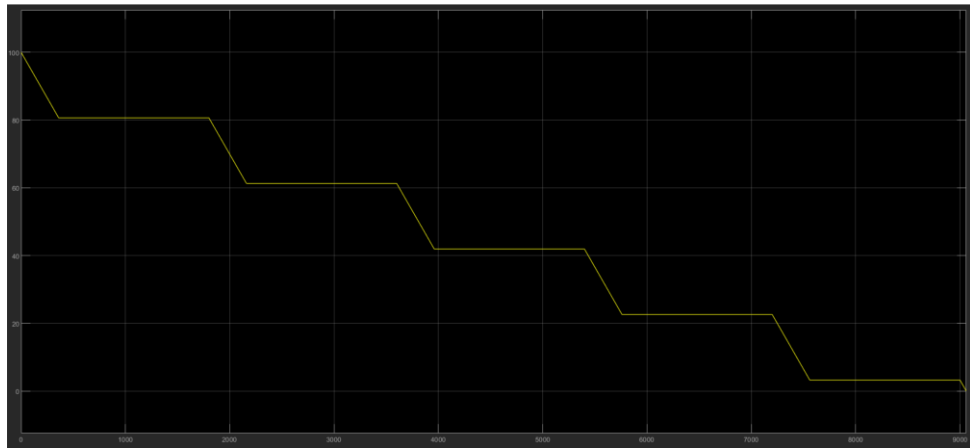


Fig. 69 Computation of SOC with Coulomb counting method (From Simulink model)

The image above shows the blocks used in the model to estimate SOC through the Coulomb counting method. Considering the advantages of the following method, I decided to adopt it during the first part of the thesis work, thus discarding the less precise voltage method and the more complex EKF. Following equation (11) I was able to implement the method mentioned in my model. In the first phase of the experiment the implemented solution was able to meet the criteria underlying the activity, both in the consideration of the single cell and for the entire module.



*Fig. 70 SOC graph obtained using the Coulomb counting method (From Simulink model)*

However, when the thesis work reached the stage where the model had to be necessarily adapted to a real-time environment, the first problems arose. Nevertheless, when there was a change from discharge to charge or vice versa, the model was not able to consider it. In fact, as soon as such change happened, in the SOC appeared a proportional jump to the value of SOC initial inserted to begin the simulation.

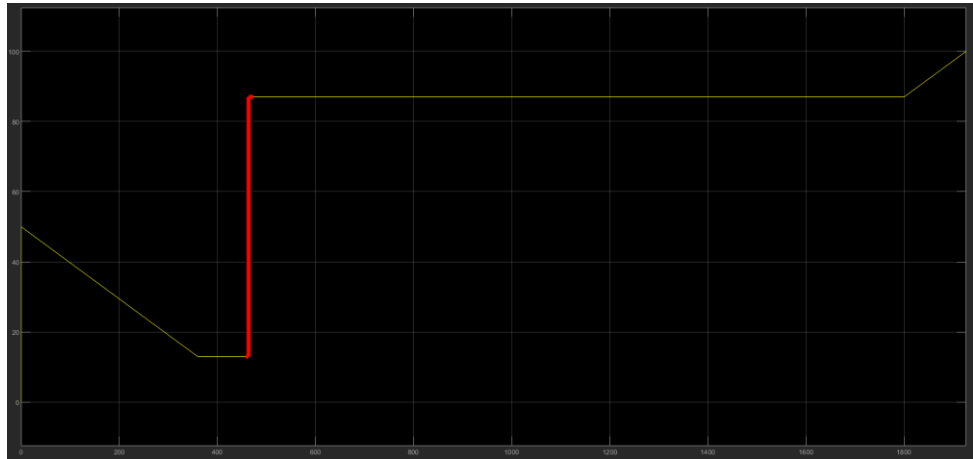


Fig. 71 Jump error obtained by changing between charge and discharge mode (From Simulink model)

So, to solve the issue I tried to insert a Stateflow logic that could track changes in the load current, implementing it to a switch that could allow the model to change freely between charge and discharge taking into account the last registered SOC value.

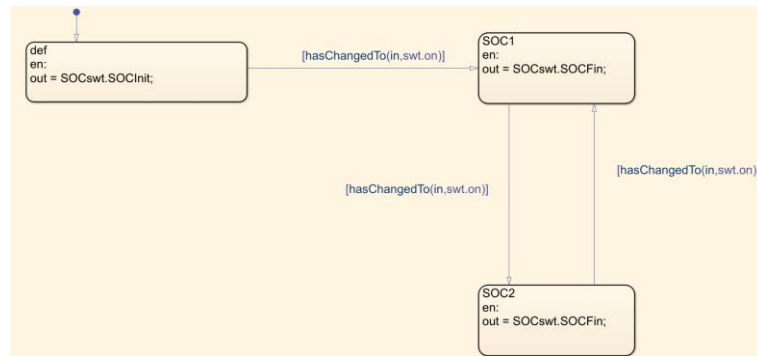


Fig. 72 Stateflow model to indicate the change from the initial and current SOC value in the SOC estimation (From Simulink model)

Unfortunately, not even this solution was enough to eliminate the jump, so I decided to change the SOC estimation method using, in the end, the voltage method.

#### 4.2.2 SOC estimation with Voltage method

The voltage method is, as mentioned above, the simplest method for estimating SOC. Based on the remaining capacity of the battery cell, the voltage method converts the read self-discharge voltage into the equivalent SOC using a known voltage and SOC curve. The main disadvantage of this method is that it is significantly affected by the load current. However, by integrating a LUT that includes the temperature for the calculation of the OCV it's possible to make it more accurate than its standards [23]. The formula used for calculating the SOC is as follows:

$$v_{soc}(t) = \pm \frac{I_{bat}}{C_Q} \cdot t + V_{soc}(0) \cdot e^{\frac{t}{\tau_{sd}}}$$

( 14 )

In which  $\tau_{sd}$  is considered as the time constant of the self-discharge resistance, while  $v_{soc}$  is the voltage of the self-discharge circuit and the SOC per unit (p.u.) [2].

On Simulink the calculation of the SOC with the voltage method has been integrated in this way:

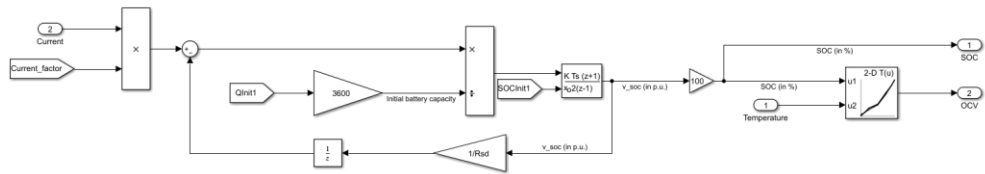
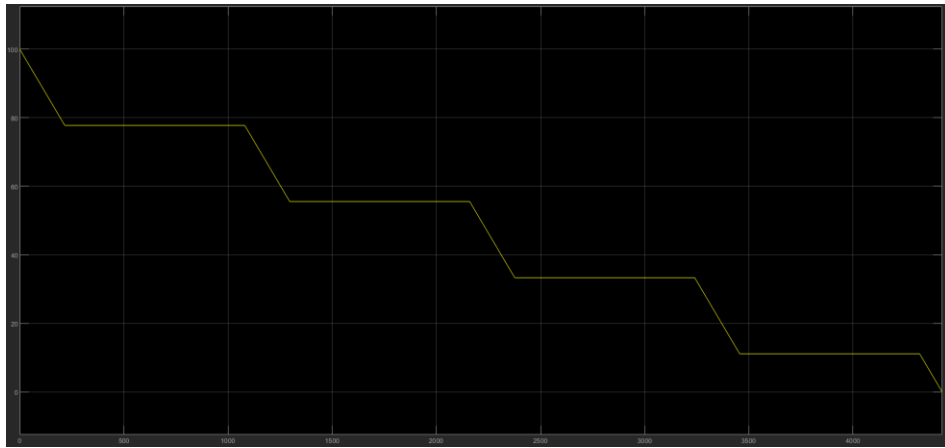


Fig. 73 SOC computation with Voltage method (From Simulink model)

In contrast to the results obtained with the Coulomb counting method, using this block model the SOC can change from the state of discharge to that of charge, and vice versa, without undergoing modifications or jumps. In this way it is possible to simulate real time running, braking (start and stop) and recharging as in reality, although the results are slightly different from the more complex estimation methods of the SOC.



*Fig. 74 SOC graph obtained using the Voltage method (From Simulink model)*

#### **4.2.3 SOC estimation with Extended Kalman Filter**

The Extended Kalman Filter is a method used when the battery system is non-linear and requires linearization. It's an online and dynamic method that needs a suitable model with great computational capabilities and precision. Through this type of estimation, it is possible to estimate the internal states of any dynamic system, including the SOC of a battery, succeeding unlike other methods to also provide any dynamic errors inherent in individual states. Therefore, when constructing the model of the battery pack it will be necessary to include in the description of the states the unknown quantities to calculate, so that they can be estimated through the EKF. Using a model-based state

estimation technique that provides for error correction directly, it is possible to calculate a value of SOC and, if desired, SOH as close as possible to the real value of the battery. With the EKF a linearization process is performed every moment of time through a linear time varying system. This system is implemented to a normal Kalman Filter to obtain the Extended. The Kalman Filter can modify system variables through feedback, reducing inaccuracies between estimated and simulated outputs.

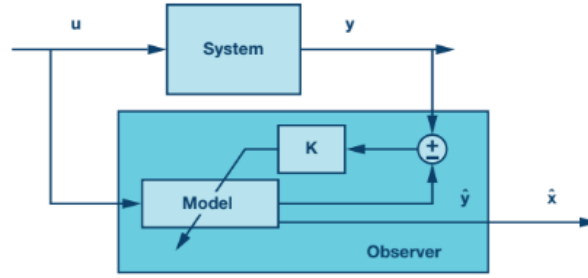


Fig. 75 Kalman filter principle (Murnane, Martin, A closer look at State of Charge (SOC) and State of Health (SOH) estimation techniques for batteries, Norwood, 2021)

Through inputs and outputs taken from the battery model, the EKF can find the minimum mean squared error assuming that the process and sensor noise are independent [23]. Considering an equivalent circuit of Thevenin, we have that the state variables are:

$$x_1(t) = SOC \quad (15)$$

$$x_2(t) = V_{cs} \quad (16)$$

where  $V_{cs}$  is the voltage on the capacitor of the RC branch. Inputs are defined as:

$$y(t) = V_{bat} \quad (17)$$

$$u(t) = I \quad (18)$$

Considering the following equations,  $w$  and  $v$  are the errors caused by the change of  $d$  and  $k$  and represent the disturbances that can occur within the system. In addition, the first two terms are independent.

$$\dot{x} = f(x, u) + w \quad (19)$$

$$y = g(x, u) + v \quad (20)$$

Bearing in mind that

$$x = [x_1, x_2]^T \quad (21)$$

and that the functions  $f(x, u)$  and  $g(x, u)$  are equal to

$$f(x, u) = \left[ \frac{\frac{u}{kC_{cb}}}{-\frac{1}{R_t C_{cs}} + \frac{1}{C_{cs}} u} \right] \quad (22)$$

$$g(x, u) = kx_1 + x_2 + R_i u + d \quad (23)$$

We generate the model presented below by linearizing the two functions for each instant of time with a first order Taylor expansion:



$$\delta \dot{x} = A_k \delta x + B_d \delta u \quad (24)$$

$$\delta y = C_k \delta x + D_k \delta u \quad (25)$$

In which  $A_k$ ,  $B_k$ ,  $C_k$  and  $D_k$  are defined as:

$$A_k = \frac{\delta f(x, u)}{\delta x} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_t C_{cs}} \end{bmatrix} \quad (26)$$

$$B_k = \frac{\delta f(x, u)}{\delta x} = \begin{bmatrix} \frac{1}{k C_{cb}} \\ \frac{1}{C_{cs}} \end{bmatrix} \quad (27)$$

$$C_k = \frac{\delta g(x, u)}{\delta x} = [k \quad 1] \quad (28)$$

$$D_k = \frac{\delta g(x, u)}{\delta u} = R_i \quad (29)$$

The model is discretized as follows, where T is the sampling time and E is the unit matrix. In addition, the individual terms that compose it are also specified.

$$x_{k+1} = A_d x_k + B_d u_k \quad (30)$$

$$y_{k+1} = C_d x_k + D_d u_k \quad (31)$$

$$A_d \approx E + T_c A_k \quad (32)$$

$$B_d \approx T_c B_k \quad (33)$$

$$C_d \approx C_k \quad (34)$$

$$D_d \approx D_k \quad (35)$$

The initialization of the P, Q, and R matrices and observation are the two phases that constitute a significant portion of the EKF estimation. The method forecasts the values of the output, error, and current state before correcting the state [23].

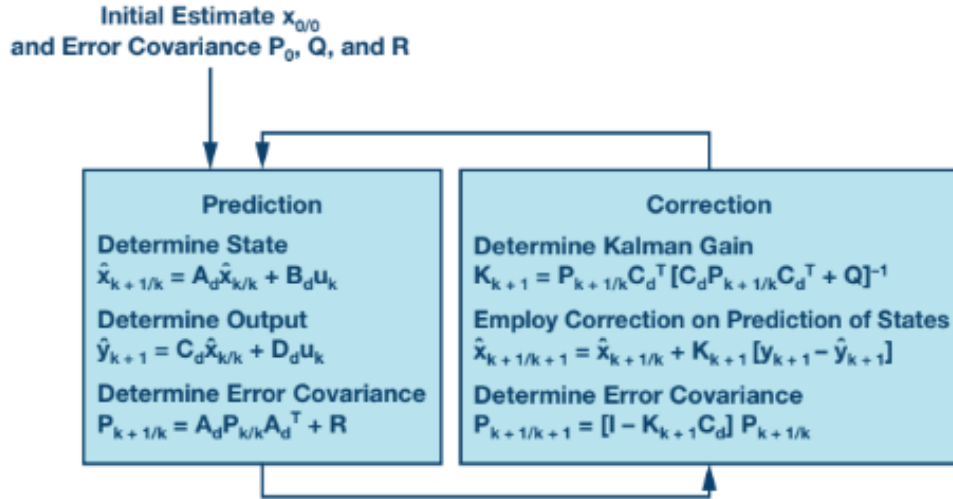


Fig. 76 Kalman filter algorithm (Murnane, Martin, A closer look at State of Charge (SOC) and State of Health (SOH) estimation techniques for batteries, Norwood, 2021)

### 4.3 State of Health (SOH) computation

A battery cell, as has already been clarified above, after cycles of use is subjected to its own degradation. The aging of a battery can also be accelerated under certain conditions, including the operating temperature of the battery pack, ambient temperature, overcurrent, or overvoltage. This is why it is important, in addition to the SOC, to estimate the State of Health (SOH) of a battery, so that the BMS can proceed to implement strategies that can preserve the life of individual cells. Just as with SOC, there are several methods to estimate SOH. Most of these are empirical models that cannot, however, keep a precise track of the aging of batteries that are inside a vehicle [24].

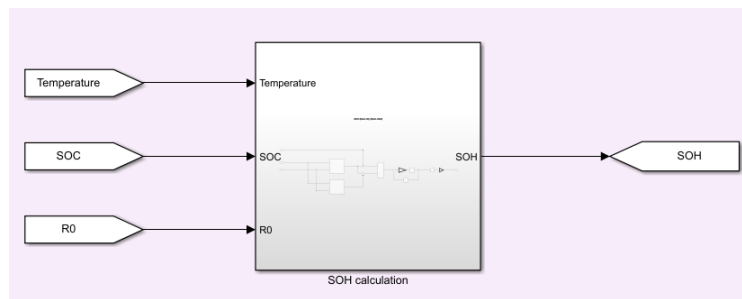


Fig. 77 SOH block in Simulink (From Simulink model)

There are two different categories of aging degradation: calendar aging and cyclic aging. The first represents the processes the vehicle goes through during rest and should not be underestimated considering that it spends most of its time in parking. The second, on the other hand, includes all the aging processes that take place during the charging and discharging cycles, so much so that suppliers provide in the datasheet of the battery cells also an estimate of the number of cycles they can reach [25].

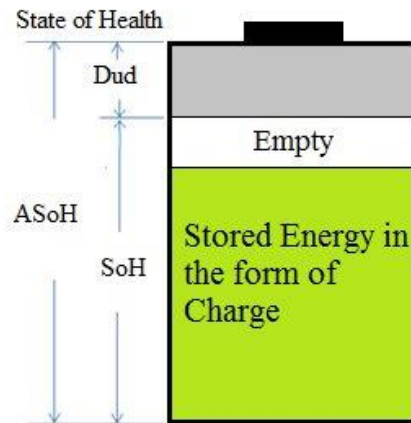


Fig. 78 Representation of battery SOH (Selvabharathi, D., Battery health and performance monitoring system: a closer look at state of health (SoH) assessment methods of a Lead-Acid battery, Thanjavur, 2019)

To be precise, there are three methods for calculating SOH. The first based on direct calibration, the second on the filter and the third on machine learning.

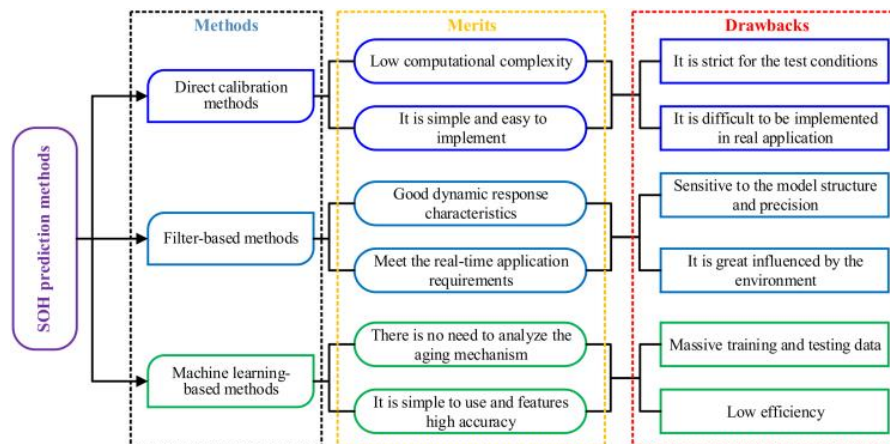


Fig. 79 Classification and comparison of battery SOH prediction methods (Shu, Xing, State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives, Kunming, 2021)

The simplest method, which was used for this thesis work, is the first. Direct calibration includes an estimate of the SOH by measuring the capacity, internal resistance, or energy of the battery cell. The second, instead, uses the already named EKF to calculate the SOH of the battery, through a series of algorithms. The latter uses machine learning algorithms and includes four different phases: offline measurement, feature generation, model training and model validation [26].

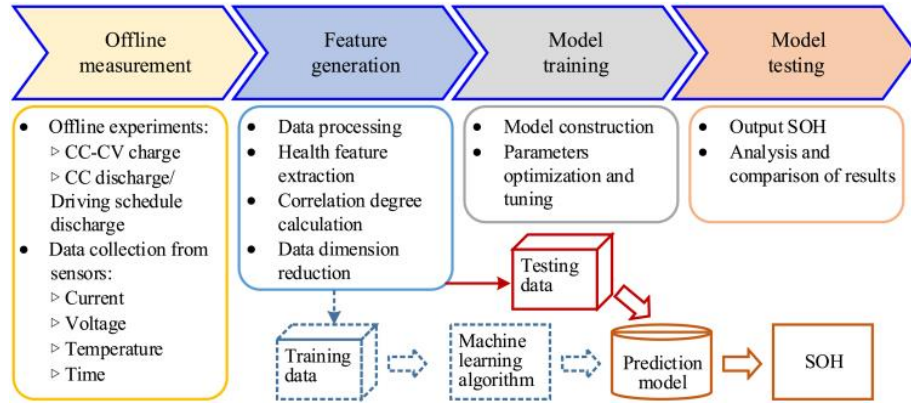


Fig. 80 Generic implementation flowchart for SOH prediction using ML algorithms (Shu, Xing, State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives, Kunming, 2021)

#### 4.3.1 SOH estimation with maximum capacity

A fully charged battery will have a maximum capacity that may be different from that indicated by the supplier. In fact, the older the battery, the more this capacity decreases, not allowing the cell to reach the levels of charge it had when it was completely new [23].

$$SOH = \frac{C_{max}}{C_{rated}} \cdot 100\%$$

(36)

Through the processes of charge and discharge it is possible to find the value of the maximum capacity, also called nominal, knowing the values of the initial and final SOC, as well as the time interval in which these processes take place [25]. From these parameters you can find the capacity as:

$$C_{max} = \frac{1}{SOC(t_f) - SOC(t_i)} \int_{t_i}^{t_f} I(t) dt \quad (37)$$

The higher the accuracy of the current sensor used, the more accurate this method is. In addition, if the difference between the two SOC's is correctly estimated using the OCV, then the accuracy of the model will be higher [25].

#### 4.3.2 SOH estimation with energy loss

Another method for calculating SOH, however, uses energy loss. This approach estimates the energy discharged or charged from the battery using the following computations:

$$\Delta E = \int_{t_i}^{t_f} I(t) \cdot V_{bat} dt \quad (38)$$

$$SOH = \frac{\Delta E}{\Delta E_{BOL}} \cdot 100\% \quad (39)$$

where  $\Delta E$  is the current change in energy and  $\Delta E_{BOL}$  the change in energy at the beginning of battery life. To ensure that the change in energy is measured under

the same conditions as the initial one, this method is often used with the charged energy. Moreover, unlike the previous one, its accuracy depends on both current and voltage sensors [25].

#### 4.3.3 SOH estimation with internal resistance

During the aging of a battery cell its internal resistance changes, increasing, thus leading to the variation of the SOH. In hybrid vehicles, for example, power loss is more important than energy loss. In these cases, the internal resistance to calculate the SOH is then considered, using the formula below:

$$SOH = \frac{R_{EOL} - R_{actual}}{R_{EOL} - R_{BOL}} \cdot 100\%$$

( 40 )

Where  $R_{EOL}$ ,  $R_{actual}$  and  $R_{BOL}$  represent respectively the internal resistance of the battery at the end of its life, at the time of measurement and at the beginning of its life [26]. It is usually possible to take  $R_{EOL}$  equal to twice the initial resistance.

$$R_{EOL} = 2 \cdot R_{BOL}$$

( 41 )

In the Simulink model I used this last approach, inserting LUTs for the calculation of the three resistors depending on the current value of temperature and SOC.

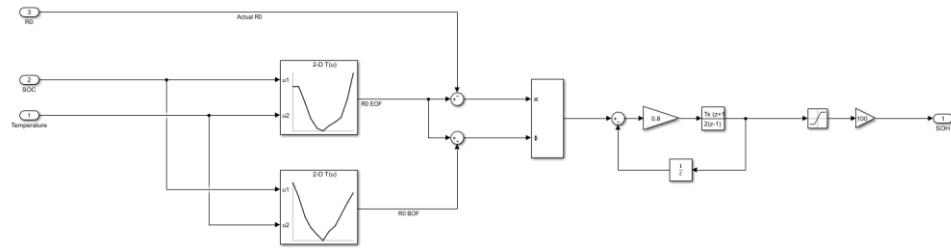


Fig. 81 Computation of SOH with internal resistance (From Simulink model)

I used a PI controller to linearise the SOH calculation. In fact, due to the variation of the resistance values, the graph obtained was too irregular because it changed according to the values of the resistances at different times, thus obtaining different percentages of SOH.

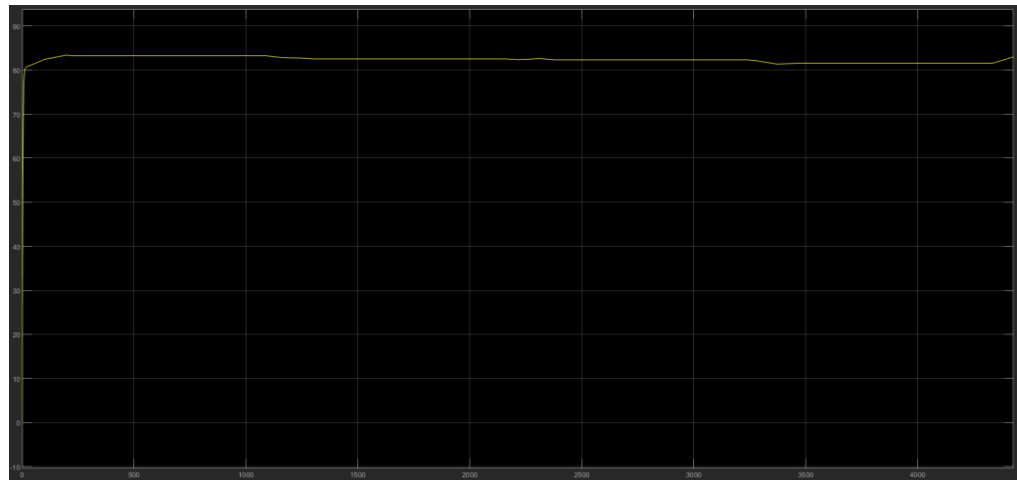


Fig. 82 SOH graph obtained using the internal resistance (From Simulink model)

## 4.4 Thermal model

As has already been addressed in the previous paragraphs, temperature significantly affects the aging, capacity, and efficiency of a battery. This leads to the study of the thermal parameters of a cell, to monitor the temperature variation during its charge and discharge cycles [27]. By studying the heat



generation inside the cell during these processes, it is possible to calculate the temperature both in case it is used alone and in relation to the other cells of the module or battery pack [28]. Heat generation is caused by electrochemical reactions within the cell and its internal resistance. Beyond these intrinsic characteristics, there are other factors that affect the thermal behaviour of a battery, such as temperature, SOC, and current load. Together, these three contribute on the degradation mechanisms that affect the capacity of the cell. For this reason, inside the HEVs and the EVs there is a need to reduce the impact of temperature through a thermal management system, so that the temperature is kept within a precise range. Obviously, depending on the chemistry of the cell there are different values of heat generation [29].

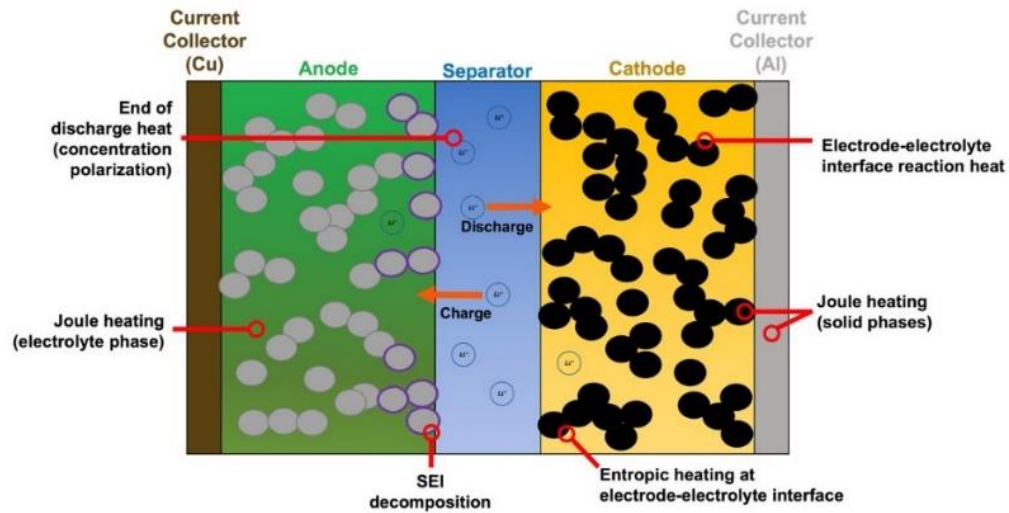


Fig. 83 Heat generation inside a battery cell (Kantharaj, Rajath, *Heat Generation and Thermal Transport in Lithium-Ion Batteries: A Scale-Bridging Perspective*, 2019)

The task of a battery thermal management systems (BTMS) is therefore to keep the cells within the temperature limits set by the manufacturer, minimizing the temperature variation between them and maximizing usable energy. To do this, the BTMS should be designed with a low complexity and so that parasitic losses are at a minimum. Having to study only the behaviour of a battery pack in

relation to its BMS, the thesis work was limited to focusing on the study of the temperature of the cells and not on the feasible solutions to add a cooling system or a heater.

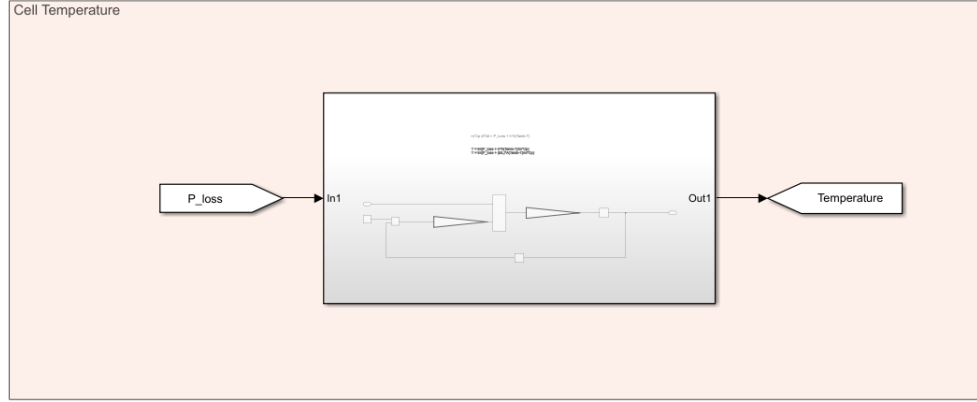


Fig. 84 Cell temperature block in Simulink (From Simulink model)

#### 4.4.1 Temperature variation within a single battery cell

A one-dimensional Simulink model has been developed to predict the temperature trend of a battery cell during its charging and discharging cycles. Energy balancing for a cell can be formulated as:

$$mC_p \frac{dT}{dt} + hA(T_a - T) + \varepsilon A\sigma(T_a^4 - T^4) - \dot{Q}$$

( 42 )

The first term represents the variation of thermal energy within the battery, in particular  $m$  is the mass of the cell and  $C_p$  the heat capacity of the cell. The second, instead, represents the heat loss by convection. In this case  $h$  is the convective coefficient of heat,  $A$  the surface of the cell from where the flow passes,  $T$  the temperature of the cell and  $T_a$  the ambient temperature [30].

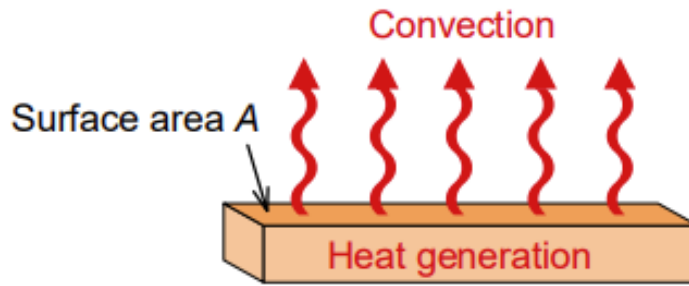


Fig. 85 Heat generation by convection (ROHM, Basics of Thermal Resistance and Heat Dissipation, 2021)

The third represents the heat loss of radiation and is formed by  $\epsilon$  which represents the emissivity,  $\sigma$  the Stefan-Boltzmann constant, the surface, and the temperature difference.

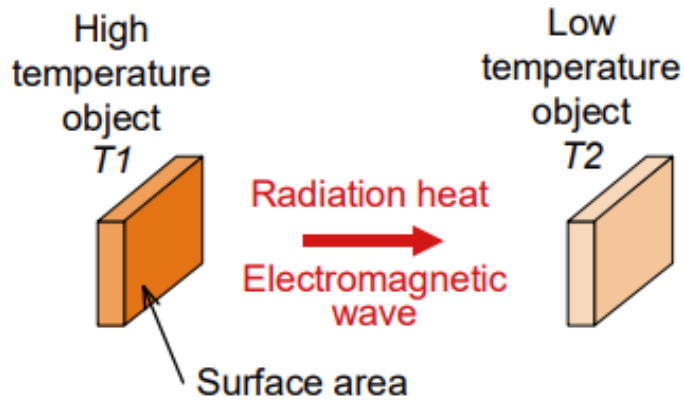


Fig. 86 Heat generation by radiation (ROHM, Basics of Thermal Resistance and Heat Dissipation, 2021)

The last term is the energy dissipation of the cell, calculated in the following ways:

$$\dot{Q} = V_{bat}I \quad (43)$$

$$\dot{Q} = I^2 R_{tot} \quad (44)$$

$$\dot{Q} = \frac{V_{bat}^2}{R_{tot}}$$

( 45 )

The power loss is calculated within the Simulink model directly in the resistor block. In fact, we consider the sum of the energy dissipation that occurs at the head of each resistance of the equivalent circuit of Thevenin, therefore both the internal one and those that are part of the RC branches.

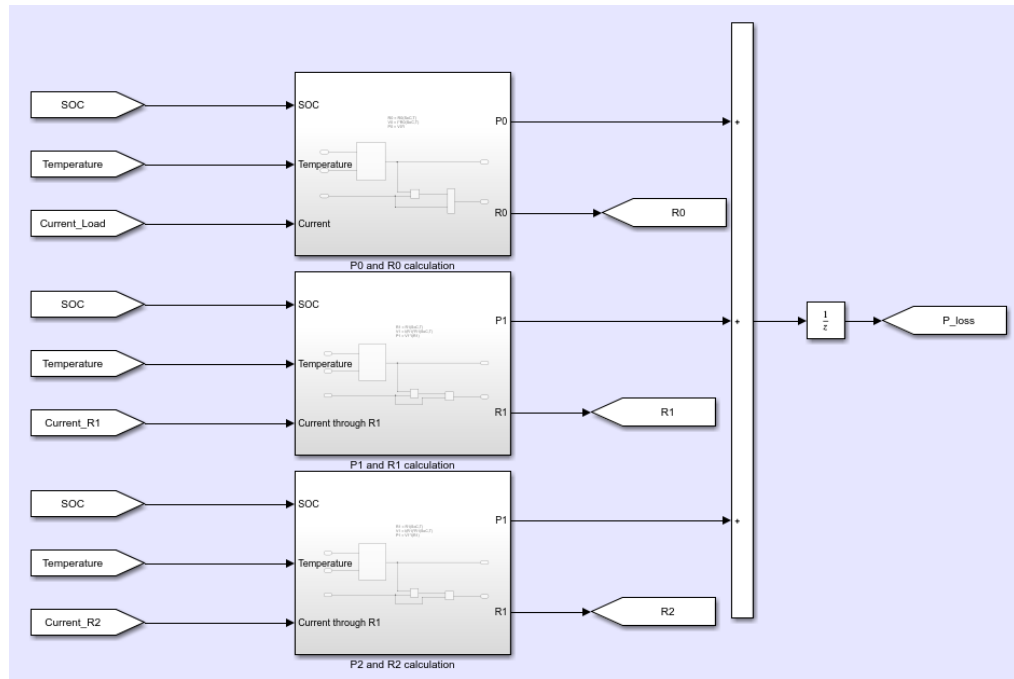


Fig. 87 Power loss computation in Simulink (From Simulink model)

To reduce the complexity of the model, it was decided to use geometric parameters that could adapt to the three architectures used by manufacturers. These are the surface where the heat exchange takes place and the thickness of the cell, identified as the diameter for cylindrical cells.

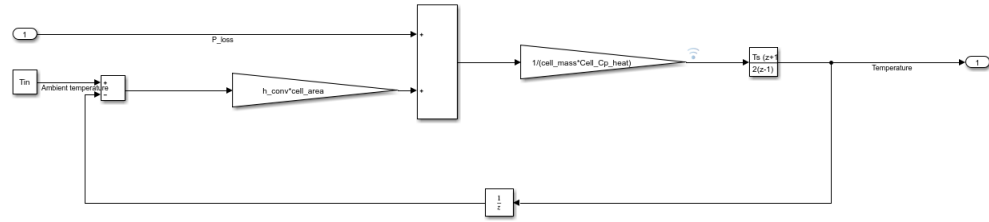


Fig. 88 Temperature calculation of a single battery cell (From Simulink model)

#### 4.4.2 Temperature variation between multiple cells

It's important to think about how the heat produced by each additional cell will affect the others when there are more cells involved. In this instance, the terms that identify the heat loss through conduction will be added to those that were originally introduced using the formula (42).

$$Q_{conduction} = \frac{kA}{L}(T_a - T)$$

( 46 )

In the above formula  $k$  represents the conductive coefficient of heat, while  $L$  is the thickness of the battery cell.

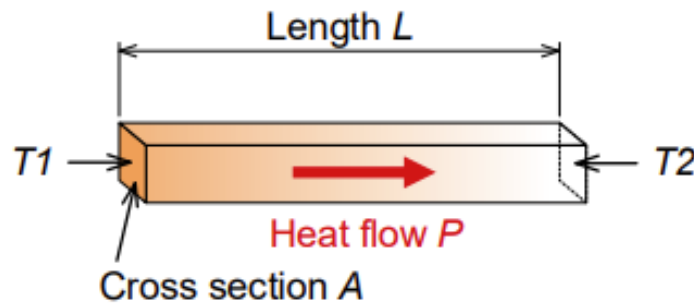


Fig. 89 Heat generation by conduction (ROHM, Basics of Thermal Resistance and Heat Dissipation, 2021)

To track the temperature fluctuation across cells, the model I designed makes advantage of the idea of thermal resistance. Thermal resistance is a property that indicates the resistance of a material to heat flow and can be defined as the reciprocal of conductive and convective terms, without considering the temperature difference.

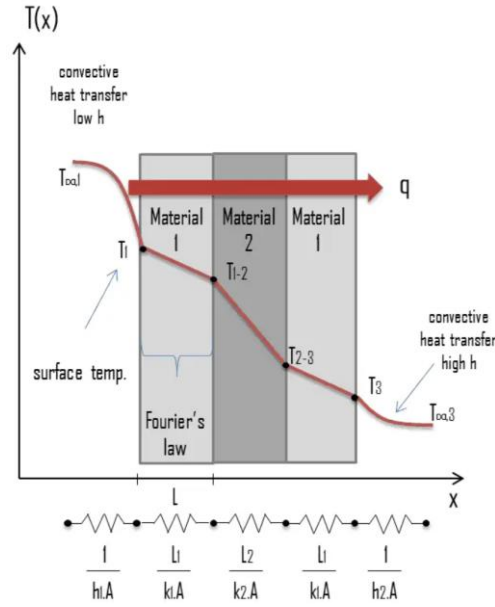


Fig. 90 Example of thermal resistance between multiple materials (<https://www.thermal-engineering.org/what-is-thermal-resistance-thermal-resistivity-definition/>)

Having a total of twelve cells in series in the module, the thermal resistance for cells located after the first can be written as:

$$R_{th} = \frac{1}{hA} + \frac{L}{kA} + \frac{1}{hA}$$

( 47 )

To make the model easier, I considered cascading heat transfer starting from the first cell. Thus, for each subsequent cell, the initial temperature was not considered the ambient temperature, but that of the cell that precedes it.

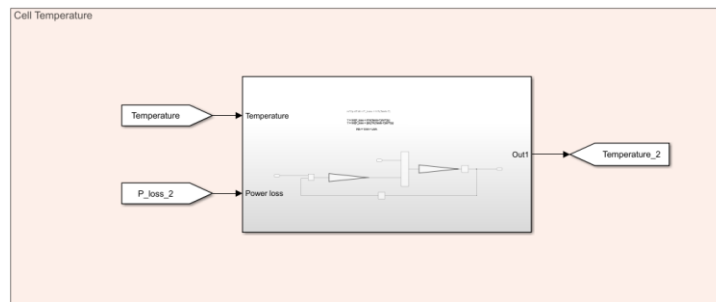


Fig. 91 Cell temperature block in relation to the previous cell in Simulink (From Simulink model)

The thermal model then, from cell 2 to cell 12, has been defined with the following block diagram in the Simulink model:

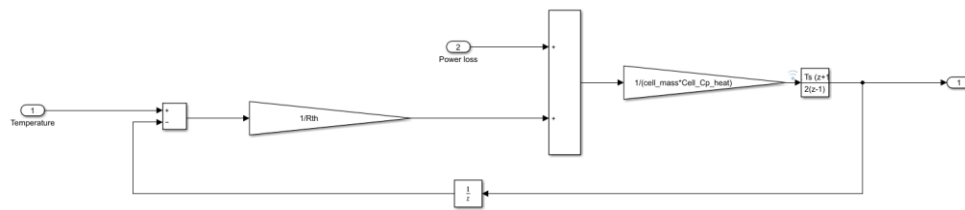


Fig. 92 Calculation of the temperature of a battery cell in relation to the previous (From Simulink model)

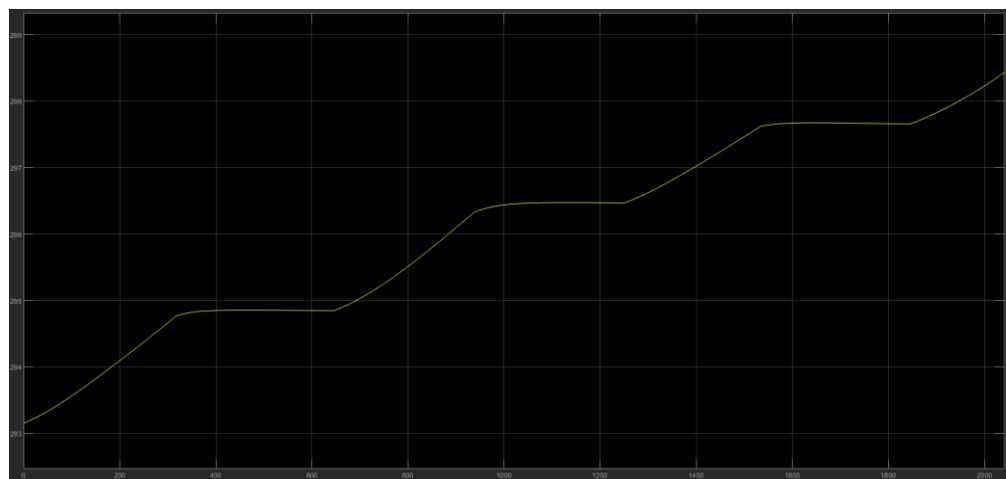


Fig. 93 First cell temperature graph (From Simulink model)

## 5. BMS CONFIGURATION ON HIL

As previously seen, HIL simulators allow you to trick the under-test control unit into making it feel like it's inside the vehicle. To do this you must first start with the study of the napkin of the control unit, to identify the channels that can be used and devise a wiring harness so that you can prepare a suitable wiring to connect the control unit to the HIL simulator. So, it is necessary to identify the control unit used and the type of I/O boards inside the HIL simulator, in my case belonging to the dSpace company. The information in this chapter has been taken from the datasheets of the TLE9015DQU [31] and TLE9012DQU [32] units of Infineon.

### 5.1 Specifications of the ECU under test

The control unit I chose is produced by Infineon and is composed of a transceiver board TLE9015DQU, an application kit TC265 TFT and more boards TLE9012DQU, which deal with the management of cell modules.

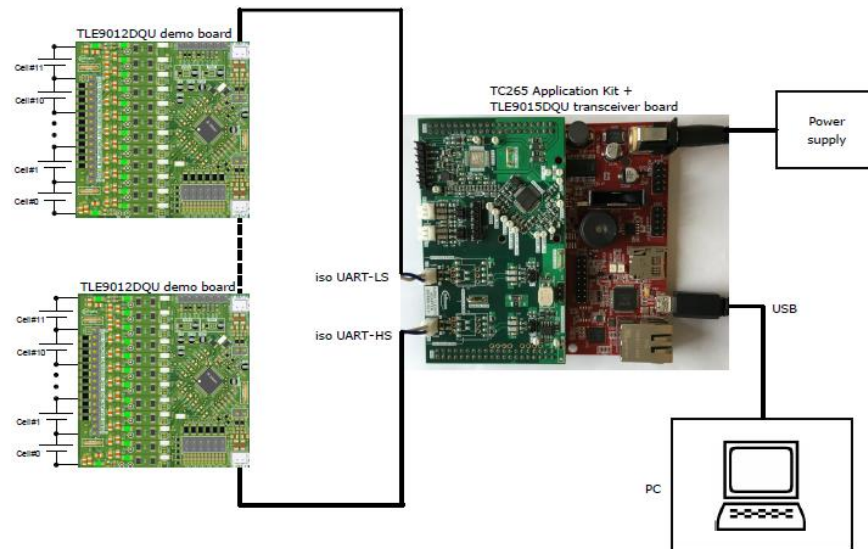
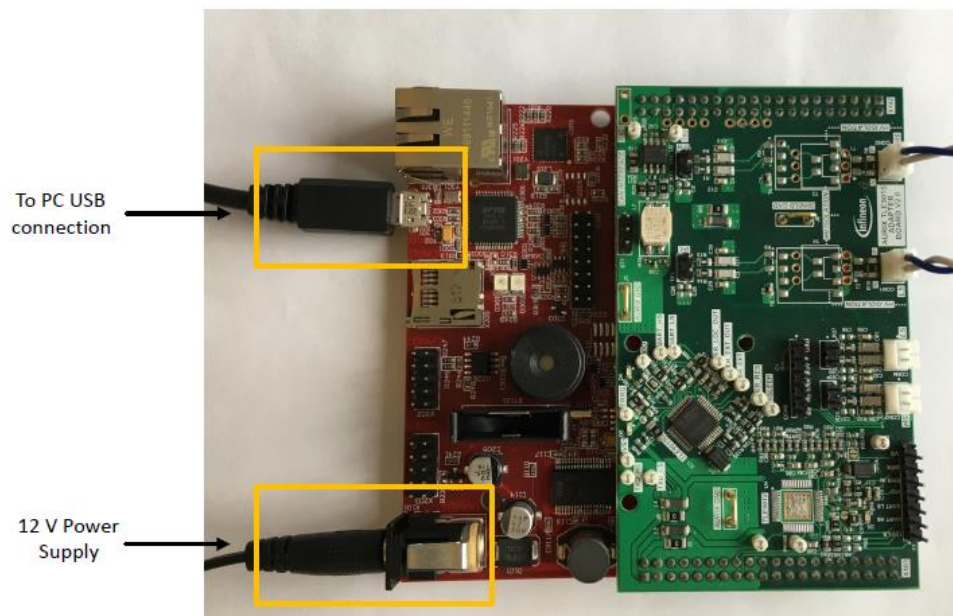


Fig. 94 Demo kit BMS (Infineon Technologies AG, TLE9012DQU, TLE9015DQU - User manual, Munich, 2022)



### 5.1.1 TLE9015DQU transceiver board

A transceiver is capable of acting both as a receiver and a transmitter. In this case, the TLE9015DQU unit is used to communicate via the iso UART modules with the integrated circuits (ICs) that manage the individual cells. The board must necessarily be connected to a 12V power supply, such as the one integrated within the dSpace simulator. The board must necessarily be connected to a 12V power supply, such as the one integrated within the dspace simulator. In addition, in order for the control unit software to be flashed, it is mandatorily to connect the transceiver to a PC Host through a USB or Ethernet cable.



*Fig. 95 AURIX power supply and communication connection (Infineon Technologies AG, TLE9012DQU, TLE9015DQU - User manual, Munich, 2022)*

The control unit can be flashed both through Infineon tools and through INCA, a software used for the measurement, calibration and diagnosis of DUT that occurs with the use of ETAS devices.

### 5.1.2 TLE9012DQU cell management circuit

The TLE9012DQU unit, instead, has the task of monitoring and managing the cells belonging to a single module of the battery pack. This board can handle up to twelve cells connected in series, being able to accurately calculate both the SOC and the SOH of the elements belonging to the module to which it is connected. The capabilities of the DUT allow it to be used in HEV, battery electric vehicles (BEV) and plug-in hybrid vehicles (PHEV) and, of course, in 12 V batteries.

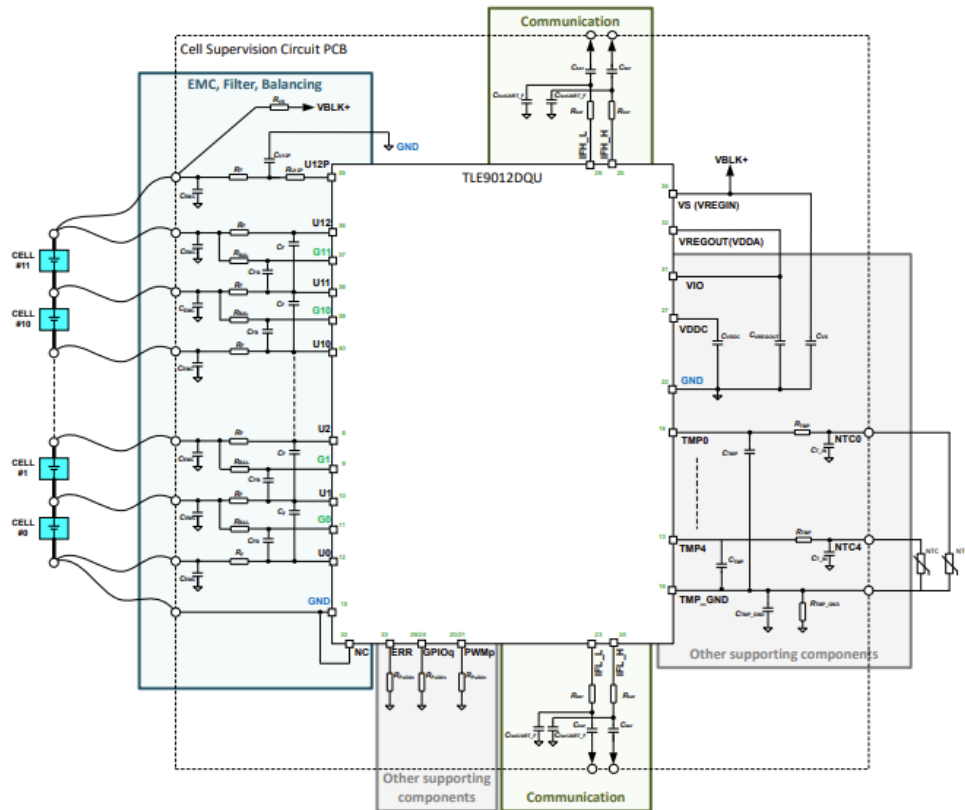


Fig. 96 TLE9012DQU and external circuitry (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The board is in turn composed of several units, which depending on their functions deal with various aspects of cell management. In the image above we

can see both the board and external circuits connected to it and recommended for its use. Alternatively, the following electrical schematic can be used to summarize the board's interior circuit.

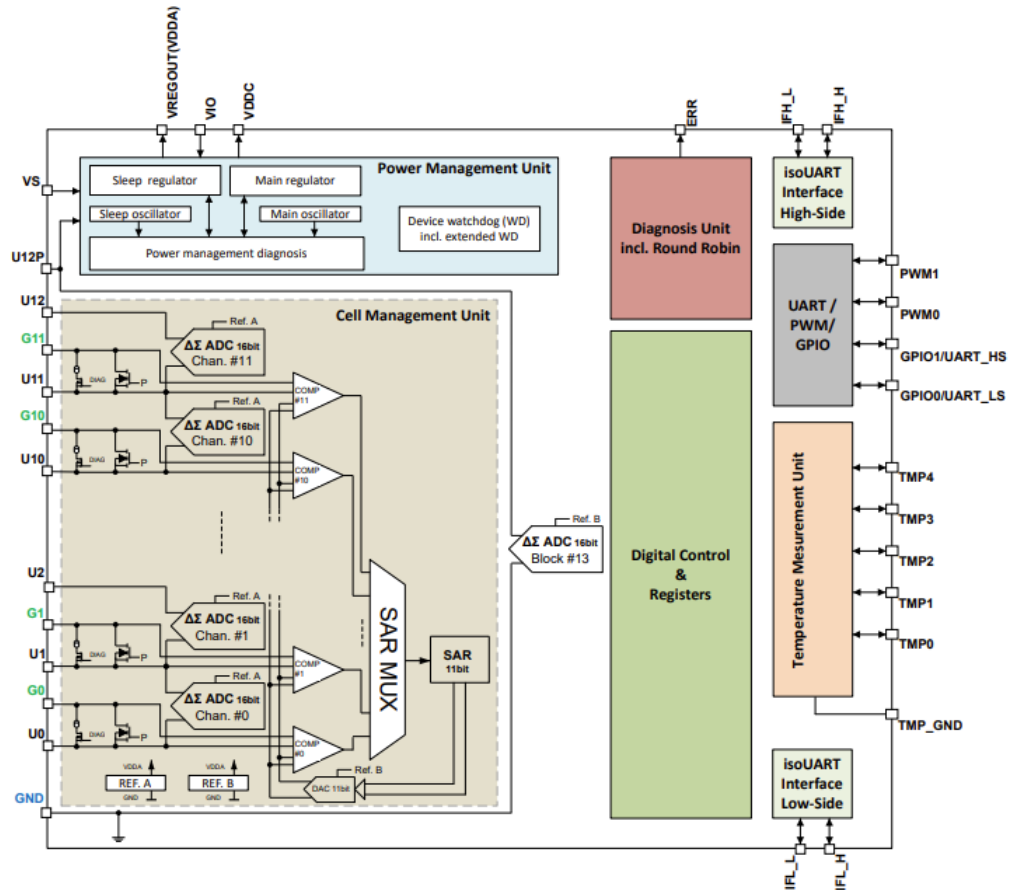


Fig. 97 IC block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

Each pin of the electrical unit must be well understood in order to comprehend the channels that may be used with the simulator.

### 5.1.3 Cell Management Unit

The Cell Management Unit consists of the following pins:

- Un: they are pins with Analog In functions that measure the voltage of the cells at their terminals.
- Gn: unlike *Un* pins, they act as both Analog In and Analog Out. They handle cell balancing.
- GND: generic ground.

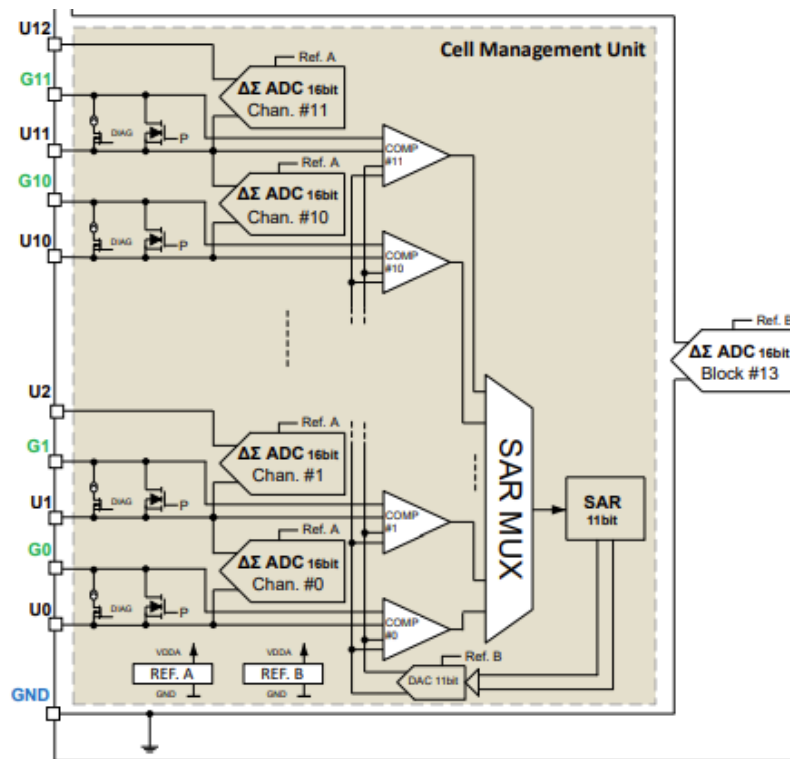


Fig. 98 Cell management unit block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The selected control unit uses several voltage measurement modules, which can also be activated manually via the communication interface. When triggered the measurement sets a lock bit that prevents any other cyclic measurement, until the device deletes it at the end of the measurement. Through the

configuration of the control unit through the Infineon tools, you can set the start bit to initialize the voltage measurement. The result obtained by the measurement is the average of the cell voltage in a precise time interval, always set by software. After completing the measurement, the starting bit is also cleared. The control unit also offers a long-running mode, where the results page is refreshed only after the conclusion of the measurement.

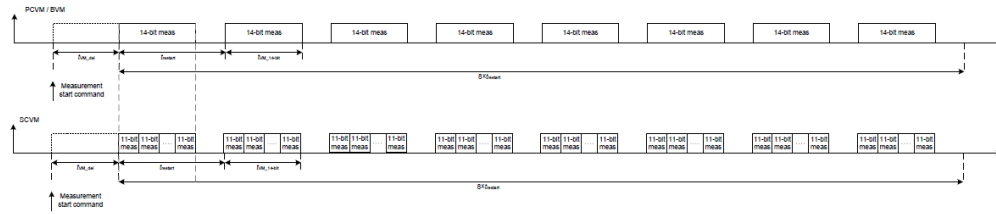


Fig. 99 Voltage measurement long-running mode (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The different measurements are as follows:

- Primary cell voltage measurement (PCVM): is an internal unit of the ECU that allows to measure the voltage detected in each cell through the pins  $Un$ .
- Secondary cell voltage measurement (SCVM): is an additional measurement type that the device permits, making it feasible to monitor the voltage of at least one cell simultaneously with PCVM. It is essential that at least one cell be set up with this kind of measurement.
- Block voltage measurement (BVM): allows the IC to measure the sum of all the voltages of the connected cells through the pins  $Un$ .
- Auxiliary voltage measurement (AVM): is an auxiliary measurement that can be, like the BVM, measured simultaneously to the PCVM and

the SCVM. However, it can only be used through the *TMP* pins used for temperature sensors. When proceeding with AVM, the temperature measurement is interrupted to allow the voltage measurement. Despite the multiple *TMP* pins, you can only proceed to AVM sequentially.

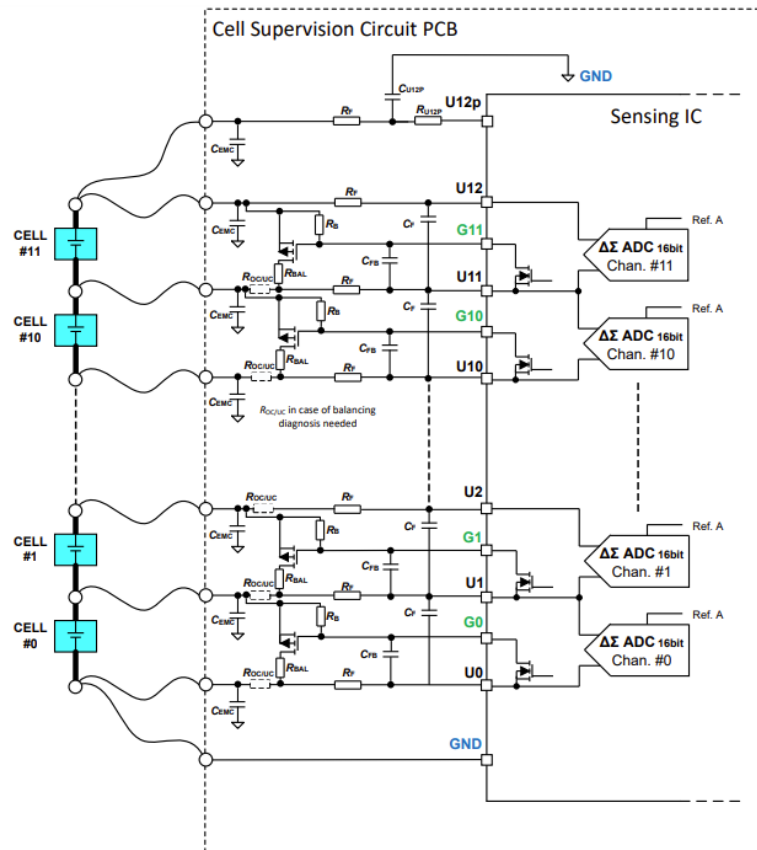


Fig. 100 External balancing device (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The device, as already mentioned, supports the balancing of each cell through *Gn* pins. The circuits indicated in the preceding picture should be used between the board's pins *Un* and *Gn* and the two terminals of each cell to apply the balancing of the battery cells, according to the datasheet for the electronic device. To activate the balancing, a start bit is recorded for all cells that, through the circuit, activates a passive balancing. Moreover, the balancing is stopped

during the PCVM, SCVM, and BVM measurements so that it cannot be used to manipulate the voltage. The balancing automatically terminates when the bit PBOFF is active.

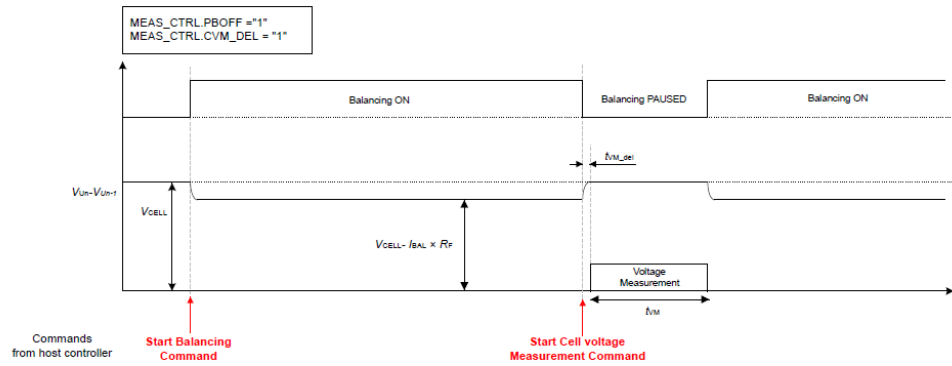


Fig. 101 Balancing and cell voltage measurement (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The *PWM0* and *PWM1* pins allow for the setting of a balance that is actuated by a PWM. It is possible to customize this feature such that it operates while the PWM is on-time and is inactive when it is off-time using the control interface. This kind of balancing is overridden by any other command, such as voltage measurement, and is interrupted by it.

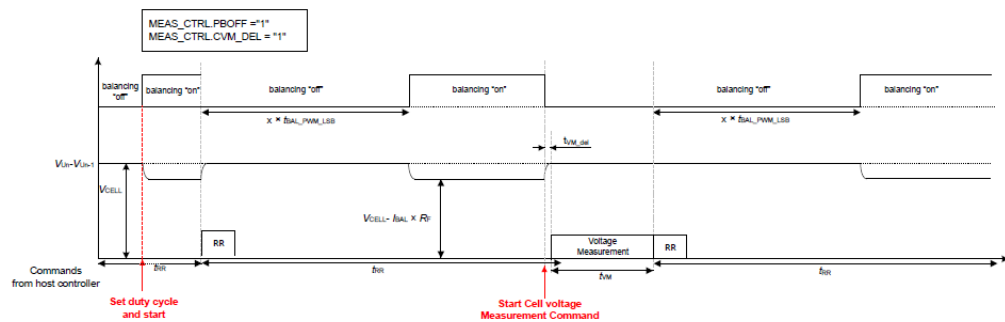


Fig. 102 PWM balancing function (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The IC also provides an overcurrent and undercurrent diagnostic function, which can be used by inserting an  $R_{OC/UC}$  resistor inside the external circuit. The control unit starts balancing by turning on the transistors used as switches through Gn pins, which are utilized as Analog Out. The IC supports up to 200 mA of balancing current, obtained directly from the voltage at the terminals of the cell so that a value equal to

$$V_{bal} = I_{bal} \cdot R_F$$

( 48 )

is subtracted from it.

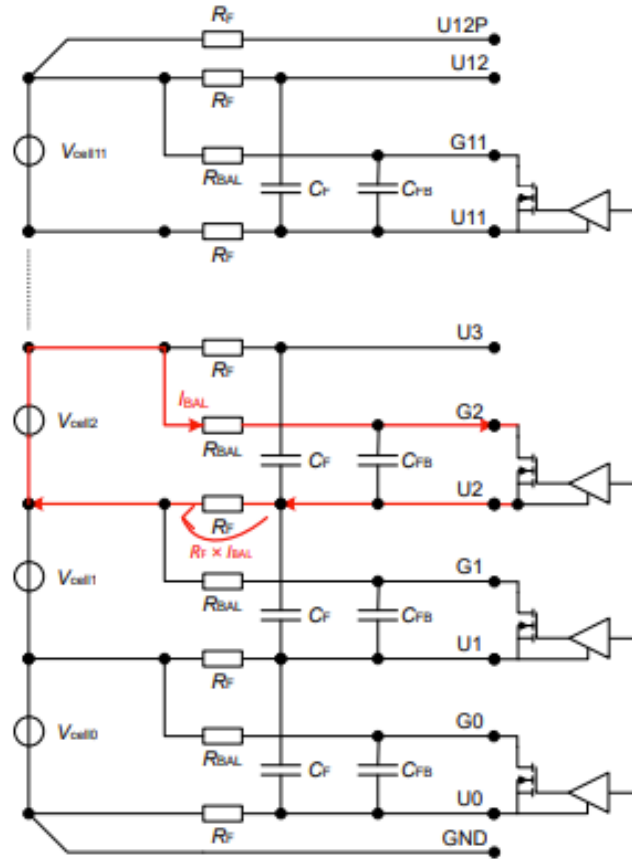


Fig. 103 Passive balancing (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)



#### 5.1.4 Power management unit

The TLE9012DQU Unit has an internal power supply system that includes  $VS$ ,  $U12P$  and  $GND$  pins. To power the IC, then, is drawn energy directly from the module of cells connected to it, so as to generate an internal voltage of supply and output voltages  $V_{VDDC}$  and  $V_{REGOUT}$ .

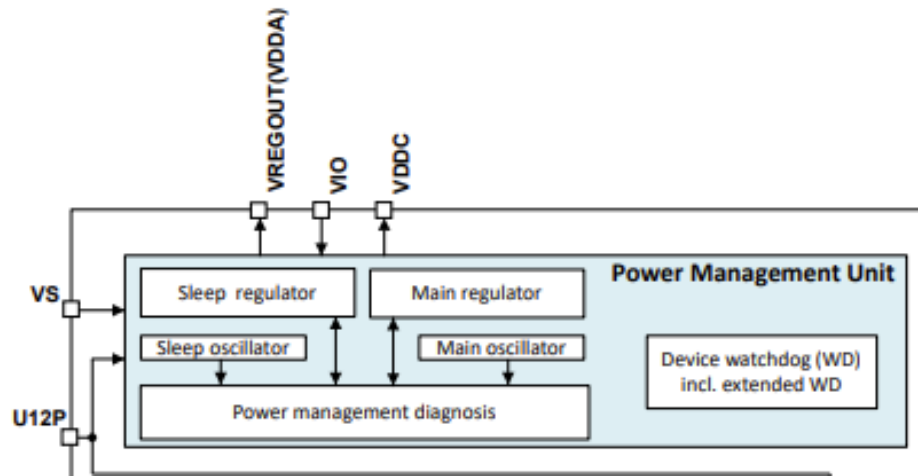


Fig. 104 Power management unit block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The output pins require a capacitor connected to the ground, moreover it is to specify that no supply current is absorbed by the voltage measurement pins. Through a configurable bit, you can start a sleep mode inside the unit that helps to reduce energy consumption through the  $U12P$  and  $GND$  pins. The  $V_{VDDC}$  pin is used to power the communication interface, while the  $V_{REGOUT}$  pin can be connected to the  $VIO$  pin to power the GPIO of the device should it be used. The latter can also be connected to an external voltage generator.

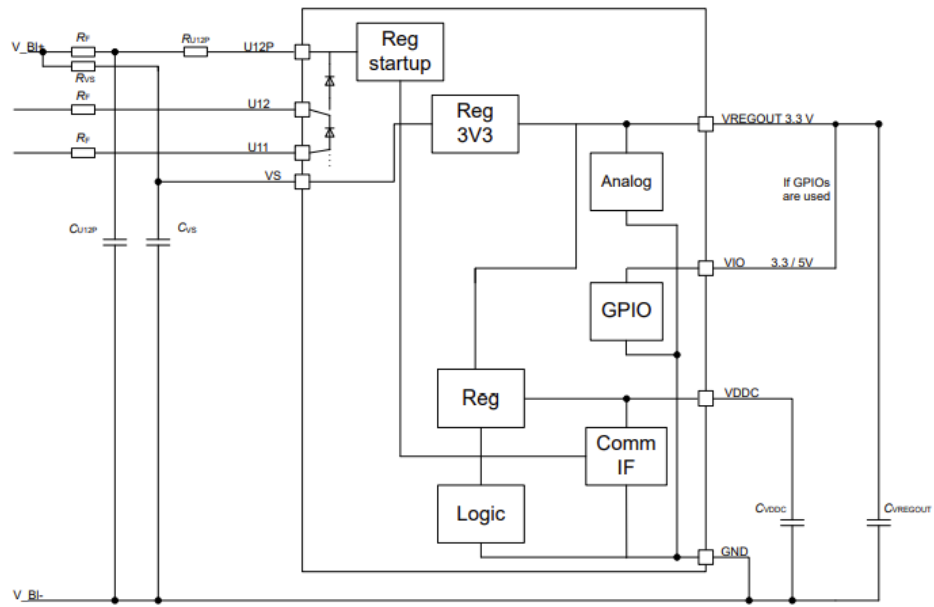


Fig. 105 Typical power supply configuration using the internal voltage regulator (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

### 5.1.5 Temperature measurement unit

Up to five different values from five external NTC thermistors can be measured through the pins of the temperature measurement unit. The device also includes two internal temperature sensors that together with the others help to record the temperature values.

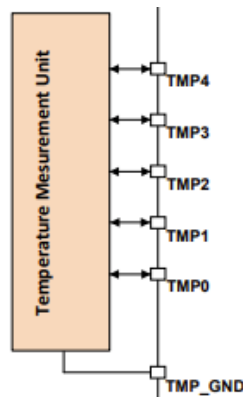


Fig. 106 Temperature measurement unit (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

Unused measurement channels must be disabled through the host computer if they are not all necessary. These can be used, as mentioned above, for AVM measurements. Starting from the *TMP0* pin, the channels must be connected consecutively. The measurements are updated every three round robin cycles.

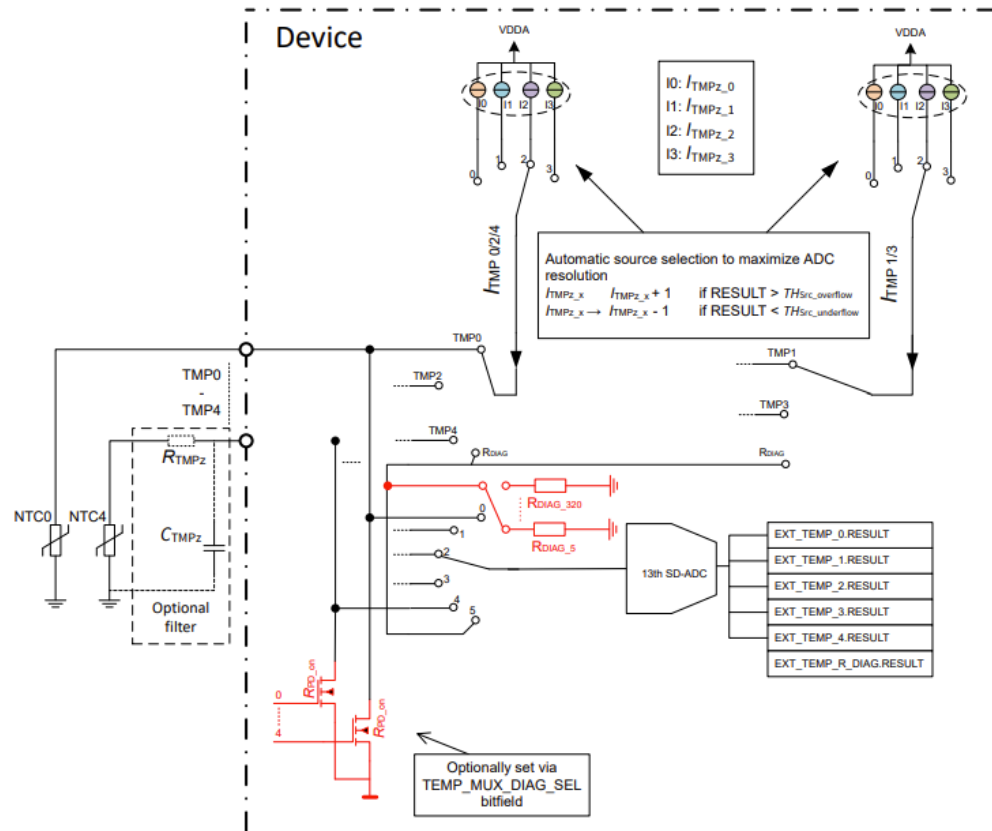


Fig. 107 External temperature measurement (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

For measurement the IC supplies up to four current sources and recognizes which of them is the best to utilize for each NTC channel. The obtained results list the perceived temperature, the current applied, the bits required to acquire new data, and any errors as information. The control unit uses a round robin diagnostic to make sure the temperature unit is operating properly. One *TMP* channel is only examined each round robin cycle.

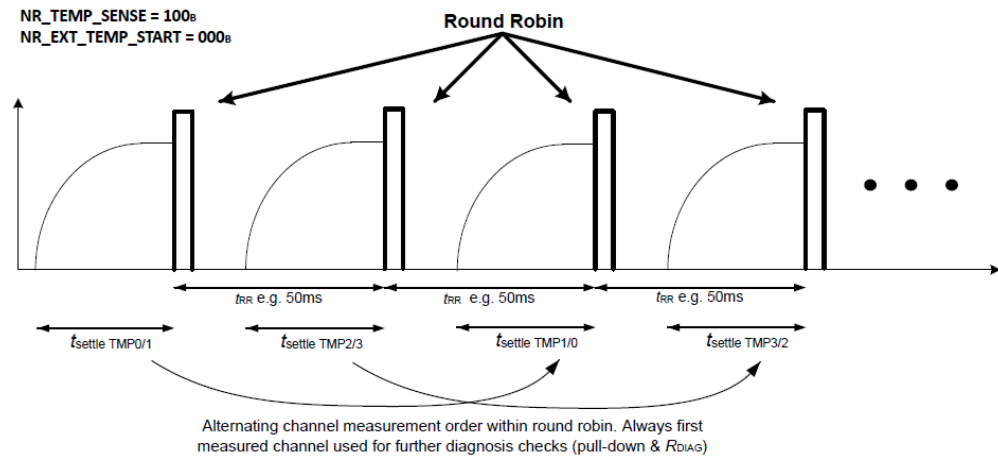


Fig. 108 TMP triggering (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

### 5.1.6 Communication interface

The device supports UART and iso UART communications, allowing users to link various devices to one another.

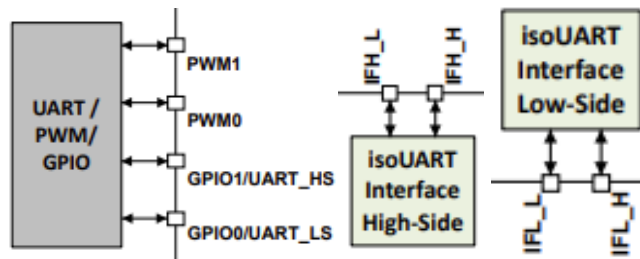


Fig. 109 Communication interface block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

Several ICs can be connected using various configurations:

- Direct UART connection for low voltage applications.
- Primary on bottom communication with EMM function.
- Primary on top communication with EMM function.
- Ring communication with EMM function.

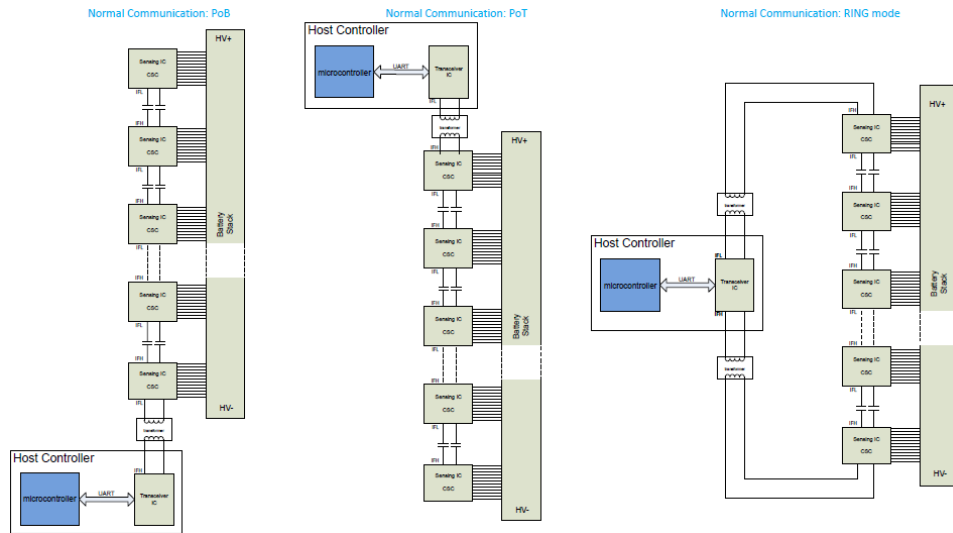


Fig. 110 Communication configurations (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The BMS configures the UART interfaces so that communication between the different ICs follows a precise direction during the wake-up cycle, which can only be changed by putting the control unit to sleep. Without using the control unit's *ERR* pin, error signals can also be sent via this connection.

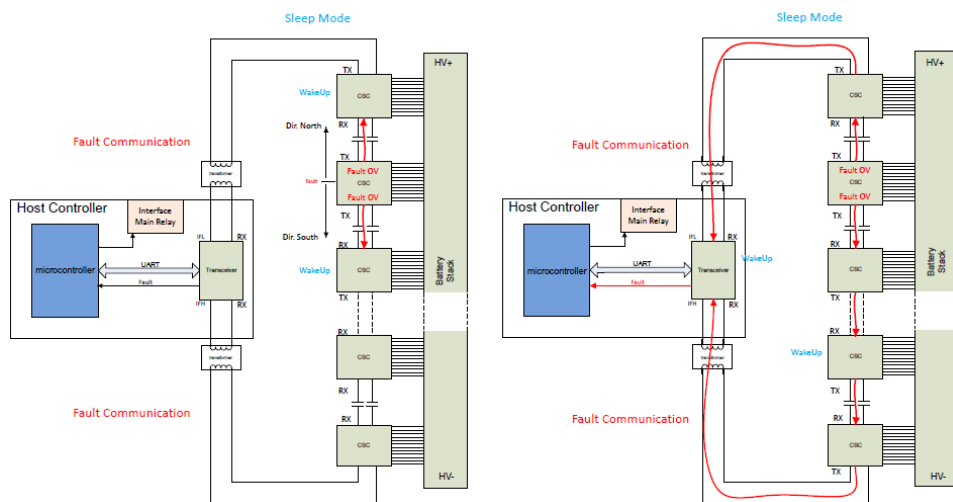


Fig. 111 Fault communication (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

## 5.2 Configuration between ECU and simulator

After carefully examining the device, and all the pins inside the control unit have been identified and their functions explained, users may move on to create an appropriate configuration between it and the HIL simulator being used. The configuration was built to connect the channels offered by the MultiCompact I/O DS2680 boards with the pins of the IC since the simulator made available for the thesis work by Kineton S.r.l. contained these boards.

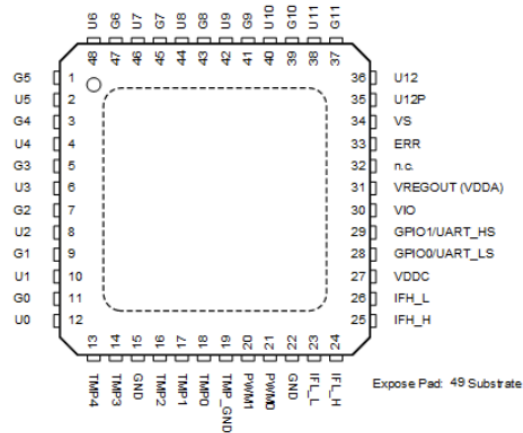


Fig. 112 Pin configuration (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022)

The first step to perform the configuration is to draw up a wiring harness that can guide to the construction of a suitable wiring to connect the BMS with the simulator.

### 5.2.1 Wiring Harness

When I created the wiring harness, I made decisions about which pins to use or not utilize as a connection between it and the simulator based on the information from the ECU datasheet. The TLE9015DQU is only able to have one supply

channel, which will be set up as a Power Switch so that the simulator's Power Supply may deliver a voltage of 12 V to the BMS.

Used Not Used	Short Description	Real Component	ECU Pin	Signal Description	In/Out looking at HIL	dSpace I/O Type
Used	G5	IC	1	Cell balancing	In	Analog In 1
Used	U5	IC	2	Cell voltage measurement	Out	Analog Out 1
Used	G4	IC	3	Cell balancing	In	Analog In 1
Used	U4	IC	4	Cell voltage measurement	Out	Analog Out 1
Used	G3	IC	5	Cell balancing	In	Analog In 1
Used	U3	IC	6	Cell voltage measurement	Out	Analog Out 1
Used	G2	IC	7	Cell balancing	In	Analog In 1
Used	U2	IC	8	Cell voltage measurement	Out	Analog Out 1
Used	G1	IC	9	Cell balancing	In	Analog In 1
Used	U1	IC	10	Cell voltage measurement	Out	Analog Out 1
Used	G0	IC	11	Cell balancing	In	Analog In 1
Used	U0	IC	12	Cell voltage measurement Ref	Out	Analog Out 1 Ref
Used	TMP4	IC	13	Temperature sensor	Out	Resistance Out 1
Used	TMP3	IC	14	Temperature sensor	Out	Resistance Out 1
Used	GND	IC	15	Ground	Ref	GND
Used	TMP2	IC	16	Temperature sensor	Out	Resistance Out 1
Used	TMP1	IC	17	Temperature sensor	Out	Resistance Out 1
Used	TMP0	IC	18	Temperature sensor	Out	Resistance Out 1
Used	TMP_GND	IC	19	Temperature sensor reference	Ref	Resistance Out 1_Ref
Not Used	PWM1	IC	20	PWM I/O	Ref	GND
Not Used	PWM0	IC	21	PWM I/O	Ref	GND
Used	GND	IC	22	Ground	Ref	GND
Used	IFL_L	IC	23	Lower isolated UART L pin		COM
Used	IFL_H	IC	24	Lower isolated UART H pin		COM
Used	IFH_H	IC	25	Upper isolated UART H pin		COM
Used	IFH_L	IC	26	Upper isolated UART L pin		COM
Not Used	VDDC	IC	27	Buffer	ECU	Communication Interface - Out from ECU
Not Used	GPIO0/UART_LS	IC	28	General-purpose input/output	Ref	GND
Not Used	GPIO1/UART_HS	IC	29	General-purpose input/output	Ref	GND

Not Used	VIO	IC	30	Supply for GPIO interface	ECU	VREGOUT - 3.3/5 V (If GPIO is used)
Not Used	VREGOUT	IC	31	Output pin internal regulator	ECU	VIO - 3.3V
Not Used	n.c.	IC	32		Ref	GND
Not Used	ERR	IC	33	Error output to microcontroller		
Used	VS	IC	34	Supply pin of internal regulator	Out	Analog Out 1
Used	U12P	IC	35	Positive supply pin	Out	Analog Out 1
Used	U12	IC	36	Cell voltage measurement	Out	Analog Out 1
Used	G11	IC	37	Cell balancing	In	Analog In 1
Used	U11	IC	38	Cell voltage measurement	Out	Analog Out 1
Used	G10	IC	39	Cell balancing	In	Analog In 1
Used	U10	IC	40	Cell voltage measurement	Out	Analog Out 1
Used	G9	IC	41	Cell balancing	In	Analog In 1
Used	U9	IC	42	Cell voltage measurement	Out	Analog Out 1
Used	G8	IC	43	Cell balancing	In	Analog In 1
Used	U8	IC	44	Cell voltage measurement	Out	Analog Out 1
Used	G7	IC	45	Cell balancing	In	Analog In 1
Used	U7	IC	46	Cell voltage measurement	Out	Analog Out 1
Used	G6	IC	47	Cell balancing	In	Analog In 1
Used	U6	IC	48	Cell voltage measurement	Out	Analog Out 1

Table 2 Wiring harness

Within, the channel and its Hypertac pin that will be linked to the simulator have been specified, in addition to the number that identifies each pin of the ECU, in order to move forward clearly with the wiring construction. All the  $G_n$  pins that deal with the balancing have been set as Analog In 1, so that through the simulated circuit on Simulink it is possible to proceed with the balancing of each cell. On the other hand, the  $U_n$  pins have been configured as Analog Out 1, so that the control unit can receive the voltage at the terminals of the simulated cells as a signal. Since these pins are used both as positive and negative terminals,  $U0$  is configured as the voltage reference of the first cell sent through the connection with  $U1$ . All the channels used for temperature sensors are set as Resistance Out 1. Since there are only five pins for temperature measurement, I decided to connect the calculated values for the



most significant cells of the module, namely the first, fourth, the seventh, tenth and twelfth. *VS* and *UI2P* are set as Analog Out 1 and for them the voltage of the entire module is taken. In total the following channels of the DS2680 board were occupied:

Channel	Used	Total	Available
Analog Out 1	14	15	1
Resistance Out 1	5	12	7
PWM Out	0	28	28
Analog In 1	11	20	9
Power Switch	1	6	5

*Table 3 Used, total and available channels*

The other channels, apart from the communication channels, are not considered to connect the unit to the simulator. Hypertac channels and pins have been identified using both dSpace documentation and ConfigurationDesk.

Used Not Used	Short Description	Real Component	ECU Pin	Ch.	HYP1	HYP2	HYP3
Used	G5	IC	1	1	A1		
Used	U5	IC	2	3			A7
Used	G4	IC	3	2	A3		
Used	U4	IC	4	4			B5
Used	G3	IC	5	3	B1		
Used	U3	IC	6	5			B7
Used	G2	IC	7	4	B3		
Used	U2	IC	8	6			C5
Used	G1	IC	9	5	C1		
Used	U1	IC	10	7			C7
Used	G0	IC	11	6	C3		
Used	U0	IC	12	8			C8
Used	TMP4	IC	13	1		A15	
Used	TMP3	IC	14	2		A17	
Used	GND	IC	15				

Used	TMP2	IC	16	3		B15	
Used	TMP1	IC	17	4		B17	
Used	TMP0	IC	18	5		C15	
Used	TMP_GND	IC	19	1		A16	
Not Used	PWM1	IC	20				
Not Used	PWM0	IC	21				
Used	GND	IC	22				
Used	IFL_L	IC	23				
Used	IFL_H	IC	24				
Used	IFH_H	IC	25				
Used	IFH_L	IC	26				
Not Used	VDDC	IC	27				
Not Used	GPIO0/UART_LS	IC	28				
Not Used	GPIO1/UART_HS	IC	29				
Not Used	VIO	IC	30				
Not Used	VREGOUT	IC	31				
Not Used	n.c.	IC	32				
Not Used	ERR	IC	33				
Used	VS	IC	34	9			D5
Used	U12P	IC	35	10			D7
Used	U12	IC	36	11			E5
Used	G11	IC	37	7	D1		
Used	U11	IC	38	12			E7
Used	G10	IC	39	8	D3		
Used	U10	IC	40	13			A9
Used	G9	IC	41	9	E1		
Used	U9	IC	42	14			A11
Used	G8	IC	43	10	E3		
Used	U8	IC	44	15			A13
Used	G7	IC	45	11	A5		
Used	U7	IC	46	1			E1
Used	G6	IC	47	12	A7		
Used	U6	IC	48	2			E3

Table 4 Wiring harness with DS2680 channels

### 5.3 dSpace Toolchain

To run real-time simulations, the SCALEXIO system must be implemented with a real-time application and controlled by a PC host. The PC host runs the software tools that work with the SCALEXIO system over Ethernet. The PC host not only creates real-time applications that model the controlled system, but also implements and manages the application on the SCALEXIO simulator. In addition, the host PC can modify the parameter values and measured variables of the real-time application. Faults in the wiring of ECU are simulated by the host PC by controlling the simulation of faults. The software tools are divided into three phases:

- Implementing, software tools and blocks for modelling and implementing real-time applications:
  - Matlab/Simulink, which is used to model only the behaviour of the controlled system model.
  - ConfigurationDesk, which is used to implement I/O functions in the simulator. This tool can be used to create a real-time application that contains the behaviour and the I/O model.
  - Model Port Blockset that contains the Simulink blocks used to read or write I/O model signals.
  - RTI CAN MultiMessage Blockset, which contains Simulink blocks used to implement CAN communication in the behavioural model.
  - RTI LIN MultiMessage Blockset, which contains Simulink blocks for implementing LIN communication in the behavioural model.
  - Flexray configuration package, which is used to implement Flexray communication in the behavioural model.
  - Automotive Simulation Models, additional blocks useful for modelling the control system in Simulink.

- RTI FPGA Programming Blockset.
- Experimenting, software tools for managing and controlling the real-time application and visualising the simulation results:
  - ControlDesk Next Generation, which can be used to download the real-time application, calibrate parameters, and measure signals.
  - MotionDesk, which displays 3D animations during experimentation with a SCALEXIO system.
  - ModelDesk for changing and downloading parameters for automotive simulation models for the real-time application.
- Testing, ECU test automation software tools:
  - AutomationDesk, which can create and manage automation tasks.
  - Real-Time Testing, which can run tests simultaneously with the model.

### 5.3.1 ConfigurationDesk

Configuration Desk is a real-time application where the virtual engine is built and is fundamental in the management of what are the models we use at HIL. The models are in real-time, they execute tasks in a well-defined time compared to the algorithms performed by a common PC. All tasks must be executed in a time that is often defined in milliseconds. With ConfigurationDesk we manage models written in a high-level language (Simulink/Matlab). Simulink is indicated to design algorithms of control systems.

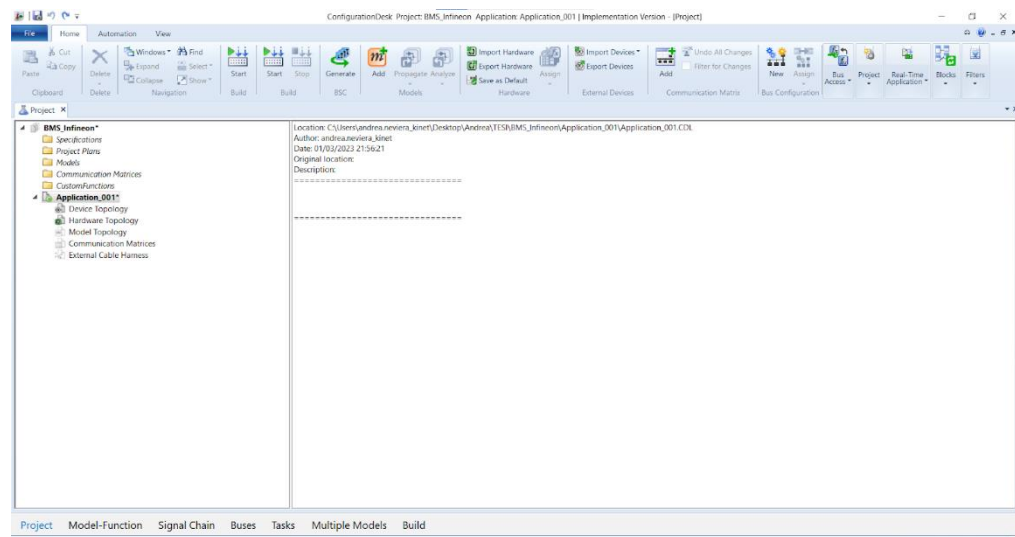


Fig. 113 Project page with configured applications (From ConfigurationDesk)

When we move from Simulink to ConfigurationDesk we have a clear division between behavioural model written in high level through simulation and development software (Matlab, Simulink and Stateflow) and the I/O model. So:

- I/O model: It contains I/O ports and logics that interface the behavioural model to a series of hardware resources made available by dSpace boards. Built in Configuration Desk and can contain the following I/O functions: crankshaft and camshaft signal generation, injection and ignition pulse measurement, lambda signal generation, knock signal generation.
- Behaviour model: built in MATLAB/Simulink, mimics the behaviour of the engine.

A multicore model is typically used, allowing us to consume less power and enhance simulation performance. We are able to create a processor language file that can be divided into distinct real-time cores through the process of code

generation. A model is split into several sub-models when it has to be built to combine multiple real control units. To have more independent units to utilize in different cores and give it a varied synchronization time, there is a software that automatically generates the various sub-models. Imagine a global model with many sub-models, one for the engine and the other for the vehicle's chassis, and so on. Two files are automatically created by the tool. We will be able to run two .slx at separate times since one will run on the first core and the other on the second. Make sure that the model, including the algebraic loop, is error-free before moving on. Simulink needs a clear input as long as a particular computation is carried out; if the output is used as input, an algebraic loop issue occurs. Another mistake can be to assign two signals with the same name to the same bus.

### 5.3.2 ConfigurationDesk interface

We proceed to the Project page to choose the project, which could include several applications.

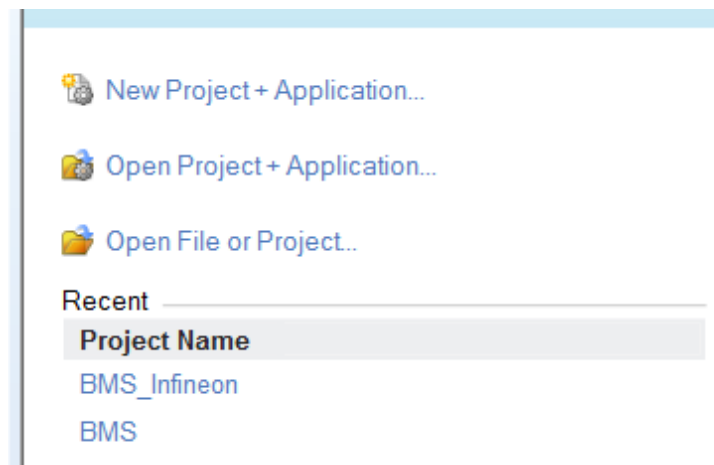


Fig. 114 Project creation and selection menu (From ConfigurationDesk)

In other words, depending on what we need to emulate, we may test out more applications on the same project.

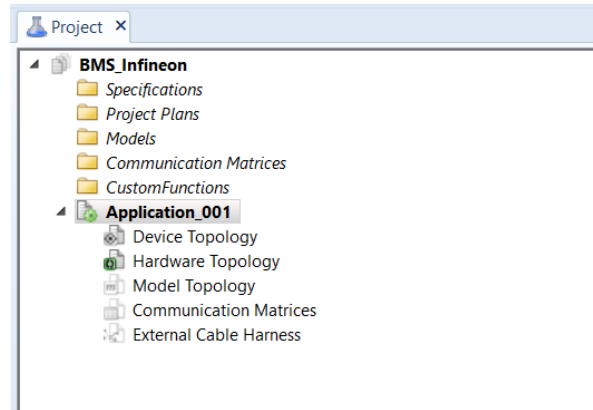


Fig. 115 Application activation and selection menu (From ConfigurationDesk)

The functions that we must configure to provide the characteristics of the channel that will connect the behavioural model with the simulator, or the control unit are found in Model Function. The fourth interface is where the CAN or LIN bus setup happens. We may customize tasks in Tasks and modify a task's execution time. Applications with several processors and cores are managed by Multiple Models.

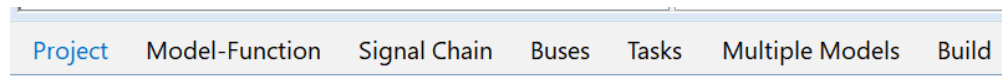


Fig. 116 Interfaces within ConfigurationDesk (From ConfigurationDesk)

I'll have blocks subdivided by application once I've completed the model separation and imported the model topology to ConfigurationDesk. The downstream model of model separation would be the application. Model separation encapsulates all signals to an application in a bus. I have to be cautious about merging signals that move from one application to another while using several models. Despite the model separation, there is communication

between the various applications. I must keep in mind to combine any modified signals in the model that pass across different applications using ConfigurationDesk. Build leads to a translation from the high-level language to a low-level platform dependent language (C language). Downstream of the build we have the following files:

- .sdf: is the most important, it contains all the information about the variable description. I can load the .sdf on ControlDesk to have real-time monitoring on all the variables of the model and go, possibly, to modify them. The .sdf allows me to see what happens if I change a constant.
- .rta: contains information about the platform, what are and how many channels I used on my simulator.
- .trc
- .map

Once you have defined the project and an application, the built model should be imported into Model Topology by entering it as a file .mcd. Hardware Topology, instead, is a mapping of the simulator resources.

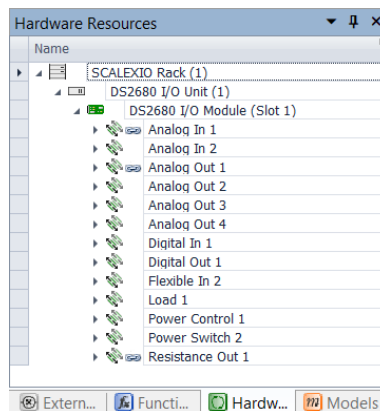


Fig. 117 Hardware Topology section (From ConfigurationDesk)



Inside the ConfigurationDesk we have three different areas: External devices, Functions and Model Port Blocks. The External devices represent the control units and do not fit into the build, they will hardly be changed unless we need them.

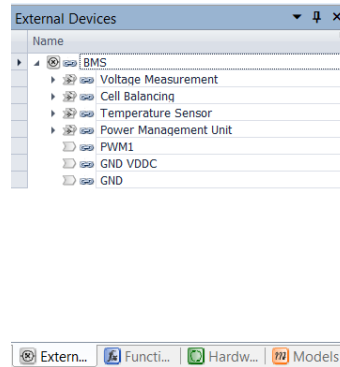


Fig. 118 External devices section (From ConfigurationDesk)

Function Blocks are elements of the dSpace function library and implement I/O functionality. Each block type has unique characteristics and must match the electrical characteristic of its signal to create the interface between the simulator and the model. Depending on the board selected by Hardware Topology, there are several I/O channels selectable by Functions.

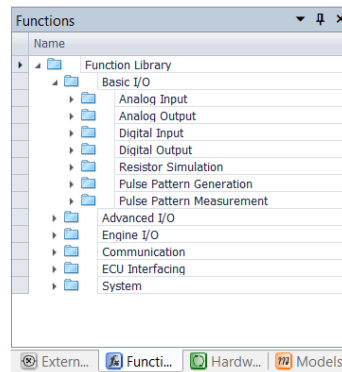


Fig. 119 Functions section (From ConfigurationDesk)

Model Port cannot be modified like Functions. The Model Port has a unique address, with an identifier assigned by the Configuration Desk that brings with it all the information about the Model Port that I have specified in the Functions. If I change the name to the Function, this change does not materialize in the model if I do not regenerate the Model Port again. The Model Port is then exported to the model. By right-clicking we can choose an option that allows us to link the Model Port to the Simulink model. I can copy the same Model Port to different models, but I must do it with the same identifier (Copy and Paste with Identity) otherwise it will copy multiple ports with different identifiers. Within the I/O section of my Simulink model, I'm going to connect all the Model Ports to what the behavioural model signals are. For digital signals we can also specify the logic that is performed for their configuration (pull up, pull down or push/pull) and polarity (active low or active high). Every time I change the model, inserting another Model Port, I must necessarily redo the code generation because otherwise the last changes will not be considered, and the simulation could prove to be fallacious. Once the Simulink build is complete and the files are obtained, the most important among them is the .sdp that will be loaded directly on the Control Desk. The .rta file represents resources.

In the Signal Chain section of ConfigurationDesk we have the following four areas:

- Device topology: External device interface. Can also be imported from an Excel file.
- External Cable Harness: Information about external wiring. Can be exported and imported with an .echx file.
- Hardware topology: Contains information on all hardware resources made available by my platform that I have previously specified. It can also be exported. Once I have realized allocation channels, resources

used and which not, through a .hpfx file I can export and import this hardware topology in another project always using ConfigurationDesk.

- Model topology: Contains interface information for the behavioural model, such as the specified Model Port blocks. Model topology can be created in ConfigurationDesk or established by analysing model interfaces in a behavioural model that is added to the ConfigurationDesk application.

Before building, you usually have to analyse the model. The build is done after assigning the different Model Ports. Analyse allows you to detect errors in the application or even various warnings, such as unused functions or Model Ports or when two functions or two Model Ports have the same name. It gives you a warning when you can't identify hardware resources because for example the platform that was previously identified was then disconnected. If there are no errors the build can be started. The build, as mentioned above, is automatic and produces a series of files and logs with which you can keep track of the code generation process, through which you can also make a possible debugging.

### **5.3.3 Channels configuration on ConfigurationDesk**

Based on the signals detected by the datasheet of the chosen electrical unit, I generated the channels in ConfigurationDesk by following the wiring harness created. Starting with Hardware Topology I chose the SCALEXIO Rack and then installed the DS2680, the MultiCompact I/O unit utilized by the simulator and already seen previously. Through External Device, I created the BMS system and then created several subsystems to identify their function and purpose within the control unit used. Within these subsystems, several ports were added, each representing a pin through which the control unit is directly connected to the simulator.

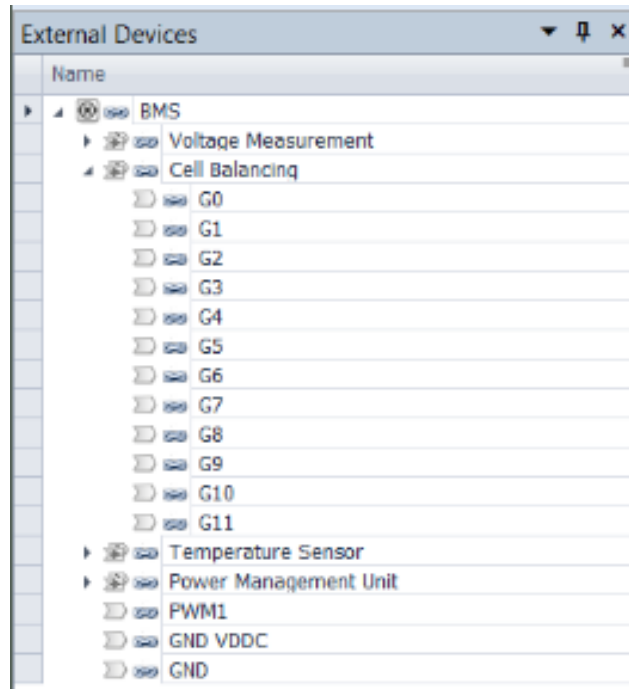


Fig. 120 External device subsystems (From ConfigurationDesk)

I then selected the Based I/O menu from the Functions section to choose the channel types I would like to utilize and drag them into the ConfigurationDesk window's middle area to create channels. Following the order of the subsystems, the first to be configured were the voltage measurement signals, which were set as Analog Out 1. Each of them was assigned a specific channel, and ConfigurationDesk automatically places the position corresponding to its Hypertac within the Electrical Interfaces located below the function name. The Hypertac position must be marked within the wiring harness to create the wiring. In the case of  $U0$ , the pin of the control unit has been set as a Reference of  $U1$  because it corresponds to the negative terminal of the first battery cell, while  $U1$  to the positive one. For subsequent channels, the Reference has not been set for convenience, as the Analog Out channels give us the option to provide a certain voltage value.

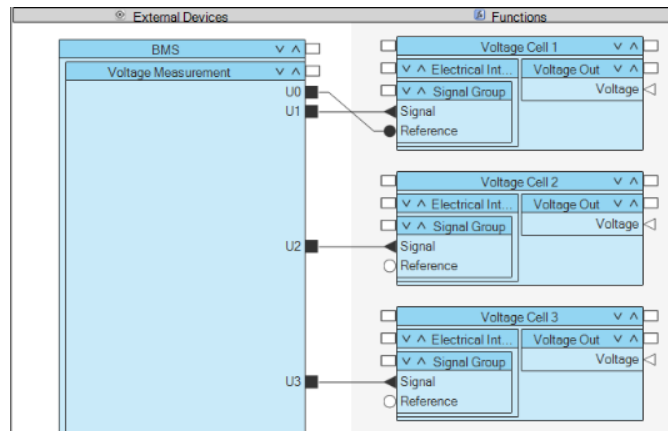


Fig. 121 Configuration of voltage measurement channels (From ConfigurationDesk)

The next section, on the other hand, contains the Analog In channels configured for cell balancing. The external balancing circuit has been modelled directly on Matlab and will be discussed in the next paragraph. The channels configured for balancing are used only to activate it, providing the necessary value to close the switches.

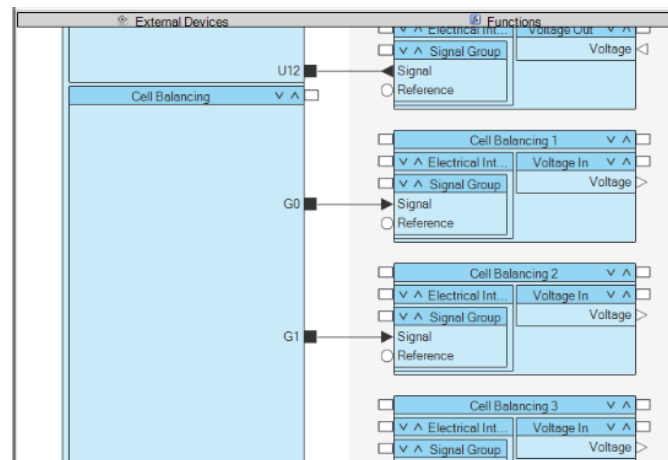


Fig. 122 Configuration of the balancing channels (From ConfigurationDesk)

I added resistance out channels for the temperature sensors. Despite each channel created has a different reference, at the wiring level the references

connected to the Hypertac will be short-circuited on *GND TMP* of the control unit.

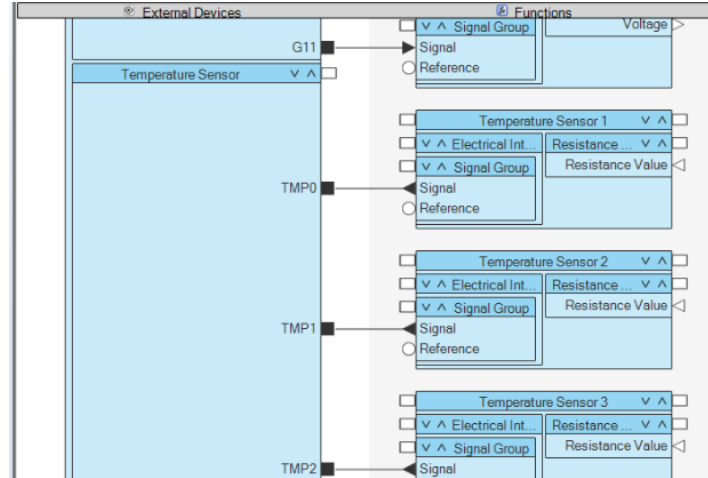


Fig. 123 Configuration of temperature sensor channels (From ConfigurationDesk)

For the internal supply of the IC, simple Analog Outs have been configured. Not being powered by a 12 V power supply, it was not possible to insert Power Switches. Analog Outs allow to set as output the voltage calculated upstream of the battery module. Pins for PWM were added if users wished to utilize them, but their use was not considered for the developing of the thesis work.

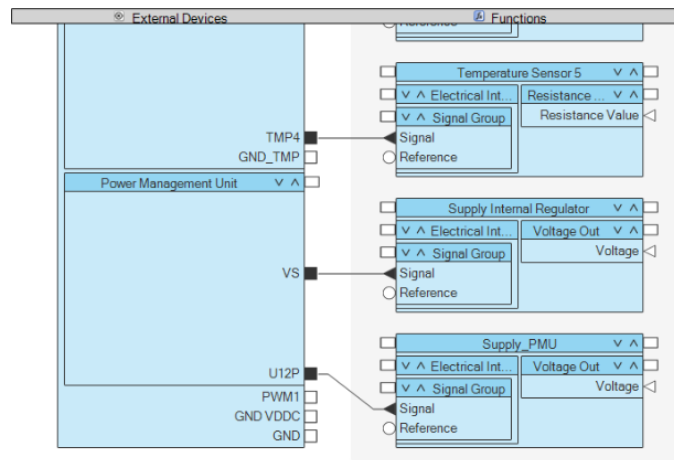


Fig. 124 Configuration of the supply channels (From ConfigurationDesk)

### 5.3.4 ControlDesk

ControlDesk performs all the necessary tasks in a single working environment, from the beginning of the experiment to the end. Some of the tasks you can use are:

- Rapid control prototyping
- HIL simulation
- ECU measurement, calibration and diagnostics
- Access to the vehicle bus system (CAN, CAN FD, LIN, Ethernet)
- Virtual validation with SCALEXIO

ControlDesk organizes work into projects and experiments. Within a project, for example, you can have multiple experiments, each of which can represent a single hardware setup. You can also assign different variable descriptions to a single platform/device in the experiment and easily switch between them. The project level also contains global documents such as specifications or reports. The experimental level is used to organize instrument panels (layout), variable descriptions, data sets, signal generators, fault simulation systems and more. All data is always well structured and can be easily accessed via ControlDesk's project manager. ControlDesk offers two different ways of representing variables in layouts. One way is to select variables via the Variable Browser, place them in the layout (by dragging or a keyboard shortcut), and assign tools to them. Another way is to select a tool using the Instrument Selector, place it in the layout, configure it, and assign variables to the tool. The Variable Browser displays the variables of the platforms/devices included in the experiment. The Instrument Selector provides a whole set of tools that can be placed in the layout. From here, the tools can be configured and the variables

assigned. You can define preferred tools so that you always have them available when working in the Layouting area.

### **5.3.5 ControlDesk interface**

On ConfigurationDesk we have the project that is always a .cdp file and below we have the experiments. ControlDesk allows you to view, monitor and manage (change in real time) the variables of the model that runs on the cores of the real-time processor. They are experiments that allow us to control calibrations and some internal variables. We can measure and change signals at constants. There are various experiments, usually we have a very precise structure in which we put the project. The platform can be entered either through ConfigurationDesk or ControlDesk. To register it just go to Register Platform in Platform. It opens the same menu that we find in ConfigurationDesk. In our case the platform is a SCALEXIO. Then enter the standard IP address (192.168.140.10). If you have more than one simulator connected, you must always give a different IP address than the first (192.168.140.12 or .13, etc.). Both on ControlDesk and ConfigurationDesk, once the platform is registered, all the resources within it are displayed. In case we have a multicore model and connect only one platform (only one real-time processor), ControlDesk warns that some resources are mapped but cannot be viewed and cannot be counted in the experiment. The .sdf is the general file downstream of the build. Below we have applications on the cores of the real-time processor. In the figure we can see three different cores. To set three different cores (applications) there is no ControlDesk tool, but - in the case of FCA- there is a tool created by the company that allows you to make the Model Separation so as to get from a single model different core. For each core we can consider a single node. The real-time processor interfaces with two DS2680 boards which in turn has two modules (I/O and Bus). The GO Online command allows you to send the



application Online. From this moment I can go and change the signals in real time of my model.

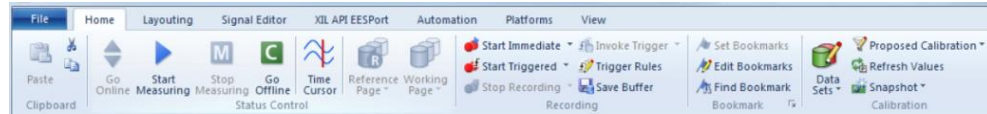


Fig. 125 ControlDesk toolbar (From ControlDesk)

Through ControlDesk you can operate or not the Power Supply, then going to turn off or turn on the low voltage battery.

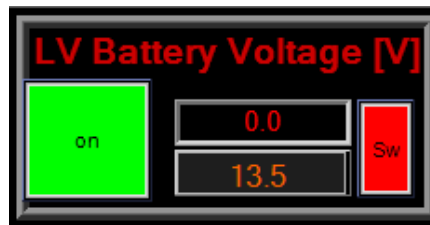


Fig. 126 Low voltage battery display (From ControlDesk)

Inside the main layout you put everything you need to check the status of the simulated vehicle. These include the ignition key, the acceleration and braking pedals and the gears.



Fig. 127 Ignition key, pedals, and gears display (From ControlDeks)

Various displays and graphs can also be used to keep an eye on different parameters such as torque or speed.

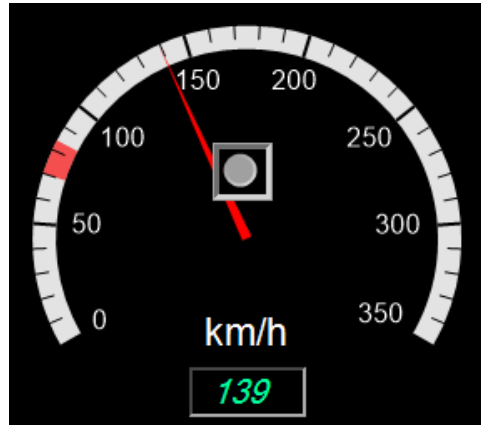


Fig. 128 Speed display in km/h (From ControlDesk)



Fig. 129 Torque display (From ControlDesk)

In addition, linking the variables of the Model you can change in real-time the basic settings, to experience the system under different conditions.



Fig. 130 Display to change the ambient temperature and ambient pressure (From ControlDesk)

## 5.4 Final project

I had to modify the Simulink model so that it would operate flawlessly with the HIL simulator to merge the functionality of the control unit and the channels defined by ConfigurationDesk. After checking that all the integrals of the model were discrete and that there were no algebraic loops, I inserted the buses to transport the signals inside the model between the simulator and the control unit. Unfortunately, the order of the BMS control unit did not arrive in time for the experimental thesis work. However, this hasn't impacted the thesis's objective because, with the help of the simulator HIL's technology, the model I developed can still act in a real-time environment and therefore construct linkages between the channels of the simulator. To do this, I just built a Soft ECU in Simulink that could mimic the logic employed by the control unit in accordance with the signals it gets and sends from the simulator, starting with the identification established with the wiring harness. As a result, we may experiment with other types of real ECU than BMS using the model developed in Simulink. First, I created two different cores using the Simulink Library feature; one for the battery model and the other to simulate the BMS.

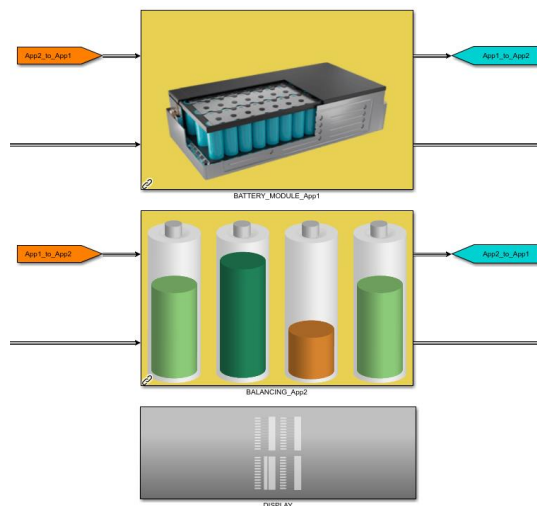


Fig. 131 Division into two cores of the model (From Simulink model)

### 5.4.1 Core 1 – Battery module model

The identification of the model inputs and outputs is essential to understanding how to integrate the new logic into Simulink. Starting from the channels established through Configuration Desk, you can identify as input the signals that will operate the balance within the battery module and that have been set as Analog In respect of the simulator. The outputs, then, will be those signals that have been configured as Analog Out and Resistance Out. In particular, the former are important for the operation of the Soft ECU because for the control unit they represent inputs.

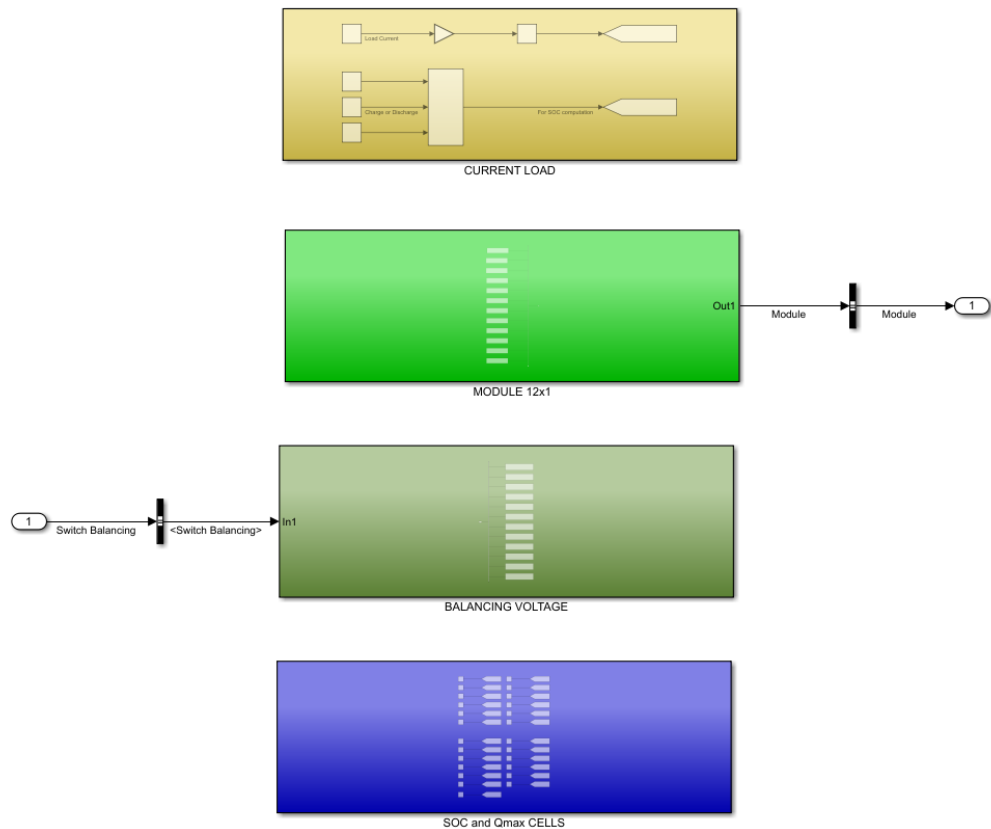


Fig. 132 Subsystems of the first core (From Simulink model)

The Analog Ins coming from the control unit and simulated with the Soft ECU will be used to activate or not the balance of each cell. The balancing model was created from the BMS datasheet, using the suggested external circuit.

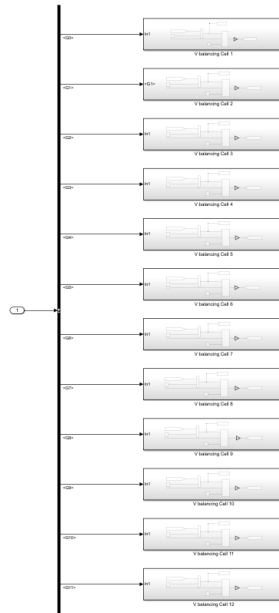


Fig. 133 Balancing voltage subsystem (From Simulink model)

With the bus selector you can divide the twelve signals coming from the Soft ECU, so that each cell can receive the dedicated one. The signals were called according to the pins of the control unit and with the name set to ConfigurationDesk.

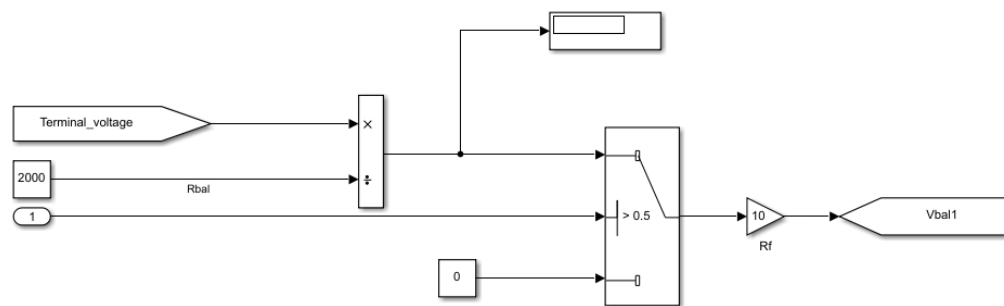


Fig. 134 Balancing voltage computation (From Simulink model)



### 5.4.2 Core 2 – Soft ECU

Since the model can't rely on a real control unit, a Soft ECU that can replicate its behaviour must be created in order for it to work with the simulator or other real ECU that aren't BMS. So I had to go from a real control unit to a simulated one that would operate in an HIL environment. I was able to comprehend the functioning of the Infineon control unit and the setup of its channels by an in-depth examination of both, which prepared me to build a model that could mimic its behaviour. For the realization of the Soft ECU I have not considered the analysis of the temperature, since the control unit is limited to measure it and, eventually, to signal it in case it exceeds the limits. What I based on is the method with which the control unit activates and deactivates the balance, thanks to the measurements it implements both for the voltage of the individual cells and for the SOC. The library created is divided into four subsystems, three of which receive the signals directly from the model of the battery module.

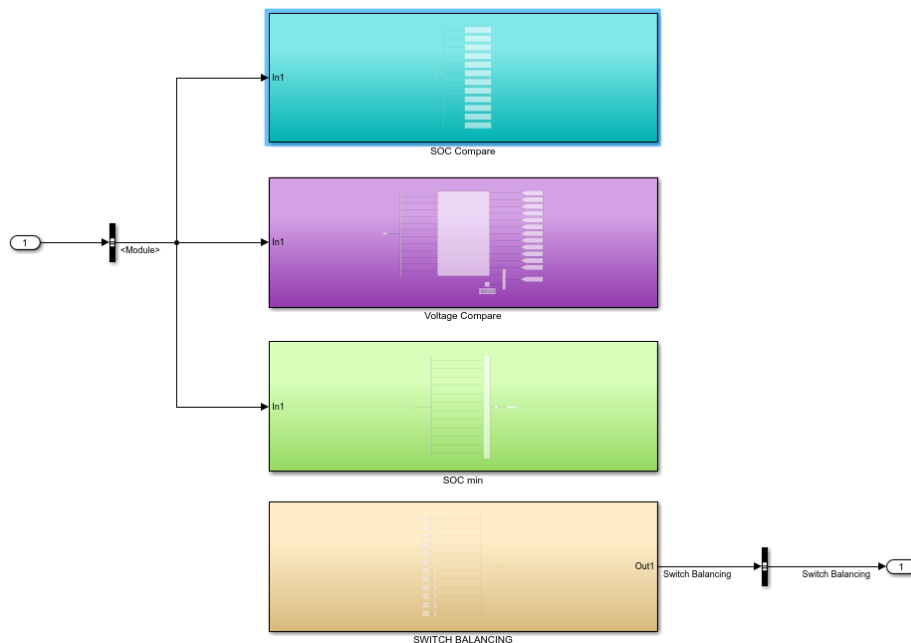


Fig. 137 Subsystems of the second core (From Simulink model)

The SOC min subsystem simply finds out how much the SOC corresponds to the cell that currently has less capacity than the others. It was done with a simple Min block and a Unit delay block was added to not create any algebraic loops.

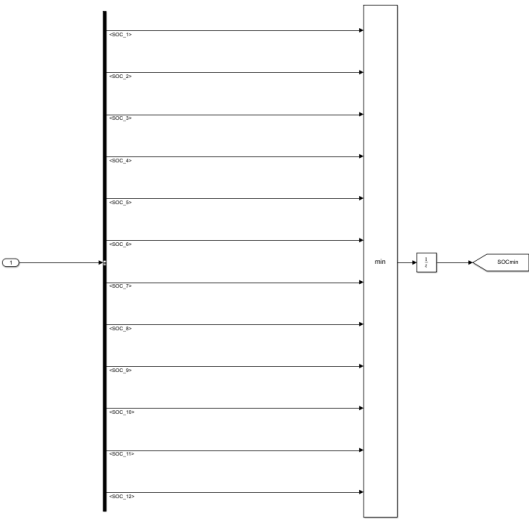


Fig. 138 Minimum SOC computation (From Simulink model)

The SOC subsystem appears, instead, is used to compare the SOC's of the various cells with the minimum SOC previously calculated.

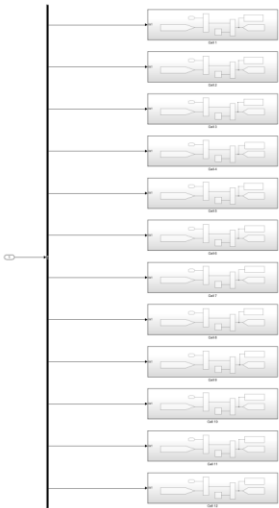


Fig. 139 Comparison between the various SOC's (From Simulink model)



Similar to the calculation made in the model of the battery module, the two SOC's are compared. If the difference between the two is greater than 2%, a signal equal to 1 is sent because of the Relational Operator block. Otherwise the signal is 0.

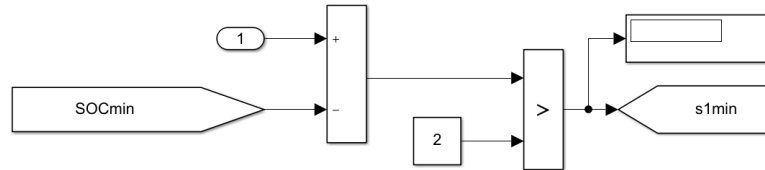


Fig. 140 Comparison between the minimum SOC and that of the cell considered (From Simulink model)

The third subsystem, Voltage Compare, deals with comparing the different voltages measured between the cells through a Matlab function. The inputs of the function are the voltages, while the outputs are values that vary between 0 and 1 to operate or not the balancing in each battery cell.

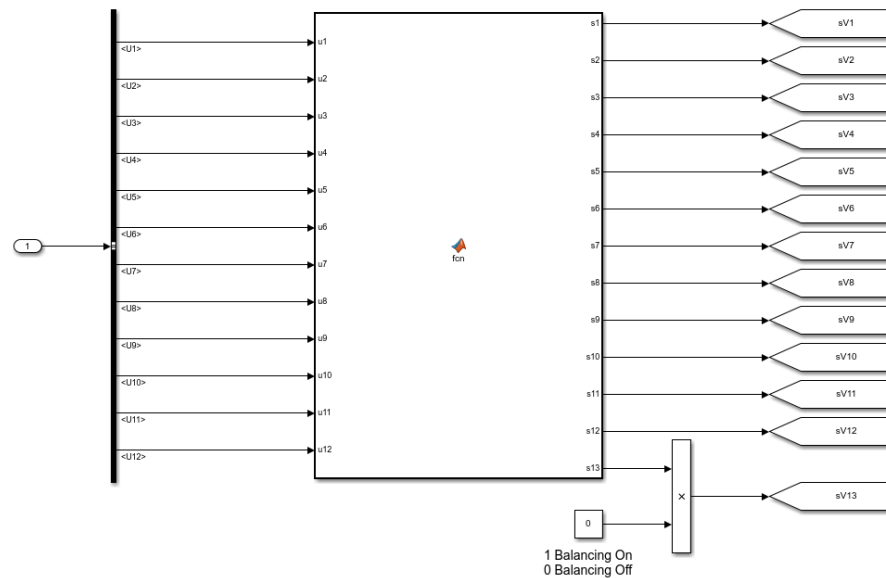


Fig. 141 Matlab function for comparison of cell voltages (From Simulink model)

The *sV13* switch is a comparison of all voltages. If these are equivalent it will give a value of 0, otherwise it will give 1. Moreover, since this switch is connected to all cells I have inserted a constant block linked to ControlDesk to operate or not the balance according to will. The code used in the Matlab function is as follows:

```
function
[s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13] =
fcn(u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12)

if ((u1>u2) || (u1>u3) || (u1>u4) || (u1>u5) || (u1>u6) || (u1>
u7) || (u1>u8) || (u1>u9) || (u1>u10) || (u1>u11) || (u1>u12))
    s1=1;
else
    s1=0;
end

if ((u2>u1) || (u2>u3) || (u2>u4) || (u2>u5) || (u2>u6) || (u2>
u7) || (u2>u8) || (u2>u9) || (u2>u10) || (u2>u11) || (u2>u12))
    s2=1;
else
    s2=0;
end

if ((u3>u2) || (u3>u1) || (u3>u4) || (u3>u5) || (u3>u6) || (u3>
u7) || (u3>u8) || (u3>u9) || (u3>u10) || (u3>u11) || (u3>u12))
    s3=1;
else
    s3=0;
end

if ((u4>u2) || (u4>u3) || (u4>u1) || (u4>u5) || (u4>u6) || (u4>
u7) || (u4>u8) || (u4>u9) || (u4>u10) || (u4>u11) || (u4>u12))
    s4=1;
else
    s4=0;
end

if ((u5>u2) || (u5>u3) || (u5>u4) || (u5>u1) || (u5>u6) || (u5>
u7) || (u5>u8) || (u5>u9) || (u5>u10) || (u5>u11) || (u5>u12))
    s5=1;
else
    s5=0;
```

```

end

if ( (u6>u2) || (u6>u3) || (u6>u4) || (u6>u5) || (u6>u1) || (u6>
u7) || (u6>u8) || (u6>u9) || (u6>u10) || (u6>u11) || (u6>u12) )
    s6=1;
else
    s6=0;
end

if ( (u7>u2) || (u7>u3) || (u7>u4) || (u7>u5) || (u7>u6) || (u7>
u1) || (u7>u8) || (u7>u9) || (u7>u10) || (u7>u11) || (u7>u12) )
    s7=1;
else
    s7=0;
end

if ( (u8>u2) || (u8>u3) || (u8>u4) || (u8>u5) || (u8>u6) || (u8>
u7) || (u8>u1) || (u8>u9) || (u8>u10) || (u8>u11) || (u8>u12) )
    s8=1;
else
    s8=0;
end

if ( (u9>u2) || (u9>u3) || (u9>u4) || (u9>u5) || (u9>u6) || (u9>
u7) || (u9>u8) || (u9>u1) || (u9>u10) || (u9>u11) || (u9>u12) )
    s9=1;
else
    s9=0;
end

if ( (u10>u2) || (u10>u3) || (u10>u4) || (u10>u5) || (u10>u6) |
| (u10>u7) || (u10>u8) || (u10>u9) || (u10>u1) || (u10>u11) ||
(u10>u12) )
    s10=1;
else
    s10=0;
end

if ( (u11>u2) || (u11>u3) || (u11>u4) || (u11>u5) || (u11>u6) |
| (u11>u7) || (u11>u8) || (u11>u9) || (u11>u10) || (u11>u1) ||
(u11>u12) )
    s11=1;
else
    s11=0;
end

```

```

if ( (u12>u2) || (u12>u3) || (u12>u4) || (u12>u5) || (u12>u6) |
| (u12>u7) || (u12>u8) || (u12>u9) || (u12>u10) || (u12>u11) |
| (u12>u1) )
    s12=1;
else
    s12=0;
end

if ( (u1==u2) && (u2==u3) && (u3==u4) && (u4==u5) && (u5==u6) &
& (u6==u7) && (u7==u8) && (u8==u9) && (u9==u10) && (u10==u11)
&& (u11==u12) )
    s13=0;
else
    s13=1;
end

```

The last subsystem is the one that deals with the generation of the outputs of the Soft ECU that will then be used to operate or not the balancing. The signals  $sVn$ ,  $sV13$  and  $snmin$  are multiplied and the result is sent to the corresponding cell to operate or not the balance as if there were the transistor used in the external circuit.

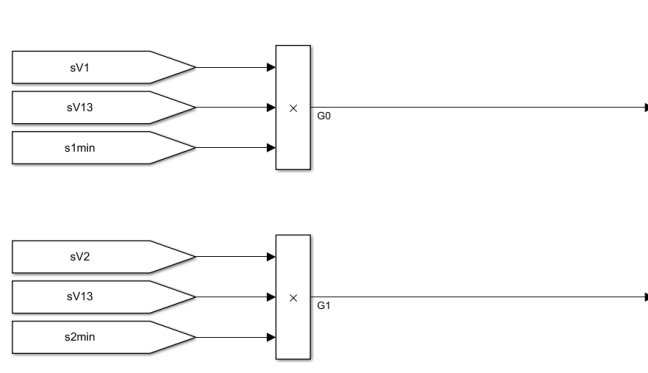


Fig. 142 Switch signals for the balancing (From Simulink model)

By simulating multiple charge and discharge cycles on Simulink, the following SOC, voltage and temperature results were obtained with active balancing:

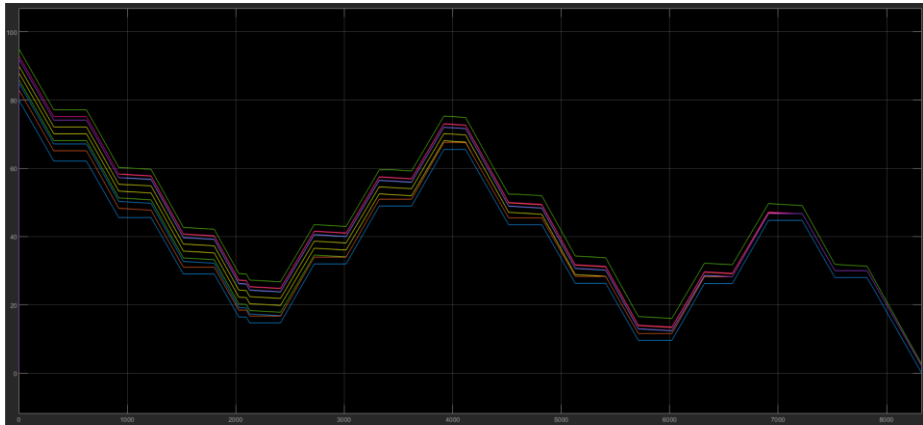


Fig. 143 SOCs graph with active balance (From Simulink model)

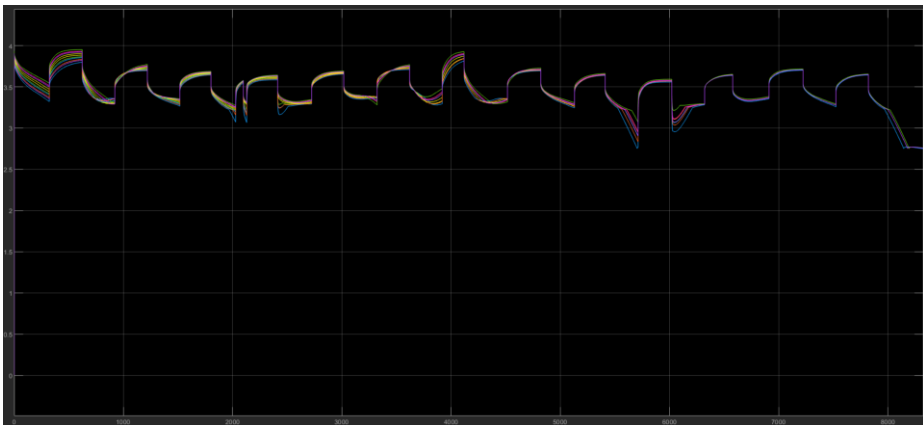


Fig. 144 Terminal voltages graph with active balance (From Simulink model)

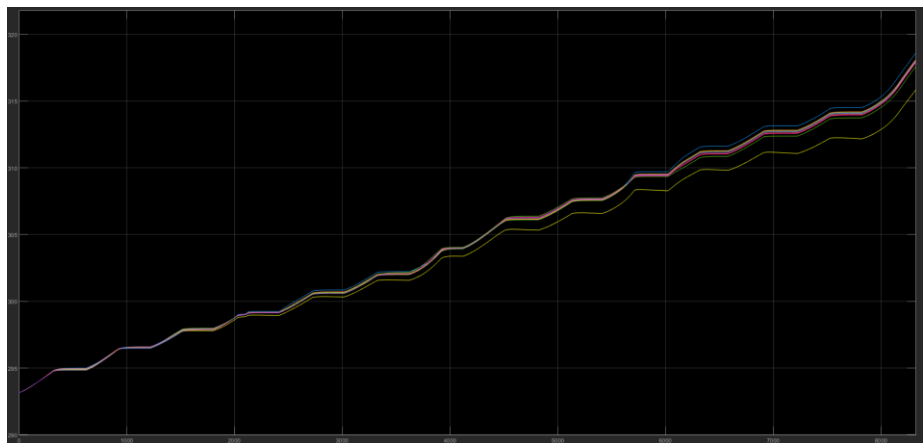


Fig. 145 Temperatures graph with active balance (From Simulink model)

The simulation was performed using a 2C current for both charge and discharge. It must be taken into account that depending on the capacity of the cell used, the current required and the configuration of the battery pack the discharge or charging time varies significantly. In fact, by connecting the cells in parallel you have the possibility to increase their capacity in Ah. The battery cell used for this simulation is the Panasonic NCR18650PF. The Matlab code to set the parameters is as follows:

```
clear all
close all
clc

%% Internal resistance
%Actual R0
R0_LUT = [0.0367 0.0367;0.0367 0.0367;0.0354
0.0354;0.0341 0.0341;0.0333 0.0333;0.0331
0.0331;0.0335 0.0335;0.0338 0.0338;0.0342
0.0342;0.0357 0.0357;0.0378 0.0378];
%R0 begin of life
R0new = [0.0292 0.0292;0.0284 0.0284;0.0273
0.0273;0.0267 0.0267;0.0263 0.0263;0.026
0.026;0.0265 0.0265;0.0268 0.0268;0.0274
0.0274;0.0281 0.0281;0.0286 0.0286];
%R0 end of life
R0end = [0.0734 0.0734;0.0734 0.0734;0.0708
0.0708;0.0682 0.0682;0.0666 0.0666;0.0662
0.0662;0.067 0.067;0.0676 0.0676;0.0684
0.0684;0.0714 0.0714;0.0756 0.0756];

%% RC branches
%Resistance
R1_LUT = [0.0687 0.0687;0.0687 0.0687;0.0082
0.0082;0.0064 0.0064;0.006 0.006;0.0058
0.0058;0.0081 0.0081;0.007 0.007;0.0067
0.0067;0.0074 0.0074;0.0104 0.0104];
R2_LUT = [0.0378 0.0378;0.0378 0.0378;0.0264
0.0264;0.0324 0.0324;0.0258 0.0258;0.0217
0.0217;0.0702 0.0702;0.0465 0.0465;0.034
0.034;0.0272 0.0272;0.0314 0.0314];
%Capacitance
```

```

C1_LUT = [25.0281 25.0281;25.0281 25.0281;133.5712
133.5712;866.5248 866.5248;974.5801
974.5801;844.86800000000001 844.86800000000001;464.479
464.479;326.3482 326.3482;316.9075 316.9075;264.6779
264.6779;31.0652 31.0652];
C2_LUT = [1146.8 1146.8;1146.8 1146.8;2155.4
2155.4;2881.7 2881.7;3093.1 3093.1;3027.5
3027.5;1782.4 1782.4;1499.3 1499.3;1500.3
1500.3;1623.3 1623.3;1513.5 1513.5];

%% SOC and OCV computation
%Self-discharge Resistance
Rsd = 1000000;
SOC_LUT = [0 10 20 30 40 50 60 70 80 90 100];
Np = 1; %cells in parallel
Qrated = 2.7*Np;

%% OCV computation
Em_LUT = [3.49 3.51 3.52;3.55 3.56 3.56;3.62 3.64
3.64;3.71 3.71 3.72;3.91 3.93 3.94;4.07 4.08
4.08;4.19 4.19 4.19]
SOC_LUT_OCV = [0;10;25;50;75;90;100];
Temperature_LUT_OCV = [278.15 293.15 313.15];

%% Thermal model
%Temperature
Tin = 293.15;
Temperature_LUT = [293.15 293.16];
%Cell architecture
cell_area = 0.0019;
cell_mass = 0.47;
cell_thickness = 0.0105 %also diameter

Cell_Cp_heat = 810.5328;
h_conv = 5; %convective coefficient
k = 239; %conductive coefficient
Rth =
[1/(cell_area*h_conv)]+[cell_thickness/(cell_area*k)
] %thermal resistance

save Parameters.mat

```

### 5.4.3 New application in ConfigurationDesk

No longer having a real control unit, even the application on ConfigurationDesk must necessarily change. In this case, to continue testing in the HIL environment the original outputs and inputs have not changed, but Analog In and Analog Out channels have been added. To be precise, Analog In 1 was added using the output from the voltage measurements of the cells generated by the Simulink model. Through the wiring, then, the voltages leaving the simulator are reinserted into the same so that they can be used in the core dedicated to the balancing of the battery cells.

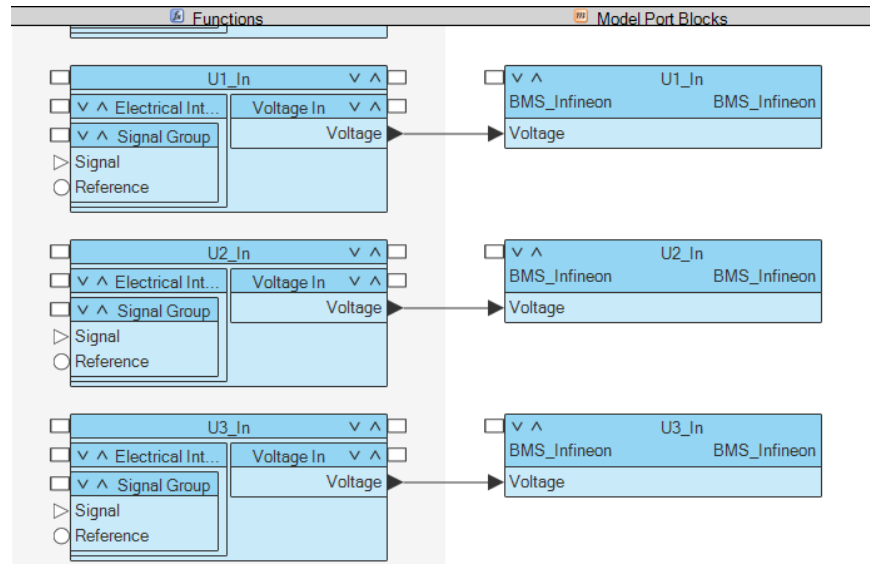


Fig. 146 Voltage Analog In functions and model ports (From ConfigurationDesk)

On the contrary, for the balancing channels, Analog Out 1 has been created, also connected via bridge connections in the wiring to the inputs already configured. This allows us to test the model and the configuration performed in real time, despite the fact that there is no physical control unit to receive or send signals. By doing this we can still understand how the model runs on the simulator and whether it is easily adaptable to a real control unit.



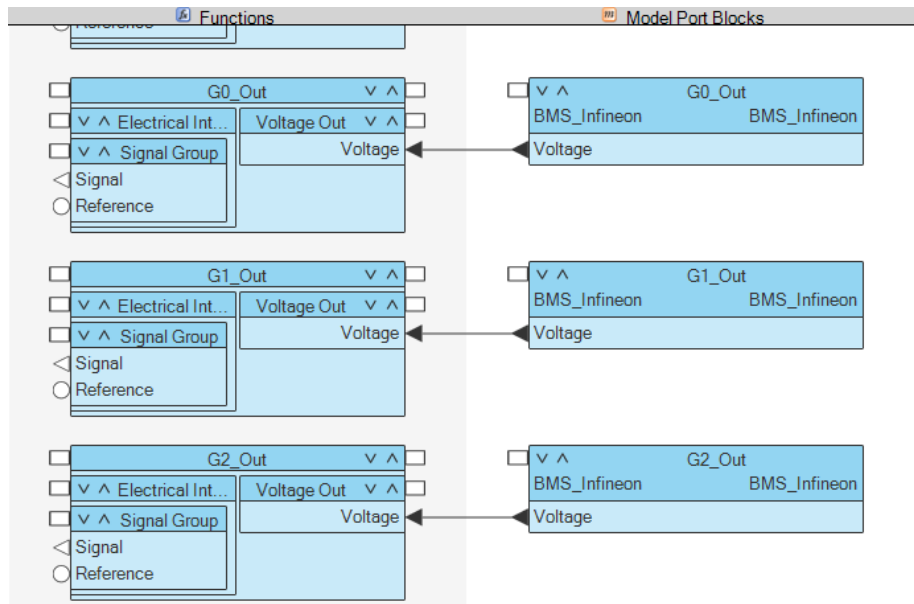


Fig. 147 Balancing Analog Out functions and model ports (From ConfigurationDesk)

Created the channels on ConfigurationDesk I have connected them to the Simulink model generating the Model Port through the command highlighted in the figure below.

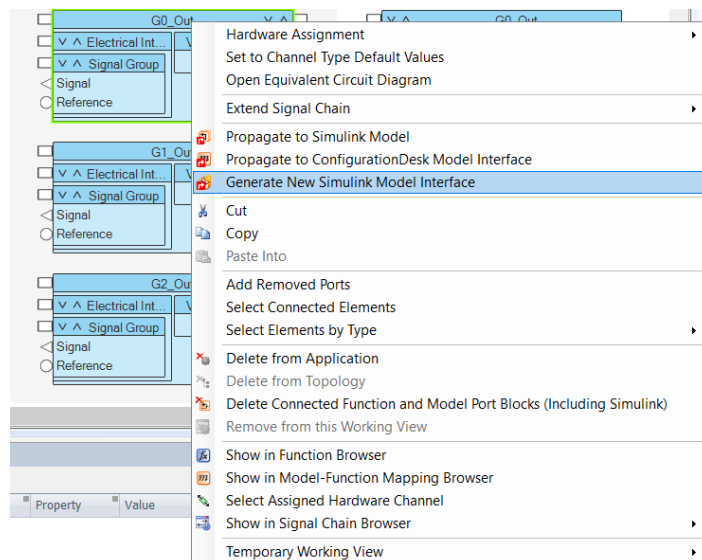


Fig. 148 Generate New Simulink Model Interface (From ConfigurationDesk)

After that, ConfigurationDesk activates Matlab/Simulink and creates a new model in which you can find the Model Port side Simulink. These must necessarily be copied and pasted with identity within the model linked to the project, to then convey the signals through bus creator and bus selector.

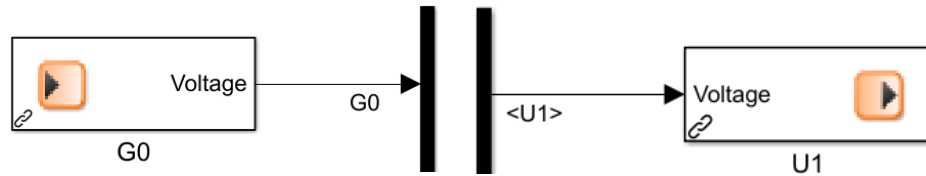


Fig. 149 Model Ports in Simulink (From Simulink model)

Once I connected all the Model Ports within the Simulink model, I proceeded with the model diagnosis to make sure that there are no errors or warnings that could affect the build success.

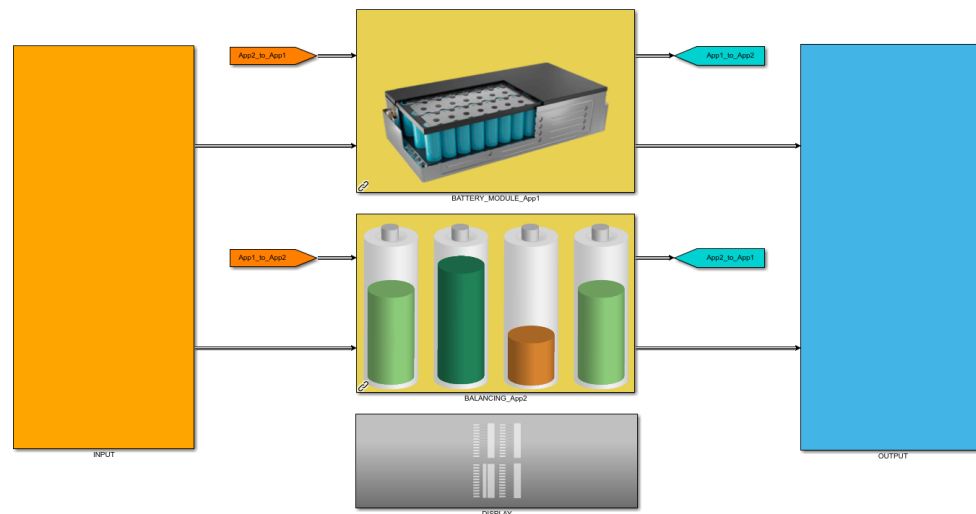


Fig. 150 Final Simulink model with I/O from ConfigurationDesk (From Simulink model)

After that, having positive response to the diagnosis of Simulink I performed the Analyze of the Model Port in ConfigurationDesk, last check before

proceeding with the build generation, resulting in code generation and creation of the .sdf file to be used for the experiment. Once the file has been created, it's possible to proceed with data analysis and real time simulation via the simulator recorded in Platform.

```

==== Starting build process =====
Model code generation started for model(s): BMS_Infincon...
Generating model code for model 'BMS_Infincon' ...
Model code generation for model 'BMS_Infincon' succeeded.
Importing model data of model 'BMS_Infincon' ...
Import of model data for model 'BMS_Infincon' finished.
Model code generation completed for model(s): 'BMS_Infincon'
Presystem integration code generation phase started...
Checking model communication...
Presystem integration code generation phase finished.
System integration code generation phase started for application process 'BMS_Infincon_Task'...
System integration code generation phase successful for the following application process: BMS_Infincon_Task.
Make phase started for application process 'BMS_Infincon_Task'...
Making 'BMS_Infincon_Task' ...
Make directory is 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\Compile'
Target compiler is 'XMC-GNU' at 'C:\ProgramData\dspace\1166892c-8928-4591-a1e9-b64339797c20\Compiler\GNU650_120\host\win32\gcc.exe'
Generating dependencies ...
Generating dependencies finished
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\SysIntCode\mdladapter_mdifunctions.cpp' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\SysIntCode\dsimengine_accesspoint.cpp' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\SysIntCode\mdlcode_ap.cpp' ...
Compiling 'C:\Program Files\dspace\RCPH11_2017-B\ConfigurationDesk\Implementation\EmbeddedSW\src\dsimengine_main.cpp' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\SysIntCode\IIOCode_Data.cpp' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\SysIntCode\rtcsal_simengineap.cpp' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\SysIntCode\rtcsal_task.cpp' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Build\BMS_Infincon_Task\SysIntCode\rtcsal_taskap.cpp' ...
Making library 'BMS_Infincon.a' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Components\BMS_Infincon\BMS_Infincon_dsrc\BMS_Infincon.c' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Components\BMS_Infincon\BMS_Infincon_dsrc\BMS_Infincon_data.c' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Components\BMS_Infincon\BMS_Infincon_dsrc\BMS_Infincon_dsrcmdifcn.c' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Components\BMS_Infincon\BMS_Infincon_dsrc\rt_nonfinite.c' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Components\BMS_Infincon\BMS_Infincon_dsrc\BMS_Infincon_dsrcmf.c' ...
Compiling 'C:\Users\andrea.neviera_kinet\Desktop\Andrea\TES1\BMS_Infincon\Voltage_measurement\Components\BMS_Infincon\BMS_Infincon_dsrc\BMS_Infincon_dsrcrtf.c' ...

```

Fig. 151 Build and code generation interface (From ConfigurationDesk)

## 6. CONCLUSIONS AND FUTURE DEVELOPMENTS

The original goal of this work was to create a model to test a battery management system in a hardware-in-the-loop environment. Due to the delay in delivery of the selected ECU, it was not possible to test the model on it, but thanks to the technology provided by dSpace simulators, it was possible to see how the model would relate to it by analysing the channels. The HIL simulators allow us to send and receive electrical signals between our model and the chosen control device, whatever that is. So once the channels are configured, even if there is no physical ECU connected, you can see the voltage values it would receive if it were connected directly to the wiring coming out of the simulator. Once you configure the channels through ConfigurationDesk and connect the associated model ports within the Simulink model, the model outputs are converted directly from simulated values to real values, so the connected devices are fooled. In this way, the user can test more than one control unit at the same time, each of them feeling immersed in the vehicle thanks to the projects created. Thus, although there is no real ECU, as long as the values of the signals coming from the simulator channels are the same that we would expect from the ECU, it is still possible to understand whether the objective of the thesis has been achieved or not. In addition, thanks to the Soft ECU developed on Simulink, it is possible to generate signals from the simulator and send them to the simulator itself to also understand how the model behaves with the input received. So in the final phase of the work, I put together the wiring according to the diagram below using simple coated copper wires and added the connectors for the Hypertac. The dSpace simulator I used had two DS2680 MultiCompact I/O boards, labelled L and R to distinguish them in ConfigurationDesk, which allowed me to use enough channels to simulate the signals from all twelve cells of the battery module modelled in Simulink.

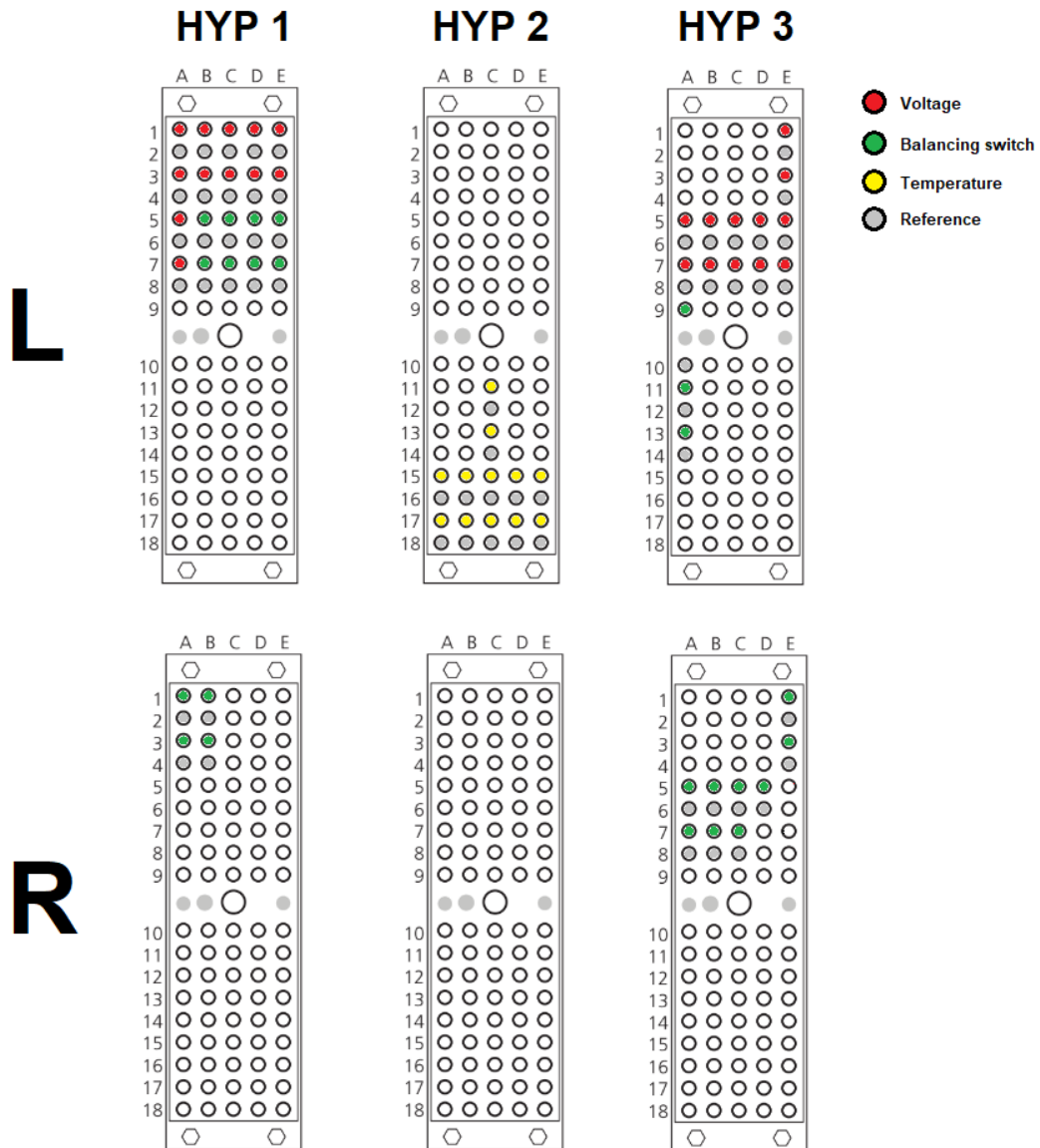
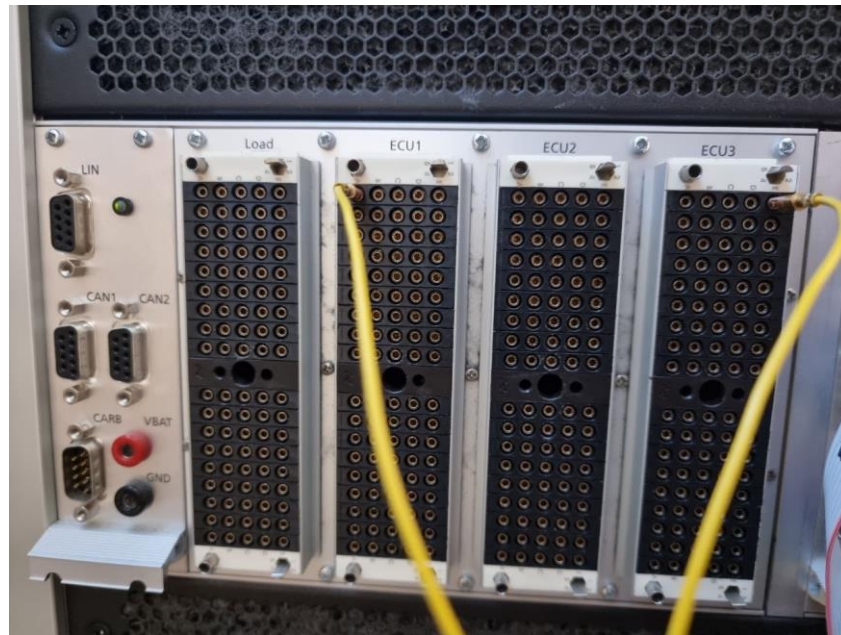


Fig. 152 Hypertac wiring configuration

Shown in red are the connections for the voltage signals that originate from the Analog Out of Hypertac 3 (HYP 3) and terminate in Hypertac 1 (HYP 1), where the Analog Ins are located. Then the battery model simulates the voltage values of each cell, which are then actually sent from HYP 3 to HYP 1, so that the Simulink model, which replicates the behaviour of the real ECU, can receive them to

compensate. Shown instead in green are the channels used to perform the equalization for each cell in the module, configured exactly as explained in the previous chapter. In addition, the yellow channels are dedicated to the temperature sensors, while the grey channels contain the references of each channel.

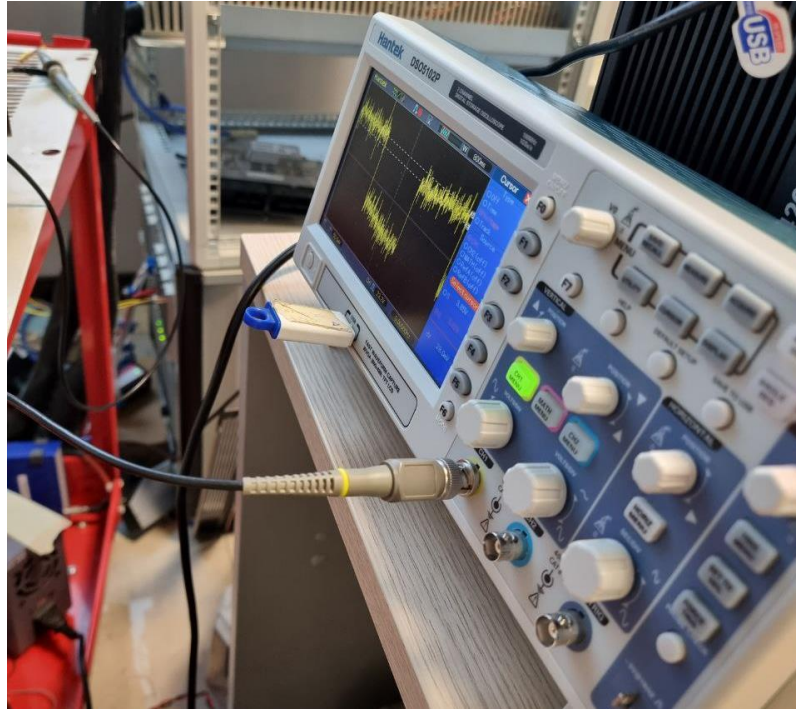


*Fig. 153 Bridge connection between Analog Out and Analog In*

In order to understand if the exact value produced by the model comes from the channels, I took measurements on the first cell using both the digital multimeter and the oscilloscope. With the first lab instrument, I was able to observe the exact voltage values displayed on ControlDesk during the change. With the oscilloscope, on the other hand, I could only get an estimate because of the noise, but the results still reflect the expected values and were measured with both the model and the multimeter. It can be concluded that if the actual ECU had been connected to the wiring, it would have received exactly the voltage values generated by the model for each cell. For the measurements, I used Simulink to simulate the discharge mode in which current is supplied from the battery cells. To do this, I set up a button



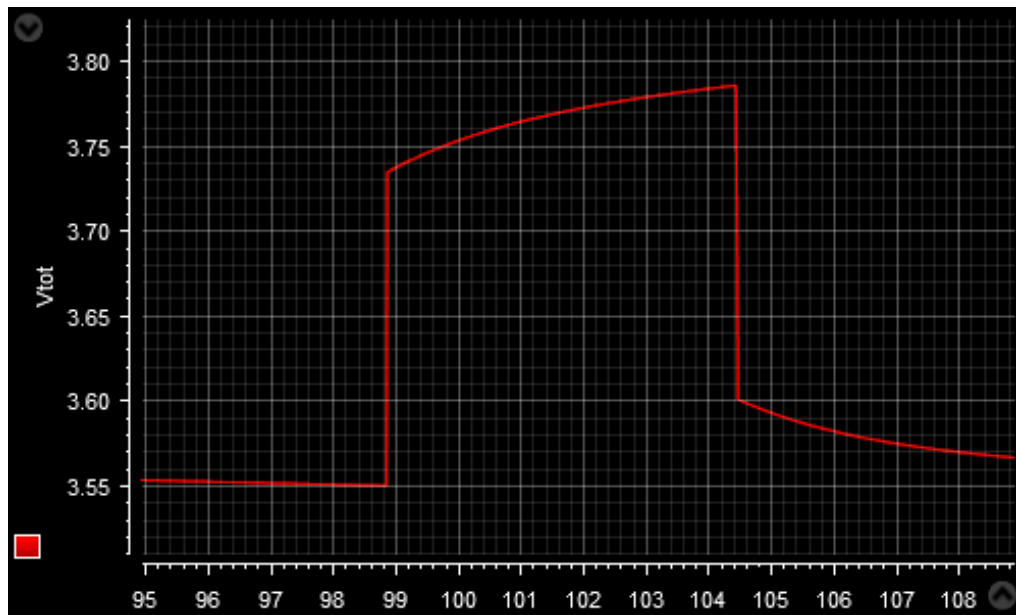
in ControlDesk that allows me to change the current delivery in real time so that I can set it to a constant pre-set value or a zero value as desired.



*Fig. 154 Oscilloscope connected to the simulator used*

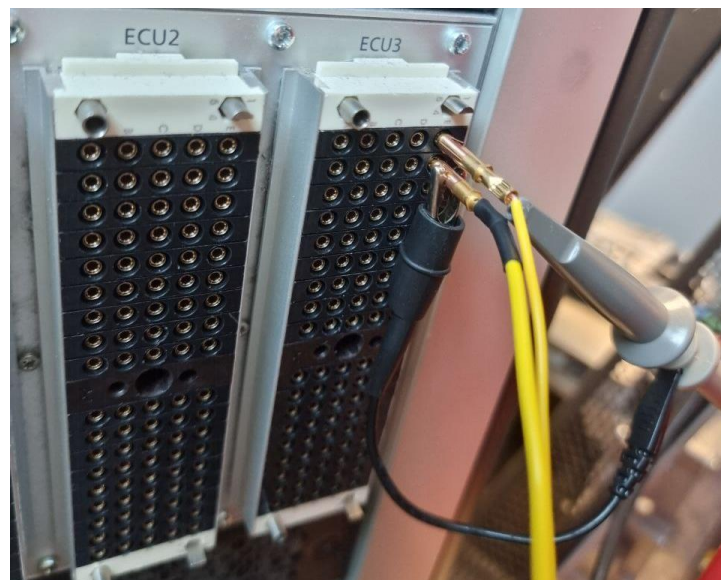


*Fig. 155 Value obtained on the multimeter in an instant of time*



*Fig. 156 Voltage measurement of the first cell on ControlDesk made during the multimeter test (From ControlDesk)*

For obvious reasons, the multimeter and oscilloscope measurements were made only on the Analog Out, using the same analog channel as the signal source and GND as the reference, as shown in the next figure.



*Fig. 157 Oscilloscope probes connected to the first battery cell channel*



When I noticed the noise and interference in the oscilloscope measurements, I changed the settings of the lab instrument to get a wave as clean and similar to the one obtained on ControlDesk.



Fig. 158 Voltage trend of the first cell recorded on the oscilloscope

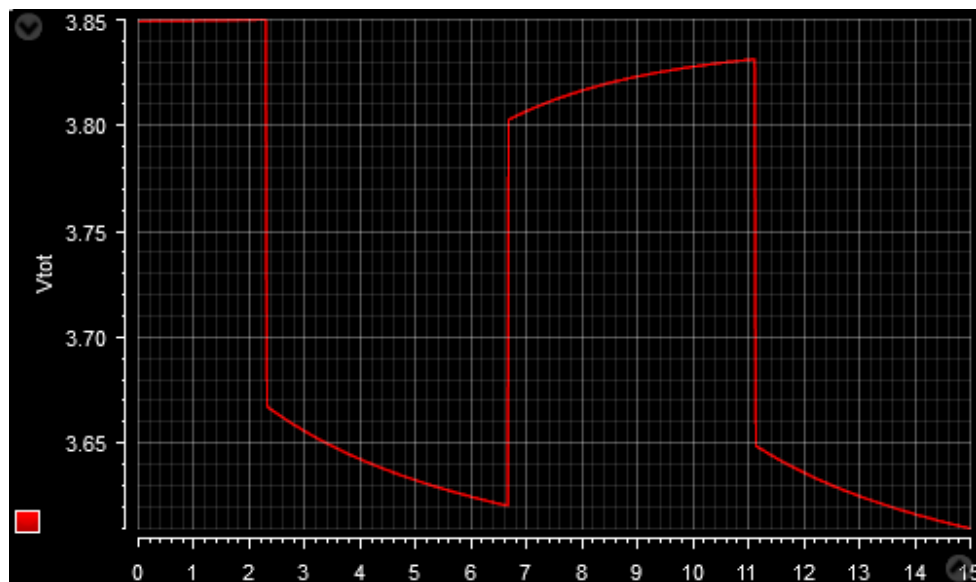
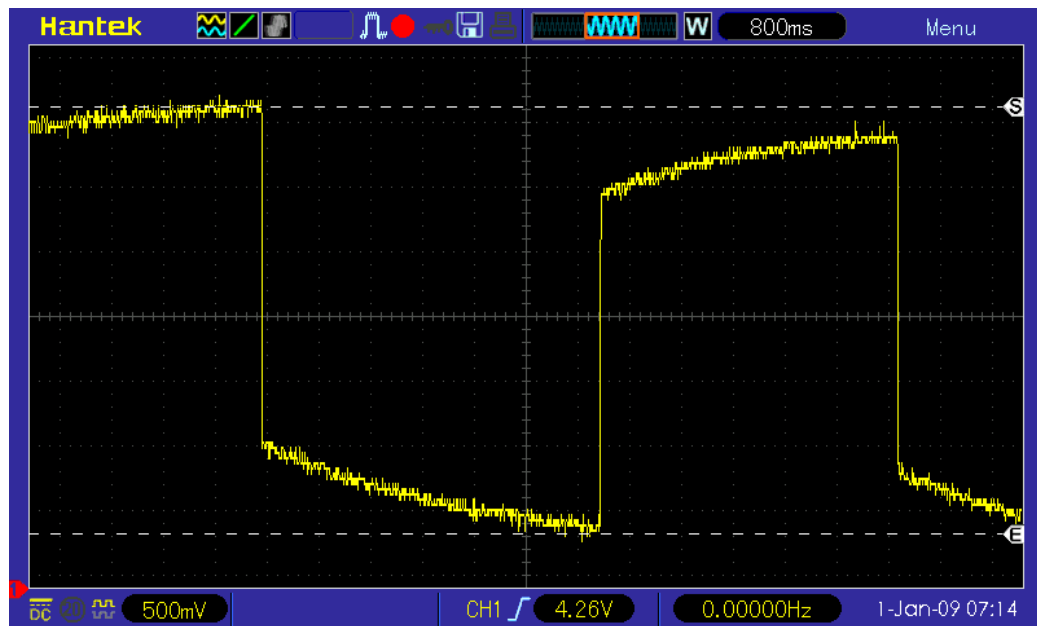


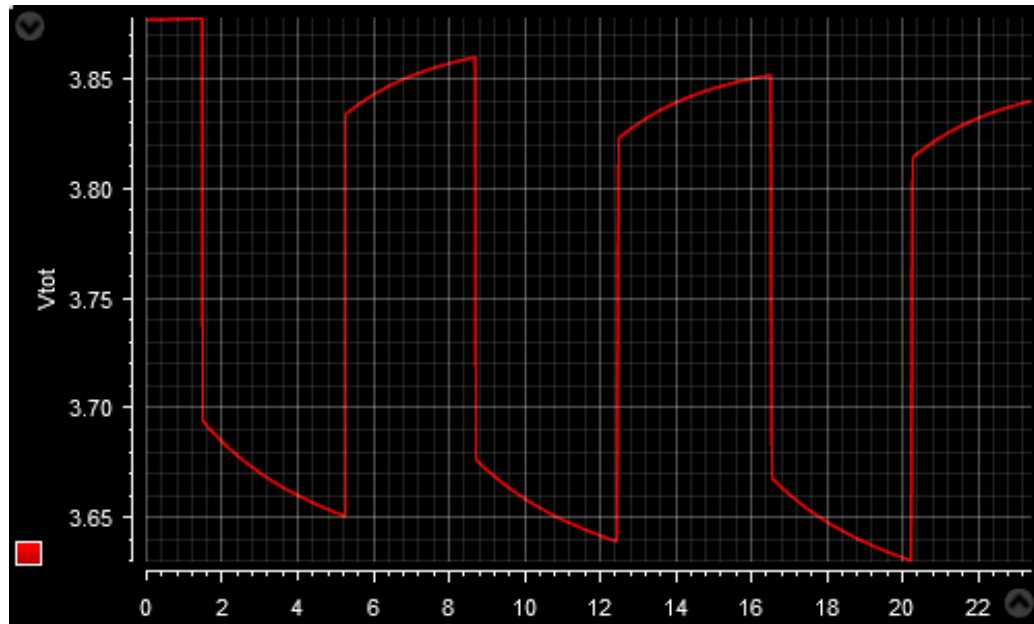
Fig. 159 Measurements of cell voltage acquired on ControlDesk concurrently with oscilloscope acquisition

First, I replaced the probe to make sure it wasn't the cause of the noise. Then, when I saw that the noise persisted, I changed the detection of the trigger, which brought a slight improvement thanks to the noise suppression. However, I managed to eliminate the noise completely by changing the analog signal reference. By moving the reference from the GND of Analog Out 1 to the general GND of the simulator (HYP 2 A05), I was able to achieve the following results.



*Fig. 160 Voltage trend of the first cell recorded on the oscilloscope without noise*

From the waveform recorded by the oscilloscope, you can see the voltage of the cell when the load current is zero. Also, you can observe the discharge process and the gradual change of the voltage during this process. Thus, thanks to the logic developed in Simulink, the simulator is able to provide a voltage equal to that of a real battery cell, even if it's simulated. To see significant changes in the oscilloscope, a load current of more than 2C was simulated compared to the cell used for the simulation. The signal displayed on the oscilloscope corresponds to the following graph retrieved via the ControlDesk interface.



*Fig. 161 Measurements of cell voltage acquired on ControlDesk concurrently with oscilloscope acquisition*

From the two figures above, it can be seen that the real-time simulation performed via ControlDesk matches the results obtained by acquiring the voltage supplied by the simulator as output. Thus, it can be seen that if the control unit had been present, it would have obtained exactly the signals generated by the model created in the Simulink environment and implemented with the dSpace toolchain in the HIL environment. The goal of the work was thus achieved, as the model created could provide the necessary information to refer to a real or simulated BMS, i.e., to give real signals in any state through the channels configured in the simulator. Of course, the channels available to an HIL simulator are not sufficient to provide a voltage for all cells within a high-voltage battery pack. However, HIL testing is primarily concerned with the operation of the connected electronic control units and the signals they receive, not the model. The test considers a BMS as a single CMC, so the rest of the cells are simulated and only the signals required to operate the control unit are used. To simulate the voltage of an entire HV battery pack I set a total of 192 cells in series to arrive at a voltage slightly higher than 800 V, connected in parallel to as many to increase the capacity for a total of 32 modules used.

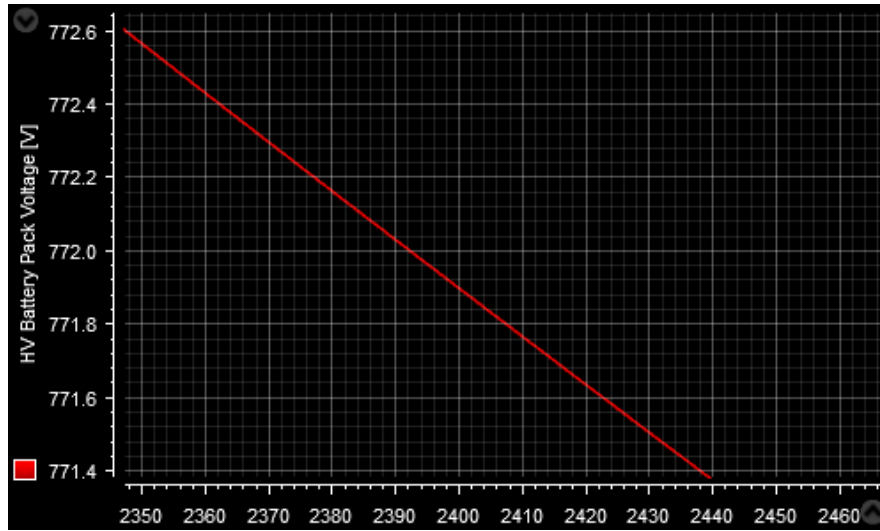


Fig. 162 HV Battery Pack Voltage during the simulation (From ControlDesk)

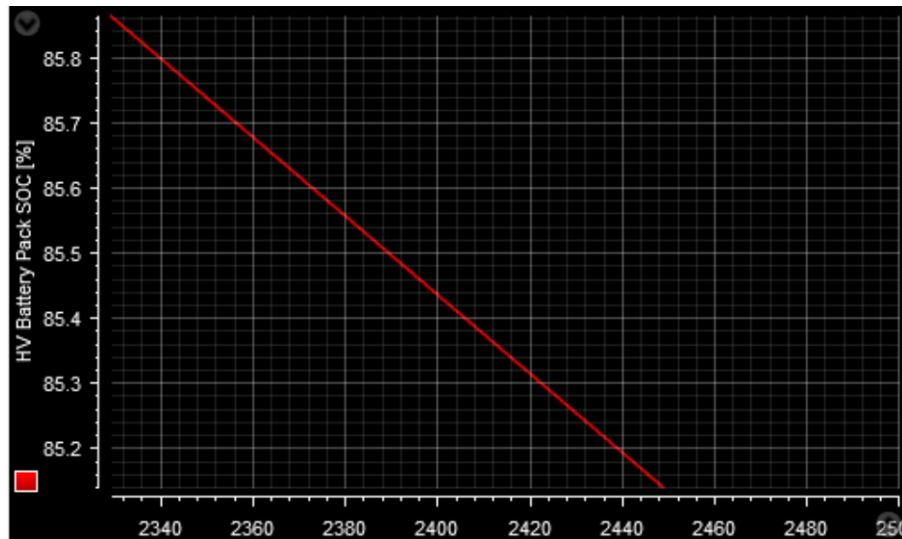


Fig. 163 HV Battery Pack SOC during the real-time simulation (From ControlDesk)

The real-time simulation and the HIL simulators allow to obtain similar results to the in-vehicle tests, but offer the possibility to repeat them, to perform critical tests and to modify them according to the needs of the project with lower costs. The future development of the thesis work depends, of course, on the type of ECU to

which the created model will be connected. The development of the same has been made in such a way that it can easily adapt to any requirement, regardless of the type of battery cell used. Due to the Soft ECU, the model is also adaptable to control units other than the BMS, such as EVCU, IDCM, HCP, TCM or any other ECU that requires signals from the BMS or from a Battery Pack Control Module (BPCM), as it is also called. Moreover, if it is to be tested with a BMS, it is always possible to use only the first core, which concerns the battery pack and which can be easily separated from the second one. Of course, the model can be improved in several ways. For example, using the HIL simulator's power switches, I can create logic that connects the HV battery to the LV battery. Another improvement would be to specify the current supplied or drawn by the battery pack, but with more models to connect to the various parts that might affect this factor, such as the APM, the inverter, the e-motors, and so on. In addition, the thermal model can be implemented by inserting a Soft ECU, which is able to replicate the behaviour of both a heater and a cooler, either with fans or with cooling fluid. The model created is a good starting point to simulate the main functions of a battery pack with a control unit and can be easily adapted if I want to improve its efficiency in future applications.

## 7. BIBLIOGRAPHY

- [1] Bathia et al., «Electrical Fundamentals - Introduction to Batteries» PDH Center, Fairfax, 2012.
- [2] S. Musumeci, «Energy sources in HEVs and EVs,» Politecnico di Torino, Dipartimento Energia "G. Ferraris", Torino, 2022.
- [3] R. M. Laine et al., Battery Basics, University of Michigan, n.d.
- [4] «LibreTexts Chemistry» 2022. [Online]. Available: [https://chem.libretexts.org/Bookshelves/General\\_Chemistry/Chemistry\\_\(OpenSTAX\)/17%3A\\_Electrochemistry/17.5%3A\\_Batteries\\_and\\_Fuel\\_Cells](https://chem.libretexts.org/Bookshelves/General_Chemistry/Chemistry_(OpenSTAX)/17%3A_Electrochemistry/17.5%3A_Batteries_and_Fuel_Cells).
- [5] W. Soutter, «Azonano» 6 July 2012. [Online]. Available: <https://www.azonano.com/article.aspx?ArticleID=3044>.
- [6] M. Wisch et al., «Crash Compatibility and Battery Safety» Eversafe, 2014.
- [7] G. Fisher, «Pouch, Cylindrical or Prismatic: Which Battery Format Will Rule the Market?» 14 April 2021. [Online]. Available: <https://www.addionics.com/post/pouch-cylindrical-or-prismatic-which-battery-format-will-rule-the-market>.
- [8] S. Melançon, «Prismatic cells vs. cylindrical cells: What is the difference?» 25 April 2022. [Online]. Available: <https://www.laserax.com/blog/prismatic-vs-cylindricalcells#:~:text=A%20prismatic%20cell%20is%20a,stacked%20or%20rolled%20and%20flattened>
- [9] S. Melançon, «Pouch Cell Assembly: New Methods to Improve Manufacturing» 13 January 2022. [Online]. Available: <https://www.laserax.com/blog/pouch-cell-assembly>.
- [10] S. Muslimin et al., «Comparison of Batteries Used in Electrical Vehicles» University of Sreiwijaya, Palembang, 2021.
- [11] R. A. Hanifah et al., «Electric Vehicle Battery Modelling and performance Comparison in Relation to Range Anxiety» Elsevier, Kuala Lumpur, 2015.

- [12] F. Baronti et al., «Experimental characterization of a commercial Sodium-Nickel chloride battery for telecom applications» University of Pisa, Pisa, 2019.
- [13] S. Vashista, «Lithium-ion vs Lithium-Polymer Batteries» 14 March 2022. [Online]. Available: <https://www.linkedin.com/pulse/lithium-ion-vs-lithium-polymer-batteries-sumit-vashista/>.
- [14] Inverted, «Types of Lithium-ion Batteries available in the market» December 2021. [Online]. Available: <https://inverted.in/blog/types-of-lithium-ion-batteries-available-in-the-market>.
- [15] Dragonfly Energy, «A guide to the 6 main types of lithium batteries» 27 September 2021. [Online]. Available: <https://dragonflyenergy.com/types-of-lithium-batteries-guide/>.
- [16] C. D. Tschritter et al., «Battery Management System (BMS) Test Stand Utilizing a Hardware-in-the-Loop (HIL) Emulated Battery» University of Texas at Arlington, Arlington, 2021.
- [17] A. Heuberger, «Flexible Battery Management System with Active Cell Balancing» Fraunhofer IIS, Erlangen.
- [18] A. Ramanath, «Active and Passive Battery Pack Balancing Methods» Eepower, 24 November 2020. [Online]. Available: <https://eepower.com/technical-articles/active-and-passive-battery-pack-balancing-methods/#>. [Consulted 20 December 2022].
- [19] W. Kai et al., «State of Charge (SOC) Estimation of Lithium-ion Battery Based on Adaptive Square Root Unscented Kalman Filter» Foshan University, Foshan, 2020.
- [20] L. Zhang et al., «Comparative Research on RC Equivalent Circuit Models for Lithium-Ion Batteries of Electric Vehicles» University of Science and Technology Beijing, Beijing, 2017.
- [21] B. University, «How does Internal Resistance affect Performance?» Battery University, 17 September 2021. [Online]. Available:

<https://batteryuniversity.com/article/how-does-internal-resistance-affect-performance>. [Consulted on 11 December 2022].

- [22] M. Razi et al., «A Review of Internal Resistance and Temperature Relationship, State of Health and Thermal Runaway for Lithium-ion Battery Beyond Normal Operating Condition» Semarak Ilmu Publishing, Johor, 2021.
- [23] M. Murnane et al., «A closer look at State of Charge (SOC) and State of Health (SOH) Estimation Techniques for Batteries» Analog Devices, Norwood, 2017.
- [24] P. Shen et al., «State of Charge, State of Health and State of Function Co-estimation of Lithium-ion Batteries for Electric Vehicles» Tsinghua University, Pechino, 2016.
- [25] K. Mawonou et al., «State-of-health estimators coupled to a random forest approach for lithium-ion battery aging factor ranking» Elsevier, Gif-sur-Yvette, 2021.
- [26] X. Shu et al., «State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives» Elsevier, Kunming, 2021.
- [27] P. Keil et al., «Thermal Impedance Spectroscopy for Li-Ion Batteries with an IR Temperature Sensor System» World Electric Vehicle Journal, Barcelona, 2013.
- [28] K.-H. Chen et al., «Air cooling concepts for Li-Ion Battery Pack in cell level» Summer Heat Transfer Conference, Bellevue, 2017.
- [29] D. Galatro et al., «Thermal behavior of Lithium-Ion batteries: aging, heat generation, thermal management and failure» Global Digital Central, Toronto, 2020.
- [30] R. Bubbico et al., «Thermal Model of Cylindrical Lithium-Ion Batteries» The Italian Association of Chemical Engineering, Rome, 2019.
- [31] Infineon Technologies AG, «TLE9012DQU, TLE9015DQU - User manual» Infineon Technologies AG, Munich, 2022.
- [32] Infineon Technologies AG, «TLE9012DQU - Li-ion battery monitoring and balancing IC» Infineon Technologies AG, Munich, 2022.



## 8. LIST OF FIGURES

Fig. 1. Cross section of a zinc-carbon dry cell ( <a href="https://chem.libretexts.org">https://chem.libretexts.org</a> ) .....	7
Fig. 2. Discharge and charge mode of a secondary cell (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	8
Fig. 3. Schematic illustration of a hydrogen fuel cell (Slaughter, Gymama, Enzymatic Glucose Biofuel Cell and its Application, Baltimore, 2015) .....	9
Fig. 4. Electrons, water and heat production in a fuel cell (Musumeci, Salvatore, Energy sources in HEVs and EVs, Turin, 2022) .....	10
Fig. 5. Schematic view of an ultracapacitor (Wasterlain, Sébastien, Hybrid power source with batteries and supercapacitor for vehicle applications, Paris, 2006).....	11
Fig. 6. Example of an ultracapacitor pack ( <a href="https://maxwell.com">https://maxwell.com</a> ) .....	11
Fig. 7. Illustration of how battery cells are arranged into modules and then assembled into battery packs ( <a href="https://www.everergi.com/electric-battery-management">https://www.everergi.com/electric-battery-management</a> ) .....	13
Fig. 8. Floor integration configuration in the Bolloré Bluecar (Wisch, Marcus, Crash compatibility and battery safety, EVERS SAFE, 2014) .....	13
Fig. 9. Tunnel mounting configuration in the Chevrolet Volt (Wisch, Marcus, Crash compatibility and battery safety, EVERS SAFE, 2014) .....	14
Fig. 10. Rear mounting configuration in the Tesla Roadster (Wisch, Marcus, Crash compatibility and battery safety, EVERS SAFE, 2014) .....	14
Fig. 11. Platform configuration in the Tesla model S (Wisch, Marcus, Crash compatibility and battery safety, EVERS SAFE, 2014) .....	15
Fig. 12. Composition of a cylindrical cell (Lebel, Félix-Antoine, Méthode de dimensionnement et modélisation de batteries lithium-ion, Sherbrooke, 2017).....	15
Fig. 13. Rolled up and stacked layers in prismatic cells (Li, Jianlin, From Materials to Cell: State-of-the-Art and Prospective Technologies for Lithium-Ion Battery Electrode Processing, Oak Ridge, 2021).....	16
Fig. 14. Example of prismatic cell structure (Zschornak, Matthias, Fundamental principles of battery design, Freiberg, 2017) .....	16

Fig. 15. Structure of a pouch cell (Yoo, Sunggoo, Analysis of Pouch Performance to Ensure Impact Safety of Lithium-Ion Battery, Gunsan, 2019) .....	17
Fig. 16. Schematic view of a pouch cell (Zhu, Pengcheng, A review of current collectors for lithium-ion batteries, Coventry, 2020).....	18
Fig. 17. Electrode production for a pouch cell ( <a href="https://djausa.com/djausa/automotive-assembly-equipment/lib/">https://djausa.com/djausa/automotive-assembly-equipment/lib/</a> ).....	19
Fig. 18. Final steps to produce a pouch cell ( <a href="https://djausa.com/djausa/automotive-assembly-equipment/lib/">https://djausa.com/djausa/automotive-assembly-equipment/lib/</a> ).....	19
Fig. 19. Battery module consisting of pouch cells and metal plates (Murashko, Kirill, Thermal modelling of commercial lithium-ion batteries, Kuopio, 2016).....	20
Fig. 20. Energy density comparison of different batteries ( <a href="https://inventuspowers.com/lithium-ion-power-for-data-storage-and-servers/">https://inventuspowers.com/lithium-ion-power-for-data-storage-and-servers/</a> ) .....	22
Fig. 21. Comparison between different types of cathodes in Li-ion batteries (Salgado, Rui, The Latest Trends in Electric Vehicles Batteries, Porto, 2021) .....	27
Fig. 22. Overview of the V-shaped workflow in Model-Based Design (Bergström, David, Model- and Hardware-in-the-Loop Testing in a Model-Based Design Workflow, Lund, 2016).....	28
Fig. 23 HIL testing concept ( <a href="https://www.rinf.tech/why-is-hardware-in-the-loop-hil-testing-important-for-automotive-solutions/">https://www.rinf.tech/why-is-hardware-in-the-loop-hil-testing-important-for-automotive-solutions/</a> ).....	29
Fig. 24 Diagram of the relationship between a control unit and its sensors and actuators ( <a href="https://www.manufacture.create.aau.dk/resources-test/sensors/">https://www.manufacture.create.aau.dk/resources-test/sensors/</a> ).....	30
Fig. 25 Example of a throttle valve for automotive applications (Bischoff, Bastian, Learning Throttle Valve Control Using Policy Search, Munich, 2013) .....	32
Fig. 26. Connections to a SCALEXIO system (Help Desk, SCALEXIO dSpace).....	33
Fig. 27. Detailed view of the SCALEXIO simulator (Help Desk, SCALEXIO dSpace) .....	34
Fig. 28. Back of a SCALEXIO system (Help Desk, SCALEXIO dSpace) .....	35
Fig. 29 SCALEXIO Processing Unit (Help Desk, SCALEXIO dSpace) .....	35
Fig. 30 Power supply unit (Help Desk, SCALEXIO dSpace) .....	36
Fig. 31. Connectors in the DS2680 I/O Unit (Help Desk, SCALEXIO dSpace).....	37

Fig. 32. Power Switch channel circuit (Help Desk, SCALEXIO dSpace) .....	38
Fig. 33. Analog In 1 channel circuit (Help Desk, SCALEXIO dSpace).....	39
Fig. 34 Analog In 2 channel circuit (Help Desk, SCALEXIO dSpace).....	40
Fig. 35. Digital In 1 channel circuit (Help Desk, SCALEXIO dSpace) .....	40
Fig. 36. Flexible In channel circuit (Help Desk, SCALEXIO dSpace) .....	41
Fig. 37. Analog Out 1 channel circuit (Help Desk, SCALEXIO dSpace) .....	41
Fig. 38 Analog Out 2 channel circuit (Help Desk, SCALEXIO dSpace).....	42
Fig. 39 Analog Out 3 channel circuit (Help Desk, SCALEXIO dSpace).....	42
Fig. 40 Analog Out 4 channel circuit (Help Desk, SCALEXIO dSpace).....	43
Fig. 41. Digital Out 1 channel circuit (Help Desk, SCALEXIO dSpace).....	44
Fig. 42. Resistance Out channel circuit (Help Desk, SCALEXIO dSpace).....	44
Fig. 43 BMS architecture ( <a href="https://www.mouser.it/new/stmicroelectronics/stm-battery-management-system-solution/">https://www.mouser.it/new/stmicroelectronics/stm-battery-management-system-solution/</a> ) .....	46
Fig. 44 Wiring overview of Orion BMS (Ewert Energy Systems, Orion BMS wiring and Installation Manual, 2018) .....	47
Fig. 45 Passive (left) and active (right) balancing circuit (Barreras, Jorge, Multi-Objective Control of Balancing Systems for Li-Ion Battery Packs: A Paradigm Shift?, Aalborg, 2014) .....	48
Fig. 46 Battery pack in an electrical powertrain (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	49
Fig. 47 Equivalent circuit of battery according to the Rint model (Hossain, Eklas, A Comprehensive Review on Second-Life Batteries: Current State, Manufacturing Considerations, Applications, Impacts, Barriers & Potential Solutions, Business Strategies, and Policies, Khulna, 2019) .....	50
Fig. 48 Equivalent circuit of battery according to the Thevenin model (Ding, Xiaofeng, An improved Thevenin model of lithium-ion battery with high accuracy for electric vehicles, 2019).....	51
Fig. 49 Equivalent circuit of battery according to the PNGV model (Saldana, Gaizka, Analysis of the Current Electric Battery Models for Electric Vehicle Simulation, Bilbao, 2019).....	51

Fig. 50 Equivalent circuit of battery according to the GNL-model (Kai, Wang, State of Charge (SOC) Estimation of Lithium-ion Battery Based on Adaptive Square Root Unscented Kalman Filter, Foshan, 2019).....	52
Fig. 51 Simplified battery Thevenin circuital model (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	53
Fig. 52 R0, R1, R2, C1, C2 blocks in Simulink (From Simulink model).....	53
Fig. 53 NCR18650PF Lithium-ion battery specifications (Panasonic, Datasheet NCR18650PF, 2016).....	54
Fig. 54 Computation of C1 and C2 in Simulink (From Simulink model) .....	55
Fig. 55 Computation of R1 and R2 in Simulink (From Simulink model) .....	55
Fig. 56 Computation of R0 in Simulink.....	56
Fig. 57 SOC variation and self-discharge circuit (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	57
Fig. 58 Usable capacity considering the cycle number and temperature of the battery cell (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	58
Fig. 59 Behaviour of the self-discharge circuit (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	58
Fig. 60 Voltage-current characteristic in Thevenin equivalent circuit (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	59
Fig. 61 Block diagram of the current through a hybrid vehicle (Kumar, Siddhant, Battery charging topology, infrastructure, and standards for electric vehicle applications: A comprehensive review, Mandi, 2021).....	60
Fig. 62 Input current and Current Factor for the computation of SOC (From Simulink model) .....	60
Fig. 63 Terminal voltage block in Simulink (From Simulink model) .....	61
Fig. 64 Computation of $V_{bat}$ in Simulink (From Simulink model) .....	62
Fig. 65 Voltage trend with respect to current pulses (Musumeci, Salvatore, Energy sources in HEVs and EVs, Torino, 2022) .....	63
Fig. 66 Terminal voltage graph of a single battery cell (From Simulink model) .....	63

Fig. 67 Stored energy status of a battery (Espedal, Ingvald, Current trends for State-of-Charge (SoC) Estimation in Lithium-Ion Battery Electric Vehicles, Trondheim, 2021) .....	64
Fig. 68 SOC block in Simulink (From Simulink model) .....	65
Fig. 69 Computation of SOC with Coulomb counting method (From Simulink model) .....	66
Fig. 70 SOC graph obtained using the Coulomb counting method (From Simulink model) .....	67
Fig. 71 Jump error obtained by changing between charge and discharge mode (From Simulink model).....	68
Fig. 72 StateFlow model to indicate the change from the initial and current SOC value in the SOC estimation (From Simulink model).....	68
Fig. 73 SOC computation with Voltage method (From Simulink model).....	69
Fig. 74 SOC graph obtained using the Voltage method (From Simulink model).....	70
Fig. 75 Kalman filter principle (Murnane, Martin, A closer look at State of Charge (SOC) and State of Healt (SOH) estimation techniques for batteries, Norwood, 2021) .....	71
Fig. 76 Kalman filter algorithm (Murnane, Martin, A closer look at State of Charge (SOC) and State of Healt (SOH) estimation techniques for batteries, Norwood, 2021) .....	74
Fig. 77 SOH block in Simulink (From Simulink model).....	75
Fig. 78 Representation of battery SOH (Selvabharathi, D., Battery health and performance monitoring system: a closer look at state of health (SoH) assessment methods of a Lead-Acid battery, Thanjavur, 2019) .....	76
Fig. 79 Classification and comparison of battery SOH prediction methods (Shu, Xing, State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives, Kunming, 2021) .....	76
Fig. 80 Generic implementation flowchart for SOH prediction using ML algorithms (Shu, Xing, State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives, Kunming, 2021).....	77

Fig. 81 Computation of SOH with internal resistance (From Simulink model) .....	80
Fig. 82 SOH graph obtained using the internal resistance (From Simulink model) ...	80
Fig. 83 Heat generation inside a battery cell (Kantharaj, Rajath, Heat Generation and Thermal Transport in Lithium-Ion Batteries: A Scale-Dridging Perspective, 2019)..	81
Fig. 84 Cell temperature block in Simulink (From Simulink model) .....	82
Fig. 85 Heat generation by convection (ROHM, Basics of Thermal Resistance and Heat Dissipation, 2021).....	83
Fig. 86 Heat generation by radiation (ROHM, Basics of Thermal Resistance and Heat Dissipation, 2021) .....	83
Fig. 87 Power loss computation in Simulink (From Simulink model) .....	84
Fig. 88 Temperature calculation of a single battery cell (From Simulink model) .....	85
Fig. 89 Heat generation by conduction (ROHM, Basics of Thermal Resistance and Heat Dissipation, 2021).....	85
Fig. 90 Example of thermal resistance between multiple materials ( <a href="https://www.thermal-engineering.org/what-is-thermal-resistance-thermal-resistivity-definition/">https://www.thermal-engineering.org/what-is-thermal-resistance-thermal-resistivity-definition/</a> ).....	86
Fig. 91 Cell temperature block in relation to the previous cell in Simulink (From Simulink model).....	87
Fig. 92 Calculation of the temperature of a battery cell in relation to the previous (From Simulink model).....	87
Fig. 93 First cell temperature graph (From Simulink model) .....	87
Fig. 94 Demo kit BMS (Infineon Technologies AG, TLE9012DQU, TLE9015DQU - User manual, Munich, 2022).....	88
Fig. 95 AURIX power supply and communication connection (Infineon Technologies AG, TLE9012DQU, TLE9015DQU - User manual, Munich, 2022) .....	89
Fig. 96 TLE9012DQU and external circuitry (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	90
Fig. 97 IC block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022).....	91

Fig. 98 Cell management unit block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	92
Fig. 99 Voltage measurement long-running mode (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	93
Fig. 100 External balancing device (Infineon Technologies AG, TLE9012DQU - Li- ion battery monitoring and balancing IC, Munich, 2022).....	94
Fig. 101 Balancing and cell voltage measurement (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	95
Fig. 102 PWM balancing function (Infineon Technologies AG, TLE9012DQU - Li- ion battery monitoring and balancing IC, Munich, 2022).....	95
Fig. 103 Passive balancing (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022).....	96
Fig. 104 Power management unit block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	97
Fig. 105 Typical power supply configuration using the internal voltage regulator (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	98
Fig. 106 Temperature measurement unit (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	98
Fig. 107 External temperature measurement (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	99
Fig. 108 TMP triggering (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022).....	100
Fig. 109 Communication interface block diagram (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022) .....	100
Fig. 110 Communication configurations (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022).....	101
Fig. 111 Fault communication (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022).....	101

Fig. 112 Pin configuration (Infineon Technologies AG, TLE9012DQU - Li-ion battery monitoring and balancing IC, Munich, 2022).....	102
Fig. 113 Project page with configured applications (From ConfigurationDesk).....	109
Fig. 114 Project creation and selection menu (From ConfigurationDesk) .....	110
Fig. 115 Application activation and selection menu (From ConfigurationDesk).....	111
Fig. 116 Interfaces within ConfigurationDesk (From ConfigurationDesk).....	111
Fig. 117 Hardware Topology section (From ConfigurationDesk).....	112
Fig. 118 External devices section (From ConfigurationDesk).....	113
Fig. 119 Functions section (From ConfigurationDesk) .....	113
Fig. 120 External device subsystems (From ConfigurationDesk) .....	116
Fig. 121 Configuration of voltage measurement channels (From ConfigurationDesk) .....	117
Fig. 122 Configuration of the balancing channels (From ConfigurationDesk) .....	117
Fig. 123 Configuration of temperature sensor channels (From ConfigurationDesk)	118
Fig. 124 Configuration of the supply channels (From ConfigurationDesk) .....	118
Fig. 125 ControlDesk toolbar (From ControlDesk).....	121
Fig. 126 Low voltage battery display (From ControlDesk).....	121
Fig. 127 Ignition key, pedals, and gears display (From ControlDeks) .....	121
Fig. 128 Speed display in km/h (From ControlDesk).....	122
Fig. 129 Torque display (From ControlDesk).....	122
Fig. 130 Display to change the ambient temperature and ambient pressure (From ControlDesk) .....	122
Fig. 131 Division into two cores of the model (From Simulink model).....	123
Fig. 132 Subsystems of the first core (From Simulink model).....	124
Fig. 133 Balancing voltage subsystem (From Simulink model).....	125
Fig. 134 Balancing voltage computation (From Simulink model) .....	125
Fig. 135 Integration of balancing in the model (From Simulink model).....	126
Fig. 136 Voltage balancing within the model (From Simulink model) .....	126
Fig. 137 Subsystems of the second core (From Simulink model) .....	127
Fig. 138 Minimum SOC computation (From Simulink model).....	128



Fig. 139 Comparison between the various SOC's (From Simulink model) .....	128
Fig. 140 Comparison between the minimum SOC and that of the cell considered (From Simulink model).....	129
Fig. 141 Matlab function for comparison of cell voltages (From Simulink model) .	129
Fig. 142 Switch signals for the balancing (From Simulink model) .....	132
Fig. 143 SOC's graph with active balance (From Simulink model) .....	133
Fig. 144 Terminal voltages graph with active balance (From Simulink model) .....	133
Fig. 145 Temperatures graph with active balance (From Simulink model) .....	133
Fig. 146 Voltage Analog In functions and model ports (From ConfigurationDesk)	136
Fig. 147 Balancing Analog Out functions and model ports (From ConfigurationDesk) .....	137
Fig. 148 Generate New Simulink Model Interface (From ConfigurationDesk) .....	137
Fig. 149 Model Ports in Simulink (From Simulink model).....	138
Fig. 150 Final Simulink model with I/O from ConfigurationDesk (From Simulink model) .....	138
Fig. 151 Build and code generation interface (From ConfigurationDesk) .....	139
Fig. 152 Hypertac wiring configuration.....	141
Fig. 153 Bridge connection between Analog Out and Analog In .....	142
Fig. 154 Oscilloscope connected to the simulator used .....	143
Fig. 155 Value obtained on the multimeter in an instant of time.....	143
Fig. 156 Voltage measurement of the first cell on ControlDesk made during the multimeter test (From ControlDesk).....	144
Fig. 157 Oscilloscope probes connected to the first battery cell channel .....	144
Fig. 158 Voltage trend of the first cell recorded on the oscilloscope .....	145
Fig. 159 Measurements of cell voltage acquired on ControlDesk concurrently with oscilloscope acquisition .....	145
Fig. 160 Voltage trend of the first cell recorded on the oscilloscope without noise.	146
Fig. 161 Measurements of cell voltage acquired on ControlDesk concurrently with oscilloscope acquisition .....	147
Fig. 162 HV Battery Pack Voltage during the simulation (From ControlDesk) .....	148

Fig. 163 HV Battery Pack SOC during the real-time simulation (From ControlDesk)  
..... 148