

POLITECNICO DI TORINO

Corso di Laurea Magistrale in
Ingegneria Elettrica

Tesi di Laurea Magistrale

**Modeling and simulation of the
electric powertrain of a small BEV**



Relatore

Prof. Radu BOJOI

Candidata

Eleonora NACCI

Dicembre 2020

Abstract

The thesis concerns the study, the modeling and the simulation of the energy management system of an electric vehicle powertrain. More specifically this vehicle is a quadricycle L7e, whose name is *Microolino 2.0*.

The aim of this work is to create the powertrain model in order to evaluate the State Of Charge (SOC) of the battery. Model implementation and computations have been made in MATLAB/Simulink framework. Two models are going to be analysed and studied about the two prototypes, entitled *Mule Prototype* and *Prototype 4*.

The thesis is organised as follows:

- *Introduction* chapter with a literature review concerning ICEV problems and consequences on global warming, together with a *Microolino* presentation, story, concept and prospective;
- *Powertrain theoretical background* chapter reports a description of the main Powertrain elements like battery, inverter and electric motor and focus on the main parameters of *Microolino 2.0* powertrain componets;
- *Modeling* chapter is dedicated to the implemented *Simulink* models, describing the single subsystems and the main difference between the two prototypes;
- *Simulation results* chapter is dedicated to the simulation results obtained and model validation, using CAN data on track.
- *Conclusion* chapter reports the thesis conclusion, personal contribution and future work.

The thesis work is made in collaboration with *Bylogix S.r.l.*, a company near Turin operating as a tier1 in automotive field.

Contents

List of Figures	iii
List of Tables	vii
1 Introduction	1
1.1 Electric Vehicles overview	1
1.2 Thesis use case	3
1.2.1 <i>Microfino</i> production plan	5
1.3 <i>Bylogix</i> role in <i>Microfino 2.0</i> development	6
2 Powertrain Theoretical Background	7
2.1 Comparison between ICE and EV powertrain	7
2.2 BEV powertrain components	8
2.2.1 Battery	8
2.2.2 Power Converter	13
2.2.3 Electric Motor	16
3 Modeling	20
3.1 Powertrain Model	20
3.1.1 Electric Model	20
3.1.2 Energy Model	21
3.2 <i>Simulink</i> implementation	22
3.2.1 <i>Mule prototype</i> vs <i>Prototype 4</i>	22
3.2.2 Main <i>Simulink</i> parts	24
3.3 Inputs	25
3.3.1 <i>Mule prototype</i>	25
3.3.2 <i>Prototype 4</i>	26
3.4 Energy Management - BMS	30
3.4.1 Driveline efficiency ($\eta_{driveline}$)	32
3.5 Electric Motor	35
3.6 Inverter	40
3.7 Battery	42
3.7.1 Model structure	42
3.7.2 Battery subsystem inputs	46
3.7.3 <i>Mule prototype</i>	48
3.7.4 <i>Prototype 4</i>	52

4	Simulation results	53
4.1	<i>Mule Prototype</i>	54
4.1.1	Inputs	55
4.1.2	Energy Management - BMS	56
4.1.3	Electric Motor	57
4.1.4	Inverter	59
4.1.5	Battery	61
4.2	<i>Prototype 4</i>	67
4.2.1	Inputs	68
4.2.2	Energy Management - BMS	69
4.2.3	Electric Motor	71
4.2.4	Inverter	73
4.2.5	Battery	74
5	Conclusion	80
5.1	Personal contribution	80
5.2	Future work	81
	Bibliography	82
	Appendices	83
A	List of Acronyms	84
B	List of Symbols	86
C	<i>Prototype 4</i> - simulation results	88
C.0.1	Ride track 2	88
C.0.2	Ride track 3	91
C.0.3	Ride track 4	94
C.0.4	Ride track 5	97

List of Figures

1.1	Pie chart displaying CO_2 emission by different sectors.[1]	1
1.2	<i>Microolino</i> front opening door, key feature.[4]	3
1.3	Iso <i>Isetta</i> 1956 vs <i>Microolino</i> 2016.[6]	3
1.4	<i>Microolino</i> .	4
1.5	<i>Microolino 2.0</i> on the left and <i>Microolino 1.0</i> to the right.	5
1.6	<i>Microolino 2.0</i> production time line.[4]	6
2.1	ICE powertrain components.[9]	7
2.2	BEV powertrain scheme.	8
2.3	Battery basic unit: the cell.[10]	9
2.4	BOSCH 48V-LEM-Battery.	12
2.5	BMZ battery module exploded view.	12
2.6	Power electronic converter base schematic.[10]	13
2.7	DC/AC Inverter schematic block.[10]	14
2.8	TM4 Tautronic™ AC-M1-S/C Low-Voltage Inverter.	15
2.9	Classification of AC motor drives.[10]	16
2.10	Examples of synchronous machine types.[10]	17
2.11	Torque of synchronous machines.[10]	17
2.12	Flux production of synchronous machines.[10]	18
2.13	IPM cross section.[12]	19
2.14	<i>Microolino</i> IPM motor.	19
3.1	Electric power flow in the powertrain.	20
3.2	Energy power flow	21
3.3	Powertrain macro block.	21
3.4	<i>Mule Prototype</i> in <i>Bylogix</i> Workshop.	22
3.5	<i>Prototype 4</i> in <i>Bylogix</i> Workshop.	23
3.6	<i>Simulink</i> general model.	24
3.7	<i>From Workspace</i> Inputs blocks.	25
3.8	Inputs block <i>Simulink</i> model <i>Prototype 4</i> .	27
3.9	Normal Pedal Mapping subsystem in <i>Simulink</i> model.	28
3.10	<i>Simulink</i> model of the regenerative braking part, inside the Normal Pedal Mapping .	29
3.11	Saturation dynamic block icon in <i>Simulink</i> .[13]	30
3.12	Discharge condition relative to the Upper limit .	30
3.13	Charge condition relative to Lower limit .	31
3.14	Scheme structure Energy Management - BMS subsystem.	32
3.15	Driveline efficiency maps.	33

3.16	Driveline efficiency maps 3-D plots. CASE 1, 2, 3, 4, 5	34
3.17	Driveline Efficiency subsystem computation on <i>Simulink</i> model.	34
3.18	Driveline Efficiency blocks computation details.	35
3.19	Electrical Motor general block.	35
3.20	Electric Motor model schematic.	36
3.21	Symplified Electric Motor modeling on <i>Simulink</i>	36
3.22	Example of working point check in the I quadrant.	36
3.23	<i>Simulink</i> working point check model.	37
3.24	Motor Efficiency Map I and IV quadrant.	38
3.25	Electric Motor <i>Efficiency maps</i> 3-D plots. CASE 1, 2, 3, 4, 5.	39
3.26	<i>Simulink</i> model for the Electric Motor efficiency computation.	39
3.27	Inverter subsystem scheme.	40
3.28	Inverter Efficiency Maps for the I and IV quadrant.	40
3.29	Inverter <i>Efficiency maps</i> 3-D plots. CASE 1, 2, 3, 4, 5	41
3.30	<i>Simulink</i> Inverter Efficiency computation.	41
3.31	Detailed battery Thevenin circuital model.[10]	42
3.32	Voltage trends: (a) Open-circuit voltage; (b) Battery voltage and battery current step variation.[10]	43
3.33	Thevenin circuital second order model.[10]	43
3.34	SOC variation and self-discharge circuit schematic blocks.	44
3.35	Voltage-current characteristic.[10]	45
3.36	Schematic block of the voltage-current characteric circuit.[10]	45
3.37	Final simplified Thevenin battery electrical model.[10]	46
3.38	Simplified battery model.[10]	46
3.39	Battery subsystem.	46
3.40	Battery current computation <i>Simulink</i> model.	47
3.41	Modeling of Manual Switches.	48
3.42	<i>Simulink</i> modeling dedicated to the computation of the number of activated modules.	49
3.43	Initial Open-Circuit voltage computation from initial SOC.	49
3.44	Battery module 1 modeling.	49
3.45	Battery model for module 1.	50
3.46	Battery model: V_{OC} LUT.	50
3.47	Battery voltage and SOC computation model.	51
3.48	<i>Simulink</i> battery model <i>Prototype 4</i>	52
3.49	Battery model: V_{OC} LUT.	52
4.1	Track distance covered by <i>Mule Prototype</i>	54
4.2	[<i>Mule Prototype</i>] Input: Reference torque T_m^* and angular speed ω_m waveforms.	55
4.3	[<i>Mule Prototype</i>] Input: Maximum discharge ($I_{discharge,[2s]}$) and charge currents ($I_{charge,[2s]}$) waveforms.	55
4.4	[<i>Mule Prototype</i>] Energy Management - BMS: Output Reference Torque T_m^{**} waveform.	56
4.5	[<i>Mule Prototype</i>] Energy Management - BMS: Comparison between T_m^{**} and T_m^* waveforms	56
4.6	[<i>Mule Prototype</i>] Energy Management - BMS: Motor output power $P_{m,out}$ waveform (electrical convetion).	57

4.7	[<i>Mule Prototype</i>] Electric Motor: Motor Torque T_m waveform.	57
4.8	[<i>Mule Prototype</i>] Electric Motor: Motor Angular Speed ω_m waveform. .	58
4.9	[<i>Mule Prototype</i>] Electric Motor: Motor efficiency η_{motor} waveform. . . .	58
4.10	[<i>Mule Prototype</i>] Electric Motor: Motor input power $P_{m,in}$ waveform (Electric convection.	59
4.11	[<i>Mule Prototype</i>] Inverter: Efficiency $\eta_{inverter}$ waveform.	59
4.12	[<i>Mule Prototype</i>] Inverter: Battery supply power $P_{dc,in}$ waveform. . . .	60
4.13	[<i>Mule Prototype</i>] Inverter: Comparison between $P_{m,in}$ and $P_{dc,in}$ wave- forms.	60
4.14	[<i>Mule Prototype</i>] Battery: Battery Current I_{batt} waveform.	61
4.15	[<i>Mule Prototype</i>] Battery: Battery Voltage V_{batt} waveform.	61
4.16	[<i>Mule Prototype</i>] Battery: <i>Mule Prototype</i> SOC.	62
4.17	[<i>Mule Prototype</i>] Battery: State Of Charge waveform obtained by CAN acquisition on ride track.	62
4.18	[<i>Mule Prototype</i>] Battery: Comparison between model SOC and CAN acquisition SOC.	63
4.19	[<i>Mule Prototype</i>] Battery: Battery Current waveform given by equation 4.1.	64
4.20	<i>Simulink</i> model used to compute the SOC.	64
4.21	[<i>Mule Prototype</i>] Battery: SOC mean values waveform.	64
4.22	[<i>Mule Prototype</i>] Battery: Comparison between CAN SOC, Model SOC and mean values SOC.	65
4.23	SOC model validation.	66
4.24	<i>Club des Miles</i> racetrack on street view.	67
4.25	[<i>Prototype 4</i>] Inputs: Accelerator and Brake pedal position and Vehicle Speed waveforms.	68
4.26	[<i>Prototype 4</i>] Inputs: Maximum discharge and charge currents waveforms.	69
4.27	[<i>Prototype 4</i>] Energy Management - BMS: Reference Torque T_m^* waveform.	69
4.28	[<i>Prototype 4</i>] Energy Management - BMS: Motor Angular Speed ω_m waveform.	70
4.29	[<i>Prototype 4</i>] Energy Management - BMS: saturation action on T_m^* . . .	70
4.30	[<i>Prototype 4</i>] Energy Management - BMS: Torque comparison T_m^* and T_m^{**}	71
4.31	[<i>Prototype 4</i>] Energy Management - BMS: Output motor power $P_{m,out}$ waveform.	71
4.32	[<i>Prototype 4</i>] Electric Motor: Motor torque T_m	72
4.33	[<i>Prototype 4</i>] Electric Motor: Motor Efficiency η_{motor} waveform.	72
4.34	[<i>Prototype 4</i>] Electric Motor: $P_{m,in}$ waveform.	73
4.35	[<i>Prototype 4</i>] Inverter: Efficiency $\eta_{inverter}$ waveform.	73
4.36	[<i>Prototype 4</i>] Inverter: Battery supply power $P_{dc,in}$ waveform.	74
4.37	[<i>Prototype 4</i>] Inverter: Comparison between $P_{m,in}$ and $P_{dc,in}$ waveforms.	74
4.38	[<i>Prototype 4</i>] Battery: battery current I_{batt} waveform.	75
4.39	[<i>Prototype 4</i>] Battery: Battery Voltage V_{batt} waveform.	75
4.40	[<i>Prototype 4</i>] Battery: Model SOC.	76
4.41	[<i>Prototype 4</i>] Battery: State Of Charge waveform obtained by CAN acquisition on ride track.	76
4.42	[<i>Prototype 4</i>] Battery: comparison between model SOC and CAN ac- quisition SOC.	77

4.43	[<i>Prototype 4</i>] Battery: Battery Current waveform given by equation 4.3.	78
4.44	<i>Simulink</i> model used to compute the SOC mean.	78
4.45	[<i>Prototype 4</i>] Battery: SOC mean values waveform.	78
4.46	[<i>Prototype 4</i>] Battery: comparison between CAN SOC, Model SOC and mean values SOC.	79
4.47	[<i>Prototype 4</i>] Battery: SOC model validation.	79
C.1	[<i>Prototype 4</i>]: Model Input waveforms: <i>GasPedalInput</i> , <i>BrakePedalInput</i> , <i>VehicleSpeed</i> .	88
C.2	[<i>Prototype 4</i>]: Reference torque T_* waveform.	89
C.3	[<i>Prototype 4</i>]: Battery current I_{batt} waveform.	89
C.4	[<i>Prototype 4</i>]: Voltage Battery V_{batt} waveform.	89
C.5	[<i>Prototype 4</i>]: Voltage Battery comparison with CAN acquisition waveforms.	90
C.6	[<i>Prototype 4</i>]: SOC waveform comparison between model and CAN acquisition.	90
C.7	[<i>Prototype 4</i>]: SOC validation.	90
C.8	[<i>Prototype 4</i>]: Model Input waveforms: <i>GasPedalInput</i> , <i>BrakePedalInput</i> , <i>VehicleSpeed</i> .	91
C.9	[<i>Prototype 4</i>]: Reference torque T_* waveform.	91
C.10	[<i>Prototype 4</i>]: Battery current I_{batt} waveform.	92
C.11	[<i>Prototype 4</i>]: Voltage Battery V_{batt} waveform.	92
C.12	[<i>Prototype 4</i>]: Voltage Battery comparison with CAN acquisition waveforms.	92
C.13	[<i>Prototype 4</i>]: SOC waveform comparison between model and CAN acquisition.	93
C.14	[<i>Prototype 4</i>]: SOC validation.	93
C.15	[<i>Prototype 4</i>]: Model Input waveforms: <i>GasPedalInput</i> , <i>BrakePedalInput</i> , <i>VehicleSpeed</i> .	94
C.16	[<i>Prototype 4</i>]: Reference torque T_* waveform.	94
C.17	[<i>Prototype 4</i>]: Battery current I_{batt} waveform.	95
C.18	[<i>Prototype 4</i>]: Voltage Battery V_{batt} waveform.	95
C.19	[<i>Prototype 4</i>]: Voltage Battery comparison with CAN acquisition waveforms.	95
C.20	[<i>Prototype 4</i>]: SOC waveform comparison between model and CAN acquisition.	96
C.21	[<i>Prototype 4</i>]: SOC validation.	96
C.22	[<i>Prototype 4</i>]: Model Input waveforms: <i>GasPedalInput</i> , <i>BrakePedalInput</i> , <i>VehicleSpeed</i> .	97
C.23	[<i>Prototype 4</i>]: Reference torque T_* waveform.	97
C.24	[<i>Prototype 4</i>]: Battery current I_{batt} waveform.	98
C.25	[<i>Prototype 4</i>]: Voltage Battery V_{batt} waveform.	98
C.26	[<i>Prototype 4</i>]: Voltage Battery comparison with CAN acquisition waveforms.	98
C.27	[<i>Prototype 4</i>]: SOC waveform comparison between model and CAN acquisition.	99
C.28	[<i>Prototype 4</i>]: SOC validation.	99

List of Tables

- 2.1 BOSCH Battery 12
- 2.2 BMZ Battery 12
- 2.3 *Dana TM4* Low-Voltage Inverter 15
- 2.4 IPM key features 18
- 2.5 *Dana TM4 (SME Line)* Low-Voltage Motor IPM 19

- 4.1 Summary values of 8 time sub-intervals. 63

Chapter 1

Introduction

1.1 Electric Vehicles overview

One of the main problems humanity is facing with concerns global warming, that is correlated with unwanted by-products actually associated with fossil fuel burning due to energy needs. Global warming is a severity consequence of climate change, who reached a threatening level.

As it is known, preventive actions and climate policies are needed to slow down impacts. Pressure on fossil fuels is reched, so most countries are adopt policies of moving towards sustainable, reliable, efficient, economic and green energy resources. Fossil fuels are one of the main threats to Earth's environment as they contribute to many CO_2 emissions. According to the International Energy Agency ¹, it is possible create a pie chart[1] to shows the percentage contribution of. CO_2 emissions by the main sectors in Figure 1.1.

- electricity and heat sector (42%);
- transportation sector (22%);
- industry sector (21%);
- residential sector (6%);
- other sectors (9%).

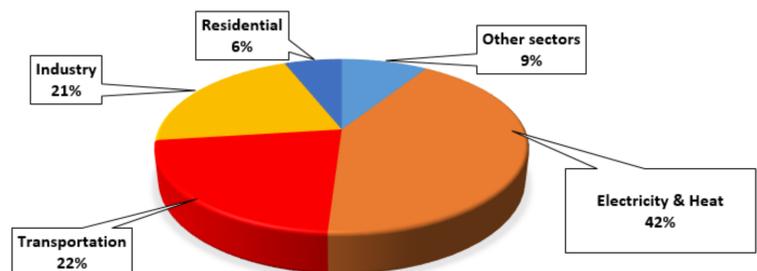


Figure 1.1: Pie chart displaying CO_2 emission by different sectors.[1]

Referring to Figure 1.1, one of the most impacting sectors is transportation, contributing to 22% of the total amount of CO_2 emissions in 2020[1].

¹The **International Energy Agency (IEA)** is a Paris-based autonomous intergovernmental organisation established in the framework of the **Organisation for Economic Co-operation and Development (OECD)** in 1974 in the wake of the 1973 oil crisis. The **IEA** was initially dedicated to responding to physical disruptions in the supply of oil, as well as serving as an information source on statistics about the international oil market and other energy sectors. It is best known for the publication of its annual *World Energy Outlook*.^[15]

Most public and personal vehicles run on Internal Combustion Engines, which can be considered as one of the major causes of climate change.

Indeed, various efforts are being undertaken to reduce the CO_2 emission of the transportation sector, focusing in new fuels development and introduction of clean technology features for vehicles. So the aim is to reduce green-houses gases emissions and also improve vehicle performance.

Transportation electrification is one of the promising approaches with many benefits. Electric Vehicles (EVs) could improve energy security by diversifying energy sources, foster economic growth by creating new advanced industries and most importantly, protect environment by minimizing tailpipe emissions.

EVs show a better performance than internal ICEVs due to the usages of more efficient powertrains and electric motors.

The advancement of EV technology has increased the social and economic benefits in both the transportation and energy sector. On the other hand, EVs do not directly emit CO_2 or are less susceptible to the high oil price. Studies are being carried out to analyze the impact of EVs takeup, with focus on economic, environment and technical issues on power. The EVs impact cost is highly dependent on the generation mix used for EVs. As EVs rely on electricity from power grid to propel, the cost of power generation strongly affects the cost of EV usage.

The economic impact of EV deployment can be evaluated from the view point of power grid and view point of EV owners. Power grid need to have more generation capacity for the additional EV load demand while EVs owners have to pay the high initial purchase cost of EV at the present time. However, with the implementation of coordinated charging, energy trading and various electricity rates policy, EV deployment can be profitable for the operation of power grid and EV owners.

The environmental impact of EVs roll out is subjective. The obvious observation is EVs have zero tailpipe emissions, which is clean and green to environment. However, EVs use electricity generated from power grid and the process of electricity generation does produce GHG emissions. Hence, the environmental impact of EVs usage depends on the sources of electricity. Renewable energy is widely employed lately and has favoured EVs to be more environmental friendly than conventional ICEVs. The interconnection of EVs to powergrid to receive charges raise concerns about the negative impacts of EVs charging on powergrid. Based on extensive search on literature, the anticipated problems associated with EVs charging are harmonics, system losses, voltage drop, phase unbalance, increase of powerdemand, equipment overloading and stability issues grid.

1.2 Thesis use case

The powertrain that is analysed in the thesis belongs to *Microlino*.

The *Microlino* is an all-electric, space saving, lightweight four-wheeled vehicle produced by *Micro Mobility Systems* and designed for urban mobility.

Its key feature is the front opening door, inspired by the *bubble cars* from the late 1950s: it enables users to get out directly onto the pavement after parking. Figure 1.2 displays the vehicle with its key feature



Figure 1.2: *Microlino* front opening door, key feature.[4]

More specifically, *Microlino* was inspired by the Iso *Isetta*² of 1953. Like the *Isetta*, *Microlino* features a single front door to access its interior, which is designed for two people, and a tailgate that opens onto a chest of 300 liters.



Figure 1.3: Iso *Isetta* 1956 vs *Microlino* 2016.[6]

²The **Isetta** is an Italian-designed microcar built under license in a number of different countries, including Argentina, Spain, Belgium, France, Brazil, Germany, and the United Kingdom. Because of its egg shape and bubble-like windows, it became known as a bubble car, a name also given to other similar vehicles. In 1955, the BMW Isetta became the world's first mass-production car to achieve a fuel consumption of 3 L/100 km (94 mpg-imp; 78 mpg-US). It was the top-selling single-cylinder car in the world, with 161,728 units sold. Initially manufactured by the Italian firm Iso SpA, the name Isetta is the Italian diminutive form of Iso, meaning "little Iso".[5]

Microlino is provided of two seats, a trunk, a folding sunroof and it can be charged with either a household or a type 2 plug. The *Microlino* is a small BEV, legally designated and functionally intended as a lightweight quadricycle (L7e³). It aims to combine the advantages of both: a car and a motorcycle.

Based on the approach 'reduce to the max' applied in the engineering manufacturing, the *Microlino* has 50% less parts than a regular car, weighing 513 kg and consuming less energy when driving.

At the 86th Geneva Motor Show in 2016, *Micro Mobility Systems* first showcased the *Microlino* which was developed together with the *Zurich University of Applied Sciences*. The *Microlino* idea originated from Wim, Oliver, and Merlin Ouboter, the Swiss family behind the *Micro Kickscooter*, asking themselves:

How much car does one really need for daily driving?

After finding out that the average car in Switzerland is occupied with 1.6 passengers and covers 36.8 km on an average journey, as well as that car parking spaces are scarce in urban areas; they realized that modern cars are overengineered for urban utilization, especially when one considers environmental factors like global warming.

The Swiss family decided to design a vehicle that would fulfill urban mobility activities while also working towards a more eco-friendly, space-saving, and efficient future. The *Microlino* is designed to be small and manoeuvrable like a motorcycle, but at the same time protected from the weather and with enough space to transport grocery shopping like a normal size car.



Figure 1.4: *Microlino*.

³**Quadricycles (L7e)**, also referred to as Heavy quadricycles, are defined by Framework Directive 2002/24/EC as motor vehicles with four wheels "other than those referred to (as light quadricycles), whose unladen mass is not more than 450 kg (category L7e) (600 kg for vehicles intended for carrying goods), not including the mass of batteries in the case of electric vehicles, with a design payload not more than 200 kg (passenger) or 1000 kg (goods), and whose maximum net engine power does not exceed 15 kW.[7]

1.2.1 *Microolino* production plan

Microolino first version is known as *Microolino 1.0*, while the manufactured and sold version is the *2.0* version: also this thesis study concerns the *Microolino 2.0* prototypes.

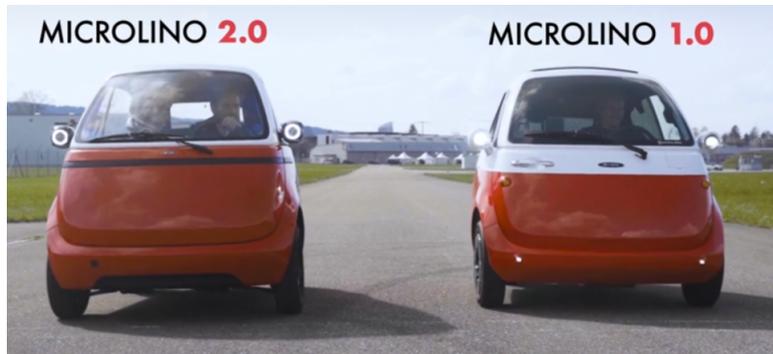


Figure 1.5: *Microolino 2.0* on the left and *Microolino 1.0* to the right.

Around May 2019, Ouboter family searched for a new partners and they contacted directly *Cecomp S.p.A.* Manufacturing Operations Director, through *LinkedIn*. In the same year, a collaboration between *Microolino* and *Cecomp S.p.A.*, an Italian automotive company established in 1978 and part of Turin industrial reality, was started in order to refine both the design as well as the engineering of the *Microolino 1.0* to the more modern *Microolino 2.0*.

Cecomp S.p.A. is also the manufacturing partner for the series production starting in 2021.

Microolino 2.0 development starts with the *Mule Prototype*, first model implemented in this thesis, that was tested in December 2020. Then, since the current year beginning, 2021, there was the '*Prototype phase*'. Indeed, a total of five prototypes was planned that were built iteratively about one month apart from each other. So in February 2021 there was the first 2.0 prototype completion, then in March 2021 the Prototype 2, followed by the third.

Then, the phase, '*Homologation and Pre-series*', started: that includes the homologation process to get the *Microolino* street-legal. Once received the official papers, '*Production Phase*' could start, about the end of current year, ready to deliver the first *Microolinos*.

The family target is to ramp-up the production as fast as possible, while maintaining a good product quality.[4]

The *Microolino 2.0* will come in three trims: Urban, Dolce, and Competizione. The Urban set the price benchmark at 12,000 euros.

Deliveries are expected first in Switzerland, and then Germany, before reaching the rest of Europe.

In Figure 1.6, there is a time line displaying the *Microolino* production plan.

Figure 1.6: *Microlino 2.0* production time line.[4]

1.3 *Bylogix* role in *Microlino 2.0* development

This thesis work is made in collaboration with *Bylogix srl*, a company near Turin operating as a tier1 in automotive field. In *Microlino 2.0* project, it has a development key role: design E/E (electric/electronic) architecture, supply the vehicle ECU (Electronic Control Unit) which operates both as BCM (Body Control Module) and VCU (Vehicle Control Unit) and HMIs (Human Machine Interface) cluster like screen and touchpad.

This thesis is made with the *Bylogix* working team who manage the high-level software development, which targets consists in handle all the functionality such as little part, like position light insertion, but also one pedal driving management and longitudinal vehicle dynamic. Team task is also about different components integration, overview, analysis, control strategies and communication between components. So control strategies stays both auxiliary part and powertrain-driveline. Main instrument employs is *MATLAB-Simulink*: so the team actions consists in simulations, code autogeneration and integration.

Chapter 2

Powertrain Theoretical Background

2.1 Comparison between ICE and EV powertrain

In this second chapter, the theory behind electric Powertrain components is presented. The Powertrain, as the name suggested, provided power to the vehicle: it refers to the set of components that generate the power required to move the vehicle and deliver it to the wheels. EV powertrain is a simple system: comprising of far fewer components compared to a vehicle powered by an ICE.

Referring to Figure 2.1, an ICE vehicle has hundreds of moving parts. Main components of its powertrain are Engine, Transmission and Driveshaft. Power generation is handled by the engine and transmitted to the driveshaft. Other internal parts and components of the engine, differential, axles, emission control, exhaust, engine cooling, system are also included in the powertrain.

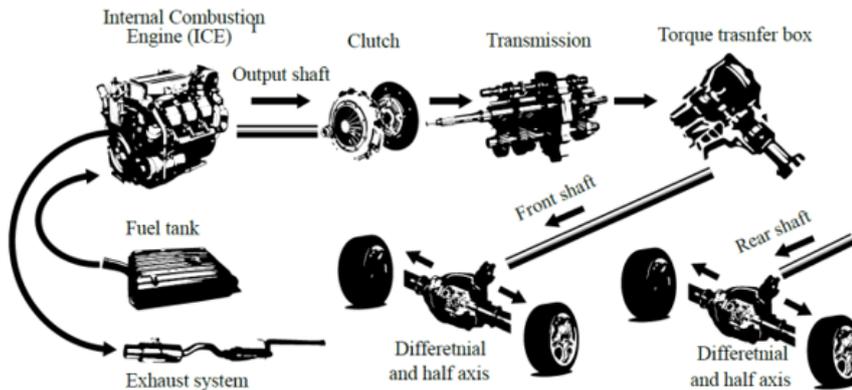


Figure 2.1: ICE powertrain components.[9]

An EV powertrain has 60% fewer components than ICE one. As it is reported in Figure 2.2, the components are mainly four: **Battery Pack**, **Electronic Converter (DC/AC)**, **Electric Machine**, **Reduction Gear**.

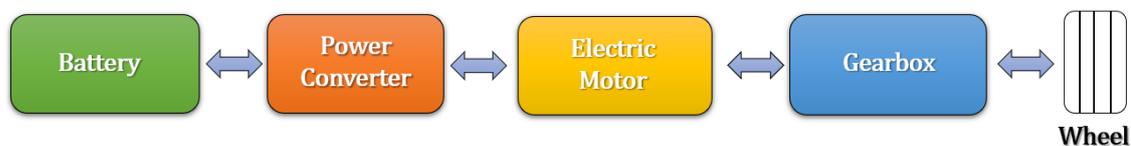


Figure 2.2: BEV powertrain scheme.

In the following chapter sections, a theoretical overview of the PWT main components is presented.

2.2 BEV powertrain components

As shown in Figure 2.2, the main powertrain components are:

- **Source:** holding the energy and electric kind;
- **Converter:** holding the conversion;
- **Motor:** holding the conversion electric-mechanic;
- **Gearbox:** holding the conversion to the wheel, so mechanic-mechanic kind.

An analysis on the three electric components is done in the below subsections.

2.2.1 Battery

In EV or HEV, the battery is an electrochemical storage element representing a bidirectional energy source.

Usually, there are two different batteries installed on the vehicle:

1. **High Voltage (HV) battery** (where voltage is equal or greater than 48 V), this kind of battery primarily roles are
 - provide power to the Electric Motor through the traction inverter;
 - charging by
 - On-Board Charger (OBC) with AC input;
 - Off-board chargers (DC).
2. **Low Voltage (LV) battery** is about
 - 12V for auxiliary services;
 - feeding bidirectional DC/DC converter between LV battery and HV battery as a separated conversion unit.

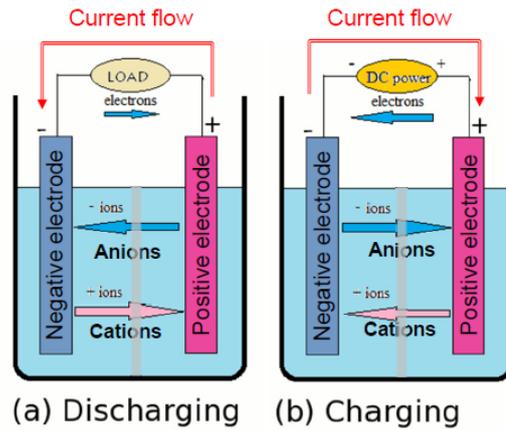


Figure 2.3: Battery basic unit: the cell.[10]

In Figure 2.3 is presented battery basic unit: the cell. It consists of the following elements:

- **Cathode** plate that is a positive electrode;
- **Anode** plate that is a negative electrode;
- **Electrolyte** that allows the movement of cations (positive ions) and anions (negative ions);
- **Separator** that allows ions flowing through it while preventing short-circuit between the electrodes.

Assuming a charged battery, then the potential of the anode is higher than the potential of the cathode.

Figure 2.3 displays two cases, indicated with letters *a* and *b*:

- a. **Discharging condition** is the case when the battery cell is connected to the load. Current flows from the cathode to the anode through the load: the electrons flow from anode to the cathode, creating a charge imbalance that is compensated by the ions inside the cell. Anions flow through the electrolyte towards the anode to donate electrons in an oxidation reaction. Cations flow through the electrolyte towards the cathode to gain electrons in a reduction reaction.
- b. **Charging condition** is the case when current flow reversal due to an external voltage source and the chemical reactions are reversed.

Battery is characterized by the following rated parameters:

- **Capacity [Ah]**: the value of the charge Q that the battery is able to deliver during a complete discharge process;
- **Rated Voltage [V]**: the output voltage by the battery (fully charged);

- **Energy [Wh]**: energy stored in the battery. The rated energy (Wh or kWh) is the product between the rated capacity and rated voltage

$$E_{rated} = V_{batt,rated} \cdot Q \quad (2.1)$$

E_{rated} expressed by equation 2.1 allows an idea of the time needed to recharge a battery of the recharging power is known.

- **State Of Charge (SOC)**: the residual charge of the battery, typically provided in [%] or p.u.

$$SOC(t) = \frac{Q_{res}}{Q_{max}} \cdot 100[\%] \quad (2.2)$$

$$SOC(t) = \frac{Q_{res}}{Q_{max}} \cdot (pu) \quad (2.3)$$

where

Q_{max} is the maximum charge and Q_{res} is the remaining available charge.

Typically, a traction battery SOC is between 10% and 90%.

- **Discharge/recharge rate**, specified as nC , where n is an number typical an integer one, so $n > 0$;
- **Depth of discharge (DOD)** is the amount of depletion that the battery has undergone during a discharge process

$$DOD(t) = \frac{Q_0 - \int_{t_0}^t I_{batt}(\tau) d\tau}{Q_0} \cdot 100(\%) \quad (2.4)$$

where Q_0 is the initial charge, corresponding to SOC = 100% (full charge).

- **Specific power** $[\frac{W}{kg}]$ and **power density** $[\frac{W}{L}]$;
- **Specific energy** $[\frac{Wh}{kg}]$ and **energy density** $[\frac{Wh}{L}]$;

Nowadays, most popular kind of battery is **Li-Ion battery**.

The Li-Ion battery has a higher cell voltage (3.7V) with respect to other technologies, so it is possibile to get HV with less cells in series.

As already mentioned, the cell is the battery basic unit, connection in series or parallel of several cells to get a building block allows to realise what is called battery module. At the end, connection of several modules (typically in series) in an independent housing, called battery pack. The latter is what it called battery in a an electric vehicle.

The traction battery is a battery pack which also contains the **BMS (Battery Management System)** and cooling (mostly liquid).

Traction battery is a complete independent system, characterized by

- **electrical part** contains battery pack, electrical breaker to connect/disconnect the battery pack from outside load (inverter) and precharge of inverter DC link, current and voltage measurement
- **electronic part** contains digital microcontroller with CAN communication with vehicle and measurement of cells voltages for monitoring
- **Cooling** as said mostly liquid cooling or, in some kind of EV, air cooling or no cooling.

Microolino battery

As explained in the introduction chapter 1, *Microolino* development part, referring to the *Microolino 2.0*, starts with *Mulo Prototype*, followed by other Protos from 1 to 5.

In this subsection, attention is turned to the battery typology and specific parameters. First of all, there is a difference: the *Mulo prototype* battery is manufactured by BOSCH with some specifications, while on the other *Protos* the battery is manufactured by BMZ. The difference was due to a commercial and economic choice, due to the fact that BOSCH stopped manufacturing of that battery type.

This distinction is important because thesis *Simulink* implemented models are two: one about the *Mulo prototype* and one about *Prototype 4*.

Summarising the parameters of the two battery types:

- **BOSCH Battery Pack**

Mulo prototype battery pack is composed by six parallel of 48V-LEM-Battery commercialized by BOSCH.

In Table 2.1 are resumed the main parameters of one battery module, used in the *Simulink* model, while in Figure 2.4 is displayed one of 6 six packs mounted on the this prototype.

Table 2.1: 48V-LEM-Battery

<i>Nominal Capacity</i>	50	Ah
<i>I_{Discharge}^{max}</i>	120	A
<i>I_{Charge}^{max}</i>	20	A
<i>V_{nom}</i>	48	V
<i>Weight</i>	≤ 15	kg
<i>Width</i>	260	mm
<i>Lenght</i>	364	mm
<i>Thikness</i>	100	mm
<i>Impedance</i>	≤ 200	mΩ
<i>Energy content</i>	2.4	kWh
<i>Max continuous storage temperature</i>	-30 ~ 40	°C



Figure 2.4: BOSCH 48V-LEM-Battery.

- **BMZ Battery Pack**

As BOSCH stopped the 48V-LEM-Battery manufacturing, the battery chooses for the prototypes "fleet" and for the commercialized of *Microfino* is BMZ battery pack.

BMZ battery pack is composed of two modules.

Table 2.2 summaries the main battery module parameters. Figure 2.5 displays the an exploded view of battery module.

Table 2.2: 1S3P-V6-Module á Terra-E-35E

<i>Nominal Capacity</i>	221	Ah
<i>I_{Discharge}^{max}</i>	250	A
<i>I_{Charge}^{max}</i>	75	A
<i>V_{nom}</i>	48	V
<i>Weight</i>	121	kg
<i>Width</i>	260	mm
<i>Lenght</i>	364	mm
<i>Thikness</i>	100	mm
<i>Impedance</i>	≤ 200	mΩ
<i>Energy content</i>	10.4	kWh
<i>Max temperature</i>	55	°C
<i>Min temperature</i>	-18	°C

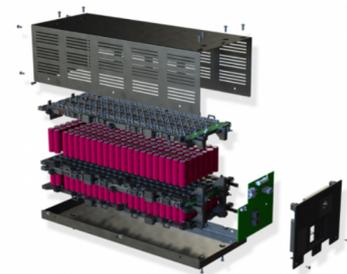


Figure 2.5: BMZ battery module exploded view.

2.2.2 Power Converter

In a EV powertrain, the Battery feeds the power electronics: more specifically the power converter. A power converter is an interfacing element between an electrical energy source and an electrical load. Its main tasks is adapting the parameters of the electrical power provided to the load, according to the load characteristics, and into regulate the power provided to the load.

Power converter types can be

- electromechanical or electromagnetic, that means an electrical machine or system that uses mechanical breakers/switches. Its features are costly, bulky but very robust;
- static converters, without any moving parts.

Power electronic converters are composed by

- **Power unit** that handles the power conversion thanks to
 - **active parts (switches)** whose states are imposed by switching functions and/or the power circuits;
 - reactive parts (such as inductors, capacitors)
 - sensors and protection elements
 - cooling, auxiliary supply
 - power and signal connectors
- **Controller**, constituted by a Microcontroller with analog and digital signal conditioning, implements the regulation law related to the power conversion. Referring to Figure 2.6, the block dedicated to the controller is characterized by
 - **reference command** such as reference torque for traction e-Motors or output current/voltage for battery chargers;
 - **feedback from the system** such as input voltage(s) and current(s), output currents, load feedback (motor position, temperature), converter state signals;
 - **outputs switching functions** to impose the state of switches for the implementation of the power conversion.

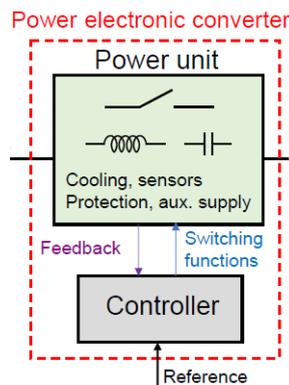


Figure 2.6: Power electronic converter base schematic.[10]

Advantages of power electronic converters respect to the electromechanical converters found high efficiency, lower cost, less bulky, less maintenance (no moving parts). Main drawback is less overload capability respect to the their electromechanical counterparts.

A power converter is always designed for overload condition (max output current) and therefore its rated power is higher than the load rated power.

In a BEV powertrain (shown in Figure 2.2), power converter is a DC/AC converter, known as Inverter. Summing up its key parameters to understand operating principles, in Figure 2.7 is displayed a basic block schematic.

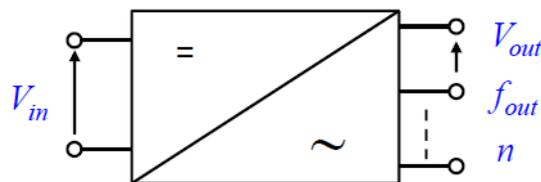


Figure 2.7: DC/AC Inverter schematic block.[10]

- **Input** is the DC voltage $V_{in,dc}$ considered constant slowly changing.
- **Outputs** are $V_{out,rms}$ rms line-to-line voltage (constant or variable), frequency f_{out} (constant or variable), n that is the phases number.

The number of output phases is not limited and can be higher than 3 for multiphase systems.

Microino Inverter

Focus the attention on *Microino* powetrain, its inverter is TM4 Tautronic™ AC-M1-S/C Low-Voltage Inverter, manufactured by *Dana TM4*. This kind of device is ideal for Off-Highway Applications, like the thesis subject.

Dana TM4 inverter provide advanced control of AC synchronous motors for traction or pump functions of any electrical vehicle working with speed or torque control algorithms.

Tautronic is an integrated controller which can manage multi-function and fully configurable I/O pins for any I/O functions like digital and analogue inputs and outputs, capable of driving fans, relays' and hydraulic valves' coils, contactors, negative brakes and many others inductive/resistive loads.

In Table 2.3, there are all the specifications of the TM4 Tautronic™ AC-M1-S/C Low-Voltage Inverter, while in Figure 2.8 is displayed the device.

Table 2.3: TM4 Tautronic™ AC-M1-S/C Low-Voltage Inverter

<i>Nominal Voltage (V_{dc})</i>	36 - 48	V
<i>Input voltage range (V_{dc})</i>	22 - 64.8	V
<i>Continuous current</i>	188 - 250	A_{rms}
<i>Nominal current S2- 2 min</i>	375 - 500	A_{rms}
<i>Output voltage V_{AC}</i>	3 x 0 to 24 (@36 V_{dc})	V
	3 x 0 to 32 (@48 V_{dc})	V
<i>Power terminals</i>	M8(U/V/W/-B), M10(+B)	
<i>Switching frequency</i>	9	kHz
<i>Efficiency</i>	95	%
<i>Output frequency</i>	0-300	Hz
<i>Ambient temperature range</i>	-40 to 55	$^{\circ}C$
<i>Maximum heat-sin temperature</i>		
<i>@ Full current</i>	80	$^{\circ}C$
<i>@ linear de-rated current (down to 50%)</i>	80 - 95	$^{\circ}C$
<i>50% current</i>	95 - 100	$^{\circ}C$
<i>IP protection</i>	IP65	



Figure 2.8: TM4 Tautronic™ AC-M1-S/C Low-Voltage Inverter.

2.2.3 Electric Motor

In an EV, when the driver pushes the accelerator, the battery in the car supplies power to the motor, causing the rotor to turn, and subsequently provide mechanical energy to turn the vehicle's gears. Once the gears are rotating, the wheels turn too.

According to electric machine theory, there are two different type of electric motors: ones operating with alternating current (AC) and ones in direct current (DC). Nowadays, DC motors can be found in an electric vehicle, but only as small, mini motors used, for the windshield wipers and electric windows, for example, but not as traction motors. For the traction of an electric vehicle, an AC motor is used.

Two types of AC electric motors are mainly used as traction motors: Asynchronous (as known as Induction) and Synchronous.

In the Figure below (2.9), the classification of AC motor drives is shown:

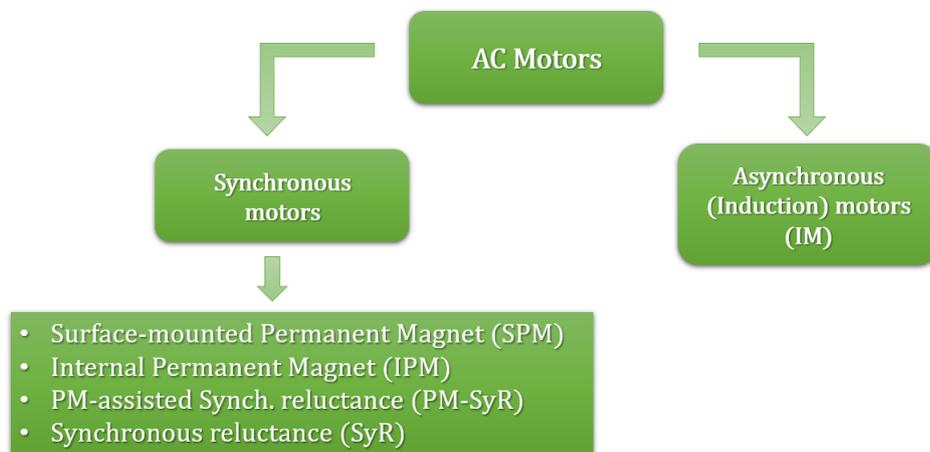


Figure 2.9: Classification of AC motor drives.[10]

An EM basically consists of two main parts: **stator**, that is stationary part of a rotary system, and **rotor**, the moving component of an electromagnetic system. The stator houses the phase windings, that are sets of insulated copper coils placed in slots, according to two main technologies: distributed or concentrated. The motional part structure define the distinction between electrical motor types cited in Figure 2.9.

The focus of the thesis is on synchronous motors, due to the electric motor mounted on *Microfino* powertrain.

In **synchronous motors** the rotor position must be measured (through sensors like encoder, resolver) for synchronization purposes. Stator and rotor cores must be laminated. For synchronous machines, independently of the stator winding (distributed/concentrated), the rotor design imposes the motor type and name.

In Figure 2.10, there are examples of synchronous machine types.

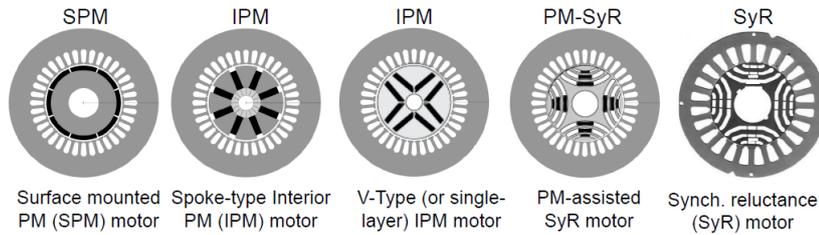


Figure 2.10: Examples of synchronous machine types.[10]

The torque produced by the synchronous motor T_m has two main components:

$$T_m = T_{PM} + T_{rel} \quad (2.5)$$

where

- T_{PM} is the **Permanent Magnets torque**. Each stator current produces a torque component in association with the PM magnetic field along the airgap, basing on Lorentz law.
- T_{rel} is the **Reluctance torque**. If the rotor is anisotropic, i.e. the inductance is not constant along the airgap, the rotor reacts to the stator magnetic field and tends to follow the wave of currents. In other words, the rotor tends to align with the direction of maximum inductance along the magnetic field imposed by the stator, to minimize its magnetic energy.

So also the torque control strategy of synchronous machines depends on the rotor type. The target consists into maximize torque for minimum stator current (and Joule losses).

In Figure 2.11¹ and 2.12, types of synchronous machine are assembled based on structure and torque components.

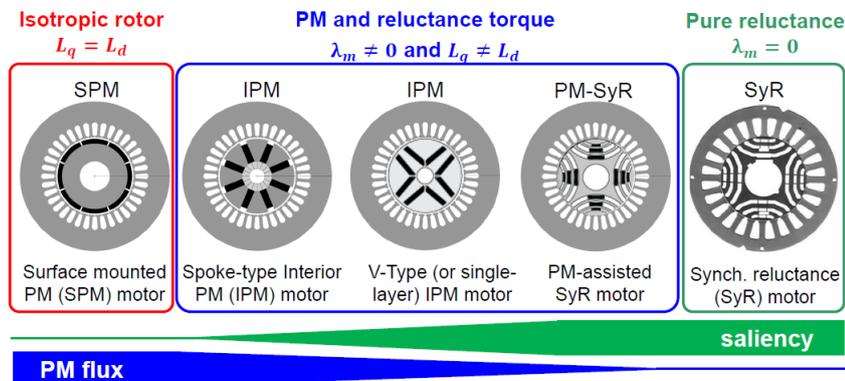


Figure 2.11: Torque of synchronous machines.[10]

¹In Figure 2.11: L_d and L_q represent the inductances and are defined respectively as *direct synchronous inductance* and *synchronous inductance quadrature* of the machine, λ_m represents *PM flux*.

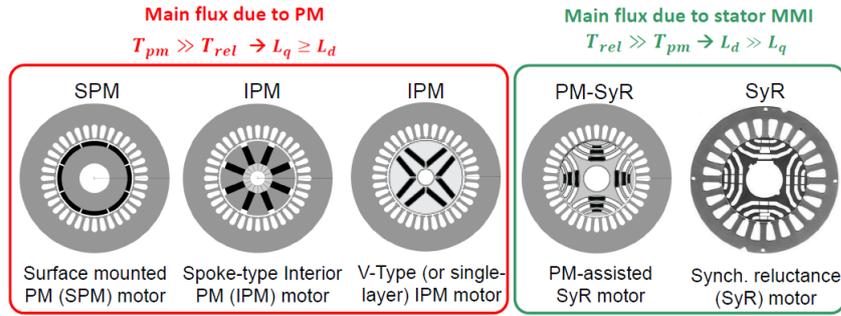


Figure 2.12: Flux production of synchronous machines. [10]

Microfino electric motor

Microfino Electric Motor is an IPM motor. An IPM motor is an AC motor that uses magnets imbedded into the motor's rotor. An **IPM machine** Permanent magnets are of type NdFeB or SmCo, buried in the rotor.

Key features lies into:

- normally high torque density;
- the anisotropy $\xi = \frac{L_q}{L_d}$ is weak but not null;
- both PM and reluctance torque are developed, so the motor torque formulation is given by equation 2.5;
- the torque maximization requires coordinated control of the two current components i_d and i_q .

Table 2.4: IPM key features

$L_q > L_d$	$\xi = 2 \div 3$	$T_{PM} \gg T_{rel}$
-------------	------------------	----------------------

In Figure 2.13, the IPM section is displayed.

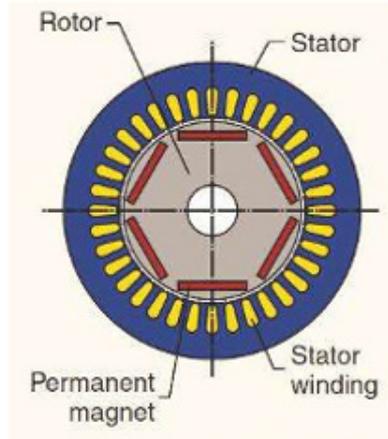


Figure 2.13: IPM cross section.[12]

Focus the attention on *Microlino* powertrain, its Electric Machine is IPM50I125-21T48/54 Low-Voltage Motor, manufactured by *Dana TM4 - SME Line*.

In Table 2.5, there are some main specification of this IPM Motor.

Table 2.5: Dana TM4 Low-Voltage Motor

<i>Technology</i>	SRIPM	
<i>Voltage range (V_{dc})</i>	48 - 144	V
<i>Peak power range</i>	14 - 70	kW
<i>Continous power range [S2, 60min.]</i>	6 - 20	kW
<i>Peak Torque range</i>	100 - 300	Nm
<i>Continous torque range</i>	35 - 90	Nm
<i>Maximum Operating Speed</i>	9.000	rpm
<i>IP rating</i>	20 / 54 / 66	

Figure 2.14 displays the *Microlino* Electric Motor.



Figure 2.14: *Microlino* IPM motor.

Chapter 3

Modeling

The Powertrain modeling, simulation and energy management is the main thesis topic. The modeling is made through *MATLAB/Simulink* environment. Two models have been implemented for the two prototypes:

1. *Mule prototype*;
2. *Prototype 4*.

Prototypes definitions and models differences are described in section 3.2.

3.1 Powertrain Model

First of all, a little digression about distinction between an electric and an energy powertrain model is required, using the power flow.

3.1.1 Electric Model

The power flow sequence is directed from the **Battery** to the **Drive Shaft** through **Inverter** and **Electric Motor**, as shown in Figure 3.1.

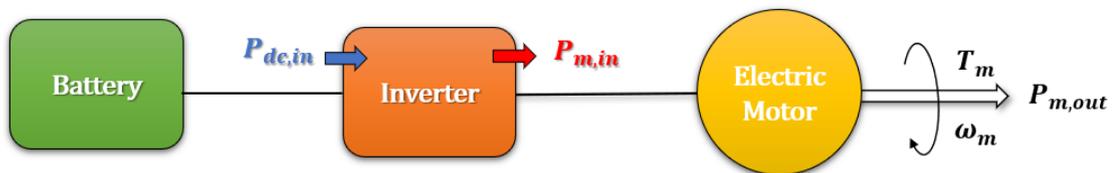


Figure 3.1: Electric power flow in the powertrain.

Referring to Figure 3.1, Battery provides power (denominated $P_{dc,in}$) to Inverter, that in turn provides power to **Electric Motor** (denominated $P_{m,in}$). The latter subsystem supplies power to the Drive Shaft (denominated $P_{m,out}$), in terms of motor torque (T_m) and angular speed (ω_m).

This is the typical powertrain point of view in electric convention, due to the fact that this thesis concerns an energy study: perspective has to be changed.

3.1.2 Energy Model

Switching to an energy perspective, the energy model of the powertrain is presented in Figure 3.2.

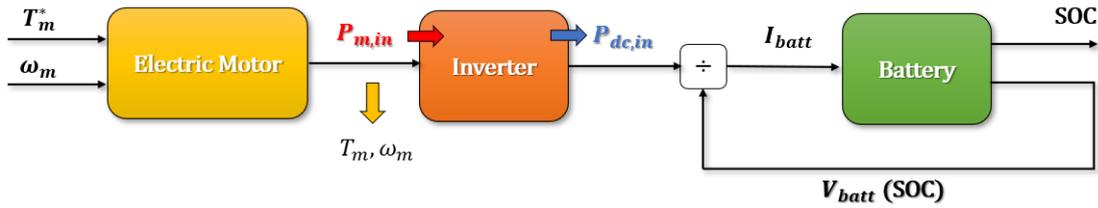


Figure 3.2: Energy power flow

System inputs are the reference motor torque (T_m^*) and angular speed (ω_m). These variables are processed by the first yellow block, **Electric Motor**, that computes a power denominated $P_{m,in}$, due to the choice to main an electric convention, even if in this kind of model it is a block output. Clearly, the motor power is expressed as the product of a motor torque (T_m) and motor speed (ω_m).

The $P_{m,in}$ is the Inverter block input, while as output there is $P_{dc,in}$, still an electric convention is maintain. Dividing $P_{dc,in}$ by the battery voltage, indicated as $V_{batt}(SOC)$ due to its dependency by the SOC, the battery current is computed. The latter is the input for the Battery block.

Therefore, the energy model outputs are the SOC and the battery voltage $V_{batt}(SOC)$, that represents a feedback to compute the battery current.

The blocks in Figure 3.2 can be considered macro blocks, that constitute the main parts of a big block, denominated **Powertrain** (Figure 3.3).



Figure 3.3: Powertrain macro block.

In the successive sections, every block is studied and analysed, presenting the adopted modeling. The models require a torque and angular speed information and the response lies in the SOC computation.

3.2 *Simulink* implementation

The studied powertrains in this thesis are related to the implemented models:

1. *Mulo Prototype*
2. *Prototype n°4*

The differences between the models are:

- battery parameters and model: distinction in BOSCH battery pack for *Mule prototype* and BMZ battery pack for *Prototype 4*
- model inputs

The thesis target lies into get a model who reach the same responses as acquisition obtained with CAN and made on track with the two prototype, during Prototype phase testing.

First of all, a little excursus about definition and difference in the prototypes is done.

3.2.1 *Mule prototype vs Prototype 4*

In the automotive industry, a *Mule prototype* is a testbed vehicle equipped with prototype components requiring evaluation. The *Mule* is necessary because automakers must assess new aspects of vehicles for both strengths and weaknesses before production. A *Mule* is drivable, often pre-production, sometimes years away from realization and coming after a concept car, that preceded the design of critical mechanical components. Usually, the *Mule prototype* may also has advanced chassis and powertrain designs from a prospective vehicle that need testing, which can be effectively concealed in the body and interior of a similarly sized production model. Development mules are often used very heavily during testing and scrapped. Occasionally they are acquired by members of the automaker's engineering team or executives overseeing the design process.



Figure 3.4: *Mule Prototype* in *Bylogix* Workshop.

Prototype 4 is the fourth implemented proto. In automotive industry, it is also known as a *concept car*, a prototype is a vehicle made to show off new developments. These prototypes are not designed for large-scale production. Instead, they are meant to gauge the practicality and popularity of implementing new designs and technologies. Prototypes are a fixture at auto shows, where their futuristic design often draws a crowd: indeed, *Prototype 4* was presented at *IAA Mobility*¹ in Munich on September 2021.



Figure 3.5: *Prototype 4* in *Bylogix* Workshop.

¹The **International MobilityShow Germany** or simply **International Mobility Show**, in German known as the **Internationale Automobil-Ausstellung** (*IAA* – International Automobile Exhibition), is the world’s largest mobility show. It is held annually, with mobility solutions (including passenger vehicles, motorcycles and bikes) being displayed as the *IAA Mobility Show* in odd-numbered years in Munich, and *IAA Transportation* for commercial vehicles in even-numbered years in Hanover, Germany. Before 1991, the show was held solely in Frankfurt. The *IAA* is organized by the *Verband der Automobilindustrie* (*VDA* – Association of the German Automotive Industry) and is scheduled by the *Organisation Internationale des Constructeurs d’Automobiles*. Formerly, the show in Frankfurt occupied twelve buildings.

3.2.2 Main *Simulink* parts

Simulink models structure is more complex than what displayed in Figure 3.2 of Section 3.1.2. Indeed, in Figure 3.6, it is displayed a scheme of the macro blocks modeled in *Simulink* framework.

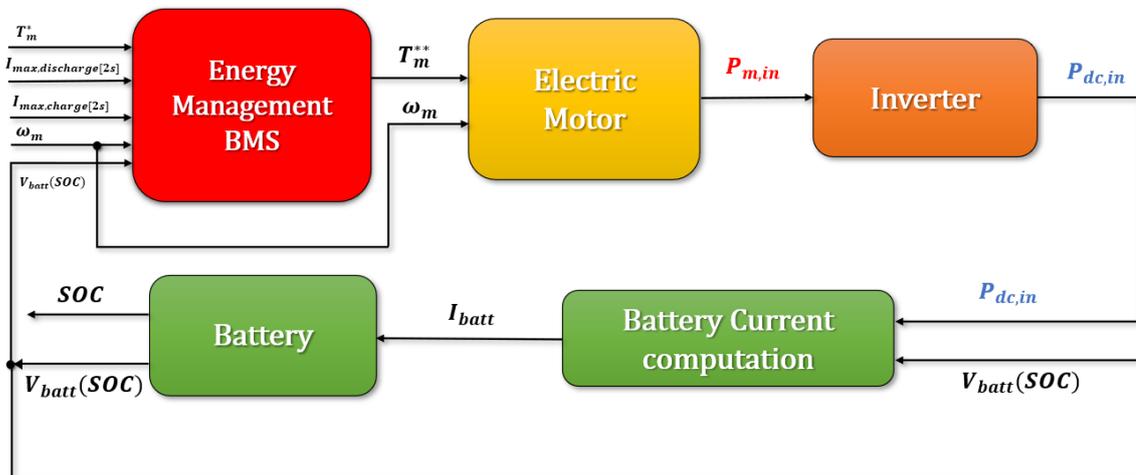


Figure 3.6: *Simulink* general model.

Starting from the left, the models inputs are:

- Reference Torque, indicated as T_m^* ;
- Angular Speed, indicated as ω_m ;
- Maximum Discharge Current, indicated as $I_{discharge,[2s]}$;
- Maximum Charge Current, indicated as $I_{charge,[2s]}$.

The inputs are taken from *Workspace* environment using a dedicated *MATLAB* script. Their values correspond to CAN acquisition data caught during ride track.

Following the power flow, the inputs supply the first block, **Energy Management - BMS**, it can be seen as an energy management's feedback action, where a check on the maximum torque value is analysed, through a saturation: indeed the block output is the **Electric Motor** Reference Torque T_m^{**} . Block details are reported in section 3.4.

Energy Management - BMS output and angular motor speed information feed the yellow block, that stands for **Electric Motor**. Details reported in the section 3.4.

As presented in Figure 3.2, **Electric Motor** block feeds **Inverter**, orange block one (Figure 3.6).

The remaining model part is composed by two green blocks, dedicated to:

- **Battery Current computation;**
- **Battery Model.**

The main Battery model output is the **SOC**, while the other Battery model output is the Battery Voltage V_{batt} that is feedback to **Energy Management - BMS** and **Battery Current - computation**.

In the following Sections, attention is focused on a detailed overview of each blocks created for the two models and, where it is requested, there is a distinction between *Mule* and *Prototype 4* models. So, every subsections below is structured as a block general presentation and then a sub-subsection dedicated respectively to *Mule* and *Prototype 4*.

3.3 Inputs

One of the main differences in the two prototype models lies in the inputs.

3.3.1 *Mule prototype*

About *Mule prototype* model, Inputs are assigned using *Simulink* Data Acquisition Toolbox, *From Workspace*². The latters are denominated using the variable name:

- **Torque_CAN**
- **Speed_CAN**
- **Idischarge_CAN**
- **Icharge_CAN**

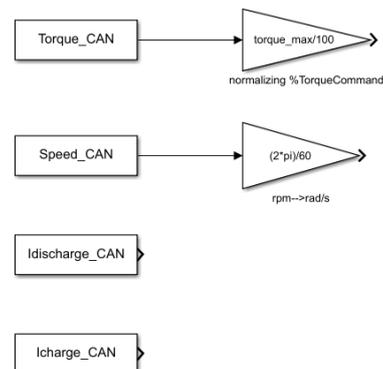


Figure 3.7: *From Workspace* Inputs blocks.

which report respectively Torque, Speed and Discharge/Charge Current [2s] from *MATLAB* script, corresponding to CAN acquisition values. They are data obtained during a track test.

²Definition: the **From Workspace** block reads data into a *Simulink* model from a workspace and provides the data as a signal or bus at the block's output. It is possible load data from the base workspace, model workspace, or mask workspace. It is possible use the From Workspace block to load signal data into any model or subsystem in a model hierarchy from a workspace accessible to the referenced model or subsystem.[13]

In Figure 3.7, the *Mule* model blocks Inputs are reported. Torque and Speed CAN informations pass through a gain block due to the conversion compared to data acquisition.

Indeed, torque CAN data are expressed as percentage of maximum torque, so the gain is formulated as

$$Torque_Gain = \frac{T_{m,max}}{100} \quad (3.1)$$

where $T_{m,max}$ is the maximum available motor torque, equal to 90.0458 Nm.

About speed gain, Speed CAN data is measured as [rpm], so a conversion to $[\frac{rad}{s}]$ is needed:

$$Speed_Gain = \frac{2\pi}{60} \quad (3.2)$$

In this way, all the variables are congruent to enter in the first block: **Energy Management - BMS**.

The reason why Inputs represent one of the main differences between the two models is that *Mule Prototype* CAN data are caught directly on the Electric Motor dedicated CAN.

Due to the fact that *Mule Prototype* is a testing prototype and not the final one, so it does not consider all the accuracies, like pedal mapping or gear ratio. *Mule prototype* task is made a PWT analysis, indeed CAN data are not the ones acquired on the wheels, differently by the Prototype 4 model, which considers also the gear ratio.

3.3.2 Prototype 4

Prototype 4 Simulink model Inputs are displayed in Figure 3.8. Using the *Simulink* Data Acquisition Toolbox, *From Workspace*, Inputs variables turn out to:

- **Gas_Pedal_Input** contains the values representative of the accelerator pedal position during the ride track;
- **Brake_Pedal_Input** is the same as previous input, but contains the values of the brake pedal position during the ride track;
- **Vehicle_Speed** contains the speed simulation values, expressed in $[\frac{km}{h}]$;
- **I_max,discharge[2s]** contains the values representative of the maximum discharge current in 2s accepted by the battery during the racetrack;
- **I_max,charge[2s]** contains the values representative of the maximum charge current in 2s accepted by the battery during the racetrack.

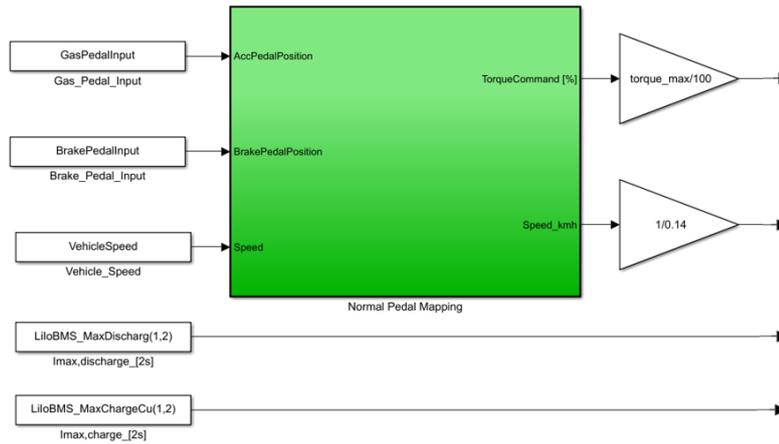


Figure 3.8: Inputs block *Simulink* model *Prototype 4*.

Except for maximum discharge/charge currents, that enter directly in the block **Energy Management - BMS**, the other inputs are used in the green block displayed in Figure 3.8. Green block is a subsystem, denominated **Normal Pedal Mapping**, that generates two signals in output:

- **TorqueCommand[%]**, that is the percentage on the maximum Motor torque. Also for this Prototype, the torque gain shown in Figure 3.8 is formulated as equation 3.1, where $T_{m,max}$ is still 90.0458 Nm , because *Mule prototype* and *Prototype 4* have the same Electric Motor, the IPM machine presented in Chapter 2.
- **Speed_kmh** is the wheels speed, requested for **Normal Pedal Mapping** and, as name suggests, it is expressed in $\frac{km}{h}$. This value is divided by a coefficient equal to 0.14, through a gain block. In this constant value, it is considered the gear reduction ratio between wheels and axle shaft and the conversion from $\frac{km}{h}$ to $\frac{rad}{s}$. In order to obtain a an Angular Speed, that can be used in the **Electric Motor** subsystem.

Entering in the **Normal Pedal Mapping** subsystem, modelization results as shown in Figure 3.9 below.

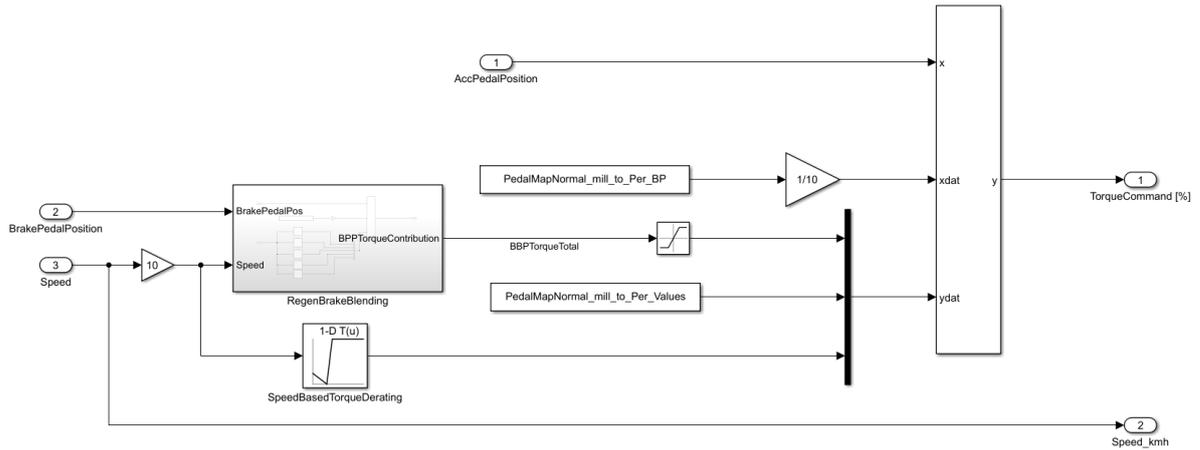


Figure 3.9: **Normal Pedal Mapping** subsystem in *Simulink* model.

Inputs, as explained, are three: accelerator and brake pedal position [*dimensionless*] and wheels speed [$\frac{km}{h}$].

Output are two, but the main computed one is **TorqueCommand[%]**, indicated with number 1. **TorqueCommand[%]** computation is obtained through a LUT Dynamic³, that interpolates the three input data and generates the **TorqueCommand[%]**. The latter represents the reference torque requested at the Electric Motor, the one indicated with T_m^* in Figure 3.6.

The interpolation is reached considering the values in x , $xdat$, $ydat$:

- x represents the first x-coordinate of the dynamic LUT, corresponding to the position of the accelerator pedal;
- $xdat$ represent the thousandth percentage (indeed there is gain of 0.1, used to enter in the LUT Dynamic as percentage value) of accelerator pedal crushing in x-coordinate, that means how much the pedal is crushing;
- $ydat$ returns torque information for the dynamic LUT, computation is obtained using a MUX⁴. Its input signals are three:
 - central one, **PedalMapNormal_mill_to_Per_Values**, is a *From Workspace* block contains values about the accelerant torques.
 - the third one still represents accelerant torques information, but based on a speed derating (1-D LUT in Figure 3.9). It returns a torque as a function of speed. This is done also to act as a torque limiter over $90\frac{km}{h}$ (maximum available vehicle speed).

³The **Lookup Table Dynamic** block computes an approximation of a function $y = f(x)$ using $xdat$ and $ydat$ vectors. The lookup method can use interpolation, extrapolation, or the original values of the input. Using the Lookup Table Dynamic block, it possible change the table data without stopping the simulation.[13]

⁴The **Mux** combines input signals of same data type and complexity into virtual vector.[13]

- the first MUX input represents a braking torque information, coming from the grey block in Figure 3.9, that is a subsystem dedicated to regenerative action of the vehicle.

In Figure 3.10, it is displayed the modelization dedicated to the grey subsystem of Figure 3.9, where an interpolation (still done with a LUT dynamic) based on speed values (x) and brake pedal crushing percentage ($xdat$) is carried out.

y is built using again a MUX, where informations by different LUTs are computed: indeed, the five LUTs in Figure 3.10 have as breakpoints data the brake pedal position and as table data a calibration of different speeds and case (for example the LUT indicated with "Speed1" considers only the accelerator pedal release).

Grey subsystem block output, **BPP_Torque_Contribution**, is a signal that, as shown in Figure 3.9, is passed through a saturation between 0 and -100 and it is the first input of the MUX in Figure 3.9.

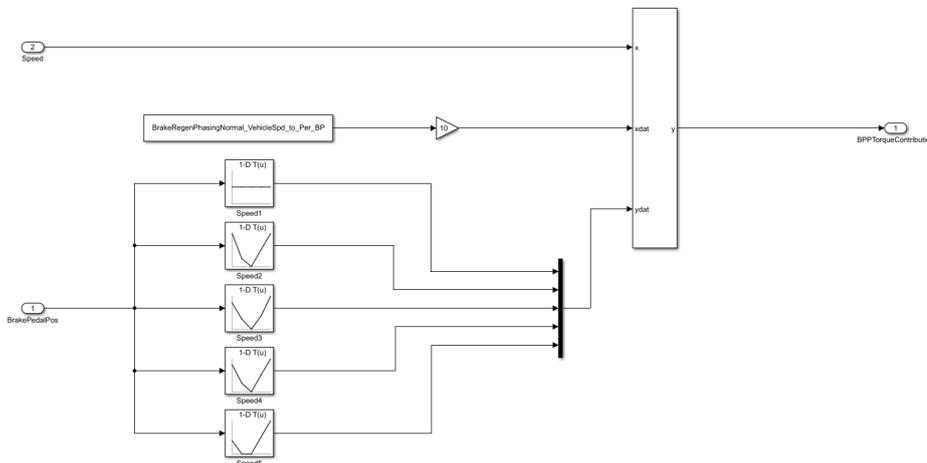


Figure 3.10: *Simulink* model of the regenerative braking part, inside the **Normal Pedal Mapping**.

LUTs in Figures 3.9 and 3.10 are created consider the *creep cost curve* principle: the LUTs are representative of a speed trends that gives back a torque information in acceleration and braking mode, recognising the pedal release condition. Their are made through calibration.

3.4 Energy Management - BMS

This section focus on the first subsystem shows in the general model power flow structure (Figure 3.6): **Energy Management - BMS**, that is the one used to simulate the role of BMS in the vehicle.

The **BMS**, **B**attery **M**anagement **S**ystem, is an acronym for any electronic system that manages a rechargeable battery (cell or battery pack), such as a protection for the battery from operating outside its SOA, monitoring its state, calculating secondary data, reporting that data, controlling its environment, authenticating it and / or balancing it.

Subsystem dedicate to **BMS** configuration is the same in *Mule* and *Forth Prototype* models. It is considered as a *feed-forward* action. Its inputs are

- Reference Torque T_m^{**} reference one, coming from Inputs dedicated part) [Nm];
- Motor Angular Speed ω_m [$\frac{rad}{s}$];
- ischarge/Charge Currents $I_{max,discharge,2s}$ and $I_{max,charge,2s}$ [A];
- Battery Voltage [V].

They are used to generate a saturation on the reference torque T_m^* , which value enters in **Electric Motor** subsystem. Saturation limits are generated considering a power balance in two conditions: discharge and charge. Saturation is made by *Saturation dynamic* block, displayed in Figure 3.11 below.

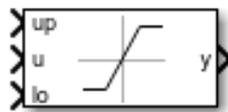


Figure 3.11: Saturation dynamic block icon in *Simulink*.[\[13\]](#)

Referring to Figure 3.11, **up** stays for **Upper limit**, while **lo** for **Lower limit**.

- **Upper limit** is a value of Torque elaborated considering a **Discharge condition**: when $P_m > 0$, power supplies wheels, as displayed in Figure 3.12.



Figure 3.12: Discharge condition relative to the **Upper limit**.

Referring to Figure 3.12, the relative power balance is expressed as:

$$P_m = \eta_{driveline} P_{batt} \quad (3.3)$$

where

- P_m is the Motor Power expressed as product of $T_{m,UP}$, Motor Torque, and ω_m , Motor Speed. $T_{m,UP}$ is the unknown variable, that represents the maximum value assumable by the T_m^{**} , input of the **Electric Motor** subsystem.
- P_{batt} is the Battery Power considered as the product of V_{batt} , battery voltage, and $I_{max,discharge[2s]}$, maximum discharging current in 2 s.

Substitute in equation 3.3 the formulation of P_m and P_{batt} , the power balance become:

$$T_{m,UP} \cdot \omega_m = \eta_{driveline} \cdot V_{batt} \cdot I_{max,discharge[2s]} \quad (3.4)$$

So, the maximum torque limit is obtained as in the formulation below:

$$T_{m,UP} = \frac{\eta_{driveline} \cdot V_{batt} \cdot I_{max,discharge[2s]}}{\omega_m} \quad (3.5)$$

$T_{m,UP}$ is not a constant value, but it is an actual value due to fact that in equation 3.5 the variable V_{batt} , ω_m and $\eta_{driveline}$ are instantaneous during simulation.

- **Lower limit** is the value of Torque elaborated considering a **Charging condition**: when $P_m < 0$, power flowing from wheels to battery, braking condition.



Figure 3.13: Charge condition relative to **Lower limit**.

In this case the power balance is expressed as:

$$P_{batt} = \frac{P_m}{\eta_{driveline}} \quad (3.6)$$

substitute in equation 3.6, the relative formulation, the lower torque limit formulation is

$$T_{m,LOW} = \frac{V_{batt} \cdot I_{max,discharge[2s]}}{\eta_{driveline} \cdot \omega_m} \quad (3.7)$$

In Figure 3.14, the structure of the **Energy Management - BMS** is displayed

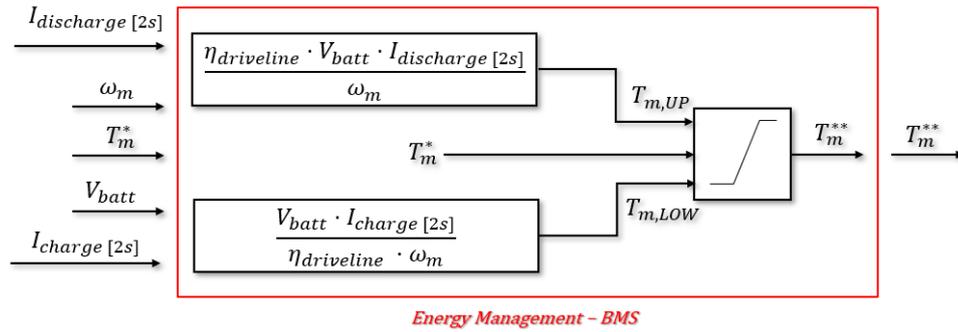


Figure 3.14: Scheme structure **Energy Management - BMS** subsystem.

Referring to Figure 3.14, T_m^* is the actual torque, coming from

- a *From Workspace Simulink* tool in *Mule Prototype* model, multiplied by the torque gain (equation 3.1)
- in *Prototype 4* model is the result of **TorqueCommand**[%], computed by the subsystem **Normal Pedal Mapping**, multiplied by torque gain (equation 3.1)

in both models T_m^* is passing through saturation dynamic block (Figure 3.11) to check that it is an acceptable torque value with electric and energy parameters. So the **Energy Management - BMS** output is a saturated and checked torque value, indicated with T_m^{**} .

3.4.1 Driveline efficiency ($\eta_{driveline}$)

This subsection is dedicated to driveline efficiency computation.

In Figure 3.14, Driveline efficiency is not displayed as subsystem input even if it is used in the limit torques formulation (equations 3.5 and 3.7). This correlated to the fact that driveline efficiency is in function of motor torque and angular speed actual values. It is obtained using an instrument called *Efficiency map*.

An **Efficiency map** is a contour plot on axes of torque and angular speed. It describes the maximum efficiency for any speed/torque combination and is a convenient way to represent the motor drive over a range of operating points defined by a driving cycle.[14]

Of course, *Efficiency maps* refer to Electrical Motor and Inverter, for this thesis it was implemented also an *Efficiency map* for the driveline considering it as product of Electric Motor η_{motor} and Inverter $\eta_{inverter}$ efficiencies:

$$\eta_{driveline} = \eta_{motor} \cdot \eta_{inverter} \quad (3.8)$$

In Figure 3.15, Driveline Efficiency maps for the I and IV quadrant is shown.

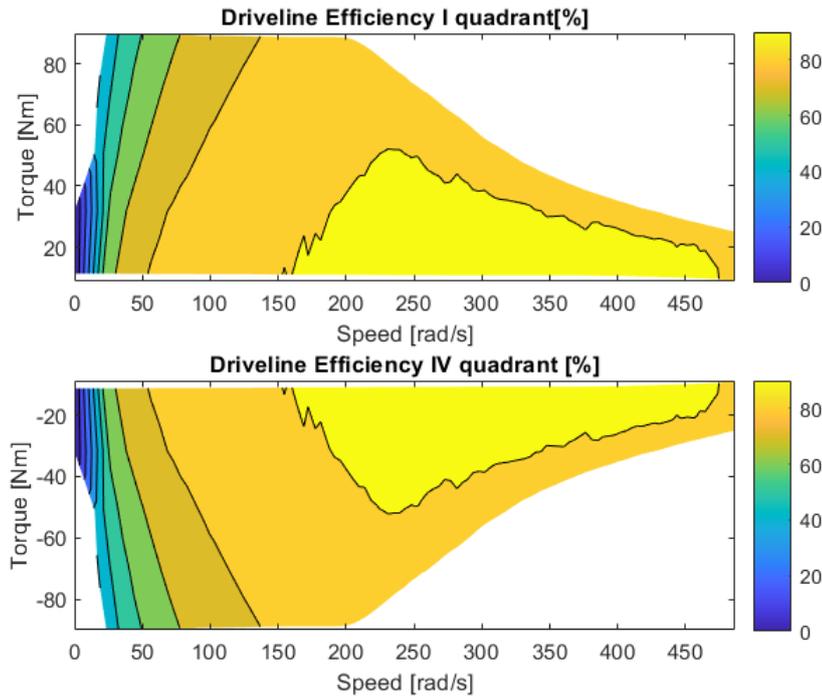


Figure 3.15: Driveline efficiency maps.

A subsystem in the *Simulink* model is dedicated to the $\eta_{driveline}$ computation, resort to *2-D LUTs*. As explained Chapter 2, Electric Motor and Inverter are provided by *Dana TM4* that produces an *Excel* file where are reported all the information in terms of Torque, Speed, Current, Voltage and, thesis concern, Efficiency in five different cases. Every case refers to a RMS current value:

- **CASE 1:** I_{RMS} is $100A_{rms}$
- **CASE 2:** I_{RMS} is $200A_{rms}$
- **CASE 3:** I_{RMS} is $300A_{rms}$
- **CASE 4:** I_{RMS} is $400A_{rms}$
- **CASE 5:** I_{RMS} is $500A_{rms}$

So, the usage of *2-D LUTs* requires the implementation of an Efficiency matrix for every five case in order to obtain a *3-D graphic*. The latter displays on *x-axes* the Motor Angular Speed values expressed in $\frac{rad}{s}$, on *y-axes* the Reference Torque values in *Nm* and on the third axes (*z* one) reports the Driveline efficiencies expressed in percentage terms.

In the Figure 3.16 below, there are reported the five cases *3-D* plots: in the *MATLAB* dedicated script, it is possible create this kind of graphic using the function *surf*.

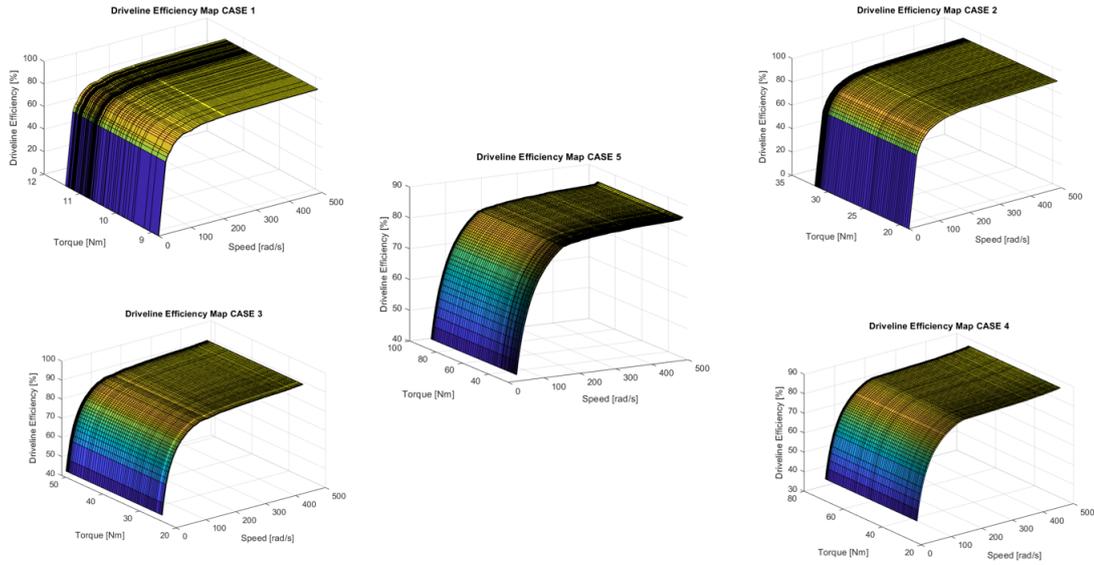


Figure 3.16: Driveline efficiency maps 3-D plots. CASE 1, 2, 3, 4, 5

The Driveline efficiency computation is done with a dedicated *Simulink* subsystem, displayed in Figure 3.17.

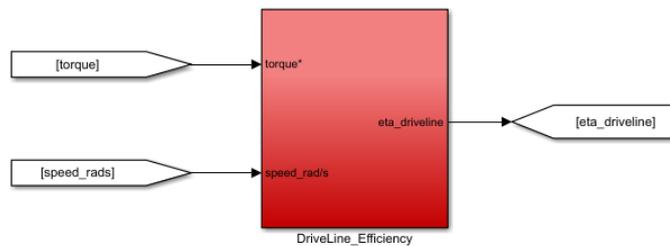


Figure 3.17: Driveline Efficiency subsystem computation on *Simulink* model.

In Figure 3.18, there are displayed the details of the red subsystem in Figure 3.17. As explained, the inputs are respectively Torque and Angular Speed values. The first variable is divided by a constant, indicated as $kt_{med} (\simeq 0.1338)$ to elaborate the corresponding current. The latter value is the green block input, in which the reference case calculation based on the current value is done. Then, a Multiport switch corresponding to reference case determinates the driveline efficiency.

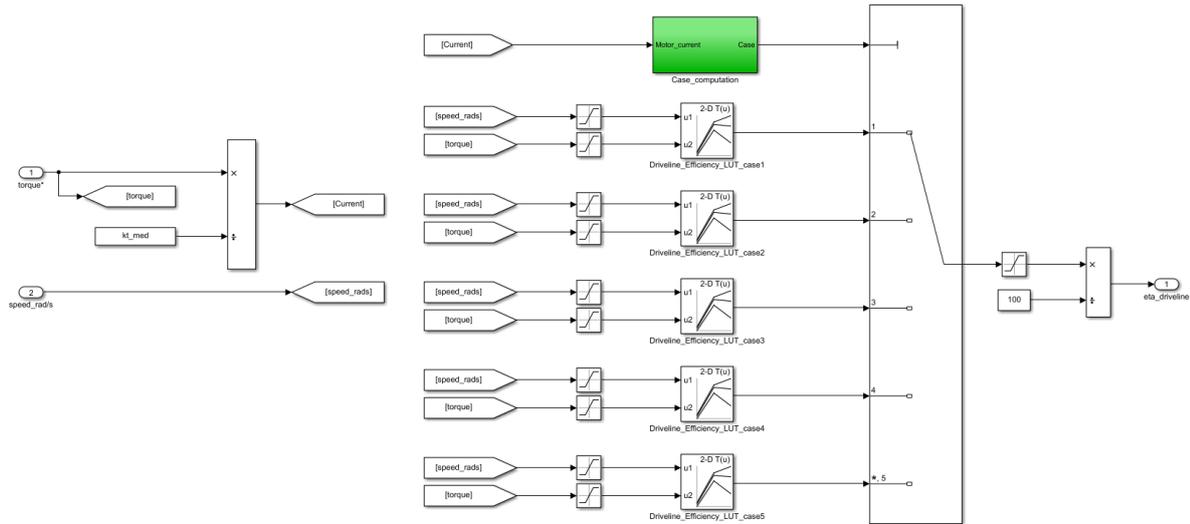


Figure 3.18: Driveline Efficiency blocks computation details.

Driveline efficiency is saturated between a minimum efficiency value, 39.30% (0.3930), to avoid division by zero, and 92.64%, maximum efficiency.

3.5 Electric Motor

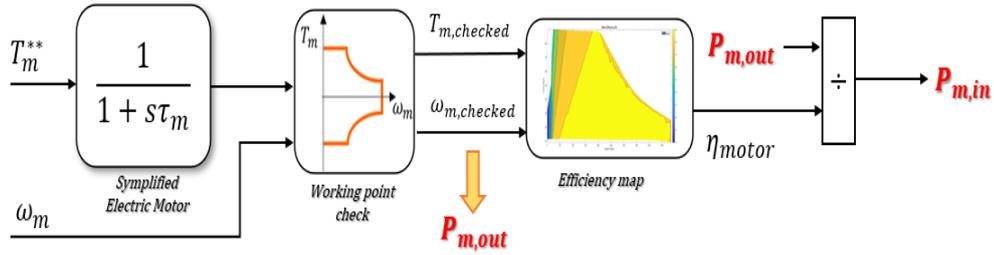
Following the power flow, the **Energy Management - BMS** subsystem outputs, the reference motor torque T_m^{**} and the Angular Speed expressed in $\frac{rad}{s}$, becomes the inputs of the consecutive subsystem **Electric Motor** shown in the general model, displayed in Figure 3.19.



Figure 3.19: Electrical Motor general block.

This subsystem is organised in the below steps:

1. Simplified Electric Motor model
2. Working point check
3. Motor efficiency computation
4. Input meccanical power computation

Figure 3.20: **Electric Motor** model schematic.

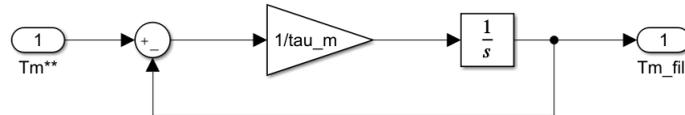
1. Simplified Electric Motor Model

For this thesis, Electric Motor modeling choice is a **LPF** (**L**ow **P**ass **F**ilter). Considering the Laplace domain, the **LPF** formulation transforms the Reference Torque T_m^{**} in a filtered torque $T_{m,filt}$ using the below transfer function

$$\frac{1}{1 + s\tau_m} \quad (3.9)$$

where the time constant τ_m is assumed with a value of 5 ms.

Figure 3.21 displays the **LPF** *Simulink* dedicated part to .

Figure 3.21: Simplified Electric Motor modeling on *Simulink*.

2. Working point check

The Filtered Torque $T_{m,filt}$ and the Motor Angular Speed ω_m are checked to be certain that every actual value during simulation (*Working point*) is inside the speed-torque curve. Concept is displayed in Figure 3.22.

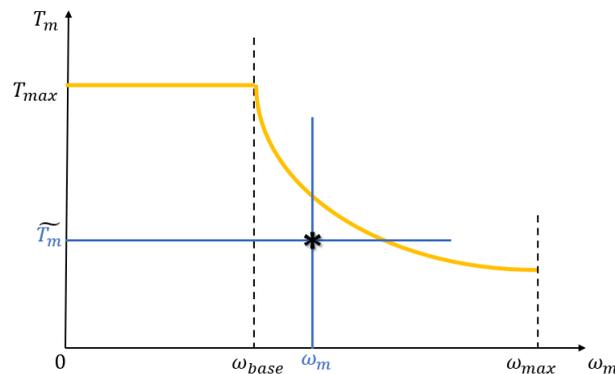


Figure 3.22: Example of working point check in the I quadrant.

Referring to Figure 3.22, the area between zero and base speed is called *constant torque* area, while the one between base and maximum speed is called *constant Power*.

Finding the maximum Efficiency for each Motor Torque $T_{m,checked}$ [Nm] and Angular Speed $\omega_{m,checked}$ [$\frac{rad}{s}$] allows to obtain the machine *Efficiency Map*, creating an appropriate combination in the Motor Torque-Speed plane. This combination is reached in order to yield the highest Efficiency while satisfying the machine voltage and current limit.[14]

In Figure 3.24, the Electric Motor Efficiency Maps are displayed for the I and IV quadrant.

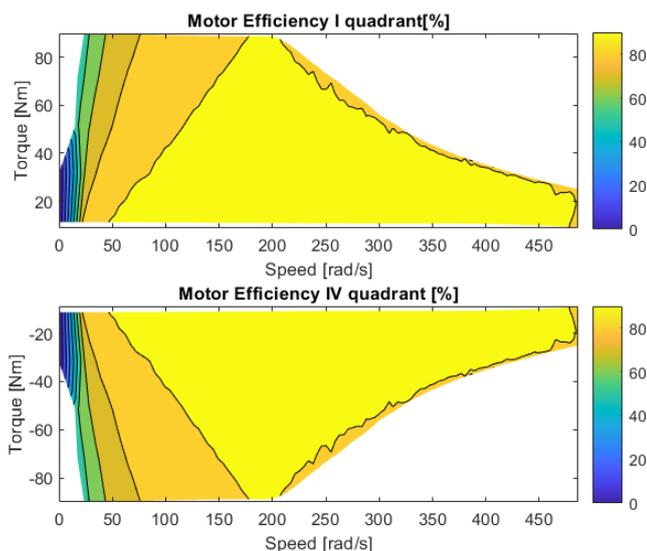


Figure 3.24: Motor Efficiency Map I and IV quadrant.

Figures reported in 3.24 are plots from the dedicated *MATLAB* thesis script. To create these kind of plots, it is used the instrument *contourf*(X, Y, Z). The latter creates a filled contour plot containing the isolines of a matrix Z , where Z contains height values on the x-y plane. In this way, *MATLAB* automatically selects the contour lines to display. The column and row indices of Z are the x and y coordinates in the plane, respectively.

Each color corresponds to a different contour interval. The intermediate contour intervals are defined by pairs of adjacent elements in the strictly increasing vector of contour levels.

X, Y, Z , *contourf* argument, are matrices, each of them is composed by 5 rows and 111 columns. Each rows represents one of the 5 different cases, displayed in the *Excel* file of *Dana TM4* Electric Motor manufacturer. The column number, 111, correspond to the values presented in *Excel* file.

- Matrix X includes Electric Motor angular speed values expressed in $\frac{rad}{s}$ of the five RMS current value cases.
- Matrix Y includes Electric Motor torque values expressed in Nm of the five RMS current value cases.

- Matrix Z includes Electric Motor efficiencies values expressed in percentage [%] of the five RMS current value cases.

Efficiency Maps have to be reported on *Simulink* model, this is done using 2-D LUTs as applied for the Driveline efficiencies. Also for the **Electric Motor** subsystem, 2-D LUTs require a 3-D plots, displayed in Figure 3.25.

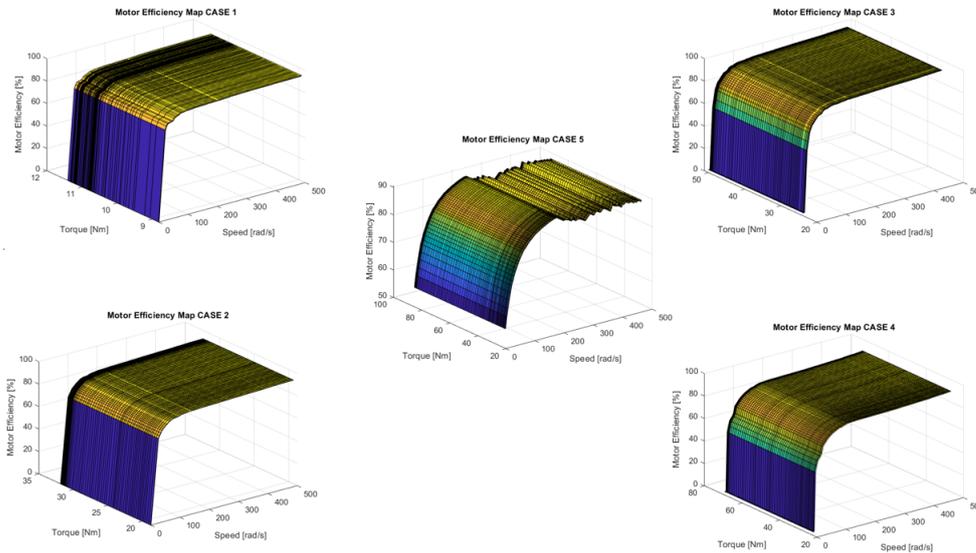


Figure 3.25: **Electric Motor** *Efficiency maps* 3-D plots. CASE 1, 2, 3, 4, 5.

Simulink principle modeling is the same: through a RMS current discrimination, a multiport switch assigns the RMS current case and the related LUT is used.

In Figure 3.26 is displayed the *Simulink* model for the Efficiency computation.

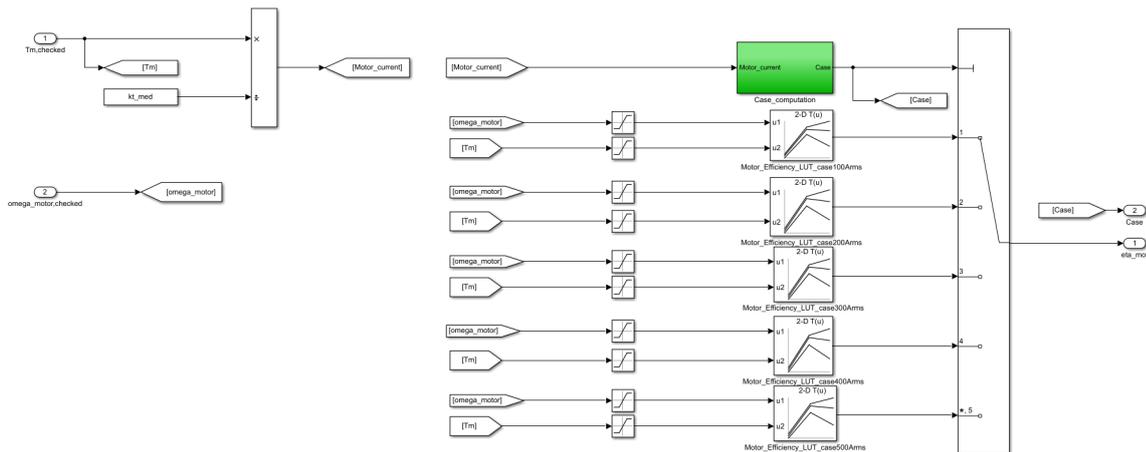


Figure 3.26: *Simulink* model for the **Electric Motor** efficiency computation.

Once computed the **Electric Motor** Efficiency, η_{motor} , the input motor power can be calculated with the power formulation:

$$P_{m,in} = \frac{T_{m,checked} \cdot \omega_{m,checked}}{\eta_{motor}} \quad (3.12)$$

η_{motor} is an actual value ranging from 52.32% (0.5232) to 96.37% (0.9637)

3.6 Inverter

This section is dedicated to the **Inverter** model implemented in *Simulink* environment. The **Inverter** general subsystem is reported in Figure 3.27 below.



Figure 3.27: Inverter subsystem scheme.

Inverter subsystem is modeled in an energetic way, where $P_{m,in}$ is the output of the **Electric Motor** subsystem, that in the Inverter block becomes an input. The output is $P_{dc,in}$, that is the power supplied by the battery pack, so the power in input at the Inverter: an electric convention is kept considered.

Modeling is based on the *Efficiency Maps* of this power converter. Computation is the same as seen about the Driveline and the **Electric Motor**. In the below Figure, 3.28, is displayed the filled 2-D filled contour plots of the *Efficiency Maps* about the I and IV quadrant.

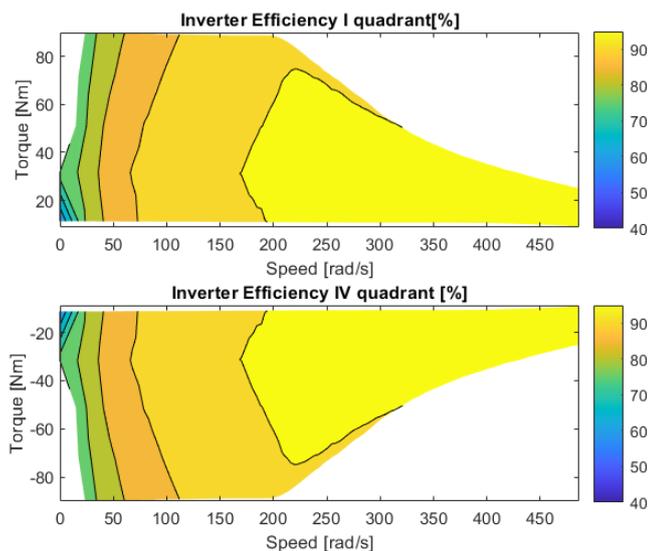


Figure 3.28: Inverter Efficiency Maps for the I and IV quadrant.

Modeling is based on 2-D LUTs, through 3-D Efficiency plots, displayed in Figure 3.29.

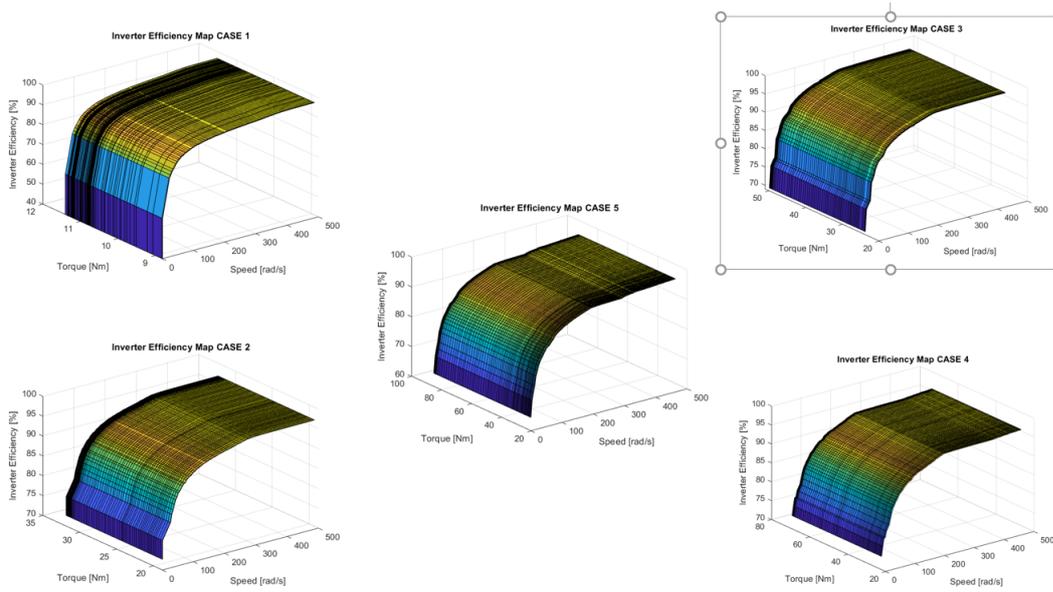


Figure 3.29: **Inverter Efficiency maps** 3-D plots. CASE 1, 2, 3, 4, 5

In Figure 3.30 there is the *Simulink* model dedicated to the $\eta_{inverter}$ computation.

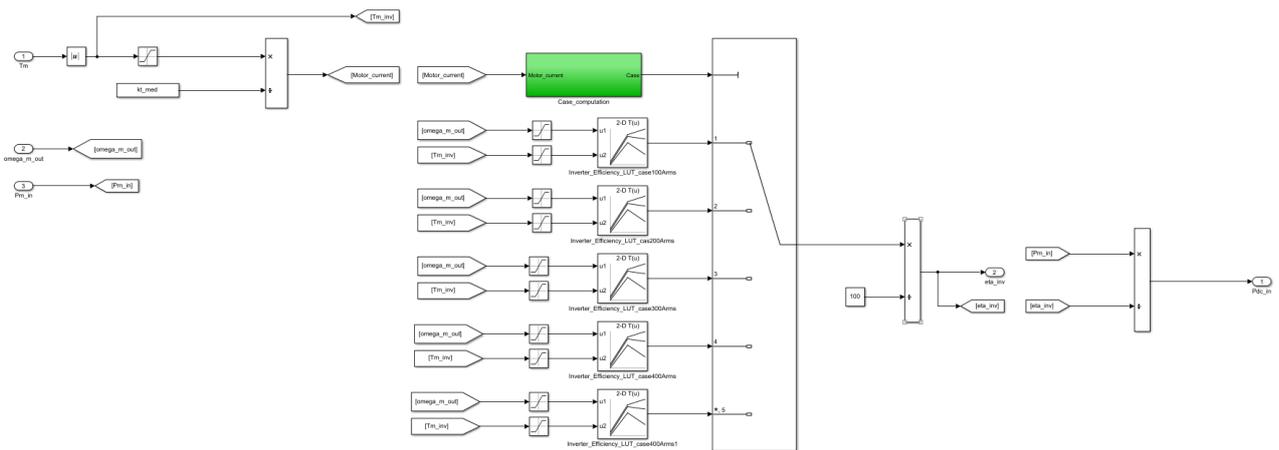


Figure 3.30: *Simulink* **Inverter Efficiency** computation.

The Inverter efficiency, $\eta_{inverter}$, is obtained as a percentage from the *Efficiency Maps*, a divide block is included in the model to reach the Efficiency as value between 0.3993 (39.93%) and 0.9806 (98.06%). These values are respectively the minimum Inverter efficiency and the maximum Inverter efficiency; both obtained by *Excel* file elaborated by *Dana TM4* manufactured.

At the extreme right of Figure 3.30, there is the divide block used to compute the $P_{dc,in}$, through power balance:

$$P_{dc,in} = \frac{P_{m,in}}{\eta_{inverter}} \tag{3.13}$$

Inverter subsystem supply the **Battery** subsystem.

3.7 Battery

3.7.1 Model structure

Battery subsystem is the block that yields the output requested by total *Simulink* powertrain models: the SOC (as displayed in Figure 3.2 and 3.3).

An EVs vehicle battery pack is affected by an high level non-linearity of electro-chemical processes, this means battery modeling is extremely challenging. A battery model depends on some parameters and effects like:

- SOC
- temperature
- ageing
- current direction
- charging/discharging rate

To reach an appropriate **Battery** modelization, used for system-level simulations, as in this thesis case, Thevenin-based circuitual models are employed.

The model adopted in this thesis is a simplification of a well-structured Thevenin circuitual model, shown in Figure 3.31.

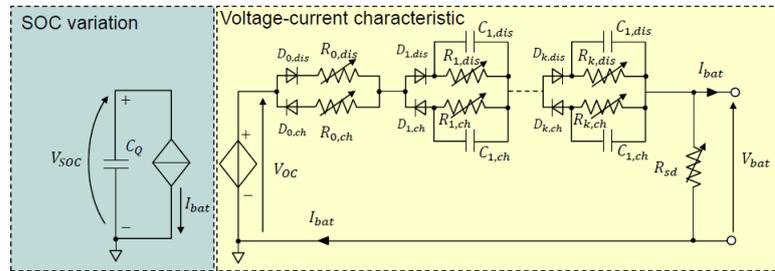


Figure 3.31: Detailed battery Thevenin circuitual model.[10]

Referring to Figure 3.31

- V_{OC} or **OCV** is the **Open-Circuit Voltage**, that represents voltage source depending on the SOC and temperature;
- $R_{0,ch}/R_{0,dis}$ are the total internal series resistance for charging/discharging, that are dependent on the SOC and temperature;
- R_{sd} is the resistance that represents the battery self-discharging;
- $R_{k,ch}/R_{k,dis}$ and $C_{k,ch}/C_{k,dis}$ represent RC groups modeling the voltage transients due to the chemical reactions, all resistances are depending on the SOC and temperature. Subscript k is value ranging from 0 to n , where n is the total number of RC groups;

- $D_{k,ch}/D_{k,dis}$ are ideal unidirectional switches (diodes) conducting the charging/discharging currents.

Typically for a step variation of the battery current, indicated as I_{batt} , the variations of the $V_{OC}|_{T=ct.}$ and battery output voltage V_{batt} . In Figure 3.32, there are the voltage trends yield with a step variation of battery current.

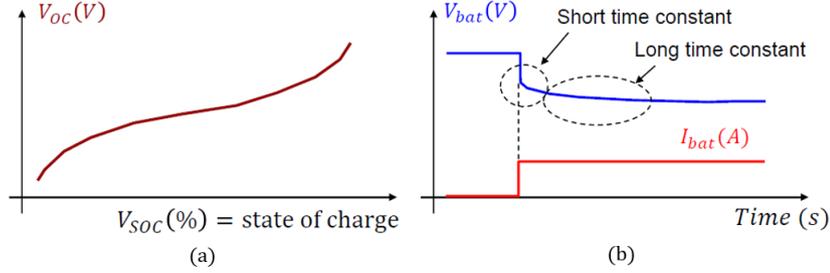


Figure 3.32: Voltage trends: (a) Open-circuit voltage; (b) Battery voltage and battery current step variation.[10]

Referring to Figure 3.32(b), V_{batt} waveform exhibits two different time constants, respectively about Short time and Long time constant, as shown in the circled parts. In order to obtain a simplified battery model, detailed one becomes Thevenin circuitual of second order model with two RC networks having two different time constants:

- $\tau_1 = R_1 \cdot C_1$
- $\tau_2 = R_2 \cdot C_2$

In theory, the RC values are SOC dependent. In Figure 3.33, it is displayed this first simplified model.

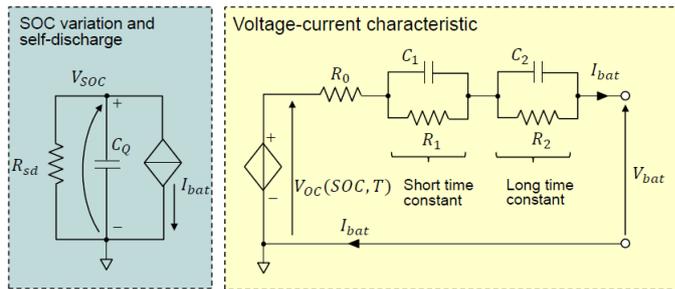


Figure 3.33: Thevenin circuitual second order model.[10]

In Figure 3.33, the blue square is dedicate to *SOC variation and self-discharge circuit*. The latter includes a capacitor, a resistor and a current source equal to the battery current.

The capacity C_Q is computed as:

$$C_Q = 3600 \cdot Q_{rated} \cdot k_{cycle} \cdot k_{Temp} \tag{3.14}$$

where

- Q_{rated} is the rated capacitance [Ah];
- $k_{cycle} \leq 1$ represents a corrective gain depending on the number of battery cycles (equal to 1 if unknown);
- $k_{Temp} \leq 1$ is a corrective gain depending on the temperature (equal to 1 if unknown).

The *SOC variation and self-discharge circuit* can be modeled with blocks displayed in Figure 3.34.

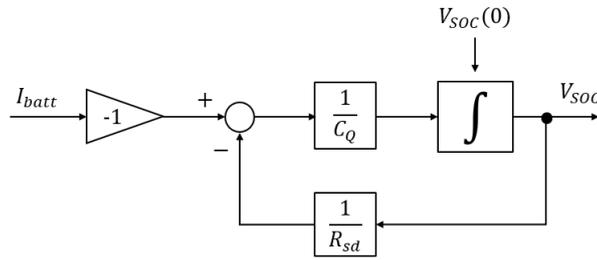


Figure 3.34: SOC variation and self-discharge circuit schematic blocks.

- R_{sd} [Ω] is self-discharging resistance (if unknown, set to tens of $k\Omega$ or eliminate)
- V_{SOC} is the SOC in [pu]

The blocks scheme in Figure 3.34 is the structure of the following equation

$$v_{SOC}(t) = -\frac{I_{batt}}{C_Q} \cdot t + V_{SOC}(0) \cdot e^{-\frac{t}{\tau_{sd}}} \quad (3.15)$$

where

τ_{sd} is the time constant expressed as $R_{sd} \cdot C_Q$.

In Figure 3.33, the other circuit displayed is the one dedicated to the *Voltage-current characteristic*, including a voltage source ($V_{OC}(SOC, T)$), a resistance (R_0) and two RC (R_1C_1 and R_2C_2) networks.

The Open circuit voltage V_{OC} depends on the SOC and on the temperature. Typically, the V_{OC} is provided as LUTs or as multiple curves (corresponding to different temperatures) using polynomial interpolation with respect to SOC.

LUTs employment is the technique adopted for this thesis about Open-circuit voltage computation.

In Figure 3.35, there is the circuit that has to be analyzed.

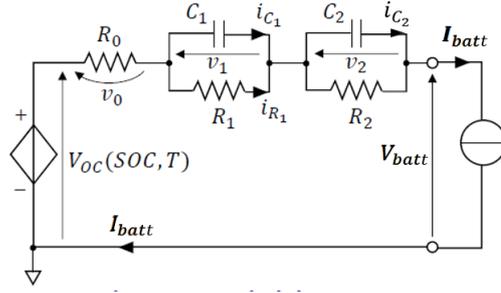


Figure 3.35: Voltage-current characteristic.[10]

Considering the battery current as an input variable, applying the Kirchhoff's voltage law, Battery Voltage is expressed as

$$V_{batt} = V_{OC}(t) - R_0 \cdot I_{batt}(t) - v_1(t) - v_2(t) \tag{3.16}$$

where it is possible discern three voltage formulation

- $v_0 = R_0 \cdot I_{batt}$
- $v_1 = I_{batt} \cdot R_1 \cdot (1 - e^{-\frac{t}{\tau_1}})$ where $\tau_1 = R_1 \cdot C_1$
- $v_2 = I_{batt} \cdot R_2 \cdot (1 - e^{-\frac{t}{\tau_2}})$ where $\tau_2 = R_2 \cdot C_2$

In this way, the Battery Voltage V_{batt} is formulated as a difference of single voltages

$$V_{batt} = V_{OC}(SOC, T) - v_0 - v_1 - v_2 \tag{3.17}$$

The blocks scheme structure is displayed in Figure 3.36.

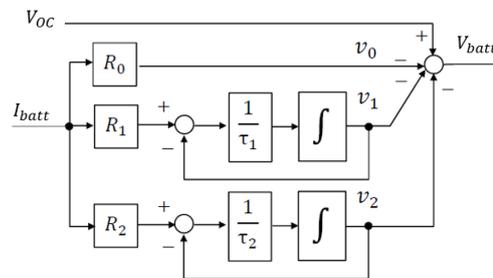


Figure 3.36: Schematic block of the voltage-current characteristic circuit.[10]

Final simplified Thevenin battery model is displayed in the Figure 3.37.

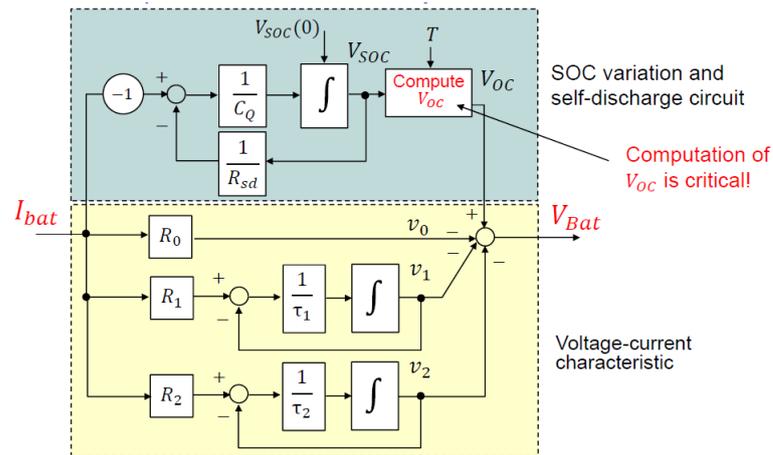


Figure 3.37: Final simplified Thevenin battery electrical model.[10]

The model in Figure 3.37 can be further simplified by eliminating the two RC networks R_1C_1 and R_2C_2 , so the voltage dynamic due to electrochemical process is not taken anymore into account. Indeed the Battery model used in this thesis, it is the most simplified one.

Thevenin battery electrical model without modeling the electrochemical transients is displayed in Figure 3.38. This is the Battery model adopted for this thesis.

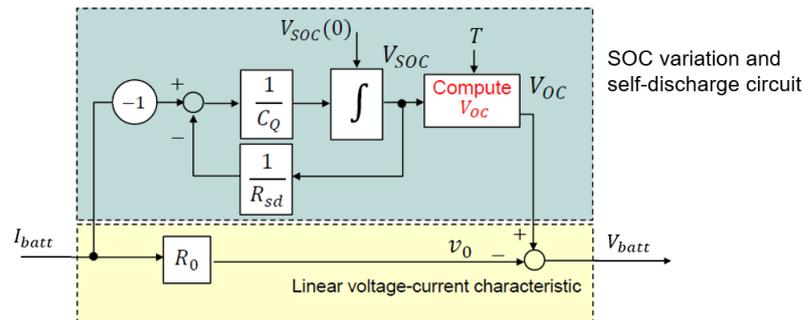


Figure 3.38: Simplified battery model.[10]

The key issue is to have a good estimation of R_0 and of the V_{OC} in function of the SOC.

3.7.2 Battery subsystem inputs

The **Battery** subsystem is modeled as shown in Figure 3.39.



Figure 3.39: **Battery** subsystem.

Referring to Figure 3.39, the subsystem shows two inputs:

- I_{batt} , the battery current.

In Figure 3.6, the power $P_{dc,in}$ enters in the block denominated **Battery Current computation**, that following the powerflow, it is before the battery subsystem. **Battery Current computation** has also as input the battery voltage V_{batt} , received as feedback by the *Battery* subsystem.

Battery current computation is a subsystem where it is applied the following conditions, based on the motor power $P_{m,in}$ sign.

If $P_{m,in}$ is positive, meaning discharging condition for the battery, than the battery current is calculated as

$$I_{batt} = \frac{P_{dc,in}}{V_{batt}} \quad (3.18)$$

otherwise, if $P_{m,in}$ is negative, meaning charging condition for the battery, than the battery current is computed as

$$I_{batt} = \frac{P_{m,in} \cdot \eta_{motor} \cdot \eta_{inverter}}{V_{batt}} \quad (3.19)$$

Simulink model of battery current computation subsystem is displayed in Figure 3.40.

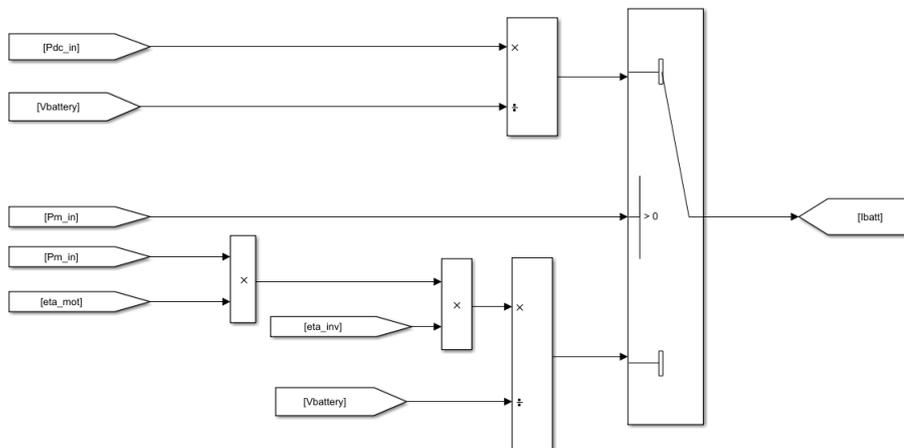


Figure 3.40: **Battery current computation** *Simulink* model.

Once obtained the battery current, I_{batt} , the variable feeds the subsystem contains the battery model.

- SOC_{in} , the initial SOC that *Microlino 2.0 Prototypes* has at the simulation beginning.

The initial SOC is received as input by the battery subsystem as a constant variable, initialized by the *MATLAB* dedicated script, taken by CAN acquisition at each simulation. SOC_{in} is important to compute the initial condition about the

integration to obtain the Open-circuit voltage.

The initial condition, indicated as $V_{SOC,0}$, is computed considering that this Battery Initial Voltage is equal to 0 when there is Full Discharged condition, while is equal to 1 when Full Charged condition.

As explained in Chapter dedicated to *Microlino* battery, battery pack is one of the main differences in the two *Simulink* models implemented. So, in the following subsections there is the description of the *Simulink* models for the battery part of the two prototypes.

3.7.3 Mule prototype

Mule prototype has a battery pack manufactured by BOSCH and composed by six modules of 50Ah of nominal capacity (as explained in Chapter 2).

Modeling is composed by six Manual Switches⁵, that are set before starting the simulations and they assign the states of each battery modules: if it is activated switch input it is set to 1; otherwise a disabled module has switch input is set to 0.

In Figure 3.41, the *Simulink* model dedicated to the Manual Switches is displayed. The choice to include this part in the modeling of the battery dedicated subsystem lies in the fact that during the *Mule Prototype* testing phase, CAN acquisitions were made with only two battery modules activated.

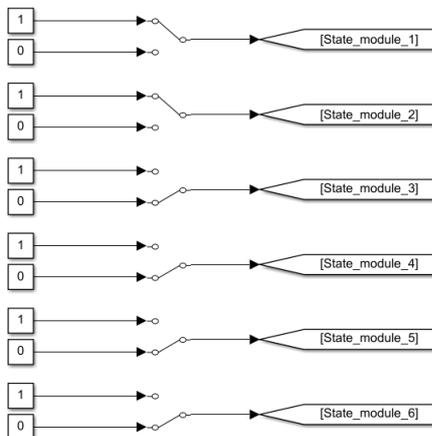


Figure 3.41: Modeling of Manual Switches.

Each *State module* is added in a dedicated block to generate a variable, n_module , that represents the number of activated modules.

⁵The **Manual Switch** block is a toggle switch that selects one of its two inputs to pass through to the output. To toggle between inputs, double-click the block. You control the signal flow by setting the switch before you start the simulation or by changing the switch while the simulation is executing. The Manual Switch block retains its current state when you save the model.[13]

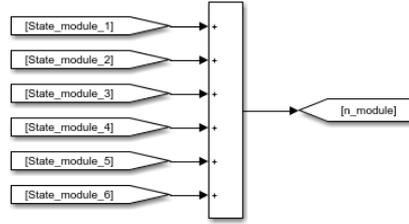


Figure 3.42: *Simulink* modeling dedicated to the computation of the number of activated modules.

The variable n_module is used to divide the total amount of battery current that enter in the battery subsystem.

The variable SOC_{in} is used to computed the voltage $V_{SOC,0}$, that is the Initial condition source of the integrator used in the modeling of the *SOC variation and self-discharge circuit*, reported in Figure 3.37.

The $V_{SOC,0}$ formulation is

$$V_{SOC,0} = \frac{SOC_{in}}{100} \cdot \frac{1}{\frac{n_parallel}{n_module}} \tag{3.20}$$

where

- $n_parallel$ is the total number of parallel modules of BOSCH battery pack, equal to 6.
- n_module is the variable coming from sum shown in Figure 3.42.

In Figure 3.43, there is model dedicated to equation 3.20.

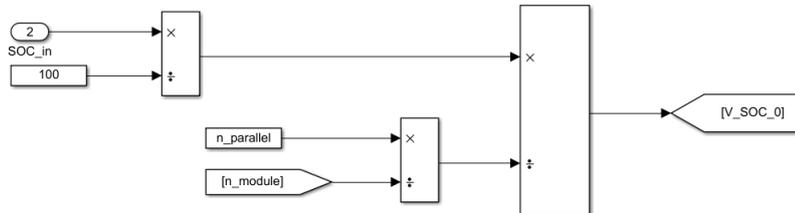


Figure 3.43: Initial Open-Circuit voltage computation from initial SOC.

Once knowing the number of active battery module and once computed the voltage $V_{OC,0}$, the battery subsystem contains the six module models. In Figure 3.44, it is displayed the subsystem dedicated to module number one.

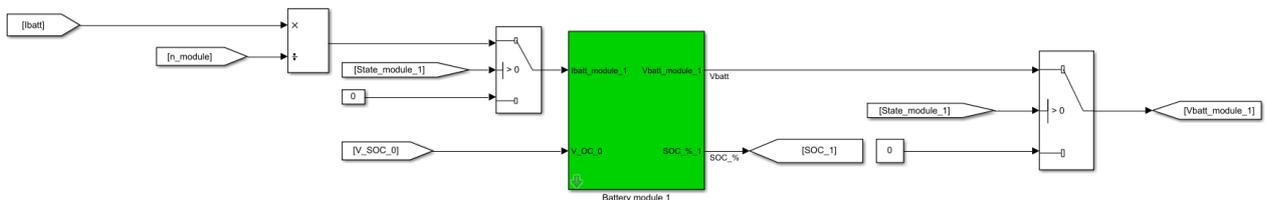


Figure 3.44: **Battery** module 1 modeling.

Referring to Figure 3.44, the green block is the subsystem contains the battery model as explained previously, that is the one in Figure 3.37. In Figure 3.45, there is the model implemented on *Simulink* dedicated to *Mule Prototype*.

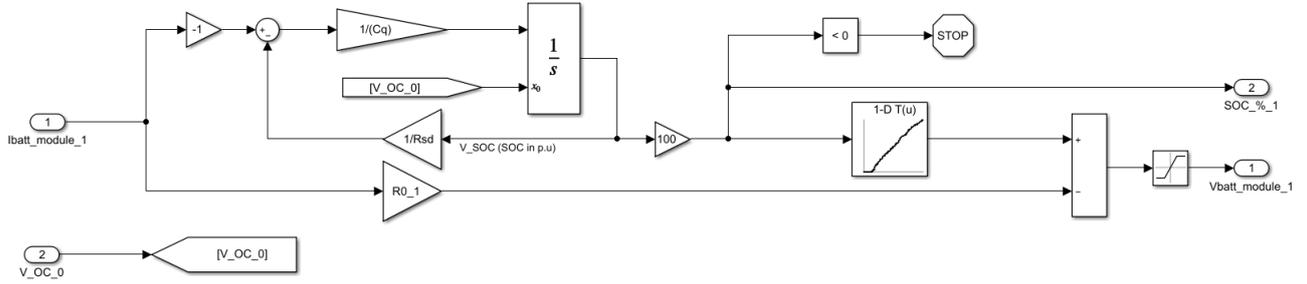


Figure 3.45: **Battery** model for module 1.

The 1-D LUT presents in Figure 3.45 is used to compute the Open-Circuit Voltage, V_{OC} .

The literature[11] reports an exponential model which shows the relationship between Open-Circuit Voltage and the State of Charge, expressed as

$$V_{OC} = a_1 e^{b_1 SOC} + a_2 e^{b_2 SOC} + c SOC^2 \quad (3.21)$$

where, a_1 , b_1 , a_2 , b_2 , and c are the coefficients with the temperature: for this thesis they are calibrated based on CAN acquisition. In Figure 3.46, the LUT used to compute the Open-circuit Voltage is displayed.

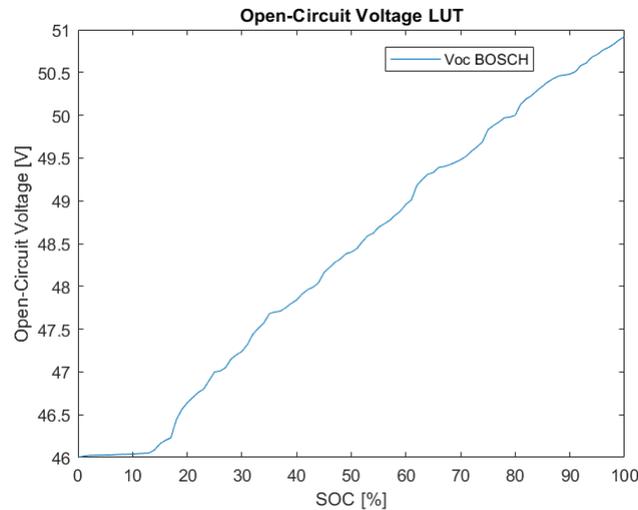


Figure 3.46: **Battery** model: V_{OC} LUT.

The green subsystem outputs are the Battery voltage [V] and the SOC, expressed in %.

Once obtained the voltage and the SOC of each active module to reach the battery voltage and battery SOC: the single voltage and SOC values obtained are sum up. The latter values are divided by the number of active module ($n_{parallel}$), in order to

obtain the battery voltage V_{batt} and the battery SOC . The model part dedicated to these computations is displayed in Figure 3.47.

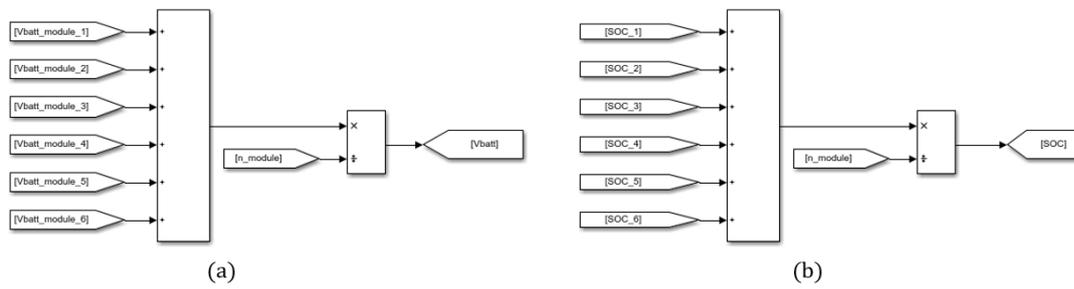


Figure 3.47: Battery voltage and SOC computation model.

3.7.4 Prototype 4

Prototype 4 is the second model implemented and tested on *Simulink*. As already explained, the main model differences lies in the Input part and in the Battery pack.

Prototype 4 battery modeling has the same structure as *Mule prototype* battery model: the main difference lies in the number of modules (n_module variable): that is 2 in the *Prototype 4*.

The total **Battery** subsystem model is displayed in Figure 3.48.

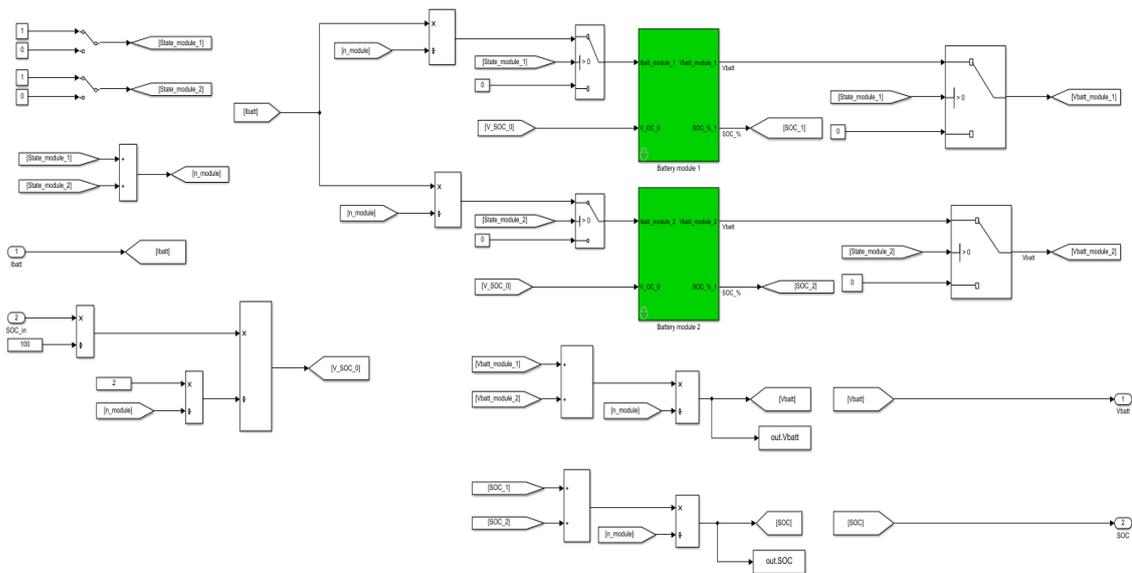


Figure 3.48: *Simulink* battery model *Prototype 4*.

Figure 3.49 displays the Open-Circuit Voltage LUT to compute the V_{OC} .

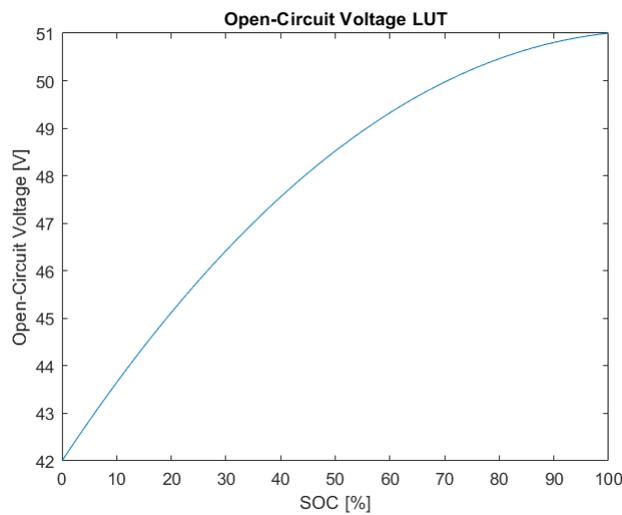


Figure 3.49: **Battery** model: V_{OC} LUT.

Chapter 4

Simulation results

This fourth chapter is dedicated to the results obtained from the *Simulink* models. The main results, like the SOC, are compared with the values and waveforms caught on CAN, during *Prototype* testing phase.

Once a consistency between CAN acquisition and model values is reached, the validation of the two model is confirmed.

First of all, the **CAN (Controller Area Network)** bus is the vehicle '*nervous system*', enabling communication. In turn, '*nodes*' or '*ECUs*' are like parts of the body, interconnected via the CAN bus: indeed informations sensed by one part can be shared with another. In an automotive CAN bus system, ECUs manages the engine control unit, airbags, audio system etc. Typically, a modern vehicle may have up to 70 ECUs, each of them may have information that needs to be shared with other parts of the network.

The CAN bus system enables each ECU to communicate with all other ECUs, without complex dedicated wiring. Specifically, an ECU can prepare and broadcast information, sensor data, via the CAN bus (consisting of two wires: *CAN low* and *CAN high*). The broadcasted data is accepted by all other ECUs on the CAN network. Then, each ECU can check the data and decide whether to receive or ignore it.

Communication over the CAN bus is done via CAN frames.

Doctor Marco Ghibaudi, *Bylogix*'s control system engineer and thesis development supervisor, handled both thesis prototypes CAN acquisitions on ride track. Data were obtained using *Vector CANalyzer* tools, to ensure compatible data were converter in a *MATLAB* file. The latter is loaded on the initialization *MATLAB* script.

This chapter is structure in two main sections, dedicated to:

1. *Mule Prototype*: CAN acquisition date back to 22th March 2021.
2. *Prototype 4*: CAN acquisition date back to 5th July 2021.

4.1 Mule Prototype

Mule Prototype was used in *Bylogix* workshop to test the main electric powertrain elements.

CAN acquisitions of the 22th March 2021 were made around the building that houses *Bylogix* headquarters: offices and workshop.

In Figure 4.1, there is a *street view* of the track distance covered (marked in yellow) during *Mule prototype* testing phase.



Figure 4.1: Track distance covered by *Mule Prototype*.

As explained in Chapter 3, *Mule prototype* is the first testing prototype, so it does not consider all the accuracies: but the *.mat* loaded file includes all the parameters requested for the test of the 22th March 2021.

The *Simulink* model is less complex and less articulate than the one implemented for *Prototype 4*. Inputs, as already explained are transferred from the *MATLAB* script through a *From Workspace* block:

- Percentage on the Maximum allowed Torque ($torque_CAN$ [%]) and Speed ($Speed_CAN$ [rpm]) are CAN data caught directly on the **Electric Motor**;
- Maximum Discharge and Charge Currents ($I_{max,discharge[2s]}$ and $I_{max,charge[2s]}$ [A]) are given by CAN parameters coming from the BMS frame.

The ride track duration is 127.3492 s. The latter value assumes the name *simulation_time*, in this way it is possible initialize the **Stop Time** in *Simulink* environment.

4.1.1 Inputs

Mule prototype simulation inputs are the torque and speed.

In Figure 4.2, there are the waveforms of input Reference Torque (T_m^*) and input angular speed (ω_m). The waveforms are plotted, inserting the *Scope* block after *Torque_Gain* and *Speed_Gain*, in order to express their magnitudes respectively in Nm and $\frac{rad}{s}$.

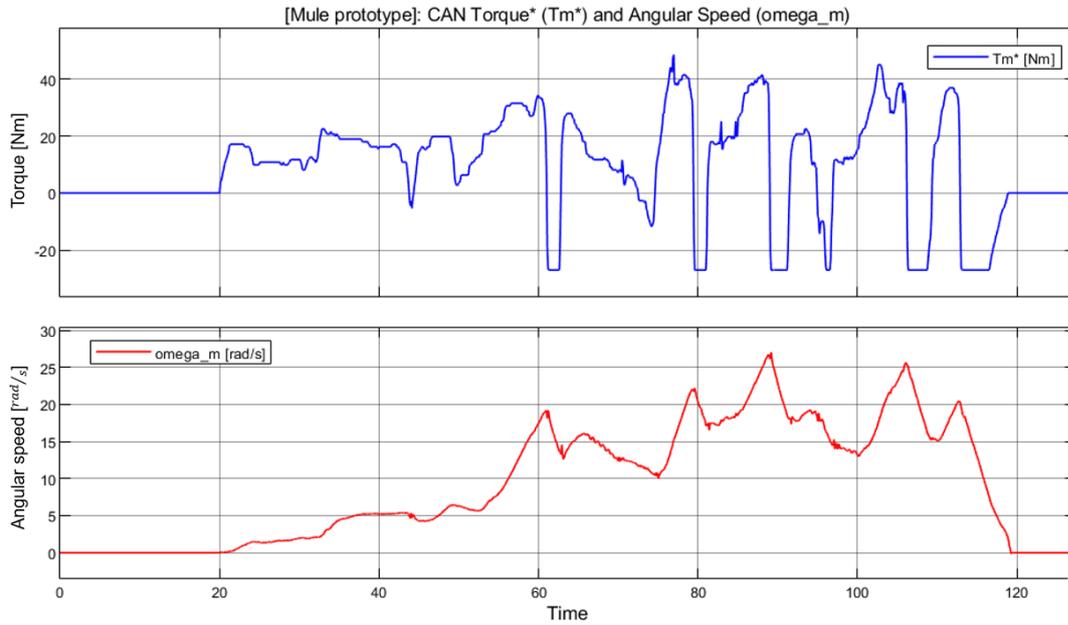


Figure 4.2: [*Mule Prototype*] Input: Reference torque T_m^* and angular speed ω_m waveforms.

The other two inputs are displayed in Figure 4.3, ($I_{discharge,[2s]}$, $I_{charge,[2s]}$ expressed in A).

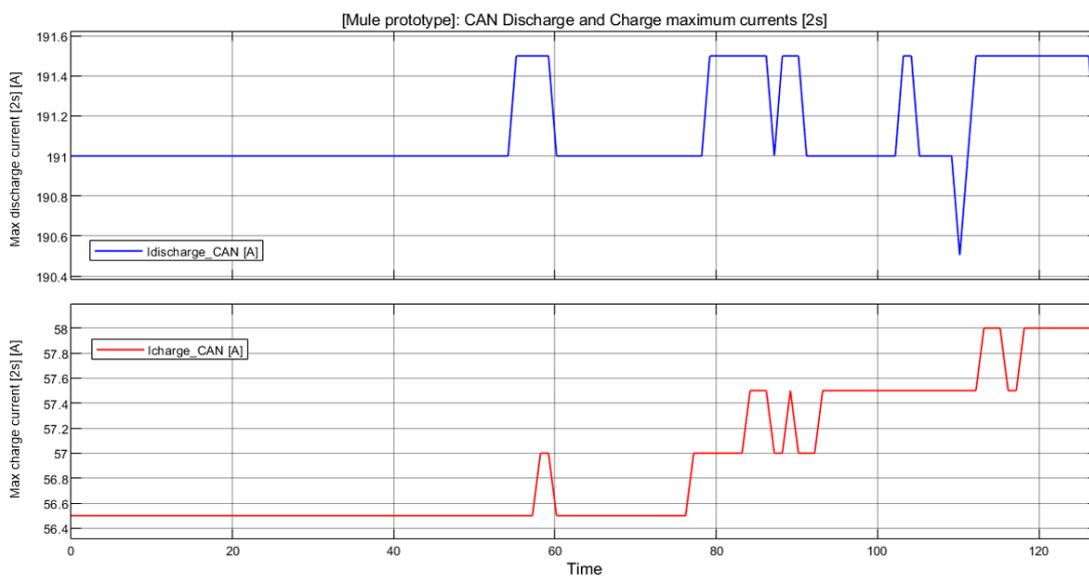


Figure 4.3: [*Mule Prototype*] Input: Maximum discharge ($I_{discharge,[2s]}$) and charge currents ($I_{charge,[2s]}$) waveforms.

4.1.2 Energy Management - BMS

First subsystem in *Mule prototype* model is the one dedicated to **Energy Management - BMS**.

Figure 4.4 displays the result of saturation action on reference motor torque (T_m^{**}) made by the BMS feed-forward action.

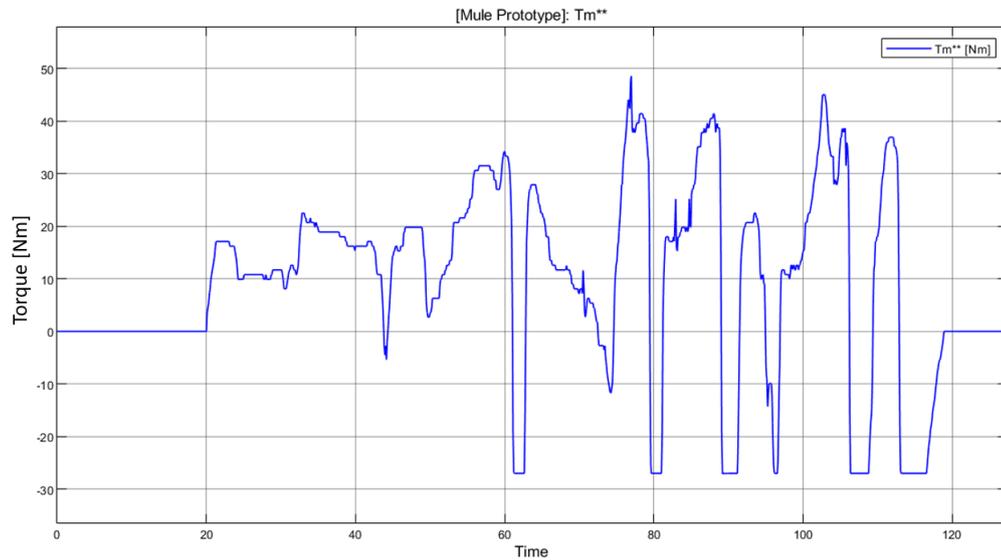


Figure 4.4: [*Mule Prototype*] Energy Management - BMS: Output Reference Torque T_m^{**} waveform.

The **Energy Management - BMS** input Torque T_m^* contains acceptable values, indeed the saturation block does not participate. In Figure 4.5, there are two scope displays respectively T_m^* and T_m^{**} : it is possible observe that waveforms are the same, so no saturation is needed.

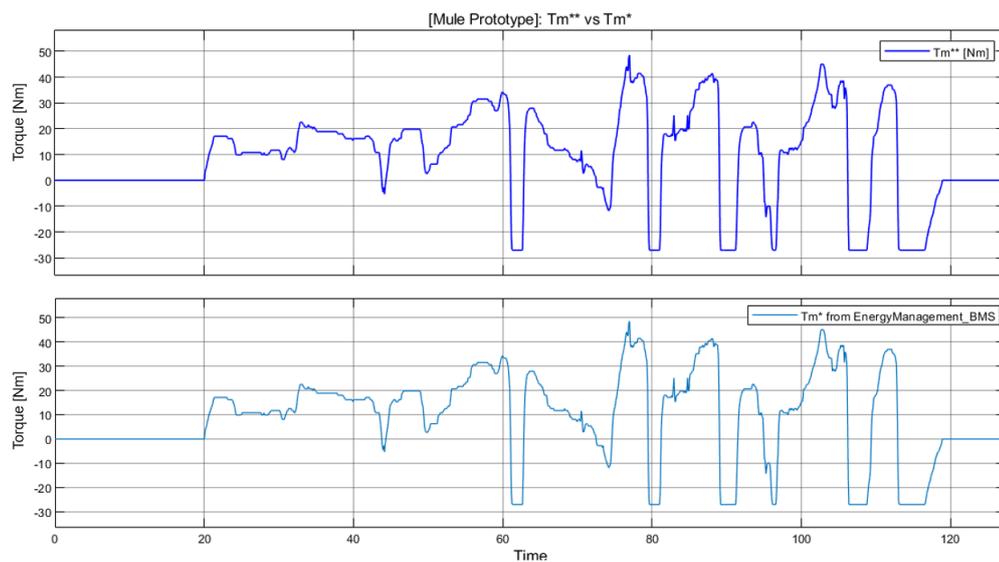


Figure 4.5: [*Mule Prototype*] Energy Management - BMS: Comparison between T_m^{**} and T_m^* waveforms

Product of Reference Torque T_m^{**} and angular speed ω_m is the output motor power, $P_{m,out}$ (keep considering an electric convention). In Figure 4.6, $P_{m,out}$ waveform is displayed.

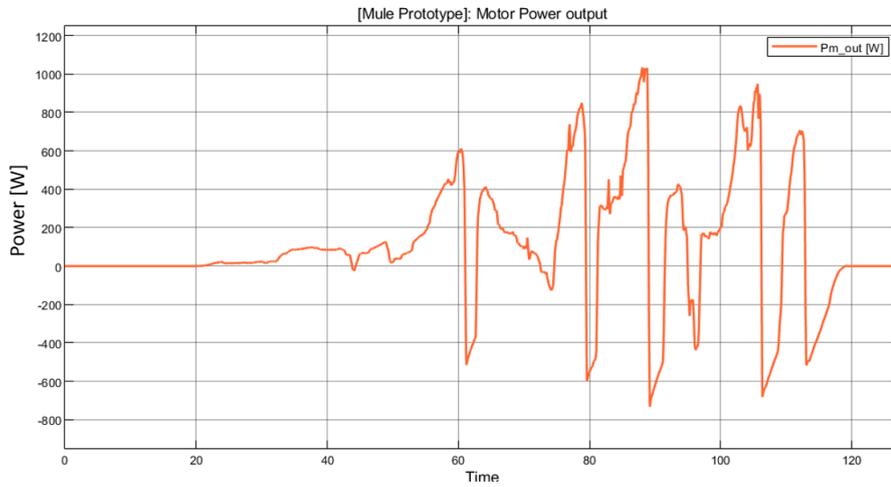


Figure 4.6: [Mule Prototype] Energy Management - BMS: Motor output power $P_{m,out}$ waveform (electrical convection).

4.1.3 Electric Motor

Reference torque T_m^{**} , in Figure 4.4, and angular speed ω_m , in the below Figure of Figure 4.2, feed the **Electric Motor** block. In the Figures below, there are the simulation result given by the latter subsystem.

Figure 4.7 displays the output of the Electric Motor subsystem, that is indicated as T_m . T_m is the consequence of the **LPF** computation and working point check.

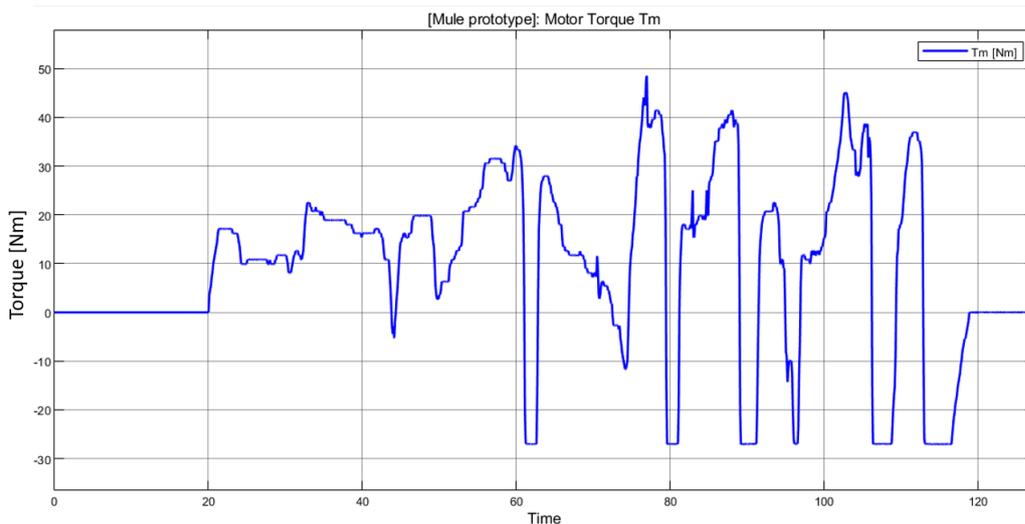


Figure 4.7: [Mule Prototype] Electric Motor: Motor Torque T_m waveform.

Figure 4.8, there is the Angular Speed of the Electric Motor, that remains the same as in the below Figure of Figure 4.2. Angular Speed undergoes just a working point check in the Electric Motor block.

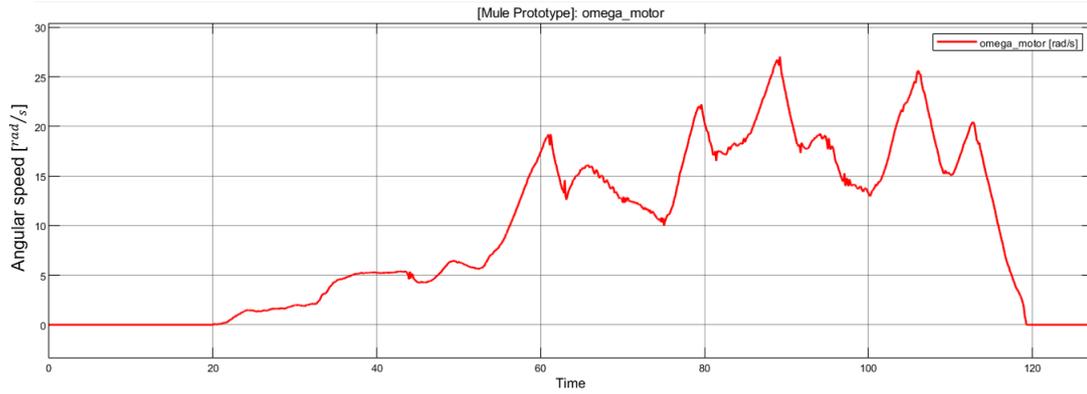


Figure 4.8: [*Mule Prototype*] Electric Motor: Motor Angular Speed ω_m waveform.

T_m and ω_m are used to compute the Electric Motor Efficiency, through *Efficiency Map*, as explained in Chapter 2. Figure 4.9 displays Motor efficiency η_{motor} values during the simulation 127.3492 s.

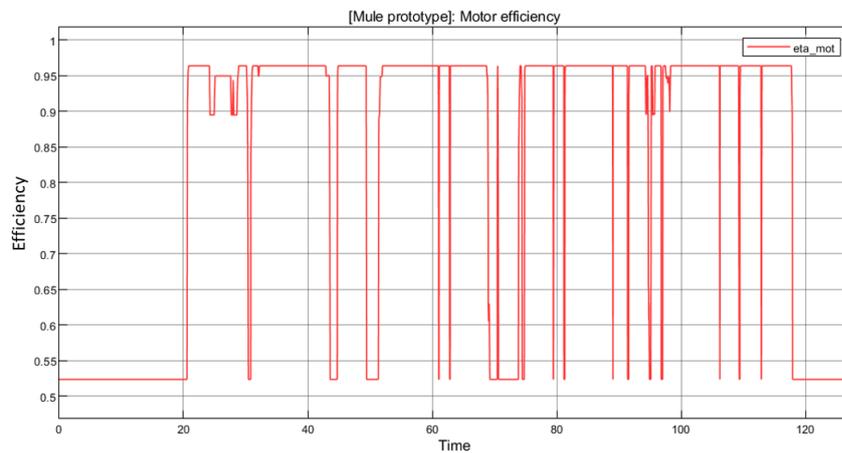


Figure 4.9: [*Mule Prototype*] Electric Motor: Motor efficiency η_{motor} waveform.

Dividing $P_{m,out}$, Figure 4.6, by the efficiency η_{motor} , Figure 4.9, $P_{m,in}$, motor input power (electric convection) is obtained.

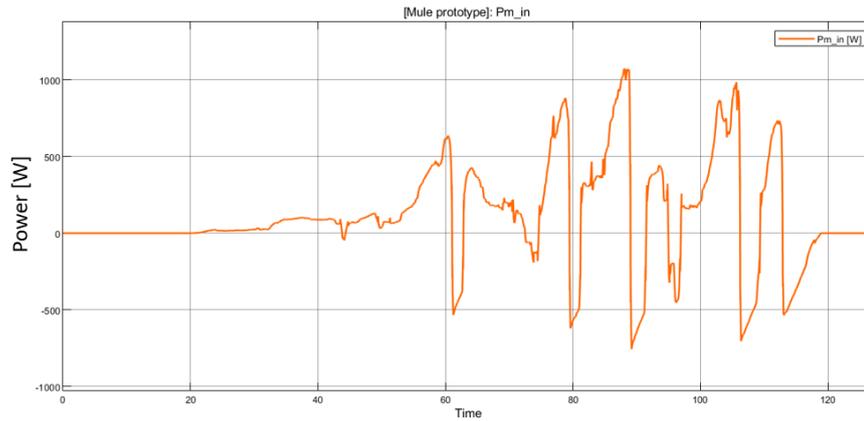


Figure 4.10: [Mule Prototype] Electric Motor: Motor input power $P_{m,in}$ waveform (Electric convection).

4.1.4 Inverter

$P_{m,in}$ is the variable that feeds the **Inverter** subsystem. Through *Efficiency Map*, so $\eta_{inverter}$, Inverter block computes the Battery supply power $P_{dc,in}$, used to determine the Battery Current (I_{batt}).

Figure 4.11 displays the Inverter efficiency values during simulation.

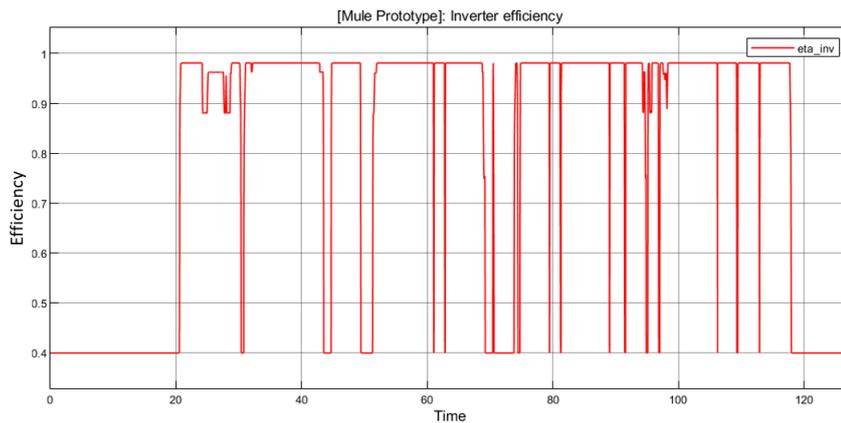


Figure 4.11: [Mule Prototype] Inverter: Efficiency $\eta_{inverter}$ waveform.

Dividing $P_{m,in}$ by $\eta_{inverter}$, the Battery supply power $P_{dc,in}$ is computed. Waveform is displayed in Figure 4.12.

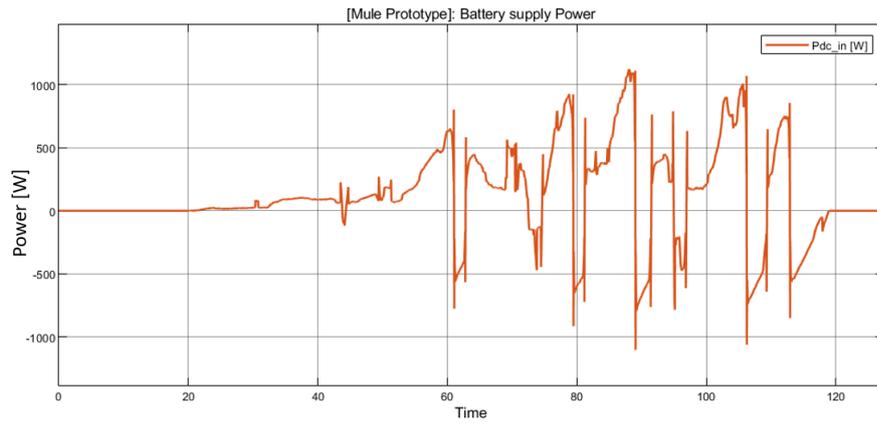


Figure 4.12: [Mule Prototype] Inverter: Battery supply power $P_{dc,in}$ waveform.

In Figure 4.13, it is displayed the comparison between $P_{m,in}$ and $P_{dc,in}$ in the same scope plot.

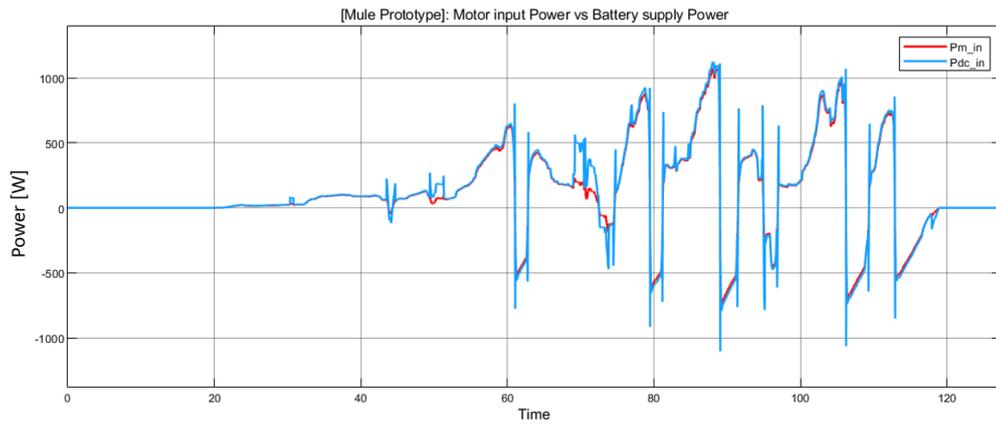


Figure 4.13: [Mule Prototype] Inverter: Comparison between $P_{m,in}$ and $P_{dc,in}$ waveforms.

4.1.5 Battery

The **Battery** subsystem is the one which reach the target of the model: SOC. Its inputs are the initial SOC and the Battery Current I_{batt} . The latter is computed considering the $P_{m,in}$ sign. This allows to understand in which quadrant (I or IV) the model is working. The quadrants are important for the Battery Current computation, as explained in Chapter 2, section 3.8.2

In Figure 4.14, Battery Current waveform is displayed.

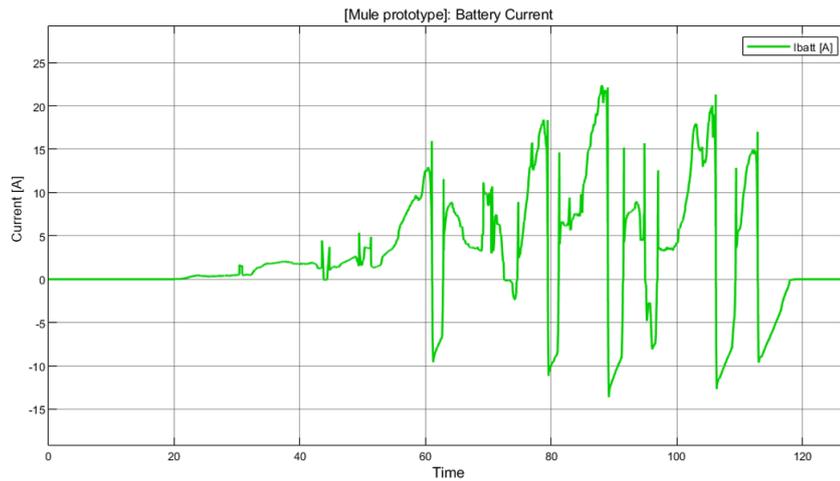


Figure 4.14: [Mule Prototype] Battery: Battery Current I_{batt} waveform.

Battery current is used in the Battery model in order to obtain

- the Battery Voltage V_{batt} , used as feedback to determinate Battery Current. Figure 4.15 displays the Battery Voltage obtained by the model during the ride track simulation.

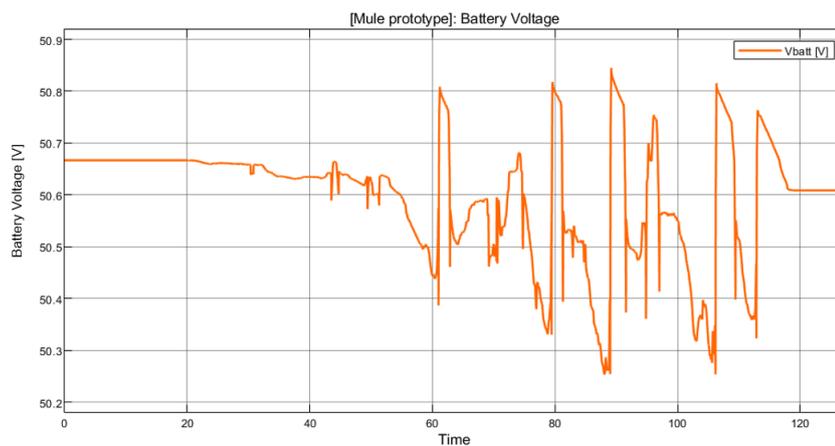


Figure 4.15: [Mule Prototype] Battery: Battery Voltage V_{batt} waveform.

- State of Charge SOC , that is the main task of this model. Figure 4.16 displays the SOC trend obtained during the ride track.

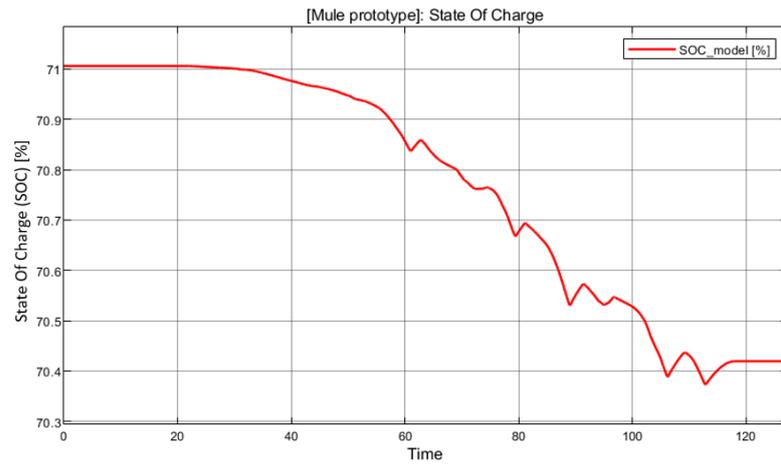


Figure 4.16: [*Mule Prototype*] Battery: *Mule Prototype* SOC.

Mule Prototype - State of Charge

Referring to Figure 4.16, SOC during simulation differs between 71.006%, that is the initial one taken from the *.mat* CAN acquisition file, and 70.4195%, in 127.3492s.

The obtained waveform of Figure 4.16 is compared with the CAN acquisition SOC waveform. The latter is displayed in Figure 4.17.

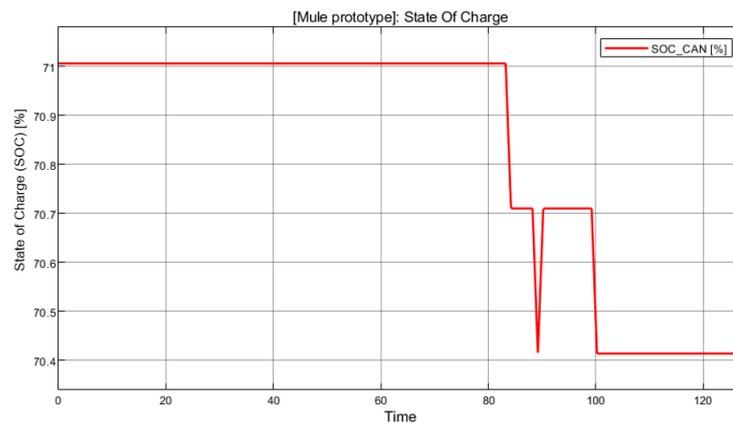


Figure 4.17: [*Mule Prototype*] Battery: State Of Charge waveform obtained by CAN acquisition on ride track.

Waveforms comparison are displayed in Figure 4.18.

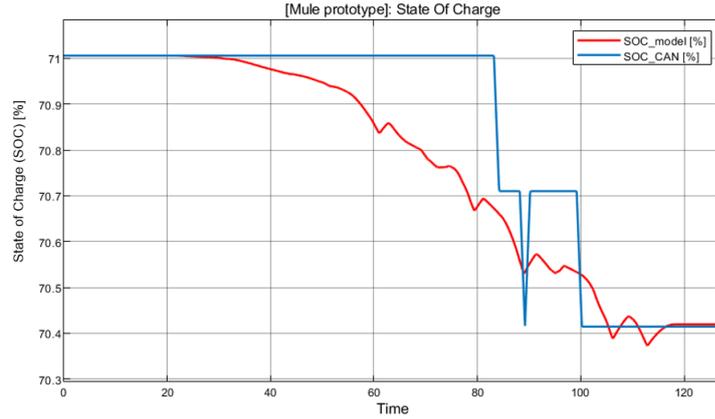


Figure 4.18: [Mule Prototype] Battery: Comparison between model SOC and CAN acquisition SOC.

The SOC acquired by CAN is a discrete waveform, while SOC given by *Mule prototype* model is a time continuous trend. Initial and final SOC values are the same, so to ensure the model validation: a study on SOC mean values is considered.

Simulation stops at $127.3492s$, in this time range it is possible recognise 8 time sub-intervals in the SOC acquired by CAN.

Table 4.1 summarises the $P_{dc,in}$ and SOC_CAN values of these 8 time sub-intervals.

Table 4.1: Summary values of 8 time sub-intervals.

Time interval	seconds [s]	$P_{dc,in}$ average values [W]	SOC_CAN values [%]
<i>Time 1</i>	0s to 84s	156.2407W	71.0059%
<i>Time 2</i>	84.1s to 85s	425.8143W	70.7101%
<i>Time 3</i>	85.1s to 89s	832.9584W	70.7101%
<i>Time 4</i>	89.1s to 90s	-629,1169W	70.4142%
<i>Time 5</i>	90.1s to 91s	-642.6028W	70.4142%
<i>Time 6</i>	91.1s to 100s	77.3692W	70.7101%
<i>Time 7</i>	100.1s to 101s	289.4568W	70.4142%

In each time sub-interval CAN SOC is considered quite constant, assuming the values of Table 4.1 last column.

The study on SOC mean values is computed considering the $P_{dc,in}$ average values in the 8 time sub-intervals and the V_{batt} instantaneous values. Evaluate the the ratio between them, a new Battery current waveform $I_{batt,ist}$ is computed:

$$I_{batt,ist} = \frac{P_{dc,in,mean}}{V_{batt}} \quad (4.1)$$

The computed Battery Current is displayed in the Figure below, 4.19.

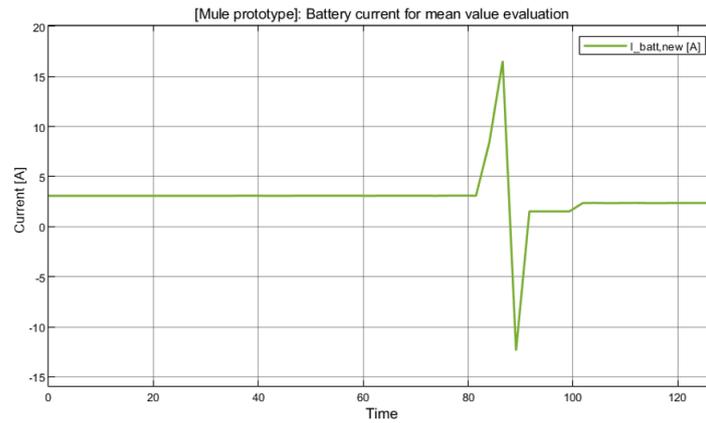


Figure 4.19: [*Mule Prototype*] Battery: Battery Current waveform given by equation 4.1.

This waveform supply a new battery model, through a *From Workspace* block. The model is displayed in Figure 4.20 and it is used to compute the SOC mean values.

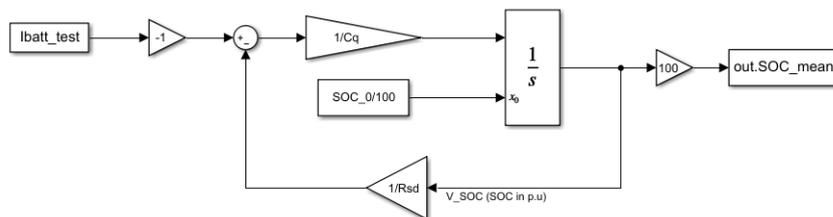


Figure 4.20: *Simulink* model used to compute the SOC.

Referring to the model of Figure 4.20, the output is a *To Workspace* block, contains the SOC mean values: waveform is displayed in Figure 4.21.

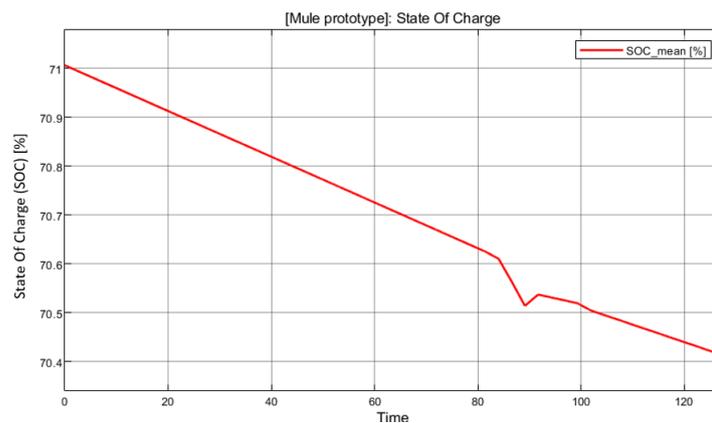


Figure 4.21: [*Mule Prototype*] Battery: SOC mean values waveform.

The computed mean values SOC is compared with the waveforms displayed in Figure 4.18. Comparison results are shown in Figure 4.22.

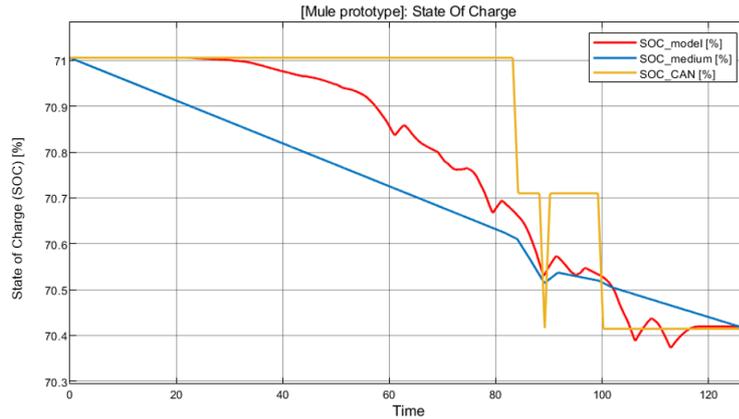


Figure 4.22: [*Mule Prototype*] Battery: Comparison between CAN SOC, Model SOC and mean values SOC.

Observing Figure 4.22, it can be said that the *Mule prototype* validation model is quite reach: due to the fact that the difference between the red waveform (SOC_{model}) and the blue one (SOC_{mean}) is very minimal.

To prove the validation standard deviation σ^1 is taking into account. Its formulation is expressed as

$$\sigma = \sqrt{\frac{\sum_N (x - \bar{x})^2}{N}} \quad (4.2)$$

where

- x represents the value assumed by the SOC in the model;
- \bar{x} is the value mean value assumed by the SOC;
- N is the samples number of the simulation.

The σ is equal to 0.1792. The latter value is used to create a band: if the SOC obtained by the model is inside this band, the model can consider validated.

Figure 4.23 displays the model validation.

¹In statistics, the **standard deviation** is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the values are spread out over a wider range.

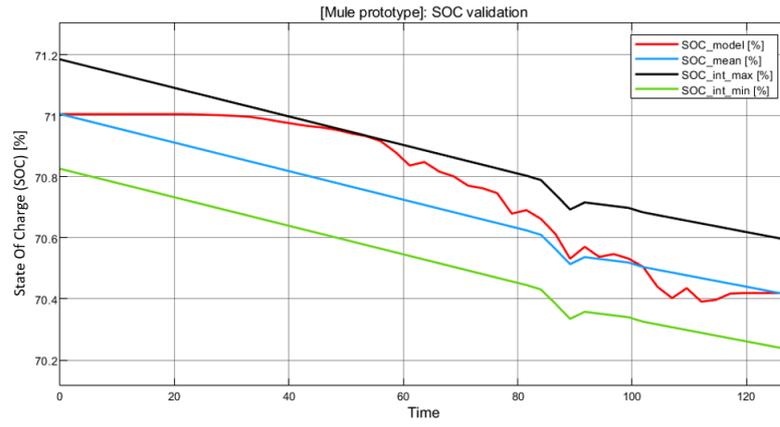


Figure 4.23: SOC model validation.

4.2 *Prototype 4*

Prototype 4 is the fourth prototype designed in prototype testing phase.

CAN acquisitions used for thesis simulations were get 5th July 2021 through ride track in collaboration with *Cecomp S.p.A.* engineers. *Cecomp S.p.A.* tests its prototypes and concept cars close by on of its plants, more specifically in Moncalieri, a city close by Turin.

The racetrack is *Club des Miles*. The track is 980 meters long in the asphalted part, with an average width between 6 and 9 meters and a dirt part dedicated to motard of 300 meters and is equipped with an Alfano magnetic strip for recording your lap times. Figure 4.24 displays the racetrack.



Figure 4.24: *Club des Miles* racetrack on street view.

The 5th July 2021 Marco Ghibaudi and *Cecomp S.p.A.* engineers did in total six ride tracks. In this section the main powertrain model results of one ride track are reported, while in Appendix C are displayed the other five results.

The ride track analysed has a duration of 277.5270 s, so quite longer than the simulation seen about the *Mule*. 277.5270 s is the value of **Stop Time** in *Simulink* environment.

As explained in Chapter 3, *Prototype 4 Simulink* model is more articulated than the one dedicate to the *Mule*. Motivation lies in the fact that CAN acquisitions on *Prototype 4* contain more powertrain informations.

4.2.1 Inputs

The *Simulink* model presents as *Inputs* the calibration on the accelerator and brake pedal pressure, differently from *Mule Prototype*. Inputs allows to start the simulation. As seen in Chapter 3, the first main subsystem is the one called **Normal Pedal Mapping**, which receives in input three informations:

- **GasPedalInput**, that is the accelerator pedal position during the track;
- **BrakePedalInput**, that is the brake pedal position during the track;
- **VehicleSpeed**, that is the *Microlino Prototype 4* speed during the track measured in $\frac{km}{h}$.

The three variables are reported in *Simulink* environment through a *From Workspace* block.

In Figure 4.25, the three inputs are displayed.

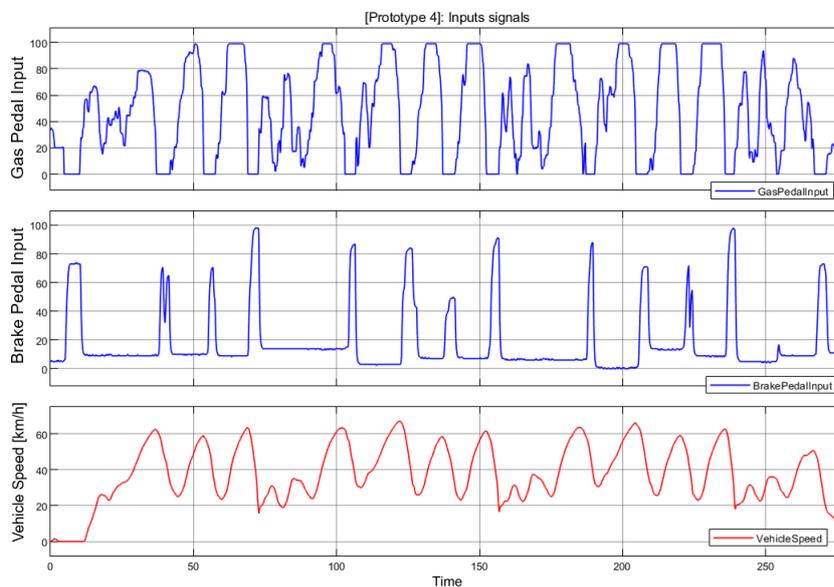


Figure 4.25: [*Prototype 4*] Inputs: Accelerator and Brake pedal position and Vehicle Speed waveforms.

The Maximum discharge and charge currents in 2s are considered as Inputs part, but they go directly to the **Energy Management - BMS** subsystem through a *From Workspace* block. Figure 4.26 displays the two waveforms.

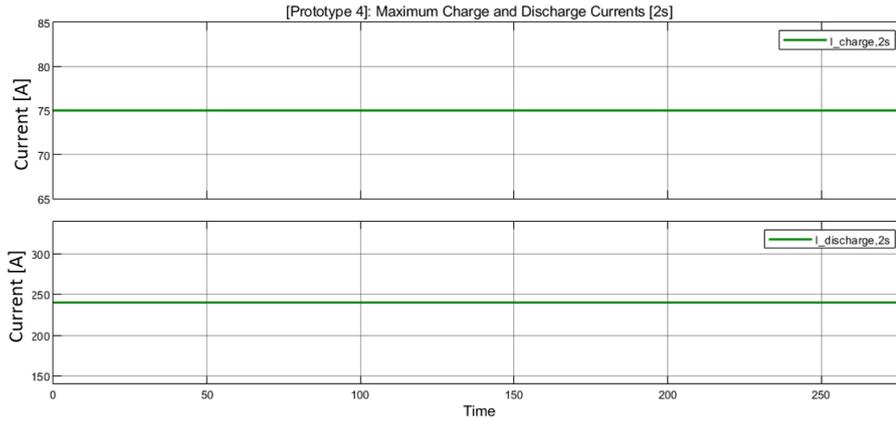


Figure 4.26: [*Prototype 4*] Inputs: Maximum discharge and charge currents waveforms.

4.2.2 Energy Management - BMS

The role of **Energy Management - BMS** subsystem is to check that the Reference Torque T_m^* , given by the **NormalPedalMapping** implementation, is acceptable doing a power balance to establish the torque upper and lower limit in the dynamic saturation action.

In Figure ??, T_m^* is displayed.

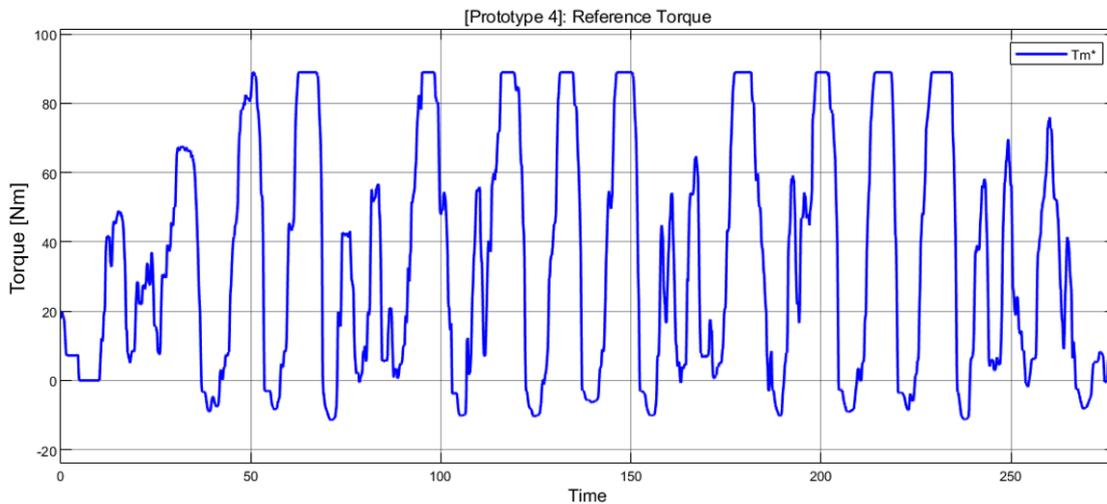


Figure 4.27: [*Prototype 4*] Energy Management - BMS: Reference Torque T_m^* waveform.

Another input of this subsystem is the Motor angular speed measured in $\frac{rad}{s}$. This variable is obtained converting the **VehicleSpeed** of Figure 4.25, due to the fact that *Prototype 4* CAN speed acquisition is taken from wheels, so a gear ratio has to be considered and also the conversion from $\frac{km}{h}$ to $\frac{rad}{s}$. In Figure 4.28, ω_m waveform is displayed.

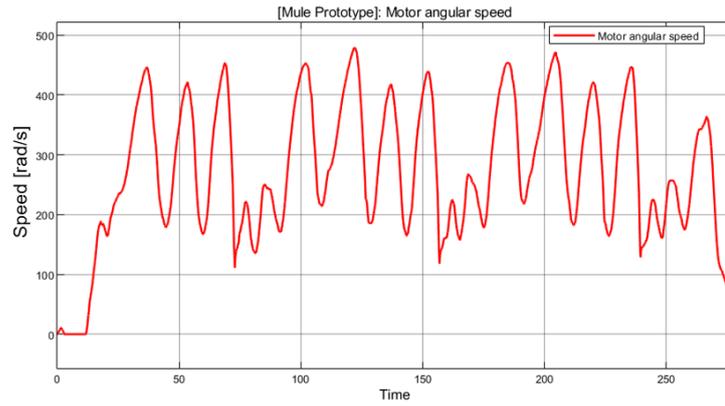


Figure 4.28: [Prototype 4] Energy Management - BMS: Motor Angular Speed ω_m waveform.

As explained in Chapter 3, **Energy Management - BMS** subsystem generates a torque T_m^{**} , which waveform and values stays inside the limits given by maximum charge and discharge currents, voltage battery and driveline efficiency.

Figure 4.29 displays this saturation action.

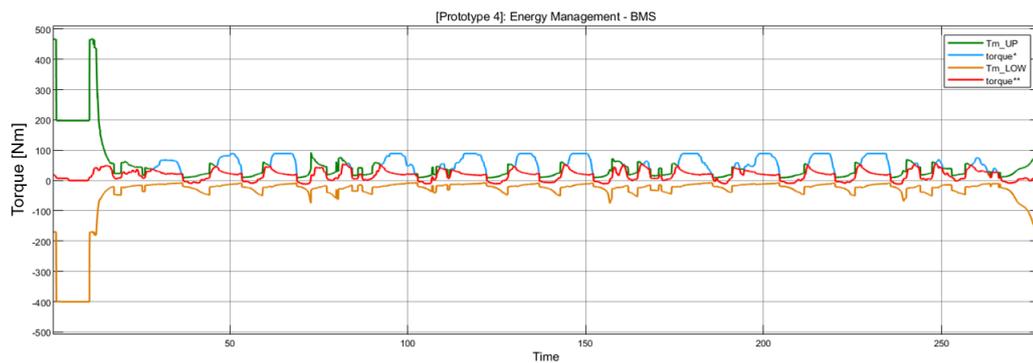


Figure 4.29: [Prototype 4] Energy Management - BMS: saturation action on T_m^* .

In Figure 4.30, there is the comparison between T_m^* and T_m^{**} .

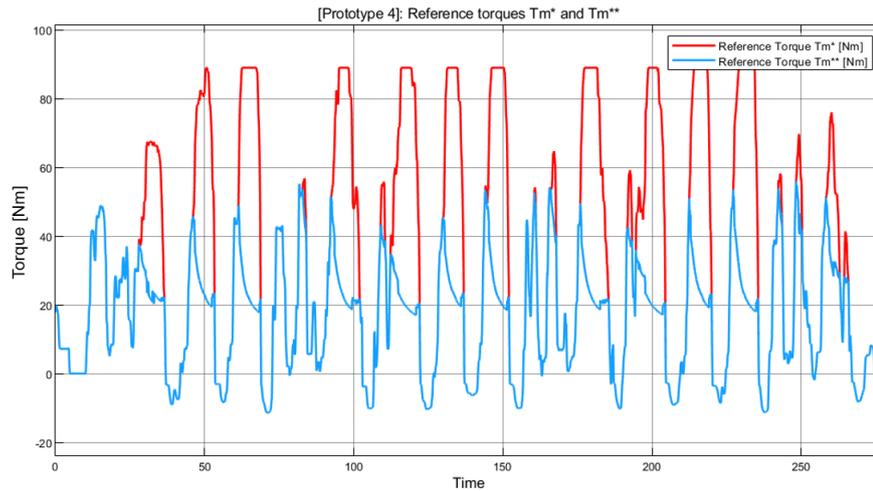


Figure 4.30: [Prototype 4] Energy Management - BMS: Torque comparison T_m^* and T_m^{**} .

Once obtained the checked motor torque T_m^{**} , multiplying it by the ω_m variable the motor output power $P_{m,out}$ is computed (an electric convention is kept main). $P_{m,out}$ waveform is displayed in Figure 4.31.

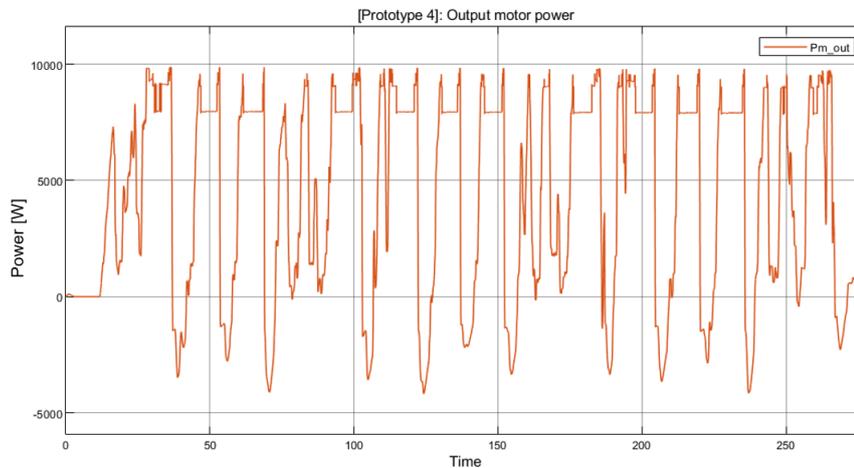


Figure 4.31: [Prototype 4] Energy Management - BMS: Output motor power $P_{m,out}$ waveform.

T_m^{**} is the reference motor torque, that supply the **Electric Motor** subsystem in the model.

4.2.3 Electric Motor

The **Electric Motor** subsystem computes the motor torque T_m through a **LPF**. Figure 4.32 displays the T_m waveform given by the filter motor model.

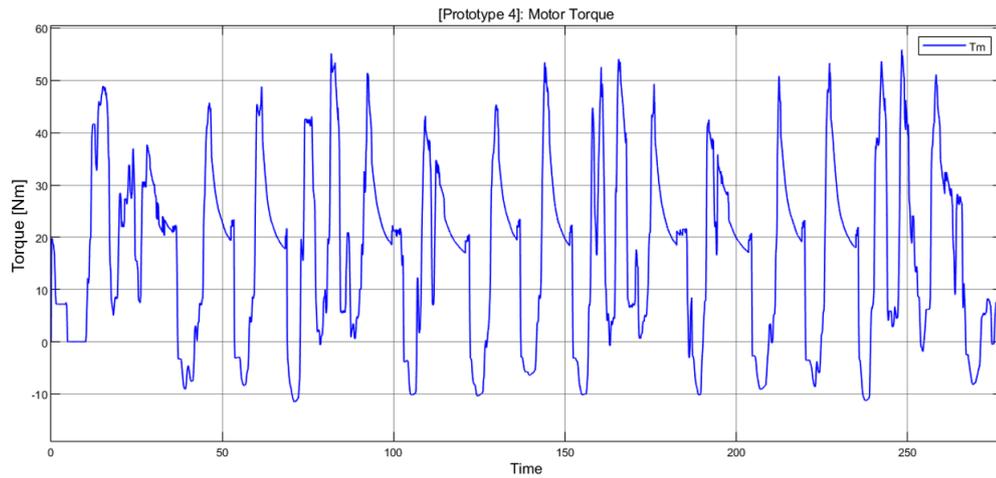


Figure 4.32: [*Prototype 4*] Electric Motor: Motor torque T_m .

The **Electric Motor** model, as seen in Chapter 3, need also the Motor angular speed value, that is the same of Figure 4.28.

T_m and ω_m are used to compute the Motor efficiency η_{motor} through *Efficiency Map*. Figure 4.33 displays the Motor efficiency values during the model simulation.

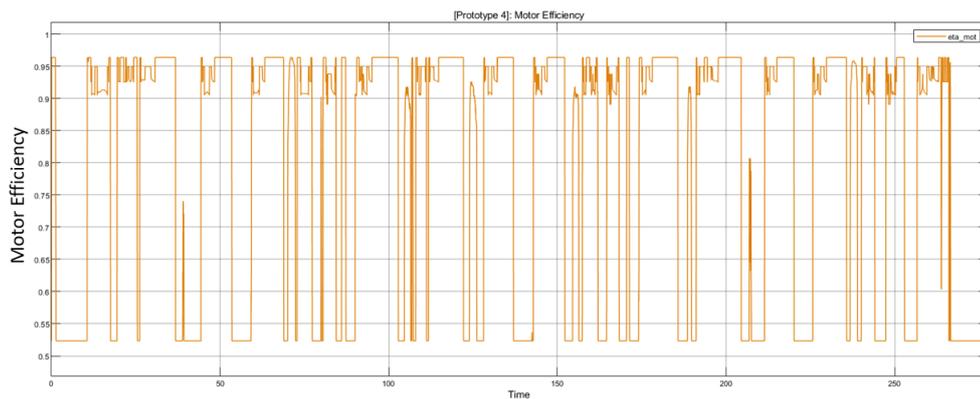


Figure 4.33: [*Prototype 4*] Electric Motor: Motor Efficiency η_{motor} waveform.

Diving $P_{m,out}$ by η_{motor} the input motor power $P_{m,in}$ is computed and its waveform is displayed in Figure 4.34.

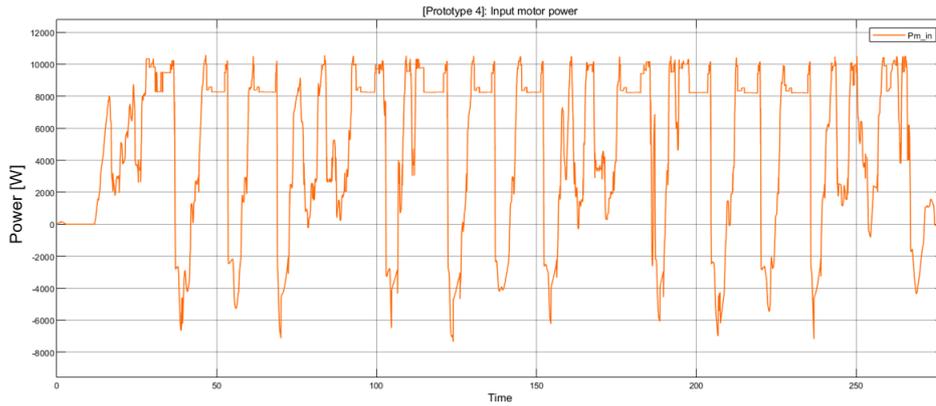


Figure 4.34: [*Prototype 4*] Electric Motor: $P_{m,in}$ waveform.

4.2.4 Inverter

$P_{m,in}$ is the variable that feeds the **Inverter** subsystem. Through *Efficiency Map*, so through $\eta_{inverter}$, Inverter block computes the Battery supply power $P_{dc,in}$, used to determine the Battery Current (I_{batt}).

Figure 4.35 displays the Inverter efficiency values during simulation.

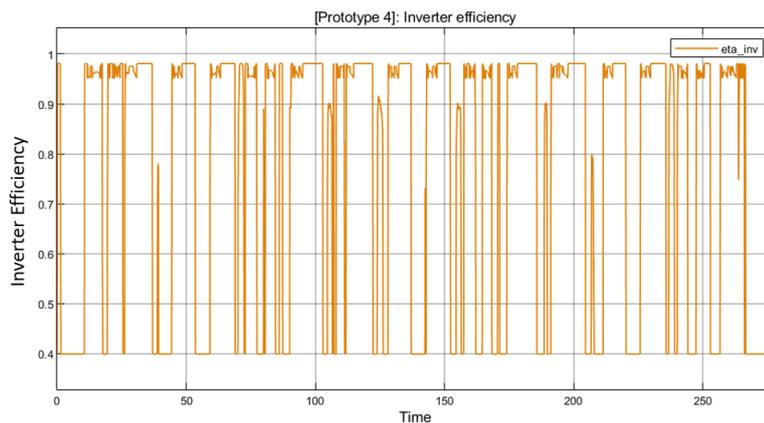


Figure 4.35: [*Prototype 4*] Inverter: Efficiency $\eta_{inverter}$ waveform.

Dividing $P_{m,in}$ by $\eta_{inverter}$, the Battery supply power $P_{dc,in}$ is computed. Waveform is displayed in Figure 4.36.

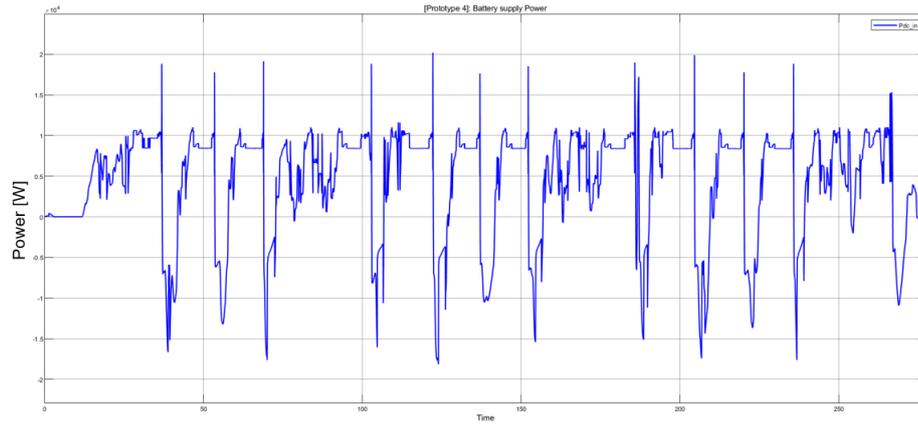


Figure 4.36: [*Prototype 4*] Inverter: Battery supply power $P_{dc,in}$ waveform.

In Figure 4.37, it is displayed the comparison between $P_{m,in}$ and $P_{dc,in}$ in the same scope plot.

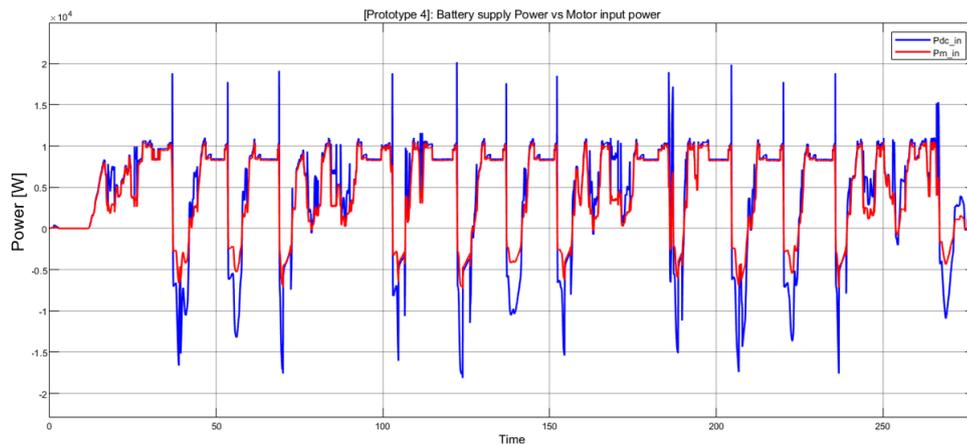


Figure 4.37: [*Prototype 4*] Inverter: Comparison between $P_{m,in}$ and $P_{dc,in}$ waveforms.

4.2.5 Battery

Also in this model, the Battery subsystem is the one which reaches the target of the model: SOC. Its inputs are the initial SOC and the battery current I_{batt} . The latter is computed considering the $P_{m,in}$ sign. This allows to understand in which quadrant, I or IV, the model is working. The quadrants are important for the battery current computation, as explained in Chapter 2, section 3.8.2

In Figure 4.38, the battery current waveform is displayed.

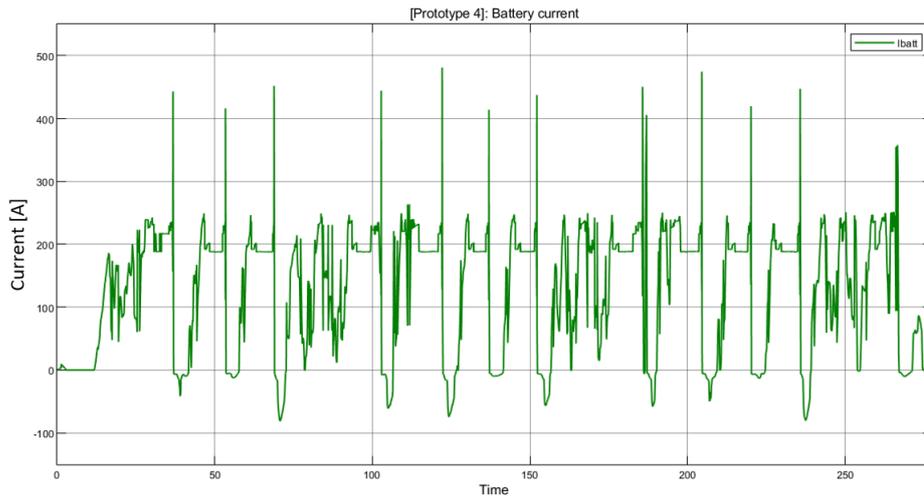


Figure 4.38: [*Prototype 4*] Battery: battery current I_{batt} waveform.

The Battery current is used in the Battery model in order to obtain:

- the Battery Voltage V_{batt} , used as feedback to determinate Battery Current. Figure 4.39 displays the Battery Voltage obtained by the model during the ride track simulation.

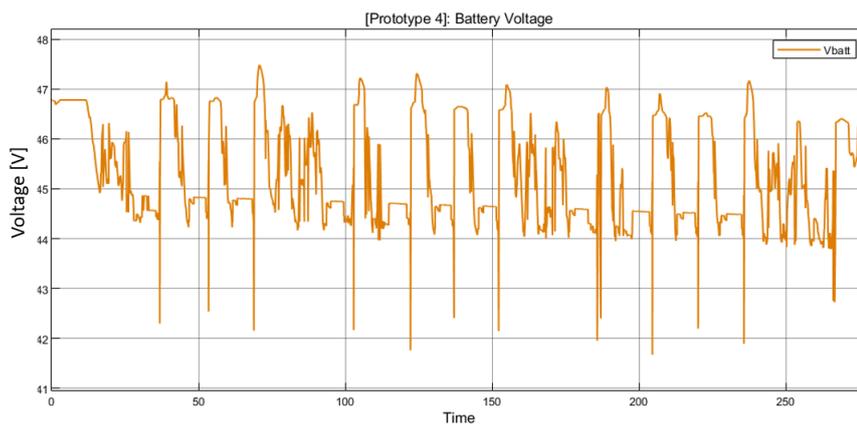


Figure 4.39: [*Prototype 4*] Battery: Battery Voltage V_{batt} waveform.

- State of Charge SOC , that is the main task of this model. Figure 4.40 displays the SOC trend obtained during the ride track.

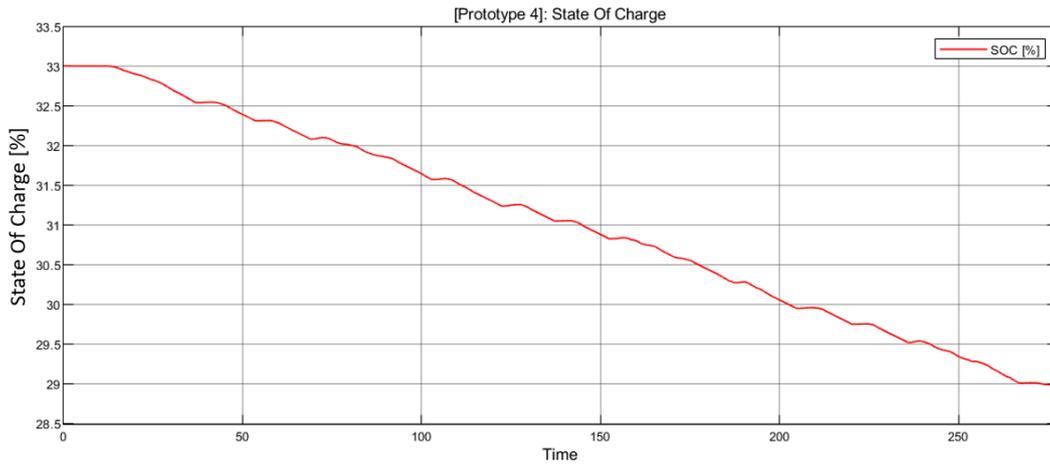


Figure 4.40: [*Prototype 4*] Battery: Model SOC.

Prototype 4 - State of Charge

Referring to Figure 4.40, the SOC during simulation differs between 33%, that is the initial one taken from the *.mat* CAN acquisition file, and 28.98%, in 277.5270s.

The obtained waveform of Figure 4.40 is compared with the CAN acquisition SOC waveform. The latter is displayed in Figure 4.41.

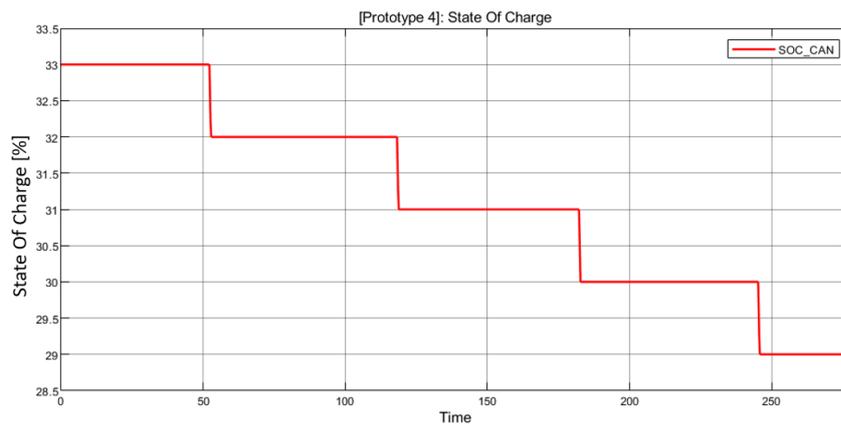


Figure 4.41: [*Prototype 4*] Battery: State Of Charge waveform obtained by CAN acquisition on ride track.

Waveforms comparison are displayed in Figure 4.42.

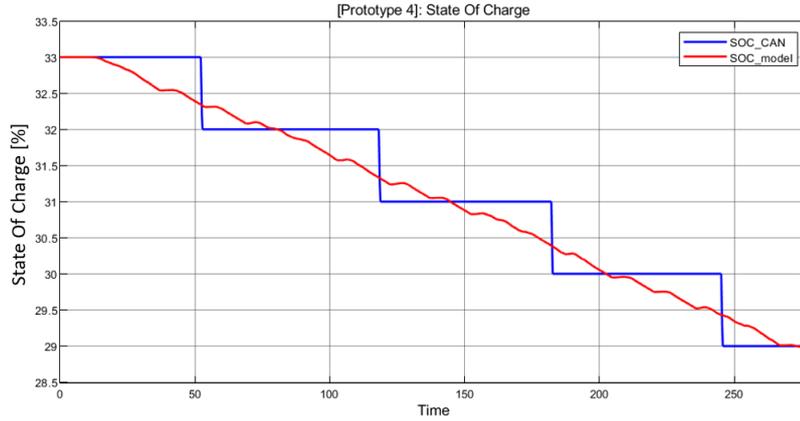


Figure 4.42: [*Prototype 4*] Battery: comparison between model SOC and CAN acquisition SOC.

The SOC acquired by CAN is a discrete waveform, while the SOC given by *Prototype 4* model is a time continuous trend. Initial and final SOC values are the same, so to ensure the model validation: a study on SOC mean values is considered.

Simulation stops at $277.5270s$, in this time range it is possible recognise 5 time sub-intervals in the SOC acquired by CAN.

Table 4.2 summarises the $P_{dc,in}$ and SOC_CAN values of these 5 time sub-intervals.

Table 4.2: Summary values of 5 time sub-intervals.

Time interval	seconds [s]	$P_{dc,in}$ average values [W]	SOC_CAN values [%]
<i>Time 1</i>	0s to 50.68s	4098.5337W	33%
<i>Time 2</i>	50.68s to 118.1812s	5360.8684W	32%
<i>Time 3</i>	118.1812s to 182.1742s	4908.1952W	31%
<i>Time 4</i>	182.1742s to 245.1806s	4816.0566W	30%
<i>Time 5</i>	245.1806s to 277.5270s	4911.8079 W	29%

In each time sub-interval CAN SOC is considered quite constant, assuming the values of Table 4.2 last column.

The study on SOC mean values is computed considering the $P_{dc,in}$ average values in the 8 time sub-intervals and the V_{batt} instantaneous values. Evaluate the the ratio between them, a new Battery current waveform $I_{batt,ist}$ is computed:

$$I_{batt,ist} = \frac{P_{dc,in,mean}}{V_{batt}} \quad (4.3)$$

The computed Battery Current is displayed in the Figure below, 4.43.

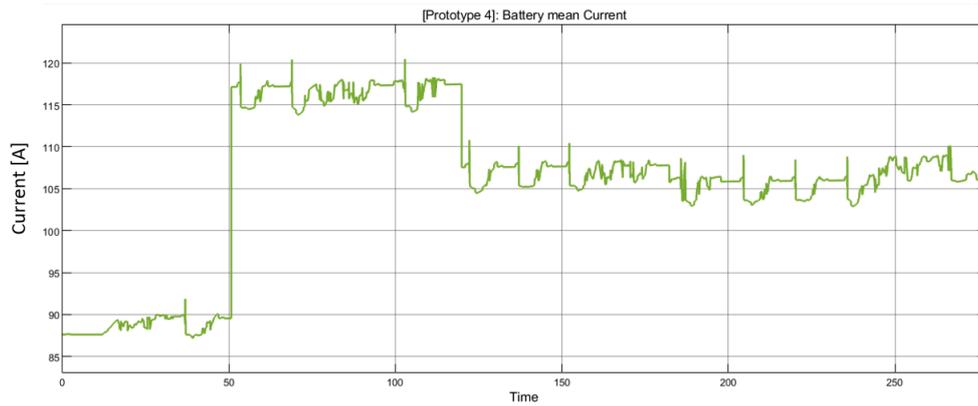


Figure 4.43: [Prototype 4] Battery: Battery Current waveform given by equation 4.3.

This waveform supply a new battery model, through a *From Workspace* block. The model is displayed in Figure 4.44 and it is used to compute the SOC mean values.

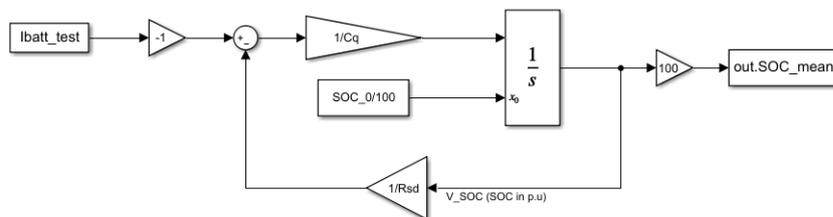


Figure 4.44: *Simulink* model used to compute the SOC mean.

Referring to the model of Figure 4.44, the output is a *To Workspace* block, contains the SOC mean values: waveform is displayed in Figure 4.45.

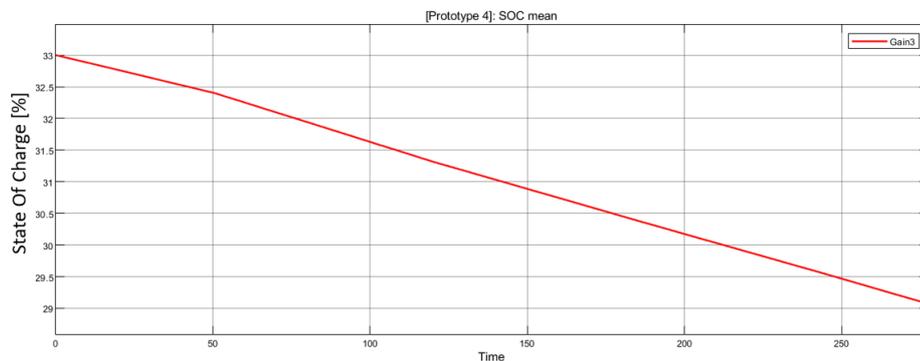


Figure 4.45: [Prototype 4] Battery: SOC mean values waveform.

The computed mean values SOC is compared with the waveforms displayed in Figure 4.42. Comparison results are shown in Figure 4.46.

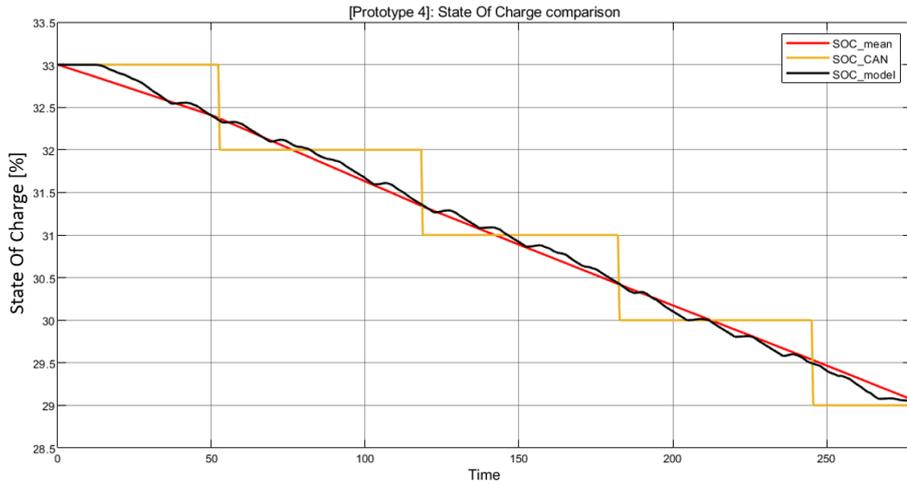


Figure 4.46: [Prototype 4] Battery: comparison between CAN SOC, Model SOC and mean values SOC.

Observing Figure 4.46, it can be said that the *Prototype 4* validation model is quite reach: due to the fact that the difference between the red waveform (*SOC_model*) and the blue one (*SOC_mean*) is very minimal.

To prove the validation standard deviation σ is taking into account. Its formulation is expressed as

$$\sigma = \sqrt{\frac{\sum_N (x - \bar{x})^2}{N}} \tag{4.4}$$

where

- x represents the value assumed by the SOC in the model;
- \bar{x} is the value mean value assumed by the SOC;
- N is the samples numeber of the simulation.

The σ is equal to 0.1549. The latter value is used to create a band: if the SOC obtained by the model is inside this band, the model can consider validated.

Figure 4.47 displays the model validation.

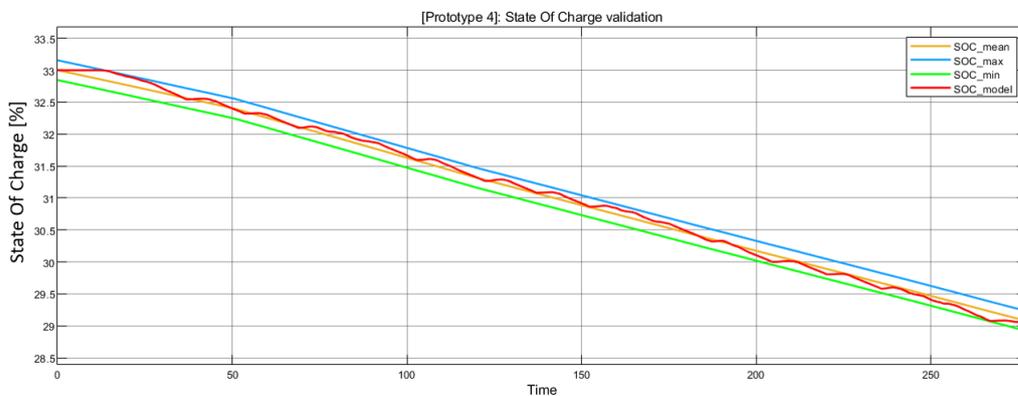


Figure 4.47: [Prototype 4] Battery: SOC model validation.

Chapter 5

Conclusion

5.1 Personal contribution

This thesis represented an important part of my training personal. Thanks to it, I got in touch with the industrial reality, during the thesis implementation.

Therefore, I was able to understand what means working in a Company and what kind of electrical engineer I want to be.

I am proud of this thesis work due to the fact that it was the first project in my career totally done alone, following the advices and guide of my supervisors.

Therefore, this section summarises my personal contributions, that are:

- literature review concerning ICEV problems and consequences on global warming, *Microfino* concept and prospective and Powertrain theoretical background;
- theory behind the main powetrain elements used for modeling;
- Driveline, Electric Motor and Inverter *Efficiency Maps* implementations from manufacturers data;
- *Simulink* models implementation;
- conversion of *CAN* acquisitions on ride track from *CANalyzer* file to *MATLAB* data file;
- simulations and relative waveform plots;
- simulation results comparison with the one acquired by *CAN*, especially about the SOC values and waveforms;
- model validation.

The powertrain energetic *Simulink* model is considered validated, through data acquired on the vehicle.

The SOC standard deviation is about 17%. This value is considered satisfying, considering the parameters data availability.

5.2 Future work

The reached results can be considered satisfying, thank to the low differentiations between the model and CAN values.

Simulink implemented models are a quite simple modeling of the electric powertrain of this small BEV called *Microfino*.

This model can be used in future if other tests or simulations are necessary on *Microfino*: the main informations needed are the driving cycles in terms of torque, speed or pedal position. In this way it is possible to reach consumptions during the cycle.

Also the model is a general Electric Powertrain model, where parameters are calibrated on *Microfino*. Through the *MATLAB* script it is possible change the main parameters and also the *Efficiency Maps*, in order to analyse another BEV or EV.

Bibliography

- [1] Arif, Syed Muhammad and Lie, Tek Tjing and Seet, Boon Chong and Ayyadi, Soumia and Jensen, Kristian, *Review of electric vehicle technologies, charging methods, standards and optimization techniques*. In *Electronics* volume 10, number 16, pages 1910; Publisher *Multidisciplinary Digital Publishing Institute*, 2021.
- [2] Yong, Jia Ying and Ramachandaramurthy, Vigna K and Tan, Kang Miao and Mithulananthan, Nadarajah, *A review on the state-of-the-art technologies of electric vehicle, its impacts and prospects*. In *Renewable and sustainable energy reviews*, Vol. 49, pages 365-385; Publisher *Elsevier*, 2015.
- [3] Kumar, Rajeev Ranjan and Alok, Kumar, *Adoption of electric vehicle: A literature review and prospects for sustainability*. In *Journal of Cleaner Production*, Vol. 253, pages 119911; Publisher *Elsevier*, 2020.
- [4] Micro, <https://microlino-car.com/en/microlino>.
- [5] Iso *Isetta*, <https://en.wikipedia.org/wiki/Isetta>.
- [6] Comparison *Microlino-Iso Isetta*, 2016, <https://www.carsales.com.au/editorial/details/microlino-isetta-for-a-new-era-101698/>.
- [7] Pavlovic, Ana, and Cristiano Fragassa, *General consideration on regulations and safety requirements for quadricycles*. In *International Journal for Quality Research*, Vol. 9.4 , 2015.
- [8] Microlino-Bylogix, (2021), <https://www.bylogix.it/2021/01/14/microlino/>.
- [9] Bicek, Matej and Pepelnjak, Tomaz and Pusavec, Franci, *Production aspect of direct drive in-wheel motors*. In *Procedia CIRP*, Vol. 81, pages 1278-1283; 2019.
- [10] R. Bojoi, (2021). *Electrical drives for eMobility* [PowerPoint slides].
- [11] Zhang, Ruifeng, et al., *A study on the open circuit voltage and state of charge characterization of high capacity lithium-ion battery under different temperature*. In *Energies*, 11.9 (2018): 2408.
- [12] M. Agne, (2020). *Dana TM4, SME Motors Overview*, [PowerPoint slides].
- [13] MathWorks, (2021). <https://it.mathworks.com/>
- [14] Mahmoudi, A., Soong, W. L., Pellegrino, G., Armando, E. *Maps of electrical machines*. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 2791-2799; 2015.
- [15] International Energy Agency website, <https://www.iea.org/>.

Appendices

Appendix A

List of Acronyms

AC – Alternate Current

BCM - Body Control Module

BMS - Battery Management System

CAN - Control Area Network

DC - Direct Current

E/E - Electric/Electronic

ECU - Electronic Control Unit

EM - Electrical Motor

EV - Electric Vehicle

GHG - Green-Houses Gases

HEV - Hybrid Electric Vehicle

HMI - Human Machine Interface

ICE - Internal Combustion Engine

ICEV - Internal Combustion Engine Vehicle

IPM - Internal Permanent Magnet

LPF - Low Pass Filter

LUT - Look-Up Table

NdFeB - Neodymium magnet

PM - Permanent Magnets

PM-SyR - PM-assisted Synchronous Reluctance

PWT - Powertrain

RMS - Root Mean Square

SmCo - Samarium–Cobalt magnet

SOA - Safe Operating Area

SOC - State Of Charge

SPM - Surface-mounted Permanent Magnet

SyR - Synchronous Reluctance

VCU - Vehicle Control Unit

Appendix B

List of Symbols

ξ - anisotropy

L_d - direct synchronous inductance of the machine

L_q - synchronous inductance quadrature of the machine

λ_m - PM flux

T_{PM} - Permanent Magnets torque [Nm]

T_{rel} - Reluctance torque [Nm]

T_m^* - Reference torque given by CAN [Nm]

T_m^{**} - Reference Electric Motor torque [Nm]

$P_{m,out}$ - Output Motor Power [W]

$P_{m,in}$ - Input Motor Power [W]

$P_{dc,in}$ - Battery Supply Power [W]

I_{batt} - Battery Current [A]

V_{batt} - Battery Voltage [V]

$\eta_{driveline}$ - Driveline efficiency

η_{motor} - Electric Motor efficiency

$\eta_{inverter}$ - Inverter efficiency

$I_{charge,[2s]}$ - Maximum charging current [A]

$I_{discharge,[2s]}$ - Maximum discharging current [A]

ω_m - Electric Motor Angular Speed [$\frac{rad}{s}$]

C_Q - Battery Capacity [F]

R_{sd} - Self-discharging resistance [Ω]

R_0 - Internal Resistance [Ω]

SOC_0 - Initial SOC

V_{OC} - Open-Circuit Voltage [V]

Appendix C

Prototype 4 - simulation results

This part of the Appendix summarises other main model simulation results about *Prototype 4*, to demonstrate that the *Simulink* thesis model can be consider validated.

CAN acquisitions of the 5th July 2021 were in total 5, done on the same racetrack.

C.0.1 Ride track 2

Ride track 2 was done after the ride track analysed in Chapter 4. Racetrack duration, so simulation time, is 191.3940s. Main waveforms are displayed in the below figures.

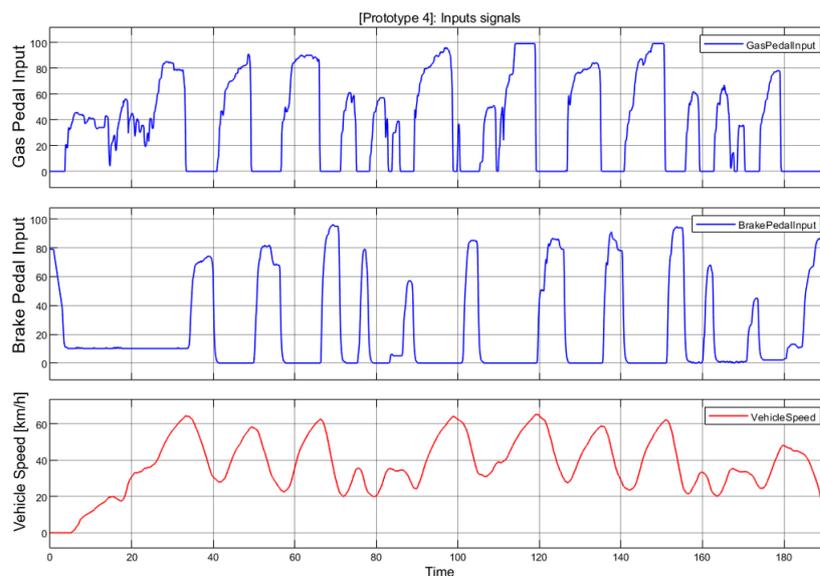


Figure C.1: [*Prototype 4*]: Model Input waveforms: *GasPedalInput*, *BrakePedalInput*, *VehicleSpeed*.

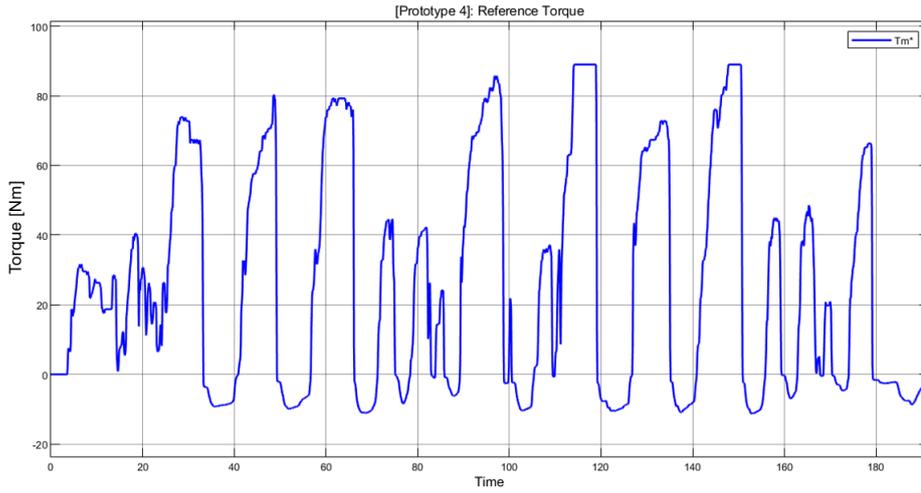


Figure C.2: [*Prototype 4*]: Reference torque T_* waveform.

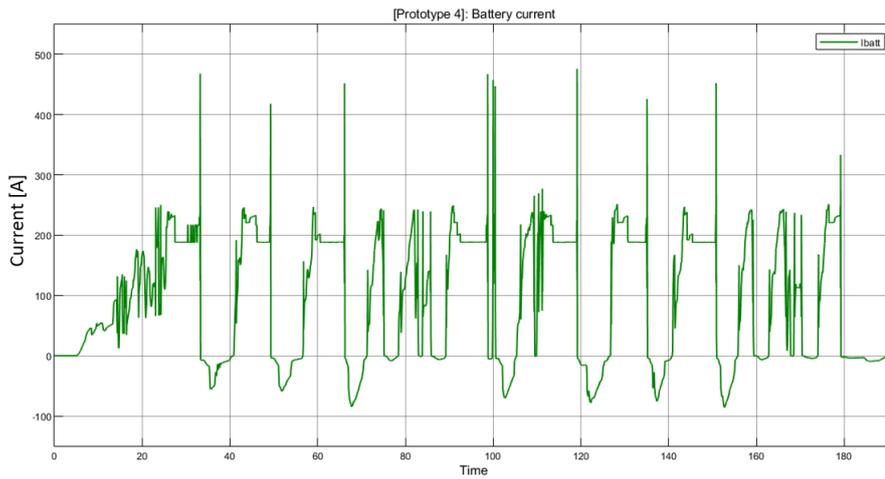


Figure C.3: [*Prototype 4*]: Battery current I_{batt} waveform.

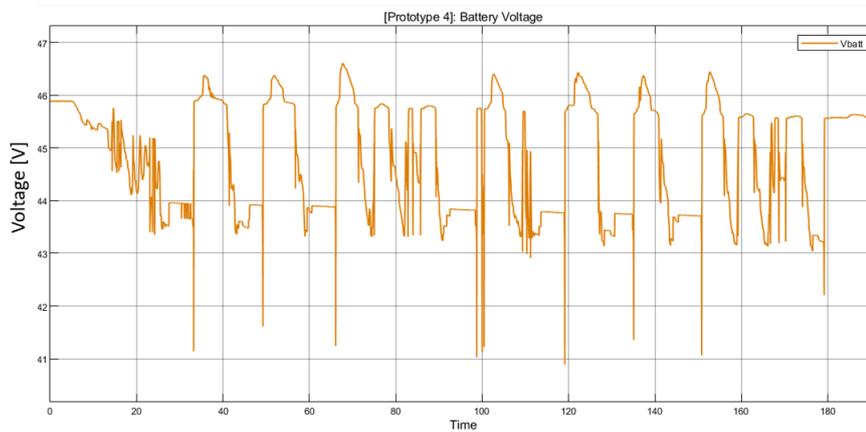


Figure C.4: [*Prototype 4*]: Voltage Battery V_{batt} waveform.

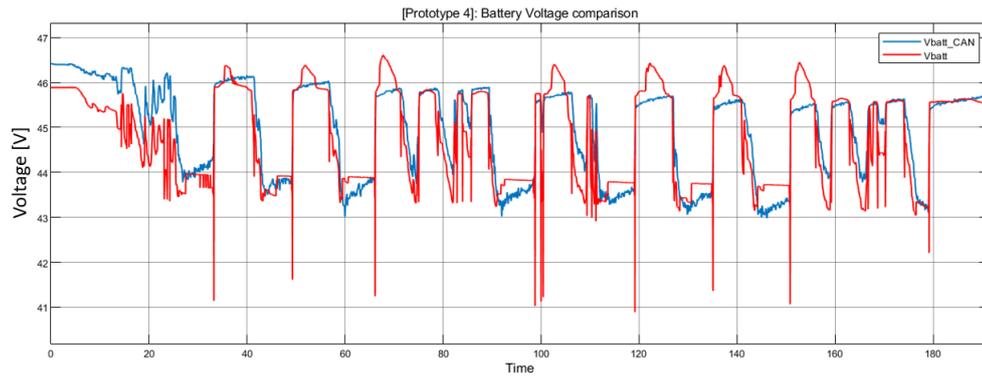


Figure C.5: [Prototype 4]: Voltage Battery comparison with CAN acquisition waveforms.

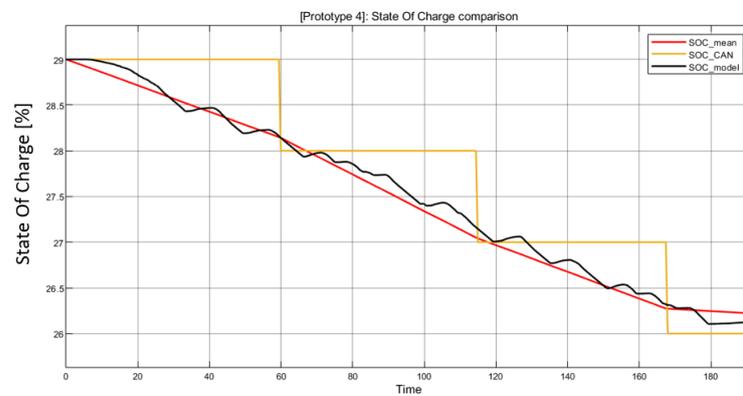


Figure C.6: [Prototype 4]: SOC waveform comparison between model and CAN acquisition.

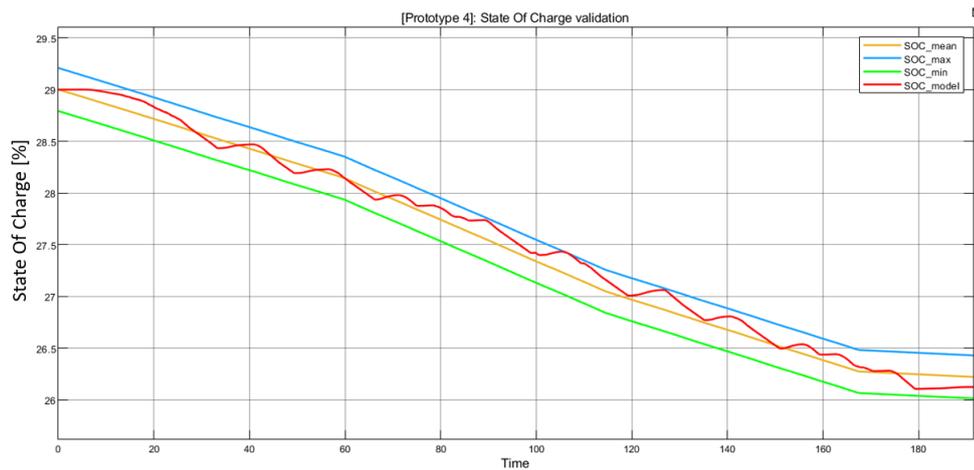


Figure C.7: [Prototype 4]: SOC validation.

C.0.2 Ride track 3

Ride track 3 was done after the ride track 2.

Racetrack duration, so simulation time, is 217.8930s.

Main waveforms are displayed in the below figures.

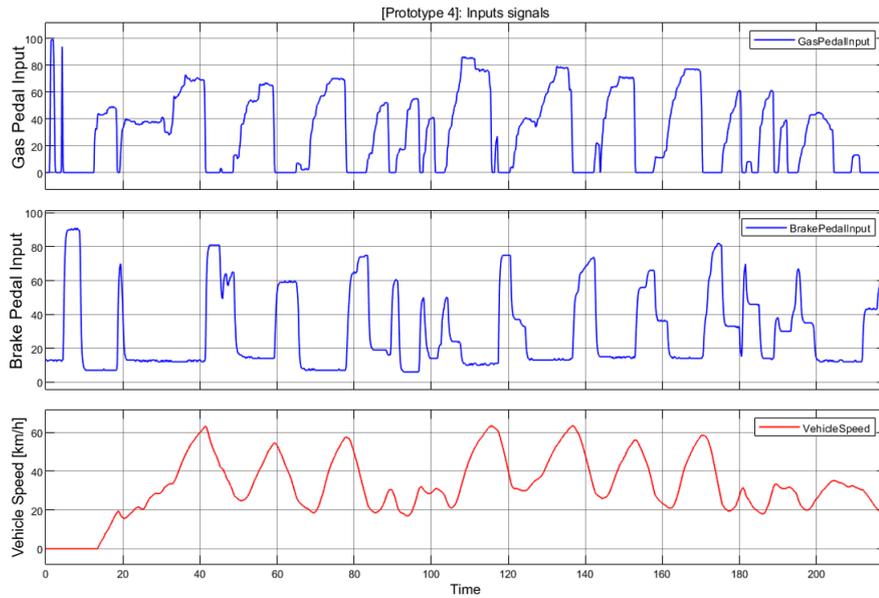


Figure C.8: [*Prototype 4*]: Model Input waveforms: *GasPedalInput*, *BrakePedalInput*, *VehicleSpeed*.

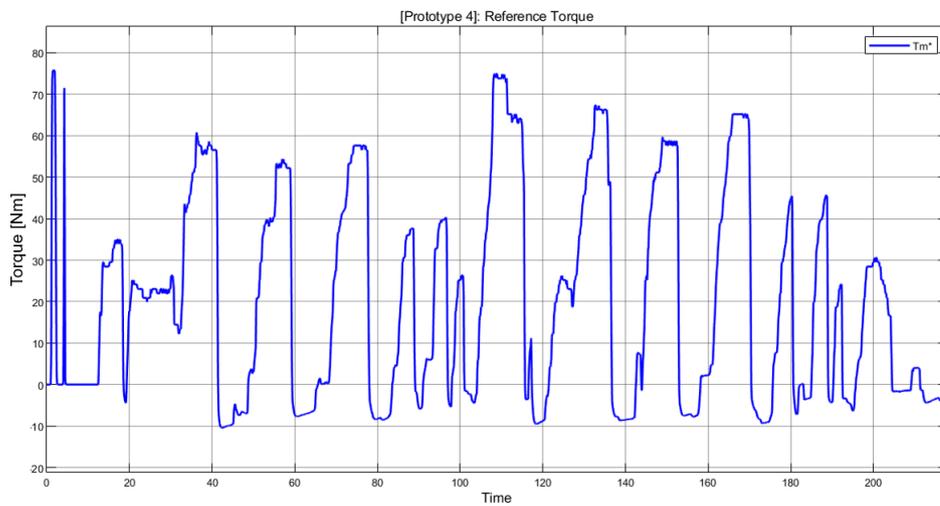


Figure C.9: [*Prototype 4*]: Reference torque T_* waveform.

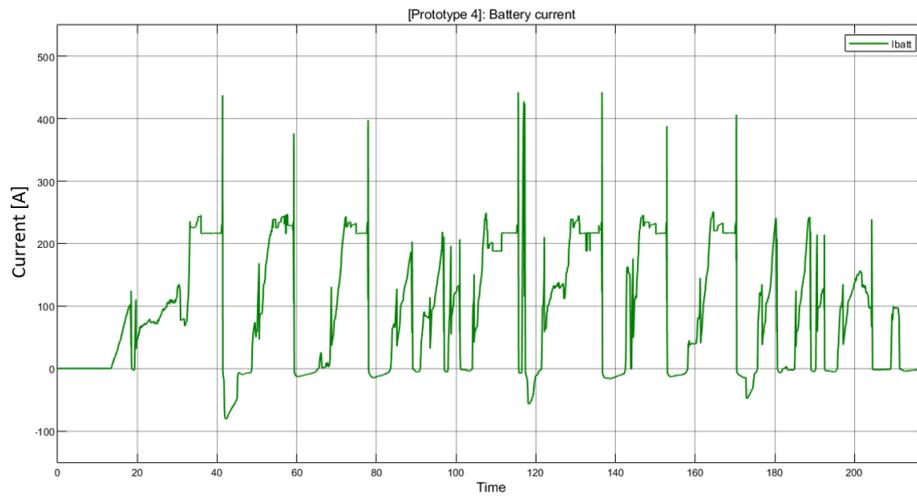


Figure C.10: [Prototype 4]: Battery current I_{batt} waveform.

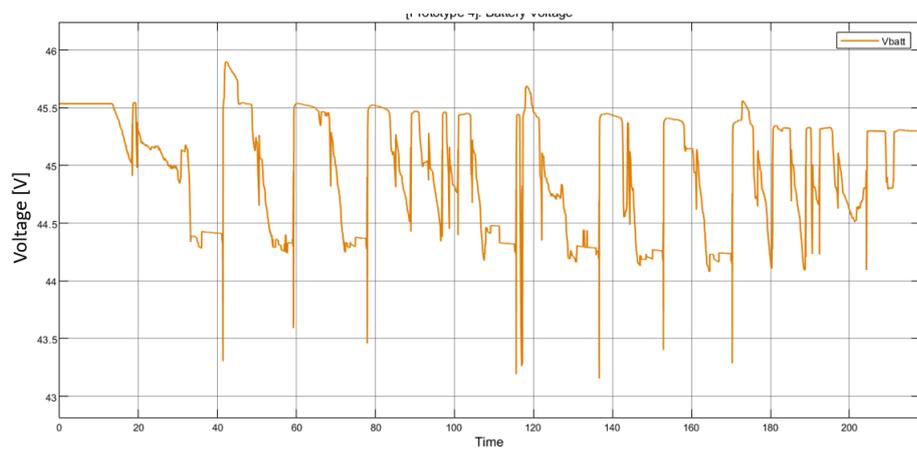


Figure C.11: [Prototype 4]: Voltage Battery V_{batt} waveform.

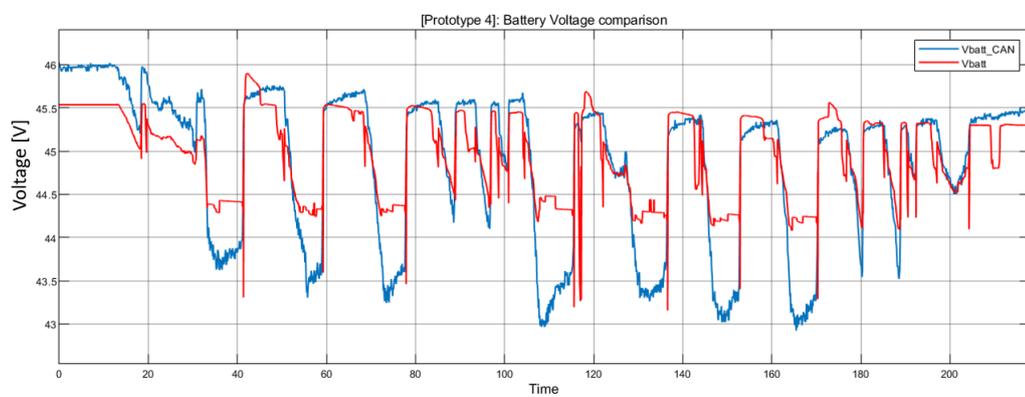


Figure C.12: [Prototype 4]: Voltage Battery comparison with CAN acquisition waveforms.

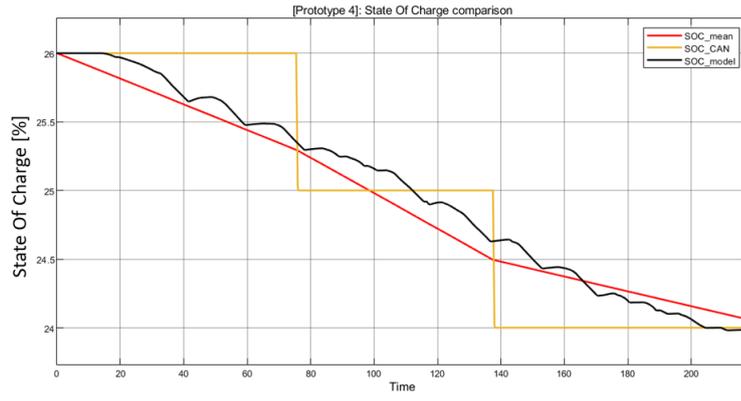


Figure C.13: [*Prototype 4*]: SOC waveform comparison between model and CAN acquisition.

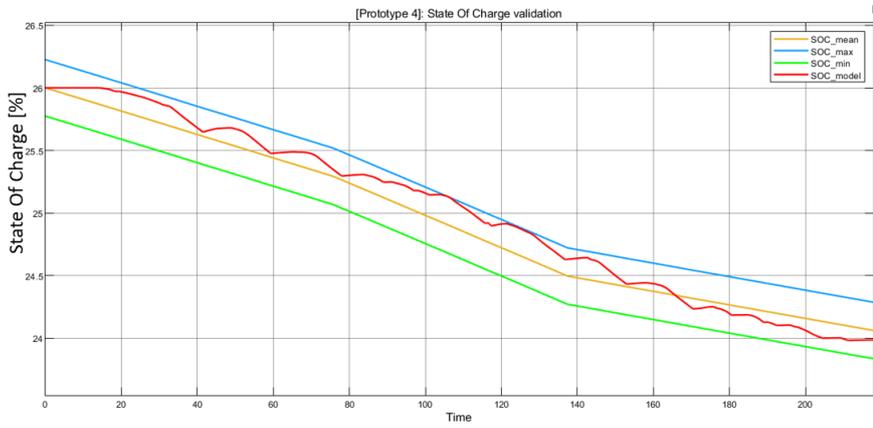


Figure C.14: [*Prototype 4*]: SOC validation.

C.0.3 Ride track 4

Ride track 4 was done after the ride track 3.
 Racetrack duration, so simulation time, is 218.1749s.
 Main waveforms are displayed in the below figures.

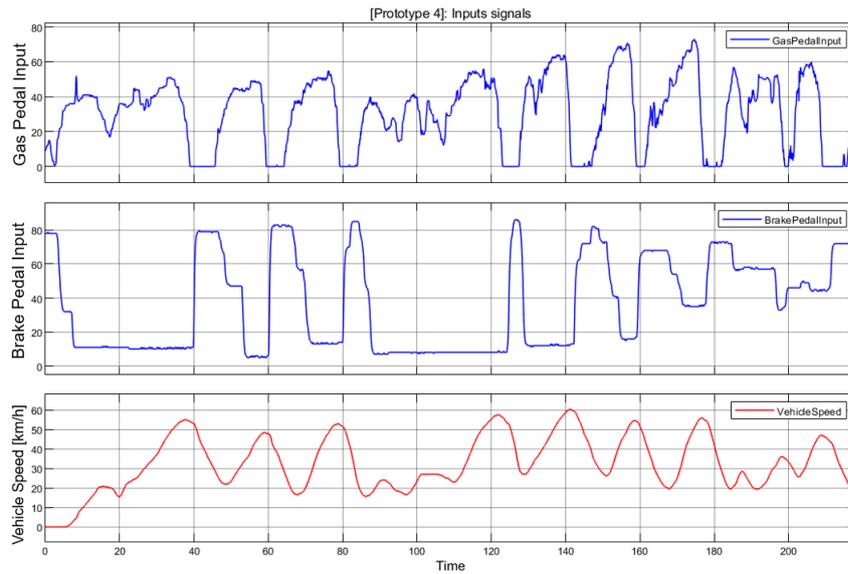


Figure C.15: [*Prototype 4*]: Model Input waveforms: *GasPedalInput*, *BrakePedalInput*, *VehicleSpeed*.

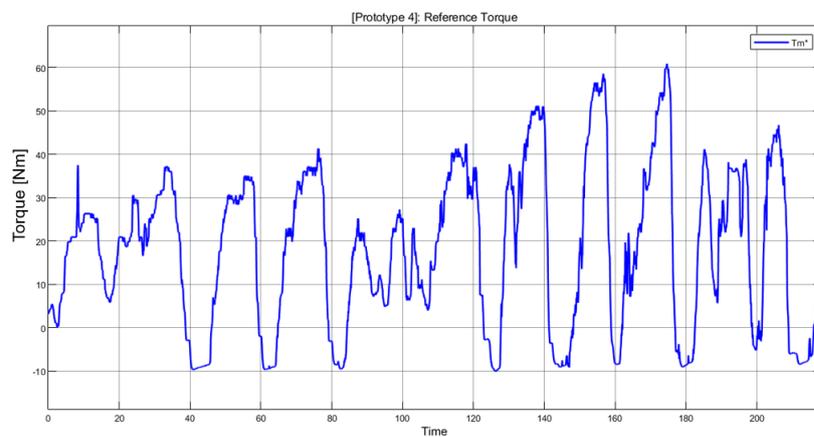


Figure C.16: [*Prototype 4*]: Reference torque T_* waveform.

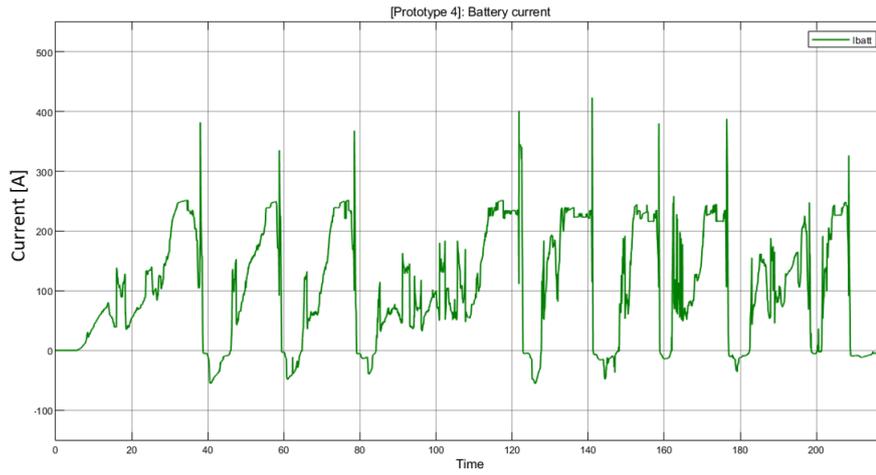


Figure C.17: [*Prototype 4*]: Battery current I_{batt} waveform.

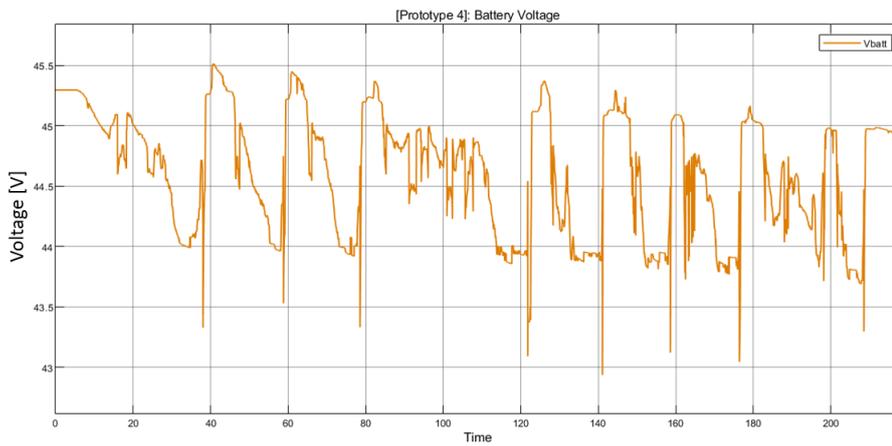


Figure C.18: [*Prototype 4*]: Voltage Battery V_{batt} waveform.

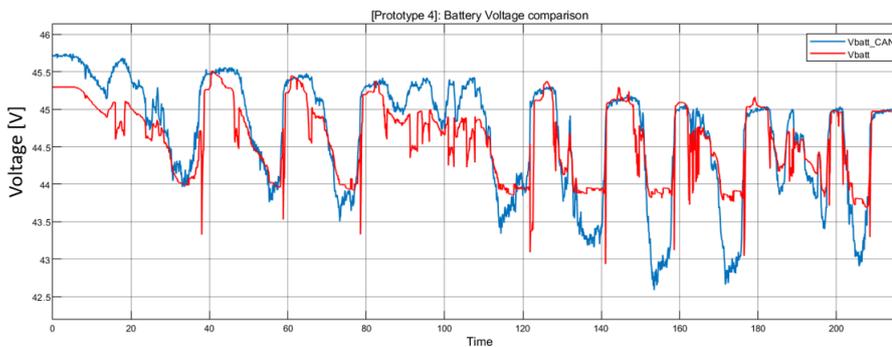


Figure C.19: [*Prototype 4*]: Voltage Battery comparison with CAN acquisition waveforms.

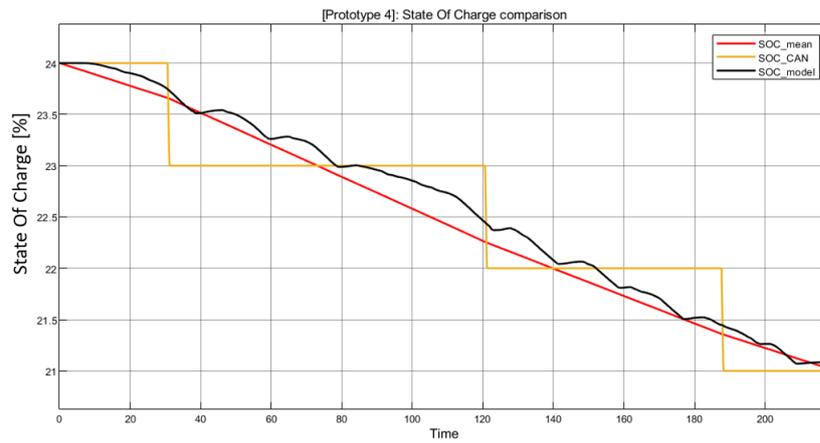


Figure C.20: [*Prototype 4*]: SOC waveform comparison between model and CAN acquisition.

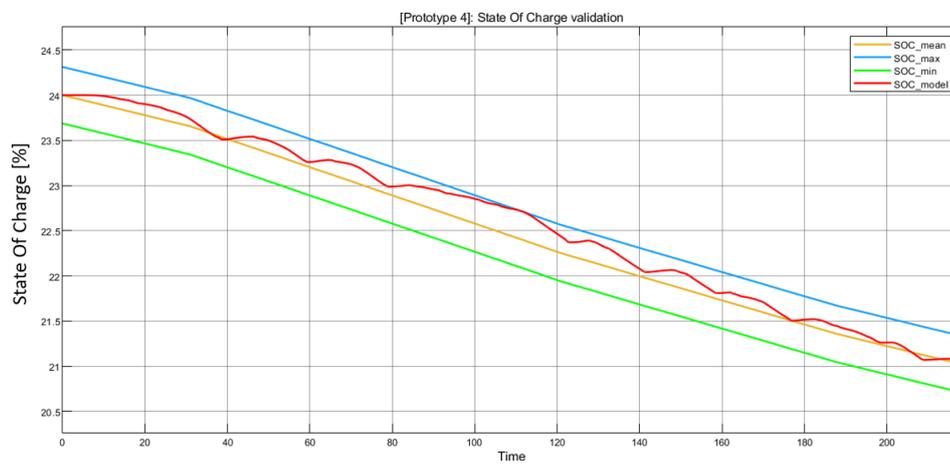


Figure C.21: [*Prototype 4*]: SOC validation.

C.0.4 Ride track 5

Ride track 5 was done after the ride track 4.

Racetrack duration, so simulation time, is 246.1490s.

Main waveforms are displayed in the below figures.

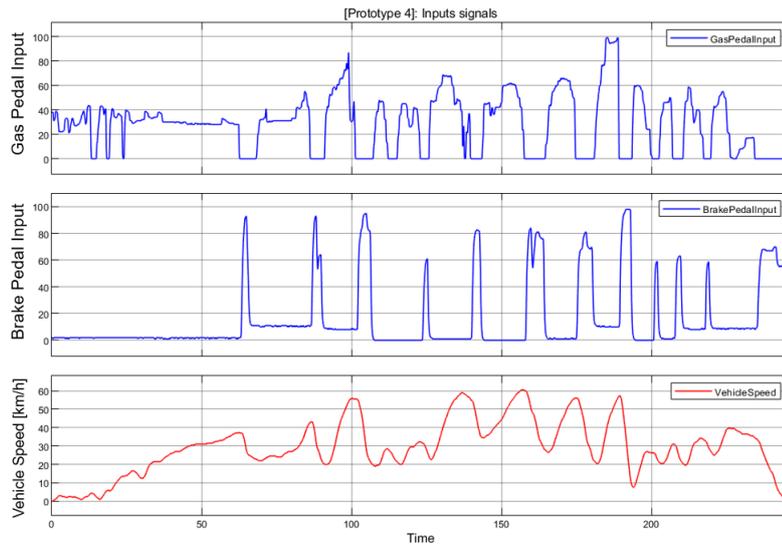


Figure C.22: [*Prototype 4*]: Model Input waveforms: *GasPedalInput*, *BrakePedalInput*, *VehicleSpeed*.

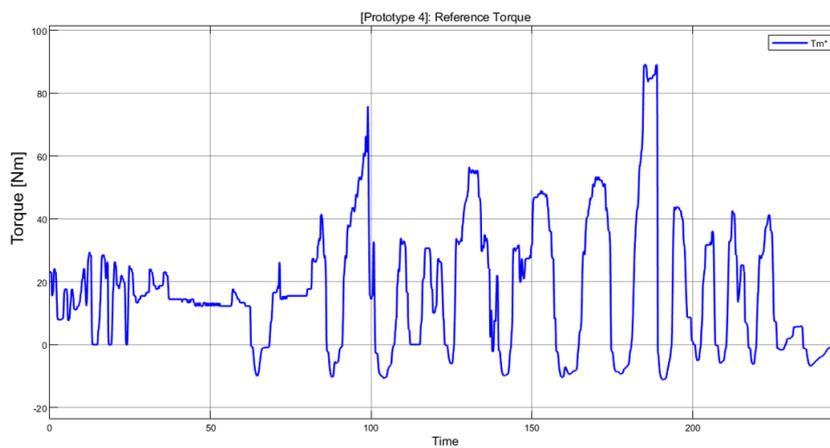


Figure C.23: [*Prototype 4*]: Reference torque T_* waveform.

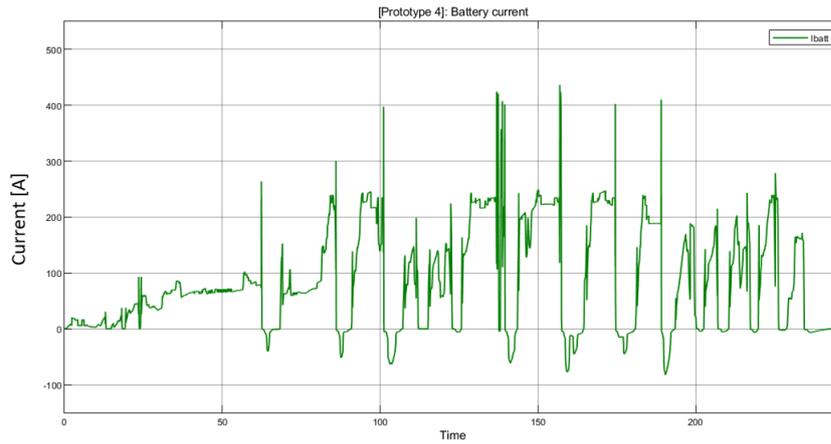


Figure C.24: [Prototype 4]: Battery current I_{batt} waveform.

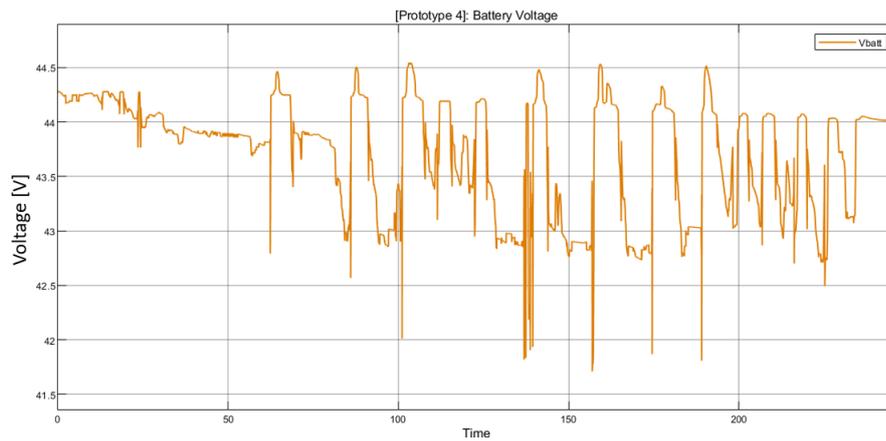


Figure C.25: [Prototype 4]: Voltage Battery V_{batt} waveform.

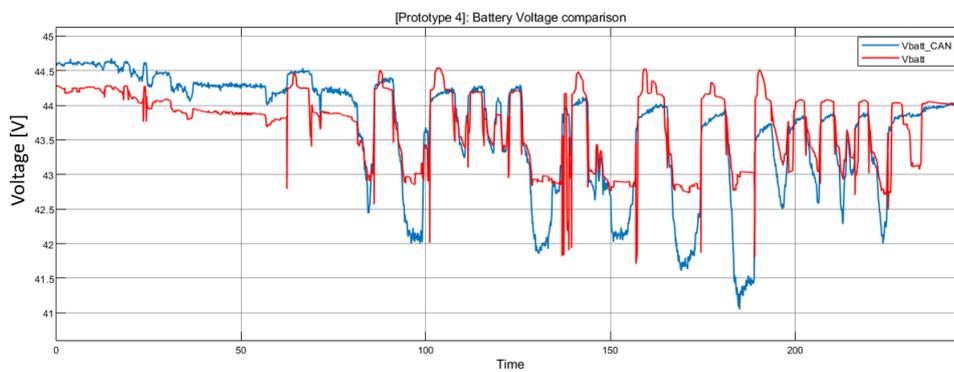


Figure C.26: [Prototype 4]: Voltage Battery comparison with CAN acquisition waveforms.

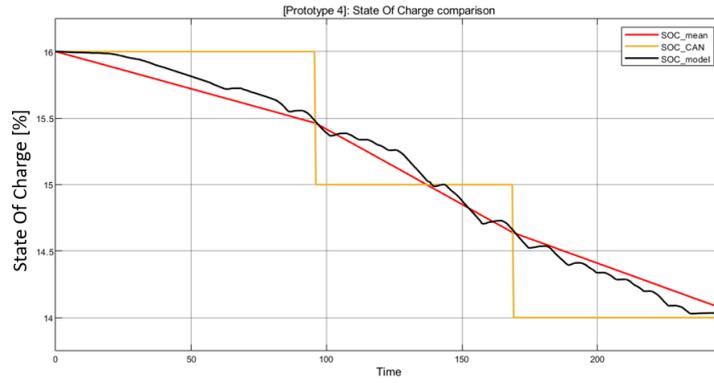


Figure C.27: [*Prototype 4*]: SOC waveform comparison between model and CAN acquisition.

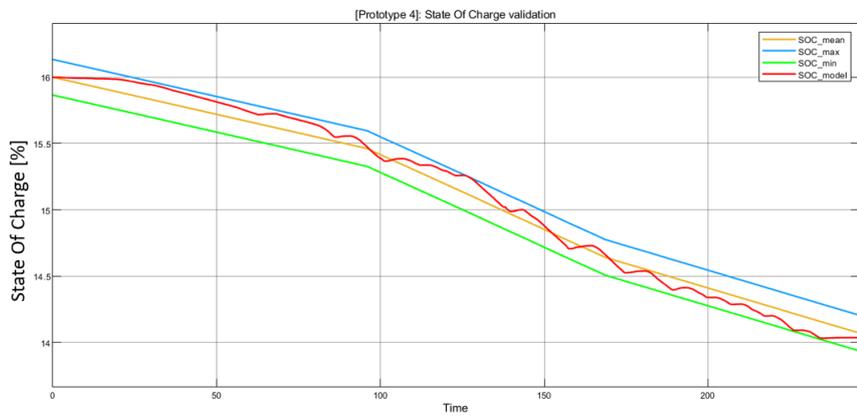


Figure C.28: [*Prototype 4*]: SOC validation.