POLITECNICO DI TORINO

Corso di Laurea Magistrale
in **Ingegneria ELETTRICA**

Tesi di Laurea Magistrale

ELECTRIC POWERTRAIN CONTROL: AN ADAPTABLE SIMULINK MODEL OF TORQUE PATH AND VALIDATION BY MODEL IN THE LOOP

Relatore

Prof. Paolo Guglielmi

Candidato

Domenico Barile

October 2018

Acknowledgement

After six long and intense months, the day finally arrived: it was a period of deep learning, not only on a scientific level, but also on a personal level. Writing this thesis has had a strong impact on my personality. I would like to spend two words of thanks to all the people who supported and helped me during this period.

Firstly, I would like to thank “AVL-Technical Centre Italy” for giving me the opportunity to be a part of this great company. The financial support is gratefully appreciated. Thanks to the colleagues I met during my internship for their fantastic collaboration. In particular I address my supervisors Ing. Francesco Mastrandrea and Ing. Andrea Piccioni, thanking them for being so supportive and encouraging in conducting my research for my thesis.

Special thanks to my university supervisor, Professor Paolo Guglielmi for his valuable advices, leading me to take the right path and complete my thesis.

Finally, I would like to thank my parents for their wise advice and their ability to listen to me. You have always been by my side.

Last but not least, my friends. We have always supported each other, in good times and in bad luck, during the hardships that have characterized our path, in moments of joy and satisfaction to reach the goal.

Thanks to everyone!

Domenico Barile

Turin

September 2018

Abstract

In this thesis, the main features of a powertrain's control for a Battery Electric Vehicle have been studied with a focus on the *Torque's Management*. About the computation of the set-points of torques for the electric machines, a model of a control algorithm for the longitudinal drive has been proposed and tested by *'Model in the Loop'*. This model can be adapted to different powertrain configurations since we can set the number of electric motors and their size. In order to test the algorithm, a basic model of vehicle has been implemented by Simulink as well as simplified models for foundation brakes and electric motors.

The *first chapter* begins with an overview of Battery Electric Vehicle's powertrain and a description of pure electric driving features. The main aspects of *Torque's Management* have been discussed with a focus on the role of the *'Torque Path'* among the systems which interact with the Vehicle Control Unit in the powertrain's control. Following some important issues about software modelling have been mentioned.

In *second chapter* the main functional requirements of a control algorithm for a longitudinal drive have been analysed considering the issues related to the safety; therefore, the corresponding solutions have been described and illustrated in a qualitative manner.

The requirements have been verified in *chapter three* by testing the proposed model with certain inputs generated in a separate section of the Simulink model; various driver request's pattern have been simulated and the results of the simulation have been shown.

Finally, some considerations about further developments and '*Software in the Loop'* phase have been made.

Contents

List of figures

List of tables

1. Introduction

1.1 Electric vehicle

1.1.1 State of the art

Electrification of transport, including standard road vehicle, plays a large role in decarbonisation of the energy system giving the benefit of reducing emissions of local air pollutants in urban areas **[1].** In the last decades many efforts are been made worldwide to improve combustion-engine vehicle efficiency developing new technologies of hybrid-electric powertrain (HEV) pushed by legislation about mandatory emission reduction targets for new passenger cars (PLDV) and light commercial vehicles. That is why, along with strong support from several national governments, many car-makers and automotive company are developing a national value chains in this emerging industry **[2]**. Among possible scenarios of energy technology pathways, battery electric vehicle (BEV) remains a good solution to have *zero emission* and at the same time great performance in terms of maximum torque-speed range but unfortunately, they are limited by a long time of charging as well as a shorter range if compared to a combustion engine vehicle of the same class. As shown by a study, the perceptions of BEV limitations or drawbacks seems to be private-functional in nature **[3]**. Furthermore, the substantial lack of public charging points does not encourage people to overcome the '*range anxiety'*. With battery costs declining rapidly, automakers are rushing to get in the game with their own all-electric models but for people, driving a BEV, it means a raft of changes: plugging in at night instead of hitting the gas station, keeping an eye on a battery meter instead of a fuel gauge, and most importantly, a change in the way they drive.
Despite this, the global stock of electric cars surpassed 3 million vehicles in 2017 after crossing the 1 million thresholds in 2015 and the 2 million mark in 2016. It expanded by 56% compared with 2016. In 2017, China had the largest electric car stock: 40% of the global total **[4].**

In the last decade automotive companies have made significant investments in developing EV. Throughout 2018, news of upcoming electric cars will continue to increase, eliciting more interest from the public. This trend leads to the necessity to improve also electric car marketing and communication strategy in order to compete more effectively with combustion-powered vehicles.
An important issue of a BEV concerns the storage system. In particular, regarding the charging time, considerable efforts are been made to improve it. Firstly, engineers have recognized the advantages given by using a high- voltage DC in an electric car, at least 400V up to 1kV. This serves, not only to reduce the charging time but also to improve the efficiency since losses depend on the current.
In 2018 the SAE the global standard for wireless EV charging will be finalized, demonstrating another improvement of the electric vehicle's technology, making the EV experience more consumer-friendly. That is why wireless power transfer is being embraced by all automakers.
The mobility sector will have also the opportunity to develop new business models based on service and sharing models, and the new uses and services associated with EVs as decentralized energy resources **[5]**. Now, we are in the middle of a global evolution of energy systems, which are becoming cleaner and increasingly decentralized, with energy generated, stored and distributed closer to the final customers, through the acceleration of renewables and storage technologies. At the same time, digitalization allows customers and electricity system operators to control where, when and how electricity is being used, with new business models emerging.
Soon, EVs can be used even as a decentralized energy resource providing new, controllable storage capacity and electricity supply that would be useful for the stability of the energy system. In markets where regulation allows EVs to be used as a source of flexibility, energy players start betting on this vision, with cars working as "batteries on wheels". Obviously, this process is closely related to the available technology and how it much costs.

1.1.2 Electric powertrain overview

It is generally possible to equip any kind of vehicle with an electric powertrain. Basically, there are three types of electric drive vehicles: battery electric vehicles, hybrid electric vehicles (HEVs), and fuel cell electric vehicles (FCEVs). The distinction is made on the base of the type of source which supplies the storage system. BEVs use the electrical energy stored in batteries to power the drive or traction motors.
For pure electric driving, the powertrain consists of three main parts which are: the battery module, the inverter-motor units and the charging module. To manage the powertrain components, it is needed also a vehicle control unit (VCU) which is the highest level of control since it handles all the

Electronic Control Units spread in the vehicle. The battery represents the 'fuel tank' since it provides the energy to propel the car. In a Battery Electric vehicle, energy flow has no one direction, how it happens in a conventional vehicle. This is one of the pros of pure electric driving: we can recover the energy which would be wasted by the friction brakes in the form of heat's production. The voltages required for the electric drive power in passenger and commercial vehicles are several hundred volts higher than the 12/24 V supply. It is clear that high voltages are required to keep the current values within reasonable limits during the energy transfer process. Currently, while battery voltages of up to 400 V are envisaged for passenger car hybrid technology, voltages of up to 850 V are planned for commercial vehicles or high-end sport vehicles. The high-voltage level is between 60 and 1500 V DC, $30 < U \leq 1000$ V AC rms **[6]**. An auxiliary power module allows us to convert the high-voltage into 12-24 V, which supply the low voltage loads.

The battery voltage must be connected and adjusted to the relevant grid voltages to enable charging of electric vehicles. This task is performed by battery chargers mounted into modern electric vehicles. Direct current fast charging is used today to speed up the charging process of electric vehicles. The charging station is directly connected to the vehicle battery. No on-board charger is required for this charging method. However, the charging stations must be able to adapt to the battery's voltage level and the key performance data required for charging (charging state, charging voltage, max. charging current) must be exchanged between the vehicle (battery) and charging station. The vehicle controls the charging process during this communication exchange, while the EV station controls current and voltage supply.

Another important part of a BEV concerns the thermal management. The amount of heat generated by the flowing of high currents, along with environmental factors, can lead possibly to a dangerous battery overheating. If we do not want to shorten the battery life we make sure that the maximum temperature stays absolutely within specific boundaries. Since the battery is the most important part we need to design a thermal system *'ad hoc'* with high performance in terms of lightweight, low cost, easily packaged, and compatible with the location in the vehicle. To comply with these requirements, a common solution consists of using a particular HVAC system (Heating Ventilation Air Conditioning) which is supplied by the high-voltage, rather than by 12 / 24 Volts.

About the auxiliary power module, usually it is a bidirectional converter which basically allows the charging of the low voltage battery. However, it can work also in the opposite direction supplying the high voltage line when the HV battery opens the main contactors. This situation generally corresponds to an emergency, so the low voltage battery can act as a spare reservoir giving the possibility to carry out some operations which would not otherwise done.

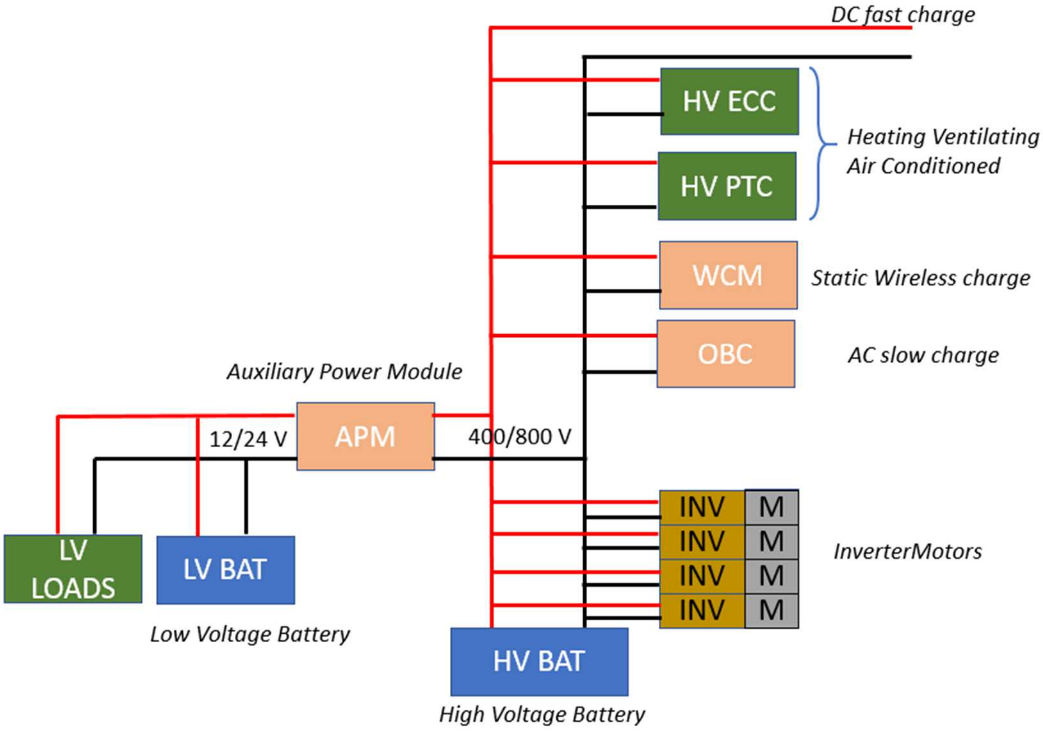


Figure 1 - Electric powertrain architecture: a generic electrical scheme of a BEV.

In *Figure 1,* we have depicted the maximum number of inverter-motor units which can be found in general in a BEV's powertrain. Today, the configuration with four independent drive wheels is not used. The reasons will appear clear later when we will talk about the torque management. In last years, automakers seem moving toward a configuration with three motors, one for the front and a two for the rear axle, but in general, we can have powertrains with only two or one motor (the last is the most used). In the following picture, the most significant cases powertrain configurations are shown.

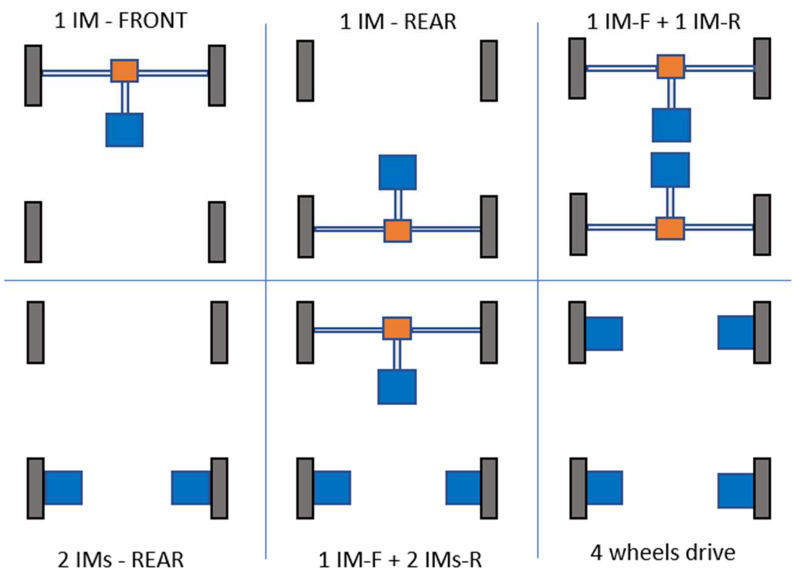


Figure 2 - Possible cases about number of motors and their allocation.

We note that differential is used to transmit the torque from the drive shaft to the wheels when only one motor is mounted on the axle, as it happens in conventional combustion engine vehicles. With regard to the twin-independent motor axle, the transmission of torque is achieved by using fixed or variable gears. In some cases, the motor shaft is directly connected to the wheel without the interposition of a gear-box. We can note that the configuration with two independent motors for the front axle is not depicted. This is because it would be not convenient or economically unviable since we can obtain much more advantages putting the same motors on the rear drivetrain.
The most complex configuration is represented by the *four-drive-wheels* that would allow us to have a total control of vehicle's motion. Even though today this solution seems to be an unpractical way because of the number of problems it carries in terms of safety and not only, it might be a common solution one day. It is clear that to control all the motors we need to have a proper system which manages the distribution of torque in every driving situation to ensure the safety. Actually, there are many systems that act together to control the vehicle's motion. In the next paragraph, we focus on these systems.

1.1.3 Embedded control systems for the automotive space

An automotive embedded system is a distributed *real-time* system with heterogeneous hardware platforms (ECUs) interconnected by different network of signal buses. This kind of system consists of many functionalities implemented in software which has to satisfy different requirements. Thereby, an automotive embedded system consists of a set of inputs (sensors), a set of functionalities (features), and a set of outputs (actuators). The set of functionalities is realized by a set of software components **[7]**.
A *real-time* system must process information and produce a response within a specified time otherwise several consequences, including failure system, can arise. If the consequences are system failure, the system is referred to as a hard-real-time system (e.g. an antilock braking system).
The aim of an embedded system is to react in a right way, as fast as possible, to any unpredictable events using the smallest amount of code and with higher level of reliability.
Automotive embedded systems differ from embedded systems used in other generic application in several important aspects as summarized in *Table 1* **[8]**. Designing real-time system poses significant challenges for engineers because of the amount of environment's complexity in which these systems has to work especially if we consider the safety systems. Today embedded automotive systems are everywhere: hundreds of components of various type and architectures are distributed across the vehicle which communicate with each other by a high-speed network by a specific communication protocol.

particulars	automotive embedded system	generic embedded system
number of sensor	more	less
CPU	multi-core	single-core
development standard	AUTOSAR for automotive application	no generic standard
costraints	high	moderate to low
nature	hard or soft real-time	can be non real-time
performance	strict and high	can be non -strict
reuggdness	high	low
reliability	high	moderate to low
cost of development	high	moderate to low
time of development	long	can be short or long

Table 1 – A brief description about differences between automotive and generic embedded systems

Following there is a list of main applications of the embedded systems in a modern vehicle: air bag, vehicle dynamic, anti- lock brake, adaptive cruise control, climate control, drive by wire, automatic parking, navigation, telematics, display control, back-up collision, tire pressure monitor and many other. One of the first embedded systems was the ABS which monitors the tire in terms of slip to prevent a lock of the wheel bringing the vehicle in a safer condition. This is a hard-real-time system which must carry out the needed action in a small interval time. In the following paragraph, we will deal with the communication network among these systems and how they are being handled.

1.1.4 Communication network architecture

Safety-critical applications, such as those in the powertrain area, need to have a robust as well as a convenient communication system which interconnect subsystem each other. Firstly, we can say that it would be disadvantageous to assign responsibility for bus distribution to just a single bus node because its failure would cause all communication to fail.

A much smarter solution is to decentralize bus access so that each bus node has the right to access the bus. This brings to the need for a serial communication technology especially for reliable data exchange between electronic control's units (ECUs).

So, at the beginning of 80s, Bosch began to develop a serial communication system called CAN (Controller Area Network). Its introduction allowed engineers to simplify by far not only project planning and installation but it also reduces wiring weight and space requirements.

A CAN network consists of a number of nodes which are linked via a unique physical transmission to each control unit (CAN bus) and its maximum data rate is 1 Mbit/s. Today, since the software complexity, has become enormous, makes it necessary to also standardize the ECU infrastructure. In

this way, a standard runtime environment as AUTOSAR provides a reference architecture for the ECU software. Its function is to decouple the network from the software components of the application. In *Figure 3* it is shown the principle CAN architecture. We can distinguish at least three parallel networks with different speed communication (baud rate) which are connected to the 'brain' of the vehicle often called Vehicle Control Unit. The choice to connect an ECU to a network with a certain baud rate depends on the minimum system dynamic requirements. However, the increase of speed corresponds to a reduction of the maximum physical length of the wire. For example, 1 Mbit/s corresponds to a maximum length of 40 m. Each subsystem connected to the network works independently from what happens elsewhere, sending the information on the network.
In the following figure, we have an exemplary network of a vehicle where we can distinguish at least three main branches called P-CAN, E-CAN, B-CAN.

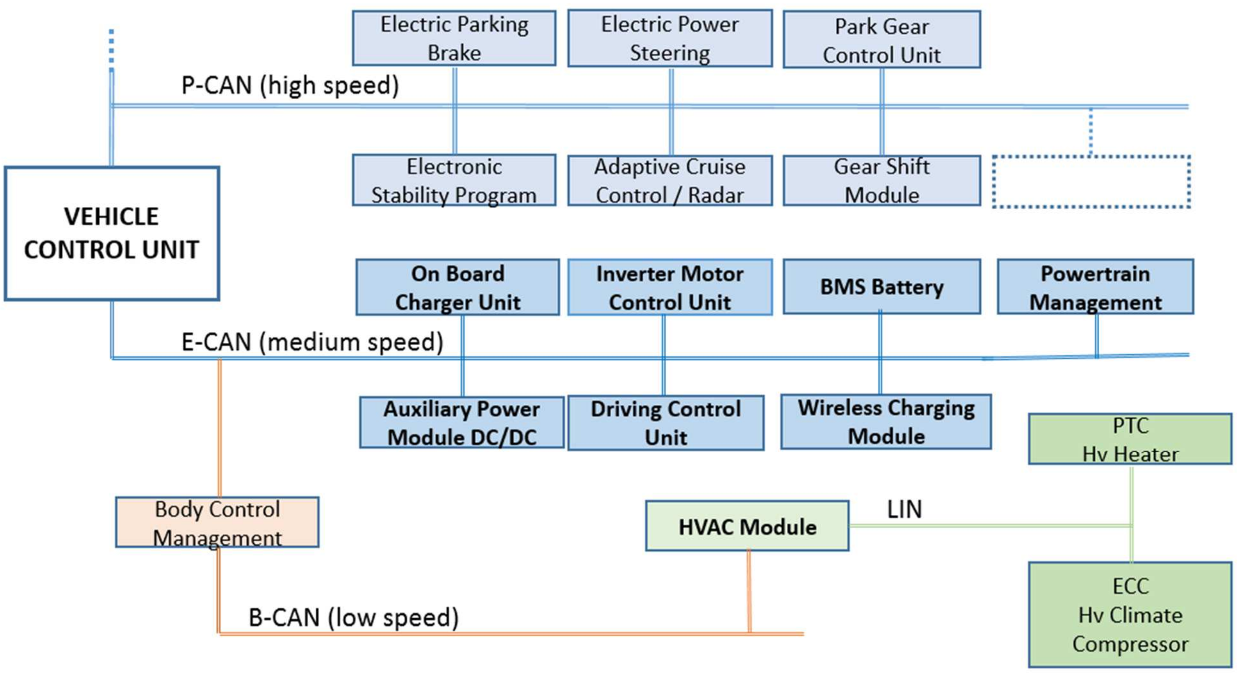


Figure 3 – Possible CAN configuration.

The P-CAN connects CAN-Nodes that implement chassis related features as well as features that can be powertrain related but have a strong connection to the braking system (e.g. brake system, gear shift module, park lock and ACC). The E-CAN connects CAN-Nodes that are part of the HV system while the B-CAN connects CAN-nodes that implements vehicle interior features such as Climate Control Unit.

The LIN bus (Local Interconnect Network) was developed to further reduce the costs of automotive networks. It was designed to complement the available and proven bus systems in areas where lower data quantities must be transmitted. All control systems can be woken up by the ignition signal (this is a hard-wired signal), or by receiving a CAN message defined in the network management specification. ISO regulation defines a multi-master architecture where each node in the CAN network has the right to access the bus without requiring permission and without prior coordination with other CAN nodes.

The vehicle control unit (VCU) ensure the torque coordination, operation and gearshift strategies, high-voltage and 48V coordination, charging control, monitoring, thermal management and much more for electrified and connected powertrains in passenger cars, commercial- and off-highway vehicles. The VCU also ensures fail-operational function for highly automated driving solutions.

Other than these drive-related functions, higher-level versions also support interconnected functions like predictive and automated longitudinal guidance, Advanced Driver Assistance System (ADAS) connection and body controller functions. Consequently, the rise in data traffic led to ever higher bus loads on the CAN buses.

ECU related to the powertrain control system executes the following steps:

- Gathering of hardwired input signals (e.g. acquisition from sensors)
- Plausibility check of these signals
- Calculation of physical values out of the hardwired input
- Provision of this information (including control unit states) which is needed as input for other module functions.

Sending of messages is basically a random activity: messages are not addressed to a particular control unit but when ECUs requiring information they select relevant messages based on the message's identifier (ID). The basic structure of a message is shown in the figure below.

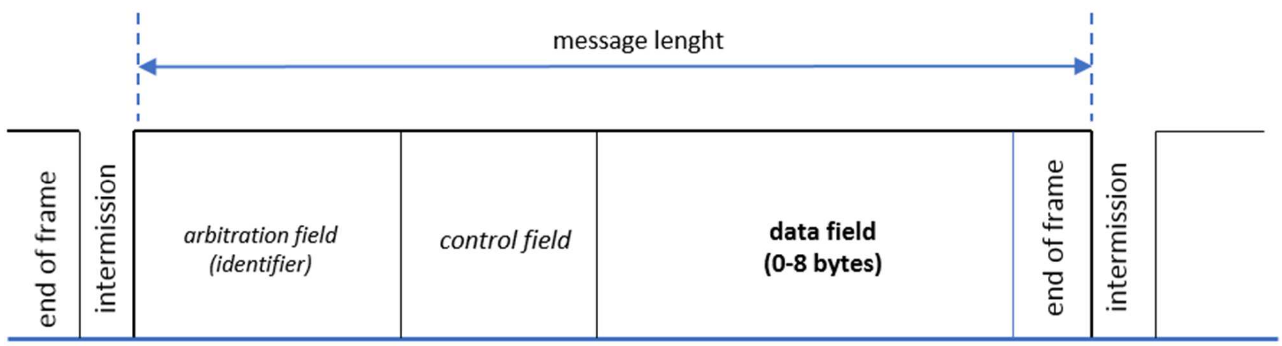


Figure 4 – Structure of a CAN message

Since each ECU have only two wires to transmit or receive the data, usually signals are organized into a matrix often called CAN matrix: it is a table that defines for each signal of a certain ECU the following essential information:

- Signal Name;
- Bit size of the signal (usually from 1 bit up to 32 bit);
- Minimum / maximum value (depending on data type and how many bits are used);
- Bit factor (e.g. 1 bit corresponds to 'bit factor' units of the physical quantity 'x');
- Unit (for example if the signal refers to rotational speed,the unit is 'RPM');
- 'Byte order' and 'Matrix Bit Position' which uniquely defines the position in the CAN matrix.

To have an idea of the its structure, we report an excerpt of the CAN matrix of a Digital Motor Controller for EV application, made by "BRUSA Elektronik".

Signal Name	*Bit Pos*	*Bit Size*	*min. Value*	*max. Value*	*Factor*	*Comment*
DMC_TrqAct	32	16	-327.68Nm	327.67Nm	0.01Nm/bit	actual torque
DMC_SpdAct	48	16	-32768rpm	23767rpm	1rpm/bit	actual motor speed
DMC_TrqMax	0	16	-327.68Nm	327.67Nm	0.01Nm/bit	max positive available torque
DMC_TrqMin	16	16	-327.68Nm	327.67Nm	0.01Nm/bit	min negative available torque
DMC_TrqSlewrate	0	16	0Nm/s	655.35Nm/s	0.01Nm/s/bit	A value of 0 will disable the slewrate
DMC_SpdSlewrate	16	16	0rpm/s	65535rpm/s	1rpm/s/bit	A value of 0 will disable the Slewrate
DMC_MechPwrMaxMot	32	16	0W	262140W	4W/bit	max motoring mechanical power limit
DMC_MechPwrMaxGen	48	16	0W	262140W	4W/bit	max regenarating mechanical power limit
DMC_SpdRq	16	16	-32768rpm	32767rpm	1rpm/bit	command value in speed mode
DMC_TrqRq	32	16	-327.68Nm	327.67Nm	0.01Nm/bit	command value in torque mode

Table 2 – An example from a real CAN matrix of the Digital Motor Control by Brusa Elektronik

This is only a small part of the whole matrix actually consisting of 63-bit positions. We have chosen just ten signals because they are essential information for a powertrain's control algorithm. The last two signals of the table represent the torque command and the speed command sent from VCU and received by the inverter's ECU. The speed command, as well as the relative slew rate, refer to the speed controller included in most of the electric motors for automotive application. The remaining signals are transmitted by the inverter unit to the VCU in order to inform it about some important actual values such as the available positive or negative torque. These feedback data are the base of the motor's torque control.

1.1.5 Electric Machines

Electric motors are already a mass-market product with a wide range of applications. They are produced by well-established manufacturers. However, the requirements for electric motors for vehicles differ from those for some aspects. In particular electric motors for automotive applications are subject to greater weight and packaging restrictions, have higher efficiency needs (due to the limited energy supply), superior power requirements and need a broader speed range.

For industrial applications, the induction motor (or AC asynchronous motor) is by far the most common type of electric motor. This is largely used because of its simple design and low production costs. As compact electronic inverters became available in the 1980s, engineers first started using this type of motor in vehicles and it is still used.

Currently, most car manufacturers are starting to use permanent magnet synchronous motors (PMSMs) or synchronous reluctance motors (SRM) for high-end applications **[9]**.

We know that electric machines can operate on all quadrants of the Torque-Speed diagram. They are: forward Braking, forward motoring, reverse motoring and reverse braking. The latter one is not used in the automotive field because the recovered energy during reverse speed is usually too low to generate a significant charging power.

Desired features of motor drives used for EV are a high-power density, a fast torque response, a high instant power, including constant-torque and constant-power regions, low cost, robustness, high efficiency over the wide speed, a high torque at low speeds for initial acceleration and with reliability.

If compared with DC motor drives, AC motor drives have some advantages such as higher efficiency at load point, less maintenance need, robustness, reliability, higher power density, effective regenerative braking. In the following illustration is shown a rough comparison between PMSM, IM, SRM in order to have a higher overview.

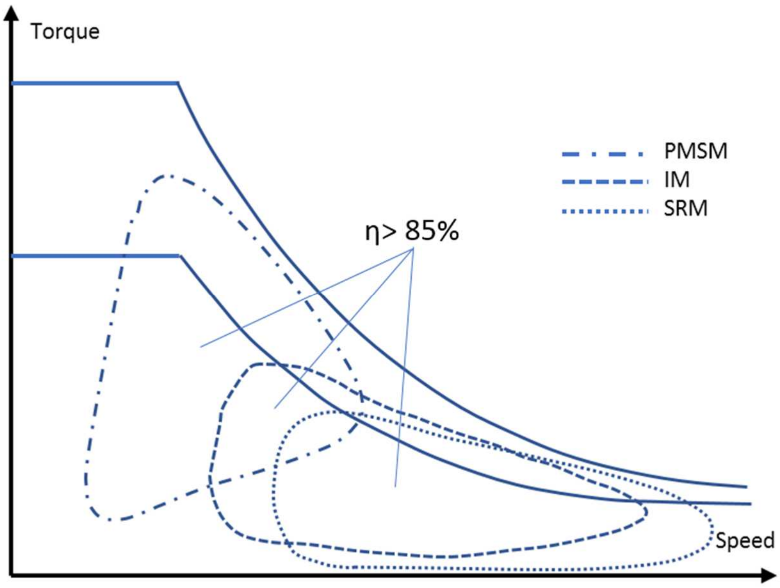


Figure 5 – An exemplary efficiency map's comparison between Permanent Magnet Synchronous Machine (PMSM), Induction Motor (IM) and Synchronous Reluctance Machine (SRM).

One of the biggest differences between electric cars and their conventional counterparts is the possibility to deliver its maximum torque at zero RPM. Furthermore, unlike an internal combustion engine, it does not need a system to disconnect it from the drivetrain to allow it to idle while the vehicle is stopped. In this way, designers of electric cars just pick a gear ratio that provides a good compromise between acceleration and top speed. Since a typical electric motor is capable to sustain 20,000 RPM we have a lot of possible ways to transmit the torque.

Gears have several functions in a conventional car, some of those functions can be skipped in an electric car. For instance, gears to drive in reverse speed are not needed in general because electric motor can be reversed in speed. However, if we have only one motor per axle, differential gears are needed to allow right and left wheels on the same axle to rotate at slightly different speeds as the car drives in a turn.

1.1.6 Drivetrain configurations

In *paragraph 1.1.1*, we have mentioned the possible configurations of an electric powertrain considering only the number of the motors but not the type of transmission.

The development of future vehicle concepts also consists of finding optimized drive trains in the general context of x-EV. A purpose design approach offers a multitude of possible drive trains, which can also offer vehicle dynamic functions, in addition to their main longitudinal dynamic functions, when using more than one electric machine. Because of this, today implemented BEV drive trains differ in both number and package of the electric machines.

Driving dynamic performance describes the capability of the specific drivetrain to integrate functions as torque vectoring, for example. In this case, the application of two electric machines (one per wheel) is mandatory **[10]**.

In the following illustration, we have an overview about possible solutions to connect the motors to the wheels: from the standard configuration with one motor per axle, using a fixed-gear or shiftable transmission, up to the in-wheel-hub motor solution.

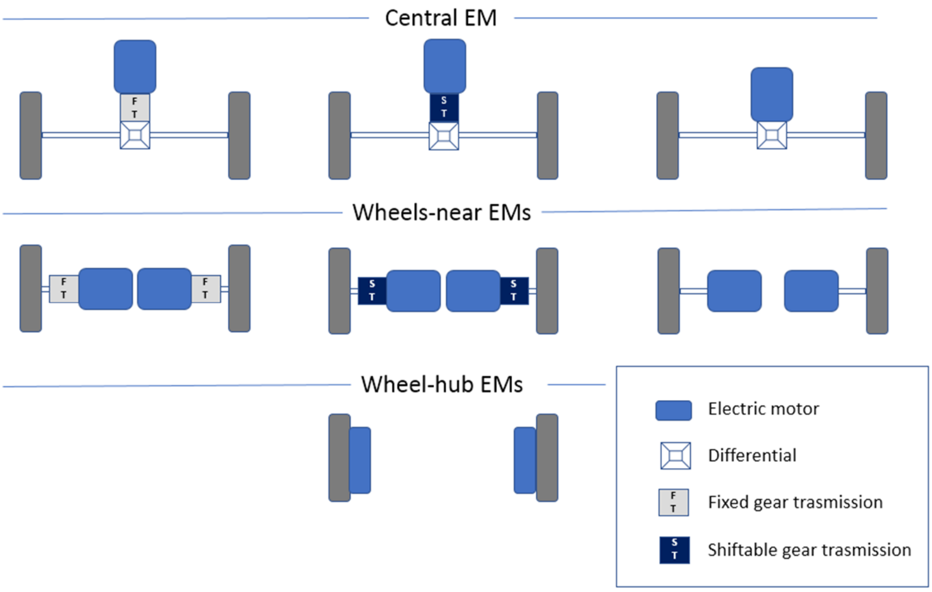


Figure 6 - Possible transmission's types between electric motor and wheels in an electric drivetrain.

It is clear that the best solution has to be found looking at system efficiency and, not least, at cost of production.

About using In-Wheel Hub motors, there are many big pros in terms of efficiency, but a single bigger con has stopped most automakers from using them in production cars. This is because the wheels are one of the components that take the most damage in a car, so it is not the best idea to put a critical and relatively expensive component in them.

A single fixed speed reductor can bring some advantages such as an increase of torque without oversizing the inverter but reduce at same time the maximum rotational speed and the system efficiency because of mechanical loss.

Optimum efficiency of electric machines is typically at medium or high speed and at high load but in reality, the most frequent use in real driving is at various speeds and at relatively low load. Therefore, to overcome these disadvantages, in the last years researchers have proposed various solutions of multi-speed transmission. The simplest is a two speeds transmission: it can increase the effective working area, without oversizing the motor's inverter **[11]**.

With a reductor, the maximum power delivered by motor does not change, less than the reductor's losses, that's why the constant power curve remains the same, but the working area is shifted toward higher values of torques or speed, depending on the speed reduction ratio. With a twin speed transmission, we have a performance's improvement, how we can see also considering the increase of the maximum efficiency area.

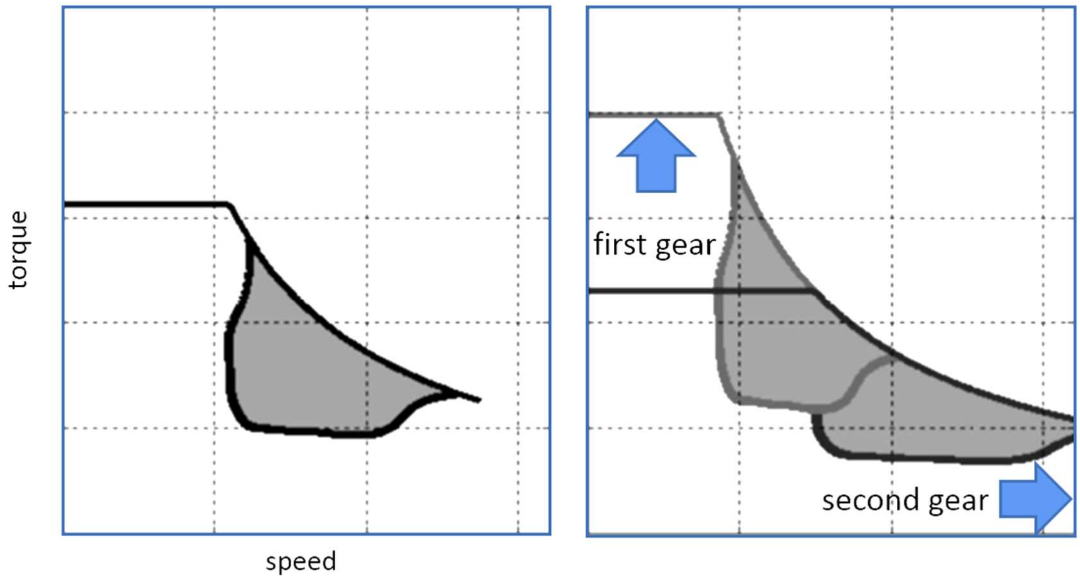


Figure 7 – An exemplary comparison between a motor map (right) and its equivalent (left) by using a twin speed transmission.

The improvement of the efficiency as well as the vehicle's performance open new possibility in the identification of optimal drivetrain by using a multi-speed reductors. In the proposed control's algorithm (named by 'Torque Path') we will consider the actual values of the gear ratio (or speed reduction) whatever it is. Since we can have even a continuous variable transmission (CVT) the value of gear ratio can be also variable in time. So, the gear ratio is an essential information if we want to have a total control of the system.

1.1.7 Functional Safety considerations

Safety is one of the key issues of future automobile development. New functionalities not only in areas such as driver assistance, propulsion, in vehicle dynamics control and active/passive safety systems increasingly touch the domain of system safety engineering. With the trend of increasing technological complexity, software content and mechatronic implementation, there are increasing risks from systematic failures and random hardware failures **[12]**.

Safety practices are becoming more regulated as industries adopt a standardized set of practices for designing and testing products. ISO 26262 addresses the needs for an automotive-specific international standard that focuses on safety critical components.
ISO 26262 covers the whole Product Lifecycle, with particular emphasis on the Development Phase.

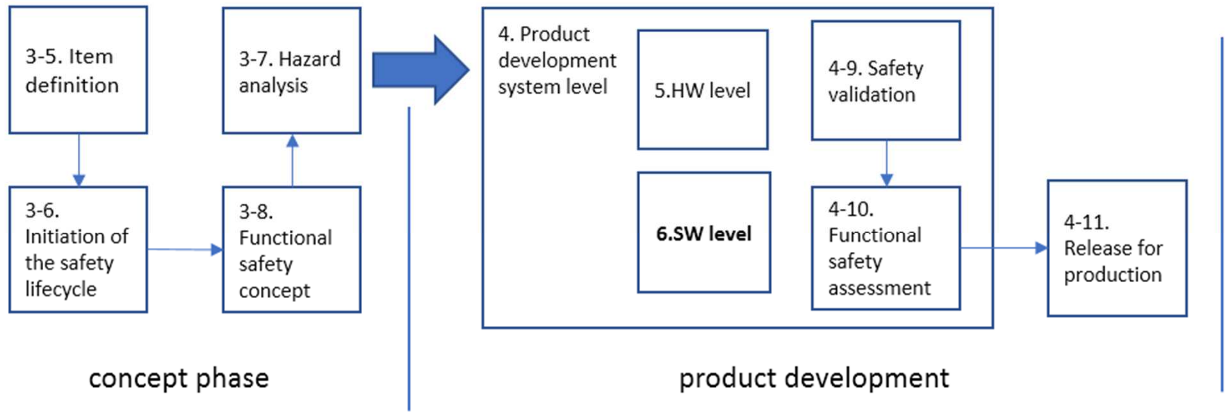


Figure 8 – ISO 26262: steps from item definition to release of production

System safety is achieved through a number of safety measures, which are implemented in a variety of technologies. This standard provides regulations and recommendations throughout the product development process, from conceptual development through decommissioning. It provides a way to assign an acceptable risk level to a system or component and document the overall testing process.
The ASIL is a key component for ISO 26262 compliance. The ASIL is determined at the beginning of the development process. The intended functions of the system are analysed with respect to possible hazards. The ASIL ask the question, "If a failure arises, what will happen to the driver and associated road users? The estimation of this risk, based on a combination of the probability of exposure, the possible controllability by a driver, and the possible outcome's severity if a critical event occurs, leads to the ASIL. The ASIL does not address the technologies used in the system; it is purely focused on the harm to the driver and other road users.
Each safety requirement is assigned an ASIL of A, B, C, or D, with D having the most safety critical processes and strictest testing regulations.

$$Exposure \quad + \quad Controllability \quad + \quad Severity \quad = \quad \mathrm{ASIL}$$

Once the ASIL is determined, ISO 26262 identifies the minimum testing requirements **[13]**.

For example, when multiple motors with individual controllers are used, an unintended yaw moment can occur because of the torque difference between left and right drive wheels. This bring to an unsafe condition which can cause accidents during the drive. Far away that never happens we make sure that this will not happen have to provide the vehicle of many systems which interact simultaneously so that the related control's logic plays a large role especially about the achievement of a greater safe condition. To do this, a large amount of calculations as well as processors and electronics with higher performances are needed. Even the number of control unit for the automotive embedded systems is growing; that means a challenge for engineers to manage all these control units.

1.1.8 V shape development cycle

To handle the technical complexity of a product under development, it is needed to classify requirements according to which part of the product they describe. The complete framework of levels and architecture elements is called Global System Structure. Starting from the vehicle specification, we derive all the requirements which fall into three main categories: chassis-, body- and powertrain-related. We focus about the latter. The V cycle demonstrates the relationship between each phase of the development life cycle and its testing phase.

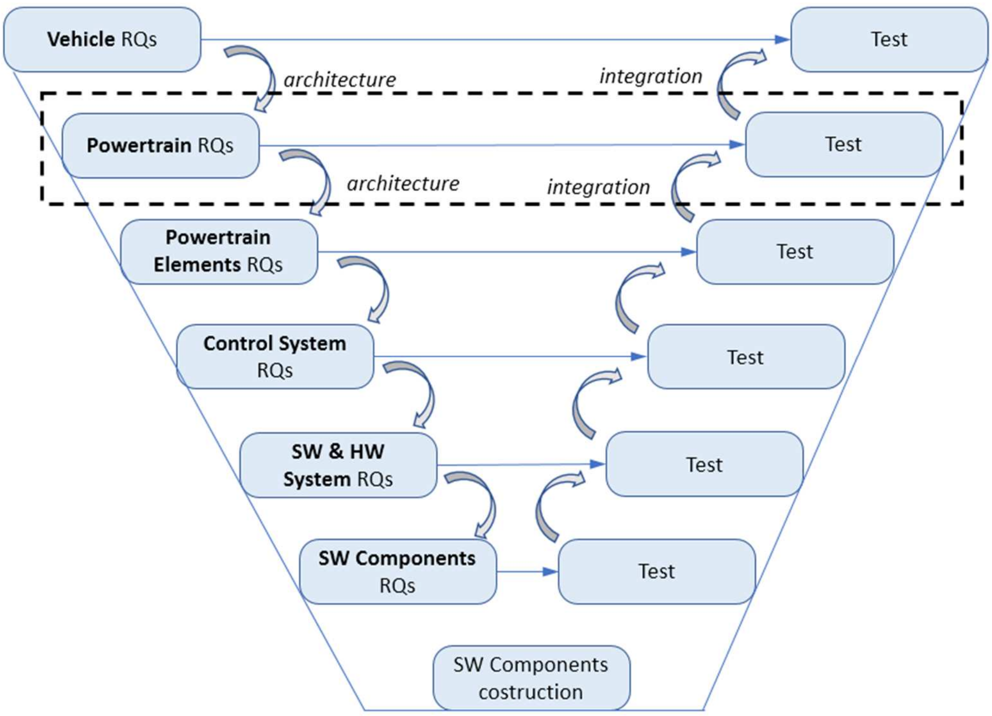


Figure 9 – V shape cycle of a vehicle powertrain's development.

The V-model represents a well-structured method, in which each phase can be implemented by the detailed documentation of the previous phase. The application of this model on the powertrain development is shown in the *figure 9*.

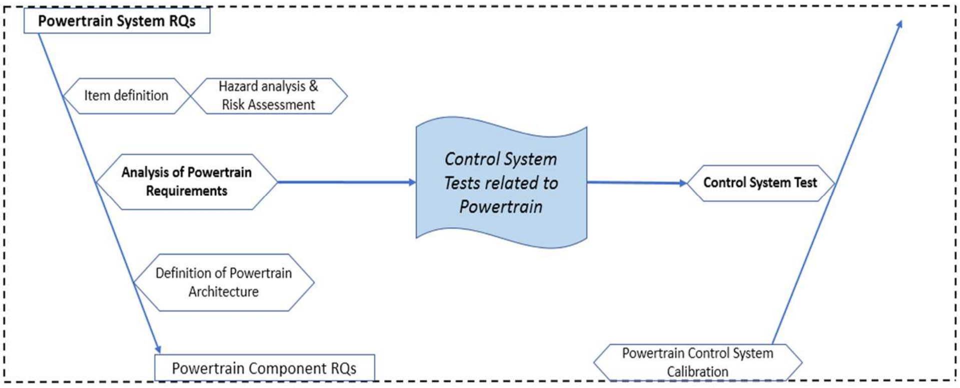


Figure 10 – A focus on the highest level of the V cycle for powertrain's development process

For each sub-system there are a set of requirements. The requirements carry the information, to which level they belong and which sub-system they describe. Requirements should be derived by considering the system or sub-system as a black-box with no interest in how they are to be fulfilled. The *Figure 10* is an illustration of the highest level of the V cycle with increasing degree of detail. Moving along the V cycle we meet the powertrain level and their requirements. These latter concerns also the related control system. In particular, the control logic can be tested before the initiation of the V cycle development. Testing activities, such as testing algorithm, start already at the beginning of the project saving a large amount of project time. There are many functions related to the powertrains which has to be tested. In this thesis we will only focus on the powertrain system level with reference to the torque management which plays an important role in the achievement of the vehicle global safety during the drive. The following chapter will give us an overview about what we mean with "Torque management" and its related issues.

1.2 Electric powertrain features

1.2.1 Torque management overview

Usually a conventional vehicle has only one engine and the power can be regulated by a throttle that limits the amount of entering fuel. In a BEV more than one motor can be mounted opening new possibilities to distribute the torque on the drive wheels. To do this, an algorithm which manage the computation of the set point of torques for the electric motors is needed. One of the issues about motion's control is how to convert the driver inputs of the accelerator pedal as well as the brake pedal in a set of requested values of torques. Then these values will be sent to each electric machine via a CAN. This operation has to face all the limits that every driving situation imposes, especially during an emergency condition. This can mean not negligible efforts if we consider the maximum complexity from powertrain's point of view represented by four independent motors configuration.

Recently some cars-makers are orienting their efforts towards the developing of an electric powertrain with three electric motors: one of these is mounted on front axle by a differential and provides most of the tractive effort while the other two independent motors (possibly smaller than the first) are mounted on rear axle. That arrangement allows for what automotive engineers call "torque vectoring". Reduced to its essence, torque vectoring means that each motor gets all the power it can use at any moment in time. Its focus is on vehicle dynamics when the road gets twisty or when the vehicle negotiates a curve. Currently, Torque Vectoring (TV) is employed in some cars (combustion or hybrid electric engine) to give to vehicles a sporting touch, improving driveability and thus the safety. This is obtained by an embedded system including a lot of sensors and actuator such as active differentials. Torque vectoring is used to improve the manoeuvrability for example reducing significant understeer behaviour at high speed cornering. The vehicle stability is achieved by correcting vehicle's yaw rate by giving to the car a calculated real-time yaw moment, starting from several variables such as steer angle, lateral acceleration and vehicle velocity. This is a critical real-time embedded system that for every driving situation, calculates and tries to get a certain value of yaw rate corresponding to the greatest safe condition. However, delivering the calculated optimal set of torques by mechanical active differential along with a single tire's brake control bring acceptable results but not the optimal. In contrast with combustion engine, electric motors do not need an idle speed to work, which means there is no need of a clutch. This way it is possible to control the torque in every wheel, obtaining a better dynamic behaviour for the vehicle in average driving, acting as a differential, as well as in emergency situations (traction control, stability control, etc.) **[14]**.

However, high degrees of freedom mean a lot of safety requirements and also high complexity about the control algorithm. To understand how to manage this complexity, it would be useful to analyse which systems come into play in the management of torque. The following scheme give us an overview about these systems and their interconnections. It is an extremely simplified illustration which shows us the powertrain management includes the function related to the computation of the torque set points but it does control directly the motors.

The Vehicle Control Unit receives torque's set point from powertrain management along with other systems and outputs the right commands to the motors according to the implemented safety control logic. Obviously, the different units communicate with the VCU sending and receiving signals via a CAN network (not shown).

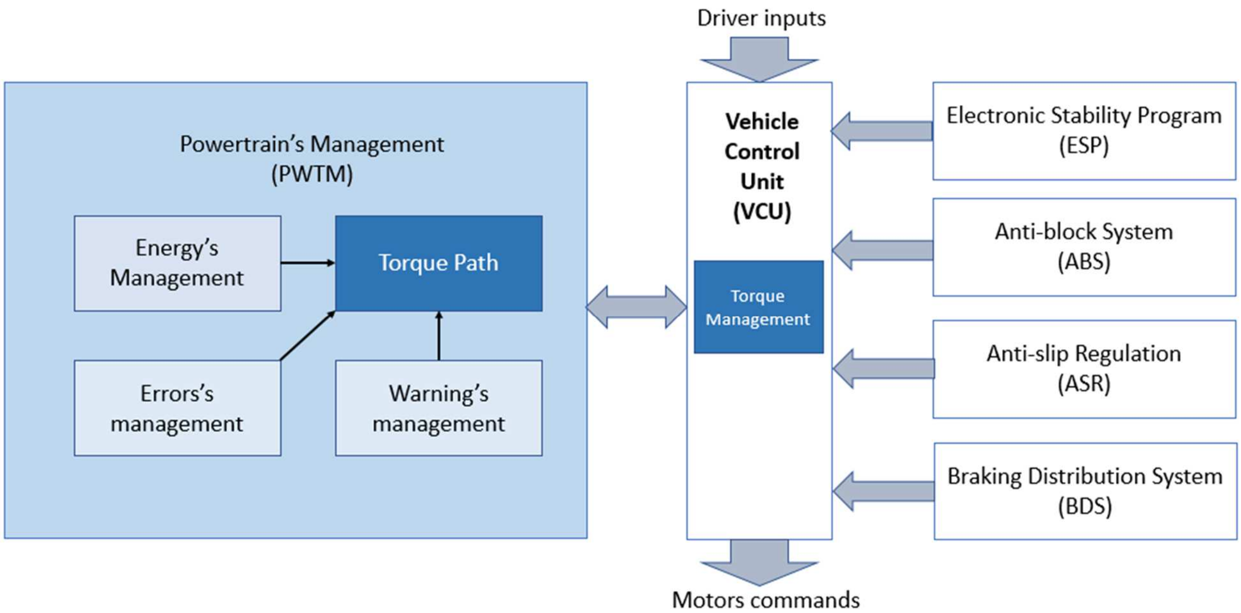


Figure 11 – A simplified scheme of interconnection between systems related to the torque management and the VCU

Referring to the picture, all these systems are involved in the highest level of control to which we will refer generically as 'Torque Management'. Most of these embedded systems are mandatory for all new vehicle both in Europe and in other countries because they prevent accidents and at the same time improve the driveability. In this thesis, we do not deal with these systems, but it is necessary an overview of them for a better understanding. Following, a quick description of these blocks will be done, keeping in the mind that they can take different names depending on the manufacturer, even though the function can be the same.

Anti-blocking system, better known by the abbreviation ABS, was one of the first safety system used to avoid the locking of the wheels of the vehicles, ensuring the driveability during hard braking. Furthermore, modern ABS apply individual brake pressure to all four wheels through a control system of hub-mounted sensors and a dedicated micro-controller, so it can be employed also to adjust the vehicle yaw rate at any moment by other stability programs such as ESP, ASR or BDS.

Electronic stability program (ESP) is the generic term recognised by many worldwide authorities even if vehicle manufacturers may use a variety of different trade names. It deals with the driveability, acting only when it detects a probable loss of steering control, for example when the vehicle is not going where the driver is steering. Its operation is founded on the ABS just as ASR (Anti Slip Regulation) and BDS (Braking Distribution System) that respectively act during a strong acceleration and a hard deceleration. They may be incorporated in other control units too. Their purpose is to exploit the road grip, depending on the weight transfer, to get the largest deliverable traction or braking force without slipping. For instance, if we want shortly to accelerate the vehicle, the load on the rear wheels are larger than the load on the front wheels. Thereby, rear wheels have more adherence to the road and they can tolerate higher values of torques. On the contrary, during a strong deceleration, we have the highest grip on the front wheels owing by a greater load on the front axle.

Powertrain's management represents the basic control of a vehicle. Many features there are in, one of this is the often-called "Energy management". It is related to the management of the battery and its main task is to regulate energy's flow in both directions. For instance, if the battery is fully charged, the Energy Management imposes that battery cannot be further charged, to avoid an undesirable event such as an overcharging. In contrast, if the battery charge is too low, the control algorithm shall decrease the output power to get the maximum recuperation rate by electric braking.

Powertrain management also deals with the warnings and errors which can arise because of a malfunction of a system. It can concern sensors, the supply system or the measurement system.

Inside this powertrain management there is our torque path which can be considered as a function of the powertrain management.

In this work, '*Torque Path*' refers only to the computation of requested torques for a *longitudinal drive* in normal operation conditions, not in a dangerous situation or during an emergency. So, if the vehicle's stability, in terms of slip ratio and yaw rate, is compromised by a motor fault or an emergency braking, the VCU should be able to ensure the safety, getting from the CAN bus the right commands which came from the various ECUs, and implementing the proper strategy to bring the vehicle in a safer condition. This may require great efforts in terms of the amount of calculations.

The model of Torque Path which is considered in this work only deals with the management of the torque for the longitudinal model of the vehicle. Furthermore, to simulate the vehicle response an ideal model of tire is used. Even though we are far from a realistic vehicle behaviour, for the purpose of the proposed control model, a simple vehicle model represents a useful way to get a fast and plausible result. It cannot be used if we want to simulate a turning because we do not deal with the stability and dynamic response. Even if we neglect the lateral behaviour this can lead to erroneous results. Turning represents a challenge if we want to control two independent steering drive wheels. The reason why this kind of drivetrain configuration is not used can be found in the growth of both complexity and the risks. The less expensive way to provide traction to a vehicle still is the configuration with one motor on the front or rear axle using the differential gear. However new future technologies can make possible the using of independent drive wheels, so it is right to have an overview of the negotiation of a curve by an electric car with a twin-motor drivetrain. The so-called electronic differential is a replication of the mechanical differential which allows delivering to the wheels the exact values of torque needed to steer the vehicle in a right way. We want also to underline that 'turn' belongs to the vehicle longitudinal model even if it is closely related to the lateral motion. If the vehicle velocity is enough low and the road friction is very high, the lateral movement can be neglected.

1.2.2 Turning

Electric vehicles with independent motors on a single drivetrain have the advantages of simple transmission mechanism, independent and precise control of the driving wheel torque; however, removing transmission and differential, they consequentially need of a system that reproduces the same behaviour when the vehicle is turning **[15]**.

All combustion engine vehicles (but also electric cars) use a differential to solve the problem of turning: a relatively simple mechanical system allows the outer drive wheel to rotate faster than the inner drive wheel during a turn. The average of the rotational speed of the two driving wheels equals the input rotational speed of the drive shaft. Since the torques at wheels do not change, the total power delivered to the wheels is split according to the rotational wheel's speed ratio.

The electronic differential is an advancement in electric vehicle technology and this is still under study by automotive engineering. In the available literature, some proposed solution consists of using speed controller to ensure that the rotational speed of the motor is always equal to the calculated speed corresponding to a certain value of steering angle **[16]**. However, using a speed controller during a turn poses some problems if we want to have a direct control of torque by the accelerator pedal.

In any case, when the vehicle negotiates a curve, drive wheels have to deliver different values of torques in order to have different rotational wheels speeds according to the well-known 'Ackerman steering model. Ackermann steering geometry is a geometric arrangement of linkages in the steering of a car designed to solve the problem of wheels on the inside and outside of a turn needing to trace out circles of different radii.

The requested rotational wheels speeds depend on many variables such as the turning radius: the latter one is a function of the geometry of the car as well as the entering velocity when you start to turn. Following, it is shown what happens in terms of motors working points changing. If the vehicle is going straight ahead, then motors working points are coincident (P_1). While, if the steering angle is not equal to zero, different values of wheels speeds are needed to avoid a wheel slip, thereby working points unavoidably change (P_2 , P_3). We can note that the delta-speed values are equal between each other while the delta-torque depends on the load's characteristic.

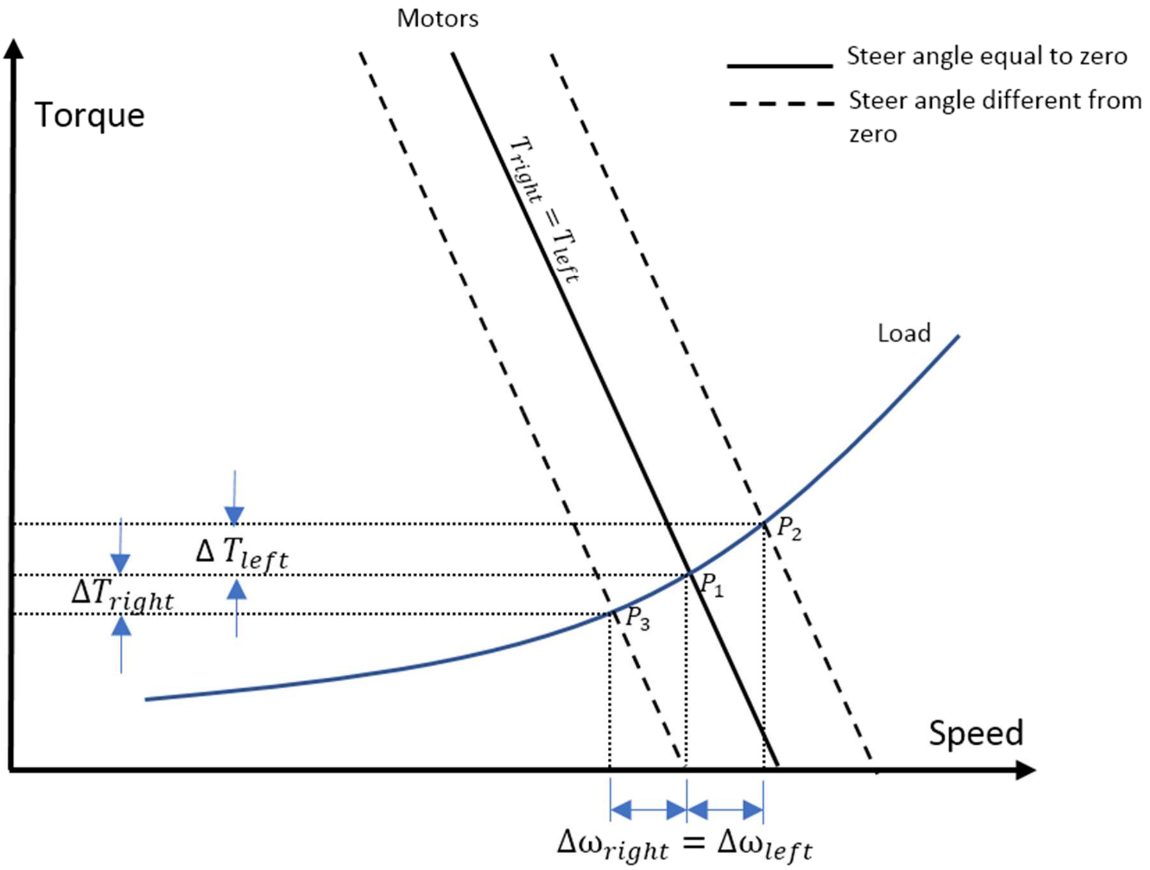


Figure 12 – Change of load points for a twin motor axle during a turn.

The computation of the two set-points of torque is carried out by splitting the problem into two parts. The basic idea consists of the computation of the torque's references for the longitudinal drive aside from the delta torques on the wheels needed to steer the vehicle.

ΔT_{right} and ΔT_{left} are the corrections to be performed by motors on the average torque set-points. We could suppose that these values are algebraically added to the output torques deriving from the block named as *Torque Path* which only produces the reference's value for a straight driving direction.

Furthermore, we have also to deduct from the actual feedback torques these corrections as shown in *Figure 13* in order to make our model insensitive from the steering angle.

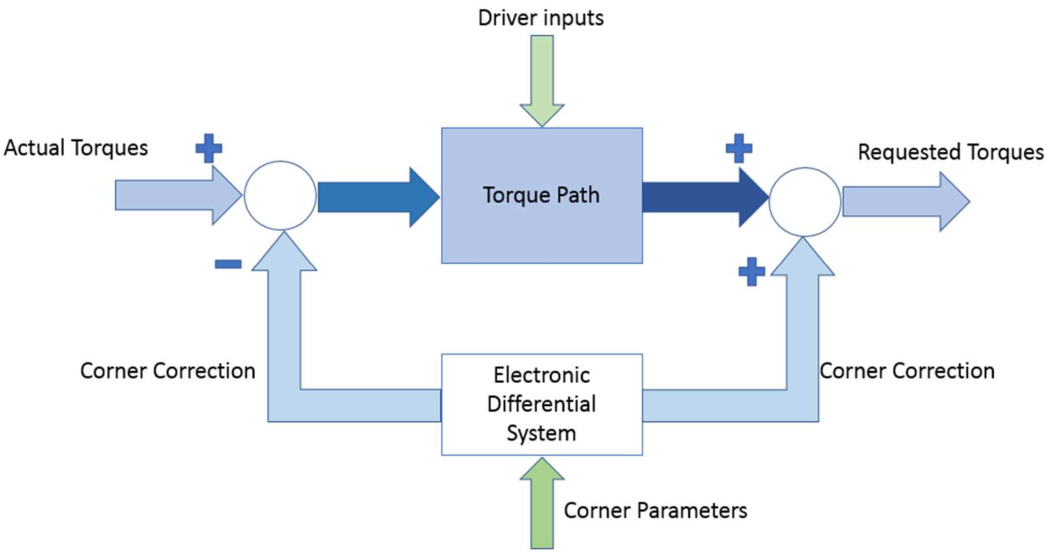


Figure 13 – A possible way to make Torque Path independent from the angle steering variation.

By this solution, we suppose to execute the Torque Path's control algorithm even during a turn. This hypothesis enables the separation of the concerns about the computation of the set-points of torques for a longitudinal drive allowing us to focus on the main control task which we have to carry out.

1.2.3 Electric Braking Recuperation

One of the most important advantage of EVs is the possibility to recover the vehicle's kinetic energy converting it back to electrical energy during braking (deceleration or downhill running). The converted electrical energy is stored in a storage device such as battery allowing to increase the driving range; thereby the powertrain must be designed considering the delivery of power as well as for the storage of it. Using regenerative braking to slow down or stop the vehicle, poses some problems concerning the cooperation of the electric machines with the foundation braking. In general, EVs are equipped with the regenerative-hydraulic hybrid braking system. Whenever the regenerative braking torque is insufficient to offer the same deceleration rate as available in conventional vehicles, the hydraulic braking torque is applied. The physics of regenerative braking systems in terms of stability is no different to hydraulic friction braking also when both kind of braking act simultaneously. A substantial difference of regenerative braking systems to friction braking systems is that while friction braking system can apply constant braking torque at all speeds, the available torque in a regenerative braking system is related to the state of charge of the battery, and the speed of the vehicle.

There are two legislative documents within the European Union with regard to braking systems in passenger vehicles: European Directive 71/320/EEC and ECE Regulation 13H: we will focus on the latter. This regulation defines two categories of regenerative braking systems. 'Category A' are systems which do not form part of the service braking system while Category B' systems which may be actuated concurrently with the foundation brakes.

Regulation 13H specifies also that '*the braking rate must remain related to the driver's braking demand'*. This means that for a giving driving situation and a certain brake pedal position the vehicle's response, in terms of deceleration rate and drive feel, must to be the same every time we repeat the experiment regardless from the availability of electric braking torque. Thus, it is necessary to modulate hydraulic braking in such a way that for a given constant pedal input, the sum of regenerative and friction torque has to be essentially constant. This is a significant constraint which has implications in terms of not only control systems, but also componentry **[17]**.

The blending of braking brings some issues about the choice of the best braking strategy but also the interaction with ABS. Even though the aim is recovery kinetic energy as much as possible the priority is always preventing a slipping event. Hence, the current standard practice, when an event of slip is detected, regenerative brake is simply disabled, then the ABS shortly intervenes using only hydraulic braking **[18]**.

There are two kind of recuperation strategy: actuating electric braking concurrently with the foundation brake (simultaneous, or parallel phasing), or before the foundation brake (sequential, or series phasing). In figure below two ways to relate the pedal brake to the request of electric and friction braking forces for parallel and series phasing strategies are shown.

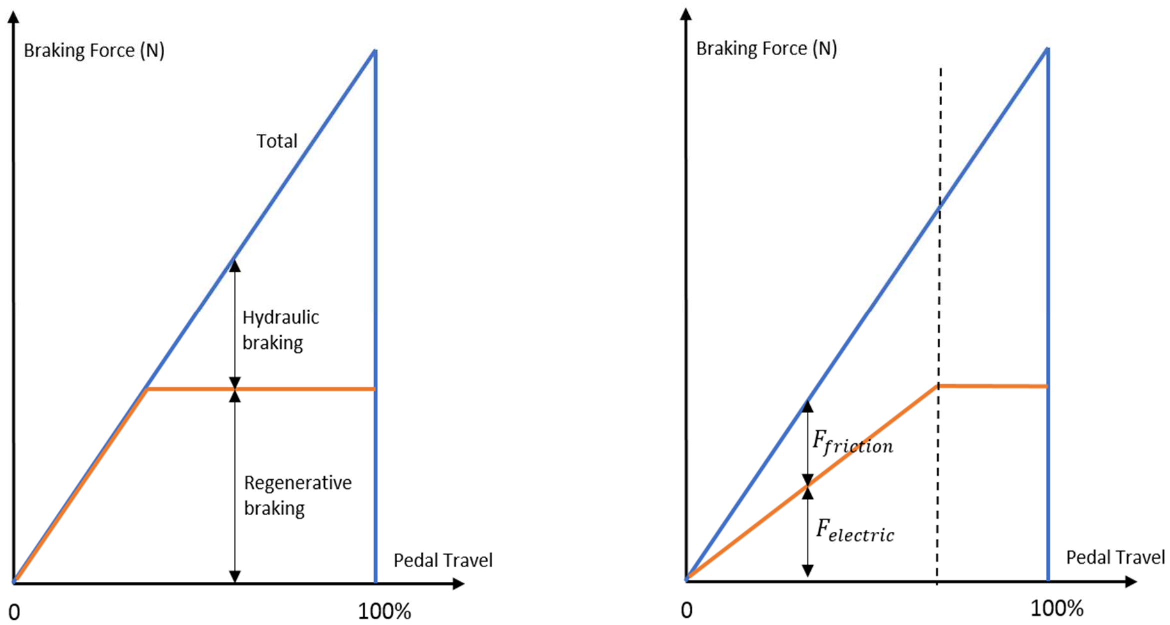


Figure 14 – Possible split strategies of total braking demand: series (left) and parallel (right) phasing.

These two strategies actually can be handled as if they are only one. We can distinguish two areas on the diagrams: the initial part where the regenerative braking is increasing up to its maximum value and the remaining part with constant regenerative braking.
Now, focusing on the first area, we can define a coefficient given by the following ratio:

$$k_{regen} = \frac{F_{electric}}{F_{electric} + F_{friction}}$$

Where k_{regen} stands for the contribution of the regenerative braking to the total braking request before the saturation of electric braking recuperation to its maximum value.
In this way, if $k_{regen} = 1$ then we have a sequential braking strategy, otherwise we have a parallel strategy. This option will be implemented in the proposed control algorithm.
Furthermore, intrinsic variations in the torque output of the electrical regenerative braking system (e.g. as a result of changes in the electric state of charge in the traction batteries, or electric motor torque characteristic) shall be automatically compensated by appropriate variation in the phasing relationship.

1.2.4 Optimal braking distribution overview

Regarding the basic theory of braking vehicle, it is right to spend few words about the optimal distribution of braking torques between front and rear axle. Theoretically, for best braking performance, the braking force distribution should be proportional to the corresponding vertical loads. On the longitudinal direction, the relative motion between tyre and road will turn from a mix of sliding and rolling to pure sliding, if the force applied on the wheel by calliper exceeds the maximum available friction force between tyres and ground. This phenomenon is also known as ‘wheel lock’ **[19]**. Therefore, to avoid the tires of one axle lock before the tires of the other axle it is necessary to modulate the total braking request according to the vehicle’s weight transfer. During braking, the vertical load (weight) on the front tires is larger than the load on the rear tires. If we consider the vehicle as rigid body, it is possible to calculate the ideal distribution of braking request on the two axles. This is shown in *Figure 15*: for a given deceleration (expressed as a fraction of gravity’s acceleration) the requested total force is the sum of the braking forces given by the intersection between the curve (blue) and the line corresponding to the desired deceleration.
Since it not realistic to follow the ideal distribution, regulation 13H recognizes a working area defining a curve (red) which forms a region along with the theoretical distribution curve.

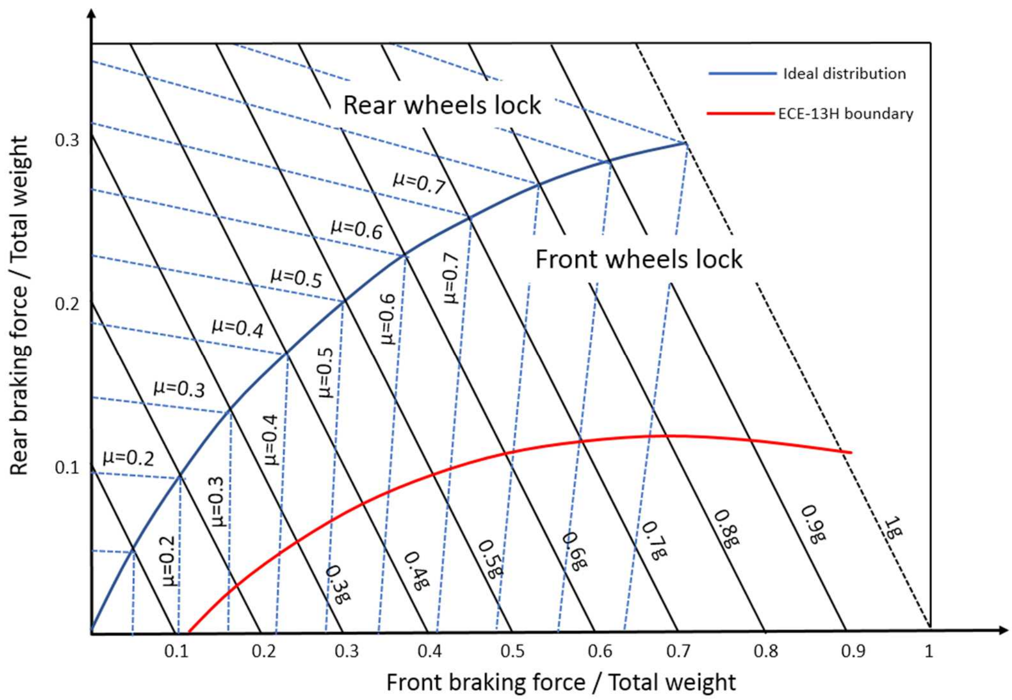


Figure 15 – Optimal braking distribution curve (blue) vs. ECE regulation 13H (Red). The latter one is the limit for the front-bias braking.

In that area, the braking is front-biased (for safety reasons) and in case of braking, a lot of pathways can be travelled across the diagram, depending on the desired braking strategy. It is not allowed to go out of this safety area: this is the function of the braking distribution system (BDS). We can make similar considerations about the acceleration phase. In this case we talk about a traction control system (often called TCS). It is important to say that in the proposed control algorithm the distribution of forces, which is the result of a complex real-time computation, is not dealt with. Basically, we will distribute the torques considering the number of electric motors and their size (in terms of power).

1.2.5 Coasting and creeping

One of the most advantages of a full electric powertrain is the possibility to exploit all the four quadrants of the torque-speed diagram, in contrast with a combustion engine which works only in one of these quadrants and with limitation about the minimum rotational speed.

In a conventional vehicle when we talk about 'coasting' we generally refer to the deceleration of the motor when no power is requested, and the transmission's gear is engaged. In a combustion engine vehicle, this happens when we release accelerator without pressing the pedal brake: thereby a further resistance force is given by the motor which slightly decelerates the vehicle. With an electric powertrain this deceleration can be avoided (sailing), or we can simulate the same behaviour by applying a negative torque when the accelerator and brake are not actuated, having the benefit of a discrete energy recuperation. Braking recuperation is perceived by users as a core vehicle behaviour.

To get the maximum benefit out of driving an electric car, the accelerator should control both the speeding up and the slowing down. Pressing the pedal makes the car go, as usual, but lifting the foot off the pedal makes the car slow down by a specific control algorithm, recovering energy.
When a combustion engine vehicle is slowing down, and so vehicle velocity is decreasing, it reaches a minimum speed larger than zero otherwise the engine would turn off.
This behaviour is often called 'Creep' and it can be reproduced in an electric car by giving positive torque when speed goes below a selectable threshold without the need to press the accelerator pedal. Furthermore, the driver could even set the vehicle speed between 0 km/h and the maximum vehicle creep. In the following illustration all possible electric driving features and their corresponding driving manoeuvres, are summarized for a better understanding.

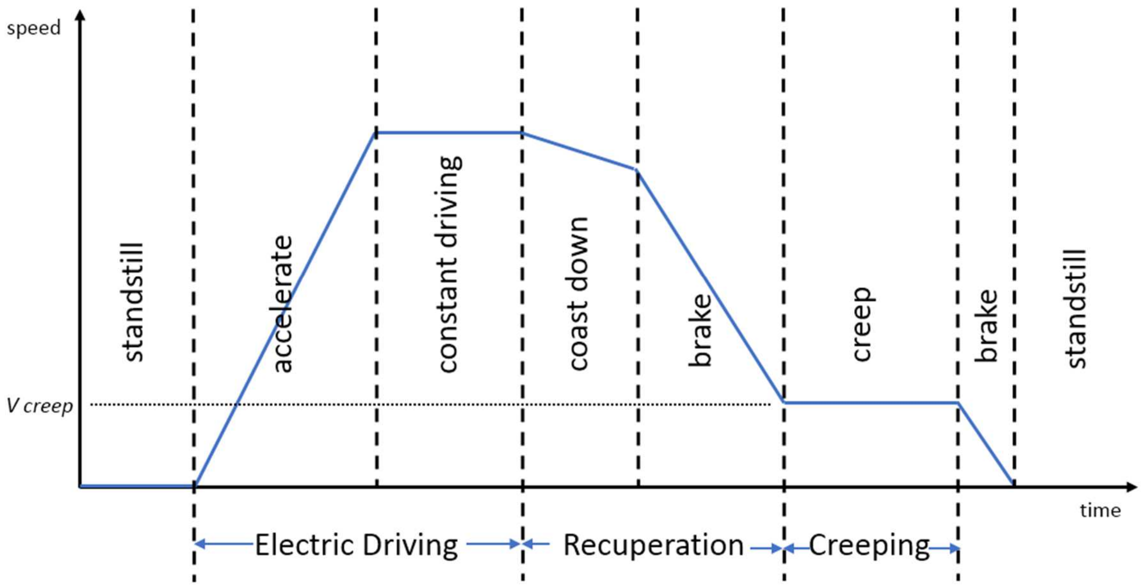


Figure 16 – An example of the electric driving features and their corresponding manoeuvres.

In a conventional car, brake pads clamp onto a metal disc, converting the kinetic energy of a speeding car into wasted heat. But when electric cars slow down, the electric motor can run as a generator, recovering some of that previously wasted energy to top up the battery. Depending on how much regeneration the software engineers allow when designing the car, the force can be powerful enough to slow the car most of the way to zero, which means drivers only need to use the brake pedal to come to a full stop. Stopping distances will be shorter too, as the car will start slowing down as soon as the driver begins to lift off the accelerator, rather than when he moves his foot to another pedal.
The motor's torque's curve is a function of the rotor's rotational speed and it indicates the maximum deliverable torque by the motor: it is conventionally placed in the first quadrant and generally it represents also the generator torque's curve because it is symmetrically placed in the fourth quadrant.

Owing to reversibility, the electric machine's behaviour does not depend on the speed's direction, therefore the same considerations are valid also for the second and third quadrants, corresponding to a negative velocity. However, electric creeping and coasting are allowed only with positive speed, that is the normal drive, both for safety reasons and because it is not much convenient in terms of energy recovery.

For *coasting*, we can have various recuperation strategy corresponding to different levels of recuperation's strategies. The basic idea is to relate the accelerator position to the request of braking torque which is extrapolated instantaneously by the corresponding functions. For instance, when we completely release the accelerator's pedal, the corresponding requested torque is the maximum allowed coasting braking torque depending on the chosen curve.

Thus, there is a threshold for the accelerator position, below which coasting is active. The 'zero position' is a calibrated value corresponding to no request of torque, above which positive torque is provided to the vehicle.

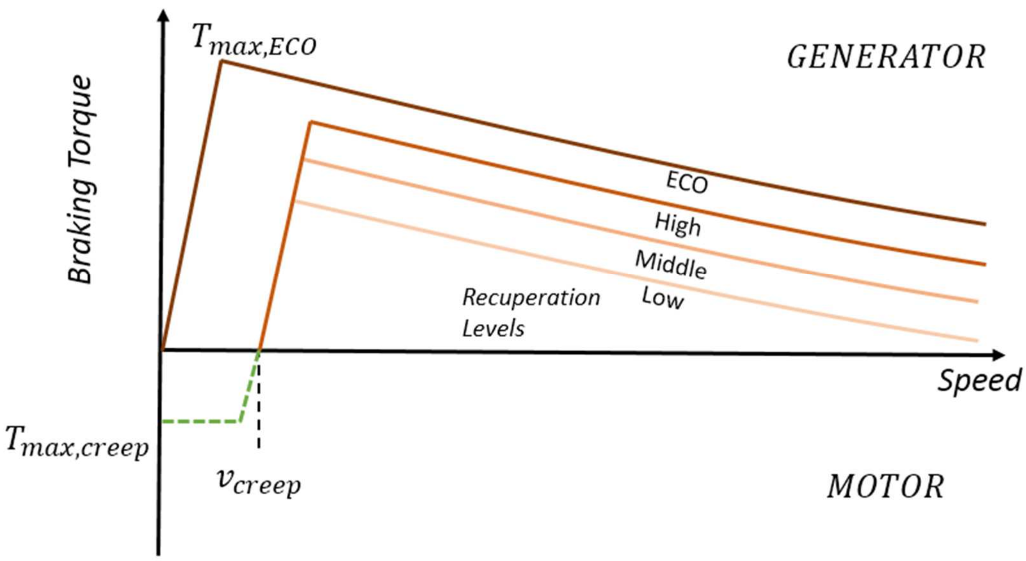


Figure 17 – Qualitative curves for different levels of coasting recuperation.

In the graph, some qualitative curves about coasting and creeping are shown, among which there are three similar curves (high, middle, and low): they impose different value of braking torque depending on the choice of the driver. The 'coasting' functions are made up in such a way that high speed corresponds to low values of braking torque and vice versa, in order to get possibly a constant power during deceleration. While the velocity is decreasing, the requested torque increasing until the vehicle reaches a certain value of speed (called v_{creep}) where the torque becomes positive (motoring) because of the activation of 'creeping' function.

So, if the vehicle is slowing down and both the accelerator and brake pedals are not pressed, the torque delivered by motor follows one of these curves while if it reaches v_{creep} , torque changes sign in order to maintain the established speed corresponding to the intersection between the axis of the abscissas and the creeping-coasting functions.
Creeping can be also deactivated if we set the ECO mode to get the maximum benefit about recuperation energy. This drive mode is often called 'one pedal drive' allowing drivers to obtain a complete stop without pressing the brake pedal.
An electric powertrain could support also a pure electric hill hold function: this means the vehicle's roll back is avoided not by applying brake torque (hydraulic brakes) but by applying positive propulsion torque to the axles. To do this, electric machines need to have a speed controller. In that case we talk about 'Electric Standstill' (not dealt with in this work).

1.2.6 XBW technology

The transition from mechanical or hydraulic systems to electronically controlled systems has been under study for almost a decade. Historically, the 'throttle pedal' operates by a direct mechanical linkage. However, modern combustion engines are equipped by so called drive-by-wire systems because there are sensors which monitor the driver request and in response, an embedded system controls the flow of fuel and air.
X-by-wire technology (x is the generic component driven by wire) replaces most of traditional mechanical control systems with electronic control systems using electromechanical actuators and human-machine interfaces such as accelerator pedal (commonly called throttle-by-wire), brake pedal, steer, gear shift, park lock and so many others **[20]**.
Electronic throttle control (accelerator) is one of the first major successes for an XBW system. Basically, the pedal translates foot pressure into a variable Voltage (0-5 V) to let the controller know how fast you want to go so there is no more direct mechanical linkage with the powertrain.
Electrohydraulic brakes are the first step toward full by-wire technology for braking. In these systems, traditional hydraulic brakes apply the braking force but a sensor on the brake pedal provides the driver's input to an electronic control unit instead of the brake pedal linkage.
These embedded systems on one hand improve the controllability and precision but on the other increase the complexity as well as the efforts to improve their reliability. One of the main reasons that automakers can investigate and implement XBW approaches today is the availability of a variety of semiconductor ICs that meet the cost targets to provide the control, power and communications required for these systems. Hence, in this thesis when we talk about the driver input signals we refer

to the outputs of these systems. They only inform the VCU about the driver intention. No direct mechanical action is done, so these signals can be handled by VCU as it wants.

1.3 About software engineers

1.3.1 Separation of Concerns

Since software development is a creative activity, informal techniques are very widespread in software design and programming. But the informal approach is in contrast with good software engineering practices **[21]**.

Separation of concerns is a way to simplify the management of the program by dividing the task in many sections corresponding to different purposes. In this way each sub-system can be developed regardless from other sections giving the possibility to update it independently.

This is a great advantage since we can improve or change the program when we want and without having to make corresponding changes to other sections. Furthermore, this also allows the creation of specialities groups, in order to allocate the problem in a proper way. These groups work efficiently and mature independently and at different rates, enabling an organization to be more flexible in adapting to new technologies and opportunities.

Within the Software Engineering context, we can distinguish at least three organizational areas: Systems Engineering, Software Engineering and Controls Engineering.

From software development's point of view, the left side of the "V" cycle activities concerns business requirements, definition of specifications and some aspects of architecture and high-level designs. These activities are usually addressed to Systems Engineering. Controls Engineering focuses on the development and testing of the control algorithms, coding and finally on the physical system which plays a key role in the requirement verification of the program.

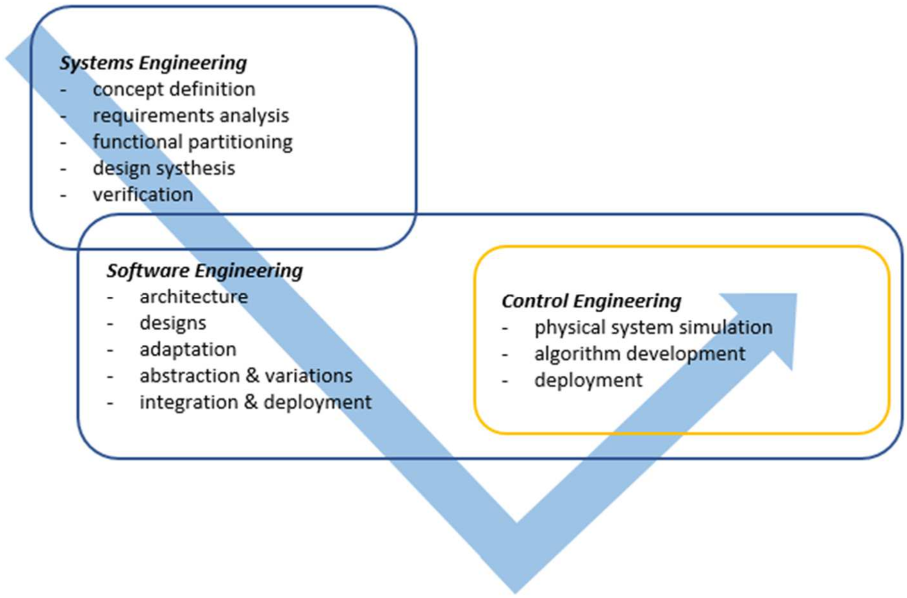


Figure 18 – Organizational areas among Software Engineering.

Software Engineering instead, deals with of creating a sustainable and scalable software architecture that allows the team to have a clear direction for the coding phase, creating an environment which does not create problems to the software developers. Software engineers are also responsible of reducing the *'learning curve'* for new team members and enabling Model-In-Loop and Software-In-Loop testing with the integration of software components, control algorithms and libraries **[22]**.
Regarding this work, it touches the domain of each area especially the Control Engineers since the aim of this thesis is the development and testing an algorithm which is capable of providing the motor torque's set-points, according to the driver request and the chosen powertrain's configuration. Obviously, to test the algorithm, we need to have a physical model of the vehicle.

1.3.2 Model-based testing

In the last decade, the growth of software modelling in the automotive field brought to a development process which uses model-based technologies. This led having many advantages especially with regard to the reduction in development time. Model-based technologies such as MATLAB/Simulink, support powerful mechanisms for managing and processing continuous signals and events which are the most important data types in the automotive domain. These model-based technologies allow the development of high-level models that can be used for simulation in the earliest stages of the development process. This is an important fact since automotive development is an interdisciplinary activity with software, electrical, and mechanical engineering aspects inextricably entwined. graphical models of functions such as a black box allow engineers to find a common functional understanding in the concept phase. Model-based development help developers to communicate with each other, with customers, or between car manufacturers and suppliers. It reduces time to market since these models can be reused and at the same time reduce costs by validating systems and software designs before the implementation. Consequently, models are often handled as part of the requirements specification, since the models indicate the required functionality in an executable way. Furthermore, model-based development provides a development process from requirements to code, ensuring that the implemented systems are complete and behave as expected **[23]**.

1.3.3 ISO 26262 applied on Software modelling

ISO 26262 recognizes that software safety and security must be addressed in a systematic way throughout the software development life-cycle. This includes the safety requirements traceability, software design, coding and verification processes used to ensure correctness, control and confidence both in the software and in E/E systems to which that software contributes. A key element of this standard (part 4) is the practice of allocating technical safety requirements in the system design and developing that design further, to derive an item integration and testing plan, and subsequently testing themselves. It implicitly includes software elements of the system, with the explicit subdivision of hardware and software development practices being dealt with further down the V model.
ISO 26262 (part 6) refers more specifically to the development of the software aspect of the product. It is concerned with:

- Initiation of product development at the software level
- The derivation of software safety requirements from the system level (following from part four) and their subsequent verification
- The software architectural design
- Software unit design and implementation
- Software integration and testing

Moreover, there are several things that are to be covered by modelling and design guidelines. Among these, the enforcement of low complexity is highly recommended for all ASILs **[24]**.

1.3.4 Testing algorithm by MIL, SIL, HIL

In general, the V-model cycle is the reference for a design process of a project. One of the main issues during the product development is to verify whether all the requirements that were posed to it at the beginning of the process are met. For this purpose, it is necessary to carry out tests during the development of the product. Model-based design approach allows for a rapid development of isolated section. From a system engineering point of view, it makes easier customer requirements definition as it provides capability to show and modify the expected behaviour in the early stage of the project **[25]**.

Model-in-the-loop (MIL) simulation concept is usually the first method of verification of the new developed algorithms, allows to reduce strongly the final cost of the project by removing errors and defects **[26]**.

In MiL testing the controller and the environment are simulated to test the functionality in the modelling framework without any physical hardware components. The controller is implemented in Matlab-Simulink and the plant usually is a Simulink model.

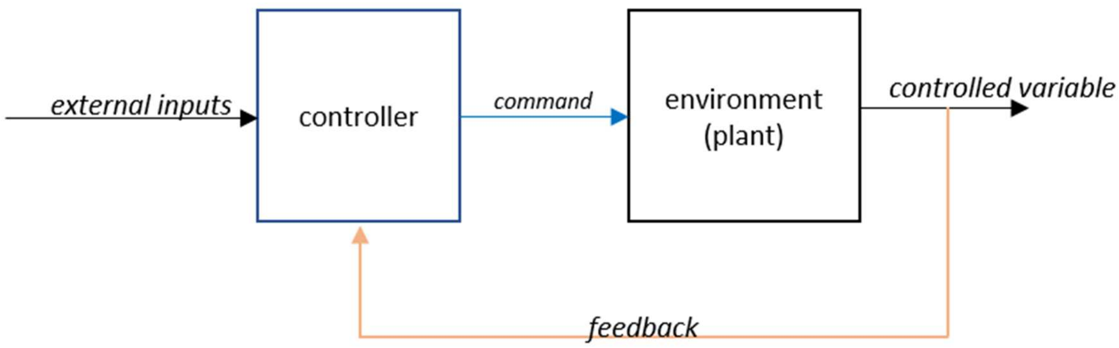


Figure 19 – Testing loop for the controller. The plant is a virtual model.

To make sure that the code is working correctly it is tested using Software-in-the-Loop (SiL).

Software-in-the-Loop (SiL) testing begins with code being generated from the controller model. This code is then tested in a virtual environment, without any hardware, to test how well the software handles the simulated system. Tests are made to make sure the code works identical when using different types of input conditions, functions, and mathematical algorithms. SiL testing is a good approach when simulating a real-time system that requires fast iterations, to ensure that the software is able to meet the requirements. Once the generated code is verified to work on a real-time boardnet the next step is Hardware-in-the-Loop (HiL) testing. The code is now implemented in the final hardware setup. HiL simulation has to have some sort of actuators or sensors, real or simulated that represent the hardware of the plant. By performing these tests developers may find errors and problems early in the developing stage instead of finding them when the control system and the plant are integrated **[27]**.

2. Requirement Specification

2.1 Global analysis

2.1.1 Why adaptable control?

Since it is not known yet what type of configuration will be mainstream for future powertrain, researchers and car-makers are looking into the possibility to build modular and scalable systems. This means that you can mount one, two, three or four modules in a vehicle, depending its requirements and purpose.

Identification of modularity and scalability comes in place especially when regarding 4-wheel drives and the option of application in other vehicle concepts as well. The meaning of modularity, in general, describes the possibility of exchanging different components for the same application. An example situation would be the case, when the electric motor of a drive train is changed without the need of adapting the other components. Changing the axle from front to rear could be also an approach for modularity as an example. Anyway, this criterion also includes a conclusion concerning the scalability in power for a drive train concept by the allowance of exchange of the components (e.g. variation of motor diameter for higher torque) **[28]**.

For these reasons it would be convenient to design a single adaptable algorithm that computes requested motor torques, based on the driver's inputs (accelerator and brake pedals) and taking into account at least the following:

- Number of electric machines (for a maximum of four);
- Rated power of each motor.

Furthermore, the control has to implement the following basic features of an electric vehicle:

- Regenerative braking;
- Coasting function;

- Creeping function;

Starting from this idea a flexible control is proposed, enriched with some essential functions described in the next paragraphs.

2.1.2 Analysis of functional requirements

Requirements specification is used to describe the desired behaviour of a system to be developed. Usually a functional analyst receives the requirements and draws up a document with all the specifications. In order to define the requirements, we need a clear and complete understanding of the products to be developed. Usually, the requirements do not refer to constraints on design or execution (such as performance requirements, quality standards, or design and stability constraints). In this work, we refer to the requirements related to the control of powertrain which derive from the highest level of the V cycle, thus they concern the main functions of the system which are concretely perceived by the driver. This is because a powertrain's control manages (along with other systems) the driver request coming from the accelerator and brake.

Some basic requirements for an electric powertrain have been discussed in the chapter 1.2. In addition to the typical electric driving features we have to analyse also the safety requirements resulting from the hazard analysis.

The simplest essential function of a powertrain's control algorithm is to manage the sign of the torque on the base of the engaged gear: if '*Drive Mode*' is engaged the motors have to deliver the tractive effort to move the vehicle forward (positive) while in '*Reverse Mode'* torque has to be negative. Furthermore, when Neutral Gear is engaged no torque shall be delivered as well as when parking brake is active. Hence, when parking gear is engaged we suppose that only foundation brakes are active and so no torque can be delivered.

Some of basic powertrain's control features consists in an energy management (or battery management) which evaluates, how much energy can flow through the battery in the time unit. The allowed maximum charging power depends on several factors, firstly the battery SOC status. So, we will consider the maximum power deliverable by the battery as well as the maximum power which it can absorb. Torque Path should adjust its outputs to comply these requests which will in turn be managed after by VCU.

Another important requirement for a torque path is the ability to take into account also possible speed reductor which different values of gear ratio between front and rear axle. As said in the previous chapter these reductors can bring advantages in terms of vehicle performance improvements but make

also the torque’s control more complex especially during the recuperation braking because of the regulation UNECE-13H. Furthermore, we have to consider also the possibility to set the desired braking strategy (parallel or series phasing) by the coefficient defined in the *paragraph 1.2.3.*
A torque’s control should include a slew rate limiter, as well. This essential function serves to limit the first derivative of the torque’s signal passing through, depending on the type of step response we want. It is important to underline that motors with integrated inverter include a rate limiter most of the way, both for the torque and the speed (if a speed controller is present). We could change these values by sending the desired slew rate to the inverter via CAN considering that there is a certain threshold for slew rate set by manufacturers.

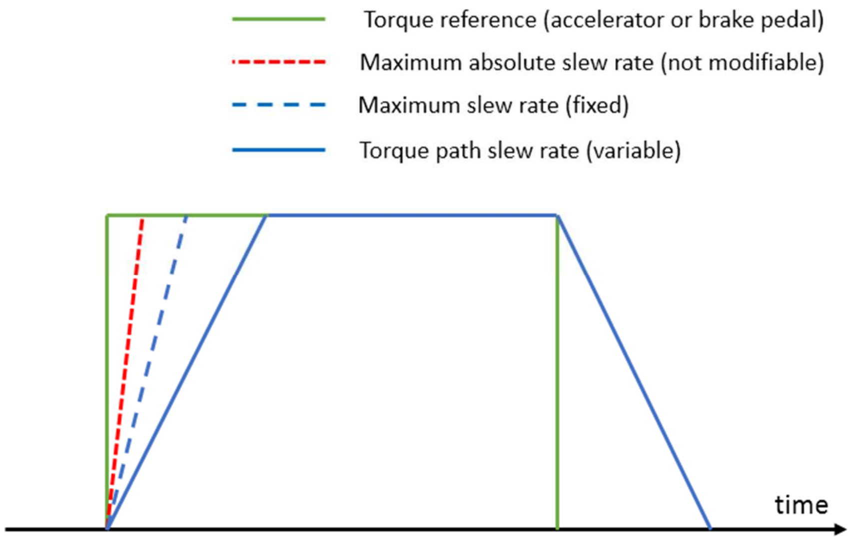


Figure 20 –Typical step response of motor’s torque when rate limiter is active.

The step response has an absolute limit in terms of slew rate corresponding to the limit of inverter (red dashed line) and a fixed maximum value which can be changed (blue dashed line). Within this boundary torque path imposes the desired slew rate depending on the chosen driving strategy.
Since torque can be positive or negative, to make the control more flexible, it can be useful to consider at least two values of slew rate: one positive, valid when torque is increasing up and one negative when torque is decreasing. These values can be set as equal or different from each other.
Anther requested function consists of the possibility to disable or enable at least the following:

- Accelerator;
- Creeping;
- Coasting;
- Torque on one axle.

The latter one deserves a dissertation. During driving, a warning or an error due to a possible malfunction of the system, can occur. In this case, we shall be able to shut down both the motors of the axle in which the fault occurs, even though the problem concerns only one motor. This is a further improvement of the safety. When an axle is disabled, we could think also to compensate the lack of the torque by the other axle if it is possible. We will implement a logic which can fulfil the latter requirement: when we decide to shut down an axle, the corresponding availability of torque is set to zero by the algorithm. Then the distribution's coefficient has to change, in order to deliver the requested tractive effort.

With regard to the distribution of torques in a vehicle we do not deal with about torque vectoring, as mentioned in previous chapter. The proposed control algorithm shall distribute the torque only considering the power availability of each axle. The basic idea is very simple: it consists of definition of two complementary coefficients, less or equal than one, which represent the potential contribute of one axle to total deliverable power. Actually, we do not consider the power, but the torque, as indicated in the definition:

$$m_{front} = \frac{maximum\ available\ torque\ front\ axle}{total\ available\ torque} = \frac{T_{av,front}}{T_{av,rear} + T_{av,front}}$$

$$m_{rear} = \frac{maximum\ available\ torque\ rear\ axle}{total\ available\ torque} = \frac{T_{av,rear}}{T_{av,rear} + T_{av,front}}$$

$$m_{front} + m_{rear} = 1$$

The available power as well as the actual power of the motors are essential to execute the distribution algorithm. We cannot control anything without these data: for this reason, availability of torque has to be sent to the control algorithm by VCU regularly, with a specific frequency. In the followings paragraphs we will give more details about this.

In the list below, we have summarized the Torque Path's requirements; the subject is always the control algorithm:

- it shall be able to distribute torque to the axles proportionally, according to the driver request as well as the available torque of each single axle;
- it shall be able to limit the delivered total power (positive) and the recovery total power (negative) according to the corresponding maximum input values;
- it shall be able to split the total braking request according to the defined coefficient k_{regen}.
- it shall be able to control the torque's slew rate, both for positive and negative gradient of torque.

- it shall be able to adjust (reduce) the power of one axle (front, rear or both), and also compensate the lack of torque if it is possible.

2.1.3 Analysis of safety requirements

The aim of the functional safety concept is to comply the safety goals and to specify the basic safety mechanisms in the form of functional safety requirements which are allocated to elements in the system architecture.

To specify safety mechanisms the functional safety concept addresses the following:

- Fault detection and failure mitigation;
- Transition to a safe state;
- Fault tolerance mechanisms, where a fault does not lead directly to the violation of the safety goals and which keeps the system in a safe state.
- Arbitration logic to select the most proper control request from multiple requests generated simultaneously by different functions

In this work we only focus on some of torque-related safety goals: basically, they concern the reaction to the detection of a torque error or deviations from the reference value.

For instance, we must make sure that we never deliver different values of torques between right and left wheels. This situation could happen when, in a twin-motor axle, one motor saturates its power because of an overheating.

We do not deal with the management of a motor fault because this can bring the motor to an Active Short Circuit condition (ASC) even known as Uncontrolled Generator Operation. When running permanently excited synchronous machines, impermissibly high voltages can occur at the inverter in a fault scenario. For example, when in high-speed operation under field weakening the active control of the inverter fails. To protect the inverter, the motor alternating current lines can be short-circuited via the high or low side of the bridge in order to deflux the machine and shut down the inverter **[29]**. Obviously, the negative torque applied by the motor during an active short circuit to the driving wheels must not lead to vehicle instability under all driving conditions.

This means that, if an ASC occurs on a twin-motor axle, both the motors must deliver the same braking torque to avoid an unintended yaw moment, keeping the vehicle in a safer transition. This emergency's situation is managed by the vehicle dynamic control usually by predictive strategies of recovery.

Aside from those particular conditions, in normal driving situation, the safety requirements refer to the unintended yaw moments and the mechanism to avoid them. About the control algorithm we can say that the delivered torques by the axle's motors always shall be reciprocally equal, for each driving condition. This is valid for the longitudinal model of the vehicle and for a straight speed's direction. This means that if a torque deviation occurs on one of these motors, we must take the necessary measures, for example by changing the torque's references to prevent that different torques are provided for the axle.

Another important safety requirement concerns the 'coordination of torque'. Since target values of torque can be different between front and rear axle, we make sure also the signs of the actual torques (front and rear axle) are not different between each other. It could happen when a front-axle's motor has different response (in terms of time constant) with respect to the rear one, whatever the causes. Thereby we could have some critical situations during a braking: for instance, while an axle is delivering negative torque as requested, the other axle is still delivering positive torque, bringing the vehicle to an unsafe condition. Since it cannot be acceptable, a safety mechanism to avoid this, is needed.

During the analysis of safety requirements some issues arise when we consider all possible cases about the accelerator and brake pedal position.

The following basic safety requirements must always to be fulfilled, regardless the powertrain configuration or any other factors:

1. if both accelerator and brake pedal are not pressed and the vehicle's speed is less than a certain threshold, no electric torque must be delivered;
2. if both accelerator and brake pedal are simultaneously pressed, accelerator signal must be overridden by the brake.

Let is complete the previous list of safety requirements in order to sum up what we have said:

3. no deviations of torque (no delta torque) are allowed between right and left drive wheels less than a certain tolerance;
4. no different torque's signs between front and rear axle are allowed.

The proposed solutions to comply these requirements are described in the following paragraphs.

2.2 Concept phase

2.2.1 Vehicle model

In this section we focus on the vehicle model which can allow us to test the control algorithm. Simulation of the vehicle behaviour is an important issue in today automotive engineering because of the high complexity of the environment. In an electric powertrain new risks can arise especially if we have more than one motor which provide the tractive effort. Hence, engineers need to simulate what happens in every driving situation to avoid problems in the future developments, before starting any production process.

The simulation of the powertrain control concerns many aspects touching the domain of the vehicle dynamic engineering. One of these aspects concerns the torque distribution system which is responsible along with other systems of the vehicle's driveability and the safety. Since we do not deal with the vehicle lateral motion as mentioned, we will consider only the vehicle longitudinal model without slip. Following there is a summary of the mathematical formulas to calculate the velocity of the vehicle starting from the delivered torques to the drive wheels.

The first step is to produce an equation for the required 'tractive effort' which is the force propelling the vehicle forward, transmitted to the ground through the drive wheels.

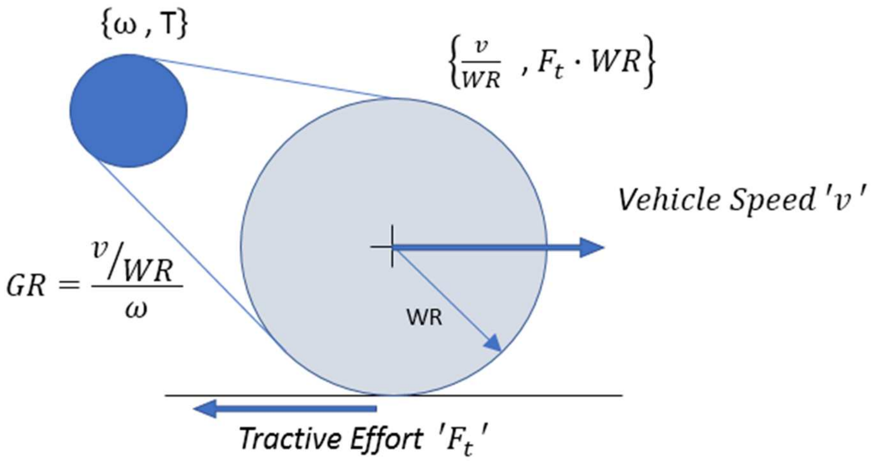


Figure 21 – Pure rolling physical model. In blue the drive shaft (motor).

To move the vehicle, we have to provide the sum of the following quantities **[30]**:

$$F_{tractive} = F_{rolling} + F_{drag} + F_{car,accel} + F_{rot,accel} + F_{climb}$$

where:

1. $F_{rolling}$ is rolling resistance force (or rolling friction): is the force resisting the motion when the tire rolls on the ground. This is a result of non-linearities of the tire because of the mechanical hysteresis of a not rigid body. This complex behaviour is usually described very

simply considering this force proportional to the vehicle weight by a constant coefficient so: $F_{rolling} = \ \mu_{rr} \cdot VehicleMass$

2. F_{drag} is the air resistance, acting opposite to the relative motion of any object which moves through a fluid. Unlike the resistance rolling it is closely related to the vehicle speed because it depends on its square as well as by other factors concerning the geometry of the vehicle and the fluid density. Since we do not know the geometry of vehicle we will use a global air drag coefficient:

$$F_{drag} = \mu_{ad} \cdot (VehicleSpeed)^2$$

3. $F_{car,accel}$ is the force needed to accelerate the vehicle, in addition to the other resistance forces; according to the famous Newton's Law:

$$F_{car,accel} = VehicleMass \cdot a$$

where 'a' is the desired acceleration. (This is true only if consider a pure rolling, neglecting slip).

4. $F_{rot,accel}$ is the force that the motor has to deliver to accelerate only the rotor mass. Since a gear reductor can be used in the torque's transmission, gear ratio has to be considered. So, if 'GR' is the gear ratio of the system connecting the motor to the axle defined as $\mathrm{GR} = \frac{\omega_{\mathrm{wheel}}}{\omega_{\mathrm{motor}}}$ and 'T' the motor torque, the force transmitted by wheel to the ground is:

$$F_{tractive} = \frac{GR}{WR} T$$

where 'WR' is the effective wheel radius. Thereby motor angular speed is giving by $\omega_{rotor} = \frac{GR}{WR} v_{car}$, where v is the tangential wheel's speed corresponding in our case to the vehicle speed (because of pure rolling assumption). Similarly, the motor angular acceleration is the derivative of the previous formula, so we can say: $\dot{\omega}_{rotor} = \frac{GR}{WR} \ a_{car}$

Therefore, the required torque for this angular acceleration is given by:

$T = J \times \dot{\omega}_{rotor} = J \times \frac{GR}{WR} \ a_{car}$

where 'J' is moment of inertia of the rotational parts. Now we can say that the minimum effort needed to accelerate the rotor is:

$F_t = \frac{GR}{WR} T = \frac{GR}{WR} J \times \frac{GR}{WR} \ a_{car} = J \left(\frac{GR}{WR}\right)^2 \ a_{car}$

If we want also to consider the global gear efficiency 'η_g' we can incorporate it in the equation in this way:

$$F_{rot,accel} = J \frac{1}{\eta_g} \left(\frac{GR}{WR}\right)^2$$

5. F_{climb} stands for the force required to drive a vehicle up a slope. Simply, it corresponds to the component of weight along the inclined plane:

$$F_{climb} = VehicleMass \cdot g \cdot \sin(\theta)$$

Where 'g' is the gravitational acceleration and 'theta' the angle of the inclined plane.

Now, we are able to calculate the acceleration of the vehicle for a given motor torque summing every single contribute to the total tractive effort by the following basic formula:

$$F_{tractive} = \frac{GR}{WR} T = (\mu_{rr} + g\, sin\theta) \cdot VM + \mu_{ad}\, v^2 + \left(VM + J \frac{1}{\eta_g} \frac{GR^2}{WR^2}\right) \dot{v}$$

How we can see, the contribution of rotor's inertia can be interpreted as a further increase of total vehicle mass and thus it can be included in it. That is a first-order differential equation in the variable $'v'$. As usual, to solve this differential equation, it is brought in the Laplace's domain. The entry variable is the algebraic sum of the following two contributions: the overall delivered wheel's torque by the motors and the total torque provided by foundation braking system.

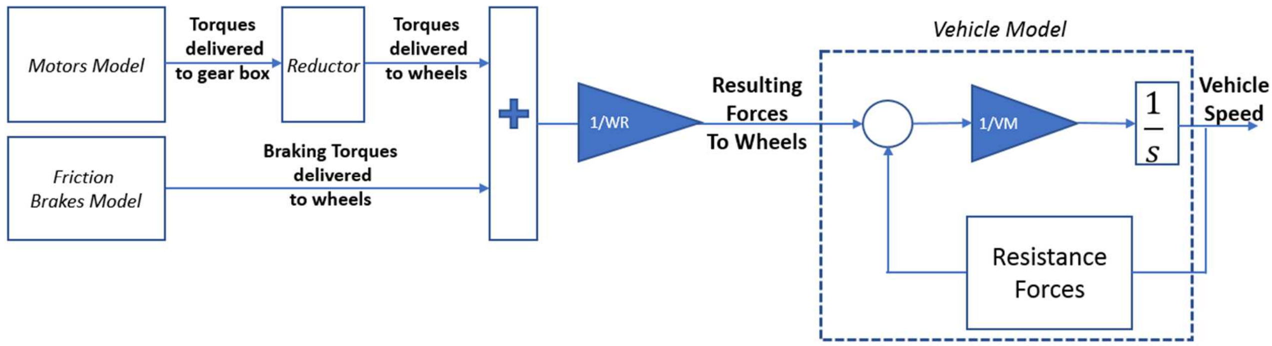


Figure 22 – Block diagram of the physical layer.

Aside from the analysed resistance forces, we also must consider the contribution of the friction braking to complete the model. Theoretically it does no matter where we put it; however, we want to think in terms of braking torque, so we will add this contribute to the delivered motor torque at level of wheels.

Then, the resulting torque can be converted in the tractive force, dividing it by the wheel radius. This force is transmitted by the drive wheels to the road. If there are more than one motor, the model's input shall be the sum of all delivered motor torques while the value of friction torque represents the total braking torque provided by foundation brakes.

Hence, from this 'tractive effort' we subtract all the resistance forces in order to calculate the vehicle acceleration and thus its velocity. Referring to the figure above, we note that, among these resistance

forces, another one can be added in the model. This is because sometimes we might use an equivalent model of all frictions by the identification of three coefficients called $f_{0,}\ f_{1,}\ f_2$ resulting from the well-known experiment 'coast down'.
Furthermore, we should add to the model a block which decides the sign of the resistance force, if we want to consider both the directions of speed, forward and reverse without having to change the model each time.

2.2.2 Electric motor and brakes

Electric motors along with friction brakes are the only systems which allow the vehicle's movements or the vehicle's stop; they are responsible about the driveability as well as the driving safety, therefore higher values of reliability and a strong fault-tolerance are needed for these systems.
Since we have already discussed about general aspects of these systems we want only say how these systems are modelled in our case.
Friction brake system will be considered like an ideal actuator: this is not so wrong because of the strict requirements about the response time and the precision required as mandatory. This means that we can assume the transfer function of braking system as equal to one; so, the requested value of friction braking torque is given directly to the physical model.
Since the tire's slip is neglected, we will not consider the emergency braking too, because in this case Anti Block System is activated and no torque to the wheel can be delivered by motors.
In general, to carry out hundreds of simulations within an acceptable time, it would be convenient to have simple model of the systems we want to simulate: this allows us to focus on the control algorithm behaviour. Once the model validation is made then we can increase the degree of details of the physical model. In our case modelling a motor-inverter with a lot of details could only lead to a waste of time, so we will only consider a first-order model by implementing a transfer function's Simulink block considering the inverter as an ideal actuator.
Since Torque Path always needs to know the availability of the motor torques to execute the algorithm, a simple look-up-table of the motor's torque-curve we be implement by Simulink. Furthermore, we cannot take into account all the possibilities about the torque-speed characteristics, so a simple way to manage this issue, is to implement a motor torque curve with a top value of '1 Nm' (likely 'per unit' calculations). Then the output torque will be multiplied by a gain that is the desired value of maximum torque. Regarding the shape of motor characteristic, to define the 'constant torque' area, we impose a 'speed base' equal to 30% of the maximum motor rotational speed. Since we are not so interested in the calculation's accuracy, for the flux weakening area a straight line is

conducted from the maximum torque (corresponding to the motor speed base) to zero (corresponding to the maximum motor speed) as we can see from the following figure. We will define the motor curves by giving only three values: the maximum torque, the base speed and the maximum rotational speed. By these values we will be able to implement an elementary motor curve by a look-up-table loaded in the input generator. We must notice that this motor characteristic is valid also for generator's operation.

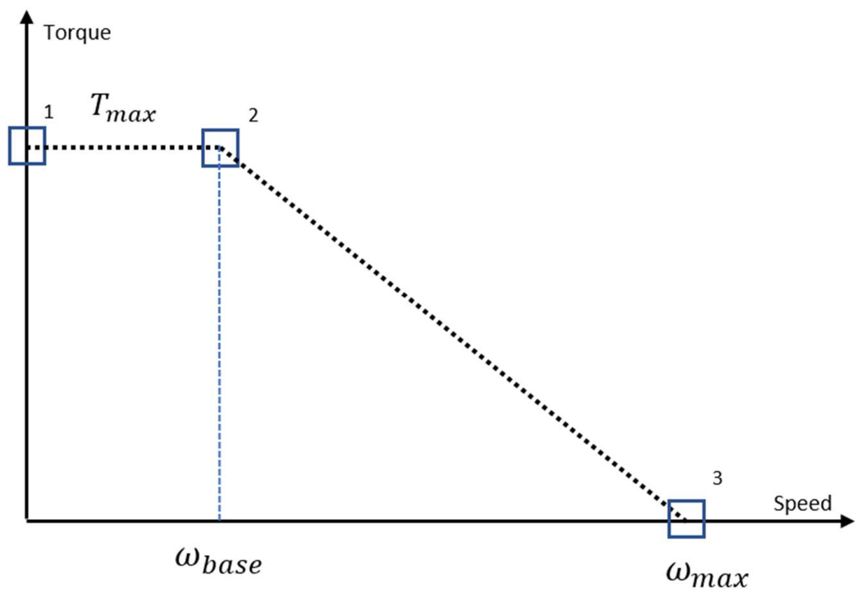


Figure 23 – An extremely simplified torque curve used to simulate the electric motor.

With regard the availability of torque, the proposed control algorithm can consider the value of the available torque as it wants. For example, if we want to shut down an axle the algorithm will consider null the available torques of the corresponding motors, regardless from the effective availability. This also allows us to set a certain ratio between the torques of front and rear axle which can mean the possibility to carry out a certain braking strategy in view of further developments.

2.2.3 Inputs Generator

To carry out a simulation we have to provide to the control model the inputs which are needed to execute the algorithm.

In the following table we have gathered all the considered inputs for a torque's path and their theoretical range of variation, distinguishing them in two main categories: commands and feedback signals. We have taken as reference the motor CAN matrix, mentioned in the *paragraph 1.1.4*.

	Torque Path Inputs	Type	Minimum	Maximum	Unit
driver commands from VCU	accelerator pedal	float	0	1	-
	brake pedal	float	0	1	-
	engaged gear	integer	1	4	-
	k_regen	float	0	1	-
	coasting mode	integer	0	4	
	maximum mechanical power - positive	float	neg. inf.	pos. inf.	kW
	maximum mechanical power - negative	float	neg. inf.	pos. inf.	kW
	slew rate - positive	float	0	pos. inf.	Nm/s
	slew rate - negative	float	0		Nm/s
	gear ratio - front	float	0	pos. inf.	-
	gear ratio - rear	float	0	pos. inf.	-
	enable accelerator	bit	0	1	-
	enable creeping	bit	0	1	-
	disable front axle	float	0	1	-
	disable rear axle	float	0	1	-
feedback	vehicle speed	float	neg. inf.	pos. inf.	km/h
	actual torque	float	neg. inf.	pos. inf.	Nm
	available positive torque	float	neg. inf.	pos. inf.	Nm
	available negative torque	float	neg. inf.	pos. inf.	Nm
	rotor rotation speed	float	neg. inf.	pos. inf.	RPM

Table 3 – A list of the main variables considered for the Torque Path.

The distinction about the type of the transmitted or receiving signal is needed because we have to know the quantity of bits that the ECU has to manage. Basically, the variable's type depends on the purpose of the signal: for example, to enable or disable a system or a function it is sufficient only one bit because of its binary logic differently the torque command, for example, which requires 16 bits (2 bytes).

The last four inputs of table above correspond to only one inverter control unit, so we have to manage group of signals as many as the number of motors. Number and type of input signals along with the amount of needed calculation, will determine the final hardware technical specification of the corresponding control unit. Since the needs to provide to the algorithm the feedback signals from the inverter-motor, we have to say in which way we will carry out the exchange of information between the inverter and the controller.

Referring to the following figure, the controller's inputs are the actual values of: torque delivered by the motor (deriving from the output of the motor-model block), the rotational motor speed (from the output of the vehicle model) and the actual value of the available torque (from VCU).

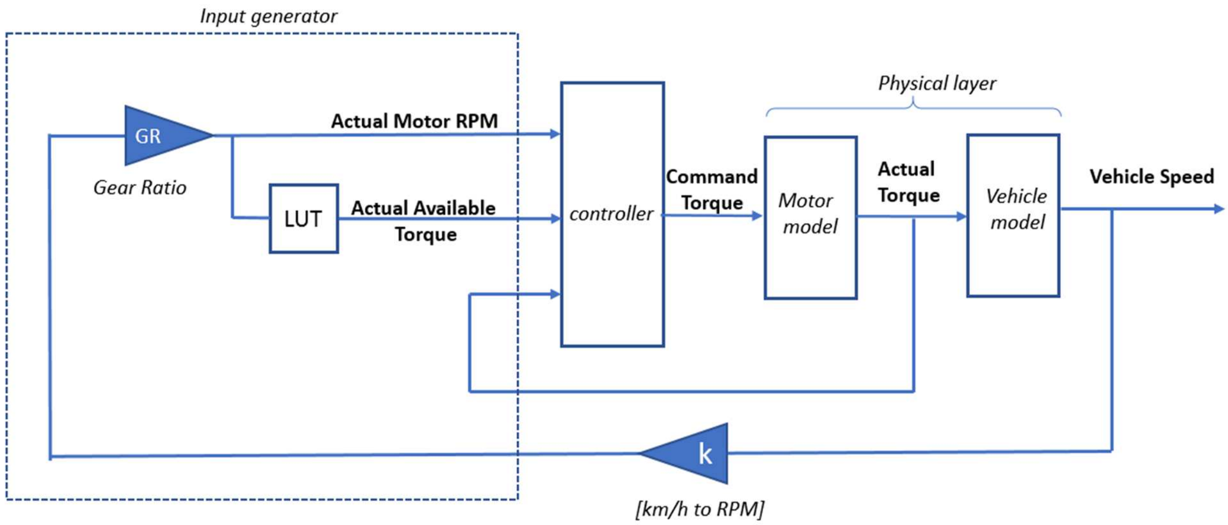


Figure 24 – A focus on the input generator and how feedbacks from motor' model are managed.

Since it is no difference between the vehicle and wheel's tangential speed as we have supposed, the latter one is sent to the algorithm by a simple conversion from km/h to rpm considering both the wheel radius and the gear ratio. Once the motor speed is determined, we can extrapolate the actual availability of torque both for motoring and generating torque from a look-up-table loaded in the input generator.

In the proposed algorithm the input variables named 'Disable Front Axle' and 'Disable Rear Axle' refer not only to the possibility to shut down the motors of a single drivetrain but they also allows us to reduce the power of a single axle as we want.

2.2.4 Model overview

When we approach to control's software modelling an essential practice to handle complexity is firstly to keep the control's model separate from everything we need to test it.

Generally, we can distinguish three main parts: one is precisely the proposed control model, while the remaining two parts consist of an input generator and a physical model by which we can test the control algorithm.

The block named 'Simulation Results' includes the physical layer and the scopes which allows us to display the results of the simulations. Regarding the Event Layer, we have not mentioned it yet.

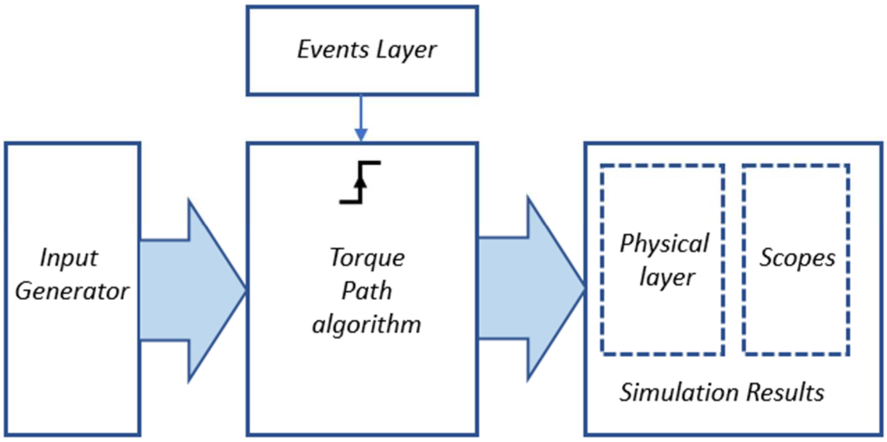


Figure 25 – General overview of the model: main parts

Basically, this block simulates the triggering of the control task by a series of impulses during normal operations ('medium time'). This block provides also the impulse to initialize all the variables when the ignition's signal is turned on.

The rising edge of these impulses imposes the triggering's frequency of the control task. Since the EV's motor time response absolutely must be less than 500 milliseconds (for safety reason), the frequency of the control task is usually imposed around 10 milliseconds. This value is a good compromise between a fast response's requirement and the need to have enough time for the outputs computation.

Whenever a trigger event occurs, torque path executes its algorithm by using a "scheduler" block after that the trigger signal is converted into a "function call" as shown in the illustration below.

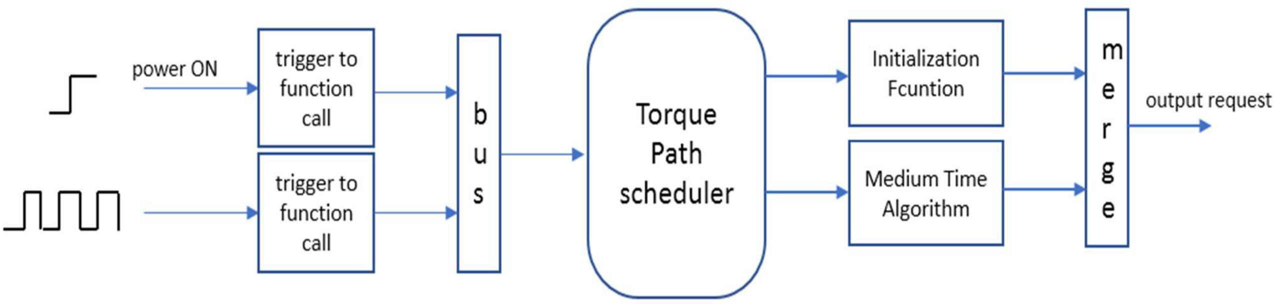


Figure 26 – How the signals from event layer are converted in a 'function call' by a Scheduler

Torque path scheduler serves to determine the order of execution for specific Simulink subsystems: so, the first impulse (ev_PowerOn) imposes the execution of block which initialize the requests, then the following impulses (ev_MediumTime) trigger the main control task.

The initialization values of the set-points are merged with the outputs of the main task as soon as the event of medium times occurs.
It is also important to say how we manage the possibility to have different powertrain configuration. Since the outputs of torque path consist of four motor torque requests, when we set a number of motors less than four, there are some outputs which shall be equal to zero. For this purpose, we will call the front axle's motors by the number 'one' and 'two' and for the rear motors by the numbers 'three' and 'four'. We suppose that if we set to one the number of motors for the front (or rear) axle, then 'motor 2' and 'motor 4' respectively are shut down.
Now, have a look at main blocks which are inside of the torque path by giving a simplified illustration of the functions with a quick description.

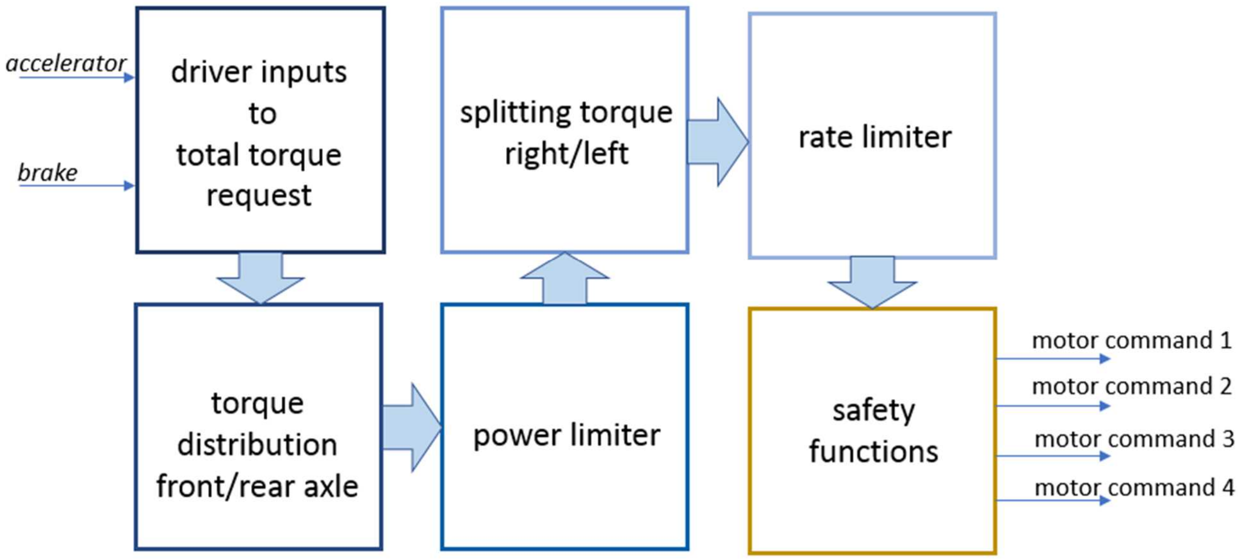


Figure 27 – Torque Path block diagram: from driver inputs to the motor commands.

The order of blocks is not arbitrary. The 'safety functions' shall to be placed at the end of the chain because we make sure that these functions are not affected by further operations.
Power limiter is put after the torque distribution (front- rear) because we have to consider the gear ratio for each axle which can be different.
The logic of this control is based on a simple principle: all the available torques coming from each inverter ECUs are cumulated by the algorithm in order to manage only one variable. The driver intention shall be converted into a total request of torque according to the pedal position.
The output torques are generally four but they can be less than four. In this case some motor commands are equal to zero depending on the selected configuration.

To manage the driver request about the direction of speed, we have to create a layer where we can choose among the possible cases setting the signal called ‘Gear’ by a simple multi-port switch as we can see from the following picture:

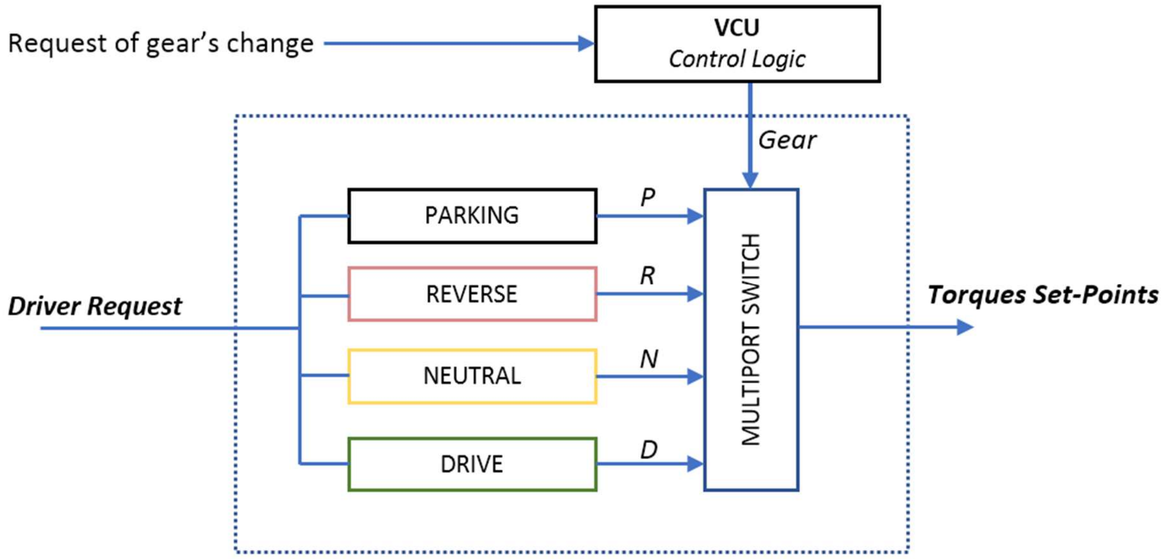


Figure 28 – How Torque Path can manage the gear's change request from VCU.

Each block includes all the functions related to the corresponding engaged gear. The global requirements of these blocks are summarized in the following list:

- If ‘Parking’ mode is active no torque shall be delivered to the wheels with exception during the ‘electric standstill’ and ‘electric hill hold’ features (we do not consider these functions);
- If ‘Neutral’ mode is active no torque shall be delivered to the wheels;
- If ‘Reverse’ mode is engaged only negative torque can be delivered. Furthermore, during reverse speed, the regenerative braking is usually not used.
- If ‘Drive’ mode is engaged the delivered torque can be both positive and negative. These torque set-points are managed finally by the VCU which manages the outputs as it wants.

Since the first two requirement are trivial case of control, the remaining two represent the core of the control. With regard the reverse mode, we suppose to deliver only negative torque without braking recuperation. This means that we can incorporate the block of reverse mode in the block of ‘*Drive Mode’* so we can put this block at the beginning of the chain described in the previous paragraph.

2.3 Torque path description

2.3.1 Reverse mode: from pedal force to the request.

The first action torque path has to carry out, consists of converting the signals coming from the driver (accelerator and brake) into a request of power or torque. This requirement can be implemented in different way. One of these consists of using the available torque from the motors to create a single representative value in order to relate it to the pedal position. This solution can be implemented both for reverse and forward speed direction.

The implemented logic consists in summing the available torque deriving from the motors and multiplying the resulting value by the percentage of request of the driver. The pedal signal is converted in a number between zero and one by a look-up-table according to the desired input response.

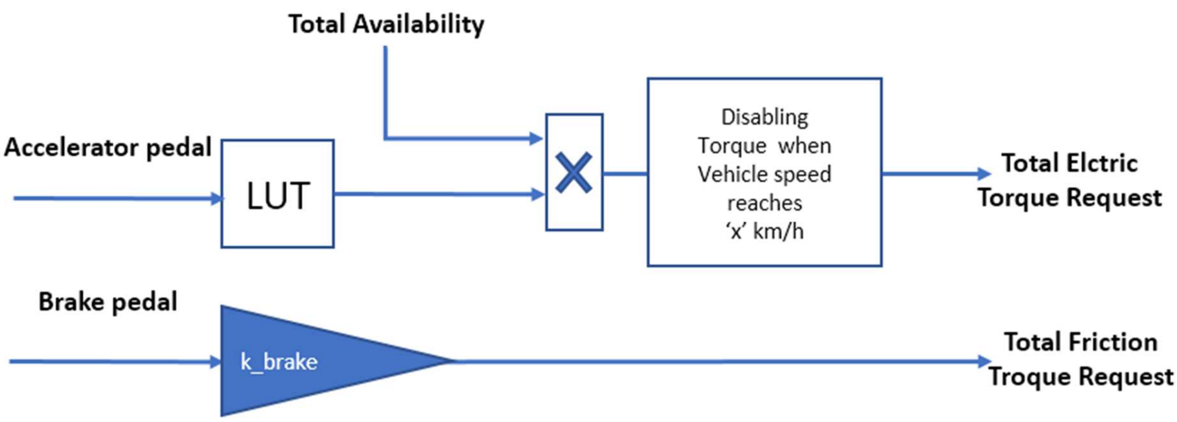


Figure 29 – Reverse mode: driver inputs conversion into a request of torque

Obviously the requested torques are negative for the motor while for the braking system we consider always the request as positive. We note that there is no interconnection between the two path which we call as 'Acceleration Path' and 'Deceleration Path' (even for drive mode). This is because regenerative braking is not allowed in reverse direction so, only the friction brakes can stop the vehicle.

Usually the speed in reverse direction is limited to a maximum value. In our model, we shall disable the torque when the vehicle speed reaches a certain value, like an ON/OFF controller. This can be implemented without problem if we consider the system's dynamic much slower with respect the controller's dynamic.

2.3.2 Drive mode: from pedal force to the request

As for the reverse mode, we focus on the block which calculates the set-points of overall requests starting from the driver input. Here the electric driving features are implemented by functions mentioned in the previous chapter. By the following illustration we can have an overview about these functions and their connections.

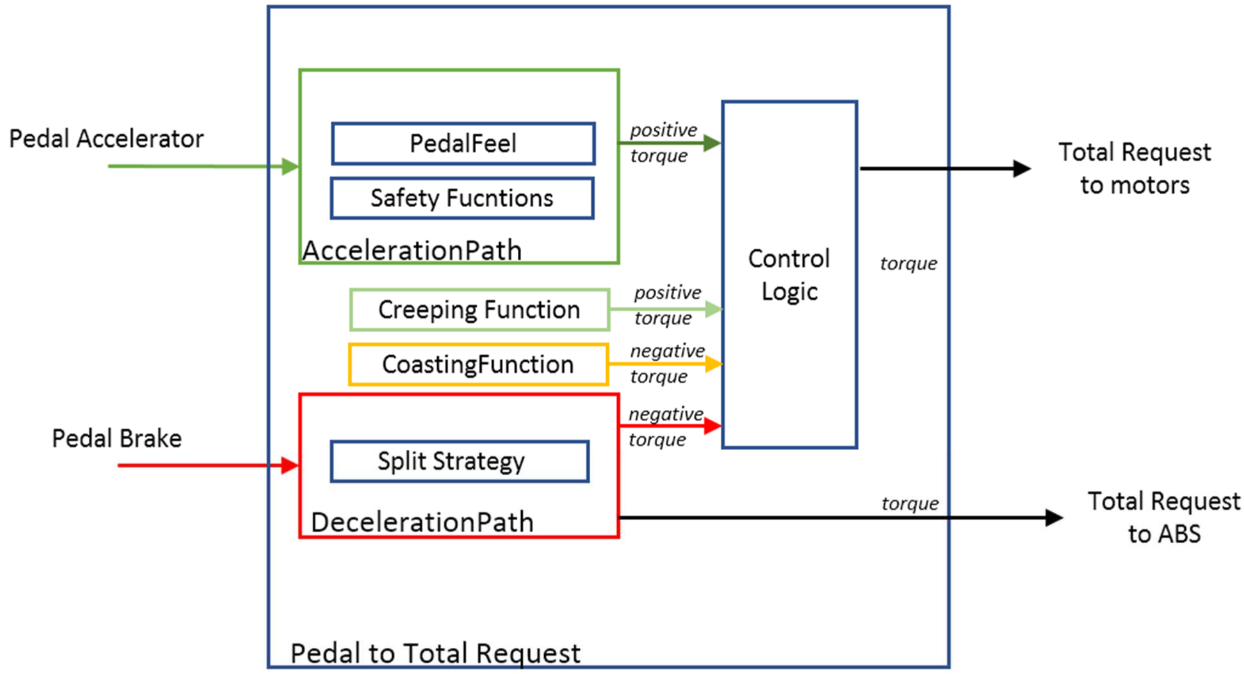


Figure 30 – Drive mode: pedal force to total request

Inside the main block there a series of subsystems which have a specific function. The outputs of these functions are sent to the control logic which manages these signals depending on many factors such as the driver inputs as well as the vehicle speed. The computation of the requested torque for each function are based on a product between the percentage of the request (pedal) and the corresponding availability. In this way we calculate all the torque's set points in per unit. The next sub-paragraphs give us a brief description about these functions and finally how their outputs shall be handled by the algorithm.

Creeping and Coasting functions

These functions will be implemented by simple look-up-tables with a variation range from zero to one for the axes of ordinates (requested torque) while the entry variable is the vehicle speed. To select the coasting strategy a simple multi-port switch will be used.

The only requirement for these functions consists of extrapolating from the tables the maximum values of torque for coasting and creeping, depending on the chosen strategy. The resulting value shall be multiplying by the driver request according with specific tables which convert the pedal position in a percentage of request for each function. The final output value will be managed by a block called ‘Control Logic’. Creeping function represents the complementary part of the coasting and provides the request of a motoring torque when accelerator and brake pedals are not pressed and the vehicle is slowing down.

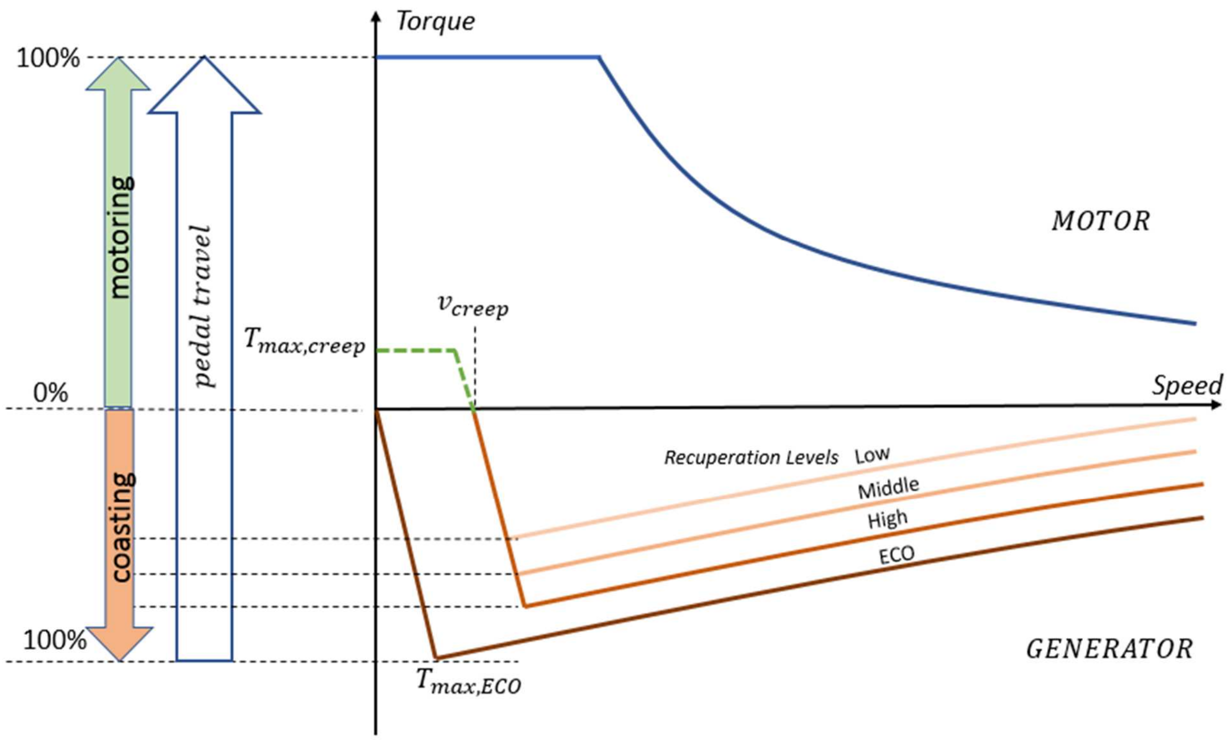


Figure 31 – A graphical description of the implemented logic about coasting (orange) strategies and creeping (green).

According to the diagram we shall be able to set the maximum torque deliverable to creep and coast. With a simple product between the output of the implemented functions (in per cent) and the corresponding maximum torque we can obtain a certain value of torque which is sent to the control logic. We shall be able to select the recuperation level by a multi-port switch according to the entry variable called ‘MODE’. The recuperation levels are implemented by look-up-tables which have as input the vehicle speed and as output the maximum torque corresponding to each coasting strategy. We consider only four level of recuperation:

- ECO: setting this function, creeping is disabled since we want to stop the vehicle, and the recuperation rate is the maximum allowed.
- High, Medium, Low corresponds to descending maximum values of braking torques. Selecting one of this strategy means that creeping function is active.

Acceleration path

In this subsystem the signal of the driver input is converted into a request of a percentage of total torque both for motoring and coasting request, according to table below.

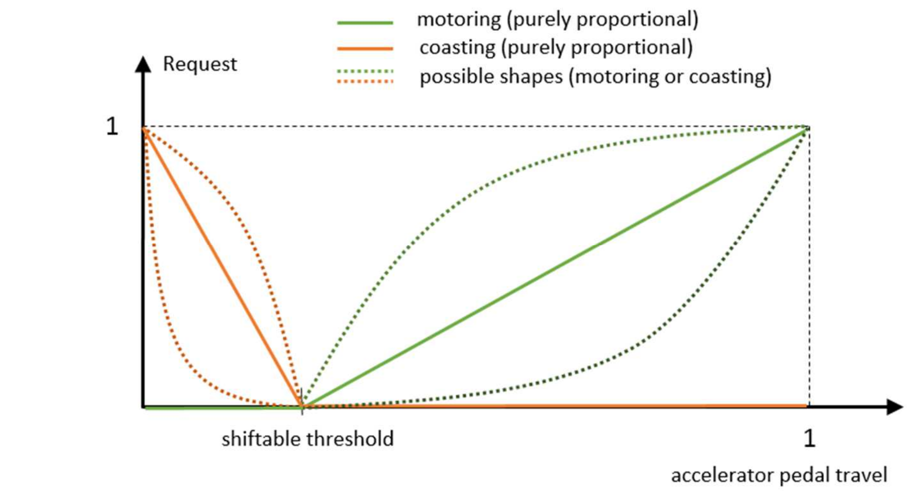


Figure 32 – A possible way to convert the accelerator position into a request of coasting or creeping torque.

The shape of these function can be variable depending on the 'feel' we want to get when we press or left the foot from the pedal. The shiftable threshold represents the value below which motoring torque's request is equal to zero; from that point, coasting torque's request starts to arise up to one. The dotted lines are shown just to say that there is not only one way which we can follow.

After calculating the requests according to the table, the resulting two values will be multiplied by the corresponding availability of torques and then will be send to the control logic which will be manage these values. The percentage of request for coasting is multiplied by the output value of the corresponding function as described in *Figure 33*.

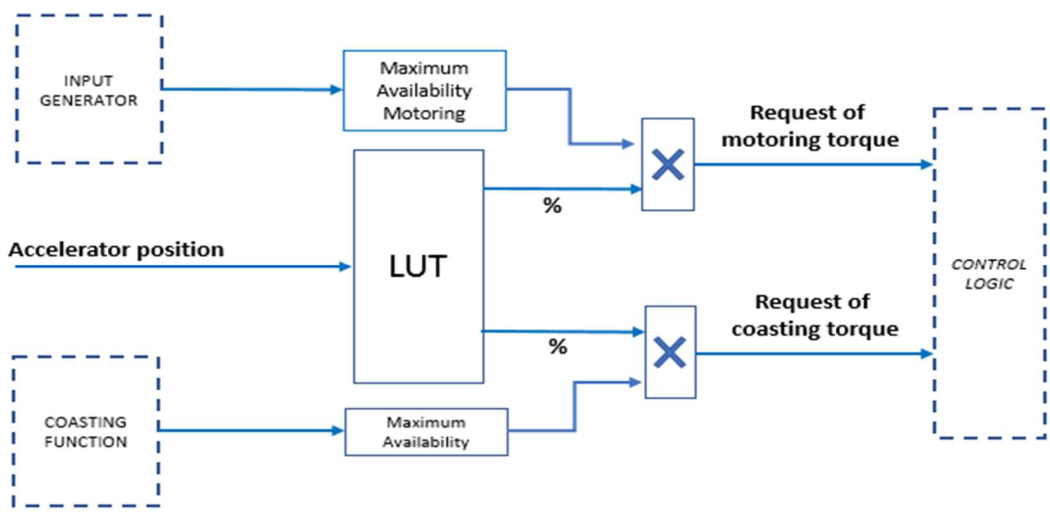


Figure 33 – Implemented logic to manage the request of driver (accelerator position).

The block named as LUT stands for 'Look-Up-Table' in which we will implement the functions described above. These latter are handled as if they were only one because the two curves are not independent from each other.

Deceleration path

It manages the computation of two main important quantities obtained by splitting the total request of the braking torque into an electric and a friction torque. The latter one will be sent to the ABS (or the braking control system which carries out the request) while the electric brake torque's demand is an input of the control logic.

The requirements for this block are the following:

1. It shall to able to set a certain ratio between the electric torques with respect to the total request, depending on the input variable '*k_regen'*.
2. It shall be independent from the gear ratio of possible reductors.

How mentioned in the *paragraph 2.1.2*, we will distribute the torque between front and rear axle only by considering their maximum available torques both for positive and negative torques with respect to the overall availability.

The first step is to convert the input signal from pedal brake in a possible value of overall braking torque. This is implemented by a product between the brake signal and a coefficient which represent the maximum braking torque deliverable by foundation braking named 'k_{brake}'. This value corresponds to the force needed to move the wheel when the brake's calliper is clamped onto the friction pad at maximum of its possibilities. With respect to the braking force delivered by electric motors, the maximum friction brake's torque is much bigger.

Once we have calculated the quantities called as m_{front} and m_{rear}, we can split the overall request of braking between front and rear axle. After then, for each axle we have to carry out a further division into an electric torque and a friction torque based on the input variablek_{regen}. Finally, the calculated friction torques are added together. Following the formulas used to carry out the described operations are shown:

The total request of braking torque is given by:

$$T_{brake} = k_{brake} \cdot BP$$

where k_{brake} is the proportional coefficient mentioned above and BP (brake pressure) is the signal from brake pedal with unit variation's range. As for the acceleration path we can also have different shape about the function which relates the brake pedal and the total request.

Since there can be different values of gear ratio between front and rear we have to split the model into two parts.

Then, we can calculate the distribution of torque, on the base of the factors m_{front} and m_{rear}defined in the *paragraph 2.1.2*:

$$T_{brake,front} = m_{front} \cdot T_{brake}$$

$$T_{brake,rear} = m_{rear} \cdot T_{brake} = (1 - m_{front})T_{brake}$$

Now, splitting these quantities into the requests respectively to the motor and foundation braking, by the entry variable k_{regen} (between zero and one):

$$T_{electric,front} = k_{regen} \cdot T_{brake,front}$$

$$T_{electric,rear} = k_{regen} \cdot T_{brake,rear}$$

$$T_{friction,rear} = \left(1 - k_{regen}\right) \cdot T_{brake,rear}$$

$$T_{friction,rear} = \left(1 - k_{regen}\right) \cdot T_{brake,front}$$

We have notice that the factor k_{regen} could be different in general between front and rear but we will consider it the same for both axles, for reasons of simplicity.

The previous formulas do not consider the gear ratio of the reductors. We cannot ignore this possibility if we want to comply the requirement number 2. We shall to be able to control the effective torque provided to the wheel.

If we define GR as:

$$\mathrm{GR} = \frac{\omega_{\mathrm{motor}}}{\omega_{\mathrm{wheel}}}$$

To remove the dependency of the gear ratio, since the wheel torque is GR times T_{motor}, just divide the motor command by the same gear ratio as shown in the following illustration.

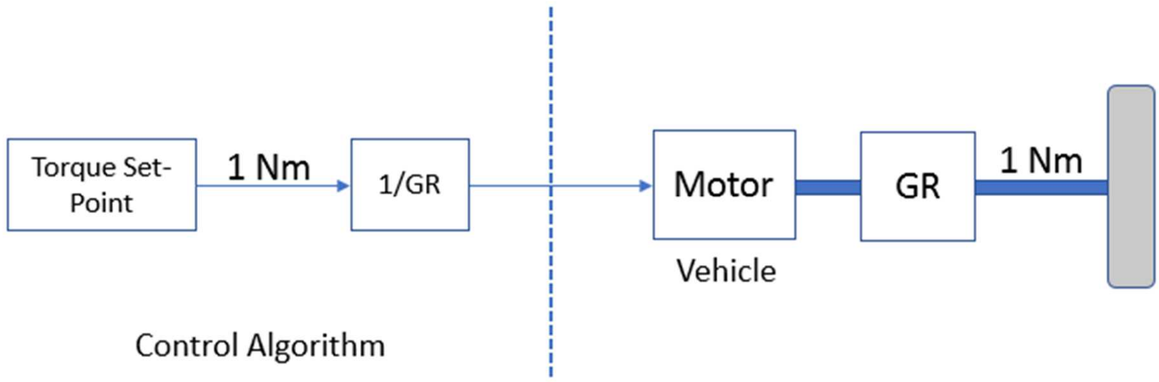


Figure 34 – How to make the control independent from the gear ratio.

After then we calculate the requests of electrical and friction braking torques by the discussed previous formulas. Keeping separately the sum of the electric torques from the sum of the friction torques, we have obtained two values of braking request which will managed by the control logic. All calculations are carried out considering as positive the values of torques.

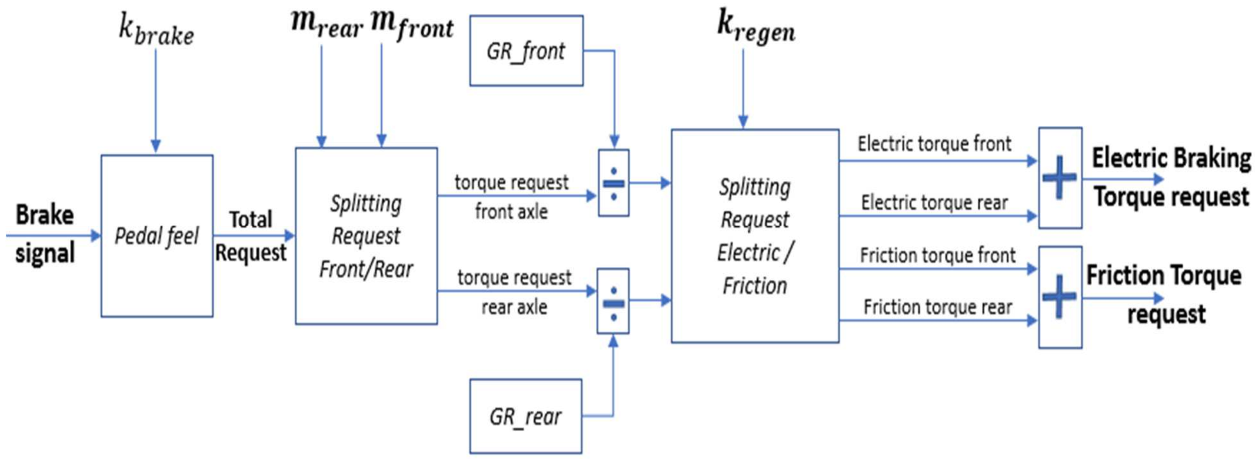


Figure 35 – Deceleration path: from brake signal to the request of electric and friction braking torque according to the split strategy and taking into account the gear ratio and the distribution factors.

3. *Control logic*

This block serves to manage all the output deriving from the functions described in previous paragraph. The requirements for this block are valid only for '*Drive Mode*'.

1. Electric driving features can be active according to the requirements summarized in following table:

		ACCELERATOR	
		1 (pressed)	0 (not pressed)
BRAKE PEDAL	1	only electric brake can be activated	coasting and electric brake can be activated
	0	electric drive can be activated	creeping or coasting can be activated

Table 4 – Electric driving features: allowed combination between accelerator and brake pedals

2. The sum between the requested braking torque to the motor and to the foundation brake shall be always equal to the total braking demand.
3. Braking torque shall be disabled when vehicle speed is less than certain threshold.

We focus now on the requirement number 2. This is easy to fulfil since we have to carry out only a subtraction between the total request and the actual electric braking torque even considering the friction torque deriving from the split strategy implemented in the deceleration path.
We have seen what the outputs of the deceleration path are. The proposed solution is shown in the following figure.

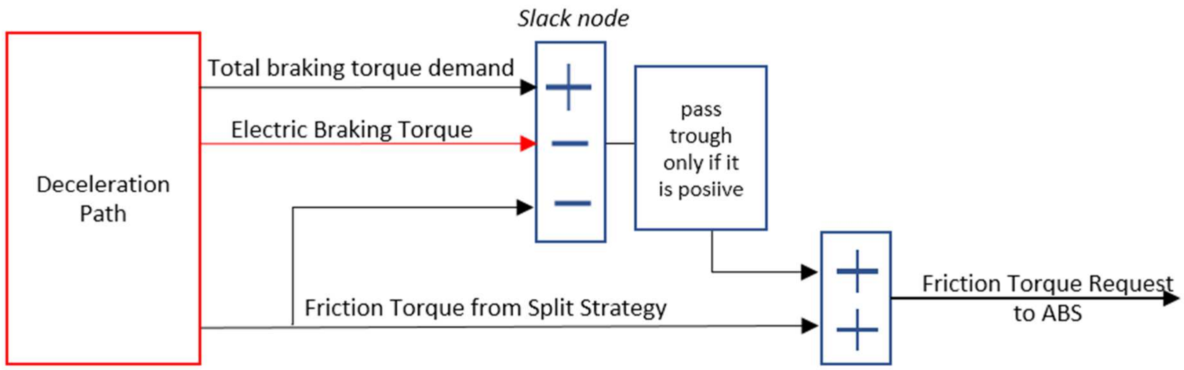


Figure 36 – Final computation of the friction torque request to the foundation braking system.

We must say that we consider all braking torque (as well as motoring torque) always as positive. The change of the sign will be carried out at a later time by the control logic.
The first requirement gives us the allowed combinations for the activation of the electric driving features. The flow chart in *Figure 37*, shows us the implemented solution to comply this requirement. Coasting torque and electric braking torque can be active together as we can see from the figure above, since a sum between the two is carried out. After then there is a condition to pass through this block: the vehicle speed has to be less than a certain threshold which we have set to 8 km/h.

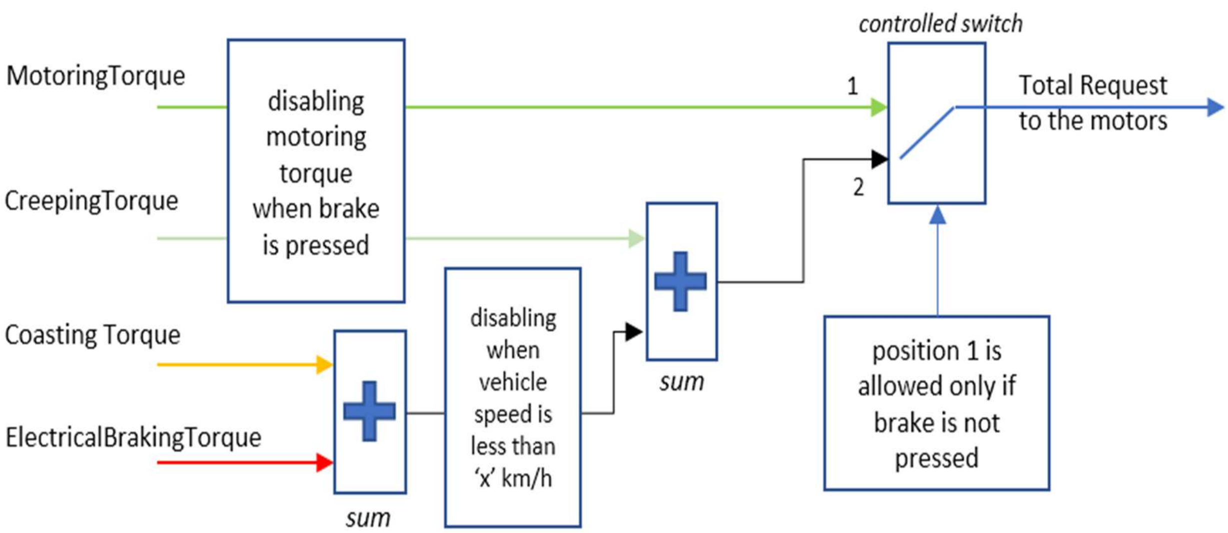


Figure 37 – Control logic: management of the function's outputs.

In turn the resulting value is summed with the creeping function output since the latter one is disabled when the brake is pressed. These operations are made because we want to control the requested torque by only one controlled switch. The condition to turn on/off the switch is pressing/not pressing the brake pedal. The logic is the following: if accelerator signal is greater than a certain threshold (to define) then input one passes through, otherwise passes input 2.

2.3.3 Torque distribution

We have already discussed about the torque distribution saying that we do not deal with the vehicle stability. There are no particular requirements for these functions aside from the computation of the defined coefficients *m_front* and *m_rear*.
Torque Distribution's block simply outputs the reference torques for the front and rear according with the calculated distribution factors. With regard the distribution of the torque to right and left wheels, we consider only the number of motors mounted on the axle to distribute proportionally the set-points value.

2.3.4 Power limiter

One of the Simulink blocks which plays a key role in the control of a variable is the Dynamic Saturation Block that allow us to saturate a variable as we want, giving to the upper and lower limits. These limits can be also variable in time; hence this block is called Saturation Dynamic. We can make a chain using these blocks if we want to manage more than one saturation of a variable. Obviously in this series of blocks the bottleneck is represented by the limits with the lowest absolute value. Dynamic Saturation's block is used here to saturate the request of motor torque in order to limit the output power. Power limiter is essential if we want to have a complete control about the delivered power both when it is positive (motoring) and negative (generator). The limitation of power is implemented by the mentioned Dynamic Saturation's block. Therefore, we have to produce the actual lower and upper limits of the requested torque, starting from the inputs of maximum positive and maximum negative power. Since the power is the torque times the rotational speed of the rotor, torque's limitations are obtained by a simple division between the allowed maximum power and the actual rotational motor speed. Thereby if the request of torque is higher than these limits, it will be truncated by the saturator. The computation of the upper and lower limits is carried out after the maximum values of the allowed power (both positive and negative) are split on the base of the coefficients m-front and m-rear. In this way, we will have four values, used to bind the range of the requested torques to each axle.

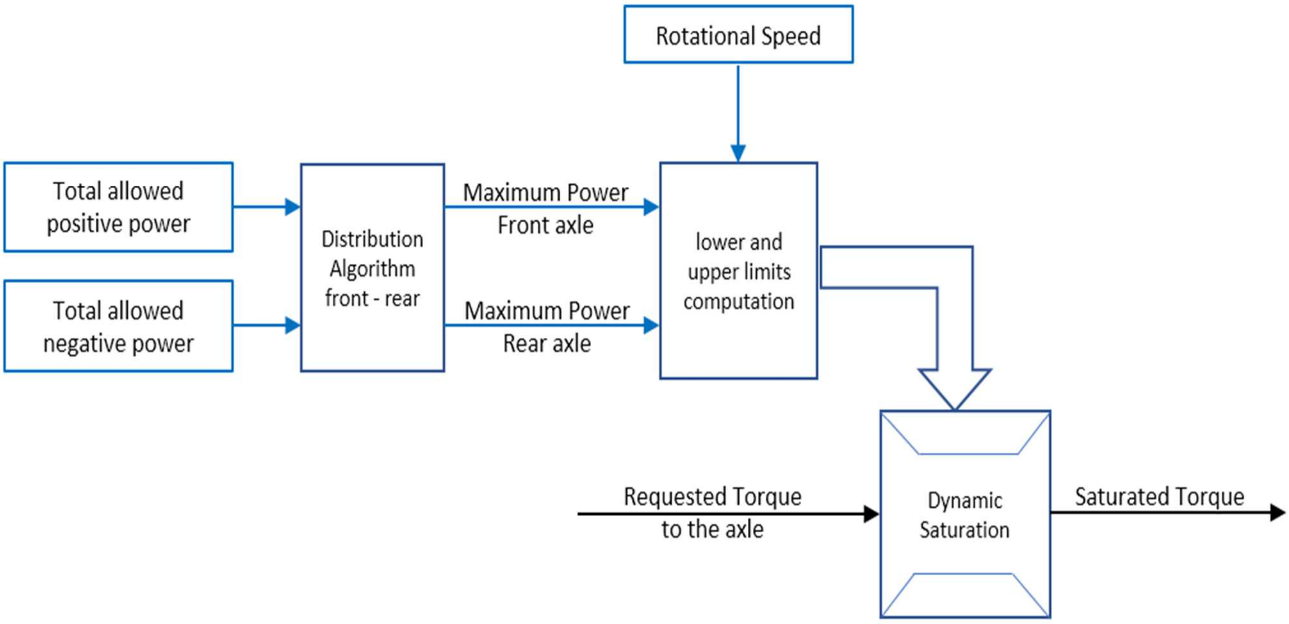


Figure 38 – Power limiter logic: the computation of the lower and upper limits before they are sent to the dynamic saturation block.

About the rotational speed of the motor we have supposed that it corresponds to the vehicle speed less than a proportional factor since we have neglected the tire's slip. Following it is shown an overview of the executions of the operations.

2.3.5 Rate limiter

The maximum variation's rate of the requested torque is another important basic feature of a torque control algorithm. There are many ways to implement this requirement, but the result is the same.
Usually when we talk about the slew rate we refer to the allowed variation in the time unit measured by Nm / s. However, in this thesis, we do not consider the maximum variation of torque in one second but in the time step unit of the 'interrupt service routine'. The effective variation of torque is also a function of the motor response.
A simple algorithm truncates the requested torque to a maximum value if the first one exceeds the second, otherwise no actions will be carried out. The truncated set point's value is valid for all the duration of the time step which we have fixed to 12 milliseconds.

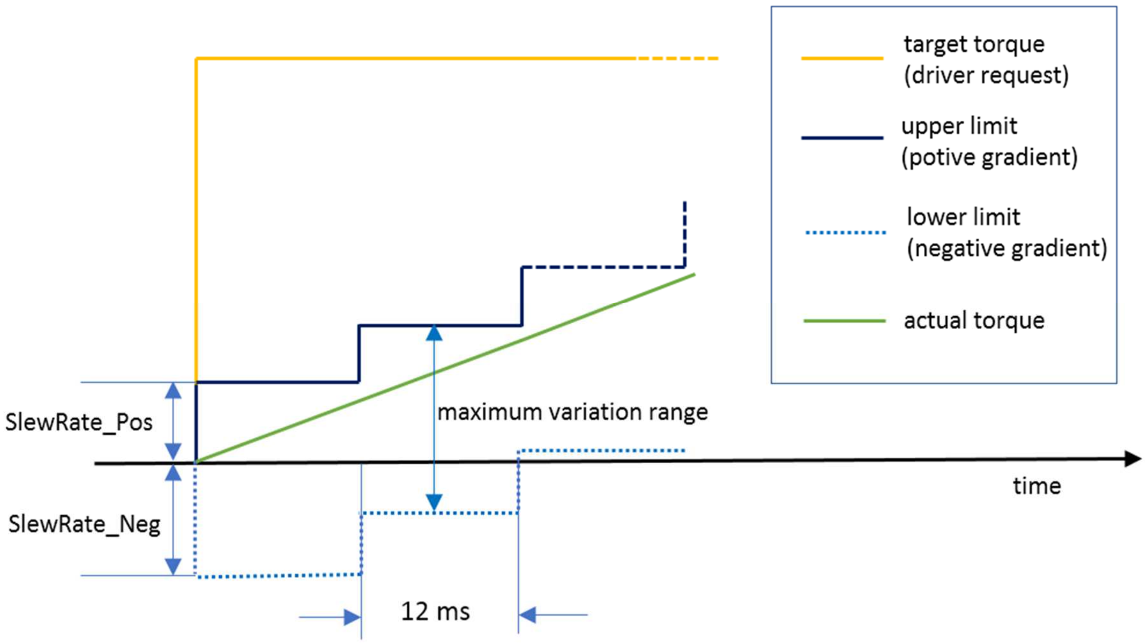


Figure 39 – Rate limiter logic. The maximum variation range is shifted for each time step according to the actual value of torque.

Similarly, the basic idea consists of using the Dynamic Saturation Block in order to make the lower and upper limits variable in time by adding (and subtracting) to the actual value of the torque, the maximum variation we want in that time step.

In the following figure, the maximum relative variation for positive and negative gradients are intentionally depicted different from each other just to say that we can set the value as we want.

The actual torque remains always within the lower and upper limits which changes at each iteration.

The input values named as '*SlewRate_pos*' and '*SlewRate_neg*', refer to the allowed variation respectively for positive and negative gradient. So, if the delivered torque increases (or decreases) the lower and upper limits consequentially change, following the actual torque.

2.3.6 Safety Functions

We have talked about safety requirements related to the powertrain's control in the pargraph. In particular, one requirement consists of delivering any deviations of torque between right and left drive wheels ever. This is valid for a control of the longitudinal drive, obviously. For example, we can have a situation where a motor reacts to the command later or earlier rather than the corresponding motor of the same axle. Therefore, we must think how to manage this eventuality.

The implemented algorithm is based on a simple but effective idea. Since the output of the Rate Limiter depends on the actual value of the torque, we shall consider the lowest value between the actual torques of the two axle's motors to calculate the upper and lower limits which shall be the same for both the motors. In other words, we set the requested torques equal to the minimum availability between the two. This is easy to implement.

Now we focus on the requirement about the coordination of the torque between front and rear. To understand what it means it can be helpful making an example. Suppose to have different torque availabilities between torque and rear axle so, when we pressed the accelerator different target torques are produced by the algorithm. Consequentially, if we consider the accelerator's step response, the time needed to reach the requested torques by the front and rear motors are different, if the slew rate is the same for both the axles. We can see it in following illustration.

When we shortly release the accelerator, according to the implemented logic, a negative reference of torque can occur because of the possible activation of coasting function. For this reason, electric motors have to change the corresponding working points from positive to negative torque's values.

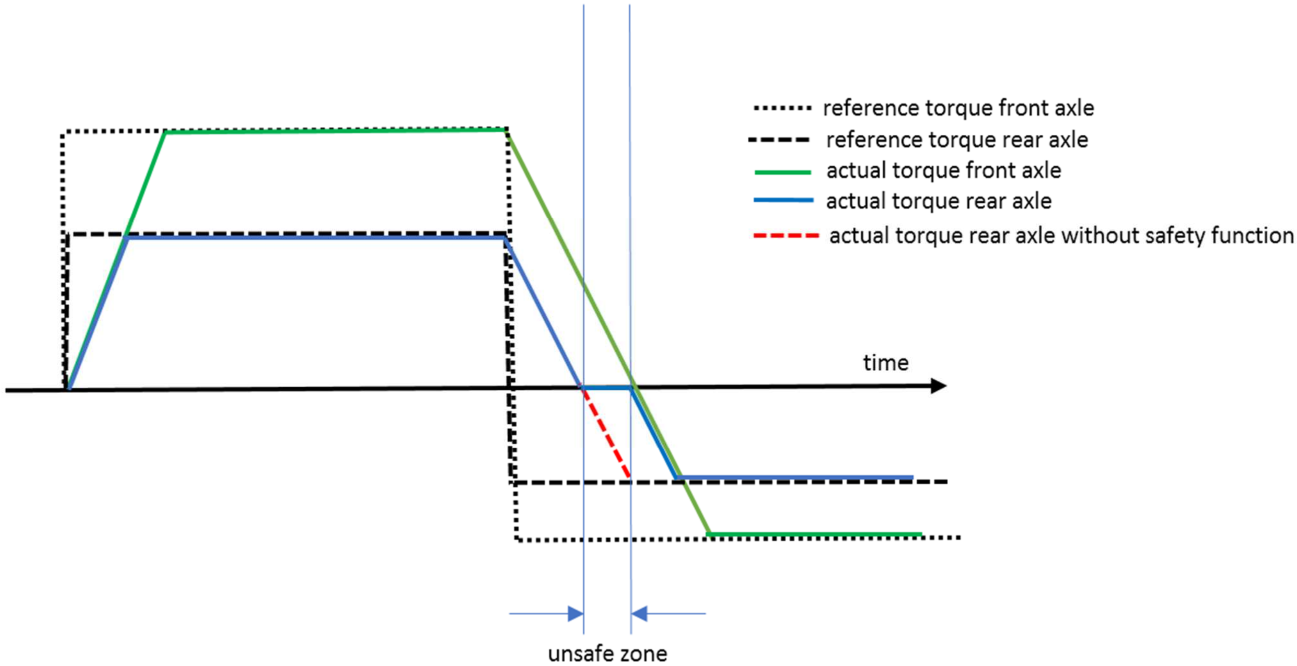


Figure 40 – An example of torque coordination between rear and front axle.

Referring to the *Figure 40*, we can note that there is a certain time's interval in which the torque's signs of the two axles can be different if we do not carry out any action to prevent that. This is because the torques start to decrease from different values with a certain value of slew rate which we have supposed to be the same for all motors.

To avoid this potential unsafe situation, we could change the slew rate of one axle in order to get the synchronicity between the two. However, implement this kind of algorithm, could bring further problems to solve. Therefore, it is proper to implement a safety mechanism which allows to hold the

torque which intersects first the time axis, until the remaining torques match the zero. Then the actual torques will be synchronized again.
Like in previous functions, we will use the '*Dynamic Saturation'* block to control the variation range of the variable according to the requirement which have to comply. We note that the upper limit is the minimum values of the available torque between right and left drive wheel as we have said.
The lower boundary instead is managed in another way. The operations which are carried out, hold the lower limit to zero until the torque of the other axle become less a certain threshold. The pseudo-code for this requirement is the following:

% for the lower limit

> **if** [***max*** (***abs*** (MinActualTorque_Front), ***abs*** (MinActualTorque_Rear)) > threshold]
> Low_Lim = ***min*** (AvailableTorque_Right; AvailableTorque_Left)
> **else**
> Low_Lim = 0;
> **end**

It is important to say that this algorithm is valid for those powertrain configurations which have more than one active axle, otherwise it does not make sense to carry out the above algorithm.
To manage all the powertrain configurations a simple logic to choose the proper algorithm is based on a controlled multi-port switch.

3. Model in the loop

3.1 **Model implementation**

3.1.1 Simulink model overview

Torque path receives from Input Generator (IG) the signals coming from the VCU and from each inverter-motor (IM); events layer triggers the initialization event and provides the frequency of the execution of the algorithm (set 12ms). Each narrow from IG to Torque Path (TP) is a signal bus which carries the informations related to the specific ECU. Same consideration can be made for the block named 'Simulation Results'. The output are the four set-points of torques plus the signal to the braking system. The following picture show us the highest point of view of the model.

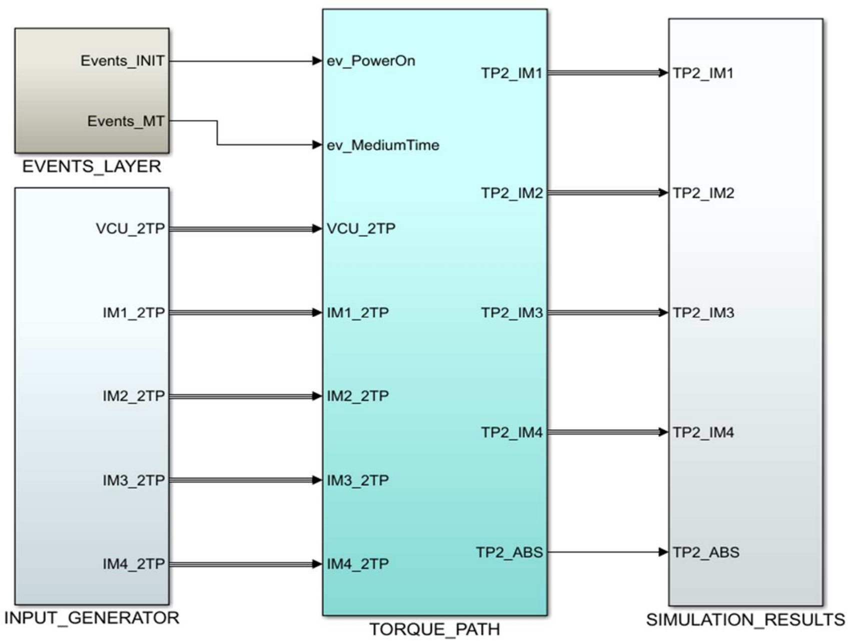


Figure 41 – A global overview about Torque path and its auxiliary blocks.

In software modelling, a common practice to manage all the functions of the system consists of defining the hierarchical structure of the layers which make up the model. Each layer corresponds to a specific function and each subsystem has to be tested regardless from other block. In this work we focus on testing of the main block (TP) in which there are all the described functions.
The layers are organized to get a good compromise between the degree of details and the need to visualize the variables and blocks properly.

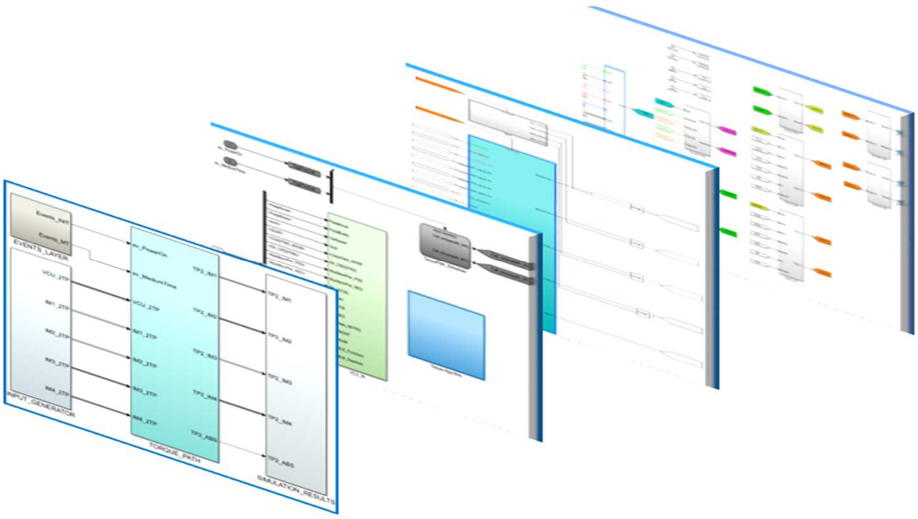


Figure 42 – Torque Path layers.

Starting from the highest level, the first sub-level handles the input and output as well as the 'function's calls' by using the scheduler. Then we have the layer where the signals coming from the 'initialisation' and 'medium time' tasks are merged, as described in *paragraph 2.3.5*. Inside the 'medium time' block we have all the main functions of the algorithm. The first link in the chain is the block which converts the inputs of driver into the overall requests of motoring and braking as we have said. The first sub-layers of this block manage the output according to the engaged gear. To carry out this operation we have used a multi-port switch controlled by the variable 'Gear' which selects the corresponding out-port. The last layer includes all the described functions.

3.1.2 Physical layer

The mathematical model used to simulate the vehicle response was discussed in the *paragraph 2.2.1*. Following, it is shown an overview of physical layer where we can see the position of the implemented model of the vehicle. On the left we have the requested torques which enter in the 'first order motor-model'; after then the output torques are sent as feedback to the control algorithm (we have considered also the possibility to add a white noise if we want to make the feedback torques more realistic).

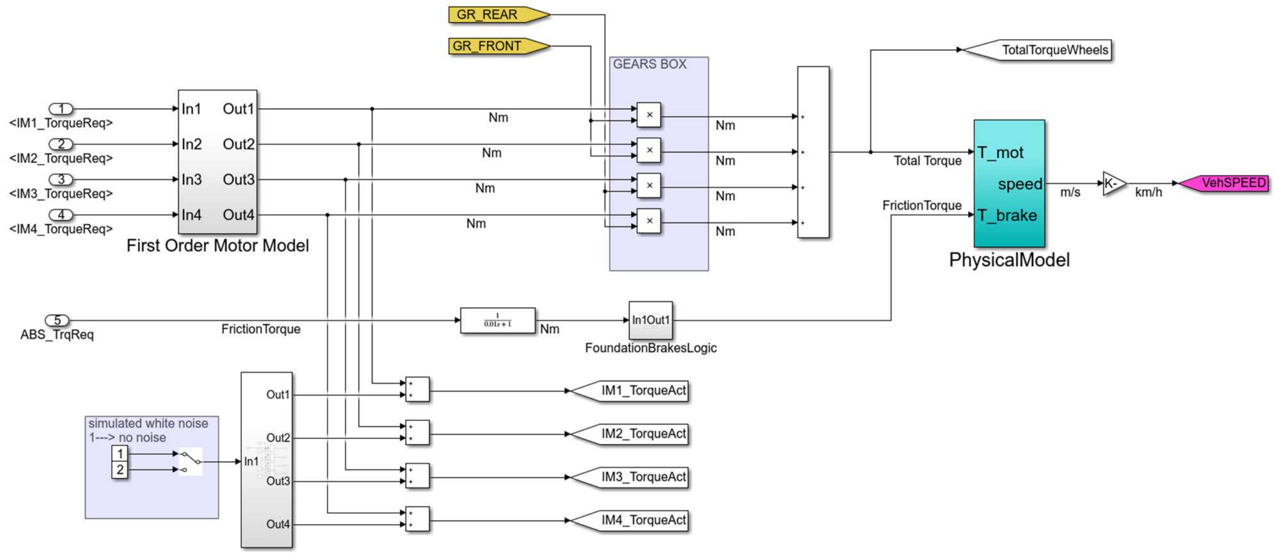


Figure 43 – The implemented physical model and its corresponding layer by Simulink.

To obtain the effective torque delivered to the wheels we have to carry out the product between the output motor torque and the corresponding gear ratio. The motor torques are then counted together, finally sending the resulting value along with the friction torque to the vehicle model.
About the block named 'Foundation Brakes Logic', it only serves to change the sign of the braking torque as soon as the vehicle speed changes its direction according to the physic principle.
The implementation of the vehicle model is shown in the following picture:

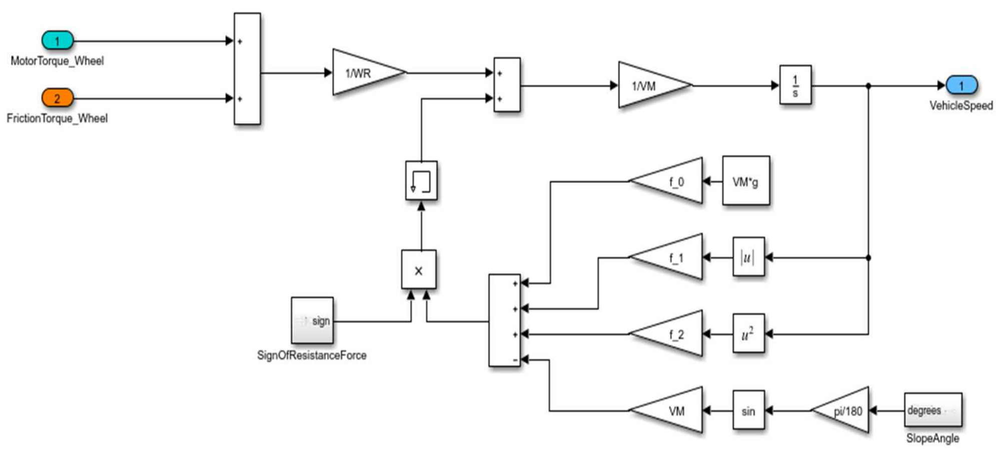


Figure 44 – Implementation of the vehicle model in Simulink.

Where 'WR' is the wheel radius, 'VM' is the vehicle mass, 'g' the gravitational acceleration.

The coefficients of friction are f_0, f_1, f_2. Before the sum of resistance forces are algebraically summed with the resulting propelling force, there is a block which changes the sign of the resistance force as soon as the vehicle speed changes direction. A ‘memory’ block is also added to solve problems about running the simulation.

3.2 **General purposes**

3.2.1 Settings

To define all the constant of the simulation as well as the powertrain’s configuration, we use a MATLB script by which we can set the number of the motors for each axle and their torque-speed characteristic by giving some points of the torque’s curve mentioned in the *paragraph 2.2.2*.
The constants are divided into two categories: simulation’s parameters and Torque Path’s parameters which are summarized in *Table 5*.
These values, which shall be set before running a simulation, are only the most important. There are other constants and thresholds which have to be set. For example, about the strategy of filtering the pedal’s signals, we shall to fix the shiftable values of the zero position of the accelerator.
Regarding the Input Generator, we generate the inputs by using common Simulink Block in particular called ‘Signal Builder’. The latter one allows us to create interchangeable groups of piecewise linear signal sources defining the output waveforms, variable in time as we want. We can also quickly switch the signal groups into and out of a model to facilitate testing. Accelerator and brake signals are built by this block.
Referring to the *Table 5* we note that, among the simulation parameters, there are also four time-constant corresponding to the transfer function which simulates the motor response. Therefore, we have the possibility to set different time constants in order to verify whether the algorithm behaviour complies with the discussed safety requirements.
The wheels radius and the vehicle mass are set equal respectively to 0.4 m and 2000 kg for all the simulations.

Torque Path parameters		
number of motors - front axle	integer	-
max torque - front motors	float	Nm
base speed - front motors	float	rpm
max speed - front motors	float	rpm
number of motors - rear axle	integer	-
max torque rear motors	float	Nm
base speed - rear motors	float	rpm
max speed - front motors	float	rpm
max braking friction torque	float	Nm
max creep torque	float	Nm
max coast torque	float	Nm
creeping speed	float	km/h
min vehicle speed for electric braking	float	km/h
Simulation parameters		
simulation time	float	s
time step simulation	float	ms
gravity acceleration	float	m/s2
rolling resistance	float	m
air drag	float	N s2/m
wheel radius	float	m
vehicle mass	float	kg
time constant motor 1	float	s
time constant motor 2	float	s
time constant motor 3	float	s
time constant motor 4	float	s

Table 5 – Reference table for parameters used to test the algorithm

About the creeping speed, it should be around to 10-15 km/h while the minimum speed to deliver negative torque (recuperation) is usually not less than 7 km/h.

Another important thing to say, is about the simulation settings. All the following simulation are carried out with a time step calculation of 1 millisecond while the control task is triggered each 12 milliseconds. This is good compromise between the duration time of the simulation and calculations accuracy.

Rolling resistance and air drag are set to the average values for passenger cars, because are not so important, since the purpose of the simulation is only to test the algorithm.

3.2.2 Test paths

To test the functions of an algorithm, control engineers usually resort to a ‘test path’ which consists of testing a single algorithm’s component, giving to it a set of inputs and verifying whether the outputs correspond to the expectations. Usually test path is used to verify a single requirement while in our case we will carry some tests, each of which serves to verify more than one requirement. This is because the number of possible tests we should carry out, is huge, since each function's requirement has to be verified for each powertrain configuration with different environment condition as well as load levels. Therefore, we will focus on the most important aspects only. The aim is to display the results of simulations to check the plausibility of the model. To do this, we will carry out some significant experiments to put in relevance the key-points about the requirement which we have analysed.

In this way, we will refer especially to the configurations with four and three motors, because they are the most significant case in terms of complexity. The tests we will carry out, correspond to different patterns of driving cycle which are the result of driver's input. Following, a quick description for each type of driving pattern is given.

1. 'Full acceleration to Full deceleration' test.

   This test consists of a simple acceleration and deceleration in forward direction by using two consecutive steps for the pedal's inputs. Basically, this test allows us to verify some important requirements about the rate limiter's behaviour, distribution of torque between front and rear axle, regenerative braking, split's braking strategy.

2. 'Full acceleration to Coasting' and 'Coasting to Creeping' test.

   Coasting and creeping functions are active in this case; this test consists of releasing gradually the accelerator during the drive; below a certain threshold of the accelerator position we expect a braking torque which decelerates the vehicle until the speed reaches the creeping velocity which is maintained by providing motoring torque without pressing the accelerator.

By mixing these two driving patterns we are able to validate all the described function such as power limitation, axle shut down, torque's compensation and other. About the safety function we will test the algorithm's behaviour when an axle's shut down occurs during a constant drive. Even the coordination of torque between front and rear axle will be checked. All the following tests are carried out setting the slope's angle equal to zero.

3.3 Test n. 1: reverse mode

3.3.1 Requirements

We start from testing the algorithm when reverse speed is engaged because is the simplest case. We notice that reverse mode and drive mode represent the first link of the chain (torque path) and it calculates only the total request of torque. This request is then managed by the next blocks which distribute the torque according to the implemented functions. The requirements for this this block are defined in the *paragraph 2.3.1*. Following a brief recap about the main features:

1. Accelerator signals shall be converted into a number between zero and one according to the implemented look-up-table (about the latter there are no requirement). The resulting variable shall be multiplied by the availability of negative torque.
2. The input signal shall be converted into a request of friction torque by a simple product between the maximum friction torque and the percentage of the request.
3. Delivering positive torque is not allowed. Negative torque shall be disable when the vehicle speed reaches a shiftable threshold.
4. Negative motoring torque shall be disable when brake signal is greater than a certain value (we set it to 0.1).

The implemented solution is shown in the picture below:

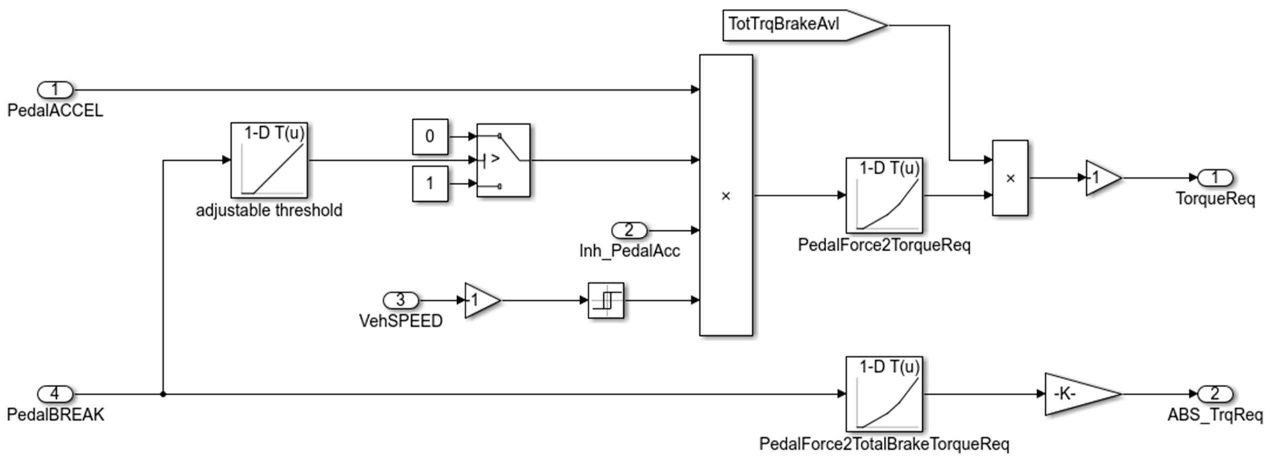


Figure 45 – Reverse mode: acceleration and deceleration path implementation.

To verify whether the requirements are met, we carry out a simple test by only giving a step of accelerator without pressing the brake.

In order to focus on the requirements of this block, we deactivate all the involved function such as rate limiter and power limitation. We carry out a simulation with only one motor.

3.3.2 Simulation parameters

In the following table are gathered together all the inputs for the simulation:

GENERATED INPUTS		***unit***
engaged gear	**R**	-
k_regen	(not allowed)	-
maximum mechanical power - positive	inf	kW
maximum mechanical power - negative	inf	kW
slew rate - positive	inf	max Nm/timestep
slew rate - negative	inf	max Nm/timestep
gear ratio - front	10	-
gear ratio - rear	0	-
coasting mode	**not allowed (0)**	-
enable creeping	**not allowed (0)**	-
reduce power rear axle	1	-
reduce power front axle	0	-
TORQUE PATH PARAMETERS		
time step	12	ms
number of motors - front axle	**1**	-
max torque - front motors	100	Nm
base speed - front motors	maxspeed*30%	rpm
max speed - front motors	10000	rpm
number of motors - rear axle	0	-
max torque rear motors	0	Nm
base speed - rear motors	0	rpm
max speed - front motors	0	rpm
max braking friction torque	15000	Nm
max creep torque	not allowed	Nm
max coast torque	not allowed	Nm
min vehicle speed (negative torque)	8	km/h
creeping speed	not allowed	km/h
max reverse vehicle speed	**40**	**km/h**
SIMULATION PARAMETERS		
simulation time	**80**	**s**
time step simulation	1	ms
rolling resistance	0.02	-
air drag	0.3	N s2/m
wheel radius	0.35	m
vehicle mass	2500	kg
time constant motor 1	**0.1**	**s**
time constant motor 2	-	s
time constant motor 3	-	s
time constant motor 4	-	s

Table 6 – Simulation parameters for test n.1.

3.3.3 Simulation results

The variables we have chosen to display for all the simulations are:

- Total electric power in kW (light green) which is the sum of each power delivered or absorbed by motors. In this case we have only one motor.
- Vehicle speed in km/h (black) as result deriving from the physical model
- Accelerator and brake signals (dark green and orange) represent the driver input in percent of pedal's pressure.
- Total motor torque in Nm (blue).
- Friction torque in Nm (red).
- The dashed lines stand for the theoretical accelerator and brake demand. In other words, it is the product between the pedal signal (%) and the corresponding maximum allowed torque.

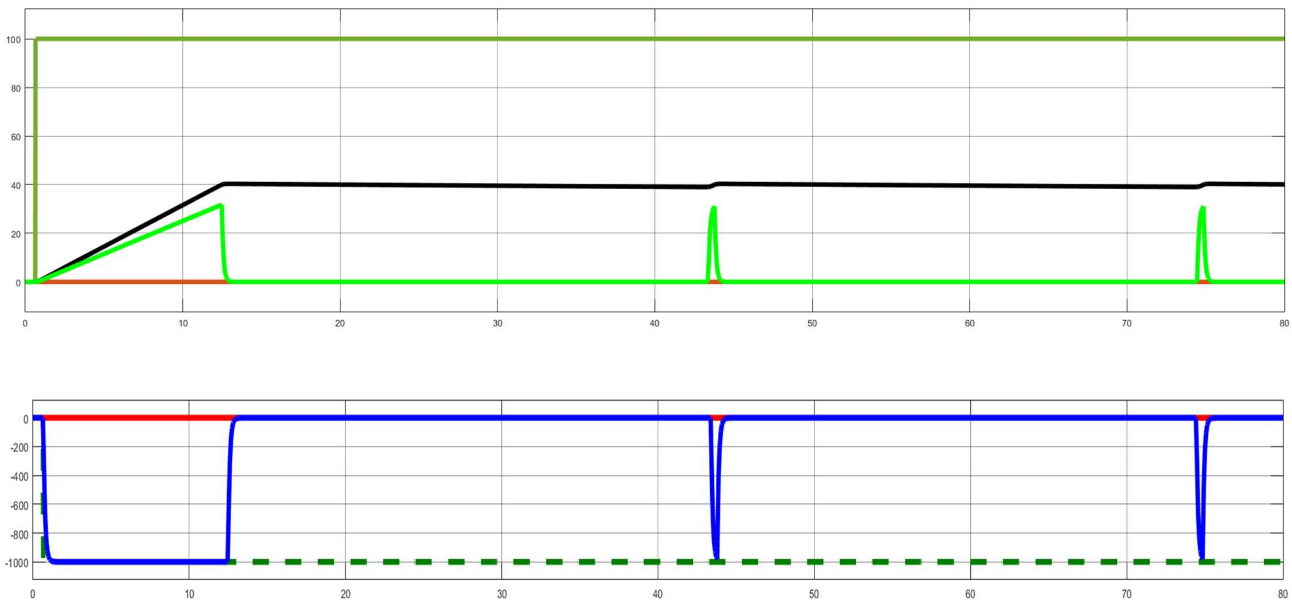


Figure 46 – Simulation results: test n.1.

The torque delivered (blue) is negative as well as the speed (even if we see it as positive). We can see the effect of the relay about the speed limitation. The vehicle speed not exceed the maximum velocity we have imposed. Since the rate limiter is no active we have a step of reference for the motor which responds to the command with a time constant of 100 ms.

Brake pressure is not actuated in this test because during reverse speed only friction braking can be active. We will show the braking phase in the next test when the electric braking torque cooperate along with the foundation braking. This is allowed only for Drive mode.

3.4 Test n. 2: drive mode

3.4.1 Requirements

The following test path is referred to the basic requirements of the algorithm for 'drive mode' which are summarized in the list below.

1. Activation of the electric driving features has to comply with the *Table 4*.
2. The sum between the electric braking torque and friction torque has to be equal to total braking demand.
3. Electric braking torque shall be disabled below a certain threshold of speed.

The implemented model which fulfilled the requirements 1 and 3 is shown in following picture:

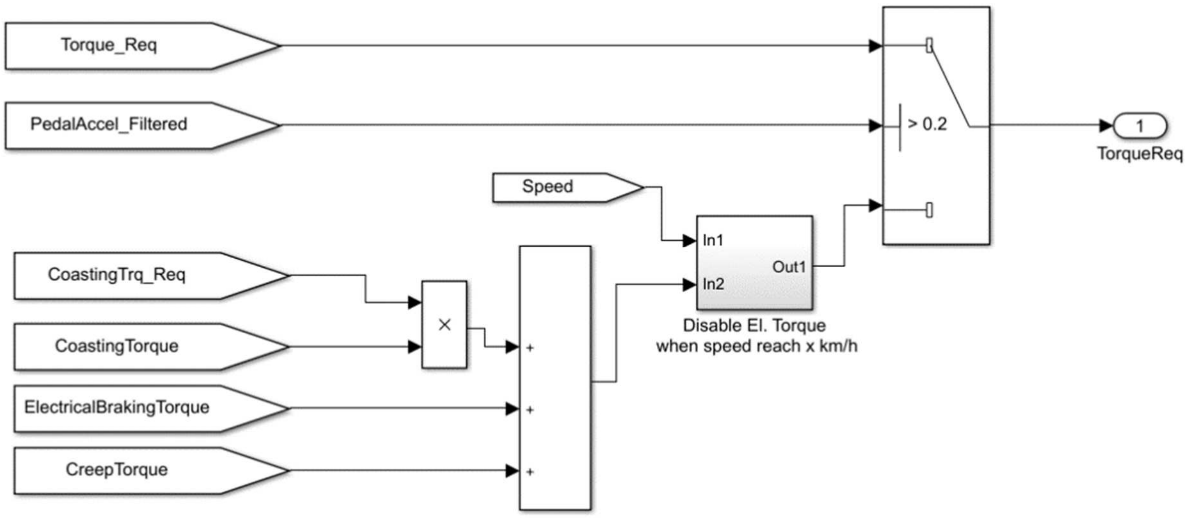


Figure 47 – Drive mode: control logic implementation

We can recognize the logic we have described in the previous chapter as well as the switch controlled by the percentage of accelerator pressure we have set to 0.2.

The next picture instead refers to the compliance of the second requirement as described in the *paragraph 2.3.1*.

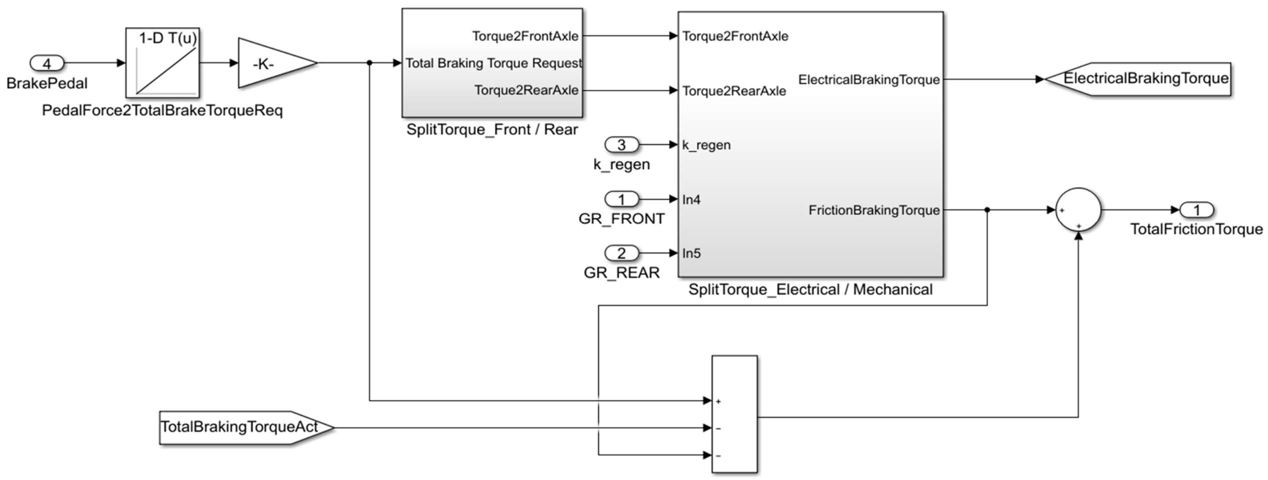


Figure 48 – Drive mode: deceleration path and computation of the finale request of braking friction torque.

Regarding the rate limiter, its implementation is made according to requirements discussed in the previous chapter. We can see from the next picture that the requested torque (referred to each axle) is saturated by the upper and lower limit which are respectively calculated by the sum between the actual torque and slew rate (positive) and the subtraction between the actual torque and the slew rate (negative).

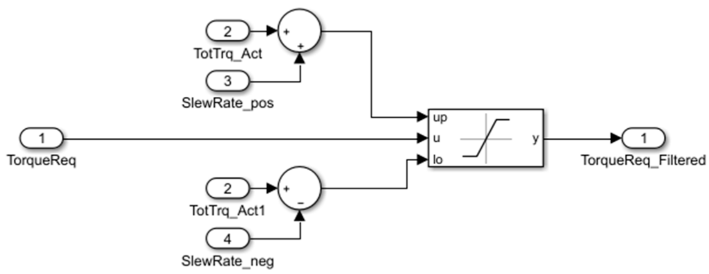
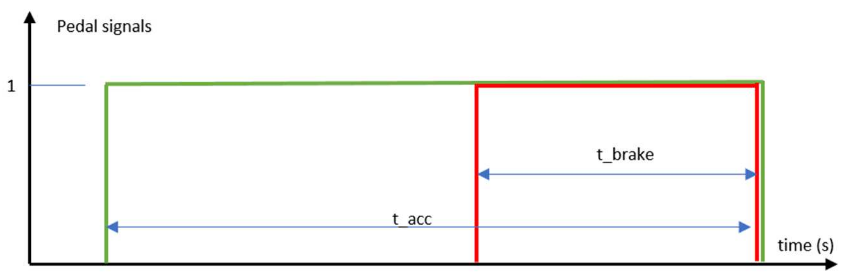


Figure 49 – Rate limiter implementation.

In this test, coasting and creeping are disabled in order to focus on the main functions of the controller, so the corresponding outputs are equal to zero.

We will make a test using the following driver input's pattern:



In green we have the accelerator's signal and its corresponding duration time. These considerations are valid for the brake's signal, too. It is clear that the figure above cannot correspond to a real situation, since it is not possible to have an ideal step of accelerator or brake but the aim here is only to test the algorithm's behaviour. By this test we are able also to verify the safety requirement about the priority of the brake pedal. So, we can expect that if accelerator and brake pedal are pressed simultaneously, accelerator pedal shall be disabled.

3.4.2 Simulation parameters

For this simulation creeping and coasting are disabled in order to focus on the main features. Regarding the split strategy we set the value of k_ regen is set to one, corresponding to the maximum allowed electric braking we can request. It is clear that the latter one depends on the function implemented by the look-up-table. In *Table 6*, the most important settings for this test are shown.

GENERATED INPUTS		***unit***
engaged gear	**D (drive)**	-
k_regen	**1**	-
maximum mechanical power - positive	inf	kW
maximum mechanical power - negative	inf	kW
slew rate - positive	5	max Nm/timestep
slew rate - negative	5	max Nm/timestep
gear ratio - front	10	-
gear ratio - rear	10	-
coasting mode	**0 (disabled)**	-
enable creeping	**0 (disabled)**	-
TORQUE PATH PARAMETERS		
time step	12	ms
number of motors - front axle	**2**	-
max torque - front motors	100	Nm
base speed - front motors	maxspeed*30%	rpm
max speed - front motors	10000	rpm
number of motors - rear axle	**2**	-
max torque rear motors	100	Nm
base speed - rear motors	maxspeed*30%	rpm
max speed - front motors	10000	rpm
max braking friction torque	15000	Nm
max creep torque	50	Nm
max coast torque	disabled	Nm
min vehicle speed (negative torque)	8	km/h
SIMULATION PARAMETERS		
simulation time	**40**	s
time step simulation	1	ms
rolling resistance	0.02	m
air drag	0.3	N s2/m

wheel radius	0.4	m
vehicle mass	2500	kg
time constant motor 1	**100**	ms
time constant motor 2	**100**	ms
time constant motor 3	**100**	ms
time constant motor 4	**100**	ms

Table 7 - Simulation parameters for test n.2.

3.4.3 Simulation results

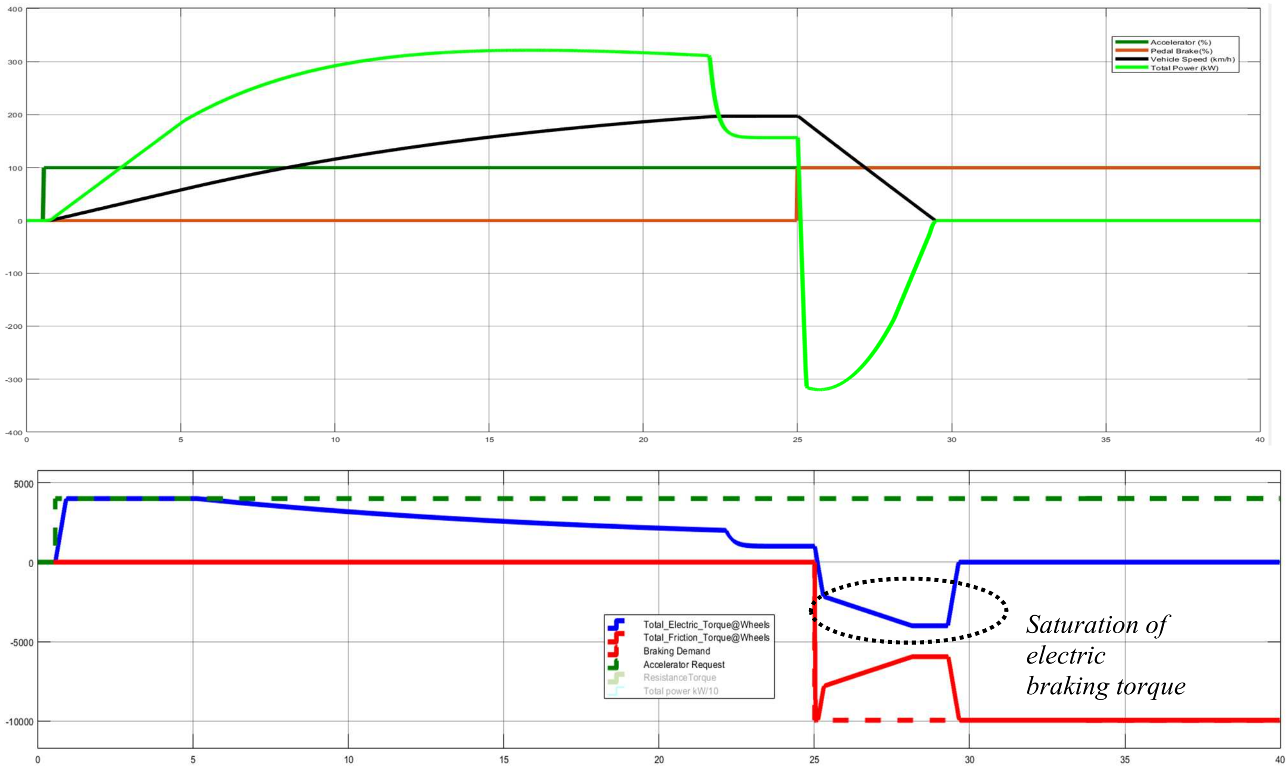


Figure 50 - Simulation results: test n.2.

We note that the power (in kW) increases when accelerator is pressed, until the resistance force become equal to the motoring torque (the equilibrium between the forces is reached around 200 km/h).

The maximum torque is 4000 Nm for motoring as we expect. Looking at the results, we can say that the requirement number one, is partially fulfilled. In fact, when the brake pedal is pressed, accelerator signal is overridden. The requirement number 2 is met, too. We can verify this if we look at the symmetry of the red and blue curves which are complementary. Indeed, if we have a step of braking torque, we expect that the vehicle speed constantly decreases until it reaches 0 km/h as we can see in first graph. The saturation of electric torque is compensated by the foundation brakes in order to maintain the braking rate related to the driver's braking demand as demanded by the ECE-13H regulation.

With regard to the third requirement, we need to zoom in the region of interest, when vehicle speed approaches the zero. We must verify that when the vehicle speed goes below the corresponding input value, electric braking torque is not delivered. This requirement is needed not only because the recovered energy at low speed is not relevant but also for safety reasons. It is important to say that we have to verify only whether the algorithm's response works correctly. We are not interested in the calibration of the settings or parameters. These latter have to be set according to the vehicle specifications and the behaviour we want.

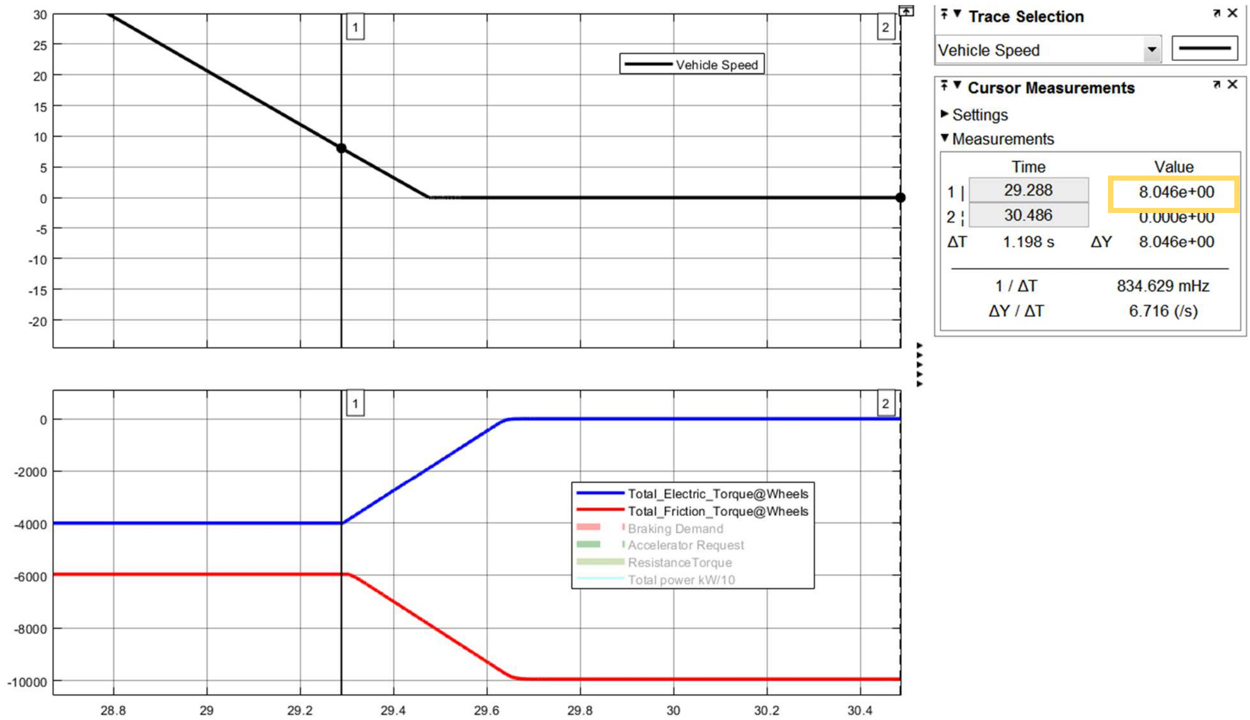


Figure 51 – A focus on the electric (blue) and friction (red) braking torque when the vehicle is coming to a stop.

We note that the value of speed corresponding to the deactivation of regenerative braking, is approximately 8 km/h, how we expect. Finally, we can see that, when electric torque is going down to zero, there is a time interval where the vehicle speed is not equal to zero yet. We don't have to bother about this because we can change the slew rate as we want. If we increase the value of negative slew rate the torque will reach the zero before the vehicle stops, in order to avoid an unintended negative torque. Obviously, this is possible only if the motor response is enough fast to decreasing before the vehicle is stationary.

3.5 **Test n. 3: coasting and creeping**

3.5.1 Requirements

In this paragraph we focus on testing creeping and coasting functions. The requirements for them are summarized in the following list:

1. Activation of the electric driving features has to comply with the *Table 4*.
2. Coasting torque has to be delivered according to the implemented function and it has to depend on the accelerator position as described in *paragraph 2.3.2*.
3. Creeping torque has to be delivered according to the function described in the *paragraph 2.3.2*
4. Both Creeping and coasting functions become no active below a certain threshold.

To select the coasting strategy, we use a simple multiport switch controlled by the variable 'MODE' as shown in the following picture:

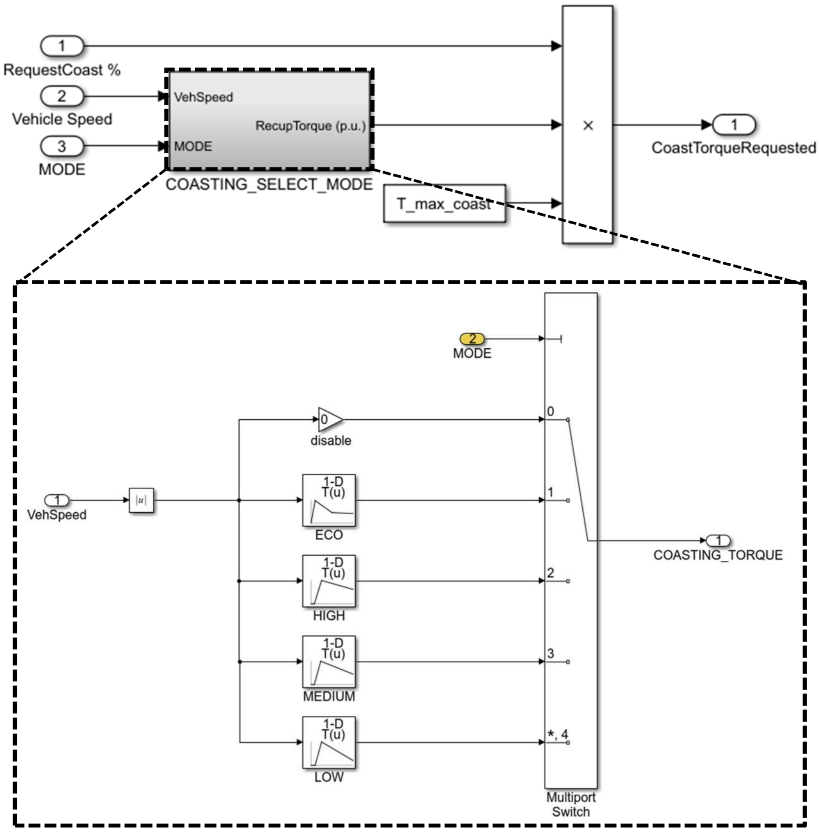
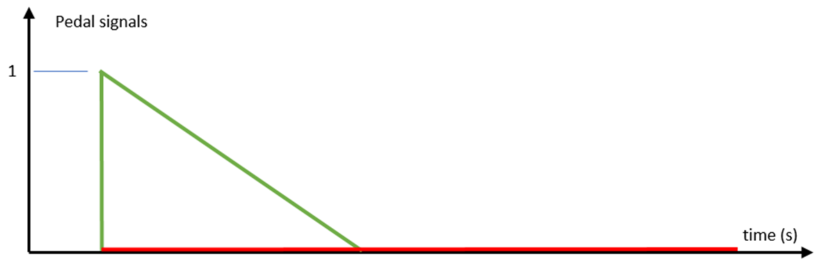


Figure 52 – Coasting function implementation

The functions implemented by the look-up-table of each level are exemplary curves and they are designed with reference to the diagram shown in the *paragraph 1.2.5*.

To test these functions, we carry out a test giving to the algorithm the following acceleration pattern:



Brake is not pressed in this case while acceleration pattern is like a saw tooth: in this way we can see if the requirement about the accelerator position and the corresponding request of motoring or coasting is fulfilled.

3.5.2 Simulation parameters

For this test, all the settings of the previous simulation remain unchanged, except the following:

- Positive slew rate is changed from 5 Nm to 1 Nm. This corresponds to reduce the dynamic's response during acceleration phase. Negative slew rate is unchanged.
- Creeping is enabled by set to 1 the corresponding bit.
- Coasting function is enabled by setting the value named 'MODE' to the second position corresponding to the maximum allowed energy's recuperation.

3.5.3 Simulation results

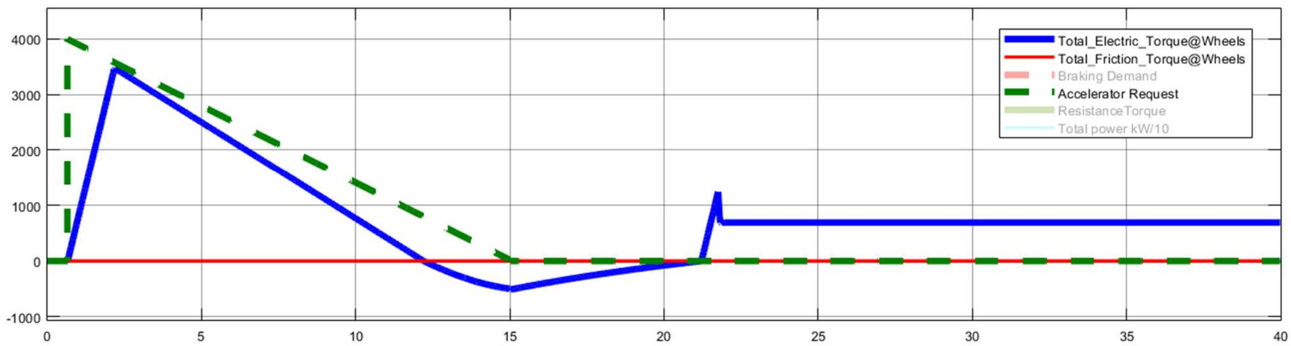


Figure 53 - Simulation results: test n.3

Starting by analysing the effect of rate limiter we can say that it acts in a proper way. Finally, the effective variation rate of the torque depends on the motor response, too.

By this test we are able to verify the requirements previously described. Firstly, we note that the accelerator position is related to the coasting torque as we expect: in fact, from a certain value of accelerator position (set to 0.2), torque changes its sign becoming a negative torque. From this point, without pressing pedals, the coasting torque still continue to decelerate the vehicle until the creeping speed is reached. The latter one is set to 15 km/h. We note that to maintain this velocity, a sort of torques 'overshoot' occurs but this does not represent a problem.

3.6 **Test n. 4: split strategy and power limitation**

3.6.1 Requirements

Now we focus on the requirements related to the power limiters and split braking strategy which are summarized in the following list:

- The algorithm has to be able to set a certain ratio between the electric braking torque and the overall braking demand. This is valid until the motor torque reached its maximum allowed value.
- The power limiter has to restrict the delivered power and the absorbed power, according to their corresponding input values.

To comply the latter requirements the solution described in the *paragraph 2.3.4* are implemented in this way:

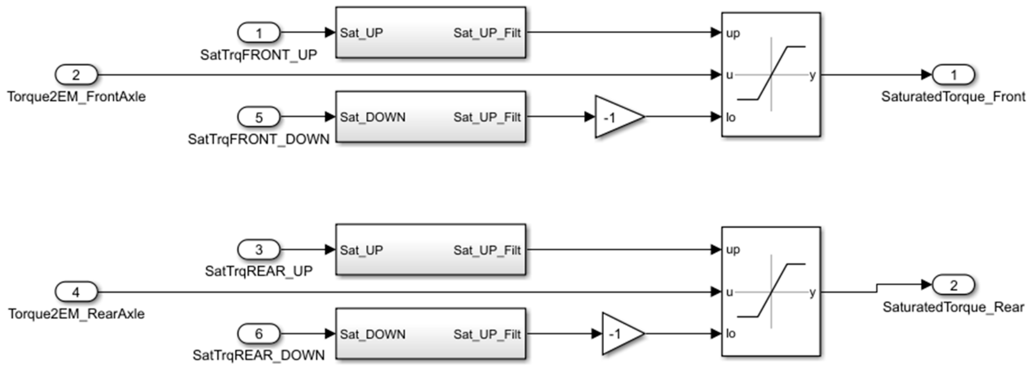


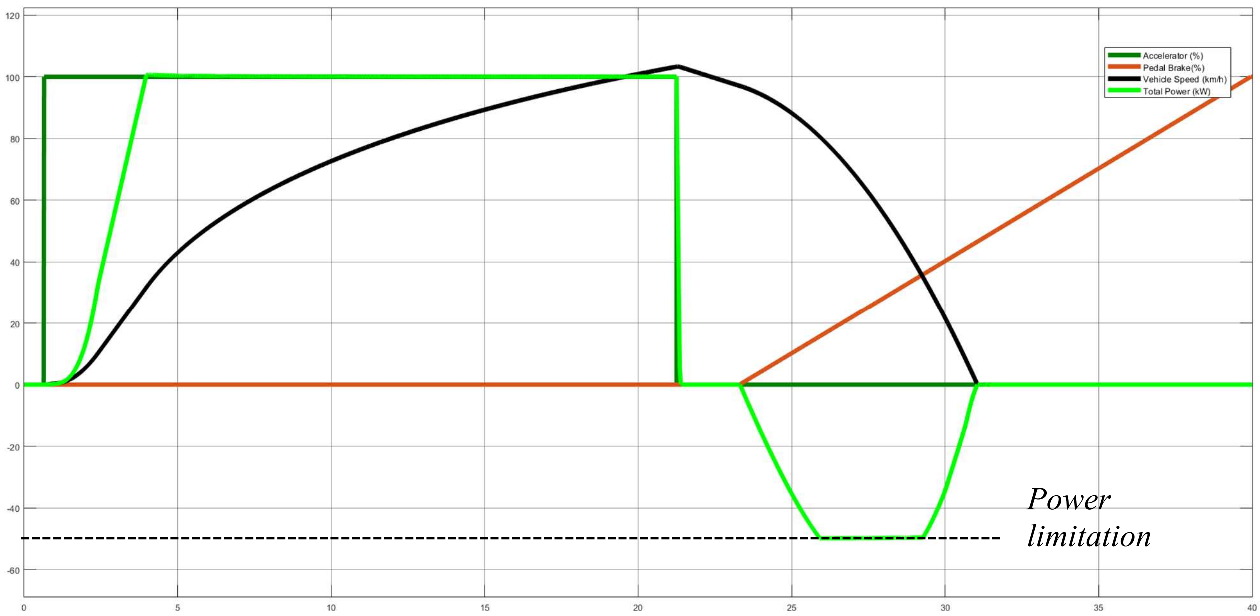
Figure 54 – Power limiter implementation

3.6.2 Simulation parameters

The settings are the same of the previous test path, excepting for the following input.

k_regen	**0.5**	-
maximum mechanical power - positive	**100**	kW
maximum mechanical power - negative	**50**	kW

3.6.3 Simulation results



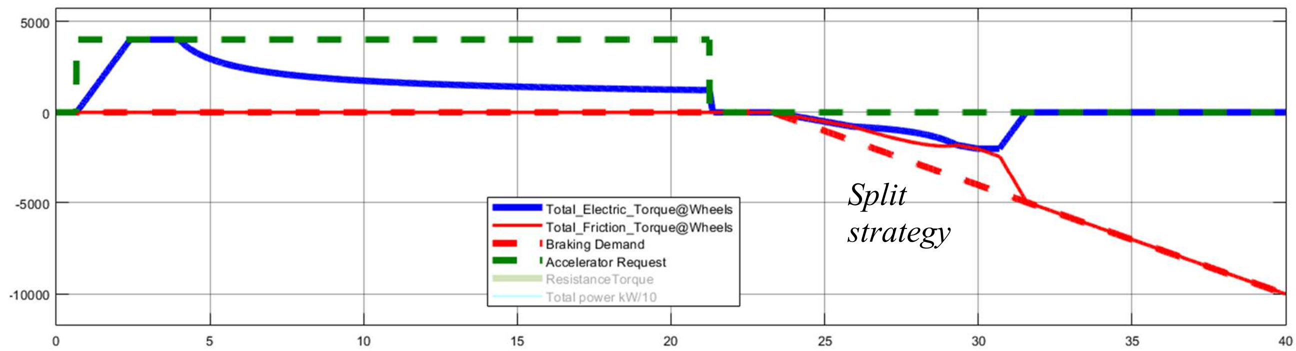


Figure 55 - Simulation results: test n.4

Looking at the results we can say that the power limiter fulfils its task both for negative and positive power limiting the total delivered torque when the power become equal to 100 kW for positive power and 50 kW for negative power. The split strategy is also successfully implemented: since the value of k_regen is set to 0.5 we expect that for a costant increase of braking demand, the latter one is split in equals parts. We can see indeed that at beginning of the brake's signal slope, the two curves (red and blue) match each other until the motor torque is saturated by the power limiter when the power reaches its maximum value (in this case 50 kW). Obviously the sum between the curves remains always equal to the total braking demand.

3.7 **Test n. 5: safety functions**

3.7.1 Requirements

In this section we will focus on the implemented safety mechanisms. We have discussed about that and we said there are basically two requirements which we consider. They are:

- no delta-torque between right and left drive wheels are allowed;
- no different torque's signs between front and rear axles are allowed;

The implemented model of the safety function described in the *paragraph 2.3.6* are shown in the following picture (only shown for front axle).

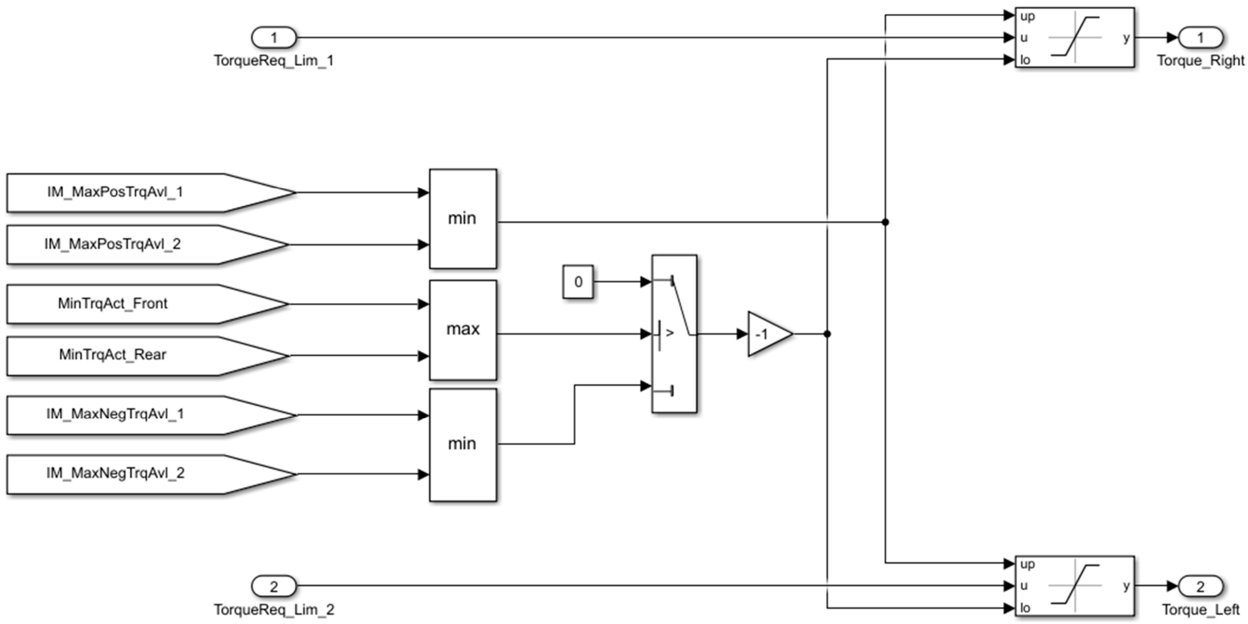
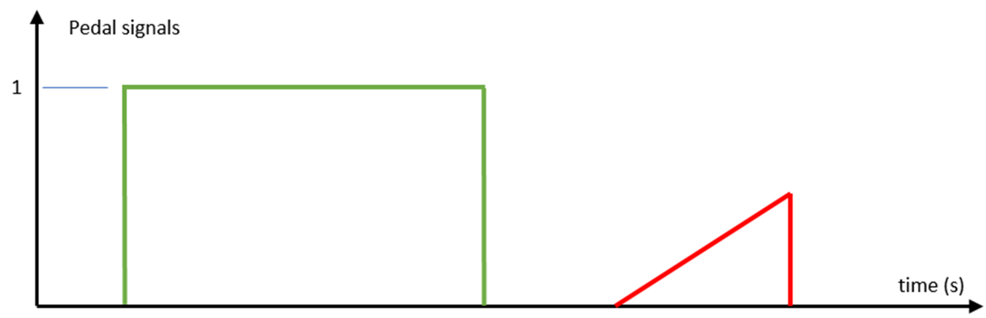


Figure 56 – Safety functions implementation

We note that the value of upper limit corresponds to the minimum availability between the motors of the axle; while the lower limit is handled according to the considerations made in the *paragraph 2.3.6*
The pattern of accelerator and brake used in this test is shown in figure below:



3.7.2 Simulation parameters

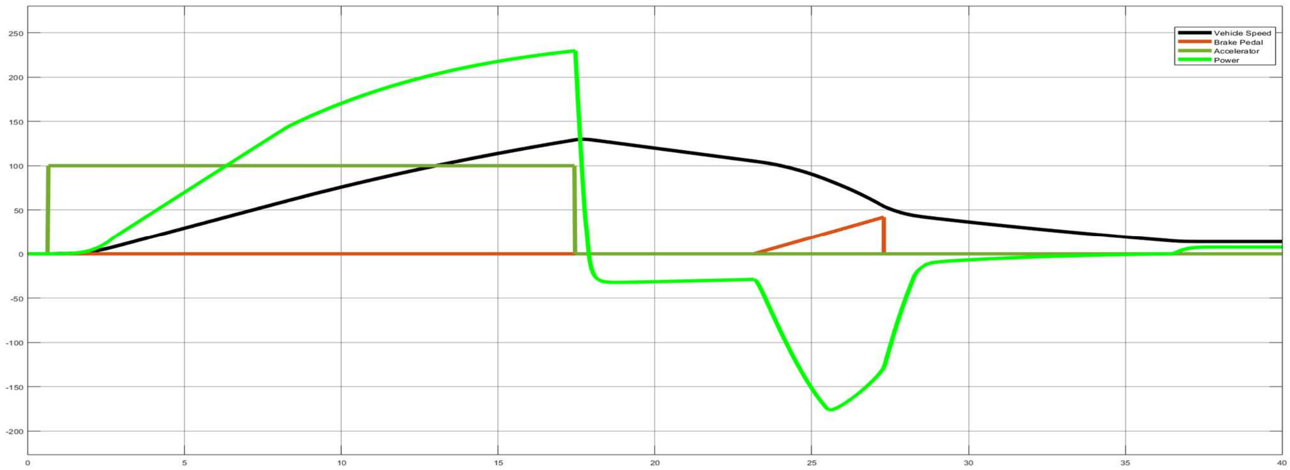
For this simulation we will change most of settings with respect to the previous simulations; in particular, the motor's sizes of the front and rear axle are different in terms of maximum torque while the maximum rotational speed are kept unchanged.
The requirement will be verified also by changing the motor's time constants. Other functions are disabled for reasons of simplicity. All the simulation parameters are shown in the following table.

GENERATED INPUTS		*UNIT*
engaged gear	D (drive)	-
k_regen	1	-
maximum mechanical power - positive	inf	kW
maximum mechanical power - negative	inf	kW
slew rate - positive	5	max Nm/timestep
slew rate - negative	**20**	max Nm/timestep
gear ratio - front	10	-
gear ratio - rear	10	-
coasting mode	**2 (high)**	-
enable creeping	**1**	-
TORQUE PATH PARAMETERS		
time step	12	ms
number of motors - front axle	2	-
max torque - front motors	**100**	Nm
base speed - front motors	maxspeed*30%	rpm
max speed - front motors	10000	rpm
number of motors - rear axle	2	-
max torque rear motors	**50**	Nm
base speed - rear motors	maxspeed*30%	rpm
max speed - front motors	10000	rpm
max braking friction torque	15000	Nm
max creep torque	50	Nm
max coast torque	disabled	Nm
min vehicle speed (negative torque)	8	km/h
SIMULATION PARAMETERS		
simulation time	40	s
time step simulation	1	ms
rolling resistance	0.02	m
air drag	0.3	N s2/m
wheel radius	0.4	m
vehicle mass	2000	kg
time constant motor 1	**100**	ms
time constant motor 2	**50**	ms
time constant motor 3	**150**	ms
time constant motor 4	**200**	ms

Table 8 - Simulation parameters for test n.5

We have imposed four different time constants to test the safety functions as we can see from the table above, in order to consider an extremely situation.

3.7.3 Simulation results



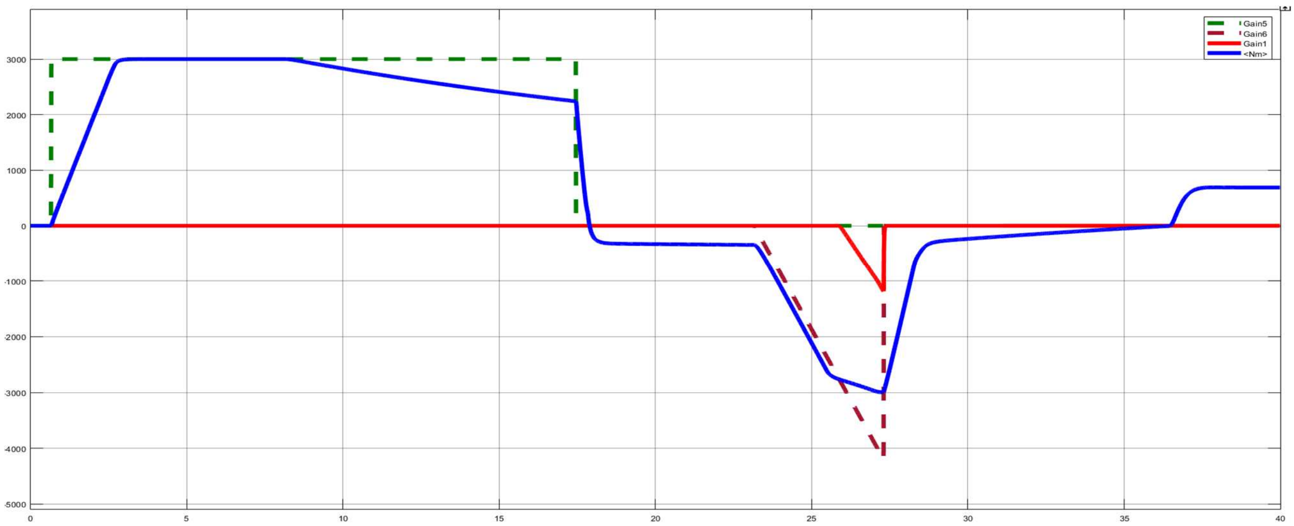
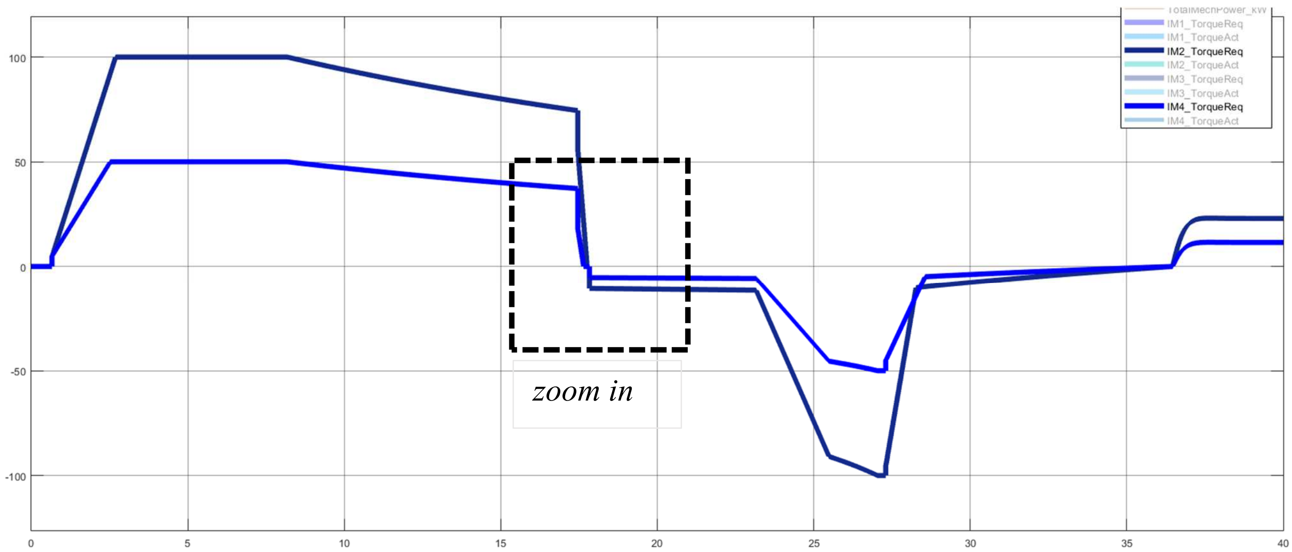


Figure 57 - Simulation results: test n.5



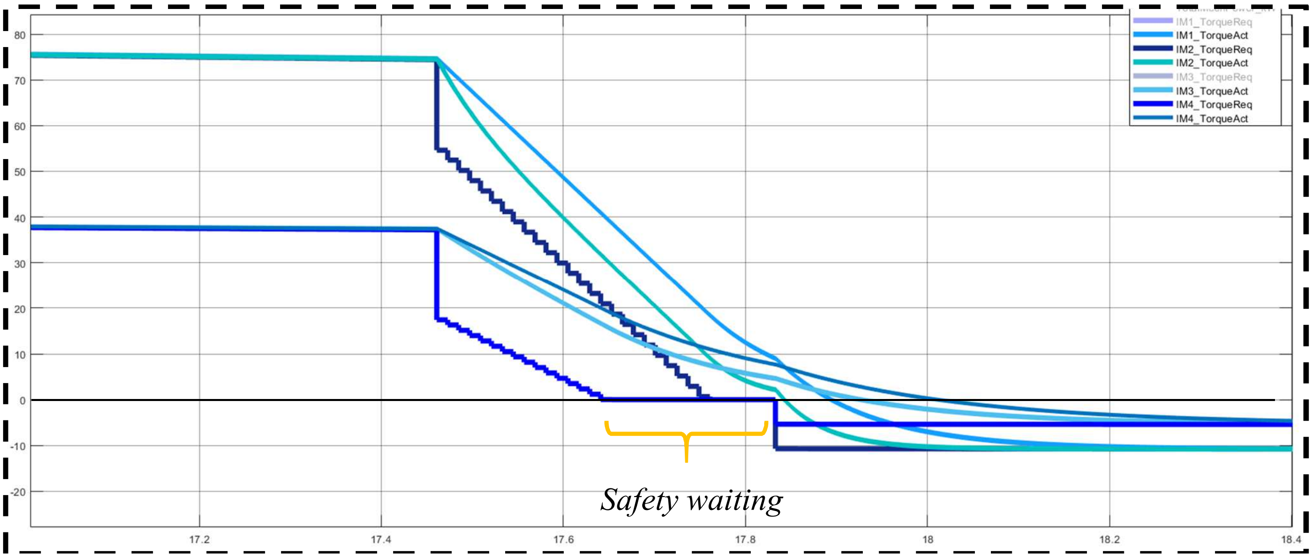


Figure 58 – Test n.5 A focus on the actual motor torques

To verify whether the safety requirements are met we need to display the torque of each motor. In the *Figure 58* the requested torque (front and rear) are shown for reasons of clarity, while in the next picture, we have zoomed in the area where the actual torques and their corresponding commands approach to the zero in order to see better what happens when the target value become negative as a result of releasing the accelerator. Firstly, we can say that the first requirements is fulfilled because the torques of each axle are very closed to each other unless than a certain quantity corresponding in the worst case to the value of slew rate (negative). This behaviour can be acceptable because we are considering an extremely situation with different motor's time constants. We can see indeed the different motor's responses to the same command which are always the same for both motors of the axle. The second requirement is also satisfied since the coasting torques is inhibited until the torques of both the axles are less than a threshold which is set to 5 Nm as we can see from the last graph.
Aside from the coasting function and the related safety requirements we can note also the activation of the creeping as well as the different targets of torques as result of the difference between the maximum torques associated with each axle.

3.8 **Test n.6: Axle's shut down**

3.8.1 Requirements

The input variables called '*disable front axle*' and '*disable rear axle*' refer to the possibility of changing the values of the coefficients of distribution (*m_front, m_rear*) by reducing the torque's availability of the axle we want to shut down.

The solution adopted consists of simply a product between the maximum availability of the axle and the percentage of reduction. This operation is carried out before to send the availability to the torque distribution's block which calculates the corresponding distribution factors.

The Simulink model implemented to this requirement is placed in the block which manages the all the inputs of the torque path.

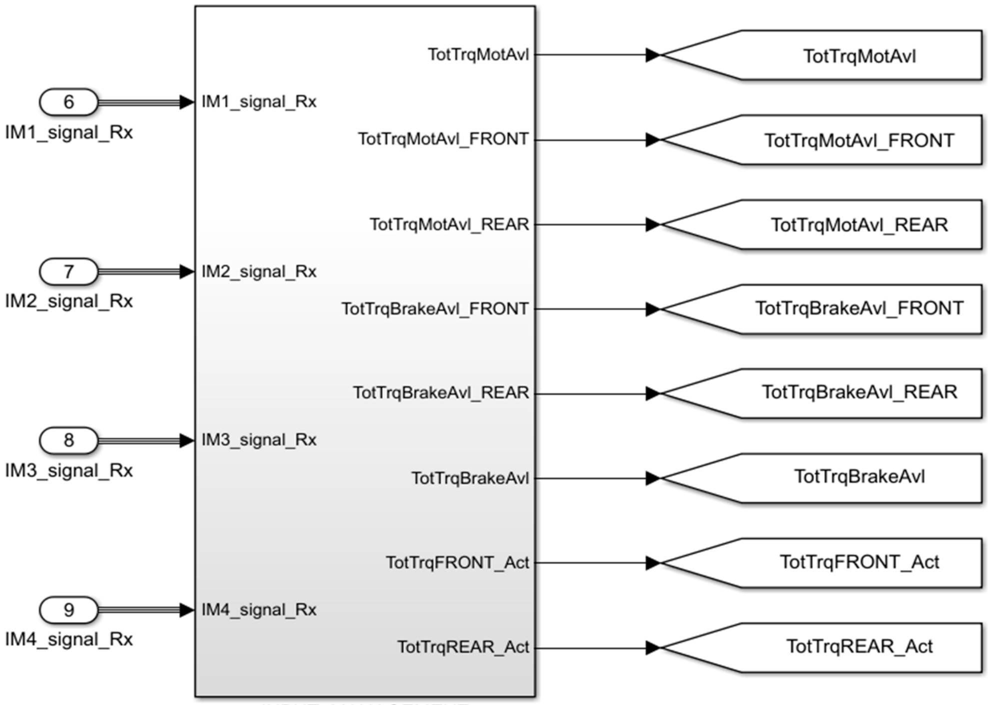


Figure 59 – Input management's block. On the right the signals coming from the motors (actual and available torques and actual speeds).

According to the selected of powertrain's configuration (number of motors), in this section we calculate all the needed values such as the total availability of torque as well as the torque available of one axle (front and rear).

To simulate the shutdown of an axle we need to build a signal variable during time simulation. We resort to a solution as shown in the following illustration:

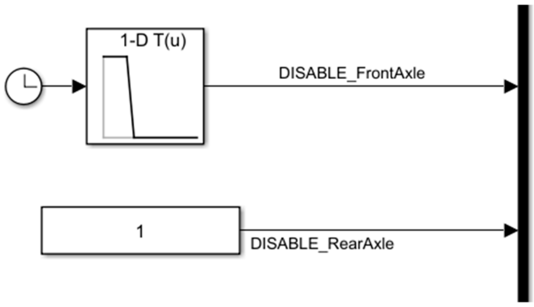
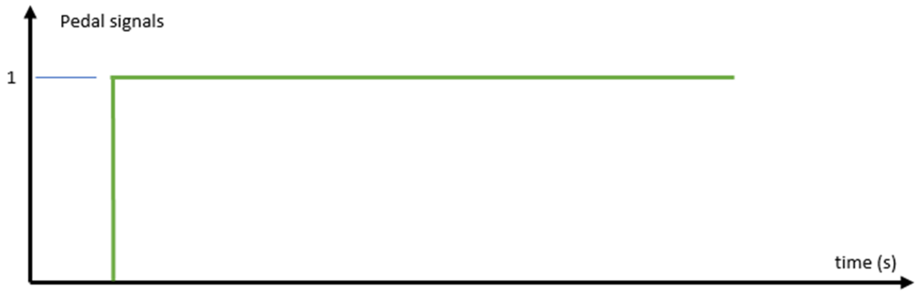


Figure 60 – Simulation of axle's power derating (front). No action for rear axle is carried out.

By a simple look-up-table we are able to create a signal depending on the time simulation using a clock. So, the curve implemented is a function of time which outputs for example the value of 'DisableFrontAxle'. For this test we use only a step of accelerator without brake pressure as shown in the figure below:



3.8.2 Simulation parameters

We choose a configuration with two motor for the front and 1 motor for the rear.

GENERATED INPUTS		*unit*
engaged gear	D (drive)	-
k_regen	-	-
maximum mechanical power - positive	**100**	**kW**
maximum mechanical power - negative	-	kW
slew rate - positive	10	max Nm/timestep
slew rate - negative	5	max Nm/timestep
gear ratio - front	10	-
gear ratio - rear	10	-
coasting mode	-	-
enable creeping	-	-
reduce power rear axle	**1 (no reduction)**	-
reduce power front axle	**function(time)**	-
TORQUE PATH PARAMETERS		
time step	12	ms
number of motors - front axle	**1**	-

max torque - front motors	**200**	Nm
base speed - front motors	maxspeed* 30%	rpm
max speed - front motors	15000	rpm
number of motors - rear axle	**2**	-
max torque rear motors	**100**	Nm
base speed - rear motors	maxspeed* 30%	rpm
max speed - front motors	15000	rpm
SIMULATION PARAMETERS		
simulation time	30	s
time step simulation	1	ms
rolling resistance	0.02	-
air drag	0.3	N s2/m
wheel radius	0.35	m
vehicle mass	2500	kg
time constant motor 1	0.1	s
time constant motor 2	0.1	s
time constant motor 3	0.1	s
time constant motor 4	0.1	s

Table 9 - Simulation parameters for test n.6

3.8.3 Simulation results

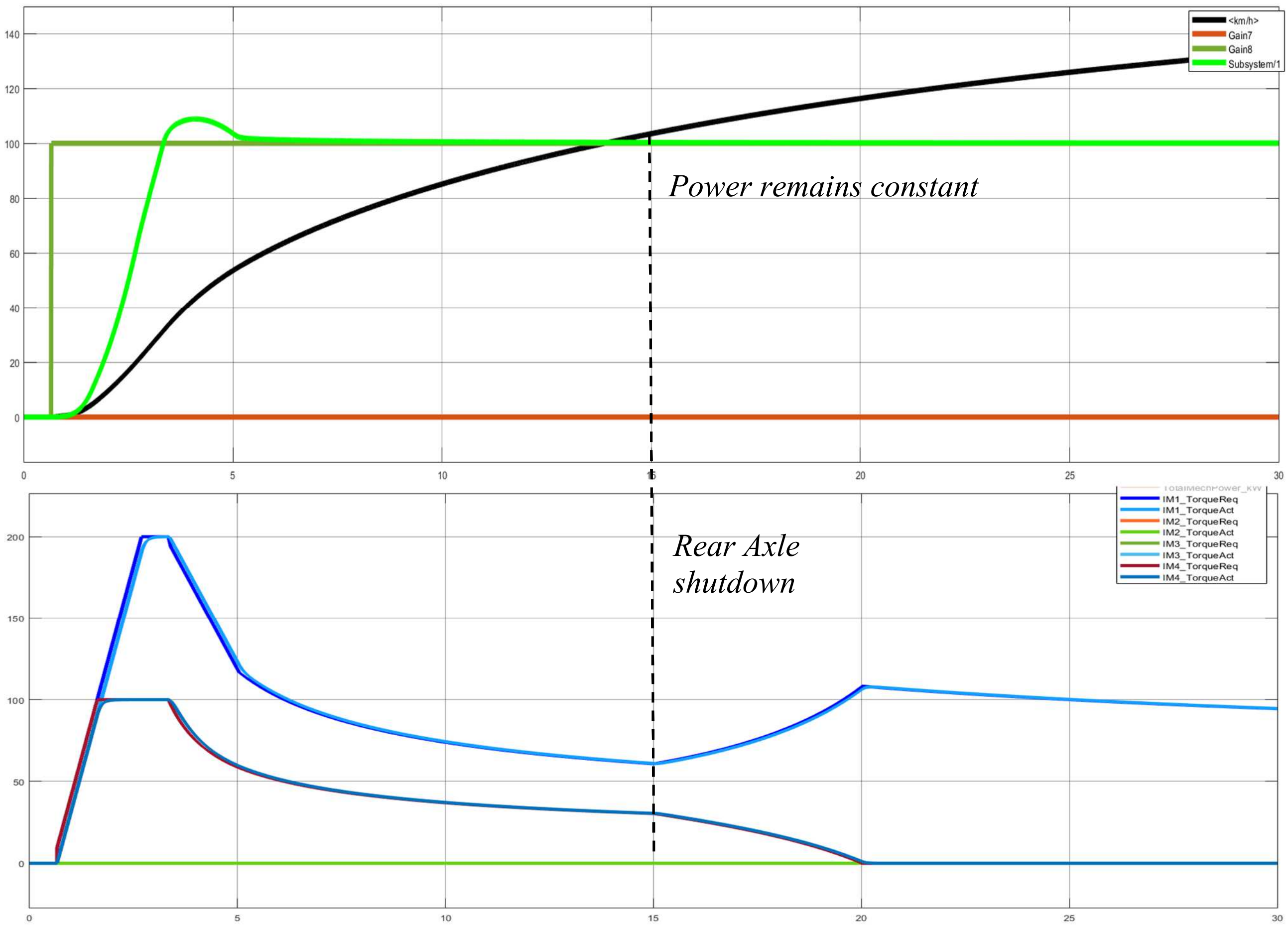


Figure 61 - Simulation results: test n.6

The shutdown of the rear axle occurs at time of 15 seconds. We note that the lack of torque is compensated as possible by the other axle. In this case the maximum power of 100 kW remains constant because of the limitation which we have imposed. This is a consequence of the changing of the power distribution coefficients due to the reset of the availability of the rear axle's power. Therefore, in this case *m_front* becomes equal to one while *m_rear* goes down to zero as we expected.

3.9 **Test n.7: two speed transmission**

3.9.1 Simulation parameters

This test serves to understand what happens when we change the gear ratio during driving. The aim of a two-speed transmission (or more than two) is to improve the performance in terms of maximum torque as we can see from the *Figure 62.* Starting from a certain value of gear ratio we can reduce this value while the speed increases until the maximum torque of the variable characteristic is equal to the value of load's torque. This is because the deliverable power is the same less than the reduction's losses. For this test we set an initial gear's ratio of 10:1. Once the working point is reached we will change the characteristic of the motor setting a gear ratio of 7:1 in order to have a further increase of torque to accelerate the vehicle.

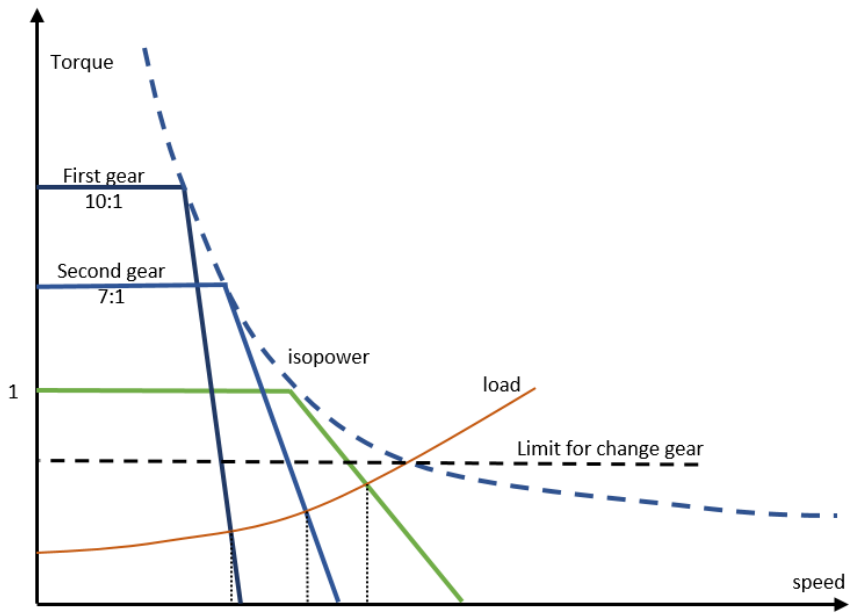


Figure 62 – Exemplary changing motor characteristic according to the gear ratio.

We carry out a test to verify whether the model corresponds to our expectation. To simulate the change gear, we will use a 'signal builder' block by which we create a variable time for the gear ratio. It is shown in the following picture:

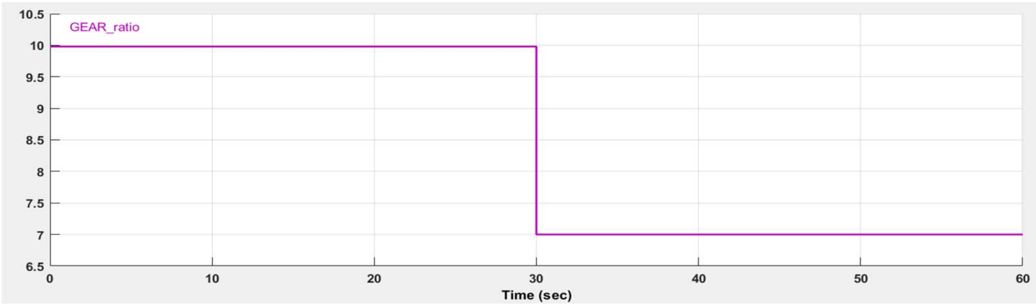


Figure 63 – The signal builder and its graphic interface: making the gear ratio variable in time.

For this test we choose a configuration with two motors for the front axle. The simulation's parameters are summarized in the following table.

GENERATED INPUTS		*unit*
engaged gear	D (drive)	-
k_regen	-	-
maximum mechanical power - positive	inf	kW
maximum mechanical power - negative	inf	kW
slew rate - positive	20	max Nm/timestep
slew rate - negative	5	max Nm/timestep
gear ratio - front	**function(time)**	-
gear ratio - rear	-	-
coasting mode	-	-
enable creeping	-	-
reduce power front axle	1 (no reduction)	-
reduce power rear axle	0	-
TORQUE PATH PARAMETERS		
time step	12	ms
number of motors - front axle	**2**	-
max torque - front motors	**100**	Nm
base speed - front motors	maxspeed* 30%	rpm
max speed - front motors	**8000**	rpm
number of motors - rear axle	0	-
max torque rear motors	-	Nm
base speed - rear motors	-	rpm
max speed - front motors	-	rpm

SIMULATION PARAMETERS		
simulation time	60	s
time step simulation	1	ms
rolling resistance	0.02	-
air drag	0.3	N s2/m
wheel radius	0.35	m
vehicle mass	2500	kg
time constant motor 1	**0.1**	**s**
time constant motor 2	**0.1**	**s**
time constant motor 3	-	s
time constant motor 4	-	s

Table 10 - Simulation parameters for test n.7

3.9.2 Simulation results

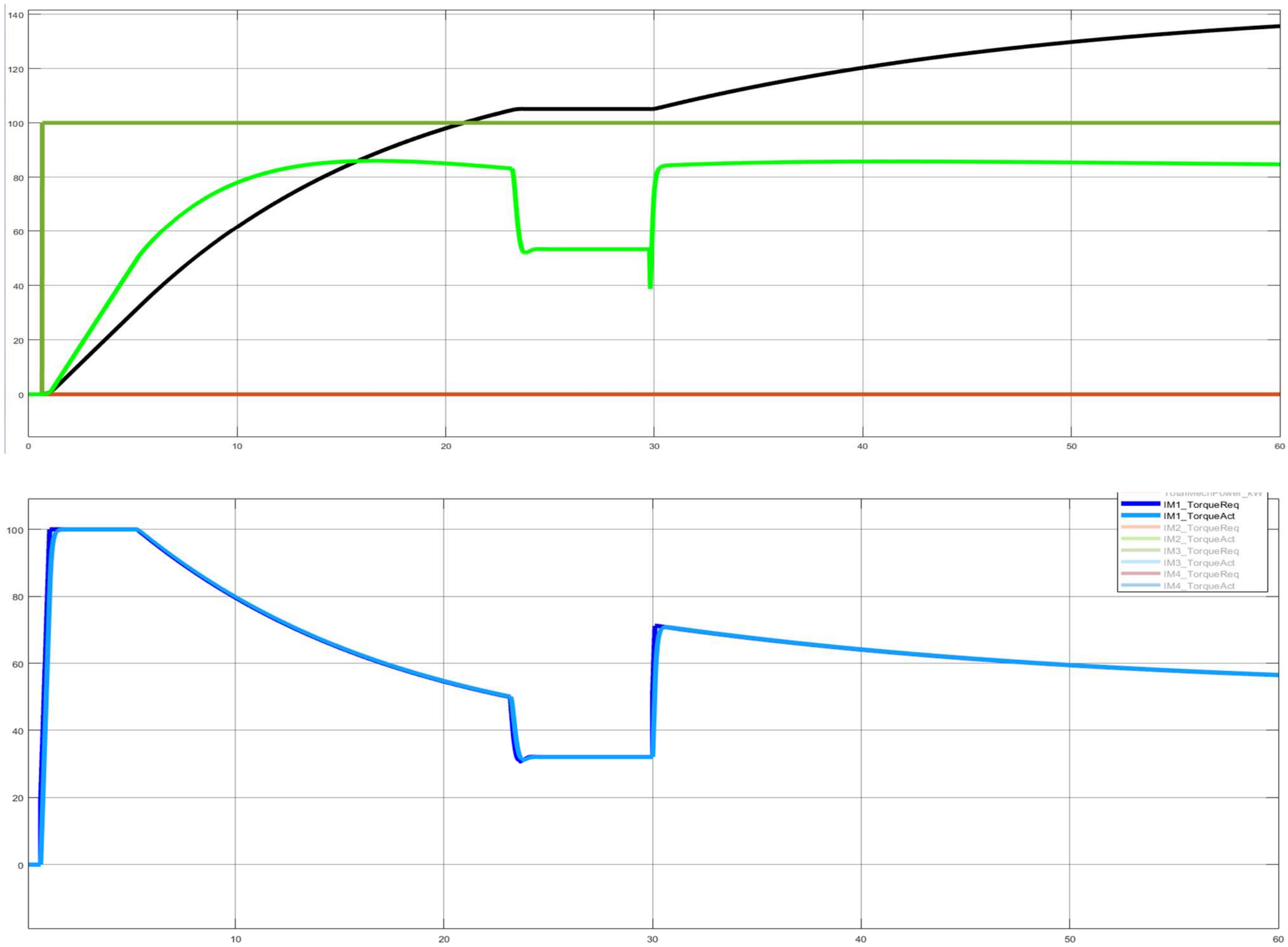


Figure 64 - Simulation results: test n.7

The advantage of changing gear ratio is clear: starting from 0 km/h, during the acceleration phase we have more torque rather than maximum torque of motor. Once we have reached the working point corresponding to the first gear, we change gear ratio in order to shift the characteristic. This solution can be convenient if we have a motor of small size and low speed. By a simple twin-speed transmission we are able to exploit all the operating area of the motor especially the working point with more efficiency. In this way we can have more than one change of gear ratio.

3.10 Future developments and conclusion

This work represents a good starting point to investigate the plausibility of the control algorithm with different powertrain configurations. An improvement of the vehicle model is required if we want to make the simulation closer to the reality.

Aside from these considerations the next step would be the integration of the control algorithm in an ECU in order to make 'Software in the Loop' tests.

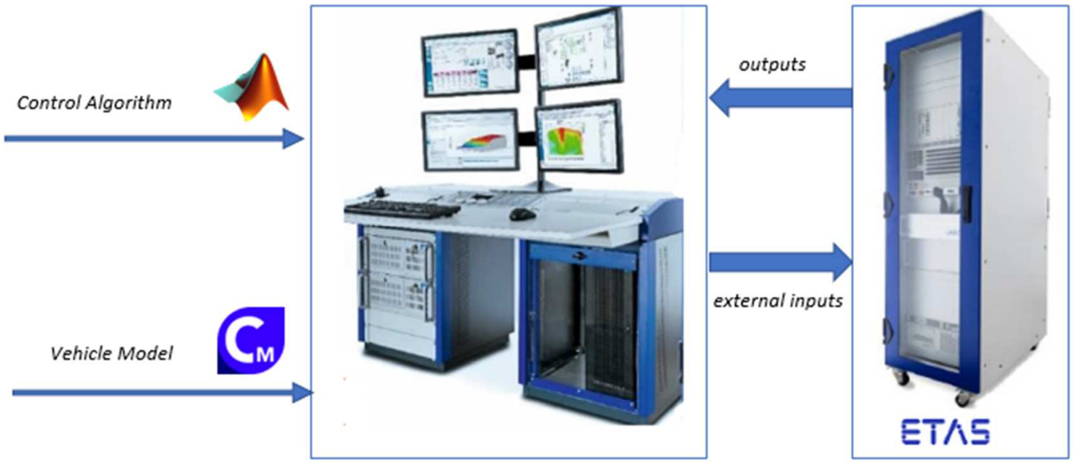
The essence of Software in the Loop consists of verifying whether the algorithm runs without problems. Model-based design is a quite comfortable way to program but it can bring some issues during the coding and implementation phase for example because of using some prohibited blocks in Simulink environment which are not recognized by the coder. In this case we have to go back to Simulink model and change the wrong things. The cycle between SiL and Mil continue until all problems are solved.

Therefore, the algorithm has to be converted automatically in C++ language by a specific solver and then the resulting code has to be run on a '*Real Time PC'* (RTPC).

During this work various tests have been made on a particular HiL workstation which actually serves to test the control algorithm related to combustion engines. Once the implementation of the control algorithm is made on RTPC, we need a vehicle model to test it. In our case we have chosen to implement not only the control algorithm but also the Simulink vehicle model in a single block which receives the input from the control console and outputs the vehicle speed. After we have verified there are no problems, instead of using the vehicle Simulink model, a more sophisticated model provided by AVL software called "*Cruise-M"* which can simulate a more realistic behaviour of the tyre and transmission but always still being in the longitudinal drive hypothesis.

To carry out *Software in the Loop* validation, the workstation uses a specific program for testing controllers in the automotive sector called 'Labcar- Operator'. It was developed taking into consideration the fundamental requirements of a system for developing and validating controller functionality. Its main function consists of the integration platforms for function modules of various kinds: MATLAB/Simulink models, I/O hardware modules, C code modules, CAN (LIN) and Flex Ray modules, modules for open-loop access and signal handling.

The *'Modelling Connector'* for Simulink makes possible to use Simulink models directly with a minimum of extensions to the model. The Simulink model can be used both for *Model in the Loop* and *Hardware in the Loop* experiments. The Real-Time Workshop makes it possible to integrate code defined by the user. This can be an s-function or other code called by an s-function.



The Connection Manager enables closed-loop operation by connecting the inputs and outputs of the existing modules accordingly. Hence, we can send to the vehicle model deriving from *Cruise-M*, the control algorithm's outputs (torques set-points) coming from MATLAB/Simulink. When the model run on RTPC we can control accelerator and brake by the console station and then we can visualize the *real time* response of the algorithm. We can also change every parameter and setting without compiling the model every time, during the execution of the algorithm. This is a very comfortable way to verify if the program runs correctly. These experiments lay the foundation for future developments about *Hardware in the Loop.*

References

[1]. IEA, 2016. World Energy Outlook 2016. International Energy Agency.

[2]. Lutsey, N., 2015. Transition to a global zero-emission vehicle fleet: A collaborative agenda for governments. The International Council on Clean Transportation (ICCT),2015.

[3]. Axsen, J.; Orlebar, C.; Skippon, S.;2013. Social influence and consumer preference formation for pro-environmental technology: The case of a U.K. workplace electric-vehicle study. Ecological Economics,Volume 95, November 2013.

[4]. OECD/IEA 2018. Global EV Outlook 2018: towards cross-modal electrification. International Energy Agency.

[5] Urbschat, M.V.; Bernhart, W., et al.; 2009. Powertrain 2020 – The Future Drives Electric. Roland Berger, Strategy Consultant. 2009.

[6] Fischer, H.M.; et al.; 2013. Voltage Classes for Electric Mobility. ZVEI - German Electrical and Electronic. 2013

[7] Prehofer, C.; Zeller, M.; 2012. A Multi-Layered Control Approach for Self-Adaptation in Automotive Embedded Systems. Fraunhofer Institute for Communication Systems ESK, Munich, Germany; Phillip A. Laplante, August 2012.

[8] Bagubali, A.; Smitha, K.; Sarath, H.;2017. Modern Automotive Embedded Systems with special mention to radars. 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017, India

[9] Yildirim, M.; Polat, M.; Kürüm, H.; 2014. A survey on comparison of electric motor types and drives used for electric vehicles. 16th International Power Electronics and Motion Control Conference and Exposition. Antalya, Turkey 21-24 Sept 2014.

[10] , [28] Felden, M.; Bütterling, P.; Jeck, P.; Eckstein, L.; Hameyer, K.;2010. Electric Vehicle Drive Trains: From the Specification Sheet to the Drive-Train Concept. 14th International Power Electronics and Motion Control Conference, EPE-PEMC 2010

[11] Walker, P.; Roser, H.; Zhang, N., and Fang, Y.;2015 "Comparison of Powertrain System Configurations for Electric Passenger Vehicles," SAE Technical Paper 2015-01-0052.

[12] ISO 26262-3:2011. Road vehicles, Functional safety, Part 3: Concept phase. International Standardization Organization,2011.

[13] National Instruments. White Paper on ISO 26262 functional safety standard. Available on http://www.ni.com.

[14] C. Montero et al.; 2015. "Modeling and torque control for a 4-wheel-drive electric vehicle," IEEE International Conference on Industrial Technology (ICIT), Seville, 2015, pp. 2650-2655.

[15] Aggarwal, A.; 2013. Electronic differential in electric vehicle. International Journal of Scientific and Engineering Research. 4; 2013.

[16] Yin, D.; Shan, D.; Hu, J.S.; 2016. A Study on the Control Performance of Electronic Differential System for Four-Wheel Drive Electric Vehicles. Shoou-Jinn Chang, January, 2017.

[17] Tawadros, P.; Zhang, N.; 2014. Alternative Fuels and Advanced Vehicle Technologies for Improved Environmental Performance; 17, Integration and performance of regenerative braking and energy recovery technologies in vehicles; Woodhead Publishing, Richard Folkson,2014; pp 541-560.
[18] Chau, K.T.; 2014. Alternative Fuels and Advanced Vehicle Technologies for Improved Environmental Performance; 21, Pure electric vehicles; Woodhead Publishing, Richard Folkson,2014; pp: 672.
[19] Ruan, J., Walker, P., Zhang, N., and Xu, G., "The Safety and Dynamic Performance of Blended Brake System on a Two-Speed DCT Based Battery Electric Vehicle," SAE Int. J. Passeng. Cars - Mech. Syst. 9(1);2016.
[20] Paret, D.;2012. Flexray and its applications; real time multiplexed network; 2nd edition. John Wiley & Sons Ltd, UK, 2012.
[21] Laplante (2007). What Every Engineer Should Know About Software Engineering. CRC
[22] Miloser, J., "Approaches to Determining Beneficial Use of Simulink and UML in Automotive Embedded Software Systems," SAE Technical Paper 2017-01-0008, 2017.
[23] Bringmann, E.; Krämer, A.;2008. Model-based Testing of Automotive Systems. Second International Conference on Software Testing, Verification, and Validation;2008.
[24]. Oshana, R.; Kraeling, M.; 2013. Software Engineering for Embedded Systems; Elsevier-Newnes, April,2013. (pp 64,478-481)
[25] Skruch, P., Dlugosz, R., Kogut, K., Markiewicz, P. et al., "The Simulation Strategy and Its Realization in the Development Process of Active Safety and Advanced Driver Assistance Systems," SAE Technical Paper 2015-01-1401.
[26] Jones, C.;2012. Software defect origins and removal methods. Namcook Analytics, Tech. Rep.,2012.
[27] Bergström, D.; Göransson, R.;2016. Model- and Hardware-in-the-Loop Testing in a Model-Based Design Workflow. MSc Thesis, Lund University; ISSN 0280-5316,2016.
[29] Enderle, E; Hammerer, H; Testing Power Electronics without an E-Motor; (available pdf at www.set-powersys.de) ATZelektronik; 2015
[30] Larminie, J.; Lowry, J.;2012. Electric Vehicle Technology Explained-Second Edition. John Wiley & Sons Ltd, UK, 2012.