



**Politecnico
di Torino**

Politecnico di Torino

Master of Science in Mechatronic Engineering

Academic year 2020/2021

Graduation Session October 2021

Analysis of the context of a rally competition for the modelling of a hybridized vehicle

following the V-Cycle development strategy

Supervisor:

Stefano Carabelli

Author:

Emanuele Carella

26 October 2021

*" I burn bright as the sun...and after that only white ashes remain. There is no halfway, only
ashes remain afterward"*

Joe Yabuki [Ikki Kajiwarra]

FOREWORD.	4
OBJECTIVES.	5
ORGANIZATION OF THE WORK.	6
PART 1: ANALYSIS OF THE PANDA RAID.	7
General Overview on the competition	7
Rules of the Competition.	8
The Path.	8
Ground characteristics.	10
Dry tarmac.	10
Dry and Dusty soil.	11
Sand.	12
Vertical and pitch displacements.	14
Road grade.	15
Other problems.	16
The starting vehicle.	17
Vehicle's setup.	17
Hybridization strategy.	18
Hybrid architecture.	18
PART 2: MODEL DESIGNING FOLLOWING THE V-CYCLE DEVELOPMENT STRATEGY.	20
Introduction of the V-cycle software development process	20
Vehicle model and degree of freedom of approximation	24
Vehicle model in Simulink	25
Model of suspension system	28
Modelling of the vertical displacement signal, slope signal ground characteristic signal	31
Model of tire.	32
Tire stiffness and damping.	33
Longitudinal Slip	33
Rolling Resistance and Shear Force distribution	36
Effect of Road conditions	37
Effect of speed	37
Tire model for longitudinal behaviour	37
Functional requirements for the tire model	38

Model in the Loop simulation for the 3DOF degree of freedom	40
3 Degree of freedom dynamics, test on a flat path in Dry condition	40
3 Degree of freedom dynamics, test in Second Scenario for different road condition	41
The 4WD-H implementation	43
Introduction	43
Control Algorithm modelling and focus on the Simulink requirements tool.....	43
Decision of the torque split strategy.	43
Function Requirements.....	44
Functional requirements of the Control Algorithm by means of Simulink Requirements tool	45
Model in the Loop simulation for the EM activation	49
Simulation for the torque split ruled by the driver.	49
Simulation test in Second Scenario for different road conditions	49
Simulation test in Third Scenario for different road conditions.....	50
Energy consumption analysis of the EM for a proper determination of the battery capacity	52
Introduction	52
Analysis for the determination of the battery capacity	53
Simulations	54
Preliminary analysis	54
Simulation of First Scenario	54
Simulation of Second Scenario	55
Simulation of Third Scenario	56
Conclusion.....	57
The Hybrid V-Cycle and the Modular Technical Model	58
Modular Technical Model (MTM) For MIL & SIL simulations	59
Code Generation for SiL performed on RCP platform	61
Code generation for the SiL on the RCP platform exploiting the rti1401 library for the I/O ports.	62
HiL Simulation attempts through the laboratory testbench.....	64
First HiL test, BSoC= $0.5 \cdot \text{BattChargeMax}/100$ and $\Omega_A=100$, Em_Bool = “1” and Mot_bool = “1”	66
Second HiL test, BSoC= $0 \cdot \text{BattChargeMax}/100$ and $\Omega_A=100$ Em_Bool = “1” and Mot_bool = “0”	66
Third HiL test, BSoC= $0 \cdot \text{BattChargeMax}/100$ and $\Omega_A=0$ Em_Bool = “1” and Mot_bool = “1”	67
CONCLUSION	68

Foreword.

The challenge the figure of the engineer must face at the beginning of a new project, regardless of its field of specialization, is to identify, synthesize and analyse the requests provided directly by the customer or the project manager and synthesize them in requirements. This phase is crucial since a superficial or incorrect interpretation of the work objectives can lead to long phases of study and simulations that could give results that are disregarded or even inconsistent with what is required.

A further challenge that the designer, in this case the **Modelling Engineer**, must face concerns the degree of the model's approximation he is building, since if this has too high level of details, it will lead to expected results but will be accompanied by a high amount of data, that may be susceptible to numerous external factors, making in many cases the product unreliable. In addition, such models can lead to long periods of design which in turn require comparison with numerous professional figures and the acquisition of additional knowledge.

These factors take a long time for the achievement of a project, but in field such as the **automotive** and **the industrial sectors**, which are characterized by a continuous evolution of technologies and a pressing demand for innovative and reliable products, it would be uneconomic and therefore unacceptable.

So, the engineer should write a model's **requirement** so that these are compliant with **industrial standards** and rely on suitable **tools** that allow to follow a standard **workflow** that guarantees the successful of a project without losing time in useless design steps.

Objectives.

The aim of this work is to create a software model of a **FIAT Panda** before the year 2003, that is intended to participate in a Rally race in Morocco called **PANDA RAID**. In addition, this car will be converted to the hybrid all-wheel drive (hence the name of the project **Panda 4WD-H**), will be adequately equipped to face the adversities of the route, and accommodate the **electric motor (EM)**. It goes without saying that a careful analysis of the requirements is needed, requiring an adequate contextualization of the project.

The design is carried out in the reality of a typical student team, where there is the possibility to carry out the **internal verification phases** through comparisons between figures from different specialization courses. On the other hand, unlike the typical work of car racing teams, the focus is on the creation of a software model that is used for the development of a control algorithm for the distribution of torque between the two axles, so that the modified vehicle can face the race in the best way.

In detail, the application calculates the **EM torque** appropriate to the situation, taking into account various system parameters such as **battery status**, **wheels speed** and the selected **working mode** of the unit, which can operate both as a **motor** and as a **generator**.

This code, after careful analysis of verification and validation, that is performed by means of tool, which is compliant with the **ISO 26262**, must be loaded on a **Rapid Control Prototype platform (RCP)** to test the behaviour of the software itself before it can be loaded on an actual VMU, that is intended to manage the entire system.

Organization of the work.

A suitable organization of thesis work is needed, so I structured this paper in two main sections:

-Analysis of the competition's rules, description of the environment in which the Rally will take place and of the starting vehicle. These is done in order to obtain an adequate design guideline that can lead to a precise and conforming design of the model.

-Design of a software following the model-based philosophy, starting from the **drafting of the design requirements** that lead to the initialization of the model, which must follow the guideline of the **V-Cycle Automotive**, technical analysis of the theoretical system that can lead to the first phase of **Model In the Loop (MIL)**, verification of each phase of development of this model, final validation and conversion to a suitable modular structure (**Model Technical Model**), which allows to extract the **control subsystem** and convert it in an code runnable on a **Rapid Control Prototyping** board (**dSpace Microautobox 2**). At the end, another simulation (**Software in the Loop, SIL**) is done to see if such code gives acceptable results with respect to the MIL simulation,

Part 1: Analysis of the Panda RAID.

General Overview on the competition

Panda Raid is a long-distance car racing rally in which hundreds of teams compete for about 2000 km where the only 2 kinds of cars are allowed to participate: a FIAT Panda, whose version must be before the 2003, and a SEAT Marbella.



Figure 1 - Panda Raid presentation

For the 2021 edition of Panda Raid there are about 300 teams registered for the competition.

After a sort of parade in **Madrid**, where all the vehicles are shown to the public, they are carried in **Nador** (Morocco), when the race will take place.

The path changes every year, so it is unknown until the competition starts. Nevertheless, some fundamental information on what the participants could face are known in advance.

It's important to underline that there will be two different tracks for 4WD and AWD vehicles, although the difference is about 20 km overall, this means that the 4WD-H system would be strictly necessary only on those parts in which the vehicle will face pool of sand where it is very easy to get stuck.

Nevertheless, the 4WD-H could be very useful for other purposes.

Rules of the Competition.

NO INTERNET: GPS and any other internet-based device are not allowed. So, the pilot and co-pilot can only use a **compass**, the **Road Book** and their personal sense of direction to navigate in the scenario.

RANKING: Panda Raid is not a challenge in terms of timing since the ranking considers the correct navigation and the mean speed in specific sectors.

Therefore, the participants should properly navigate from one checkpoint to the successive. So, it is up to the stewards to check the passage of the vehicle.

Indeed, they are allocated every 50 km and they are in charge of stamping the team's card to prove that they correctly passed the checkpoint.

The Path.

As mentioned above, the road path changes every year, so it is not possible to obtain a precise scheduling of the odds the vehicle has to face.

However, it is possible to predict a general layout looking at the local environment, searching for the information from the previous races and exploiting the descriptions of the paths obtained from former participants.

All the other data are hypothesis, but they could be considered valid for the purpose of testing our model.

The competition is composed of around 6 stages in which the vehicle is severely tested both from a mechanical and thermal point of view.

The path of the previous years should be the following:



Figure 2-Geographical map of the competition from previous years

It is possible to evidence the stages near the cities of:

1. Nador.
2. Beni Tadjide.
3. Errachidia.
4. Merzouga.
5. Tansikht.
6. Marrakesh.

I notice that the longest stage is usually about 500 km, the average every stage is about 300 km and that each one takes about 9 hours to be completed.

Although it is not so precise to obtain the properties of the single path from such a kind of map, I can create a list of the general information, that I can compare with the ones provided by the past participants.

Ground characteristics.

It is necessary to get the general condition of the surfaces of the several road stages in order to model peculiar scenarios in the context of our study.

For what concerns the interactions between the ground and the vehicle, I can focus my analysis on three main ground conditions the vehicle can face in the Panda Raid:

1. Dry tarmac.
2. Soil (Gravel).
3. Sand.

Dry tarmac.

Tarmac, short for tarmacadam, consists of a layer of crushed stone or aggregate coated and mixed with tar. It differs from the asphalt, in which the tar is replaced by bitumen (a by-product of petroleum distillation). The Tarmac differs from asphalt for being slightly harder wearing. It is used for driveway, pavement, and road surfaces like asphalt.



Figure 3 -Dry tarmac road condition

In the first stage, this kind of surface is prevalent (50% of the total length), but it decreases progressively during the race being almost not present at all in the stage from 2 to 5.

Eventually, also the last stage will mainly consist of tarmac.

This part is not problematic from the longitudinal dynamic, vertical dynamic and pitching dynamic points of view, so I could neglect it during my analysis, but it is good to know that on

this surface will be easier for the vehicle to move and so the components will not be stressed as in the other cases.

For this reason, it could be useful to exploit such a path scenario to cool down the engine and the battery, as much as charging it, as I mention in the modelling part of this work.

Dry and Dusty soil.

It represents the most common type of terrain on which the car is intended to race. This is characterized by the almost total absence of tarmac and asphalt and by the presence of gravel, which is a loose aggregation of rock fragments, on the surface.



Figure 4 - Dusty soil road condition

It is important to consider that the environment is very dusty and so it results fundamental to have a proper air filter and a snorkel in order to get always clean air from the outside.

Another inconvenient to keep in mind is the pattern of this terrain, since it could be very hard and bumpy, with rocks of various diameters (5-10cm) and many potholes.

Consequently, it results in a very challenging environment for the suspensions, for the tires and for the rims.

Furthermore, the rocks are very dangerous for the underbody, indeed it is common to have problems with tank, the oil cup and the whole front part (e.g. the engine), so a proper protection of the most fragile components is needed.

Luckily, the 4WD-H is almost not needed in this part and the EM could also work as a Generator in some parts to recharging the battery, selecting the proper velocity profile.

Sand.

The overall rally is composed around 10 % of sand path, that represents, in practice, the most challenging part of the competition since it requires to compute the proper EM's torque traction to overcome this scenario.

Generally, the most demanding parts are not longer than 1 km in a row and the longest part of continuous sand is about 15 km. Moreover, the big dunes are not present on the tracks planned for the FWD vehicles nor for the AWD ones, so I can consider the sand parts almost flat, although the problem concerning the presence of little dunes highly also influence the vertical dynamic and pitching dynamic.



Figure 5 - Sand Road condition

In particular, the kind of sand that is prevalent in the Sahara is referred as “Dune sand”, and it is characterized to be generally very well-sorted, e.g. means that all the sand grains are roughly the same size and there is almost no dust. The sand paths are formed since the grains are blown away and deposited elsewhere, possibly as a loess deposit far away from the source area. There are no gravel or boulders because wind is not capable of carrying such a heavy load.

This presents an unusual challenge to rally cars, as the constant drag of the sand increases the wear and drag on the transmissions. Furthermore, the driving technique that should be applied

on is different from other paths, as the soft sand on the road can be like a snowbank, while the braking ability exceeds that of tarmac with slick tires.

In this condition, the EM must work **always** as a Motor in order to concur with the traction together with the ICE for avoiding over-heating problems. Consequently, it is necessary to performs a suitable study for the sizing of battery keeping in mind the advantages and disadvantages of a huge capability.

Indeed, the extra weight given by the hybrid implementation results in a higher probability of slipping of the tires or of getting the vehicle stuck in the sand. Indeed, the vehicle must be as light as possible to overcome the sand and for this reason the weight must be reduced in all possible ways, removing all the unnecessary things.

Moreover, it is fundamental also to choose the most suitable kind of tire (e.g. tire tread, width, height, rim diameter...) and properly lower their pressure in order to obtain the bigger contact patch possible. The detailed study on the tires characteristics is dealt in the second part of this paper.

Vertical and pitch displacements.

The reason for studying and modelling the vertical dynamics is to find a solution for limiting the force's variations acting on the tires. In this way, it is possible to improve the drivability of the vehicle and modify the vertical response depending on the scenario.

Since the Panda's suspensions are designed to guarantee a compromise between driver's comfort and vehicle's handling, it is necessary to exploit a proper model for simulating the improvements regarding the drivability the mechanical engineers are wanting to apply.

The study of the mathematical description for what concerns the evolution of the suspension system and how it is possible to model it properly is dealt with in more detail in the second section of this paper, where the vertical displacements are referred as vertical values of a **white noise signal**.

So, the biggest problem for what concerns the obstacles is the presence of a variety of rocks, gravel, bumps, hollows, and potholes on the path that influence dramatically the dynamic of the vehicle.

This is particularly a problem while driving on soil due to the presence of a lot of small rocks of a diameter whose range vary from 5 to 10 cm and various potholes.

Each obstacle could cause variations on the vertical, longitudinal and pitch dynamics and so it is necessary to study the proper scenario.

For the example, the present of gravel could cause vertical displacements associable to noise effects that can stress the mechanical couplings of the steering arms.



Figure 6 - A steering arms broken probably by a too huge vertical impact

It goes without saying that a proper modelling of the environmental inputs (or disturbances) must be performed in order to observe their effects on the vehicle so that the mechanical designer can regulate the improvements on the physical vehicle.

Overall, the problems to be worried about are the following:

- The continuous vibrations due to the small rocks and the uneven surface that could lead to loss of traction or resonance effects.
- The presence of rocks which are completely stuck into the ground and, if hit at high speed they can break both the tire and rim.
- Potholes are unavoidable sometimes and they can be very aggressive on the suspensions and therefore on all the vehicles.

Road grade.

The Panda Raid is a characterized by a very large heterogeneous interval of values for what concerns the road slope. Indeed, each vehicle has to face scenarios as **rolling hills, large depressions, dry/wet riverbeds, steep climbs** and **dunes**. Thank to the interviews with the former participants, I can affirm that the slope can vary likely in a range between -30% and 20% during the competition.

Sand dunes are the most demanding parts, with an angle of attack of about 20-25°, while riverbeds have slopes that can reach 50% (usually there are only 1 or 2 of them during the race) but it usually for a very small length and they are easily overcome with the proper inertia.



Figure 7 - A participant facing a relevant slope

In these conditions, the EM must work as Motor most of the time in order to help the traction.

In particular, the control algorithm aims to make the weight of the entire rear axle not a **deadweight** when the car has to face such odds.

However, the driver has not got a battery whose capability is infinite, so he must follow a proper strategy for the activation of the EM.

Other problems.

Overheating: In the desert the temperatures can rise to 50 °C in certain periods of the year and so the vehicle could stand temperatures not usual for a standard urban vehicle during the hottest hours of the day., since the engine might work for a lot of hours under these temperatures.

Furthermore, I consider the presence of the batteries which are highly influenced by the external temperature, so it is mandatory to develop a proper cooling system both for the ICE engine and EM.

Thin sand and dust: The sand of the Sahara Desert, which is the context of the race, is particularly thin and it can compromise the correct functioning of sensitive components, so it is necessary to implement a proper isolation mechanism. In particular, the suspensions could stop working because the presence of sand inside a shock absorber makes not allow the system work properly and obviously also all the electrical components are at risk as much as the engine.

The starting vehicle.

As starting vehicle, I select a FIAT Panda FWD 1100 i.e. cat, whose engine can provide 54 cv (40 kW) and 88 Nm of torque.

The reasons behind this choice are the high power compared to the other versions, the high reliability due the fact that the previous versions had some issues with the piston seals and the fact that it is the most recent Panda allowed to this race.



Figure 8 - A FIAT Panda FWD 1100 young

Moreover, it has a tank of about 45 L which should allow to complete a stage in just one tank.

For detailed information about the technical aspects of this vehicle, I remind to the text *Workshop Manual - General Info & Technical Data [FIAT Technical Manual]*, and the website <http://www.fiatpanda.it/go.com/htmlfire/schedatecnica.htm>.

Vehicle's setup.

The greatest challenge of the vehicle's setup is making the vehicle strong and reliable enough to complete the race. The following improvements are directly a consequence of the problems concerning the context of competition, as I explained above, and the modification I am going to implement on the software model in order to improve their dynamic during the Panda Raid:

- **Suspension system:** it is fundamental to have proper shock absorbers and springs in order to absorb the energy due to the continuous change of surface and to overcome all the possible obstacles (e.g rocks and potholes). In practice, I aim to improve the handling of the vehicle at the cost of reducing the user's comfort. Moreover, the height reached by the vehicle from the ground must be considered, since a greater ground

displacement allow the vehicle to avoid damages on its underbody with respect to a standard configuration.

Such a modification compensates partially the extra weight of the hybrid system could lower even more the distance between the underbody and the ground.

Finally, it could also be useful to use a strut bar on the front to provide extra stiffness between the strut towers.

- **Protections:** Raising the vehicle is not sufficient to completely avoid damages so it is necessary to also add some protections. The most important parts to protect are the tank, the sump and the front part of the vehicle. It is sufficient to use a steel plate of 3mm both for the front, the sump and a similar cover for the tank. The tank could also be upgraded from the classic plastic to a steal one.

Hybridization strategy.

The goal is to create a hybrid system that can deliver enough performance to overcome a classic Panda FWD in such a challenging environment.

Indeed, after several simulation tests, I concluded that a FWD layout might not be sufficient in order to overcome the difficulties that the vehicle will face during the race.

Therefore, the vehicle must be highly modified, starting from the vehicle's alignment until the electrification.

I decided to implement the electric motor on the rear axle, with the purpose of activate the 4WD-H in case of loss of traction and consequent slip of the front tires.

Furthermore, the electric motor will work as a Motor only in some particular situations, because in more than one scenario the FWD is sufficient (e.g. long straight with a non-slippery surface) and so it is wiser to save the batteries and to use them only when it is strictly necessary.

For what concerns the regenerative strategy for the recovery of the electrical energy, several strategies are analysed.

The classic regenerative braking has been rejected due to the lack of downhills and heavy braking sectors.

Hybrid architecture.

The goal is to exploit both the ICE and the EM in order to improve the longitudinal and vertical dynamics behavior of our vehicle.

First of all, it is necessary to design the architecture of the powertrain of the Panda starting from the several hybrid architectures used in the automotive industry.

Considering the difficulty to re-designing the entire drivetrain chain that the P0, P1, P2 and P3 structure could require, the choice fell on the P4 parallel hybrid architecture. This presents some advantages fully compatible with the requirements:

- The great efficiency due the lack of any mechanical coupling between the front and the rear axle.
- The ease to design a rear axle for inserting the EM and the differential starting from the original one of the Panda FWD.
- The possibility to use the vehicle as a BEV.

Overall, the EM can help or substitute the ICE for the traction. For the details about the study for the EM's insertion on the rear axle from a mechanical point of views, I advise to read the thesis *Panda 4WD - Hybrid: design and analysis of a new suspension system for a hybrid vehicle for rally competitions* [Riccardo Breda, 2019]

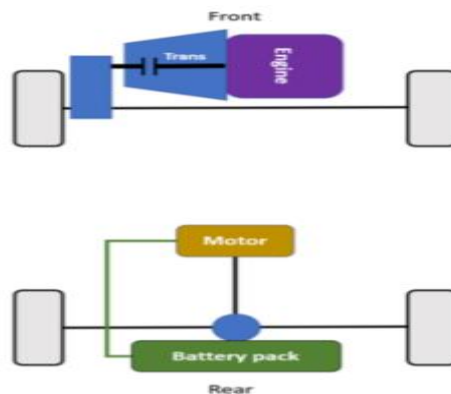


Figure 9 - Parallel P4 hybrid architecture

Part 2: Model designing following the V-Cycle development strategy.

Introduction of the V-cycle software development process

In these last years, the Automotive industry must respond to the increasingly demanding requests by the market of products ever more sophisticated for what concerns the services these can offers.

Indeed, now the vehicles are **computers on wheels** that are able to provide services such as **connectivity (Vehicle to everything, V2X)**, **autonomous driving** and so on. These services need a great amount of data that must be processed by a consistent number of **Electronic Control Unit (ECU)** implemented inside the vehicle. Moreover, these units require a proper communication protocol (**CAN, LIN, ETHERNET**) that makes the messages travel safely and such a velocity that allows the **determinism** of processes.

Consequentially, the complexity of the software running on these ECUs is growing ever more, although their longevity is decreasing.

Volvo year	Lines of code	Number of steps	Time to walk
1997	1 MLOC	Frankfurt - Brussels	70 h
2007	10 MLOC	Stockholm - Jerusalem	700 h
2017	100 MLOC	Circumference of the Earth	7000 h
2027	1000 MLOC	Earth - Moon	8 years

Figure 10 - table representing the increasing of the Miles of Line of Code in the modern vehicle (Vard Antinyan, Volvo Car Group, Sweden)

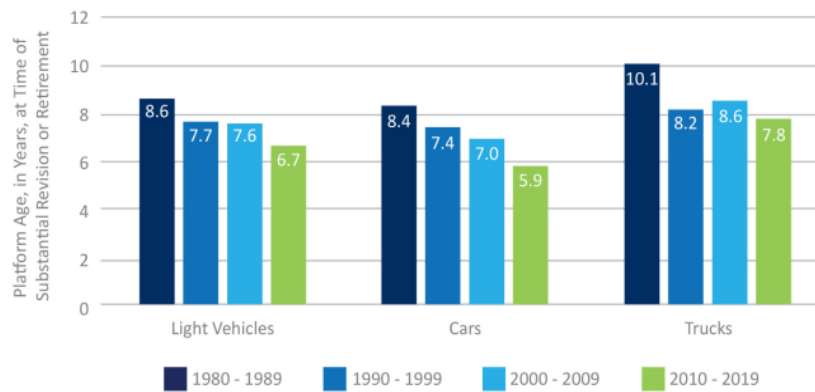


Figure 11 - Average Platform Longevity by decade of introduction and vehicle type (Centre for Automotive Research)

Due to the complexity and size of the control software, it is mandatory to use a development process that should improve the overall quality of the software, increase development efficiency, and eliminate systematic software bugs.

The standard software development process used in the automotive industry is the **V-Cycle**. The V model splits the software development process into two main phases.

- The left side of the V is the part of requirement analysis, function/software design and change management.
- The right side of the V concentrates the main verification and validation activities.

For further details about the V-Cycle, visit the website <https://x-engineer.org/graduate-engineering/modeling-simulation/model-based-design/essential-aspects-of-the-v-cycle-software-development-process/> and read the thesis *THERMAL PREDICTIVE CONTROL FOR HYBRID AND ELECTRIC POWERTRAIN* [Coletta, 2020].

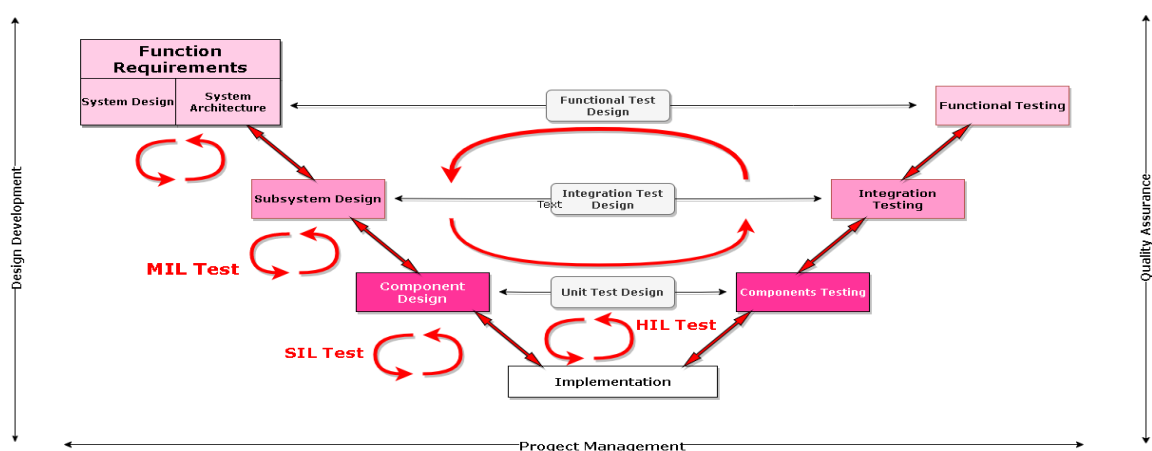


Figure 12 - An example of V-Cycle for automotive applications

The overall work of this part is structured so that I apply the V-Cycle strategy for each module of my software model. Indeed, a great property of such process consists in its **Agility** and that allows **iterative loops**, so that I can focus on the study of **single module** of my software program and verify that it passes each development phases, that are stated by the V-Cycle itself. As soon as it results verified and validated, I proceed to the successive module and finally integrate all the component obtaining my final product.

Moreover, another peculiarity of the V-Cycle is the possibly to **parallelize** the steps processes, allowing to save time keeping the **reliability** of the project.

Therefore, the project is based on the developing of each phase of the descending branch for each subsystem that constitutes the overall software:

- **Function Requirements:** it defines what is expected from the module in terms of functionality. This phase is crucial since a poor **Requirements Management** is the main cause of the embedded project fails. Usually, the requirements are come in a form of a **document (.doc, .xls)** or managed inside standard program like **DOORS, Polarion**. Furthermore, it is necessary to guarantee **traceability** between requirements and system design. This is the reason why I use the **Simulink Requirements** and **Simulink Test** tools.
- **Subsystem Design:** it consists in using a model-based development environment to design and test the required functionality. Such environment must be **independent** from the technology of the Hardware in which is intended to install it, so I use **Matlab/Simulink** program to implement the control functions. Therefore, this is developed as a **block diagram** which has the capability to be simulated and highlight potential design error. After the function design is complete, I run a series of tests called **Model in the Loop (MIL)** tests since they are using models (block diagrams) to simulate the functionality.
- **Software Development:** It consists of translating the software product in **C language** using a proper tool of **automatic coding**. Indeed, almost all the **ECUs** within a modern vehicle are programmed in **C**. At this stage there are more details regarding the **functionalities** since the software tool needs to know the **specific hardware** in which the **control algorithm** is intended to run and what **data types** needed for each of the software variable. Therefore, a result of this operation is an optimized C code, in terms of memory resources and CPU loading. Modern software development processes are

using **model-based design tools** which automatic code generation capabilities (**Embedded Coder** from **MathWorks** in my specific case). This means that I can generate C code automatically from the block diagram. I can also perform **Software in the Loop (SIL) test** with automatic code generation. This technique allows to compile and run the C code of the control algorithm on the **development workstation (DWS)** supervising the behaviour of the plant simulated on the **RCP unit**. By doing **SIL** tests, I can check if the software is providing the required functionality (compared to the model).

Although the phases of **Processor in the Loop (PIL)** and **Hardware in the Loop (HIL)** should be included in the descending branch, I decide not to perform such simulations since they require to be implemented inside programs that run in specific purpose hardware, which is not available in the lab, and that require very expensive software license.

Vehicle model and degree of freedom of approximation

From the analysis of the competition and of the kind of odds the vehicle will address, I decide to model the Panda as a **vehicle with 3 degrees of freedom (Vehicle 3DOF)**.

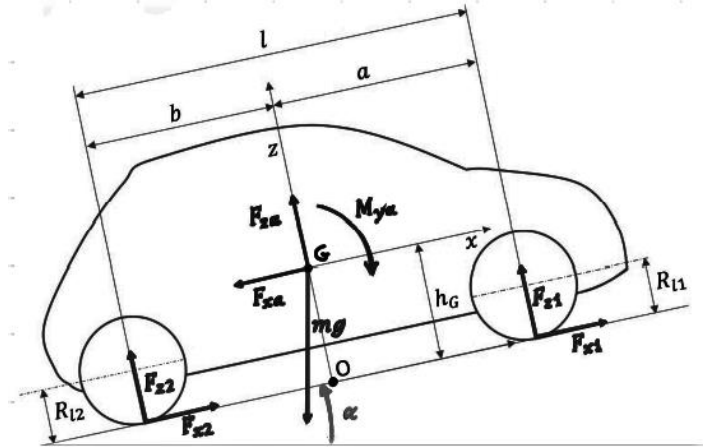


Figure 13 – Basic scheme of a 3DOF vehicle

This choice is justified since the vehicle is not intended to perform high speed corners, but it might only go straight. So, it is possible to describes properly the behaviour of the system focusing only on its **Longitudinal dynamic**, **Vertical dynamic** and **Pitch dynamic**, neglecting the lateral accelerations.

Also, I assume the aerodynamic reference frame is parallel to ground and the overall aerodynamic force is centred in the **centre of gravity (G)**

I can consider the Panda as a system constituted by a **rigid chassis** attached to two **axes**. The latter can be roughly described as system of spring, dampers, tires, and it is responsible for the interaction between the vehicle and the ground.

Such an interaction is possible modelling a **suspension system**, which converts the vertical acceleration due to the **load transfer** and the presence of vertical displacements. Then, it is necessary to build a proper **tire system** that considers the exchange of longitudinal forces between the vehicle and the ground. In particular, the latter responds in different ways according to the nature of the path, as I mentioned above in the first section.

Overall, it is necessary to make the part regarding the chassis in contact with the external inputs and disturbances that can affect these dynamics.

Looking at the Figure 13, the **Equilibrium Equations** that describe properly the dynamics of the Vehicle 3DOF are the following.

$$F_{x1} + F_{x2} - F_{xa} - mgsin\alpha = m\dot{v}$$

$$F_{z1} + F_{z2} + F_{za} - mgcos\alpha = 0$$

$$aF_{z1} - bF_{z2} + h_G * mgsin\alpha + F_{xa} * h_G - M_{ya} = -mh_G\dot{v}$$

Vehicle model in Simulink

Inside the Simulink environment, I can model the vehicle dynamics in two ways:

- Constructing a custom model writing a proper MATLAB script that runs on the **MATLAB Function** Simulink block. In this, I can run a script which implements the overall dynamic of my chassis with the desired degree of approximation.

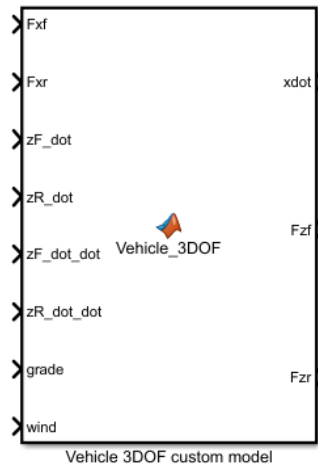


Figure 14 - MATLAB Function Simulink block

A drawback of this solution is that is very hard to calibrate the block when it is necessary to modify the parameters regarding the vehicle and/or modify/add the equations that regulate the behaviour of the model.

- Exploiting Simulink Blocks implement these dynamics. In particular, the **Vehicle Body 3DOF Longitudinal** allows to implement a three degrees-of-freedom (3DOF) rigid vehicle body model with configurable axle stiffness to calculate longitudinal, vertical,

and pitch motion. The block accounts for body mass, aerodynamic drag, road incline, and weight distribution between the axles due to acceleration and the road profile.

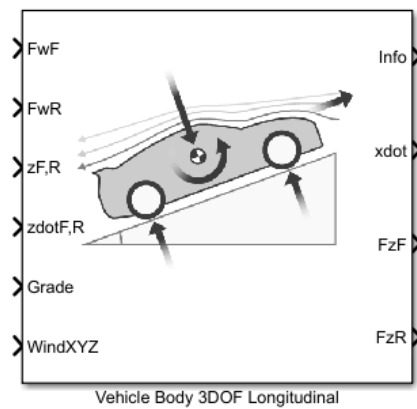


Figure 15 - Vehicle Body 3DOF Longitudinal Simulink block

I decide to use the latter solution since I can easily modify the car parameters exploiting the block's window and because it is supported by a good documentation. Moreover, it is validated by several experiments performed by the MATWORKS community.

The **Vehicle Body 3DOF Longitudinal Simulink** block allows also to specify the type of axle attachment to the vehicle. In my case, I decide to use the option “**Axle displacement**”, so that the axles have input-provided vertical displacement and velocity with respect to the road grade.

This option requires tabular stiffness and damping parameters to model the suspension forces acting between the vehicle body and axle.

The block uses **rigid-body vehicle motion, suspension system forces, and wind and drag forces** to calculate the normal forces on the front and rear axles. It also resolves the force components and moments on the rigid vehicle body frame:

$$\begin{aligned}
F_x &= F_{wF} + F_{wR} - F_{d,x} - F_{s_{\alpha}F} - F_{s_{\alpha}R} + F_{\delta,x} \\
F_z &= F_{d,z} - F_{s_{\alpha}F} - F_{s_{\alpha}R} + F_{\delta,z} \\
M_y &= aF_{s_{\alpha}F} - bF_{s_{\alpha}R} + h(F_{wF} + F_{wR} + F_{s_{\alpha}F} + F_{s_{\alpha}R}) - M_{d,y}
\end{aligned}$$

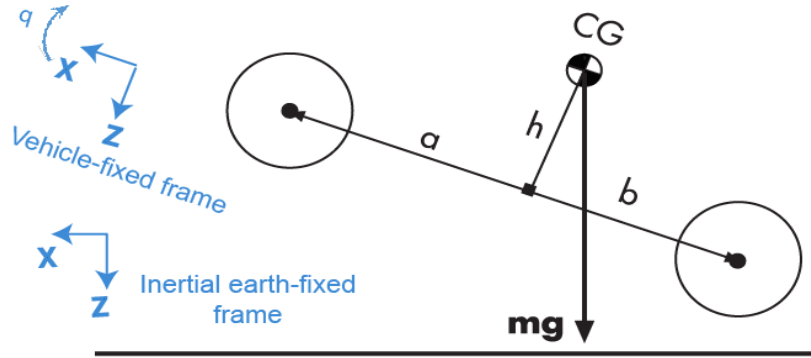


Figure 16 - References frame the block refers and equilibrium equation used

The vehicle axles are parallel and form a plane. The longitudinal direction lies in this plane and is perpendicular to the axles. If the vehicle is traveling on an inclined slope, the normal direction is not parallel to gravity but is always perpendicular to the axle-longitudinal plane.

The block uses the net effect of all the forces and torques acting on it to determine the vehicle motion. The longitudinal tire forces push the vehicle forward or backward. The weight of the vehicle acts through its centre of gravity (CG). Depending on the inclined angle, the weight pulls the vehicle to the ground and either forward or backward. Whether the vehicle travels forward or backward, aerodynamic drag slows it down. For simplicity, the drag is assumed to act through the CG.

Model of suspension system

In order to determine the vehicle motion, the **Vehicle Body 3DOF Longitudinal** uses external longitudinal force, the pitch moment and external vertical forces. The latter are outputs of a system which interacts with the chassis, i.e. **the suspension subsystem**, whose simplest representation is the **quarter car model**.

It is a modelling way most widely used in vehicle-related studies.

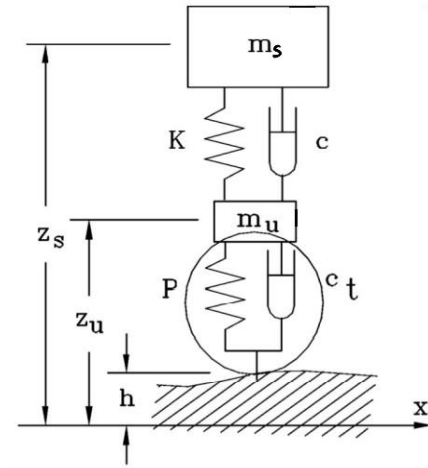


Figure 17 - scheme of a quarter car model with focus on the forces acting on the system

It consists of the most essential implementation that describes the movement of a vehicle in the vertical translational direction.

It follows the **lumped-mass** approach, that implies there are only two inertial components which are the sprung mass and the unsprung mass, related to two DOFs.

For a quarter car model, the sprung mass represents loosely the quarter mass of the vehicle body which includes all parts supported by the suspension system. Meanwhile, the unsprung mass comprises the masses of all parts of a single wheel station that are acted by the suspension. Due to the two DOFs, a typical lumped-mass quarter vehicle model is basically capable of representing **the body bounce and wheel hop modes of movement** of a vehicle.

The dynamic of a quarter car model is represented by the following equations, that are referred to Figure 17:

$$\begin{aligned}
& \begin{bmatrix} m_s & 0 \\ 0 & m_u \end{bmatrix} \begin{Bmatrix} \ddot{z}_s \\ \ddot{z}_u \end{Bmatrix} + \begin{bmatrix} c & -c \\ -c & c + c_t \end{bmatrix} \begin{Bmatrix} \dot{z}_s \\ \dot{z}_u \end{Bmatrix} + \\
& + \begin{bmatrix} K & -K \\ -K & K + P \end{bmatrix} \begin{Bmatrix} z_s \\ z_u \end{Bmatrix} = \begin{Bmatrix} 0 \\ c_t \dot{h} + Ph \end{Bmatrix}, \\
& \begin{cases} \frac{|z_{s0}|}{|h_0|} = P \sqrt{\frac{K^2 + c^2 \omega^2}{f^2(\omega) + c^2 \omega^2 g^2(\omega)}} \\ \frac{|z_{u0}|}{|h_0|} = P \sqrt{\frac{(K - m \omega^2)^2 + c^2 \omega^2}{f^2(\omega) + c^2 \omega^2 g^2(\omega)}} \end{cases}, \\
& \begin{cases} f(\omega) = m_s m_u \omega^4 - [P m_s + K(m_s + m_u)] \omega^2 + K P \\ g(\omega) = (m_s + m_u) \omega^2 - P \end{cases}. \\
& \frac{|F_{z_0}|}{|h_0|} = P \omega^2 \sqrt{\frac{[K(m_s + m_u) - m_s m_u \omega^2]^2 + c^2 (m_s + m_u) \omega^2}{f^2(\omega) + c^2 \omega^2 g^2(\omega)}}.
\end{aligned}$$

In which m_s , m_u are the sprung and unsprung mass values, z_s , z_u are **the vertical sprung and unsprung mass displacements** that represents the DOFs, their derivatives are the corresponding velocities and accelerations, z_g is the **vertical road input displacement**, while k , c , k_t are the dynamical representations of the system, namely **suspension stiffness**, **suspension damping** and **vertical tire stiffness**.

It goes without saying that this representation is very simplified or linearized in terms of dynamical component. Indeed, the suspension stiffness is usually represented by linear or constant spring rate, suspension damping is usually treated as being linearly proportional to velocity, while the tire has linear vertical stiffness, but negligible damping and its representation is assumed to have fixed point contact.

Such a model can be implemented inside the Simulink environment exploiting the interaction between the blocks “**Vehicle Body 3DOF Longitudinal**” and “**Longitudinal Wheel – Disc Brake**”. Indeed, they both allow to implement the vertical motion by means of **Look-up tables** that resume the characteristics of the dumping coefficient and the stiffness.

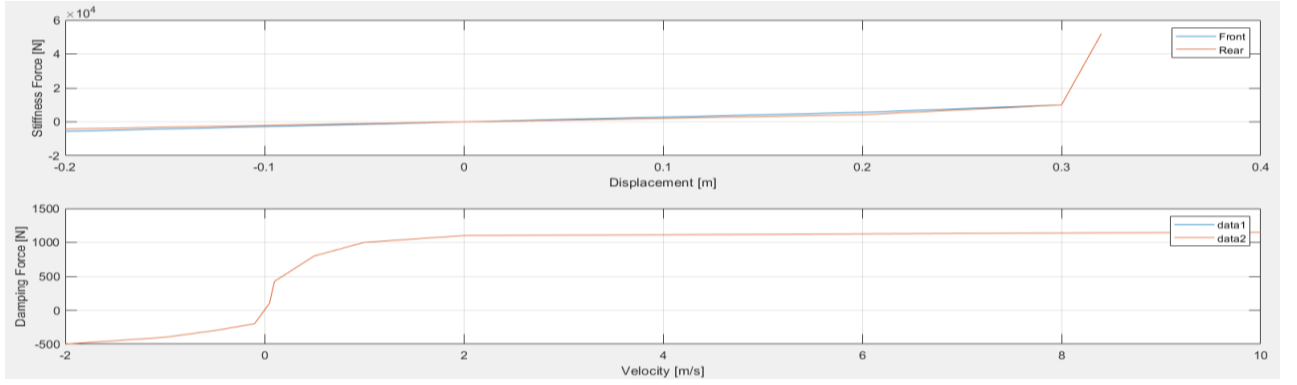


Figure 18 - Look-tables concerning the stiffness and the damping of the suspension attached to the sprung mass

The choice of the **breakpoints** obviously rules the behaviour of the overall system. Usually, the trend of a suspension architecture is a trade-off between **comfort** for the driver and **driveability** of the vehicle, that is intended to work inside an urban environment, where **the Drive-Cycle** does not predict high velocities and the odds are easier to face than the ones of a rally context.

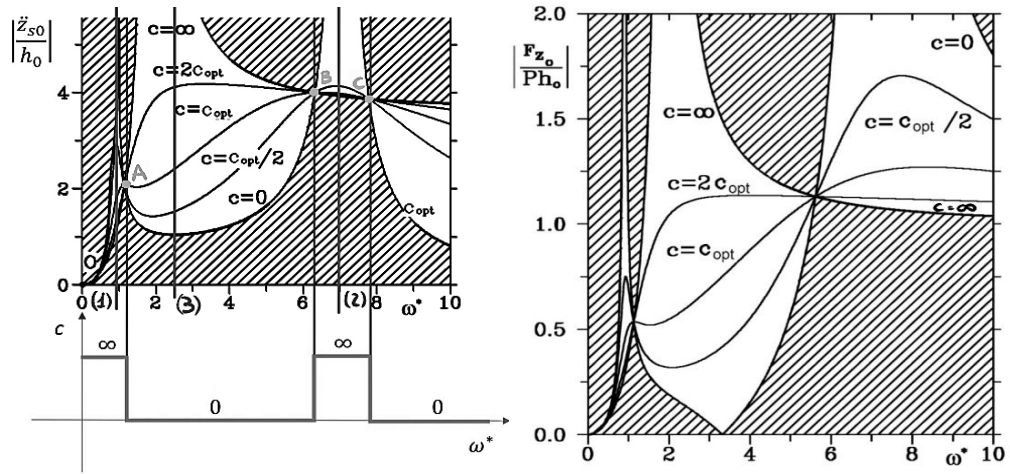


Figure 19 – Frequency responses related to both the sprung and the unsprung masses for a system with $P = 4K$ and $m_s = 10m_u$. The plots include curves obtained with different values of damping c . T

$$\text{nondimensional frequency in x-axis is } \omega^* = \omega \sqrt{\frac{m}{K}}$$

In particular, the value of the **stiffness** k can be assumed constant or linear for all the ranges of operation frequencies, while the choice of the **optimum damping** c depends on the frequency range, as it is possible to see on Figure 19 and on the level of the noise attenuation the engineer

wants to implement (Figure 20). So, there must be the possibility to modify the **Look-up Table** to tune the behaviour of the suspension system in according to the **Drive-Cycle**.

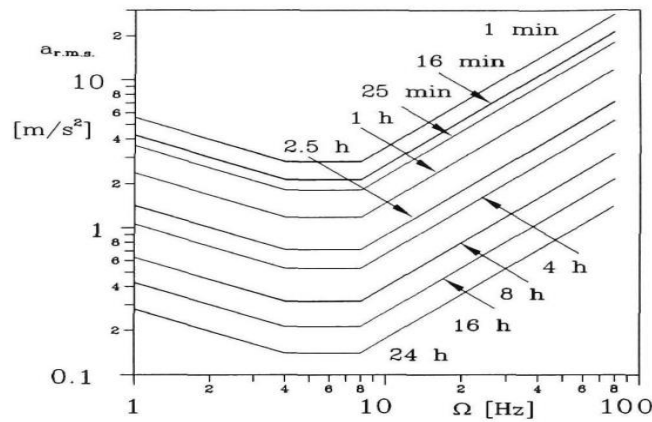


Figure 20 - rms value of the vertical acceleration causing reduced physical efficiency to a sitting subject as a function of the frequency. The curves for different exposure times have been reported (ISO 2631 standard).

Modelling of the vertical displacement signal, slope signal ground characteristic signal

A crucial requirement regarding the project on Panda 4WD-H is the design of a system that makes interact the dynamic of the vehicle with the characteristics of the road path.

This software must give:

- A signal labelled as “**u_Signal**”, that describes the terrain conditions of the path the vehicle is travelling on (dry, wet, sand, etc..).
- A signal labelled as “**Slope_Signal**”, that describes the grade of the road the vehicle is travelling on (in Grade: °)
- A signal labelled as “**Vertical_Displ_Signal**” that describes the vertical displacements (holes, bumps, stones, gravel, etc...).
- A signal that describes the slope variation of the path (hills, steep descents, etc...).

A further requirement concerning all these signals is that the software must make them strictly dependent on the velocity of the vehicle and so with the distance covered. This way, it is possible to make the simulation more realistic since the correct timing of events is ensured.

It is necessary to measure the **feedback velocity** of the Panda, to convert it into the **distance travelled**, to discretise this signal and to make it coupled with the value of the **vertical**

displacement and the **ground properties**. These signals are structured as some $[1, X]$ **vectors of integers** whose values are associate with some finite vertical displacements, expressed in cm, and ground characteristics.

In particular, each value of “**u_Signal**” must be associate a-priori with a certain road condition so that the model can recognize the path on the vehicle is travelling on. I use the following associations:

- $u = 0$; **Dry Tarmac** conditions.
- $u = 1$; **Wet** conditions.
- $u = 2$; **Soil** conditions.
- $u = 3$; **Sand** conditions

Model of tire

It is necessary to design a proper model that resumes the **tire dynamics**, strongly influenced by the **road conditions**, and how such forces influence the overall performances of my system.

Before doing a list of requirements regarding my subsystem, I need a **mathematical description** of the tire’s dynamics. Since it is necessary to model how the system converts the Torque coming from the propulsive unites (EM, ICE) into a tractive longitudinal Force F_x that makes the Panda accelerate.

My design is based on a 3DOF Vehicle model, so I am going to neglect any **steering actions**, **side slips**, **camber dynamics** and **cornering dynamics**.

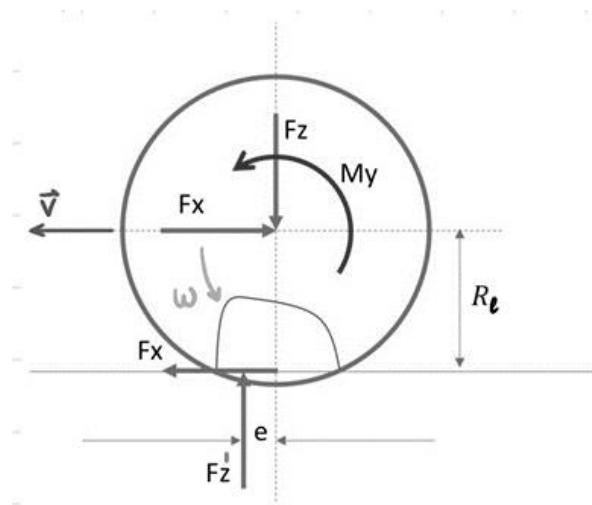


Figure 19 - Reference frame for the study of tire forces.

The tire forces under study are the following:

- F_z : **Radial (load) forces** along Z axis.
- F'_z : Vertical force, that is **net force of the pressure distribution**. It composes the **Rolling resistance** with F_z .
- M_y : torque on the Y axis. It is referred positive when it makes the car accelerate, while it is intended negative when makes the car brake.
- F_x : tractive force along the X axis.

Tire stiffness and damping

They are the main dynamic parameters for what concerns the behaviour for vertical deformations of the tires due to impulsive vertical force. These parameters are obtained from a lumped-mass modelling of the tire.

Usually, the stiffness results much greater with the analogue quantity for suspension systems attached to the sprung mass (chassis):

$$K_r = \frac{F_z}{\Delta z} ; \text{ where } \Delta z \text{ represents the } \mathbf{radial \ deformation} \text{ due to } F_z$$

For what concerns the damping coefficient, it is usually considered negligible.

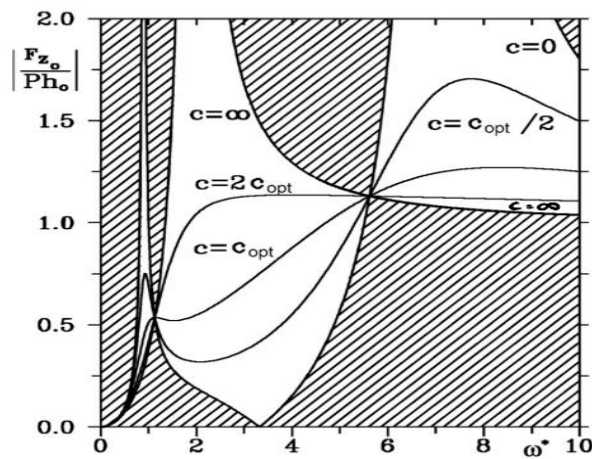


Figure 20 - Harmonic response of the vertical deformation of a tire

Longitudinal Slip

It is necessary to fix a relation that links the resultant longitudinal force F_x with the radial velocity of the tire ω . In order to do that, I start from the condition of **pure rolling** motion, where the tire is under the action of a constant vertical force F_z and F_x is null. In these conditions, the wheel rotates at velocity ω_0 with the centre of rotation is R_0 from the wheel axis.

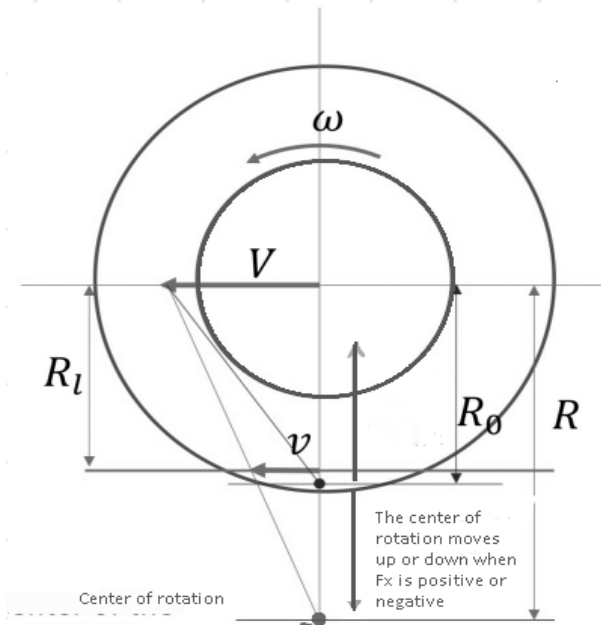


Figure 21 - wheel centre of rotation depending on the applied F_x

The velocity V of the body results:

$$V = \omega_0 R_0$$

This indicates that the rotation centre is located at a distance R_0 from the wheel axis.

Now, applying a generic F_x to the wheel, its velocity V changes. I define the **longitudinal slip** as:

$$\sigma = \frac{\omega - \omega_0}{\omega_0}$$

Where ω is a generic rotational speed caused by F_x .

σ indicates how far the wheel is from the condition of **pure rolling** in percentage and it

has not bounds for positive values (**infinite acceleration**) while it saturates at -1 (**blocking**) for braking actions.

N.B. The stiffness of a tire is not infinite rigid. So, fixed the kind of tire, the contact point with the ground changes according to the several factors like internal pressure, vertical load force, temperature, road condition, transmitted torque, etc...

N.B. the value of the forces acting on the wheel are obtained by means of empirical test, such as FLAT BELT TESTING MACHINE and/or DRUM TRACK in which suitable **dynamometers** measure those values.

- Imposed: V, F_z ;
- Parameters: ω, α, γ ;
- Measured: F_x, F_y, M_x, M_y, M_z

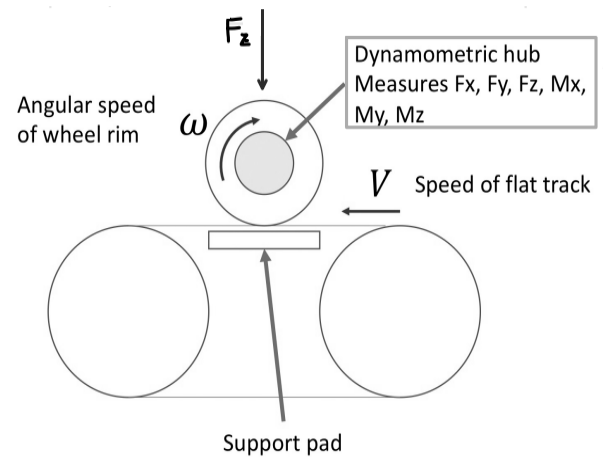


Figure 22 - Flat Belt testing machine scheme

In a general case, I consider $V = \omega R$ and so I can state:

$$\sigma = \frac{\omega - \omega_0}{\omega_0} = \frac{\frac{V}{R} - \frac{V}{R_0}}{\frac{V}{R_0}} = \frac{R_0 - R}{R}$$

Referring to Figure 30 and considering $R = \frac{V}{\omega}$, I can see that changing the value of the angular speed makes the position of the rotational centre change. More precisely, it moves down in case of braking and up in traction.

R_l is the **loaded radius** and it is the distance between the wheel axis and the ground.

Due to the deformability of the tire, this value can change. If the wheel were infinite rigid, v (the speed of the ground) would have been null, but it approximately represents the **speed of the contact path** relative to the ground:

$$v = \frac{V}{R}(R - R_l) \approx \frac{V}{R}(R - R_0) = -\sigma V \quad \textbf{APPARENT SLIP}$$

Introducing the **Friction Coefficient** as:

$$\mu_x = \frac{F_x}{F_z}$$

I can plot a diagram between σ & μ_x .

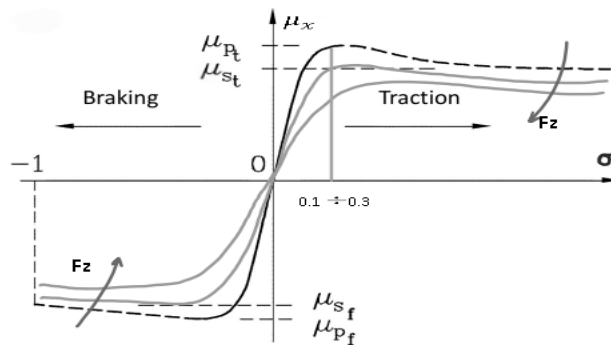


Figure 23 - Plot diagram between longitudinal slip and friction coefficient for different values of F_z

Increasing the vertical load F_z , the force that is produced by the tire F_x does not increase as much as F_z itself and it starts to saturate, creating a saturation zone: in this zone, I can say:

$$m\dot{V} = F_x \quad (\text{m indicates the weight of the vehicle})$$

$$m\dot{V} = \sum F_{x_i} = \sum F_{z_i} \mu_{x_i} \quad (\mu_x, F_z \text{ are constant})$$

$$V = \frac{F_x}{m}$$

So, less weight means more friction coefficient and so better performances from the tire point of view.

Rolling Resistance and Shear Force distribution

In case of rolling, the tire interacts with the ground through some forces.

First, there is a certain **pressure distribution** σ_z from the ground on the tire contact surface. In case of rolling, it is not symmetrical, and the peak results in front of the origin “0” of the tire reference frame.

Considering its asymmetry, the net force F'_z , that the ground applies on the wheel, is not on the same axis of the F_z and so a braking torque is created, referred as **rolling resistance**.

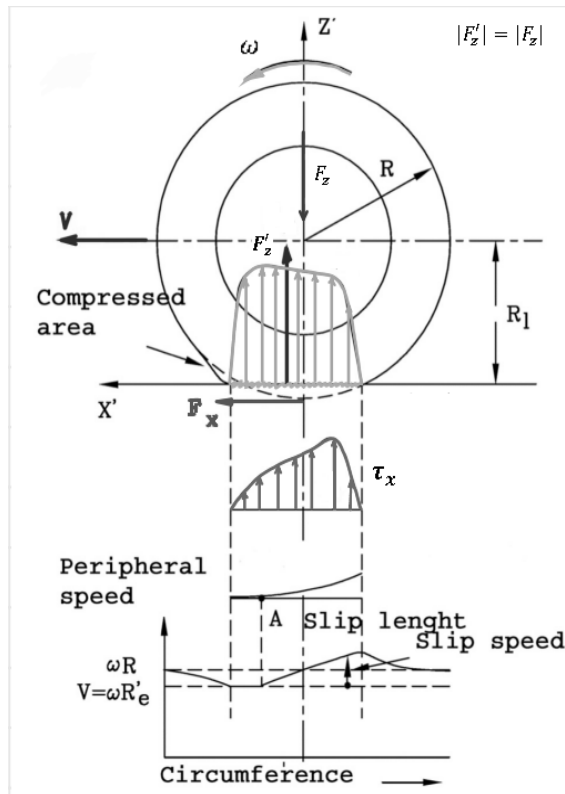


Figure 24 - pressure and shear distribution on the tire 'surface

For what concerns the Shear force τ_x , it starts from zero, it increases until a maximum and decreases when the brick of the tire's tread exits from the contact surface. When the τ_x exceeds the friction limit, the block starts to slip because it can not be attached to the road any longer.

In traction τ_x is positive, in braking is negative and it has a triangular distribution where the peak is at the rear of the origin.

As shown in Figure 33, in the initial part of the contact patch both tangential and normal forces are minimum and so are σ_z and τ_x . Then, the brick of the tire's tread experiences the maximum normal force and tangential force is very low, but it is increasing.

In the end, the last contact point of the tire experiences the maximum shear stress but the lowest normal force. So, there is slippage between the brick and the ground. This values both for braking and for acceleration.

Effect of Road conditions

The force from the tires depends on road features. More precisely, for the same tire I can have different friction coefficient depending on the type of surface. The behaviour (slope) is similar, but there is a significant difference between the curves obtained by different experiments in conditions not exactly comparable. Indeed, the saturation changes and so do the maximum braking/acceleration.

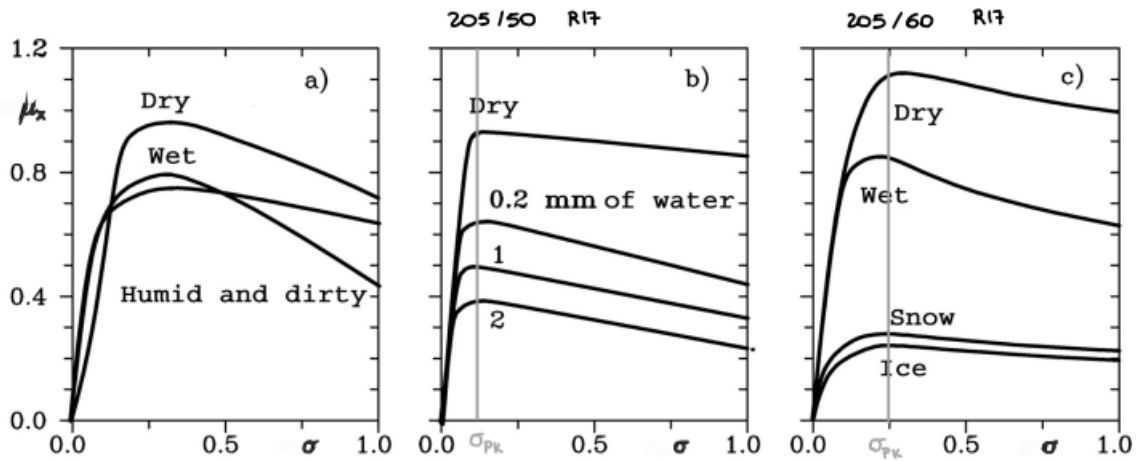


Figure 25 - wheel slip for different terrain conditions

Focus on the effect of terrain condition: speed variation on wet surface

On a dry surface, the speed does not significantly impact on the tire's behaviour, meanwhile, on a wet surface (constant thickness of water) the more the speed, the less the friction coefficient's peak.

Tire model for longitudinal behaviour

At the end, I can introduce a mathematical description that resume the tire's behaviour.

The most used in modelling for vehicle 3DOF applications is **The Pacejka Model**:

- $\mu_x = A(1 - e^{B\sigma}) + C\sigma^2 - D\sigma$
- $F_x = D * \sin \left\{ C * \operatorname{atan} \left[B(1 - E)(\sigma + S_h) + E * \operatorname{atan} [B(\sigma + S_{h_x})] \right] \right\} + S_{v_x};$

where S_{h_x} and S_{v_x} represent offsets to the slip and longitudinal force, while A, B, C, D and E represent empirical data and they are characteristics for each road conditions

It is an empirical model, i.e. it does not follow a physical model but it fits very properly the experimental points found with tire testing machine.

Functional requirements for the tire model

At this point, it is necessary to fix the functional requirements the wheel's model must respect:

- Compute the longitudinal force acting on the axle from the torque.
- Compute the rolling resistance.
- Compute the vertical force, which are caused by ground vertical stresses, acting on the vehicle. The wheel works in model the quarter car model as unsprung mass.
- Compute the longitudinal slip given a vertical load, axles torque and vehicle velocity.
- Calibrate the relation between the slip, longitudinal force, and vehicle velocity in response of a 2D discrete signal which communicates to the model on which condition the vehicle is travelling.

Considering these requirements, I implement my model exploiting the “**Longitudinal Wheel**” Simulink block and “**State-flow**” state machine tool.

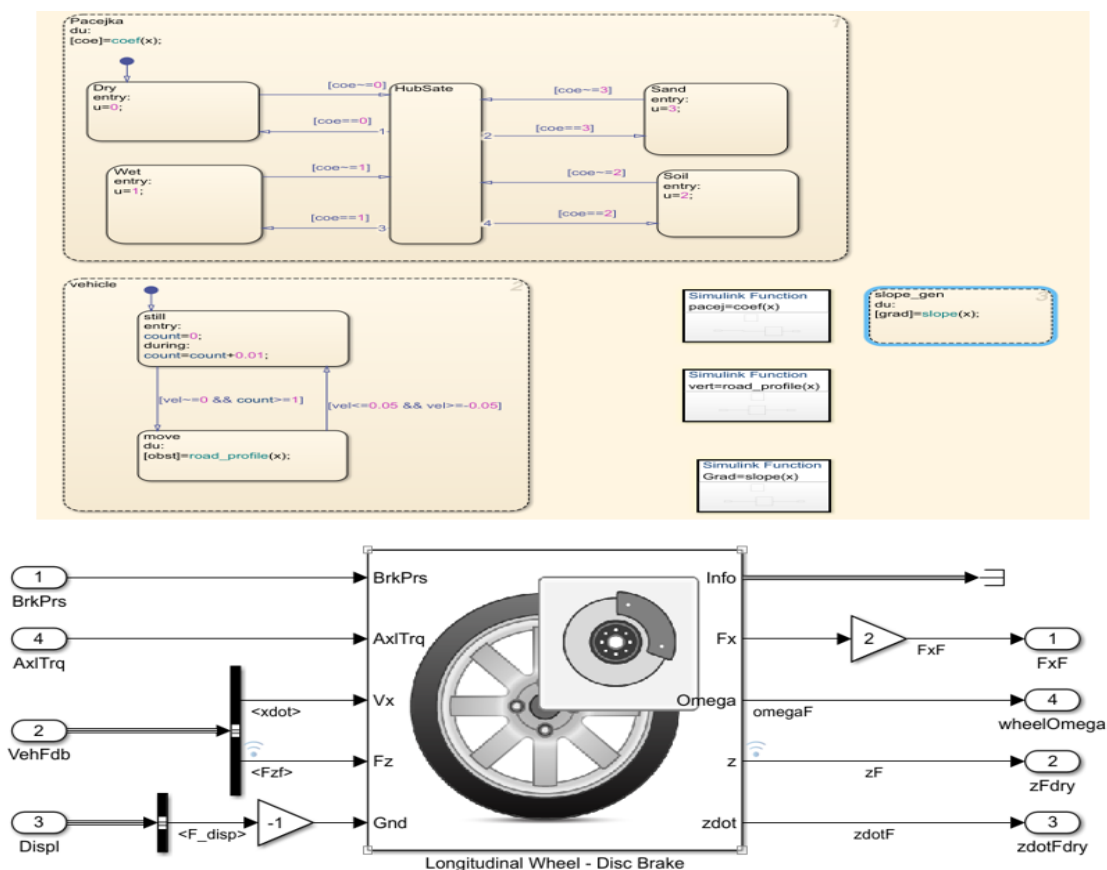


Figure 26 - State machine for the calculation of the road conditions and the Longitudinal Wheel Simulink block

The State-flow in particular computes the state, the slope, and the vertical obstacles of the ground from the vehicle velocity as requested from the requirements.

The “**Longitudinal Wheel**” allows to map the behaviour of the longitudinal slip by means of **Look-up tables** or exploiting the **Pacejka’s constant**. Since, such constants can be got only through several empiric tests and so it is very hard to obtain them for each specific terrain condition, I decide to exploit the Look-table modelling option.

So, I fill these tables of data points, through empirical simulation and calibration until I obtain the correct behaviour for each road condition:

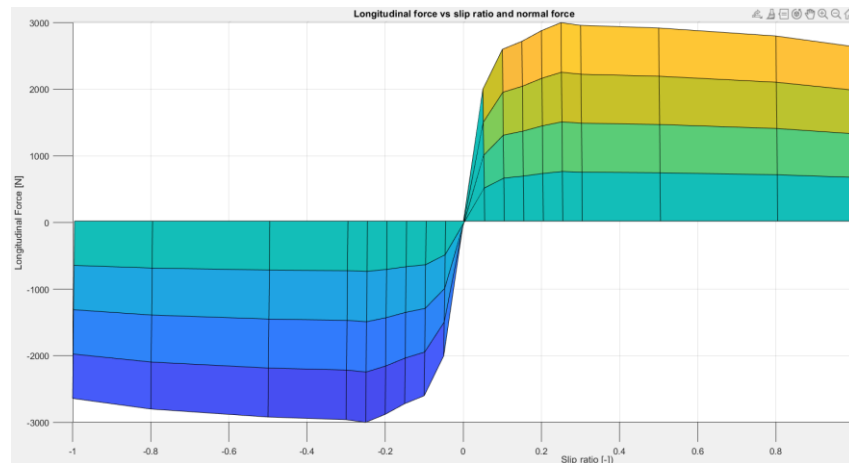


Figure 27 - Longitudinal Slip diagram for a tire traveling on wet tarmac

Model in the Loop simulation for the 3DOF degree of freedom

In this section several simulations are performed to verify if the software relative to the 3DOF system's dynamic respect the fixed requirements.

This is possible through the “**Data inspector**” Simulink tool, that allows me to follow the evolution of the signals of interest in run time.

Each simulation requires the suitable lay out of signals. This way, I can verify if the correct timing of dependent signals is respected.

3 Degree of freedom dynamics, test on a flat path in Dry condition

In this test, I aim to verify if the software effectively simulates well a vehicle's plant with 3 Degree of Freedom. In order to focus on this issue, I create a test on which the Panda has to follow a classic velocity **Drive-Cycle (ECE R15, four cycle)**. For the entire journey, **the slope is null** while the terrain condition is equivalent to a **dry tarmac** road path. Then, I insert some vertical displacements in crucial parts like acceleration phases and paths at constant velocity to see the overall behaviour of the system.

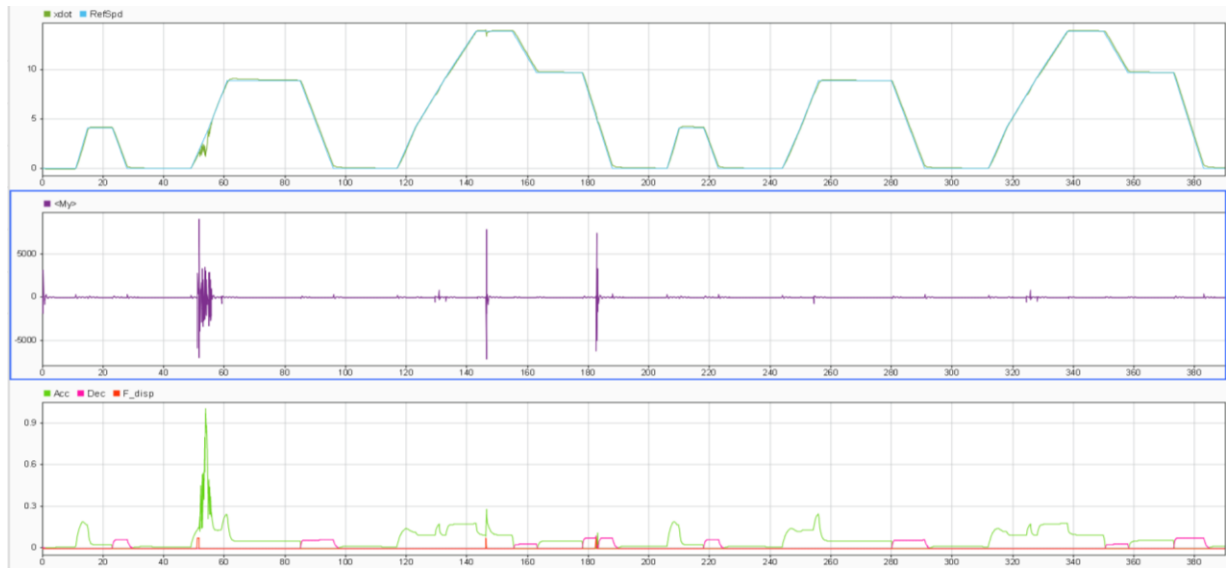


Figure 28 - Simulation number 1 for the verification of the 3 degree of freedom dynamic

Observing the vertical forces acting on the vehicle ($F_z f$ and $F_z R$), I notice that the systems responds correctly both to the vertical solicitations and to the acceleration actions. Moreover, the timing of the odds, which must be dependent on the vehicle's velocity, results verified.

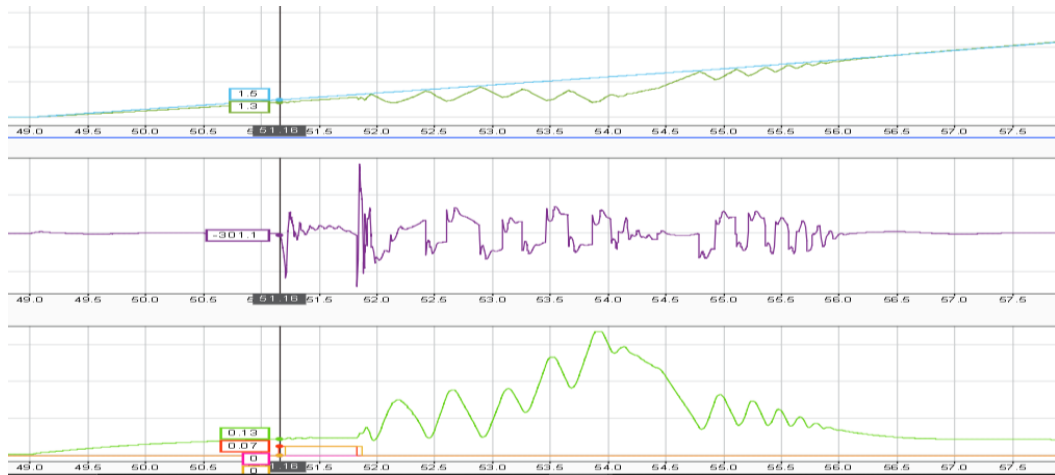


Figure 29 - Focus on test number 1 for bouncing phenomena

In figure 29, I notice that at the time 51sec the vehicle is subjected to a huge variation of the pitch moment due to a vertical obstacle of 7cm. This odd is faced at very low velocity and during a slight acceleration phase.

This could be caused by a choice of the suspensions' damping factor c that is not suitable for solicitations at low frequencies. So, it is necessary to calibrate this data for vertical low frequency applications. It is a task for the mechanical group.

3 Degree of freedom dynamics, test in Second Scenario for different road condition

This test, I aim to verify if the system responds to the variation of terrain conditions, that is represented by the “u_Signal”. It must consider the variation of the pitch moment acting on the vehicle Body Inertia and the difference of torque output compared to a scenario in which the simulation is performed entirely on a dry tarmac path. So, I simulate the system using a custom drive cycle, which is labelled as “Second Scenario”, in both situations.

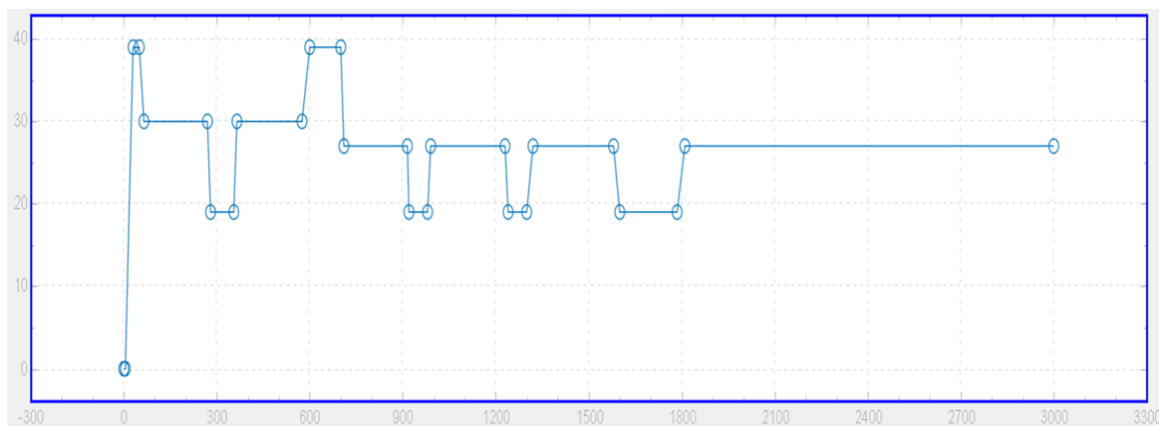


Figure 30 - Drive Cycle of the second scenario

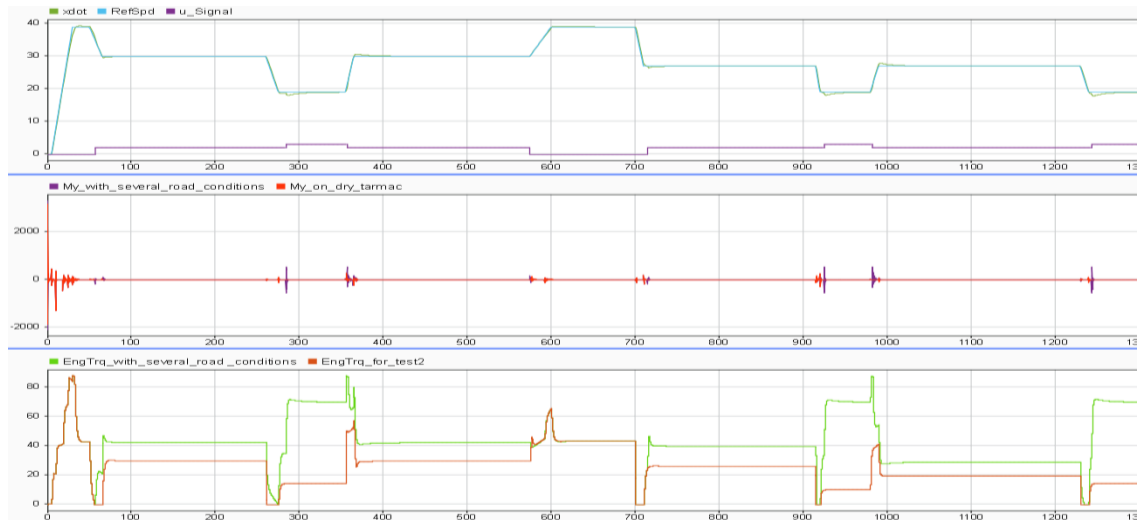


Figure 31 - Simulation number 2 for the verification of the 3 degree of freedom dynamic

Looking at the engine torque in both the simulations, I notice that its output is effectively greater on “tougher” terrain conditions than on dry tarmac path only. Consequently, also the pitch moment is greater during the acceleration phases.

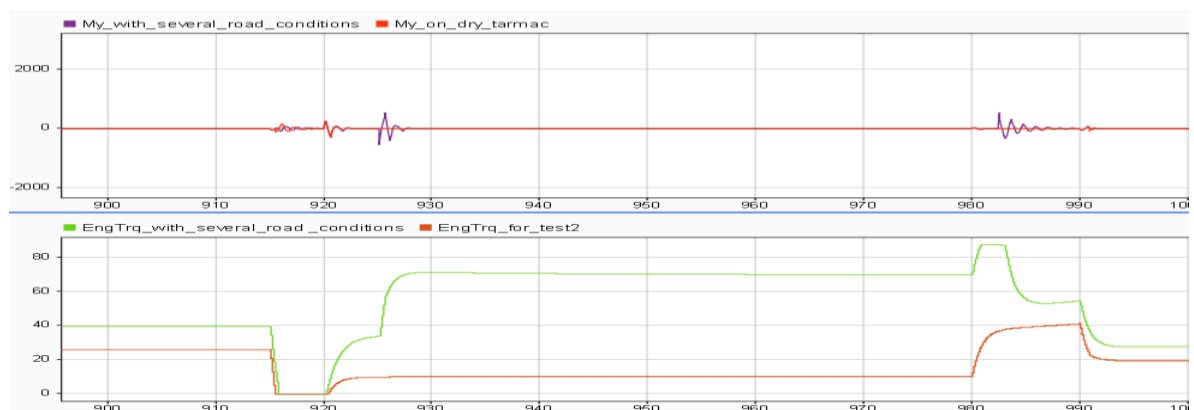


Figure 32 - Focus on test number 2 for pitch moment M_y

The 4WD-H implementation

Introduction

After the 3 degree of freedom system is validated, it must be converted to the full wheel drive through the implementation of a electric unit on the Panda's rear axle.

The aim of this work is to verify if such implementation can lead to an improvement of the vehicle capabilities when it has to face the toughest conditions of Panda Raid. For example, a classic front wheel drive Panda is not able to accelerate on sand terrain for continuous slipping. So, the model must verify if the vehicle effectively is improved by means of verification test with the full drive mode activated or not.

Moreover, it is necessary to create a proper control algorithm for the torque split between the two propulsive units. This algorithm must consider the vehicle's equipment in terms of sensors and actuators.

Control Algorithm modelling and focus on the Simulink requirements tool

Decision of the torque split strategy.

In a first place, the torque split aimed to mitigate the slipping phenomena and the loss of traction due the pitch, so the control module measured the difference in velocity between the front and rear wheels and the percentage of weight distribution on the axles due to the accelerations and static weight to modulate the output torque, by means of a proper filter, from the EM (*Modeling and Torque Split Control Strategy development of a 4WD-Hybrid Vehicle for Rally Application* [Stefano Ruscigno, 2019]).

However, there are not wheel velocity sensors on the rear wheel and accelerometers sensors on both the axles in a conventional Panda. So, there could be two possibilities for guaranteeing that implementation:

- Install aftermarket sensors on the vehicle, although these sensors have a high risk of breakage due to strong solicitations during the race. Furthermore, the installation of custom sensors could let to a re-designing stage of parts of the powertrain and/or of the chassis and this could lead to high cost.
- Obtain the values using dynamic relations, starting from known quantities.

Since both solutions lead to a very long and expensive study phase and since the aim of this work do not focus on the acquiring strategy of sensors, I decided to change the control algorithm

strategy transferring all decision-making power to the **Driver** who can decide when and how much time the EM is active.

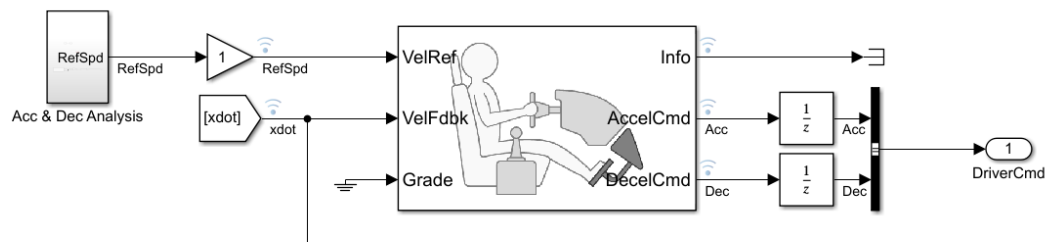


Figure 33 - Longitudinal Driver Model

This way, I can plan my control strategy exploiting signals that are computed by the existing sensors such as **acceleration pedal position** and **wheel rotational speed** to make the EM able to ease the load of the front engine and to proper charge the battery.

However, the control algorithm must monitor the decisions taken by the Driver. Indeed, it analyses **only** the “**Battery State of Charge (BSoC)**” parameter for allowing the EM to work as a Motor or as a Generator.

Function Requirements

The process starts with the analysis of the **function requirements**. This is a very important step because it defines what is expected from the control software in terms of functionality.

A good **Functional requirement** must be:

- Clear and unambiguous.
- Testable (verifiable).
- Feasible.
- Consistent.
- Atomic.
- Traceable.

After analysing the development of the product, the engineer must draft a proper **list** of the requirements the model is supposed to have.

Functional requirements of the Control Algorithm by means of Simulink Requirements tool

This section, I would like to show an example of a formal lay-out of **Functional Requirements** and their verification through suitable program tools

A poor **requirements management**, in terms of missing, incomplete, and inconsistent requirements that do not capture the intended purpose of the project, is the most common reason for embedded project's failure. *Fixing the requirements Mess, [Christopher Lindquist, 2005]*.

A rigorous lay out of “**Textual Requirements**” is fundamental on **Model-based Design Workflow** because it is the basis of the Simulink's designing and it gives the possibility to test the several functional behaviours that are strictly linked to those requirements. Therefore, it allows to find the faults and to develop the system in a rigorous way and compliant with the standards and certifications.

There are many programs (DOORS, POLARION) that allows to draft a list of requirements that is compliance with the standard certifications such as ASIL.

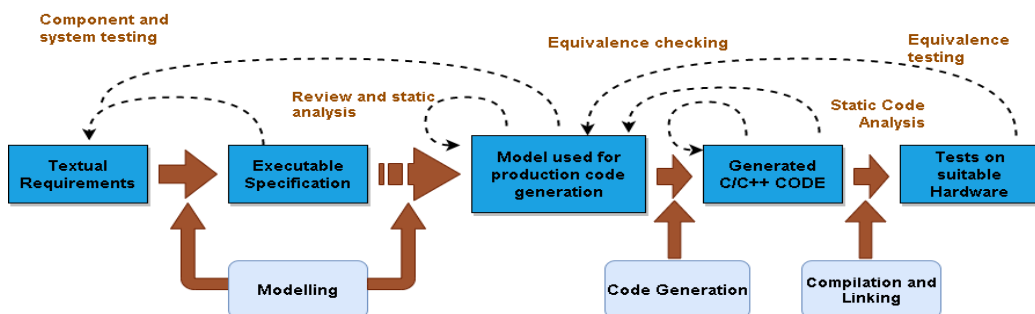


Figure 34 - Model-based design workflow for critical embedded Software. it is approved by the standard authority

Moreover, it is necessary to establish **traceability** between requirements and design. So, after the lay out of the requirements on a document creation tool like **Microsoft Word**, these are imported inside the Simulink workflow through the tool **Simulink Requirements**, which can update synchronizes changes from source, edit and add further details to import and create author requirements.

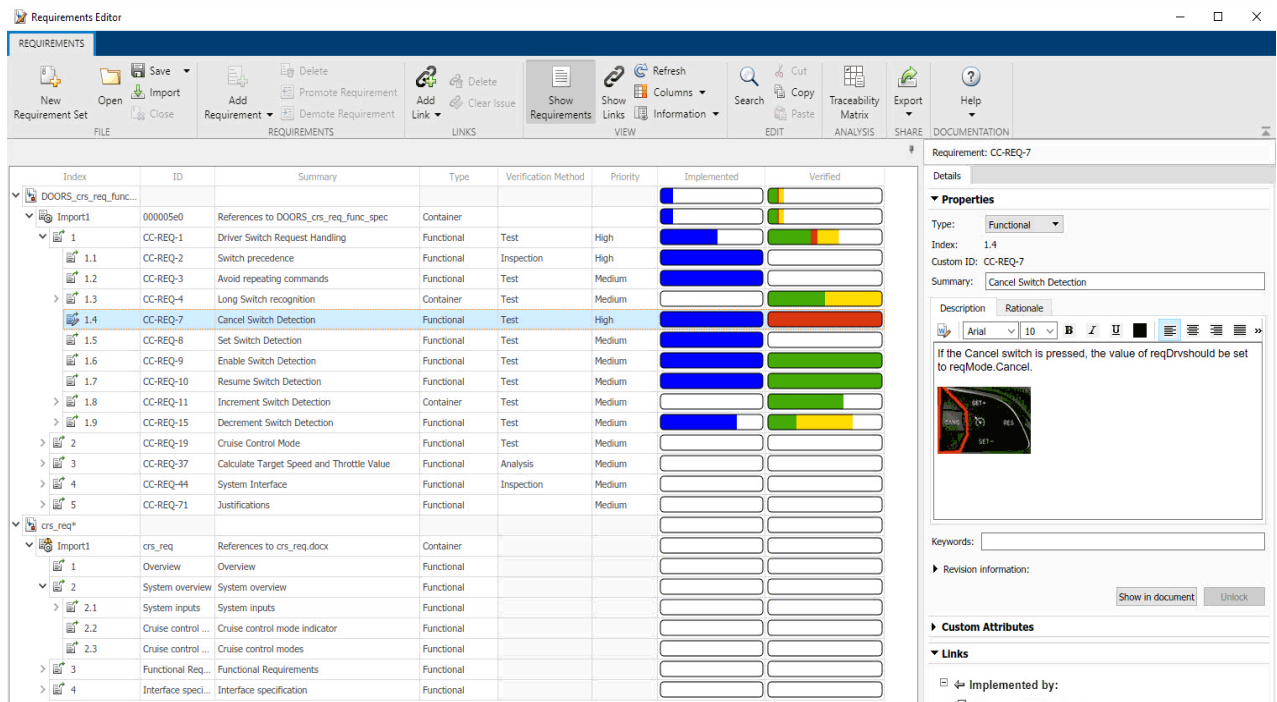


Figure 35 – Example of Simulink Requirements Editor dashboard

Such tool allows also to easily create a bidirectional link (a drag & drop operation) between the requirements and the model, establishing **traceability**. In this case, I show the links between the **control functional requirements** and of the control algorithm's **state machine**.

The functional requirements the control algorithm must respect are:

- Turn the **EM** on the state “**ON**” when the **Driver** toggles the switch named “**EM_BOOL**” on “1”.
- Compute a null torque when the **EM** is on the state “**ON**”.
- Turn the **EM** on the state “**OFF**” when the **Driver** toggles the switch named “**EM_BOOL**” on “0”.
- Compute a null torque when the **EM** is on the state “**OFF**”.
- Turn the state of the **EM** from “**ON**” to “**Motor_state**” when the **Driver** toggles the switch named “**Mot_bool**” on “1”, the switch “**EM_BOOL**” is on “1”, **BSoc** is higher than 40% and the rolling velocity of the front tire is higher than 10 rad/s.
- Compute a reference positive torque by means of the function “[TorRefMot]=cmd_torq1(Acc, RedRatio)”, when the EM is on “**Motor_state**”.
- Change the state of the **EM** from “**Motor_state**” to **ON** when the “**BSoc**” is lower than the 20%, the switch “**Mot_bool**” is on “1” and the switch “**EM_BOOL**” is on “1”.

- Change the state of the **EM** from “ON” to “**Gen_state**” when the “**BSoC**” is lower than the 95%, the switch “**Mot_bool**” is on “0” and the switch “**EM_BOOL**” is on “1”.
- Compute a reference negative torque, that is computed by means of the function “[TorRefMot]=cmd_torq3(Acc,RedRatio) when the switch “**EM_BOOL**” is on “1”, the switch “**Mot_bool**” is on “0” and the **BSoC** is below the 95%.
- Compute a null reference torque in order to stop the charge when the switch “**EM_BOOL**” is on “1”, the switch “**Mot_bool**” is on “0” and the **BSoC** is over the 95%.

So, once the traceability is done for both the requirements and design model point of views, it is crucial to perform **Unit Testing** for verifying those points.

In this operation the specified component (**Component Under Test**) is isolated from the system and tested giving to it specified **inputs** for verifying the behaviour.

The inputs can be injected manually, by dashboard signals, pre-code signals and **Simulink Editor** blocks. Same thing can be done for the outputs results. A proper tool for this operation is **Simulink Test**

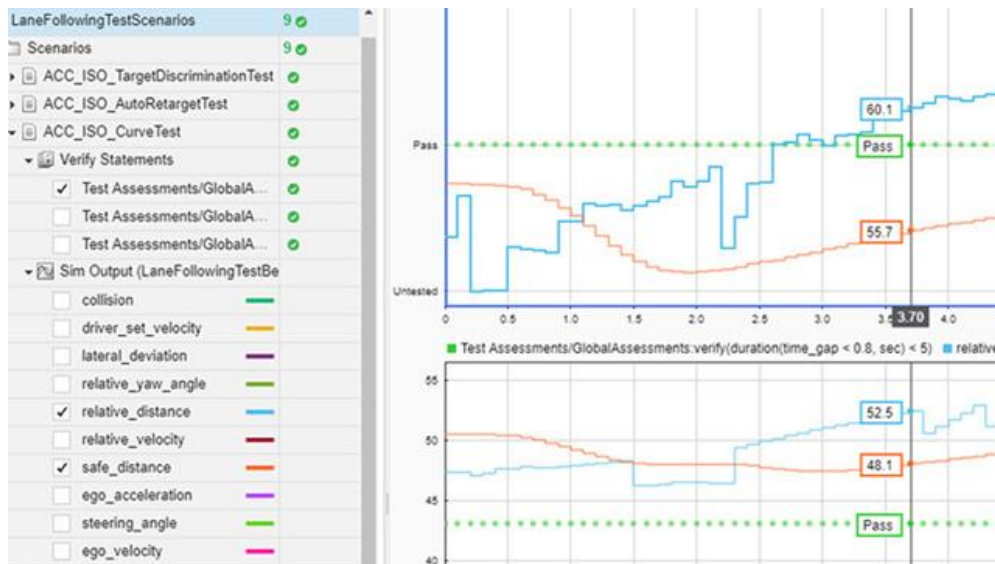


Figure 36 - Simulink Test manager dashboard

In order to verify the demands, I perform several test on the “**EM’Controller**” subsystem whose aim is to see if the conditions that allow the toggling between the EM’s states “**em_on**” and “**MotMod**” is respected.

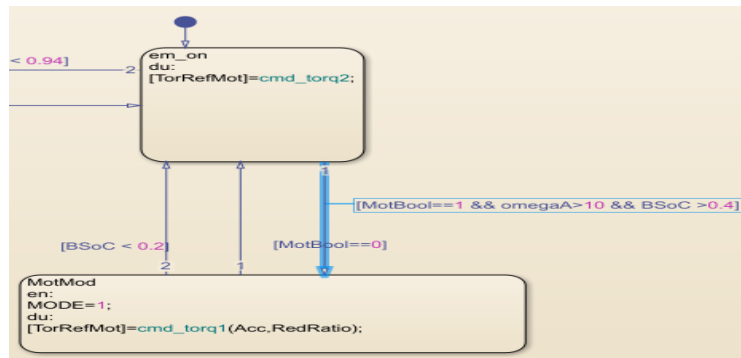


Figure 37 - Condition under test

The verification is performed through an **iteration** of several test, each one that injects different input signals to verify all the possible situations, considering the **independences of each condition**. In particular, the tool assesses the logic condition shown in Figure 45 for each test:

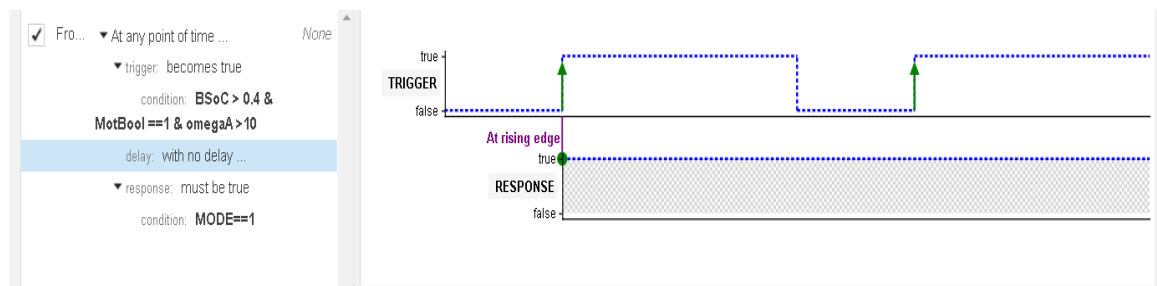


Figure 38 - Logic condition for each iteration

After the iteration, all the tests **result positive**:

Results: 2021-Oct-11 17:45:14	3 ✓
step2_vers11_sim_Harness4	3 ✓
BSoC low	✓
BSoC high	✓
OmegaR low	✓

Figure 39 - Positive iterations

It means that the requirement is verified (from the tool point of view). It is also important, in addition to the link between requirements and model design, to fix **traceability** between requirements and those assessments that verify them. It is possibly again through Simulink Test.

NAME	REQUIREMENTS	SIGNAL EDITOR SCENARIO OR SIGN
✓ BSoC low	3.1.4.6 From em_on to MotMod: From em_on to MotMod (Requirements=58)	BSoC low
✓ BSoC high	3.1.4.6 From em_on to MotMod: From em_on to MotMod (Requirements=58)	BSoC high
✓ OmegaR low	3.1.4.6 From em_on to MotMod: From em_on to MotMod (Requirements=58)	OmegaR low

3.1.4.4 From M...	From MotMod to em...	
3.1.4.6 From e...	From em_on to MotMod	
3.1.4.7 From O...	From ON to Gen_status	
3.1.4.8 From G...	From Gen_status to ON	

Figure 40 - bidirectional link between requirements and assessment

Model in the Loop simulation for the EM activation

In analogy with the 3DOF system's dynamic, I perform the MIL for verifying the functionality of the EM's controller and its implementation inside the Simulink environment together with the associated hybrid system in the plant. This is done to see if the fixed **Functional Requirements** are respected.

Also in this case, I exploit the “**Data inspector**” Simulink tool and the **Drive-Cycle profiles** I create for this simulation.

Simulation for the torque split ruled by the driver.

As reported above, the decision making for the activation/deactivation of the EM and in which modality it should work is ruled by the **Driver**.

So, I create a simulation for each custom **Drive Cycle** relative for each custom **Scenario**, where changes of terrain's characteristic are inserted. Then, I simulate the Driver's commands so that the EM works as predicted when there is a change of terrain condition or change of vehicle's velocity.

Simulation test in Second Scenario for different road conditions

During the Panda Raid, a **speedy trial** could be scheduled. It consists of a stage of around 50 km on which the vehicles must register their own **fastest lap**. On this scenario, the Panda is intended to travel at its limit, with the EM always turn on the **Motor mode** during adverse road conditions (soil, sand, etc).

First, I simulate the system with the **EM always off** to see if the vehicle can face these conditions. Then, I repeated the test with the **EM** that follows the commands from the Driver to compare the results.

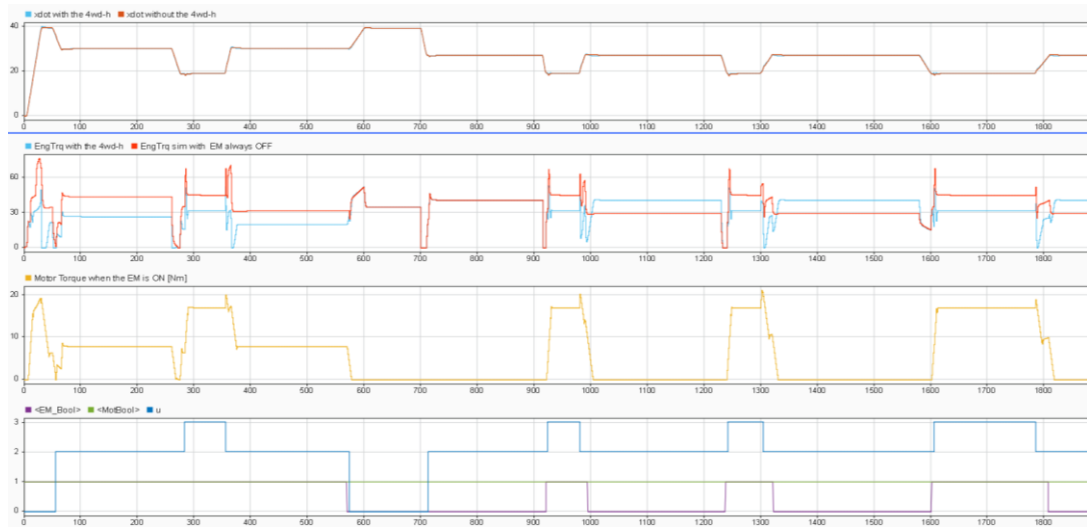


Figure 41 - Test results for Second Scenario

As it is shown in Figure 41, the **EM** effectively helps for the traction when it is ON and in Motor Mode. Indeed, when “**EM_Bool**” is equal to “1”, the **ICE** unite gives a smaller output torque with respect to the simulation where the **EM** is always off.

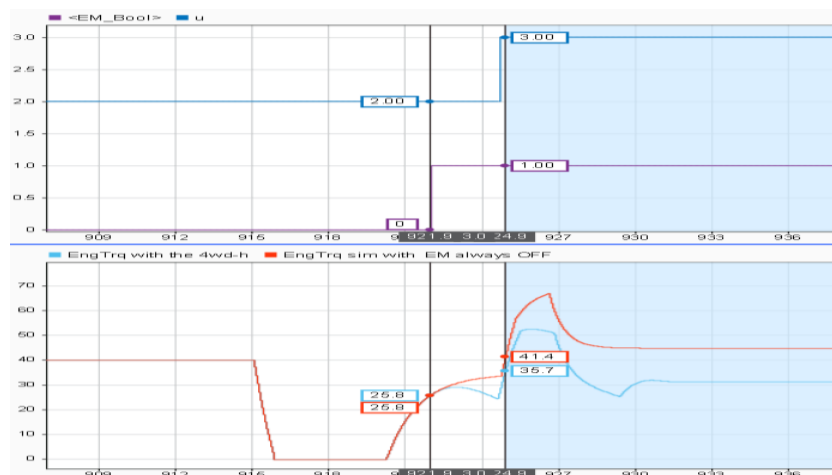


Figure 42 - Focus on the EM activation due the sand path

Simulation test in Third Scenario for different road conditions

I design a reference Drive Cycle similar as much as possible to the typical situation the vehicle can find during the competition: it is characterized by a prevalence of soil at medium-high velocity, in which the driver can both use the EM as a **Motor** or as a **Generator**, and sporadic sandy stretches on which it could be needed to use the 4WD-H traction.

From the simulation, I notice that when the EM works as a Motor (**MotBool** equal to “1”) the output torque from the ICE is lower than the one provided when the 4wd-h is disabled.

Overall, the MIL for the hybrid and torque split strategy results verified.

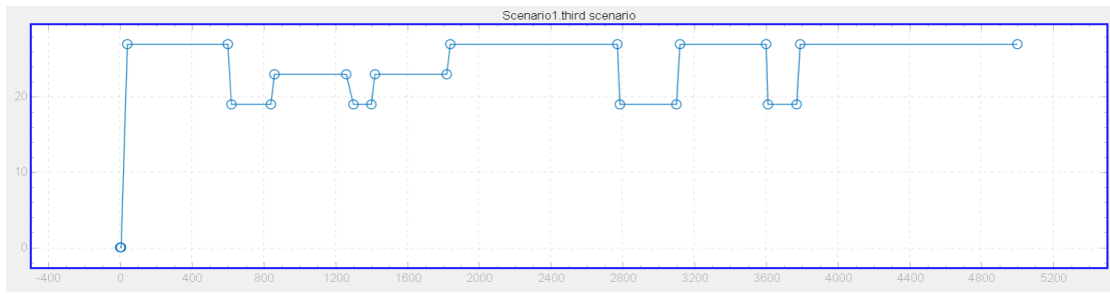


Figure 43 - Drive cycle for the third scenario

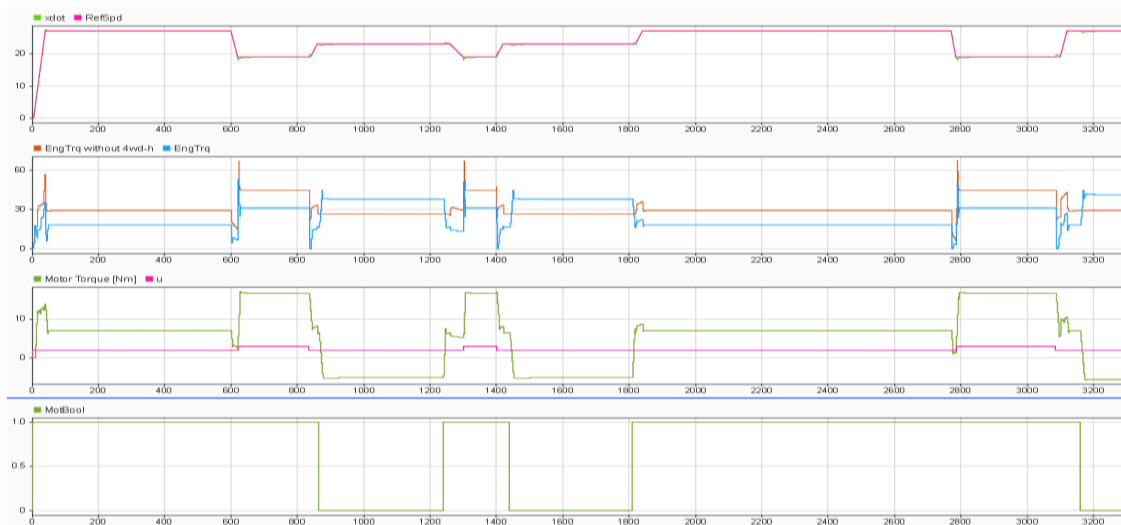


Figure 44 - Test results for Third Scenario

However, when the EM works as a Generator (**MotBool** equal to “0”), the ICE’s torque is higher. This is a consequence to the fact that the EM is requesting negative torque to be converted in current for the battery pack and so the load for the ICE is greater.

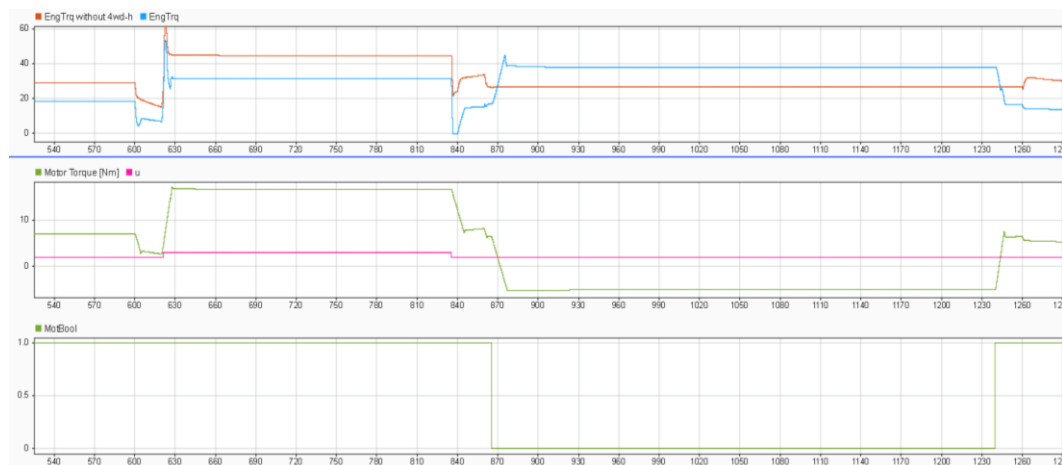


Figure 45 - Focus on the EM, both as a motor and as a Generator, activation due the variation of the terrain condition

Energy consumption analysis of the EM for a proper determination of the battery capacity

Introduction

A proper analysis of **energy consumption** is needed for the determination of the energy reservoirs' capacities.

From the technical guide of FIAT PANDA 1000 FIRE I.E and from user experiences, the average fuel consumption of the ICE is around 21.3 km/l at 90km/h, although these data are referred to urban cycle. Moreover, they are computed for a conventional vehicle, that is lacking hybrid stuffs that increase dramatically the weight although the EM should help in the traction.

So, as long as proper empiric experiences are performed with the prototype of the running vehicle, I intend such data as the minimum level of fuel consumption.

However, I need to simulate the behaviour of the EM to obtain its data about both mean energy consumptions and peak one. Such data are obtained as follows:

$$\text{Energy consumption} = \frac{\int I_{\text{batt}} * V_{\text{bat}} dt [\text{Wh}]}{\text{Run_distance}[\text{km}]}$$

The EM is implemented through the block “**MappedMotor**” Simulink block:

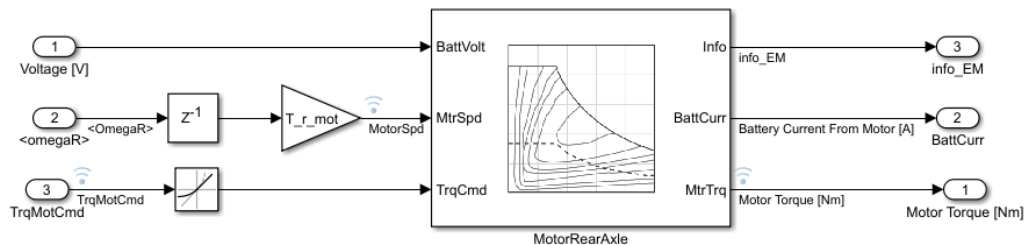


Figure 46 - Mapped motor for the EM

Whose characteristic parameters are the following:

- Maximum Torque = 88 Nm
- Torque control time constant = 0.02
- Maximum power = 3000 W

I aspect for this electric unit the following data:

- Maximum Consumption uphill: 250Wh/km
- Maximum Consumption on a flat surface: 120Wh/km
- Average Consumption on a flat surface: 100Wh/km

Analysis for the determination of the battery capacity

The aim of this work mainly focuses on the study of the dynamical vehicle in a context of a Rally competition. So, a proper detailed analysis on the battery dynamics and management by means of a Battery Management System is not needed.

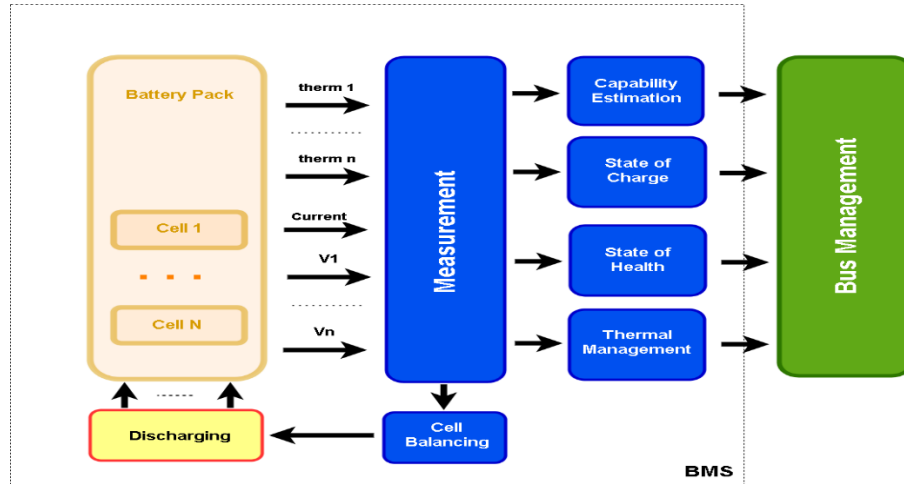


Figure 47 - Battery Management System logic scheme

However, a proper study for the determination of the **battery capacity** is a crucial problem. Indeed, it is necessary to identify the most suitable size of that component, since all the physical implementation such as wiring, weight distribution, cooling system and battery controller will be implemented starting from this data.

Basically, our vehicle is compatible with a battery whose size can be 5kWh or 10kWh.

This choice must be justified by several analysis tests on the **charging cycles** and on the **discharging cycles** that the EM could perform exploiting the occasions that the rally can offer.

Indeed, there are some road paths that force the use the **4WD-H**, so there will be a discharging of the battery, and some of them that instead will allow the vehicle to use the EM as a **Generator**.

The **Functional requirement** concerning the battery's charging status declare that the driver must try to keep the status of the battery charge into the 85%-20% interval.

Therefore, it is necessary to design several **road scenarios** the vehicle might face.

These must urge the system to as many of the situations, which stress the electrical subsystem and put in evidence its capabilities.

Simulations

Preliminary analysis

Before the analysis of on the scenarios, I perform some simulations on which the vehicle moves at constant velocities on different road conditions to analyse the **average consumptions**.

Legend of the figure:

- **Blue bar:** Dry Tarmac.
- **Orange bar:** wet road.
- **Grey bar:** Soil Road.
- **Yellow bar:** sand path.

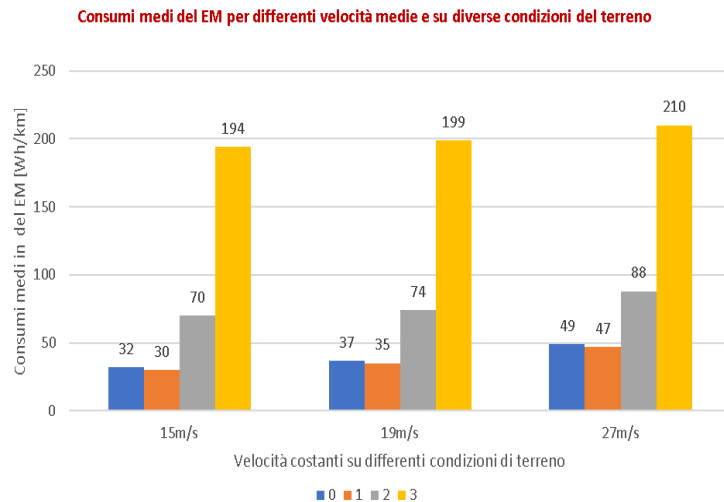


Figure 48 - Average Consumption of the EM for different constant velocities and for different road conditions

From this preliminary analysis, it is possible to make an idea on the needed size of the battery and so to verify the dynamic of the different stages.

Simulation of First Scenario

On this analysis, the Panda must address a rally stage characterized by a **very long sand path**, whose length is around 15km. This quantity is obtained thanks to the interviews with the past participants of the competition, and it represents the longest possible and so the worst scenario for what concerns the discharging cycle of the battery. First, the simulations are performed using a battery of 5kWh. Then, they are repeated with a 10kWh.

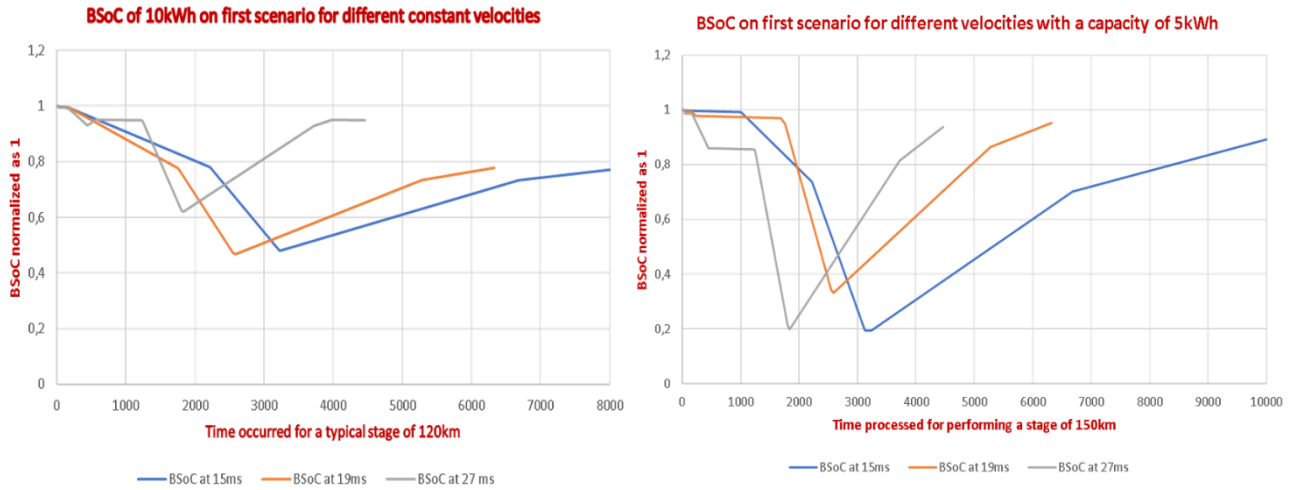


Figure 49 - Battery State of Charge for different constant velocities on first scenario with a capacity of 10KWh and 5KWh

Simulation of Second Scenario

The simulation is performed both with a 5kWh and 10kWh battery.

On the Figure 14, there are represented the Energy consumptions along all path for the second scenario and the average consumption for the overall travel.

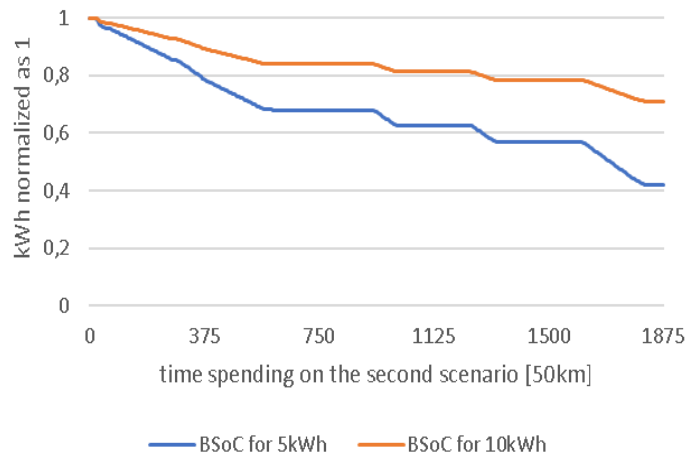


Figure 50 - Battery State of Charge for the Second Scenario Drive cycle on first scenario with a capacity of 10KWh and 5KWh

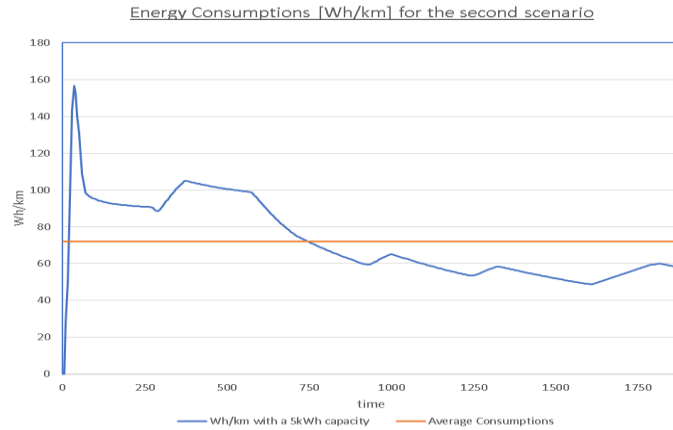


Figure 51 - Energy Consumptions [Wh/km] for the second scenario

Simulation of Third Scenario

I design a Reference drive cycle similar as much as possible to the typical situation the vehicle can find during the competition: it is characterized by a prevalence of soil a medium-high velocity, in which the driver can both use the EM as a Motor or as a Generator, and sporadic sandy stretches on which it could be needed to use the FWD traction.

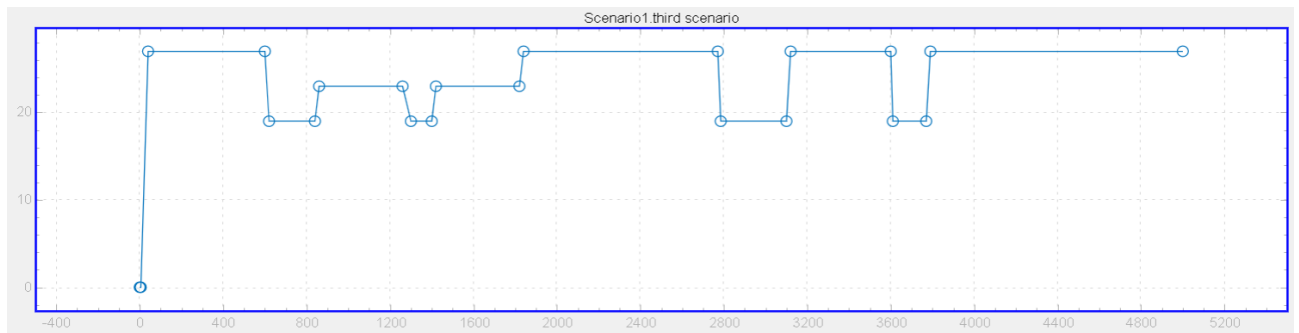


Figure 52 - Drive cycle for the third scenario

The simulation is performed both with the 5kWh and 10kWh battery size. The results are represented in Figure 16, where the data obtained match with the preliminary analysis performed at the beginning.



Figure 53 - Battery State of Charge for different constant velocities on third scenario with a capacity of 10KWh and 5Kwh

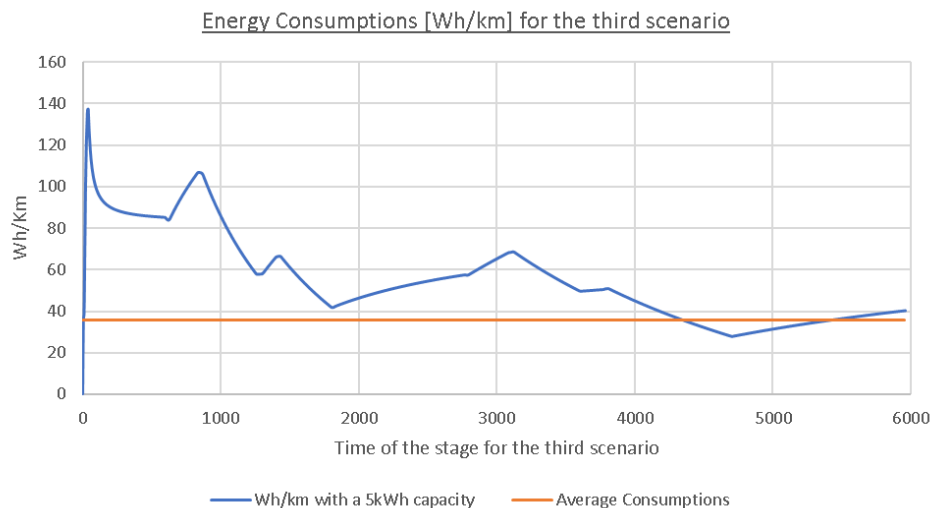


Figure 54 - Energy Consumptions [Wh/km] for the third scenario

Conclusion

From these simulations, I decide that no further scenarios are needed. The 5kWh battery capacity seems to be at least sufficient for the road paths the Panda has to face, but it makes the driver use the EM in Motor mode following a rigid scheduling for saving as much as possible energy capacity.

But in a context of a rally competition where it is usually that unexpected events happen, it is more useful to have a size capability large enough to deal with the competition maintaining a certain security lapse.

In conclusion, I think that the 10kWh battery size is the most feasible for the Panda Raid.

The Hybrid V-Cycle and the Modular Technical Model

After the verification of the system model, it is necessary to add further details about the V-cycle to see details about the different computing platforms that are used in the workflow. In this way, the structure appears expanded in the form of a **Hybrid V-Cycle** workflow.

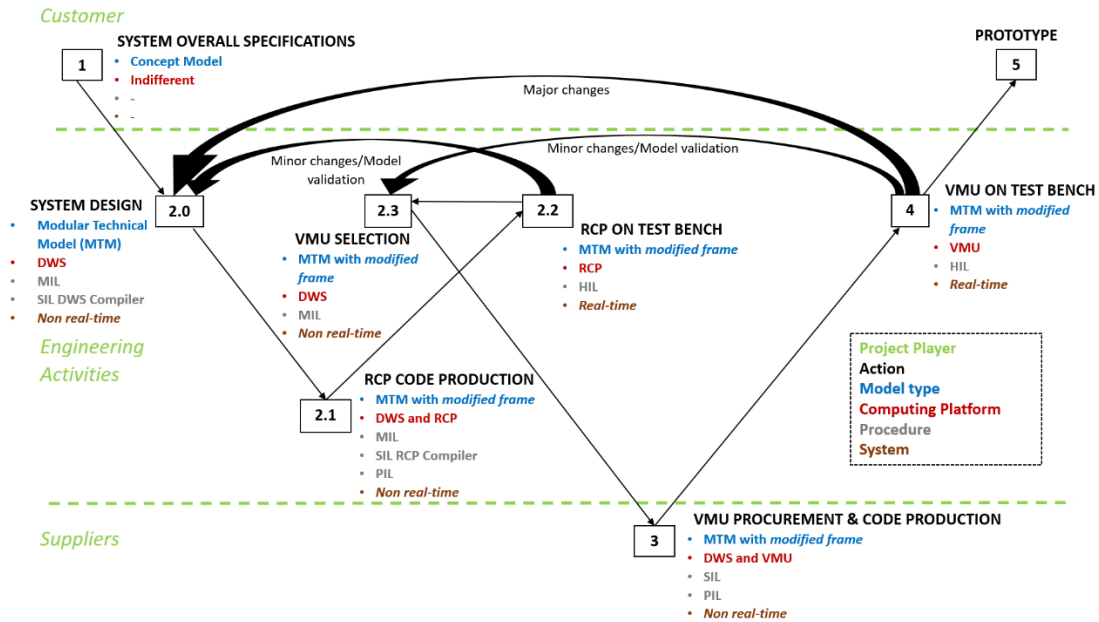


Figure 55 - Hybrid V-Cycle

In particular, the platforms in which I worked are:

- the **DWS (Development Workstation)**, which refers to my own personal computer. This is intended for the entire software implementation of my model and for simulation purpose (**MIL**). The latter are performed not in real-time, so without any considerations on the final target hardware.
- The **RCP (Rapid Control Prototyping)**, which refers to **MicroAutobox 2** from **dSpace**. This is intended for **SIL** (and **PIL**), where the control software, after a proper **code generation** from the module itself, run in RCP platform. The model built at the “**System Design**” phase is modified including all the characteristics of the RCP hardware, in particular interfaces of the input and output’s ports. The simulation is still not in real-time. During the SIL simulation the DWS is not connected to the RCP hardware.
- **VMU (Vehicle Management Unit)**, which refers to the final **Electronic Control Unit (ECU)** and it is still intended for SIL (and PIL).

For the details about each step shown in the Hybrid V-Cycle, read the thesis *THERMAL PREDICTIVE CONTROL FOR HYBRID AND ELECTRIC POWERTRAIN* [Coletta, 2020] and *Project for the implementation and validation of the Vehicle Management Unit (VMU) code of a hybrid car following the VCycle development strategies described in ISO 26262* [Serrano, 2020].



Figure 56 - MicroAutobox 2 from dSpace

Modular Technical Model (MTM) For MIL & SIL simulations

The MTM is a standard way to lay out the different subsystems inside my software in order to organize its modules into special groups that are characterized by their own functionalities. In this way, it is possible to separate all the disciplines involved in the design of the system components and to accurately define the interfaces. It is very useful when the software is transferred into different hardware platforms since the layout results independent from the specific application under study.

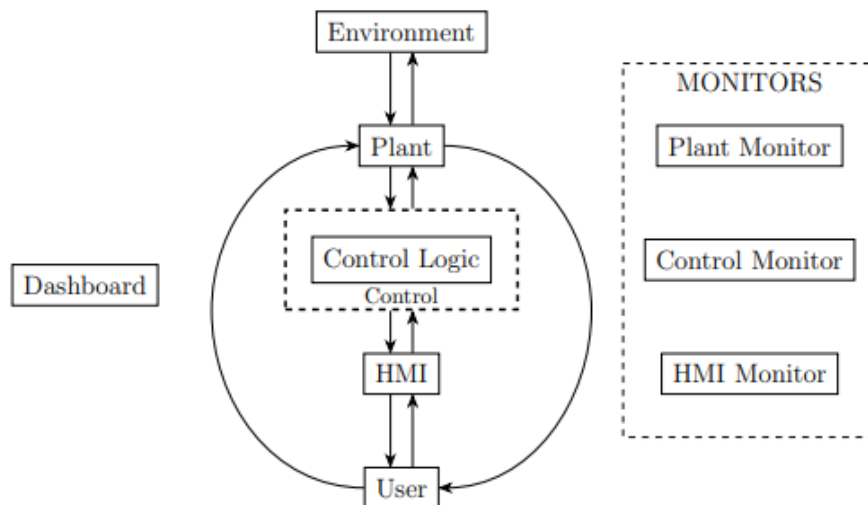


Figure 57 - Ideal Modular Technical Model

For the details on the specific component of MTM and code generation process for a dSpace MicroAutobox2, read the thesis *THERMAL PREDICTIVE CONTROL FOR HYBRID AND ELECTRIC POWERTRAIN* [Coletta, 2020].

I want to highlight the aspects regarding two modules of the MTM in particular:

- **The plant:**

It is constituted from all the components of the model that must be controlled and it handles continuous signals. So, it contains all the vehicle's parts involved in the **3 degrees of freedom dynamic, such as Vehicle Body, Wheels, and the propulsive modules (EM, ICE).**

In particular, I insert into this subsystem the module regarding the ICE's torque command. Indeed, the vehicles allowed to participate to the Panda Raid are not equipped with a **digital actuation** of the acceleration pedal (es, Ride by Wire), but the command is a task of a wire connected to the throttle valve. Since such actuation does not give a feedback signal concerning the status of the pedal, it is not possible to perform any kind of control action.

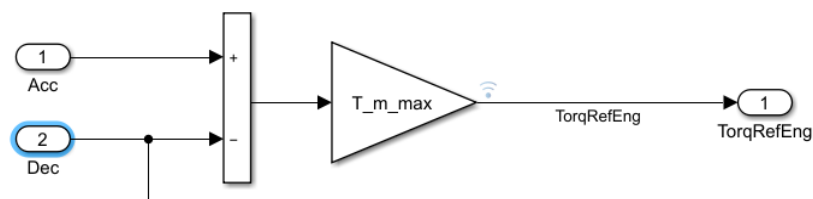


Figure 58 - ICE's torque logic actuation

- **The Control Logic:**

It is constituted from all the components of the model that perform control operation on the plant and it handles **discrete (digital) signals**. So, it is formed by the **control module of the EM** and the **supervisor** module, which monitors the status of the running program. Such components are designed using the “**State-flow**” Simulink tool. The Control Logic has a **frame** (Control) that decouples and I/O device from the control algorithm. In this way, it is possible to make the Control Logic and the target hardware interact including device specification without affecting the control law. In this way the MTM is adapted to the target hardware and so the code generation process, that extracts the application code from the Control Logic, remains feasible.

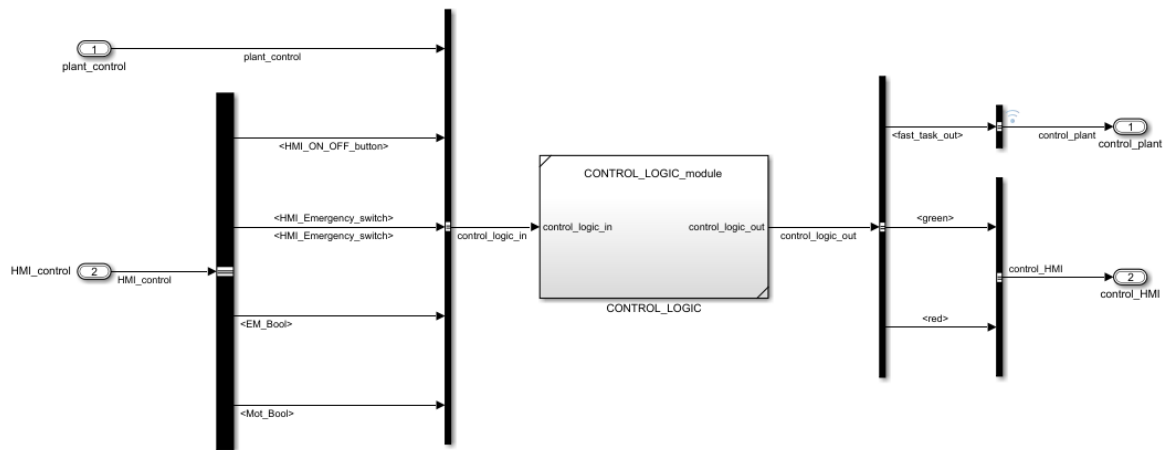


Figure 59 - Control Logic Frame

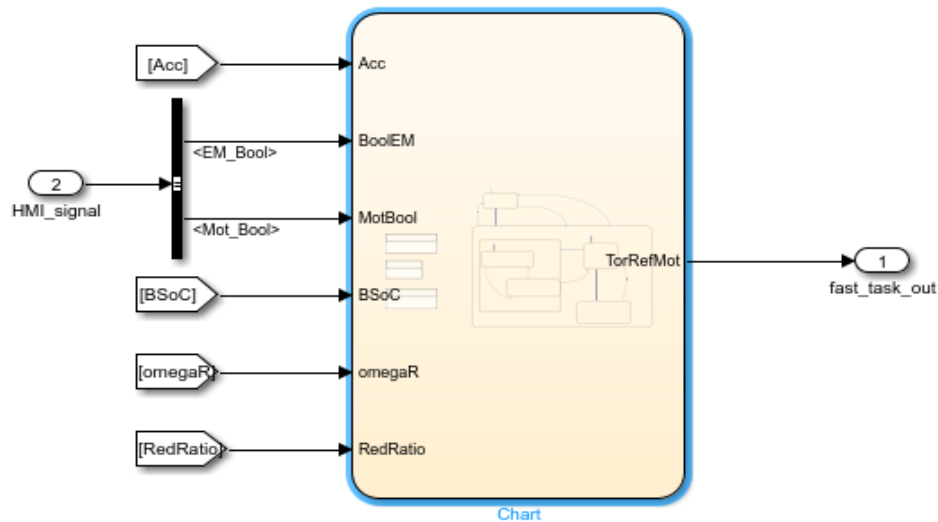


Figure 60 - EM's controller module

Code Generation for SiL performed on RCP platform

Once the **MiL** is verified from the requirements point of view on the DWS platform, it is time to verify the quality of the software through a **code generation** process and a **SiL** simulation on the RCP platform. This phase is performed automatically by the **Embedded Coder** in the Simulink environment.

But before starting the code generation, I must configure the tool such that it takes into account the **target hardware** on which the software is intended to run (in this case, a platform provided by the dSpace). Indeed, the generic interfaces defined in the frame control at MiL (Figure ...) phase are modified to match those of the RCP platform. In this way, the coder considers of software architecture, memory allocation, sampling time and specific hardware structure for what concern the I/O ports (ADC, DAC).

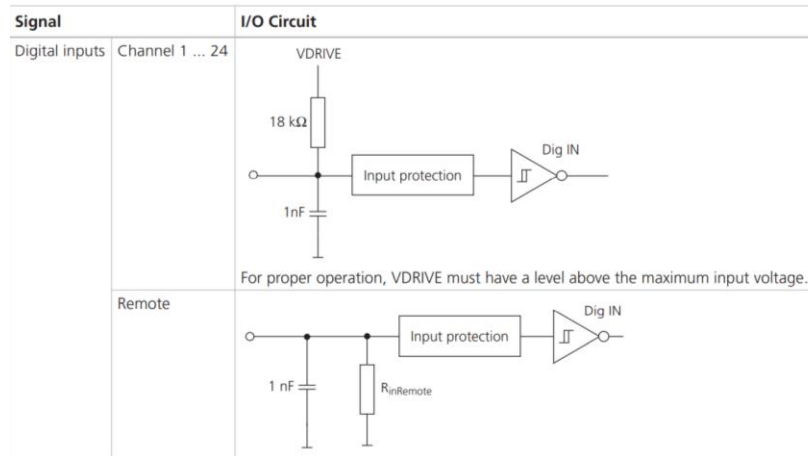


Figure 61 - Simplified diagrams of the I/O circuitry of the digital input of a MicroAutoBox II 1401/1512/1513

Actually, the **Processor in the Loop** simulation is needed after the SiL, but the former requires the use of **TargetLink software** from dSpace (if I am still using the MicroAutobox 2 hardware). But it requires a licence that the laboratory have not got. So, the tests for the PiL are skipped.

Code generation for the SiL on the RCP platform exploiting the rti1401 library for the I/O ports.

To be able to interact with the actual inputs and outputs of the RCP platform it is necessary to define these inside the frame control. These blocks are available inside the rti1401 library and replace the standard input/output block of Simulink *Project for the implementation and validation of the Vehicle Management Unit (VMU) code of a hybrid car following the VCycle development strategies described in ISO 26262 [Serrano, 2020]*.

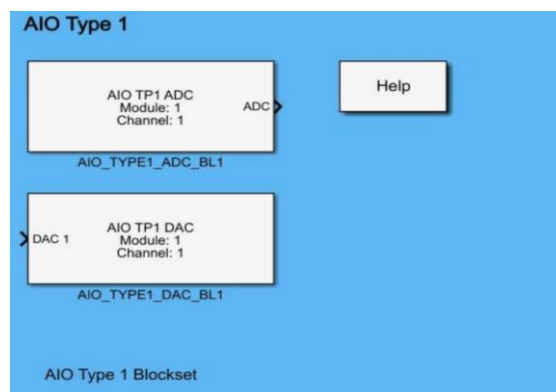


Figure 62 - DSPACE AIO library for MicroAutoBox II DS1513 AIO

Therefore, I insert in the control frame the most suitable interface blocks from the library that take into account both for the nature of the port chosen and the kind of signal I am using and recording.

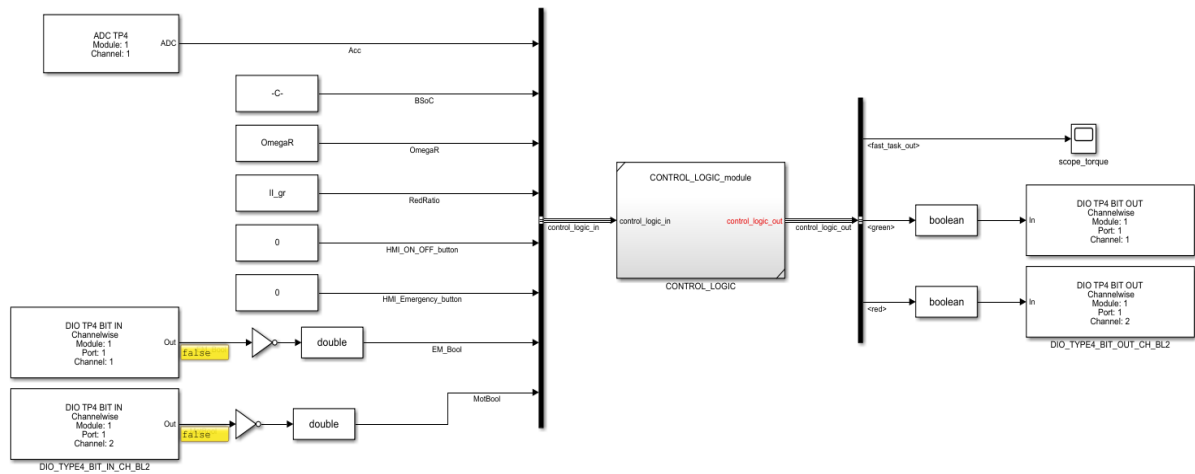


Figure 63 - Control Logic frame with the proper I/O interface blocks

Once the control frame implementation is completed, I translate such subsystem into the runnable code. This phase is perfectly done, so I consider the SiL concluded.

HiL Simulation attempts through the laboratory testbench

As I mentioned above, the PiL simulation is not possible since the license for the software “**TargetLink**” is missing. However, it is possible to perform a sort of HiL exploiting the hardwares such as multimeter, signal generator and oscilloscope, that are in the laboratory and that can simulate both the digital input signals and the analogical ones associated to the Driver’s command.

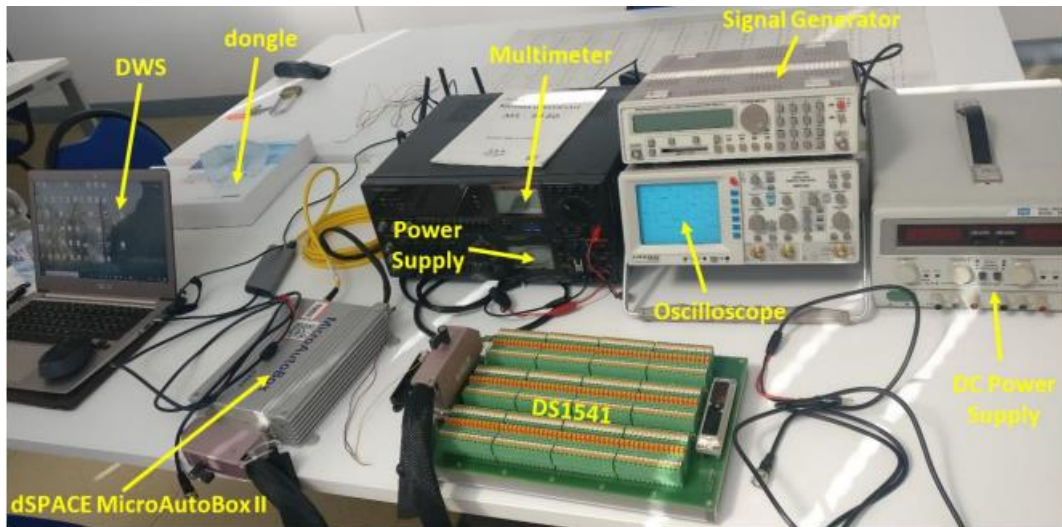


Figure 64 - Testbench equipment

To do this, it is necessary to connect the I/O ports of the **RCP breadboard (DS1541)**. This can handle both logic **signals**, that are received through digital ports driven in according to the hardware structure of the RPC platform (Figure 55), and analogical ones, through ADC ports.

The control code requires from the Driver:

- Logic signal **EM_Bool**
- Logic signal **Mot_bool**
- Analogic signal **Accelerator pedal**

Meanwhile, it receives from the plant:

- The BSoC
- The radial velocity of the front wheel

I implement the signals from the Driver through some logic switches and a **pedal** that communicate with the RCP through respectively the digital ports and the ADC port of the breadboard.

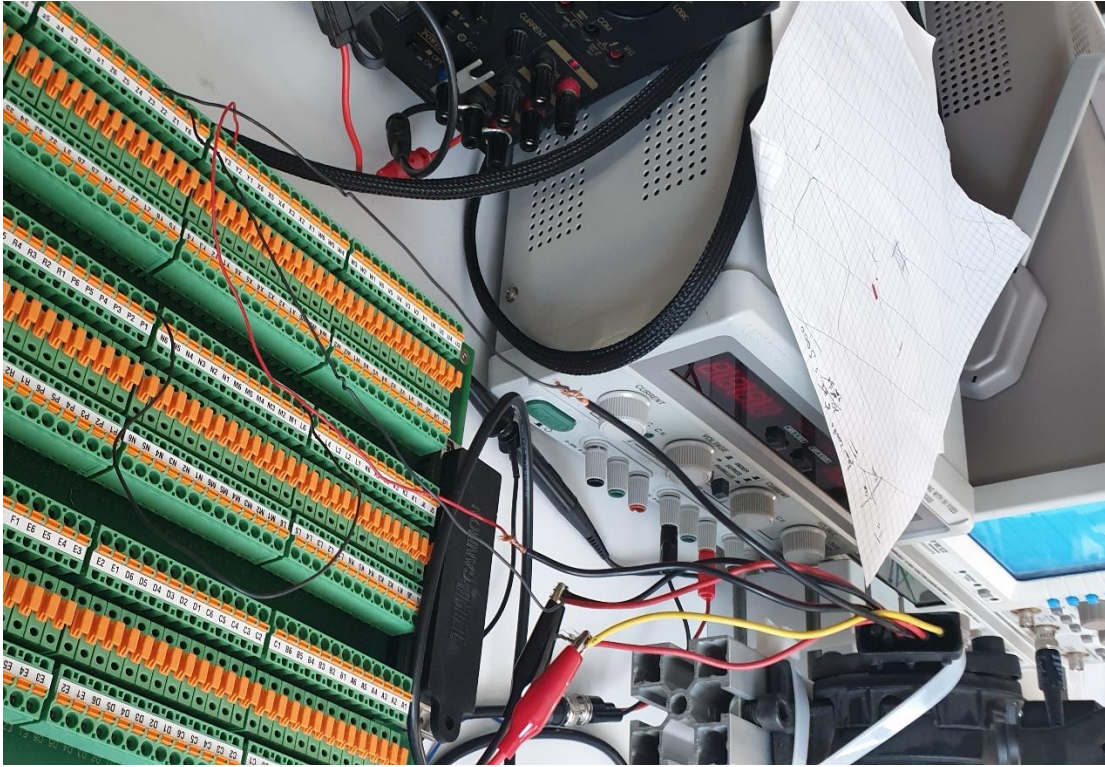


Figure 65 - implementation of the links on the DS1541 for the input signals

For what concerns the signal from the plant, I need to simulate them by means of the **Control Desk** software since there is not a suitable **Testbench** in the laboratory. This software, which runs on the **DWS**, controls the communication with the RCP platform, loads the codes, monitors their execution and allows to interact with some variables in real time.

Indeed, thanks to a custom dashboard layout I create before the simulation, I can vary crucial quantities inserted in the model in order to modify the working conditions of the controller.

In this case, I insert the **BSoC** and the **OmegaA** signals as constant, whose values can be modified in run time for verifying the control's logic. They are initialized as “1” and “20” respectively.

First HiL test, $BSoC = 0.5 * BattChargeMax / 100$ and $\Omega_{A}=100$, $Em_Bool = "1"$ and $Mot_bool = "1"$

In this test I verify if the control software gives a positive output torque given the suitable conditions.

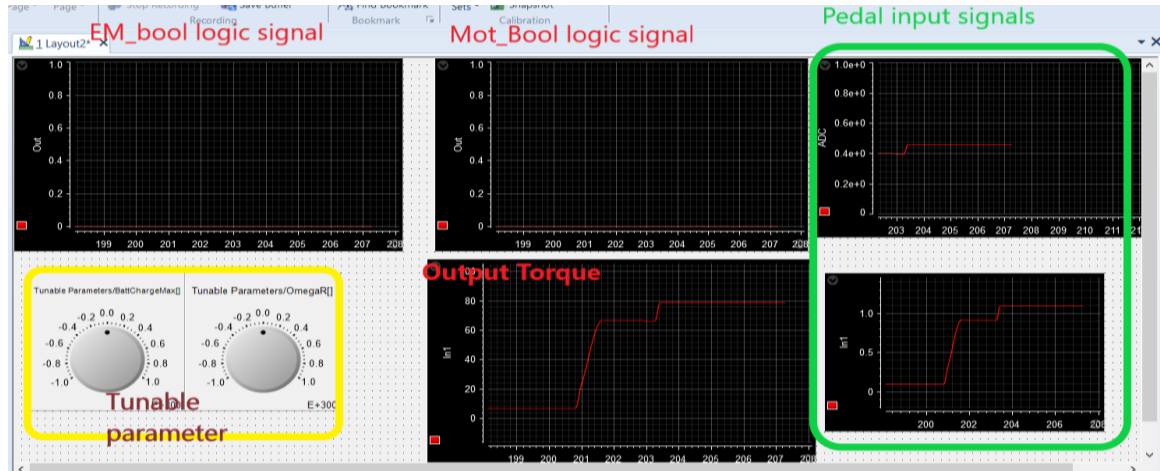


Figure 66 - First HiL test

N.B. the logic signal appears inverted with respect to the input values I am giving. This is due the hardware architecture of the input's ports.

As shown in the Figure 66, the task is accomplished successfully. So, the functionality is verified.

Second HiL test, $BSoC = 0 * BattChargeMax / 100$ and $\Omega_{A}=100$ $Em_Bool = "1"$ and $Mot_bool = "0"$

In this test I verify if the control software gives a negative output torque given the suitable conditions.

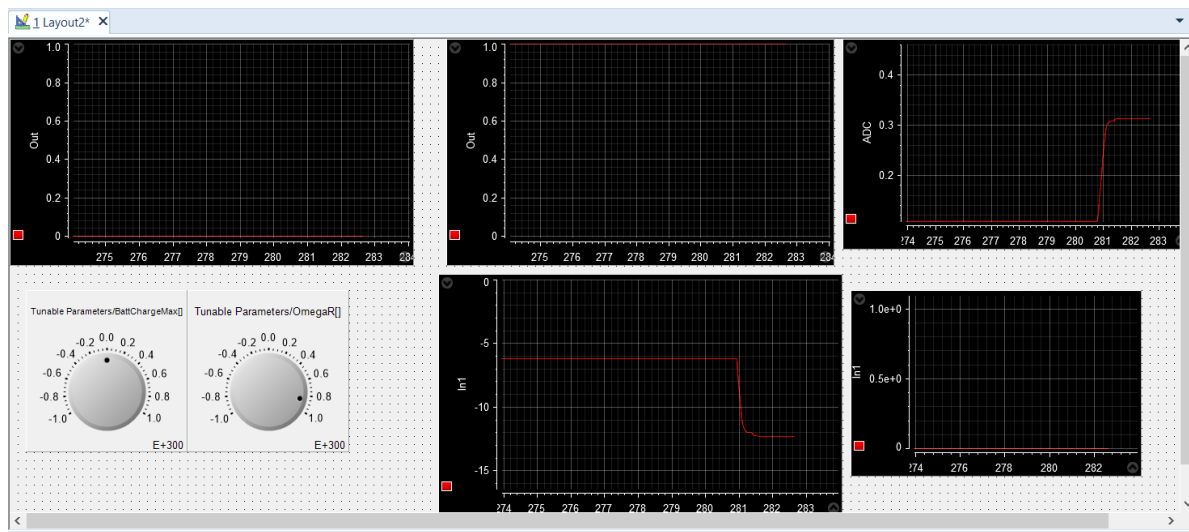


Figure 67- Second HiL test

As shown in Figure 67, the output reference torque results negative and it follows the logic I implemented inside the control logic. So, the functionality is verified.

Third HiL test, BSoC= $0 \cdot \text{BattChargeMax}/100$ and $\Omega_A=0$ Em_Bool = “1” and Mot_bool = “1”

In this test I verify if the control software gives a null output torque when the Driver asks for a positive torque, but the system’s conditions do not allow it.

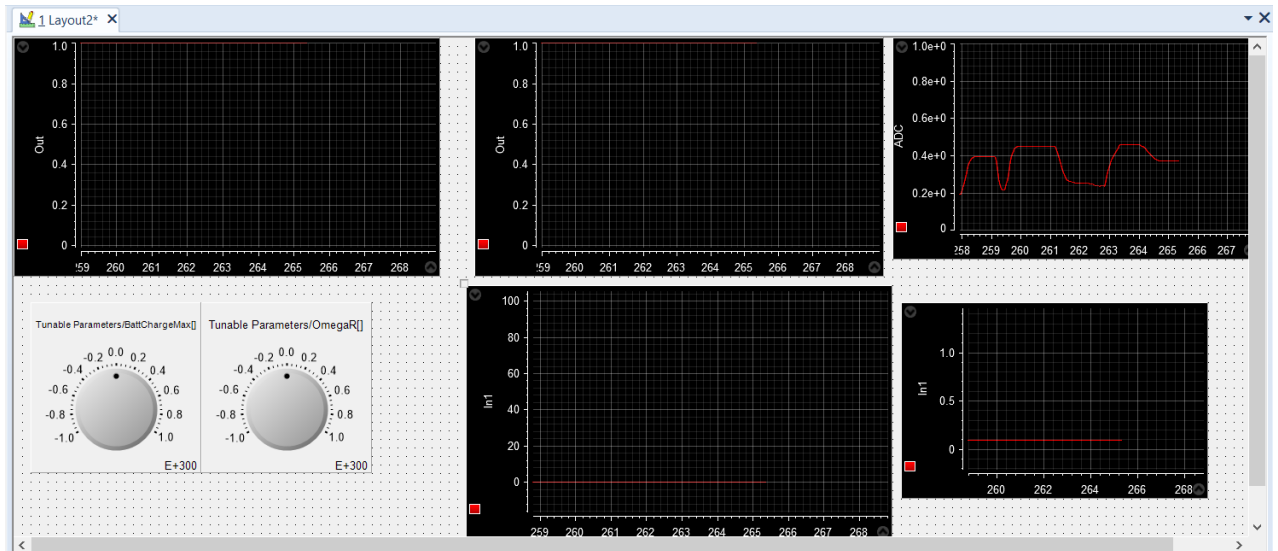


Figure 68 - Third HiL test

As shown in Figure 68, the output keeps being null although the driver asks for a positive output. So, the functionality is verified.

Conclusion

The objective of this thesis is to model the terrain condition of Panda Raid's path and verify the dynamic behaviour of our system on the variation of this environmental signals.

The overall work is performed following the V-Cycle philosophy that allows to model properly the plant and the control logic considering the requirements that an engineer MUST lay out before the model-based work. Indeed, I accomplish my tasks in a way perfectly compliant with the modelling sector thank to the use of suitable software tools.

However, the work performed in this thesis is based on a modelling design with a precise degree of detail. So, several physical process and phenomena, that could improve the analysis on dynamic and performance of our vehicle, are neglected. Moreover, parallel works are still working about this project for what concerning the modelling of Battery Management System and of CAN network inside the Panda system. Therefore, it is necessary to start a proper integration work between my work and these ones. In this case, the V-Cycle strategy workflow is still useful since it is sufficient to update the requirements our software must respect.

Then, this thesis describes the development of a control application up to step 2.2 of the Hybrid V-Cycle.

As obvious future action lines it is to continue the development process until having a functional prototype (point 5).

Bibliography

- <https://x-engineer.org/graduate-engineering/modeling-simulation/model-based-design/essential-aspects-of-the-v-cycle-software-development-process/>
- *THERMAL PREDICTIVE CONTROL FOR HYBRID AND ELECTRIC POWERTRAIN* [Coletta, 2020].
- *Modelling and Torque Split Control Strategy development of a 4WD-Hybrid Vehicle for Rally Application* [Stefano Ruscigno, 2019]
- *Fixing the requirements Mess*, [Christopher Lindquist, 2005].
- *Introduction to functional safety and ISO2626, Model-based software design course*, [Massimo Violante, 2019]
- *Project for the implementation and validation of the Vehicle Management Unit (VMU) code of a hybrid car following the VCycle development strategies described in ISO 26262* [Serrano, 2020].
- *Motor Vehicle Dynamics* [Genta, 2006]
- <https://it.mathworks.com/help/autoblks/ug/explore-the-hybrid-electric-vehicle-p4-reference-application.html>
- *Modellizzazione ed ottimizzazione di un powertrain ibrido* [Bianchini, 2012]

Thanks

To my family, that has always supported and believed in me.

To my best friend Sabrina, that has given me the strength to keep go straight.

To my course's mates Alessandro, Thomas, Luca, Francesca, Paolo, and Giuliana, that accompanied me in this journey and that made me grow up as engineer.

To my relator Stefano Carabelli, that gave me the possibilities to do this work.

To my laboratory colleague Jose, that gave me a hand in the last part of my work.

To my colleagues from the Panda-Raid team, that gave me the possibility to improve my vertical skills.