

POLITECNICO DI TORINO

Master's Degree in Mechatronics Engineering



**Politecnico
di Torino**

Master's Degree Thesis

Development of logic for pollutant emission reduction of a Hybrid Electric Vehicle

Supervisors

Prof. Angelo BONFITTO

Ing. Alessandro DAMINO

Candidate

Salvatore SOLARINO

2022

Abstract

Nowadays, environmental preservation and pollutant reduction are critical points on which the research is focused to reduce the carbon footprint of humans as much as possible. In particular, the mobility industry, responsible for almost 12% of global CO₂ emissions, is investing more and more resources in research and development of solutions to face this problem.

This thesis is collocated to improve vehicles' efficiency and reduce their pollutant emissions, being part of a project AutoECO/Pitef that aims at improving the ecology of a light-duty commercial vehicle by hybridizing it and developing low and high-level, ECO-oriented control logic.

This study developed and optimized two algorithms to pursue the project's objective. An improvement of the standard ECMS logic for power splitting management is introduced in the first algorithm. The newly proposed solution gains one degree of freedom more on the decision of the best power unit's working point through the control of the gear engaged by the drive line. The algorithm will choose the best power split and gear arrangement between all the possible conditions that make the drive-line satisfy the request of the driver by means of a suitable and complete cost function that weights the equivalent fuel consumption, the actuation of the gear shift, the transient of the combustion engine and the fulfillment of the driver's request. A reduction of the CO₂ emissions of 0.73% with respect to the standard ECMS formulation on a WLTP cycle has been achieved during the testing phase. The second algorithm improves the standard Start&Stop logic performances extending the stop time of the engine and avoiding disadvantageous start-stop events. The start of the engine is delayed until the ECMS decides that its contribution is needed through the integration of this decision into the algorithm in the form of an additional weight activated when the engine is off. The engine stop is anticipated thanks to the information about the environment given by the ADAS signals that can detect a stop situation in the near future making the algorithm decide to turn off the engine since it will not be used during braking. An increase in the stop time of 168% has been measured in simulations which causes a reduction of the CO₂ emissions of 2.13% with respect to the standard Start&Stop logic on a WLTP cycle.

Acknowledgements

Ai miei genitori, a cui devo tutto, per aver creduto in me e avermi sostenuto;

Ad Egle e Mattia, per il bene che mi volete e per esserci sempre;

A Francesca, onnipresente, per la sicurezza e la luce con cui ha illuminato ogni giorno;

Ai nonni, per essere fonte di saggezza e ispirazione;

Ai miei coinquilini, protagonisti di crescita e dibattiti, per aver condiviso con me questo percorso;

Ai miei amici e colleghi, compagni di viaggio, per le risate, le uscite, le idee e le sfide;

Ad Alessandro e alla Podium, per avermi accolto e dato la possibilità di crescere professionalmente.

- Salvo

Table of Contents

List of Tables	VI
List of Figures	VII
Acronyms	X
1 Background	1
1.1 Introduction	1
1.2 Project	3
1.3 Thesis outline	4
2 Vehicle modelization and parameters	5
2.1 Introduction	5
2.2 Hybrid Electric Vehicles	5
2.2.1 HEV topologies	5
2.2.2 HEV degrees	10
2.3 Vehicle configuration and parameters	10
2.4 Vehicle model	15
2.4.1 Plant	16
2.4.2 Controller	19
2.4.3 ADAS inputs	20
2.4.4 Simulation Interface	23
2.5 Speed profile for simulation	24
2.5.1 WLTP homologation cycle	25
2.5.2 Reference profile actuation	26
3 ECMS with Gear Control	30
3.1 State of the art	31
3.1.1 Introduction	31
3.1.2 Input description	32
3.1.3 Torque split computation	34

3.1.4	Cost function description	36
3.2	Solution produced	40
3.2.1	Introduction	40
3.2.2	Algorithm description	40
3.2.3	Algorithm realization	42
3.2.4	Gear for-cycle	44
3.2.5	Feasibility conditions	46
3.2.6	New cost function	50
3.2.7	Battery energy behavior management	54
3.3	Results	58
3.3.1	Emission reduction	59
3.3.2	Performances	61
4	Intelligent Start&Stop	63
4.1	State of the art	64
4.1.1	Previous algorithm description	64
4.1.2	Plant actuation	65
4.2	Solution produced	66
4.2.1	Restart delay	67
4.2.2	Advance shutdown	68
4.3	Results	72
5	Conclusions	77
A	Model images	79
A.1	Plant	79
A.2	Controller	81
A.3	Simulation Interface	82
B	Matlab codes	84
B.1	ECMS-GC	84
B.2	ADAS based Start&Stop	91
B.2.1	Velocity prediction analysis triggered version	91
B.2.2	Stop-triggered version	92
	Bibliography	93

List of Tables

2.1	Vehicle parameters	11
2.2	Sensor fusion characteristics	14
2.3	WLTC class 3b cycle characteristics.	25
3.1	Different ECMS logics fuel consumption and emission comparison. .	59
3.2	ECMS algorithms Performance index comparison.	62
4.1	Start&Stop algorithms performance comparison.	73

List of Figures

2.1	HEVs topologies overview.	6
2.2	Series HEV scheme.	7
2.3	Parallel HEV scheme.	8
2.4	Parallel HEV configurations scheme.	8
2.5	Series-Parallel HEV scheme.	9
2.6	HEV degrees scheme.	10
2.7	BSFC engine map.	12
2.8	EM map.	13
2.9	Vehicle model components out-view.	15
2.10	Engine simulink model.	17
2.11	Gearbox model.	18
2.12	Battery simulink model.	18
2.13	Electric motor simulink model.	19
2.14	ADAS models options.	21
2.15	Ideal ADAS model.	22
2.16	Simulated ADAS model.	23
2.17	WLTC cycle for class 3b vehicles.	26
2.18	External reference profile control loop scheme.	27
2.19	ACC input-output scheme.	28
2.20	External ACC control loop scheme.	28
2.21	ACC performance in following a vehicle that tracks a WLTC cycle.	29
3.1	Input-output representation of the standard ECMS algorithm.	31
3.2	Lookup table for maximum EM torque.	32
3.3	PI driver model	33
3.4	Simulink block scheme for requested torque computation	34
3.5	Logic scheme of the standard ECMS algorithm.	35
3.6	Fuel consumption map of the engine.	37
3.7	Available points of the engine map and maximum torque characteristic.	37
3.8	Relay Simulink block.	40
3.9	Engine's map points that are considered by the standard ECMS.	41

3.10	Engine's map points that are considered by the ECMS-GC.	42
3.11	Logic scheme of the new ECMS-GC algorithm.	44
3.12	Driveline scheme of power transformers.	45
3.13	Area of the engine map where the points could be precisely extrapolated.	48
3.14	Example of point retrieved by interpolation that is out of the known zone of the engine's map.	48
3.15	New look-up table system adopted for engine's map interpolation.	49
3.16	Block scheme of the time-dependent weight for gear change	51
3.17	PI controller for $f(SOC)$ production	55
3.18	Saw-tooth behavior assumed by the SOC in charge maintaining with the relay $f(SOC)$	56
3.19	New polynomial $f(SOC)$	57
3.20	Behavior assumed by the SOC in charge maintaining with the new polynomial $f(SOC)$	58
3.21	ICE engine map working points comparison between standard ECMS and ECMS-GC.	60
3.22	EM engine map working points comparison between standard ECMS and ECMS-GC.	60
3.23	ECMS-GC and standard logic Gear behavior time series comparison.	61
4.1	Standard ICE Start&Stop block scheme.	64
4.2	Standard ICE Start&Stop logic.	64
4.3	Throttle disabling with Stop signal	65
4.4	Resisting torque signal generation.	66
4.5	Simulink block scheme for fuel cut off when the engine is off.	66
4.6	Simulink block scheme for fuel addition when engine starting.	66
4.7	Logic scheme for i-Start&Stop restart delay.	68
4.8	Coast down experiment time series.	70
4.9	Coast down factors interpolation.	70
4.10	Logic scheme of stop prevision with velocity profile analysis.	71
4.11	Adas-retrieved stop signal behavior with respect to the velocity preview and the actual one.	73
4.12	Particular of different Start&Stop algorithms short stop handling.	75
4.13	Engine stop behavior comparison for different Start&stop architectures.	76
A.1	Plant dynamics Simulink model.	79
A.2	Plant power-train Simulink model.	80
A.3	Controller Simulink model overview.	81
A.4	Simulation configuration interface view.	82
A.5	ECMS configuration interface view.	83

Acronyms

ADAS

Advanced Driving Assistance Systems.

HEV

Hybrid Electric Vehicle.

ICE

Internal Combustion Engine.

EM

Electric Motor.

BSFC

Brake Specific Fuel Consumption.

SOC

State Of Charge.

ECMS

Equivalent Consumption Minimization Strategy.

ECMS-GC

Equivalent Consumption Minimization Strategy with Gear Control.

Chapter 1

Background

1.1 Introduction

The continuously growing trend of vehicle circulation on the roads and the simultaneous need for reducing pollutant emissions for the ecology of the planet represents one of the main technological problems of the automotive industry that is committed, also driven by stringent regulations, in the research and development of innovative and effective solutions.

The research focuses on improving the vehicle's system to improve their energy efficiency and reduce pollutant emissions without compromising performance, all looking for a solution economically sustainable by the consumer and adequately supported by infrastructure.

The objective can be pursued through two main approaches:

- Hardware Improvement [1][2]
- Software Improvement [3][4]

As for the first point, In addition to the basic aspects of hardware improvement regarding the choice of better architecture elements, such as a more efficient engine or a better transmission, the tendency to cancel emissions has shifted the focus to electric mobility where, if on the one hand the increasing advancement of technologies in the field of traction push these solutions to have more than significant efficiencies, on the other hand, on the other hand, the lack of a final solution that allows the same freedom of fuel as thermal vehicles in terms of autonomy and charging speed is a limit. These contrasting aspects have led research and the market to focus on the development of hybrid electric vehicles.

This solution represents the best compromise with the technologies currently available to enjoy the logistical advantages of combustion engines (better infrastructure network, better energy storage and charging system) increasing their ecology thanks

to the help of electric traction that decreases polluting emissions and increases the overall efficiency of the powertrain by maintaining, and in some cases increasing, the performance of the vehicle.

However, this architecture has its limitations in requiring mechanical complexity and greater weight for the use of both power sources, as well as a greater maintenance requirement due to the greater number of components. Software improvement aims to develop logics that regulates in a better and more efficient way the use of the hardware components adopted in the studied architecture.

Among the algorithm development approaches, two macro-categories can be distinguished:

- The causal logics [5][6] that exploit the instantly available information of the system as input to produce the control decision in output; they are usually lighter computationally speaking to allow the implementation of these logics in a real-time manner; an example is a rule-based algorithm or fuzzy logic.
- The non-causal logics [7][8] instead adopt the approach of studying a posteriori the trend of the outputs with respect to inputs and control variables to determine the best control sequence that leads to the optimization of a minimum problem to be pursued; they are usually computationally heavy and requiring the analysis of the future trend of the plant response to determine the actual optimal control, they are not applicable real-time but are used as a benchmark to evaluate the performance of causal logics; an example of these types of logics is Dynamic Programming.

This thesis goes into the development of causal logic, in view of a future real-time implementation in a vehicle, for the pursuit of the reduction of emissions and consumption of a hybrid electric vehicle according to the objectives of the "AutoECO/Pi.Te.F." project, described in section 1.2.

In particular, some of the logic developed proposes to exploit the information coming from the ADAS sensors, a technology in constant diffusion and development especially from the driving safety point of view [9], here used for the perception of the surrounding environment and the recognition of scenarios exploited by the algorithms produced to apply a more suitable and efficient control strategy.

1.2 Project

The thesis work contributes to advancing research within the AutoECO project, the research project funded by the Piedmont region through the "Pi.Te.F." invitation to tender, Piattaforma Tecnologica di Filiera [10].

The Call aims to encourage collaborative projects between a company and its supply chain partners, pursuing the objectives of supporting the production chains of the participating regions (Piedmont and Valle d'Aosta), encouraging commercial relations and technological development between companies working in the same field, stimulating programs aimed at technological sharing and product development, processes, and organization and support the most promising supply chains.

In this context, a collaboration has been established between the companies Dayco Europe S.r.l. [11], Tecno System S.p.A. [12], Italtcnica S.r.l. [13], Podium Engineering S.r.l. [14], and the Politecnico di Torino [15] for the development of the AUTO-ECO project, whose purpose is the development of a P2 hybrid vehicle equipped with ADAS (Advanced Driving Assistance Systems) sensors used for the optimization of vehicle performance and emissions.

In particular, the project plans to:

- Hybridise the vehicle's powertrain in such a way that it can be homologated at the European level;
- Creation of high-level logic that exploits the information regarding the surrounding environment perceived through ADAS sensors to reduce vehicle emissions;
- Study the implementation of ADAS sensors to better obtain the information required by high-level logic.

The research work described in this thesis was carried out during an internship in the company Podium Engineering S.r.l. whose research objective concerns the development of the vehicle module, and in particular, addresses the development of the logic for the reduction of emissions working on the vehicle model to study its implementation and validate the results.

1.3 Thesis outline

This thesis is organized into five chapters:

The first one, here concluded, introduced the topic and described the background of the project.

The second one, after giving an overview of the hybrid vehicle's technologies, deals with the description of the vehicle's characteristics and model, highlighting the employed ADAS setup and giving space also to the homologation normative operated for simulation and validation purposes.

The third and fourth chapters describe the solutions originated from the research work, the third contains the solution that doesn't need the environment information support, the ECMS-GC, while the fourth includes the solution that takes advantage of the ADAS sensor's information, the i-Start&Stop. Each of the solutions treated is discussed by supplying an introduction of the problem, the state of the art of the solution already present and the description of the solution produced with the collected results.

The last chapter contains the conclusion and collects the work results, with a digression on the possible future developments.

Chapter 2

Vehicle modelization and parameters

2.1 Introduction

In this chapter the hardware system considered in the project will be described, firstly introducing an overview of the classification of hybrid electric power trains to frame the context in which the treated vehicle is located, then, the technical characteristics of the car and information on the set-up adopted for the ADAS sensors are reported. Finally, the vehicle model used for the simulation and validation of the algorithms is described.

2.2 Hybrid Electric Vehicles

Hybrid electric vehicles, hereinafter referred to as HEVs, have two power sources available, the fuel and the battery that can be reversible (the electric machine can be used as a motor or generator).

The classification of HEVs is made according to the type of hardware set-up that is implemented in the power train (Topologies) or according to the possibilities and uses that the electric motor has in traction (Degree).

2.2.1 HEV topologies

The distinction between the different HEVs topologies is done according to the ICE speed or torque adjustment capability and ICE and/or EM power addition capability.

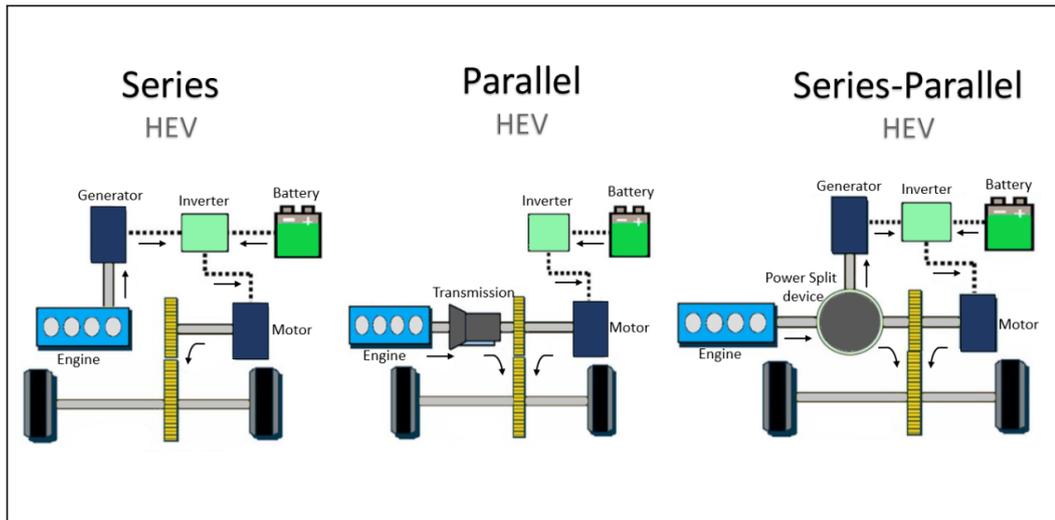


Figure 2.1: HEVs topologies overview.

The three main topologies are :

- Series HEV (ICE torque and speed adjustment capability);
- Parallel HEV (ICE torque adjustment and power addition capability);
- Series-Parallel HEV (ICE torque and speed adjustment and power addition capability).

Series HEV

For series hybrid configurations the mechanical power from the engine is immediately converted into electrical power by a generator. The engine is not directly linked to the wheels, while the electrical energy generated is firstly stored in a battery pack and then used to power an electric motor that is the only source of traction.

The pros of such a configuration are that there isn't a mechanical link between the ICE and the wheels, the combustion package can be located anywhere and there's no need for conventional transmission elements. Moreover, this configuration can let the engine operates in its narrow most efficient working points.

In contrast, the cons of this topology are that every power source of the system must be dimensioned to handle the full power of the vehicle, involving much weight, volume and cost. In addition, the power coming from the fuel has to go through the efficiency of the combustion engine and of all the electric set. For that reason, this system could be less efficient during long-distance highway driving.

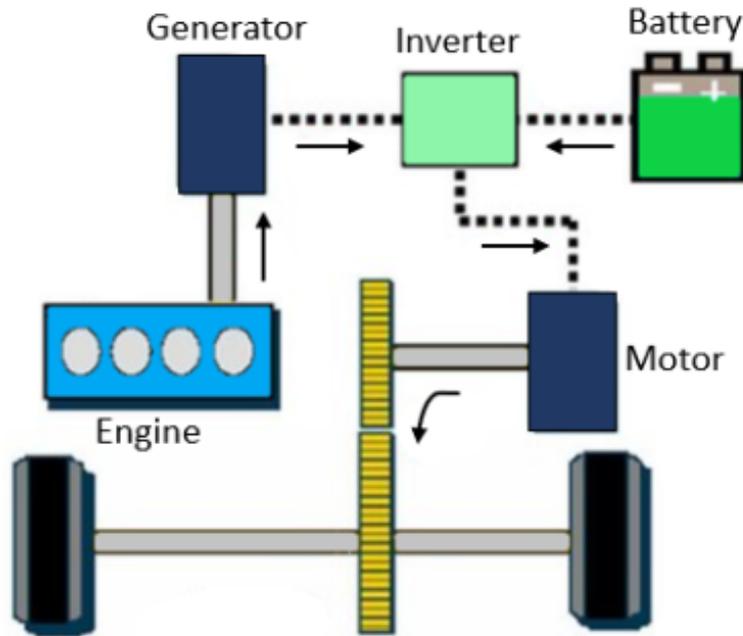


Figure 2.2: Series HEV scheme.

Parallel HEV

In Parallel hybrid configurations, the EM is in parallel with the ICE and they are both linked to wheels.

Depending on where the EM is placed, there could be different architectures:

- In P0 architectures the electric motor is mounted through a belt on the front end accessory drive and it accounts for the power engine's accessories to improve its performance and response time;
- In P1 architectures the EM is connected directly with the crankshaft of the ICE, usually on the fly-wheel and accounts for torque support and start performance improvements.
- In P2 architectures the EM is mounted between the ICE and the transmission on the transmission side and could be disconnected from the engine through the clutch; it accounts for torque support and complete electric traction;
- In P3 architectures the EM is connected through a gear mesh with the transmission;

- in P4 architectures the EM is connected through a gear mesh on the axle and is located in the axle drive or in the wheel's hub.

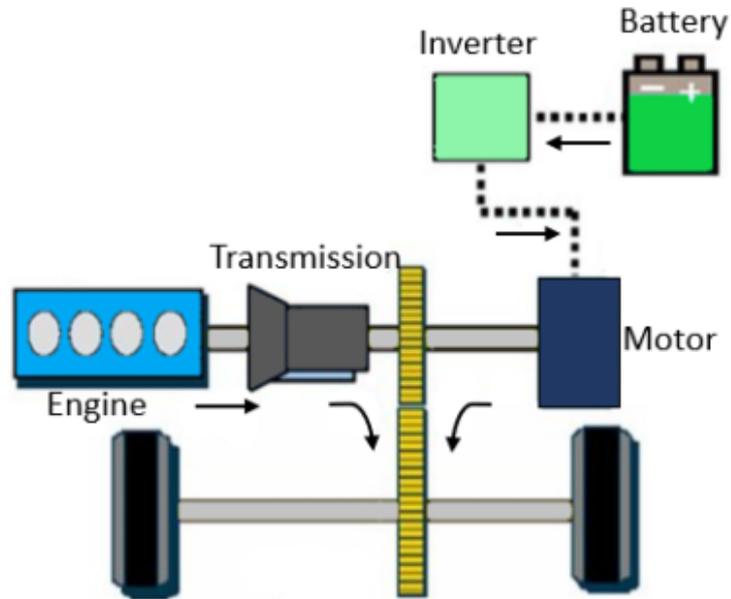


Figure 2.3: Parallel HEV scheme.

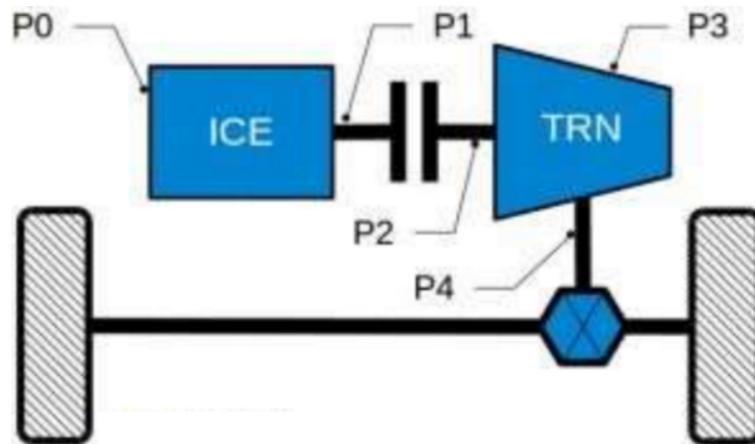


Figure 2.4: Parallel HEV configurations scheme.

From P2 on, thanks to the possibility of disengaging the engine from the motor and the wheels, it is possible to have all-electric traction and regenerative braking. In every configuration it is possible to have torque assist from the motor or battery

charging through the engine's power.

The pros of those configurations are that the efficiency gain remains valid even in long-distance highway drives and that the power sources don't need to be dimensioned to handle the whole power of the vehicle.

On the other hand, they are rather complicated systems, the engine velocity is not controlled, so it can't always operate at the best efficiency points.

Series-Parallel HEV

The series-Parallel HEVs are the most complex system from a mechanical point of view. They involve planetary gear for power summation and torque ratios.

Thanks to this configuration it is possible to gain the maximum flexibility to switch between EM and ICE power that, being decoupled from the power demand of the driver, can be of a smaller, lighter and more efficient design.

Besides, as said, they are very complicated and expensive systems and, involving such a complicated transmission, their efficiency plays an important role.

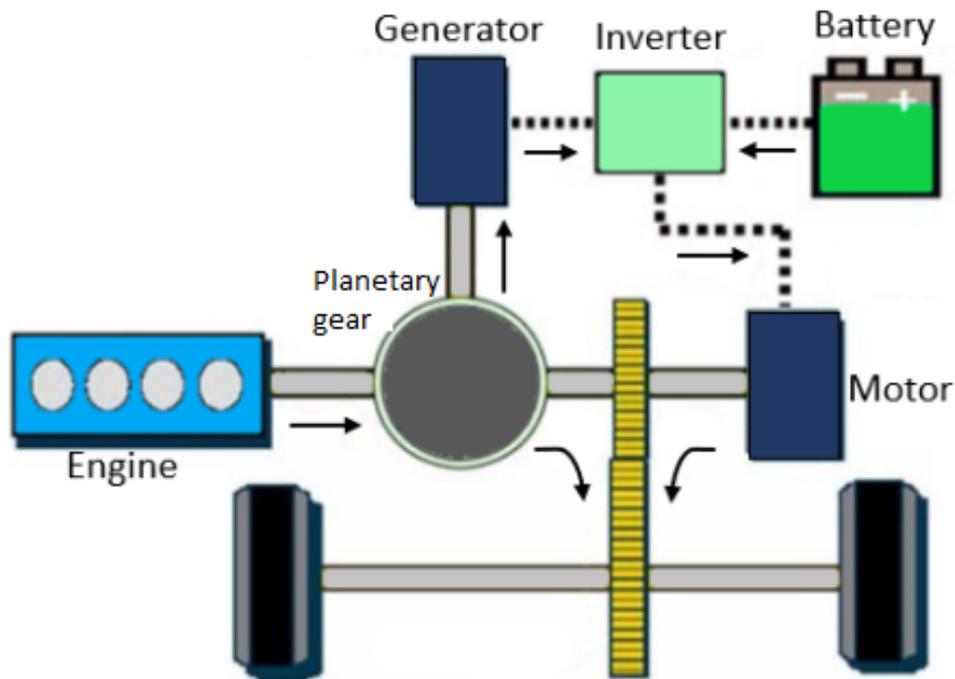


Figure 2.5: Series-Parallel HEV scheme.

2.2.2 HEV degrees

The degree classification is made with respect to the EM power level and battery pack size and the system specification like the ratio between the EM and ICE power, The Start&Stop capability, the regenerative braking capability, the power assistance to the ICE and the full-electric propulsion possibility.

They are distinguished into four categories:

- Micro hybrid involves Start&Stop capability and limited or optional regenerative braking;
- Mild hybrid involves Start&Stop capability, regenerative braking and limited power assistance to the ICE;
- Full hybrid involves Start&Stop capability, regenerative braking, power assistance to the ICE and limited full-electric propulsion;
- Plug-in hybrid involves Start&Stop capability, regenerative braking, power assistance to the ICE and full-electric propulsion, but most importantly, allows the battery to be charged by an external source.

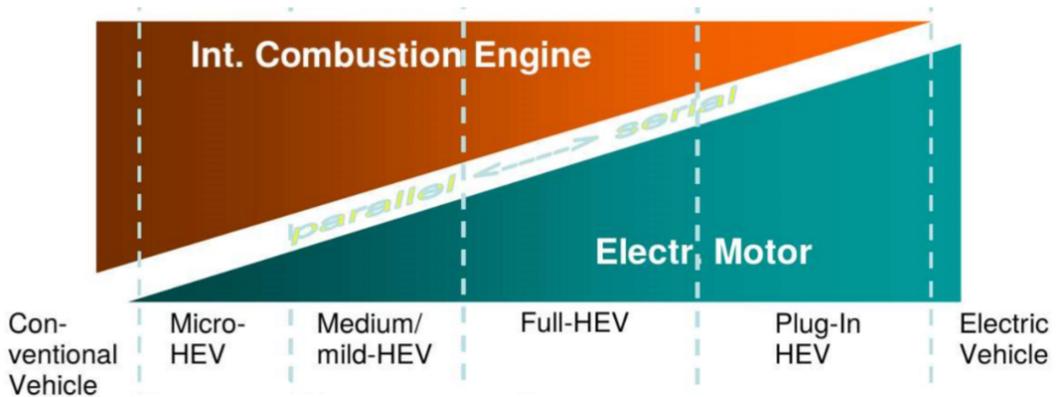


Figure 2.6: HEV degrees scheme.

2.3 Vehicle configuration and parameters

In this section, it will be described the vehicle used as an object of study and simulation, in its characteristics and main parameters.

The vehicle considered is a light-duty commercial vehicle with electric hybrid propulsion of P2 topology and full hybrid grade. As explained in section 2.2 it is a configuration that has the electric motor mounted into the input shaft to the

transmission that can be disconnected from the engine through the disengagement of the clutch. The degree of hybridization allows purely electric traction, albeit limited, the improvement of Start&Stop performance and regenerative braking but does not allow the battery to be recharged from external sources.

The characterization of the vehicle is possible thanks to the parameters provided in the Table 2.1 and to the efficiency maps of the engine (Brake Specific Fuel Consumption) and electric machine (EM efficiency map).

Element	Characteristics
ICE	BSFC and maximum Torque characteristics
ICE max Torque and speed	$T_{ICE,MAX} = 390Nm @1700rpm ; \omega_{ICE,MAX} = 4000rpm$
EM	Efficiency and maximum Torque characteristics
EM max Torque and speed	$T_{EM,MAX} = 330Nm @(0 - 2700rpm) ; \omega_{EM,MAX} = 7000rpm$
Transmission	Automatic 8 gears transmission with torque converter
Gear ratios	[4.55 , 2.96 , 2.07 , 1.69 , 1.27 , 1.00 , 0.85 , 0.65]
Coast Down	$f_0 = 0.0997N f_1 = 0.0043N \frac{h}{km} f_2 = 1.3712N \frac{h^2}{km^2}$
Vehicle Mass	3175 kg
Battery Capacity	40Ah = 14 kWh
Wheels	Inertia=0.8kgm ² Radius=0.35m

Table 2.1: Vehicle parameters

BSFC

The Brake Specific Fuel Consumption map is an experimentally-retrieved engine characteristic that links for each working point, identified by a couple (Torque, Angular velocity), a factor that is calculated as the ratio between the fuel requested by the engine and the mechanical power supplied, as an indicator of efficiency.

$$BSFC_{i,j} = \frac{\dot{m}_f}{\omega(i)T(j)} \quad (2.1)$$

As can be seen from Equation 2.1, this map can be converted into a fuel-consumed map by multiplying all BSFC points by the mechanical output power.

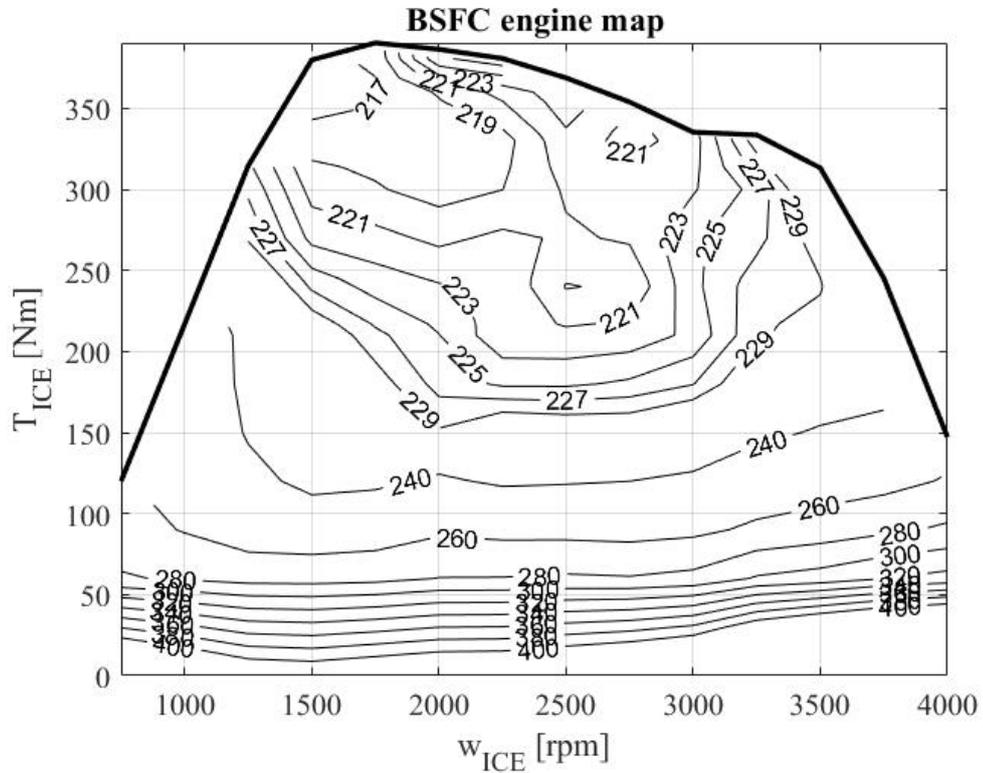


Figure 2.7: BSFC engine map.

EM efficiency map

The electric machine efficiency map is an experimentally-retrieved map that links each couple of working points to the efficiency of the electrical machine, both working as a generator (negative torque) or as a motor(positive torque). This efficiency is supposed to consider the efficiency of the whole electric drive, composed by the battery, the power electronics and the electric links.

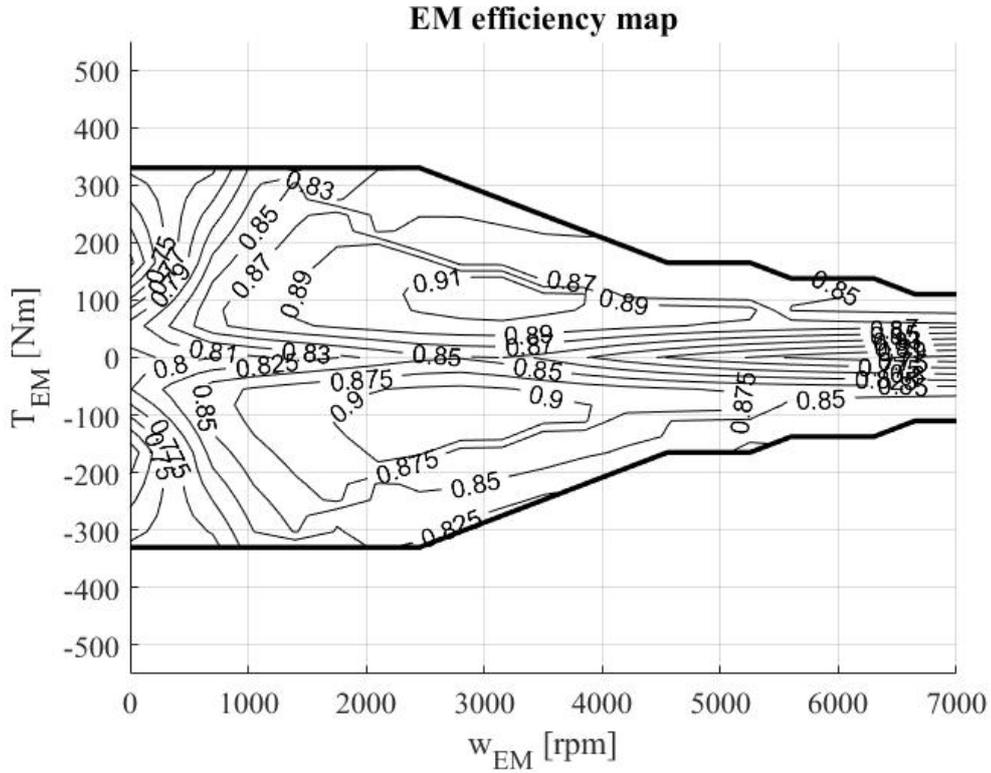


Figure 2.8: EM map.

ADAS set-up

For the project's objective reason, the vehicle under study is equipped with an ADAS hardware set-up composed of a Stereo camera and a mid-range Radar that will collaborate to supply environment information through sensor fusion.

The radar sensor is able to detect objects by calculating the distance, angle and speed of the detected objects using electromagnetic waves in the radio frequency spectrum. The biggest advantage of the radar sensor is that the environmental conditions do not affect the functionality of the device and it works perfectly even in variable lighting conditions.

Automotive radars can be divided into three categories, short-range radar, medium-range radar and long-range radar, based on their detection capabilities. The difference between these three main categories is the "visibility range". The short-range radar provides a range of up to 80 m, the medium-range radar up to 150 m, while the long-range radar up to 200 m.

For this project, the radar used uses the principle of frequency-modulated continuous wave (FMWC), which helps to measure the range and speed of the detected

objects.

The goal of using a stereo camera for the perception of the environment is to realize the detection and identification of road elements of interest, using state-of-the-art object detection algorithms. The stereo camera captures frame by frame of the road environment and the algorithm, usually based on a neural network, reads the images and detects if there are objects of interest in the field of view to be identified and classified. The classification is carried out by drawing the so-called "bounding box" around the recognized object and assigning to it a "label" with the name that represents the class to which the object belongs (any labels are "vehicle", "traffic light", "road sign").

As can be seen in Table 2.2, sensor fusion is fundamental to take advantage of both hardware systems' pros and improve the supplied information's redundancy.

Parameter	Camera	Radar	Sensor Fusion
Distance evaluation	bad	good	good
Visible distance	bad	good	good
Localization	bad	good	good
Object classification	good	limited	good
Angle estimation	good	bad	good
Bad weather conditions performances	limited	good	good
Low light conditions performances	bad	good	good
Cost	good	good	good

Table 2.2: Sensor fusion characteristics

2.4 Vehicle model

This section will describe the Simulink model of the vehicle that has been used throughout the thesis work for the simulation, validation and data collection of all the technologies described in the next chapters.

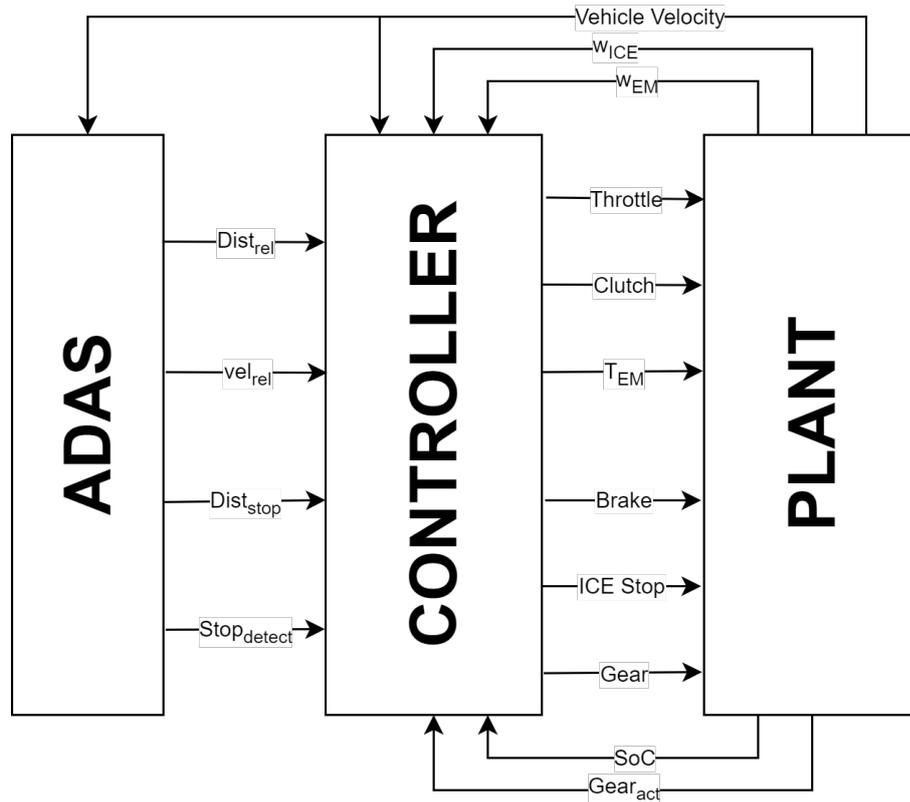


Figure 2.9: Vehicle model components out-view.

The model is structured in:

- Plant, that contains the model of the vehicle itself as a system to be controlled;
- Controller, which contains the logical systems that manage the control of the control variables of the system with respect to feedback measured by the Plant or measurements coming from the ADAS;
- ADAS inputs, where inputs from ADAS systems are simulated.

Finally, the simulation interface system that has been created to better manage the model is described.

2.4.1 Plant

The model of the Plant is organized in a more external section where the elements inherent in the dynamics of the vehicle are modeled, and a more internal section where the models of the power-train components are contained, a graphic illustration of them is given in section A.1.

Being the subject of modeling for multiple physical systems, we have exploited the Matlab Simscape add-on software, which allows you to create models of physical components based on physical links that are directly integrated with block diagrams and interfaces on Simulink.

As regards the outermost section, it shall include:

- The longitudinal dynamics model of the vehicle that, given the applied forces, calculates the longitudinal speed;
- The model of the wheels that are complete with the inertia inherent in these elements, contain the brake component and their dynamic behavior is based on the magic formula;
- There is also a differential that divides the torque coming from the power train to the two wheels of the front axle.

The power-train model consists of:

- Engine model;
- Drive-line model;
- Model of the electric power-train.

Engine Model

The engine model shown in Figure 2.10 is a model that simulates the behavior of the engine to a throttle demand in terms of fuel consumed and actual engine torque response. It also includes elements that model idle control and resistive torque of the engine. The calculation of the fuel consumed is carried out through look-up tables given the torque and angular speed conditions to which the engine is subjected and also takes into account the fuel cut-off in the presence of the engine stop signal and the amount of additional fuel necessary to turn on the engine.

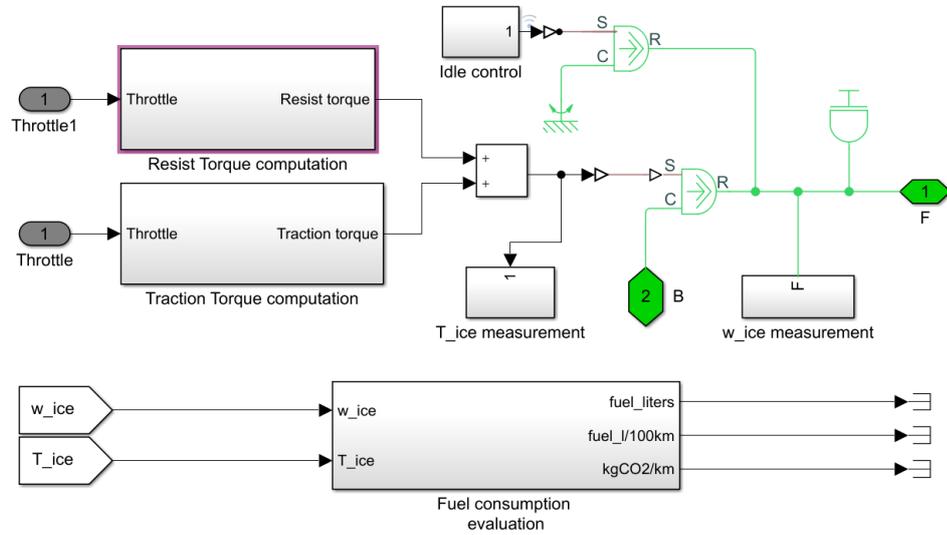


Figure 2.10: Engine simulink model.

Drive-line Model

The driveline is, in this case, composed of the clutch, the torque converter, gearbox and differential.

The differential, clutch and torque converter models were made through constructs provided by Simscape, while the gearbox model, in Figure 2.11, contains some modifications.

The gearbox is made through the model of variable ratio transmission which ratio is given as input and is the one that corresponds to the gear engaged.

The gear is decided in the area highlighted as "Gear Control" where there is both a shift logic that is based entirely on the speed of the vehicle indicating the angular speeds of shift and down-shift of each gear and the gear decided in the controller according to the logic of ECMS-GC that will be treated in section 3.2. In the "clutch k0 actuation" block, the clutch actuation is managed.

Model of the electric power-train.

The electric power-train model consists of a battery and an electric motor.

The battery model, Figure 2.12, is formed by the "Battery" block belonging to the Simscape library that simulates the dynamic trend of the battery in terms of capacity, voltage and current. The battery voltage, measured by a voltage sensor, is used to find the current required by the electric motor by dividing the required power.

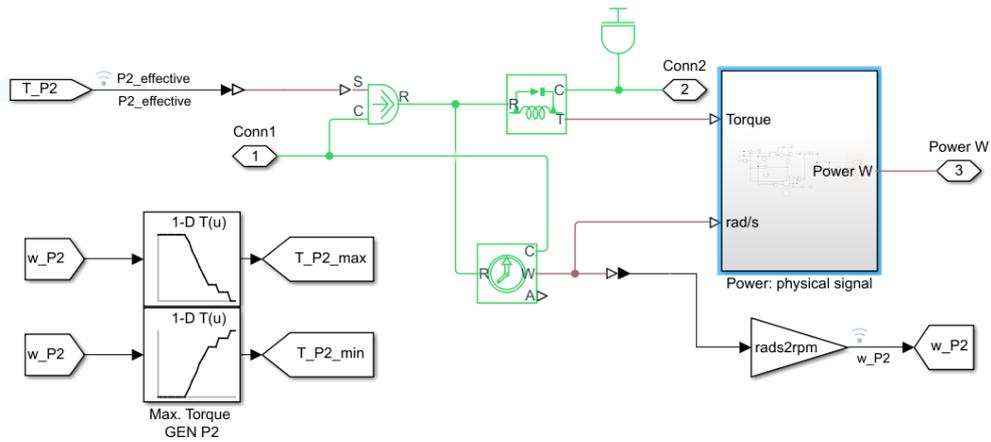


Figure 2.13: Electric motor simulink model.

2.4.2 Controller

The "controller" system contains all the control logic that governs the vehicle and the logic developed in this study, a graphic representation is given in section A.2. The controller inputs are:

- The angular velocity of the crankshaft;
- The current speed of the vehicle;
- The State of Charge;
- The gear engaged;
- The reference speed or acceleration required according to the selected reference;
- The type of power split logic to adopt;
- The speed of the electric motor;

In the model it is possible to choose, through variant subsystems managed by the Simulation Interface, the type of reference to be adopted which can be a speed profile, the command given by an Adaptive Cruise Control that follows a lead vehicle, or an Adaptive Cruise Control with the Stop profilation feature, that is to say a speed profiling for the stop that prefers energy efficiency treated in [16]. It is also possible to choose the type of throttle control, that is, the control that manages the required torque in response to the difference between reference and feedback that can be with respect to speed or acceleration and of a static or adaptive

type (through the implementation of fuzzy logics that analyze a prediction of the speed profile and regulate the aggressiveness of the controller, treated in [16]).

The controller mainly manages the power split between the engine and the electric motor according to the logic of ECMS that will be treated in chapter 3. In addition to this, the Start&Stop (described in chapter 4) is also controlled and the implementation of the clutch and brake is managed.

The outputs of the controller are:

- The torque of the electric motor;
- The torque of the engine in the form of a throttle signal;
- The pressure to be applied to the brakes;
- The signal of clutch implementation;
- The gear in case the ECMS-GC algorithm is used;
- The engine stop signal.

2.4.3 ADAS inputs

ADAS signals are a crucial point in the development of this thesis as it aims to define high-level logic based on the perception of the environment to reduce vehicle emissions.

In particular, the developed logic that will be described in chapter 4 and the Adaptive Cruise Control, needs information that can be obtained from the processing of data from the available hardware described in section 2.3 such as:

- Relative Speed and distance to the lead vehicle;
- Recognition of the stop situation;
- Distance relative to the stop.

For the validation of the algorithms, models have been developed for the simulation of signals from ADAS. The first model simulates ideal ADAS signals, calculated without being affected by any disturbance; The second simulates the hardware adopted and the perception algorithms to be implemented, making the measurement and the signal affected by disturbances and more similar to reality; the third model excludes sensors.

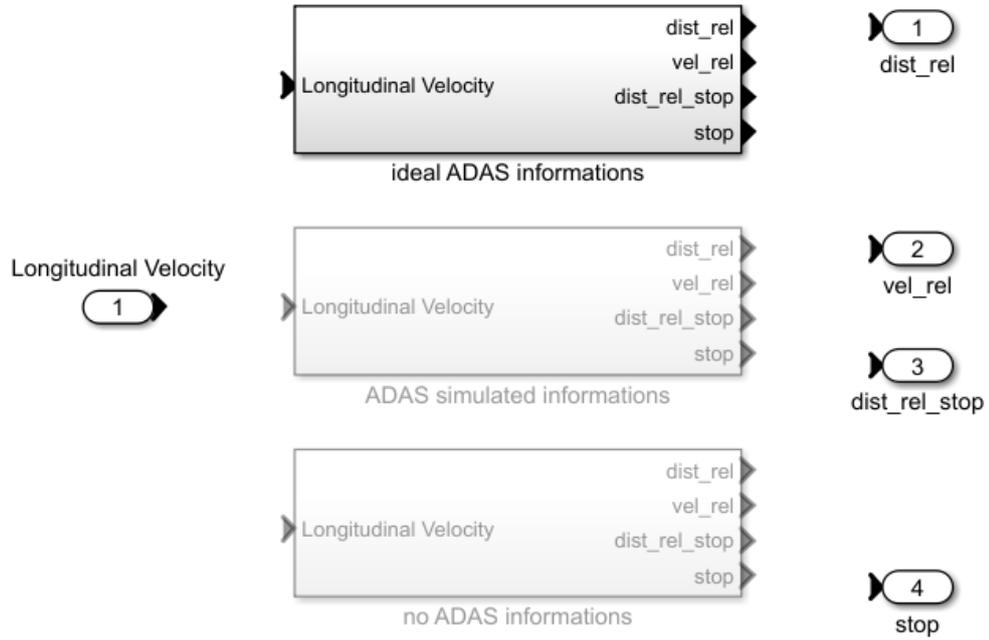


Figure 2.14: ADAS models options.

Ideal ADAS model

The model of the ideal ADAS signals has as input the current speed of the vehicle and the speed profile that follows the lead vehicle and calculates as output the distance and relative speed from the lead vehicle, the presence of a stop in the next twenty seconds and the distance from this stop.

The stop is identified by analyzing the speed profile twenty seconds in advance and determining whether the speed in this profile reaches zero. The signal is then maintained until the current speed profile reaches the stopping point. Distances shall be calculated as the difference between the distance traveled by the reference speed profile and that covered by the vehicle which are retrieved through the integration of the velocity in time, as in Equation 2.2. In particular, if it has to be calculated the distance from the stop, it must be considered as a reference profile the profile anticipated in time, otherwise, if it has to be calculated the distance from the lead vehicle it must be considered the current profile.

$$dist_{rel}(t) = \int_0^t (v_{leadvehicle} - v_{vehicle}) dt \quad (2.2)$$

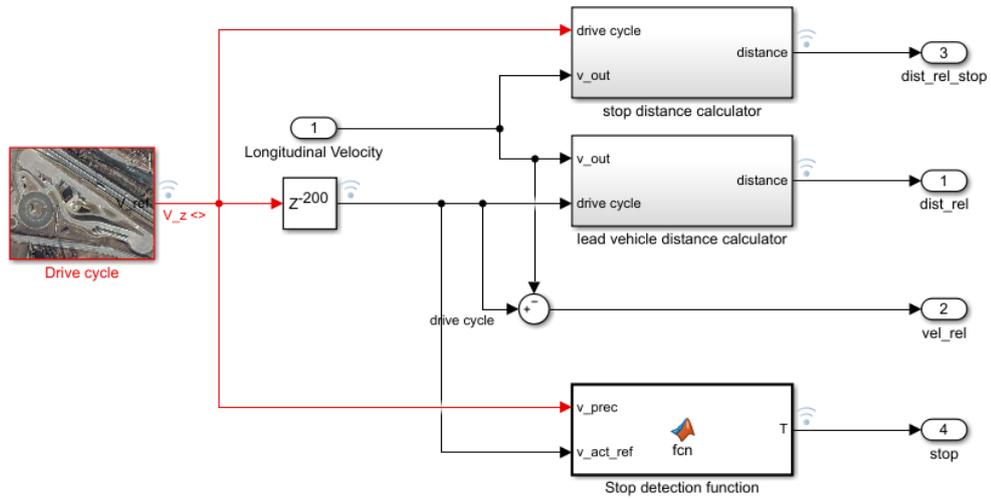


Figure 2.15: Ideal ADAS model.

The relative speed is finally calculated as the difference between the speed of the lead vehicle and that of the vehicle itself.

Simulated ADAS model

The model of the simulated sensors provides signals more similar to those that in reality could be provided by the hardware, being affected by errors and disturbances.

The model was derived by appropriately modifying the example provided by Mathworks in [17], imposing on the lead vehicle the speed profile of interest and using the blocks related to the perception of the lead vehicle, the simulation of sensor measurements and the calculation of speed and relative distance.

In Figure 2.16 is shown the model where it is possible to distinguish the subsystems "Vehicle and Environment" and "Sensor Fusion". The first contains the elements that simulate the behavior of the lead vehicle and the ADAS hardware by creating the output signals of the radar and stereo camera. The second contains the logic that analyzes the signals and extrapolates the necessary information, creating clusters of measuring points, identifying the lead vehicle to follow and calculating speed and relative distance.

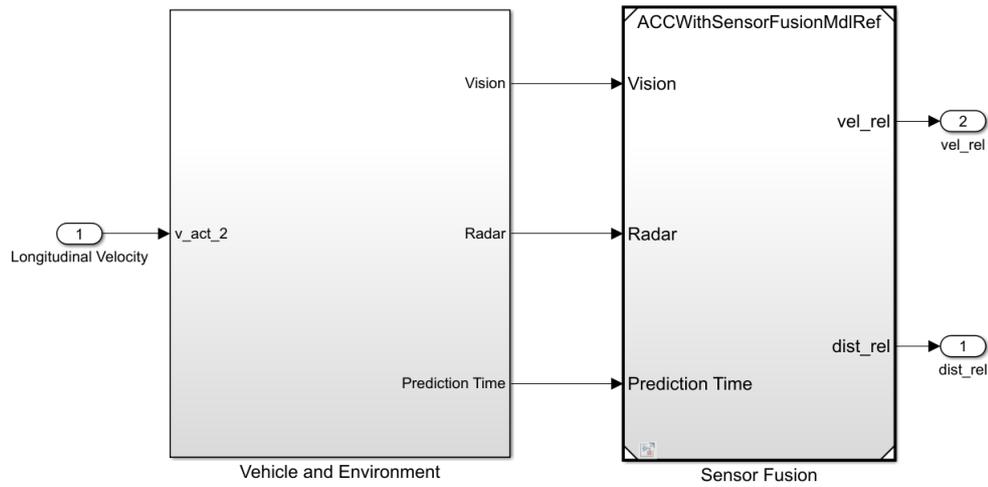


Figure 2.16: Simulated ADAS model.

2.4.4 Simulation Interface

Dealing with such a complex model requires suitable tools to make navigation and parameters set up easier. During the studies a lot of simulations were done, each of them sometimes differing by just one parameter or one tiny change in the model, and the continuous model navigation to reach the region in which the desired modification has to be done was very time-consuming, especially when a trial and error approach was used.

For that reason, a Simulation interface was developed using Matlab LiveScript, a useful tool that let generate an interactive document that combines both Matlab code and formatted text parts [18]. This kind of document can be composed of interactive sections where one can modify a parameter (edit field), choose a value from a list (drop-down), tune a parameter between a range (numeric slider) or control a Boolean value through a check-box. Each time one of those control tools is modified the script will be run and all the parameters will be saved on the workspace or in a “.mat” file, making them accessible by the model.

The Simulation Interface is what makes easy and automated the switch between one simulation set-up and another.

With this tool, all the parameters of interest are collected and accessible from one place and it is possible to modify certain parts of the model, to be able to change the hardware that is going to be simulated directly from this interface. The developed interface counts two sections, one for the simulation configuration and the other for the ECMS.

In the first section, it is possible to

- choose what kind of algorithm employ as gear shift logic (consumption reduction-based or velocity-based);
- select the power split logic(ICE only, ECMS, ECMS-GC);
- Select what kind of ADAS information are available (Ideal, simulated, none);
- select the throttle control system (velocity feedback, velocity feedback adaptive control, acceleration feedback, acceleration feedback adaptive control);
- Choose what kind of reference employ (velocity profile, Adaptive Cruise Control, ACC with stop profilation);
- choose what kind of algorithm employ as Start&Stop logic (Standard, ECMS-based, ADAS-based).
- set-up initial SOC, simulation duration, and time step.

In the second section instead, one can regulate the weight used inside the ECMS algorithm regarding the gear change (gear change weight, last gear change elapsed time weight), the amount of given torque (insufficient torque weight), or the weights concerning the combustion engine transient optimization (weight given to delta torque from the previous step) and the kind of $f(SOC)$ used.

A graphic overview of the Simulation interface is available in section A.3.

2.5 Speed profile for simulation

For the simulation of the vehicle and the obtaining of results for the validation of the algorithms, it is necessary to choose a speed profile to be performed that is as representative as possible of the various driving scenarios that the vehicle will face in real situations.

For this purpose, there are regulated and specially designed speed profiles that are used during the homologation phase to evaluate the emission class of the vehicle under consideration. These tests, as reported in [19] are:

- ECE 15 + EUDC: The original EU test cycle (also known as the MVEG-A test), including urban and extra-urban segments, performed from a hot start;
- NEDC: Effective from 2000 (Euro 3), the ECE 15 + EUDC test was modified to eliminate the 40 s engine warm-up period before the beginning of emission sampling. This modified cold start test was referred to as the New European Driving Cycle (NEDC) or as the MVEG-B test;

- WLTP: The Worldwide harmonized Light vehicles Test Procedure (WLTP) and the corresponding Test Cycle (WLTC) replaced the NEDC procedure.

Since the legislation currently in force is the WLTP, it was decided to use this cycle as a speed reference.

2.5.1 WLTP homologation cycle

The Worldwide harmonized Light vehicles Test Procedures (WLTP) published as UNECE Global technical regulation No 15 (GTR 15), define several procedures that are needed to type approve a vehicle[20].

It contains several WLTC cycles applicable to vehicle categories of different power-to-mass ratios, but, the provision for the HEVs categories specifies to consider the class 3a or 3b (depending on the maximum velocity) for those kinds of vehicles. The specification of the normative also distinguishes different categories of HEVs for which different evaluation criteria are picked:

- OVC-HEVs (off-vehicle chargeable hybrid electric vehicles);
- NOVC-HEV (not off-vehicle chargeable hybrid electric vehicles);
- PEV (pure electric vehicles).

In the case considered by this study the vehicle falls in the NOVC-HEV category, 3b class, which involves a complete Class 3b WLTC cycle done in charge sustaining mode, so that the initial and final States Of Charge are nearly the same.

The cycle velocity profile can be seen in Figure 2.17 and its specifications are available in Table 2.3.

Phase	Low 3	Medium 3-2	High 3-2	Extra-High 3	Total
Duration [s]	589	433	455	323	1800
Stop Duration [s]	156	48	31	7	242
Distance [m]	3095	4756	7162	8254	23266
% stop	26.5%	11.1%	6.8%	2.2%	13.4%
v_{MAX} [km/h]	56.5	76.6	97.4	131.3	
v_{avg} [km/h]	18.9	39.5	56.7	92	
a_{min} [m/s]	-1.47	-1.49	-1.49	-1.21	
a_{MAX} [m/s]	1.47	1.57	1.58	1.03	

Table 2.3: WLTC class 3b cycle characteristics.

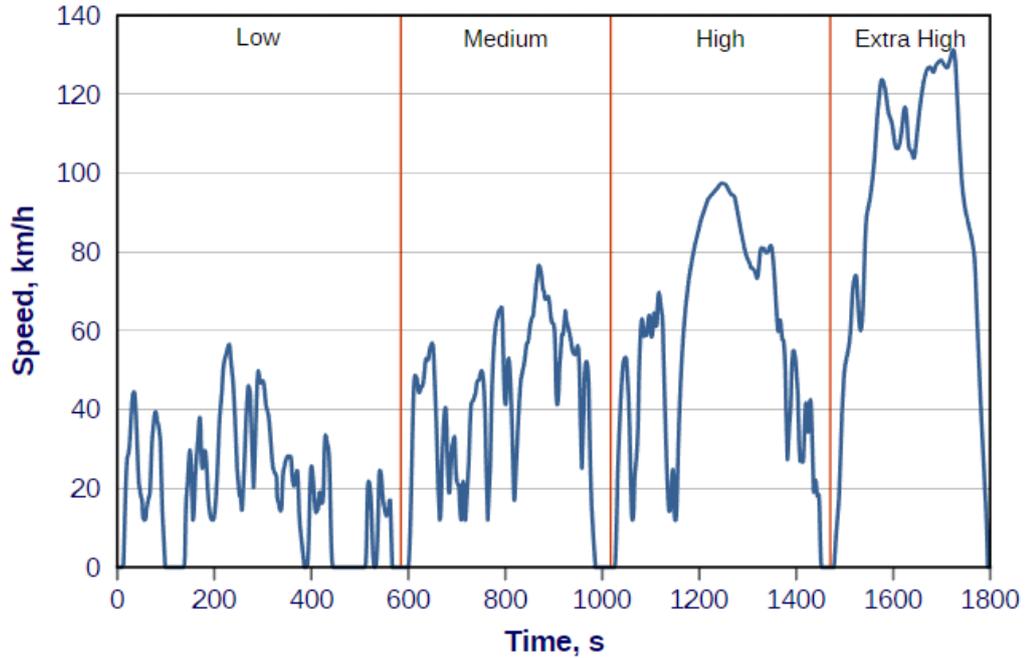


Figure 2.17: WLTC cycle for class 3b vehicles.

2.5.2 Reference profile actuation

The velocity profile chosen is used to act as a reference to the external control loop that controls the vehicle dynamics.

The control loop could be in velocity, in acceleration, or done through the command computed by an Adaptive Cruise Control.

The error between the reference and the feedback is then converted into a torque request by a throttle control module and then is managed by the power split algorithm to determine what power source has to handle it.

The controller output is the commands to the plant's power-unit element which actuation produces a response of the vehicle according to its dynamics. At this stage, the feedback can be measured, and so on.

Velocity or acceleration loop

The way the external loop is implemented is straightforward. As can be seen in Figure 2.18, the velocity profile is used as a reference for the control variable adopted, whether it is the velocity or the acceleration.

in the first case, the reference will be the WLTC velocity profile itself with the vehicle velocity as feedback, while in the second case will be considered the derivative

of it.

Those two options require a different throttle control module according to the different dynamics that the two control variables have.

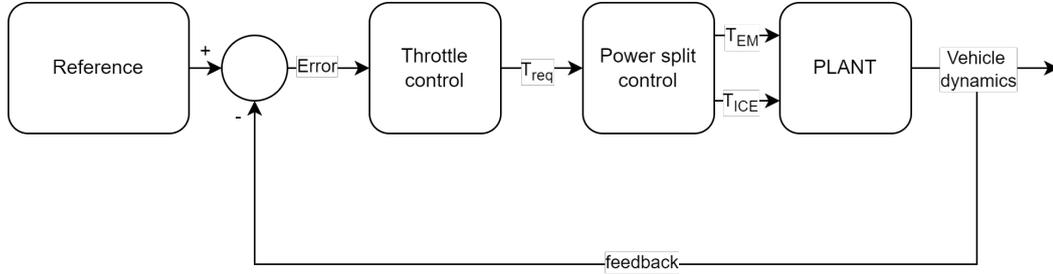


Figure 2.18: External reference profile control loop scheme.

Adaptive Cruise Control

The Adaptive Cruise Control, referred to as ACC, is a system that let the vehicle track a certain set velocity (as the normal cruise control) while maintaining a safe distance from the vehicle in front.

For the purpose of this study, it has been employed an ACC, based on the one from Mathworks example [17], to follow a lead vehicle that travels a WLTC cycle. Another application of this system was for the actuation of the velocity profile originated from the algorithm of eco stop profilation, treated in [16], which computes the best speed profile to be followed in order to stop the vehicle in the most efficient way once the stop has been detected.

As can be seen in Figure 2.19 the ACC system takes as inputs some set information, (set velocity and time gap) and some feed-backs, both from plant measurements (longitudinal velocity) and ADAS (relative distance and velocity from the lead vehicle).

Those inputs are given to an MPC controller that solves a QP optimization problem to retrieve the optimal manipulated variable (acceleration) that takes the measured output (vehicle velocity) to be closer to the reference signal (set velocity) in the prediction horizon (set by the user), taking into account the disturbances and constraints (safe distance from the lead vehicle and limits of the vehicle's dynamics). For this application, the system has been slightly modified with respect to the one in example [17] :

- The computational effort has been reduced decreasing the prediction and the control horizon;

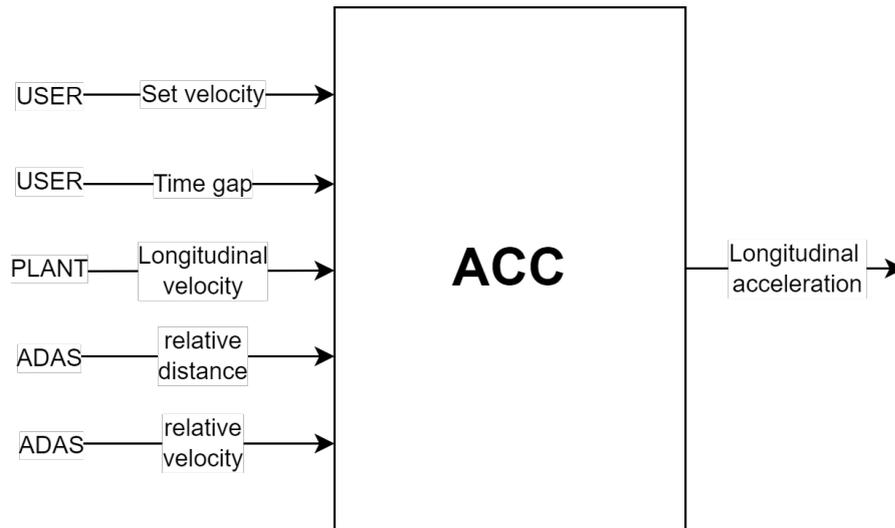


Figure 2.19: ACC input-output scheme.

- Since the scenario under study was the lead vehicle's following, it has been given its velocity as set velocity or the stop velocity profile when is used;
- The parameters of maximum velocity and acceleration have been changed with the ones of the vehicle under analysis.

The output of the system is the acceleration that the 'ego' vehicle must have to follow the lead vehicle and in the control loop is converted into a throttle command thanks to a feedback loop in acceleration, as in Figure 2.20.

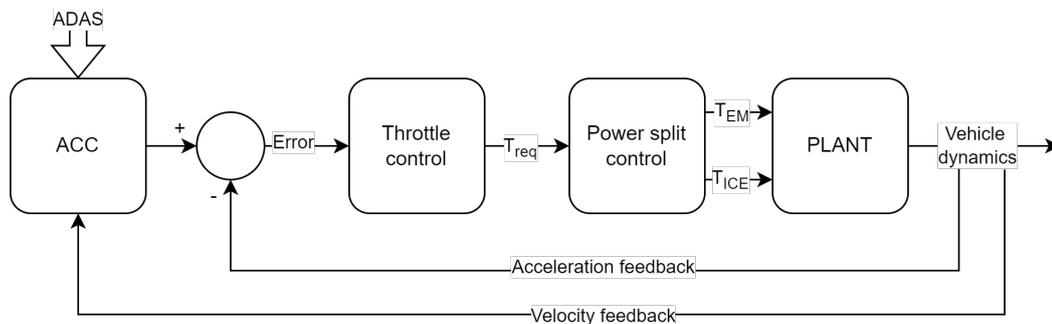


Figure 2.20: External ACC control loop scheme.

An extract of the simulation is reported in Figure 2.21 where it can be observed the performances of the system in following the lead vehicle on the first five-hundred seconds of the WLTC cycle. As can be seen, the system gives more importance in maintaining the correct safe distance instead of tracking precisely the velocity

profile, and this lead to the avoidance of unnecessary spikes on the velocity and consequently to consumption reduction.

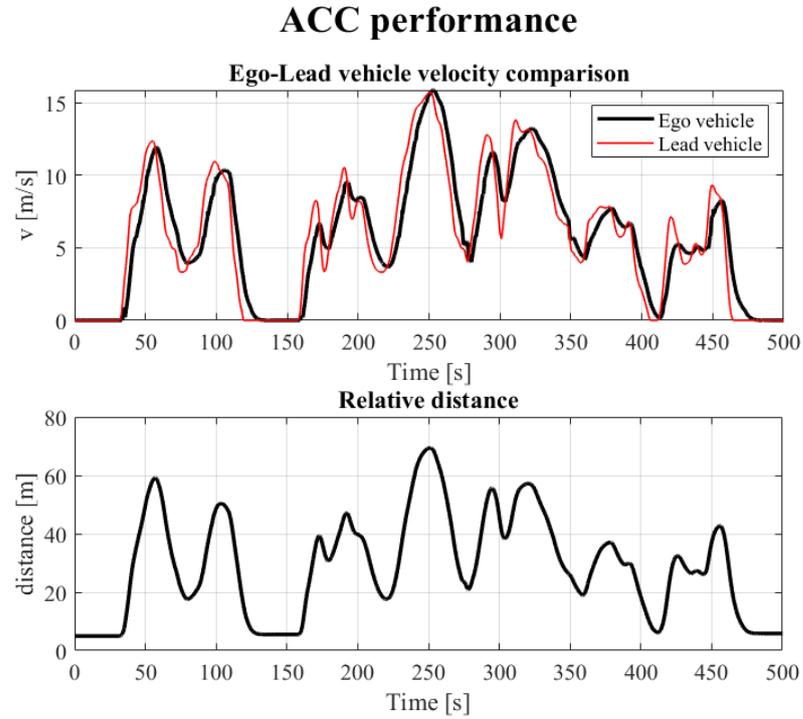


Figure 2.21: ACC performance in following a vehicle that tracks a WLTC cycle.

Chapter 3

ECMS with Gear Control

One of the major control problems related to hybrid vehicles concerns the power division between the two sources of the power train.

As described in section 2.2 a hybrid electric vehicle features two different motors and managing the way they are used heavily affects the results that the system wants to achieve, whether the objective is the vehicle's performance or the pollutant emissions as in this case.

The most common algorithm to get through this problem is the Equivalent Consumption Management Strategy, referred to as ECMS, which is going to be described in this section.

The conducted study aims at improving this kind of logic adding new features and considerations that uses information available in the vehicle to let the algorithm find a better sub-optimum configuration. In particular, the solution found includes the gear to engage as controlled output gaining the ability to analyze, chose and reach all the working points that let the power train provide the requested power.

3.1 State of the art

3.1.1 Introduction

This section will describe the concept of ECMS and the algorithm that was already implemented in the model before any modification was applied.

The ECMS is a well-known algorithm that can compute the optimum solution in terms of minimization of a suitable cost function taking as input the instantaneous conditions of the vehicle and giving as output the optimal power split between the combustion and the electric motor.

All in all this controller could be seen as a MISO system that takes as input the torque requested, the maximum torque that the two power sources could give, the shaft speed and the $f(SOC)$ and gives back the optimum torque split.

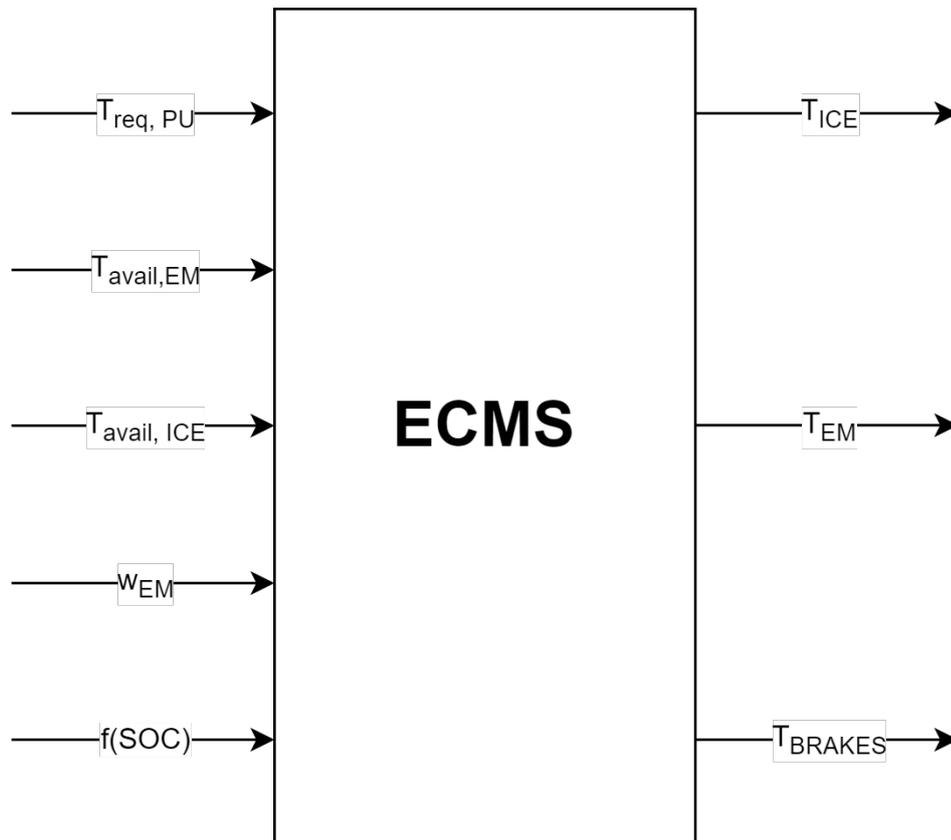


Figure 3.1: Input-output representation of the standard ECMS algorithm.

3.1.2 Input description

The starting point of the process is the input conditions and requirements, which include :

- The available torque from the motor and the engine at that specific angular velocity;
- The actual angular velocity of the motor shaft;
- The motor torque needed from the driver.

The first elements of the list are retrieved from a one-dimensional lookup table that maps the actual motor shaft angular velocity to a maximum payable torque by the motors by interpolating a table of values given from motor data. The extrapolation is made through the linear interpolation method, automatically done by the Simulink block in Figure 3.2.

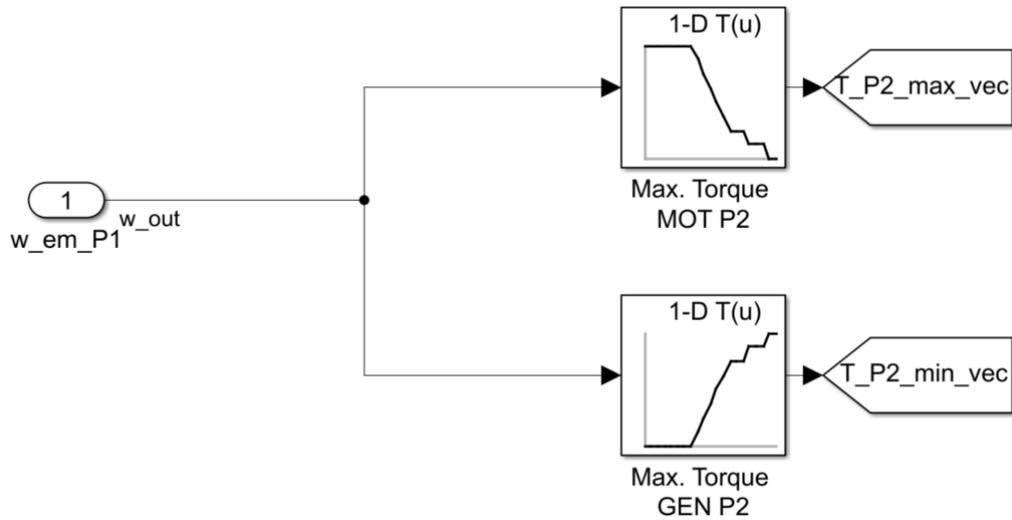


Figure 3.2: Lookup table for maximum EM torque.

The second input is directly measured from the plant through an ideal angular velocity sensor, featured by a Simscape block;

The last one is supplied by the driver model that converts an error in velocity or acceleration, depending on which kind of reference is given, into a request in torque. This subsystem models the behavior of a driver that has to follow a precise reference acting on the throttle pedal. It has been modeled through a PI controller, Figure 3.3, that computes a signal which value is included in the interval $[-1; 1]$, simulating

the position of the throttle pedal, based on the error between the reference signal and the feedback of the behavior of the plant.

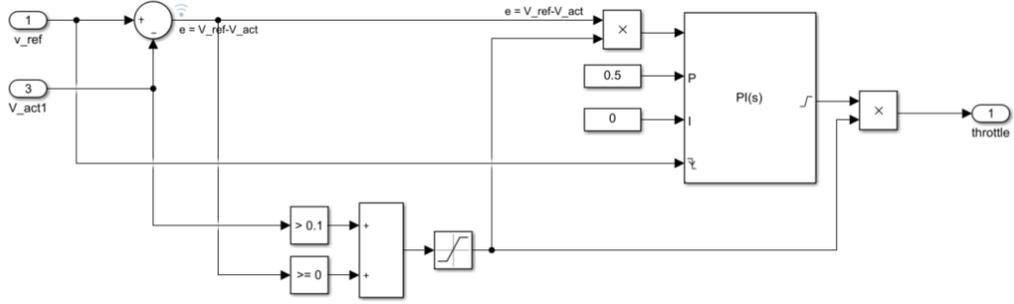


Figure 3.3: PI driver model

The throttle pedal signal is then converted into a requested torque by another Simulink subsystem that can originate both the torque requested by the power unit and the one needed from the drive line (just considering the motors and the gearbox). In the first case, the requested torque is computed as:

$$T_{req,DL} = T_{max}Thr \quad (3.1)$$

Where $T_{req,DL}$ is the torque requested from the drive-line, Thr is the throttle pedal signal and T_{max} is the maximum payable torque from the drive-line overall, calculated as:

$$T_{max} = T_{max,PU} * max(Gr) \quad (3.2)$$

Where Gr is the vector containing all the gear ratios and $T_{max,PU}$ is the maximum payable torque by the Power unit.

the calculated $T_{req,DL}$ can be converted into the requested torque by the power unit considering the actual gear engaged through the formulation in Equation 3.3 translated into the Simulink block in Figure 3.4.

$$T_{req,PU} = \frac{T_{req,DL}}{Gr(Gear_{act})} \quad (3.3)$$

The controlled variable could be acceleration or velocity. In the first case, the controller has to be faster and more aggressive to face the quicker dynamic of this signal, it results to be less precise due to the inaccuracy that characterizes the

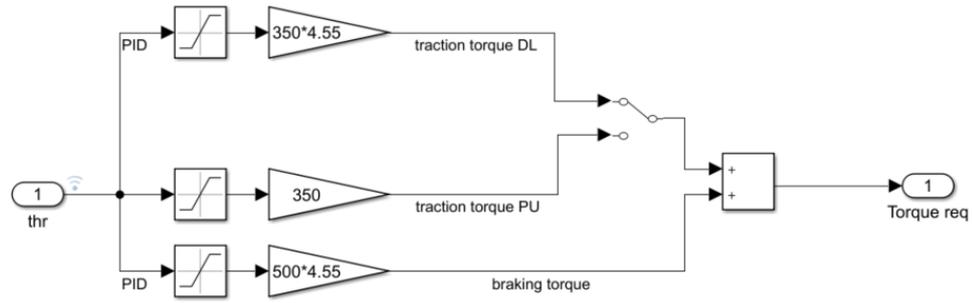


Figure 3.4: Simulink block scheme for requested torque computation

feedback, involving also the need for some low pass filters applied to that source, but results to be a little more efficient from the fuel consumption point of view. In contrast, the second option results to be easier to implement, more stable and with a better performance in following the velocity reference but worst regarding the fuel consumption.

The choice among the two depends on the application, if one wants to simulate a real driver behavior the second choice is probably more addressable, instead if an adaptive cruise control is involved, the typical output of such a device is usually an acceleration signal, so the first option looks more suitable.

3.1.3 Torque split computation

With all those information the algorithm tries through a for cycle all the possible splits of the requested torque, considering the maximum torque constraints, the avoidance of impossible or undefined map points and the completely electric regenerative braking, compute the value of the cost function and choose the optimal configuration taking the one with minimum cost, as represented in the logic scheme in Figure 3.5.

The torque split process begins by discerning the braking and traction situations. In the first one, the requested torque is completely assigned to the electrical machine, which, acting as a generator and recovering some energy is for sure the best solution for such a need.

In the second case, the algorithm calculates the vector of a hundred possible splits locating the positive (3.4) and negative (3.5) extremes as the maximum percentage of requested torque that could be taken by the electric motor or generator and generating an equally spaced vector of a hundred elements between them (3.6).

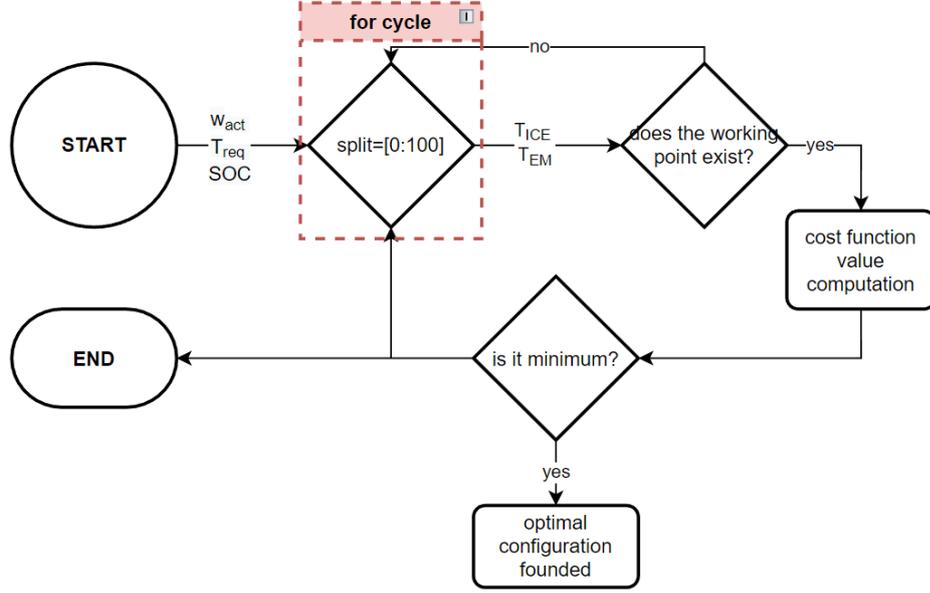


Figure 3.5: Logic scheme of the standard ECMS algorithm.

$$\bar{u} = \min \left(\frac{T_{mot,avail}}{T_{req}}, 1 \right) \quad (3.4)$$

$$\underline{u} = \max \left(\frac{T_{gen,avail}}{T_{req}}, -1 \right) \quad (3.5)$$

$$u = \text{linspace} [\underline{u}, \bar{u}, 100] \quad (3.6)$$

This vector represents all the possible percentages of requested torque that could be assigned to the electrical machine and could be positive if it is used as a motor or negative if it is used as a generator involving load shifting (the combustion engine would supply more torque than the requested one that is converted into electrical energy through the generator).

The for cycle tries one split at a time, calculating first the combustion engine torque (taking into account the saturation at the maximum torque available) (3.7) and then assigning the remaining torque to the electrical machine being sure that its limits are respected (3.8).

$$T_{ICE} = \min (T_{req}(1 - u(i)), T_{ICE,avail}) \quad (3.7)$$

$$T_{EM} = \max(\min(T_{req} - T_{ICE}, T_{mot,avail}), T_{gen,avail}) \quad (3.8)$$

Each of those torque couples is then converted into the equivalent fuel consumed and compared to each other to choose the best combination with respect to the minimization of the cost function.

3.1.4 Cost function description

With all the inputs and the process to retrieve all combinations of torque being explained it is possible to examine in depth how the cost function works.

The objective function of such algorithm (3.9) is to reduce the equivalent fuel consumption which is a computed quantity of fuel that takes into account the real fuel demanded by the ICE engine and the virtual fuel that corresponds to the electric power that is requested in the considered configuration.

$$J = \dot{m}_{ICE} + \dot{m}_{EM} \quad (3.9)$$

Minimizing this cost one should be able to achieve the configuration that better full fills the requested power wasting the minimal equivalent fuel possible. Minimizing the fuel is equivalent and proportional to minimizing the pollutant emissions, which is the main objective of this study.

To get deeper into the analysis of the two terms of the cost function, the first one represents the amount of fuel, experimentally retrieved, that the requested condition on the combustion engine would request. The data are given as a motor map, Figure 3.6, that links for each working point, identified as a combination of angular velocity and torque, to the test-measured fuel consumption, retrieved from the BSFC map in Figure 2.7.

The quality of this map heavily influences the performance of the whole algorithm. The more defined the map the more the algorithm is likely to take into analysis the behavior of the engine that is more similar to the real one. This encourages the algorithm to take the right decisions being more conscious of how the actual motor behaves.

In this case, the given map Figure 3.7 contains the combination of fourteen values of engine angular velocity and twenty-seven values of engine torque for a total of three-hundred-seventy-eight points with hundred-fifty of them that are actually non-reachable being over the maximum torque characteristic of the considered engine.

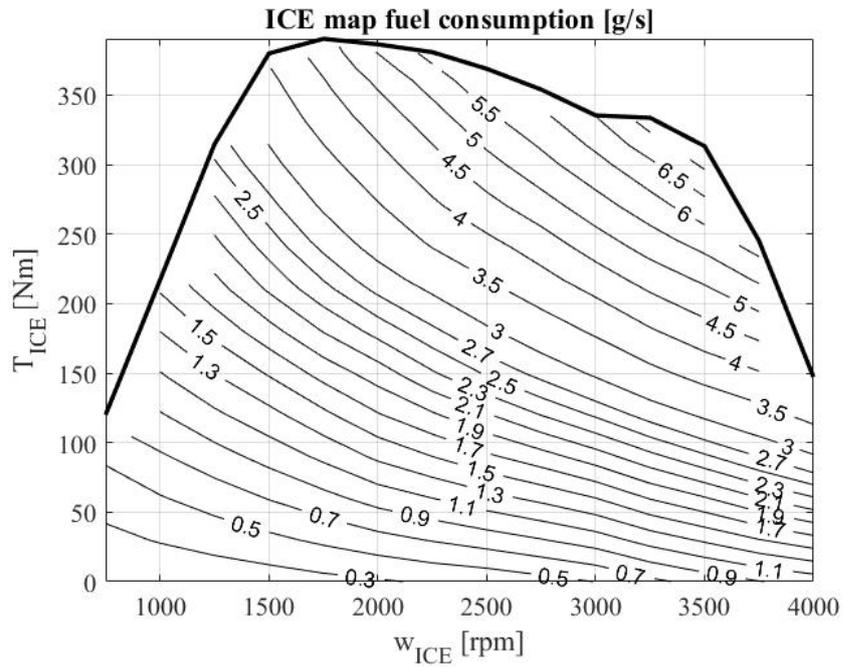


Figure 3.6: Fuel consumption map of the engine.

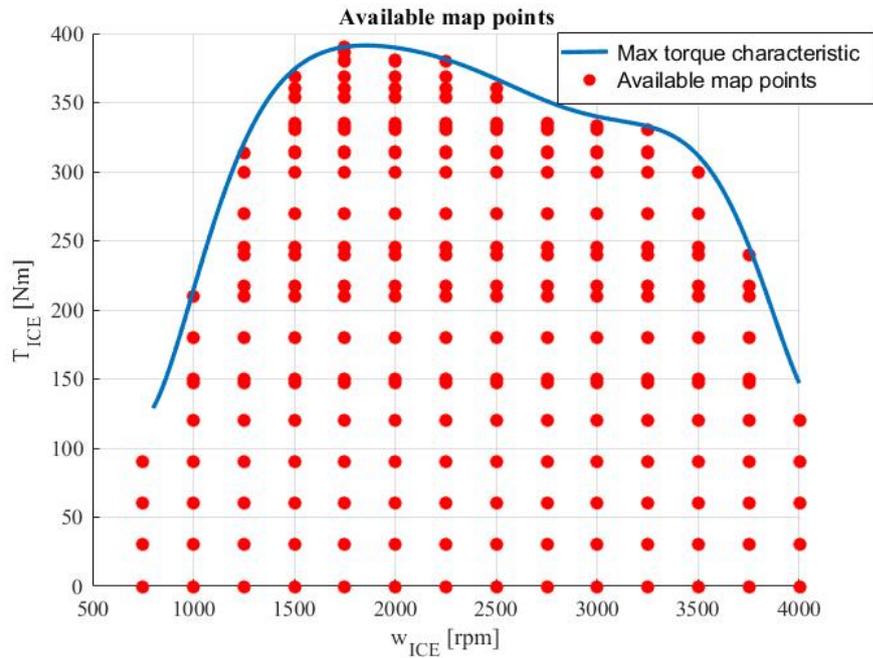


Figure 3.7: Available points of the engine map and maximum torque characteristic.

With those data available, a sharpening of the map was needed, otherwise, the information captured from it would both be imprecise or in the worst case lead situation in which the addressed working point would not exist or be outside of the maximum torque curve. To do this interpolation methods were adopted to get intermediate working points, that were combined with suitable conditions and algorithms that will be explained later to avoid infeasible points. To be redundant and improve the avoidance of such points some if-else conditions that place enormous weights as \dot{m}_{ICE} have been placed to be sure that those configurations are certainly excluded by the cost function minimization if nothing else.

The second term is instead something computed, a virtual weight that represents the equivalent fuel consumption that corresponds to the energy consumed by the electric machine. This weight is originated from thermodynamics theory computing the quantity of fuel that would be needed by the combustion engine to produce the electrical energy needed by the motor to supply the requested power.

$$\dot{m}_{EM} = K P_{bat} f(SOC) \quad (3.10)$$

where P_{bat} is the power delivered by the battery which formulation varies depending on the mode the electrical machine is used (generator or motor)

$$\begin{cases} P_{bat} = \frac{T_{EM}\omega_{EM}}{\eta(T_{EM},\omega_{EM})} & \text{if } T_{EM} \geq 0 \\ P_{bat} = T_{EM}\omega_{EM}\eta(T_{EM},\omega_{EM}) & \text{if } T_{EM} < 0 \end{cases} \quad (3.11)$$

In this formulation, one can observe how the effect of the efficiency is different depending on the situation, in fact, while considering the electrical machine as a generator the efficiency is in the conversion of mechanical to electrical energy and so the energy that will reach the battery is smaller than the mechanical energy (efficiency at the numerator), in the traction case the efficiency lead to electrical losses that require more energy from the battery than the one actually converted into mechanical energy (efficiency at the denominator).

Another thing to point out is the term $\eta(T_{EM},\omega_{EM})$ that, as indicated, is the efficiency that depends on the condition of torque and angular velocity retrieved from a given motor map produced experimentally in Figure 2.8.

As far as the term K is concerned, it links the battery's electrical energy or power to the amount of fuel that the considered combustion engine would consume to produce that same quantity. From the dimensional analysis is simple to retrieve that the unit of measurement of this constant is $[(g/s)/W] = [g/J]$ which is the same as the inverse of the calorific value of the fuel. In particular, the formulation of this factor is

$$K = \frac{1}{Q_{fuel}\eta_{ICE,MAX}} \quad (3.12)$$

Where Q_{fuel} as said is the calorific value of the fuel that is a known parameter measured in $[J/g]$ that identifies the quantity of joule that one gram of that specific fuel can produce from a chemical point of view. Since the goal is to evaluate the fuel needed by the specific engine it is important to include in the formulation of K the term $\eta_{ICE,MAX}$ that identifies the fraction of chemical energy that the engine is actually able to convert into mechanical.

The last term that has to be pointed out is $f(SOC)$ which is a multiplicative factor that influences the weight of the electric power in the cost function depending on the actual value of the state of charge. It is of fundamental importance to adapt the behavior of the algorithm to the condition of the battery to preserve the battery or even enhance the recharge through load shifting when the charge is very low, or let the battery discharge in the opposite situation. It can be also used to maintain a certain value of charge waiting as a proportional controller the variance from the reference.

There are several ways to think about an effective $f(SOC)$ and it deeply depends on the type of hybrid vehicle where is going to be applied. With reference to the homologation normative [20], it is specified that dealing with a plug-in vehicle or not changes the way the battery has to behave in a cycle. With a plug-in vehicle, the homologation process expects one or more (depending on the size of the battery) WLTC cycles in charge depleting mode, and one cycle in charge sustaining mode. Whereas if a non-plug-in vehicle is considered, the most important thing that the normative expects is that the initial and the final battery charge must be almost equal.

It seems clear that those two situations require completely different $f(SOC)$, for the first one this term must lighten the electrical energy weight in the loss function when in charge depleting mode and make it heavier when the SOC goes below the preset threshold, regulated by the normative; analyzing the second situation the purpose of this term is to let the SOC oscillating around a reference value that is chosen to be 50% so that the battery is preserved standing for sure in the 20%-80% window.

To obtain the requested performance, dealing with a plug-in vehicle, the solution adopted for this function was a hysteresis function implemented through a preexisting Simulink Relay block, Figure 3.8.

The output for the Relay block switches between two specified values. When the relay is on, it remains on until the input drops below the value of the Switch off point parameter. When the relay is off, it remains off until the input exceeds the value of the Switch on point parameter. For this application, the block takes

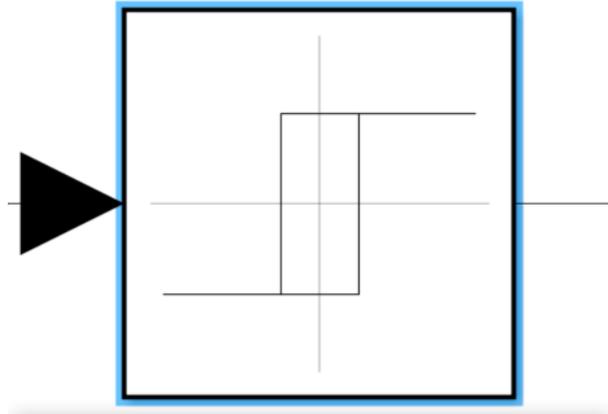


Figure 3.8: Relay Simulink block.

as input the SOC and gives as output the $f(SOC)$. When the SOC gets below the switch-off value (30%) the $f(SOC)$ became greater as to let the algorithm choose not to use the electric motor. When the SOC overcomes the switch on value (35%) the electric motor gets preferred.

3.2 Solution produced

3.2.1 Introduction

In this section will be described all the developed solutions that managed to solve some problems of the previous algorithm and make the system more efficient from an emission reduction point of view.

In particular, the focus is on the development of a new ECMS algorithm, referred to as ECMS-GC, that acquires a new degree of freedom on the control action, being able to operate a gear shift, gaining the ability to reach the instantaneous global optimum operating point of the power unit to cover the requested power output from the driver.

3.2.2 Algorithm description

The idea that lead to the development of such an algorithm is that cycling through all the possible splits of the requested torque at the power unit does not cover all the possible ways the drive-train can supply the need of the driver. As a matter of fact, one should consider that the need of the driver is to be intended as an

acceleration which is translated into the desired torque applied at the wheels and this could be converted in many different ways to a request to the power unit depending on the conversion ratio that the drive-line adopts. With that said, not considering the possible variation of that ratio, leads to finding only the local optimum point that finds the best power split among those ones with the same configuration of the gearbox without considering all the possible ones.

To better understand the problem it is useful to describe an example: let's consider a real situation and analyze the behavior of the two algorithms. The state variables of the plant are :

- $w_{ICE} = 2000$ rpm
- $T_{REQ} = 100$ Nm
- The actual gear engaged is the 6th.

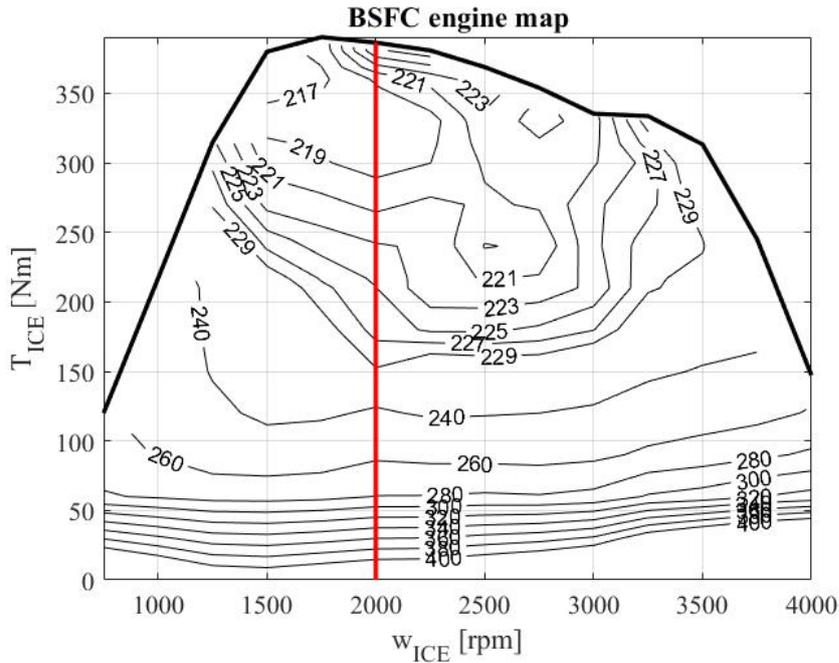


Figure 3.9: Engine's map points that are considered by the standard ECMS.

In this case, the previous algorithm would consider the map points with constant angular velocity to pick the one that, combined with the electric motor and minimizing the suitable cost function, results in the optimum. As pointed out through the picture above the considered points by the algorithm are just

the ones with the same angular velocity that represents just a small part of the map.

If the new algorithm is employed, the points that would be taken into consideration are all the ones that could be reached by changing the actual gear ratio, so, since the angular speed at the motor is directly linked to the gear ratio adopted (the variable to be preserved is the angular speed of the wheels) the reachable points are the ones at constant angular velocity for each hypothetical velocity at all different gear ratios, as shown in Figure 3.10.

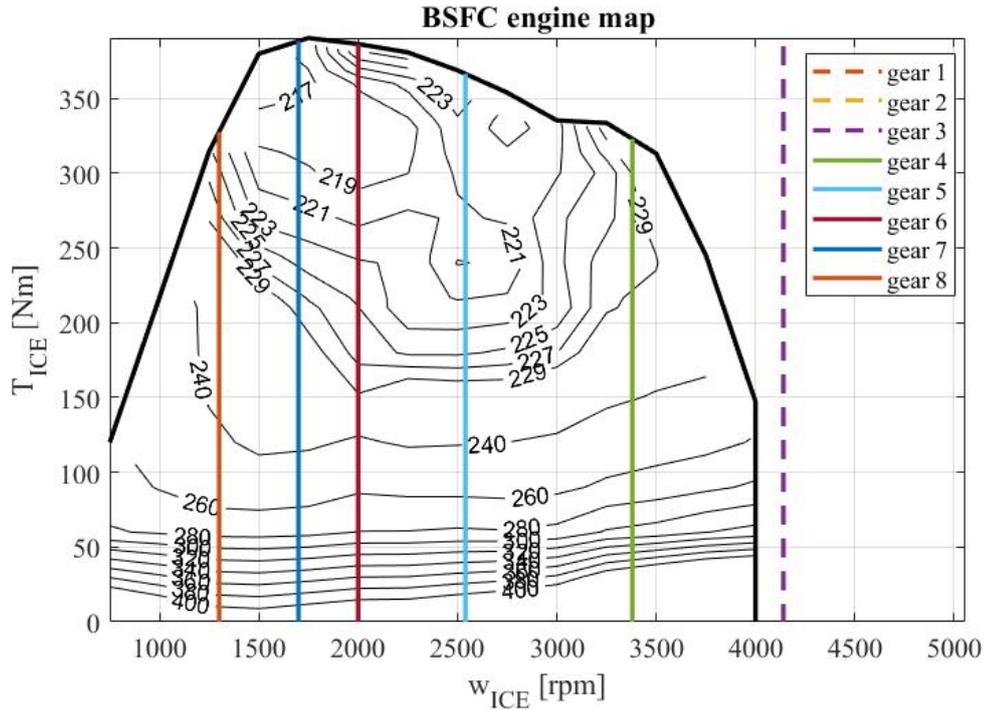


Figure 3.10: Engine’s map points that are considered by the ECMS-GC.

it is clear that the points must satisfy some feasibility and existing conditions to exclude all the points that involve conditions that the motors are not capable to withstand. That includes for example the exclusion of certain angular velocities or torque values like the dashed ones in Figure 3.10 where the points involved with the first and second gear would have an even higher angular velocity.

3.2.3 Algorithm realization

The creation of an algorithm that takes into account the gear change in the energy management strategy to aim at optimizing the consumption of a hybrid vehicle is

addressed by several authors.

In the article "Optimal integrated energy management and shift control in parallel hybrid electric vehicles with dual-clutch transmission"[21] the authors address the issue of power split control following the realization of a gear shift. Through the implementation of optimization and state prediction algorithms such as LQR and MPC, the developed algorithm manages to smooth the engine torque through torque compensation from the electric motor to prevent engine transient emissions resulting from a sudden load change.

In the article "Adaptive Equivalent Consumption Minimization Strategy With Rule-Based Gear Selection for the Energy Management of Hybrid Electric Vehicles Equipped With Dual Clutch Transmissions"[22] the authors analyze the behavior of the results obtained through the implementation of Dynamic Programming to obtain a set of rules that, if implemented, reproduce the optimal sequence of gears. The authors also introduce the concept of penalization of infeasible working points and impossible gears.

Another approach discussed in the article "Optimization of Gear Shifting and Torque Split for Improved Fuel Efficiency and Drivability of HEVs"[23] and "One-Step Prediction for Improving Gear Changing Control of HEVs" [24] introduces the use of an ECMS-based strategy to optimize shifting logic. In particular, two control algorithms have been developed that act in parallel and at different frequencies (based on the actuation speeds of the two control variables) which independently decide the best torque split under the currently engaged gear and the best gear based on the optimization of the working point through ECMS and, while the torque split control variable is chosen with a high frequency and implemented instantly, the optimal gear is first calculated with a lower frequency and then subjected to a filter that avoids oscillation and increases drivability based on the prediction of future states of the power-train.

The solution produced in this thesis addresses the gear change as a control variable for fuel consumption optimization based on the ECMS as in [23],[24] but integrates all the control variables in a single, lightweight architecture that takes into account the transient phenomenon described in [21] and the infeasible point preached in [22].

The algorithm is realized through the application of an external for-cycle that tries all the different gears and converts each time the driver requirements to the equivalent torque and angular speed at the motor shaft with the analyzed gear. After all the infeasible points are excluded and all the others are weighted, taking into consideration the transient occurrence, the more convenient is picked and given

as output, controlling both gear and torque split.

As can be seen in Figure 3.11, the differences with respect to the previously employed ECMS algorithm are consistent and are located in :

- the external for-cycle that tries all the gear
- the condition of feasibility
- the cost function
- the input and output of the controller

All the differences are going to be explained in the next sections.

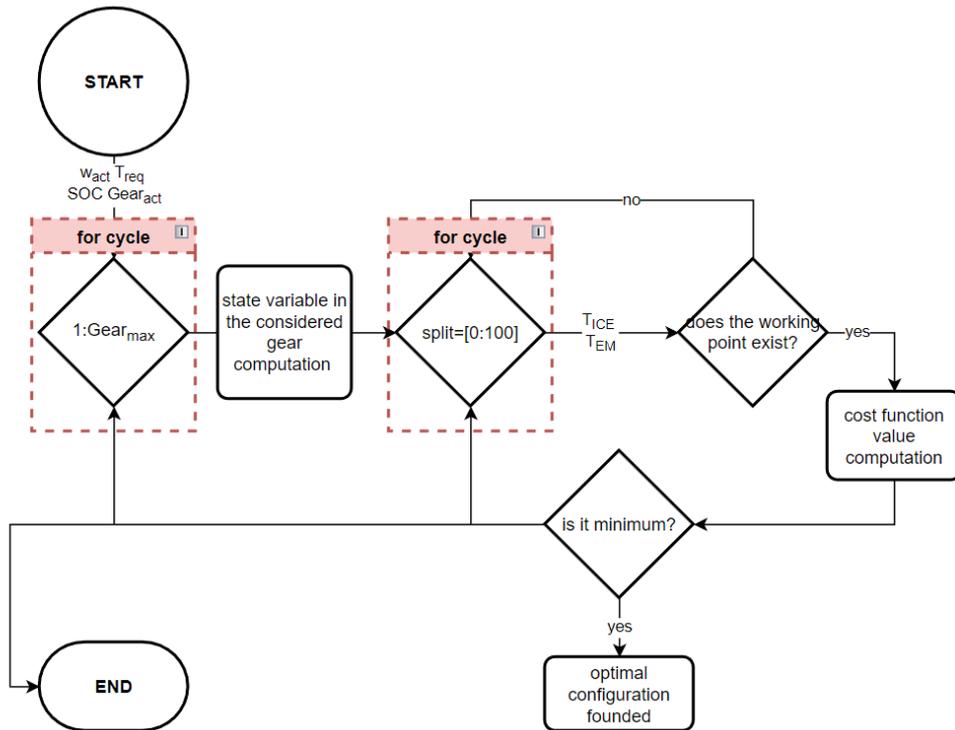


Figure 3.11: Logic scheme of the new ECMS-GC algorithm.

3.2.4 Gear for-cycle

Cycling through the gears means that the state variable must be converted into the hypothetical ones that would be present with that gear engaged. As said the need of the driver must be intended as an acceleration that is translated into a

requested torque applied at the wheel. That torque becomes the input of the new algorithm and, through the analysis of the drive-line, it is possible to calculate the equivalent torque that the power unit must supply.

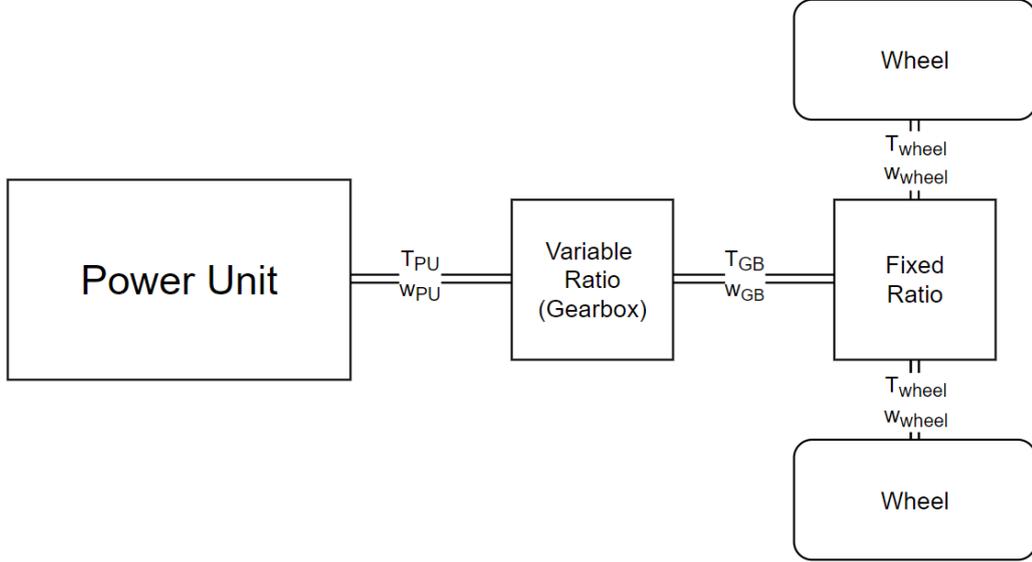


Figure 3.12: Driveline scheme of power transformers.

As the Figure 3.12 exposes between the wheel and the power unit all the drive-line can be modeled through two transformers, one that is with a fixed conversion ratio that represents the differential gear and the other that could dynamically change its ratio depending on the gear engaged that is the gearbox. With that in mind the equation to retrieve the state variables at the motor side are:

$$\omega_{PU} = \omega_{wheel} \tau(gear) \tau_{diff} \quad (3.13)$$

$$T_{PU} = \frac{T_{wheel}}{\tau(gear) \tau_{diff}} \quad (3.14)$$

Where, as indicated in Figure 3.12 T_{PU} is the computed torque at the motor, T_{wheel} is the requested torque at the wheel, $\tau(gear)$ is the gear ratio that depends on the gear engaged, τ_{diff} is the ratio of the differential gear. Due to lack of data, the mechanical efficiencies that depend also on the gear engaged are ignored at the moment, but could be easily taken into account by adding the efficiency term in the equation according to the power flow.

The inputs of the system are T_{wheel} and ω_{wheel} that for each iteration are transformed into the PU variables where the calculation in split and cost function are performed.

3.2.5 Feasibility conditions

Before doing the calculation is important to establish conditions to exclude infeasible points to unload the computational effort and restrict the decision set to only possible points.

This process guarantees that all the points that are taken into consideration and that are compared to each other are actually feasible and avoids the conditions where a working condition, due to its nonexistence in the map could be chosen by the cost function having a near-zero weight and that result to be practically impossible and probably inefficient.

The main conditions to be excluded are

- Conditions concerning the angular speed of the power unit shaft;
- Unfeasible torque conditions or working points not contained in the maps.

The first point aims to preclude from the analysis all the points that are impossible from a hardware capability point of view. In particular, its objective is to exclude the points that, due to the gear ratio adopted, are located outside of the motor maps considering the x-axis of the angular speeds.

It strongly depends on the hardware adopted since could change from motor to motor (in particular from combustion to an electrical motor, look at the two motor maps Figure 3.6 and Figure 2.8) and depends on the hybrid configuration in analysis.

Regarding this last sentence, different hybrid configurations involve different constraints on the velocity:

- If a P0 or a P1 configuration is considered, since the electric motor is directly mounted on the same shaft of the engine without any decoupling element between them, the system is constrained by the limits of the combustion engine and being those kinds of configurations usually equipped with a small electric motor, the completely electric traction is uncommon, needing to drag the fuel cut offed engine;
- If a P2, P3 or P4 configurations are employed, the two power sources are decoupled by at least one clutch element, so, the constraint depends also on the split analyzed, being the more limiting among the motors involved.

In this case, the algorithm has been developed for a P2 configuration so three different conditions have been identified. The most external one regards the highest value limit for the angular speed that corresponds to the maximum speed limit of the electric motor which is much higher than the one of the combustion engine. Over this speed, which corresponds to almost 7000 rpm, both sources of power are not able to work, so this condition involves the exclusion of that gear. The second constraint to take into account is the range of speed where the engine could work, which is indicated to be from 750 to 4500 rpm. The power split is only possible inside this range, so, for the gears whose calculated angular velocity is outside this range but still lesser than the outer upper bound, only the electric motor is made available and the engine is considered to be at idle conditions. It is to be noted that with the previous gear shift logic that is only based on an upper and lower limit of velocity that identifies the shift event, the distinction of the cases described above is not considered and, since the limits are chosen to be common to the whole power unit, a substantial quantity of possible working points is ignored and non-reachable.

Regarding the avoidance of unfeasible point and torque conditions, a lot of measures have been employed at different levels. The outer is the improvement of the look-up tables that given the motor shaft angular speed returns as output the maximum torque that each power source could supply. The analysis of this problem starts from the identification of the area of the map whose knowledge is complete from the given data, which is highlighted in the Figure 3.13.

It can be observed that the zone of interest is identified by the area subtended by the discrete connection of the known maximum torque points holding the minimum value in the interval between two consecutive points. The problem that affects the previously adopted map is that for intermediate points the output is retrieved through linear interpolation. As could be seen in image Figure 3.14 the linear interpolation method leads to problems with the upper bound of such an under-defined map. All the points that are not given by the map have been saved with a fuel mass rate of *'NaN'* so as to be easily identifiable and avoidable. But this becomes a problem when linear interpolation is applied because when an upper limit is considered, the interpolated point is going to be for sure outside of the area of the map that is known. This incorrect data is going to be a problem for the algorithm having an undefined weight (even though this would be resolved by some if-else condition that will be explained later) and in the definition of the maximum power capability of the power unit in the analyzed condition.

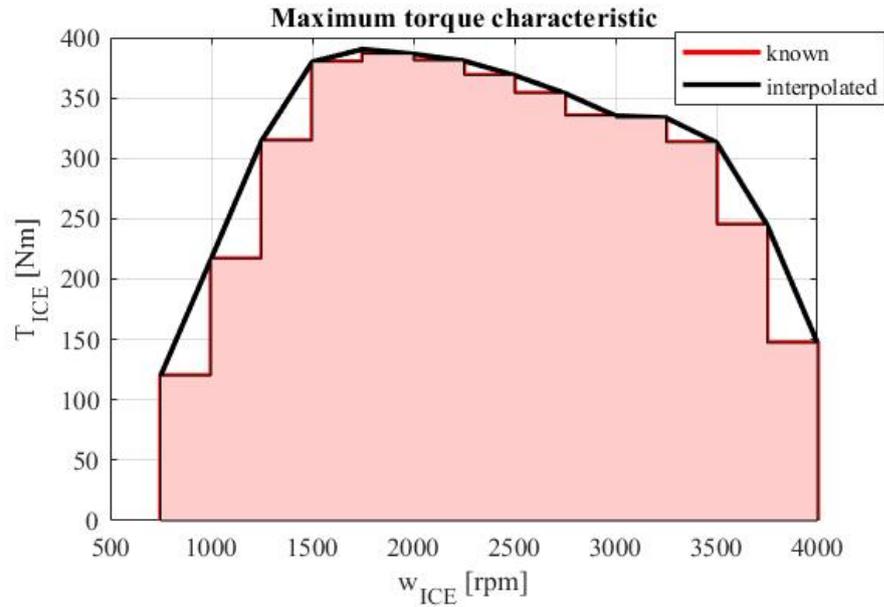


Figure 3.13: Area of the engine map where the points could be precisely extrapolated.

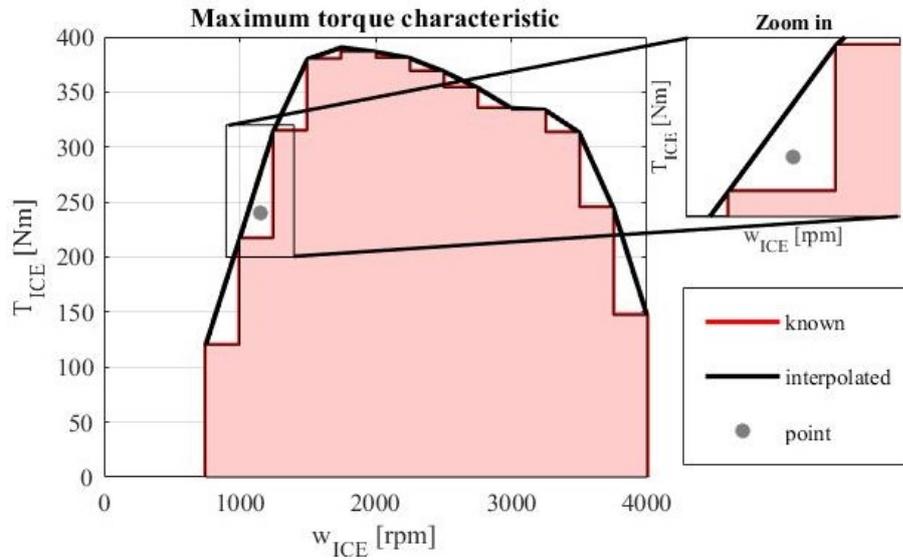


Figure 3.14: Example of point retrieved by interpolation that is out of the known zone of the engine's map.

This observation has been converted into an algorithm realized via Simulink, Figure 3.15, that takes as input the actual motor shaft velocity and finds the

bounds of the interval where this velocity is contained thanks to two Dynamic lookup table blocks that are set looking up to the upper and lower input. Then the output of the system is chosen to be the minimum among the two torque found, according to the known zone of the map.

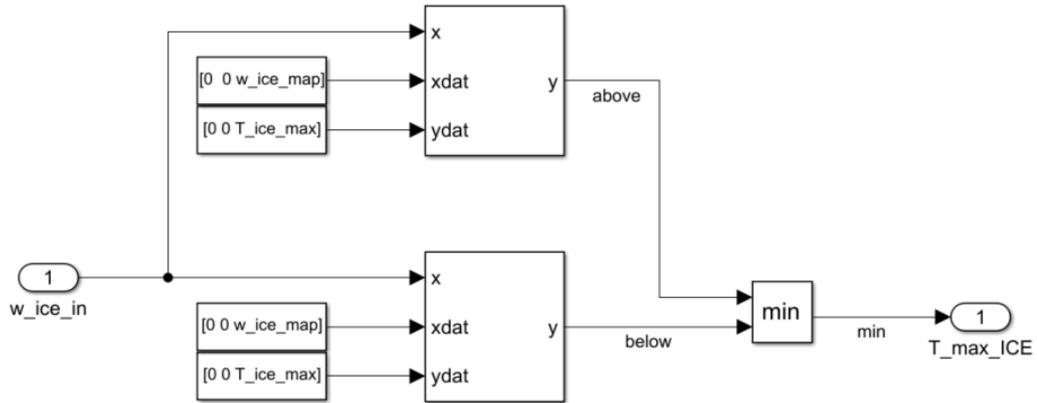


Figure 3.15: New look-up table system adopted for engine’s map interpolation.

One thing to point out is that the velocity that is given as input is the velocity that corresponds to the electric motor velocity with each gear, as a vector of eight elements. That is because with the output of this block one wants to retrieve the amount of torque that would be available from the combustion engine in the hypothetical situation when a selected gear is engaged and the clutch is closed, so when the engine shaft becomes synchronous with the electric motor shaft. With the vector of available torque for each gear, it is possible to accurately know the limits of the power unit while cycling among the gears.

Being this method of interpolation more effective with the given map, this kind of system has been employed also in the plant, in the motor model.

Moving to more inner levels, a lot of if-else conditions have been employed to identify and exclude out of the map points. As said those points have an associated fuel rate consumption that is not defined, is indicated as *NaN*, that in an equation would have a null weight. This could interfere with the right choice of the optimum since a point that doesn’t have any weight for the combustion engine component would be for sure lighter than any other.

those conditions are made to identify the infeasible condition and assign to them an enormous weight (all different to identify the error in the debug process) as to make them not interesting for the cost minimization process. In particular the

identified cases concern:

- The assigned torque to the combustion engine is greater than the available one;
- The fuel rate retrieved through interpolation from the map is *NaN* (out of the known zone) or is negative (interpolation error);
- There are errors in the interpolation of the efficiency of the electric machine.

3.2.6 New cost function

The new algorithm must be able to choose the best configuration among all the possible achievable also by implementing the gear change. The choice of the best torque split is certainly already well explained by considering only the weight of fuel equivalent given by the sum of the two thermal and electrical contributions as in the equation (3.9). The study has now focused on assessing whether two points that provide for a different gear should be evaluated in the same way or if it was right that the possible decision rate of the change was the same as the decision of the torque split.

A more in-depth analysis of the problem reveals that with the adoption of the cost function that takes into account only the split (3.9) the cost of the actuation is not considered and observing the response of the plant to this implementation, it is noted that the state variables are subject to transients that if not considered can lead the system to continuously change the working point leading to oscillations and system instability.

It is, therefore, necessary that the cost function is aware that the working point under analysis requires additional energy expenditure to be implemented or that being in a transient of the plant any improvement in analysis can be completely fictitious if applied.

The solution adopted involves inserting new factors into the cost function to ensure that a work point that provides for a cost of implementation is chosen if and only if the gain it brings is greater than its cost. Another weight is inserted to discourage a high frequency of shifting and limit the influence of transients.

$$\dot{m}_{gc} = k_{gc}|Gear_i - Gear_{actual}|w_{t,gc} \quad (3.15)$$

Both weights have been implemented through the formulation (3.15) that describes the new factor added to the cost function to take into account the gear shifting composed of a product of three elements:

- k_{gc} is a tunable fixed coefficient that identifies the cost of the actuation of a single gear shift, in this case, it has been chosen from a trial and error procedure making the contribution of this weight of the order of magnitude of the improvement that one evaluate enough to make the gear shift action worth;
- $|Gear_i - Gear_{actual}|$ represents the difference between the gear of the analyzed working condition and the actual gear engaged that in the new algorithm is given as input. its purpose is to make this additional weight proportional to the number of gear shifts that have to be actuated and cancel out it when the gear remains the same;
- $w_{t,gc}$ this element has been designed to prevent the occurrence of high-frequency gear shifts that are nor compatible with any mechanical hardware, neither optimal from a fuel consumption point of view since such behavior involves more transient events that are usually not efficient.

Among them, the one that deserves to be illustrated is the last. Its employment acts as a filter for the computed gear making the system more stable and reliable. It has been developed by weighting the time that passed from the last gear shift event, leaving the user the possibility to set the duration of this additional weight and its intensity.

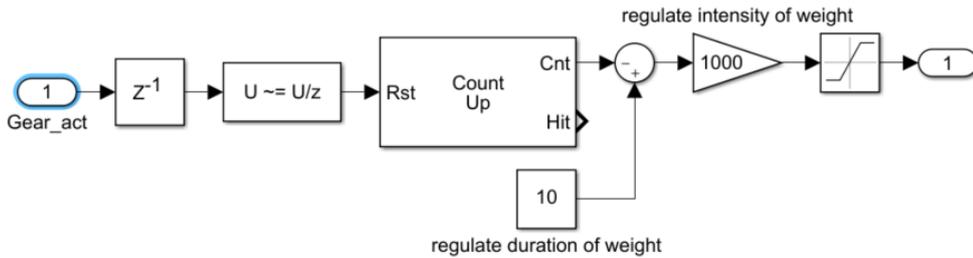


Figure 3.16: Block scheme of the time-dependent weight for gear change

It was realized in Simulink with the block scheme in Figure 3.16. The architecture involves:

- The actual gear signal that is given as input;

- A delay block to avoid the algebraic loop;
- A detect change block to create an impulse each time a gear shift event is sensed;
- A counter block reset by the gear shift event signal that starts counting up in free-running (+1 each time step);
- A constant that indicate the number of time-steps that the weight will last (in seconds $T_{w,sec} = T_{w,ts}/T_{ts}$);
- A gain to regulate the intensity of the weight;
- A saturation block that sets the lower limit to 1 such that once the set duration of the weight has elapsed, the weight becomes non-influential for the whole gear change component in the cost function without disabling it.

Two more weights have been inserted into the cost function to account for the situations that could be of interest when thinking about consumption reduction and improve the algorithm response in certain situations.

The first one has been designed to answer the need of the plant to reduce as much as possible the transients that affect the torque of the combustion engine, an argument that is deeply discussed in [25]. Those transients must be avoided for two main reasons:

- Discrepancy between the dynamic behavior of the combustion engine and the decision rate of the controller;
- Fuel surplus needed to overcome transient demand.

Since the dynamic behavior of the combustion engine is a lot slower than the decision rate of the algorithm, if not regularized, this could lead the plant to not being able to follow the requested command, ending up on the emanation of a torque smaller than the one requested by the driver. In fact, while the control behavior is a discrete function that doesn't provide for any limitation on the rate of change of the control signal, the response of the plant behaves like a first order low pass filter, originating situations where the discrepancy between what the plant is supposed to do and what it does is considerably high. As a consequence, if the plant is not able to guarantee the command, the control variable's error increases requiring more torque in the future and so increasing the fuel consumption.

Deepen analyzing the fuel consumption theme, it appears that the given map that links the fuel consumption to $[T_{ICE}, \omega_{ICE}]$ couples Figure 3.6 are collected applying statically the selected working point. In a real environment instead, the transient

that affects the motor involves a surplus of fuel consumed. With that being said, despite this behavior hasn't been modeled yet, it appears a good strategy to take it into account when developing an algorithm for a real set-up.

The formulation of this weight is :

$$\dot{m}_{dT_{ICE}} = k_{dT_{ICE}} |T_{ICE} - T_{ICE,prec}| \quad (3.16)$$

Where the factor $k_{dT_{ICE}}$ is :

$$\begin{cases} k_{dT_{ICE}} = constant & \text{if } T_{ICE} - T_{ICE,prec} \geq T_{ICE,max} \\ k_{dT_{ICE}} = 0 & \text{if } T_{ICE} - T_{ICE,prec} < T_{ICE,max} \end{cases} \quad (3.17)$$

The constant and the threshold $T_{ICE,max}$ can be tuned by analyzing the response of the plant by trial and error. The way this weight works is by adding a quantity to the cost function proportional (through the constant gain) to the difference in combustion engine torque from the one that the algorithm is analyzing to the one that was given in the previous step. The threshold $T_{ICE,max}$ has been adopted to enable this weight only for torque engine transient greater than the one that the plant is actually capable of.

The second weight that has been inserted in the cost function wants to relax the constrain on the requested torque gaining the possibility of accepting supplying a minor quantity of torque if really convenient.

The need of relaxing the requirement arises from the observation of certain situations when, despite being at high velocity, the required torque was as high as to need a lower gear to be supplied. On the other hand, engaging a lower gear implies the increase of the motor velocity and its fuel consumption which is not considered worthwhile when the requested torque is just a bit higher than the available one with the gear already engaged.

Adding a weight that takes into account this makes the controller being able to decide whether it worth or not to spend that quantity of fuel more. The formulation of such weight is :

$$\dot{m}_{dT_{req}} = k_{dT_{req}} |T_{given} - T_{req}| + K_{dT_{req}} \quad (3.18)$$

Where the factor $k_{dT_{req}}$ is :

$$\begin{cases} K_{dT_{req}} = constant & \text{if } T_{req} - T_{given} \geq dT_{max} \\ K_{dT_{req}} = 0 & \text{if } T_{req} - T_{given} < dT_{max} \end{cases} \quad (3.19)$$

The constants $k_{dT_{req}}$, $K_{dT_{req}}$ and the threshold dT_{max} can be tuned analyzing the response of the plant by trial and error.

This formulation is composed of a component proportional (through $k_{dT_{req}}$) to the discrepancy between the given and the requested torque and a constant component

that acts as a flag that highlight situation where this difference is too high (more than the threshold dT_{max}) and gives additional weight to avoid them.

Once all the new weights have been inserted, the original formulation is a complete and adjustable cost function that manages to choose the best working point taking into account all the factors and situations that have been explained in this paragraph. The regulation of factors largely influences decisions, which can make gear changes more or less frequent, reduce the transients of the thermal engine, or even be more compliant with the torque demand. The complete formulation of the aforementioned cost function is:

$$J_{traction} = \dot{m}_{ICE} + \dot{m}_{EM} + \dot{m}_{gc} + \dot{m}_{dT_{req}} + \dot{m}_{dT_{ICE}} \quad (3.20)$$

As said, the reported cost function is used only after the determination that the case under analysis is a request for traction of the driver. Otherwise, if it is a request for braking, everything is simplified by having to analyze only the torque delivery from the electric motor. In this case, the component concerning the fuel of the engine and the weight concerning the thermal torque transients lapses. Furthermore, considering that the difference between the equivalent fuel between the various working points of the electric machine is much smaller than in the case in which the heat engine is also used, it is advisable to calibrate the weights so that they are lighter and have an influence compatible with the possible gaining.

3.2.7 Battery energy behavior management

The last element whose change has yet to be analyzed is the multiplicative factor within the fuel equivalent of the electric machine which depends on the SOC, named $f(\text{SOC})$.

As mentioned in the subsection 3.1.4, the formulation of this component depends explicitly on the type of architecture in which it is to be used. As for this study, solutions have been developed for plug-in and non-plug-in configurations.

As for the case of a plug-in vehicle, the goal of the $f(\text{SOC})$ is to maintain the charge by swinging it around a set level so as not to arrive at excessively low SOC levels since the charge cannot be restored from any external source. In particular, the legislation preserves this aspect by imposing the constraint that the initial and final charge of a WLTC must be approximately equal.

To comply with these constraints and meet the needs of this configuration, the solution that has been adopted in Figure 3.17 is the implementation of a PI controller that is based on the error between the current SOC and the reference level set at 50%. For this controller, an integral part was chosen that is not too influential in order to limit but not cancel the oscillatory effect that this component brings,

an output offset was given to match the null error to the value of $f(\text{SOC})$ which corresponds to the charge sustaining of the algorithm and a proportional part calibrated to allow simultaneously a good detachment from the reference and the maintenance of the charge.

Being the process of choice of all the different coefficients extremely arbitrary and being difficult to try all the possible combinations of them, the values adopted are not guaranteed to be the best choice, but are good enough to achieve the desired results.

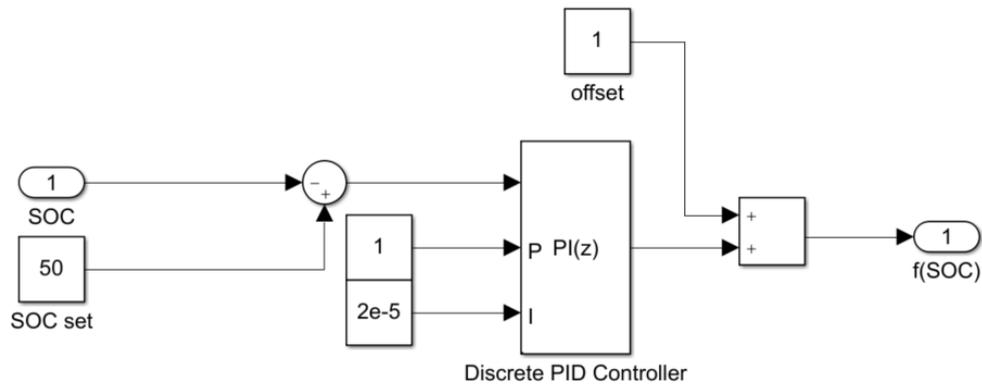


Figure 3.17: PI controller for $f(\text{SOC})$ production

Analyzing instead the case of the plug-in vehicle, as mentioned, the needs are different. The behavior of the battery must be such as to ensure the first phase of discharge and the second phase of maintenance of the battery. The solution adopted and discussed in subsection 3.1.4, although it was quite functional, presented some critical issues that were solved considering a new alternative.

The critical issues concern:

- The behavior assumed in charge maintaining ;
- Very low soc behavior;
- The possibility of assuming only two values discreetly.

The first point refers to the trend that the SOC assumes in charge maintaining, where, due to the switch between the two values of $f(\text{SOC})$ of which one favors the discharge of the battery and the other the charge, the SOC value quickly discharges up to the lower limit and after charging slowly up to the upper limit, creating a saw-tooth trend observable in Figure 3.18.

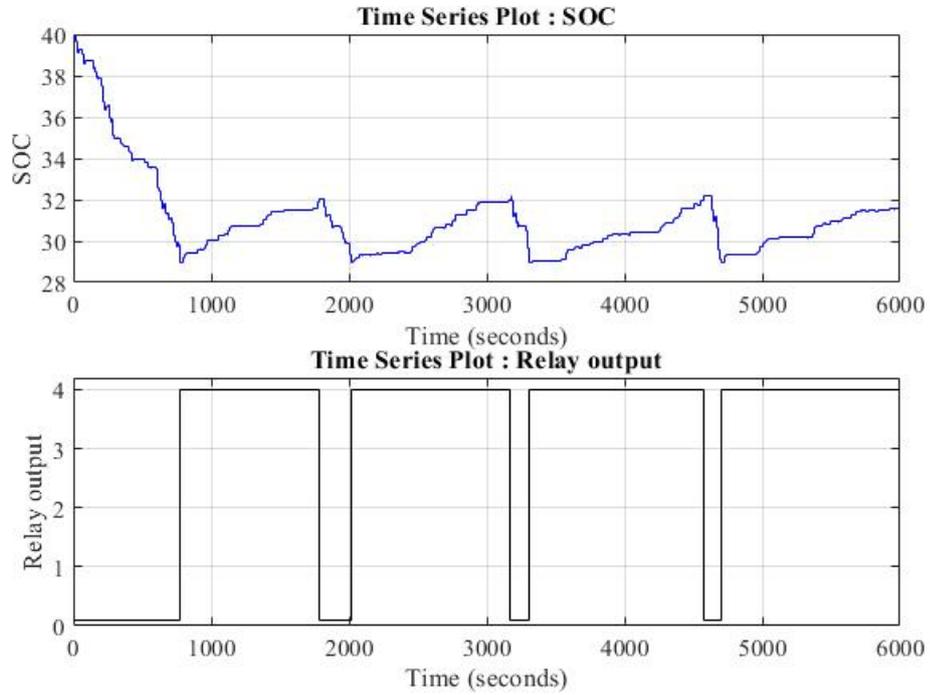


Figure 3.18: Saw-tooth behavior assumed by the SOC in charge maintaining with the relay $f(SOC)$.

Such a trend does not seem to be ideal because the choice of the working point will be influenced by the historical trend of the SOC and there will be no punctual optimization. We want to understand that, if for example the SOC is at an intermediate value at the two extremes set in the relay block, the weight that will be given to this situation and consequently the working point adopted will depend on the state previously assumed by the relay block, if on or off mode.

The second critical issue highlighted concerns the fact that once the SOC falls below the lower trigger value, the value of the $f(SOC)$ always remains the same, whatever SOC value is reached from there on. The value chosen is such as to guarantee a non-aggressive charge of the battery, taking advantage of regenerative braking and rarely using the electric motor to use torque splitting. The critical aspect that has been focused on is that there may be situations for which a more aggressive charge may be more suitable, for example when the battery level reaches critical discharge levels.

The last point whose problematicness is also reflected in the two points described above also involves the impossibility of adjusting the intensity of the torque split

depending on the current value of the SOC. Having only two discrete values it is possible to have only two types of behaviors, not adopting all intermediate situations.

To cope with these observations, a new polynomial $f(\text{SOC})$, Figure 3.19, was produced to smoothly transition from one control condition to another. The function was produced through the "poly fit" command, which returns the polynomial of the desired degree that best interpolates a provided point vector.

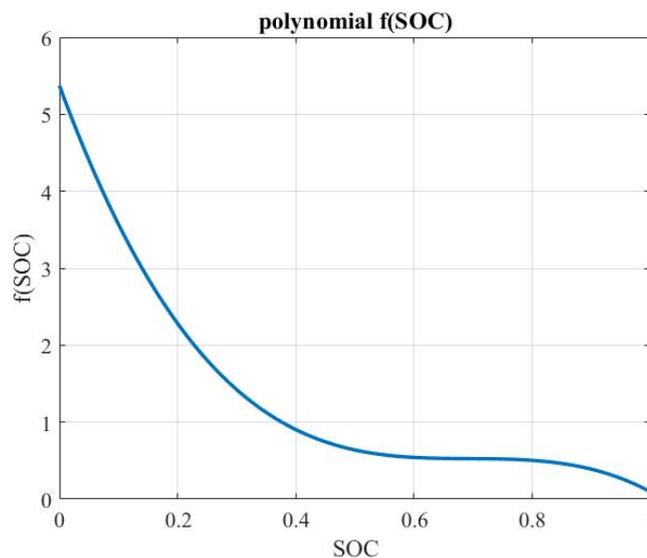


Figure 3.19: New polynomial $f(\text{SOC})$.

The points have been chosen to set values in considerable conditions of importance, giving a low value to the values that correspond to a high SOC, encouraging the use of electric traction, a value such to choose the power split for values close to the charge maintaining value of the SOC and gradually higher values going below. Having a linear-like continuous function around the value for which you want to maintain the state of charge, the behavior of the battery energy obtained (Figure 3.20) is similar to that of a proportional controller according to a constant that corresponds to the slope of that linear part. The slope was chosen to be steep enough not to allow an excessive deviation from the set value of the SOC. This specification has always been implemented in relation to the legislation that specifies that the use of the battery must be absolutely limited at this stage.

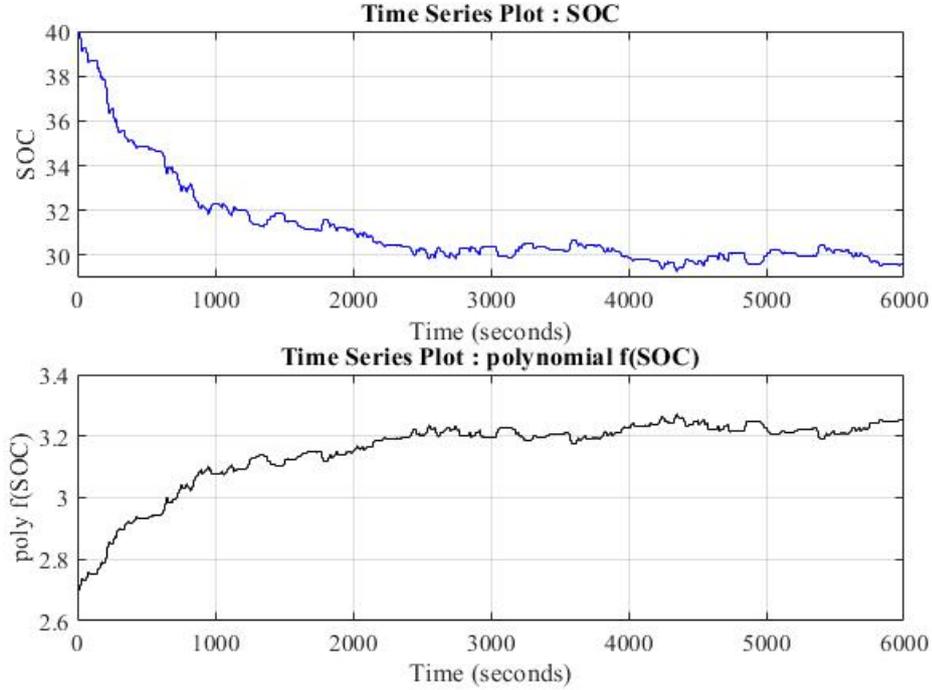


Figure 3.20: Behavior assumed by the SOC in charge maintaining with the new polynomial $f(\text{SOC})$.

A feature of the implementation of the new $f(\text{SOC})$, analyzing the situation in which the state of charge of the battery is critically low, is that it will be given in output a value so high as to favor the charging of the battery through load shifting, using the heat engine to charge the battery faster and first solve the critical state that could damage the battery itself.

The implementation of this function can take place through the Simulink block that executes the "polyval" command, which receives as input the value of SOC and calculates the value of the polynomial whose coefficients are given and calculated, as mentioned, through the "polyfit" command.

3.3 Results

In this section will be presented the results that have been obtained considering a non-plug-in P2 hybrid configuration setup running a WLTP homologation cycle. The results will be presented as a comparison of the new ECMS-GC, which formulation in code is available in section B.1, with the previous ECMS formulation in section 3.1 from a performance and emission reduction point of view.

3.3.1 Emission reduction

The reduction of pollutant emissions is the main purpose of this algorithm and has been evaluated considering three parameters:

- $CO_2/100km$ emitted;
- Fuel required by the heat engine;
- Fuel equivalent that also takes into account the consumption of electricity.

As seen in subsection 2.4.1 while the fuel required by the heat engine is derived from interpolation of engine maps with respect to a given torque request at a certain engine rpm condition, $CO_2/100km$ is calculated considering a fixed gain that transforms a fuel consumption into a CO2 emission, and the equivalent fuel is calculated as extensively described in subsection 3.1.4.

This last term is actually not far from the actual fuel consumed because, being under consideration a non-plug-in configuration, it is recalled that a requirement for carrying out the WLTP cycle is that the initial and final SOC must be the same, so the delta electricity during the cycle is negligible.

	ECMS	ECMS-GC	Reduction	improvement
CO_2 [kg/100km]	0.2741	0.2721	0.002	0.73%
Fuel [l]	1.77	1.76	0.01	0.6%
Equivalent fuel [l]	1.7	1.762	0.008	0.5%

Table 3.1: Different ECMS logics fuel consumption and emission comparison.

As evident in Table 3.1, the algorithm can bring a significant reduction in emissions managing to save 2.29% of the emissions of a basic ECMS which corresponds to about $5.9 gCO_2/100km$. This reduction in emissions is directly related to fuel savings of about the same amount.

The best behavior in terms of efficiency is also evident by observing which points of the motor maps are used.

From the comparison between the maps of the thermal engine, it can be seen that when the ECMS-GC algorithm is adopted there is a shift in the fashion of the points used towards the lower power section of the map, where there is the lowest fuel consumption, and there is a better exploitation of all the possible areas of the

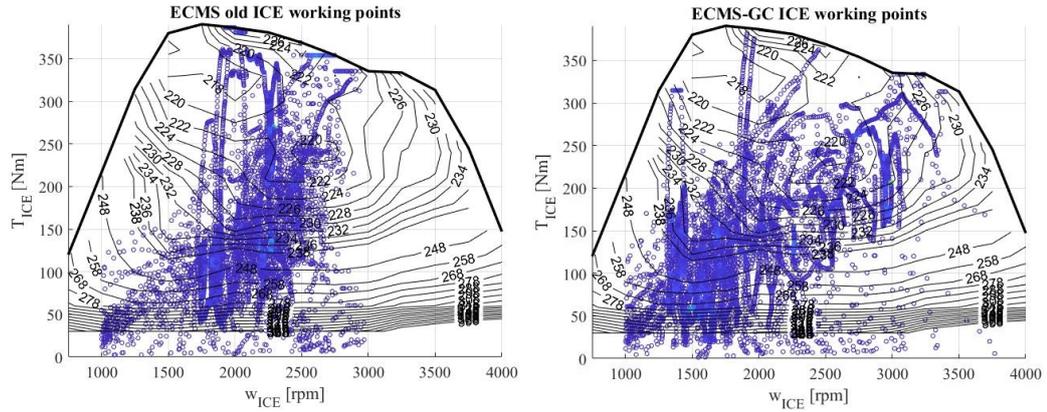


Figure 3.21: ICE engine map working points comparison between standard ECMS and ECMS-GC.

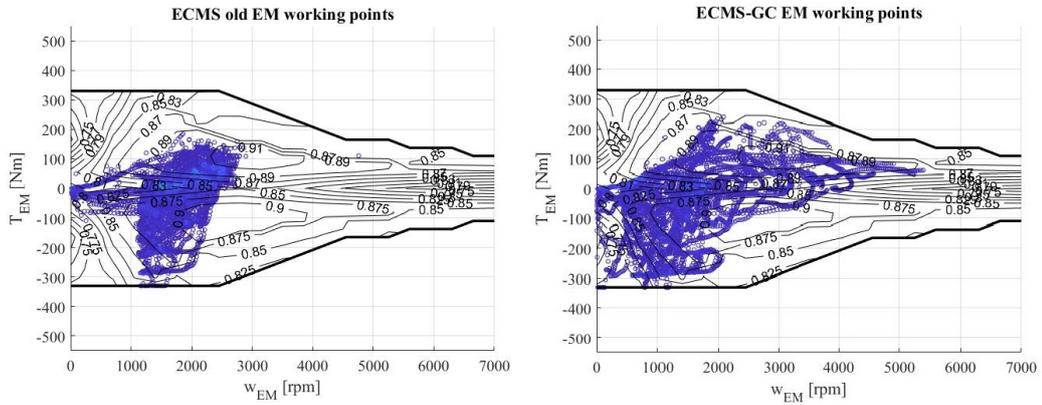


Figure 3.22: EM engine map working points comparison between standard ECMS and ECMS-GC.

map having the most distributed points and not limited to the angular speed of change.

As far as the map of the electric motor is concerned, it is even more evident that the absence of the lower and upper limits of the gearbox leads to better exploitation of the possible working points. In particular, with the electric machine, it is possible to reach very high angular speeds while maintaining high levels of efficiency or having torque available from zero speed; these features are better managed and exploited with the new algorithm and reducing gear changes and taking advantage of the lower speeds of the idle speed of the heat engine if the traction is fully electric.

3.3.2 Performances

Focusing now on the performance of the vehicle simulated with the new algorithm we want to compare three particular aspects:

- The progress of the gears engaged
- An index that characterizes the powertrain's response to the driver's command ;
- An index that characterizes the response of the combustion engine to the controller control.

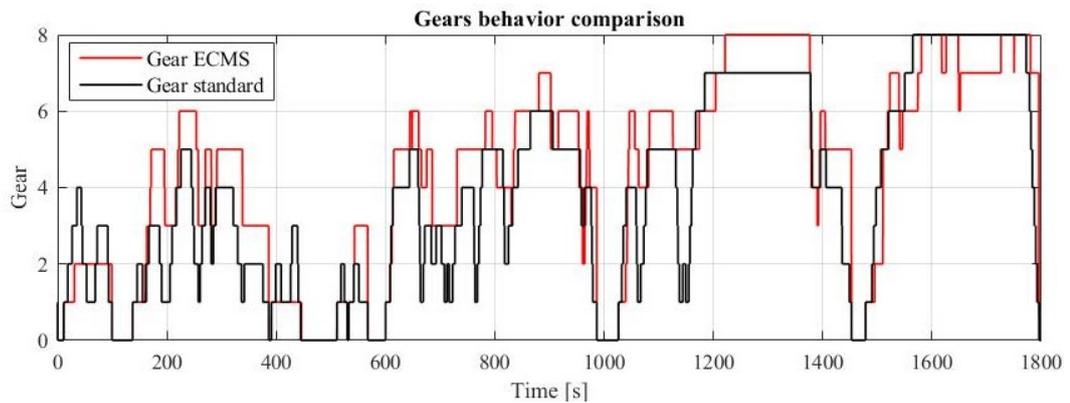


Figure 3.23: ECMS-GC and standard logic Gear behavior time series comparison.

The first aspect, shown in Figure 3.23, shows how the gear shift algorithm behaves very differently from the standard algorithm. In the first section, it can be noted that in the presence of only electric traction the algorithm tends to maintain a fixed gear considering the shift not favorable enough to be completed. In general, the trend of the gears is very similar to the basic one when the torque split is in force, although usually keeping above to maintain a lower number of revolutions and consume less. It is also noted that when braking it tends not to change gears so as not to have torque interruptions due to the operation of the clutch during the shift. In the last section, it can also be noted that the new algorithm tends to better meet the driver's requests where more torque is required at high speeds using a lower gear.

	ECMS	ECMS-GC	Improvement
dT_{ICE} index	5539	2375	57%
dT_{req} index	11800	1889	84%

Table 3.2: ECMS algorithms Performance index comparison.

The two indices, in Table 3.2, are indicators of how much the torque command deviates from the quantity delivered. They were calculated as the integral of the difference between the control and the measured delivery.

$$\int T_{command} - T_{measured} \quad (3.21)$$

From the results it is clear that the new algorithm has better results on both fronts, being able to better follow the torque requests of the driver and taking into account the dynamics of the combustion engine.

Chapter 4

Intelligent Start&Stop

The Start&Stop system is one of the main players in reducing vehicle emissions in an intuitive and economically sustainable way.

The system eliminates idle fuel consumption when the vehicle is stationary by turning off the combustion engine and turning it back on almost instantaneously at the driver's request for movement [26] [27].

The start&stop is, therefore, able to optimize the consumption of the vehicle during stops avoiding that the engine remains running in idle condition and is particularly effective, for example, in city contexts where the need for intersections managed by traffic lights, rows and stops is particularly frequent and results in significant reductions in fuel consumption and pollutant emissions [28].

The systems currently widely used are static and decide the implementation of the stop according to variables such as the speed of the vehicle, the torque demand and in some cases the passage of a fixed threshold time. These strategies can cause the occurrence of start&stop events so short that they are not efficient both from the point of view of consumption [29] and driving comfort.

Several studies focus on improving the performance of this type of system, some focusing on controlling the hardware protagonists of the actuation to improve its dynamics [30] [31], others exploiting the outside information to avoid inefficient start&stop events [32].

This study focuses on the development of start&stop logics to maximize their duration by exploiting decision-making algorithms based on cost functions, in this case, ECMS, to delay the Start and speed profile predictions obtained through ADAS to anticipate the Stop.

4.1 State of the art

4.1.1 Previous algorithm description

The already implemented algorithm had a pretty standard formulation. It was implemented through a Stateflow flow chart MISO system that takes as input the vehicle velocity and the requested torque signal and produces as output a boolean signal that identifies the ICE stop status, as displayed in Figure 4.1.

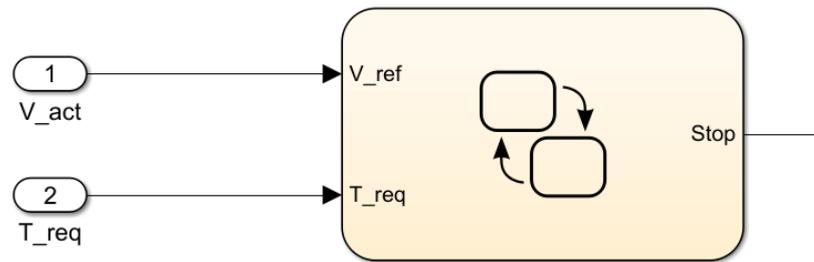


Figure 4.1: Standard ICE Start&Stop block scheme.

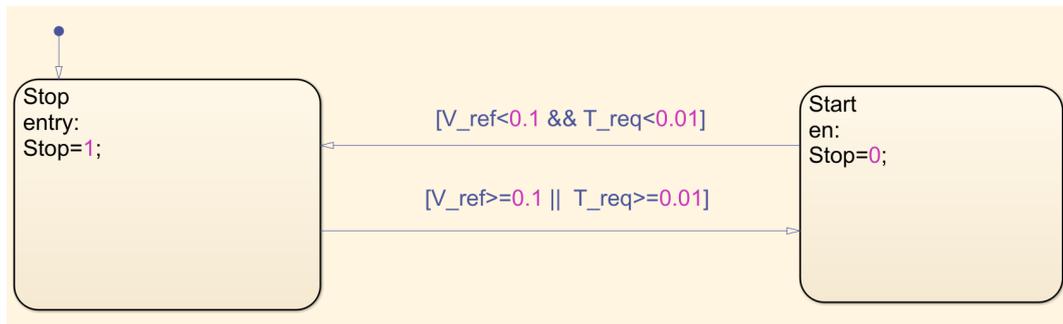


Figure 4.2: Standard ICE Start&Stop logic.

A Stateflow flow chart is a very intuitive graphical construct that models logical patterns. The blocks in Figure 4.2 represent all the different states that the output could assume, in this case are only two, engine on or engine off, while the arrows that connect them model the state transition that is associated with a specific condition. The algorithm employed with the model in Figure 4.2 follows this logic:

- The initial condition is set as state "Stop = 1" that is associated with the engine off;
- If the measured velocity of the vehicle gets closer to zero (not exactly to be

compliant with the noises that affect this variable) and the driver doesn't request any torque, the engine turns off, setting " $Stop = 0$ ";

- The engine starts again when at least one of the two conditions is not respected.

4.1.2 Plant actuation

Once the boolean $Stop$ signal was created, it has been sent through the model in different ways for the actuation.

On the controller side, the signal was sent in the block that shapes the throttle signal so that to be in logic and relation with him.

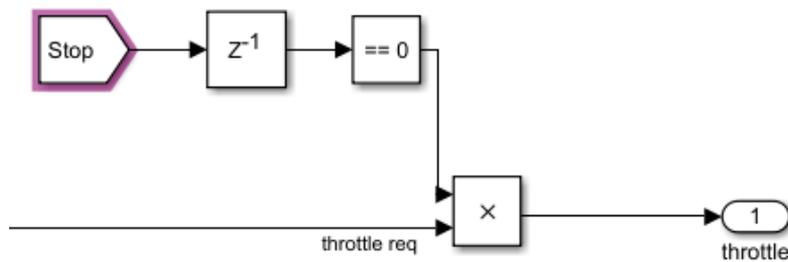


Figure 4.3: Throttle disabling with Stop signal

As can be seen in Figure 4.3 the throttle signal is multiplied to the comparison between the delayed (to avoid algebraic loop) signal $Stop$ and 0 so that when the motor is turned off the throttle signal gets deleted.

On the Plant side, the signal is used to:

- Enable the resisting torque of the combustion engine when turned off as displayed in Figure 4.4.
- Disable the fuel consumption when the engine is off (cut off) as shown in Figure 4.5.
- Add a fuel quantity needed for turning on the motor, that is given by the engine datasheet and identified by the falling edge of the $Stop$ signal, block scheme in Figure 4.6

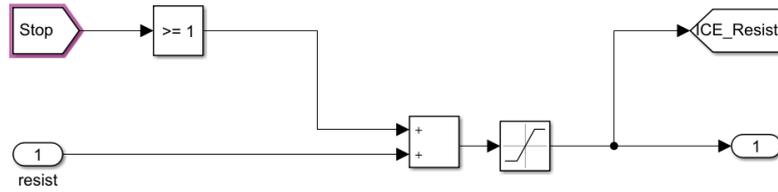


Figure 4.4: Resisting torque signal generation.

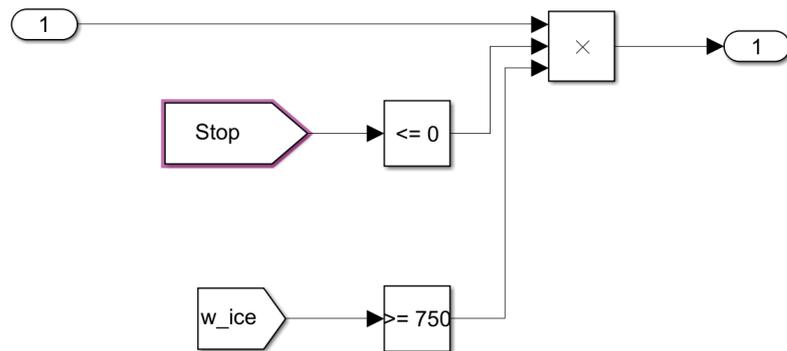


Figure 4.5: Simulink block scheme for fuel cut off when the engine is off.

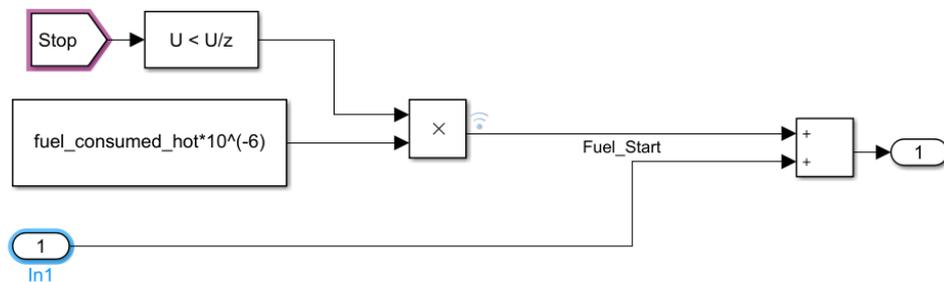


Figure 4.6: Simulink block scheme for fuel addition when engine starting.

4.2 Solution produced

The objective of the new algorithm is to enhance the stop time of the engine and take advantage of the possibility of using the electric motor as the only source of power, making it possible to shut down the engine on the move.

The objective has been pursued acting in two ways:

- Set forward the engine restart as far as the engine power is not needed.
- Trying to move up the engine's stop through the prediction of the velocity profile available through the ADAS signal analysis;

4.2.1 Restart delay

The idea that lead to the development of this aspect is that, since the P2 hybrid configuration in analysis has an electric motor that can supply enough power to be the only source of the vehicle, the conditions discussed in section 4.1 are not enough to imply the need for the engine to be turned on. In fact, considering that the departures involves frequently bad working point for the engine and that they are often assigned to the electric machine, one can think that as far as the engine is not involved in the traction, it can be maintained off, saving the idle fuel that would be consumed in the process.

Moreover, when the battery is charged, the ECMS typically uses only the electric motor until the SOC reference value is reached. In such a situation the engine is not needed at all and keeping it off could gain an important amount of fuel and emissions savings. This last aspect is particularly evident in plug-in configurations where, during the charge depleting phase, with the new algorithm the engine would rest off for the whole cycle, while with the old one the idle consumes are present.

The logic, represented in Figure 4.7, has been realized by exploiting the decision cost function of the ECMS, integrating into the algorithm the decision of turning on the engine weighting it through an additional cost.

The engine stop starts once the basic engine start and stop logic described in section 4.1 commands it, that disables the engine use in the ECMS algorithm. Once this signal ends the condition to restart the motor becomes the presence of a torque request from the ECMS to the combustion engine. The ECMS decides whether it is convenient or not to turn on the engine considering that the configurations that provide for the engine usage will need an expense in terms of fuel consumption for the engine start.

This aspect is taken into account adding a new weight to the ECMS cost function for the maintaining of the stop condition, defined as follow:

$$\begin{cases} m_{restart} = Stop * m_{restart} & \text{if engine involved} \\ m_{restart} = 0 & \text{if only electric motor} \end{cases} \quad (4.1)$$

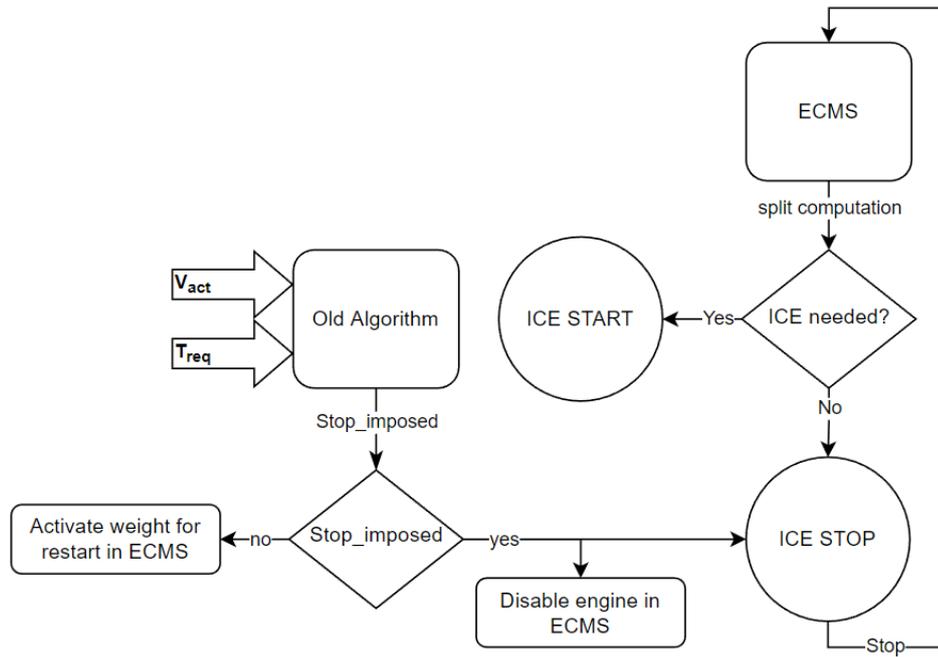


Figure 4.7: Logic scheme for i-Start&Stop restart delay.

Where $m_{restart}$ has been chosen to be the Fuel addition for the engine starting in Figure 4.6 but could be modified making the restart of the engine earlier or later and $Stop$ is the status of the combustion engine, so as to disable this weight once the engine is already on.

4.2.2 Advance shutdown

This section will describe the developed logic that takes advantage of the preview of the velocity profile in the future to decide to anticipate the engine turning off, increasing the stop time and reducing the fuel consumption.

It has been developed in two different ways, the first one analyzes the actual prevision of what the velocity of the vehicle would be in the future, while the second one triggers the system whenever a stop is pictured in the future.

The first logic aims at founding periods in which the combustion engine would certainly not be used that have to be long enough to let the turning off and on be convenient. In order to do this the algorithm needs a prevision of the velocity profile in the future that could be given working on the ADAS signals that could be used in different ways:

- Store the values of velocity in time of a lead vehicle that is followed through an Adaptive Cruise Control thanks to the sensors such as radar and stereo camera;
- Predict the values of velocity in the future through an MPC controller involved in the ACC algorithm.

Both solutions supply a vector of velocity in time that could be analyzed to retrieve data that leads to the identification of the scenario in which the engine could be turned off.

The question was how to discern those scenarios. The approach adopted focused on the fact that in the situation on analysis, the engine is not used in two situations, when braking and during stops. In particular, braking, if intended to be a deceleration, is not enough since there are situations that involve a certain quantity of traction force despite being decelerating. This is due to the resisting forces that the vehicle has to overcome when moving that produce a certain deceleration, they depend on the velocity and are described by the coast-down parameters.

The coast-down parameters are experimentally retrieved coefficients that give a good knowledge of the resisting forces that affect the vehicle to be moved. An accurate overview is given in [33].

To match as much as possible the calculation of the algorithm to the behavior of the simulated vehicle, a coast-down experiment has been performed in the simulated environment. So the vehicle was led to 100 Km/h and then a step signal disengaged the power line by opening the clutch and cutting the electrical power of the motor, as represented in Figure 4.8. After that a data processing on the results was made, looking for the equation that describes the behavior of the coast down deceleration in function of the velocity of the vehicle. This relation is expected to be of the same order as the forces involved, so the results were retrieved through second-order polynomial interpolation, Figure 4.9.

Once this relation was computed one can precisely know what would be the deceleration of the vehicle at any velocity if no traction force is applied. This deceleration could be interpreted as a threshold, since, if a greater deceleration is needed, then it must be applied a braking force, while in contrast if a smaller deceleration is needed, traction power must be supplied.

This factor will be the discriminant that let the algorithm understand whether the velocity profile of the future could involve the usage of the engine or not. The algorithm computes the accelerations expected and counts the seconds that the period when the accelerations are below the threshold least and compare it with a predefined time computed as follows:

$$T_{set} = \frac{m_{fuelhotrestart}}{m_{idle}} \quad (4.2)$$

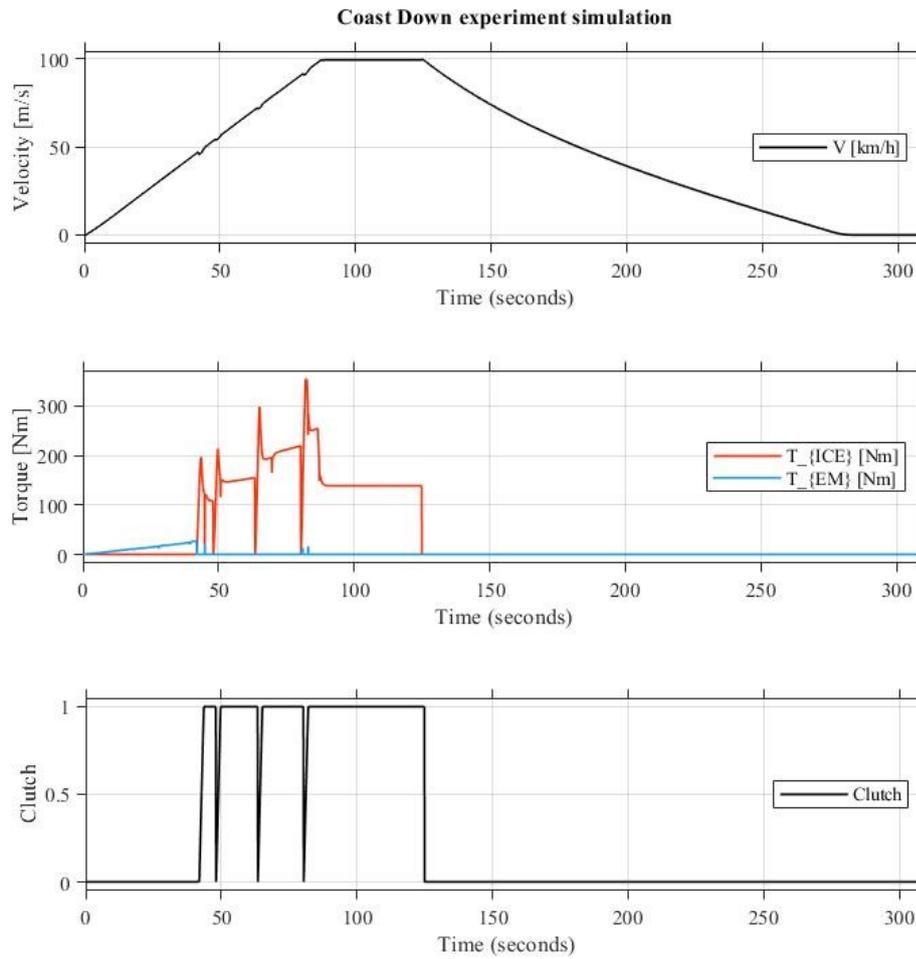


Figure 4.8: Coast down experiment time series.

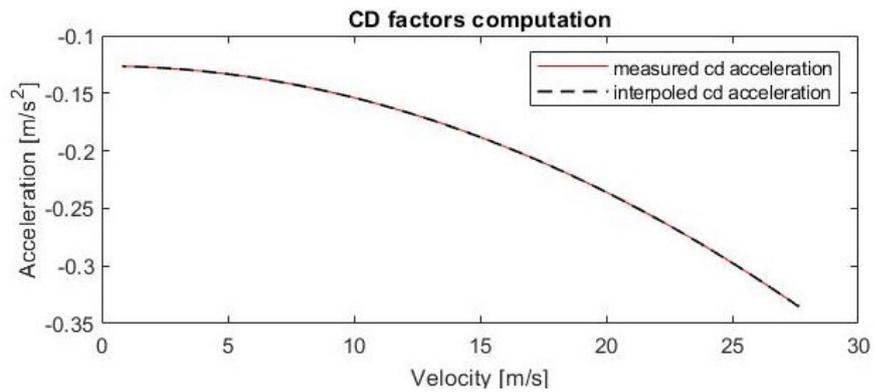


Figure 4.9: Coast down factors interpolation.

Where $m_{fuelhotrestart}$ is the fuel mass [kg] that is spent for starting up the engine and m_{idle} is the fuel mass rate that is requested in idle condition [kg/s]. This computed time represents the minimum time for which it is worth to turn off and on the engine rather than maintain it in idle condition.

Once this condition is respected and the engine is turned off, the algorithm doesn't need anymore to consider the whole prediction time, but it has only to decide to maintain or not the stop status of the motor, so, only the actual velocity is compared with the coast down acceleration.

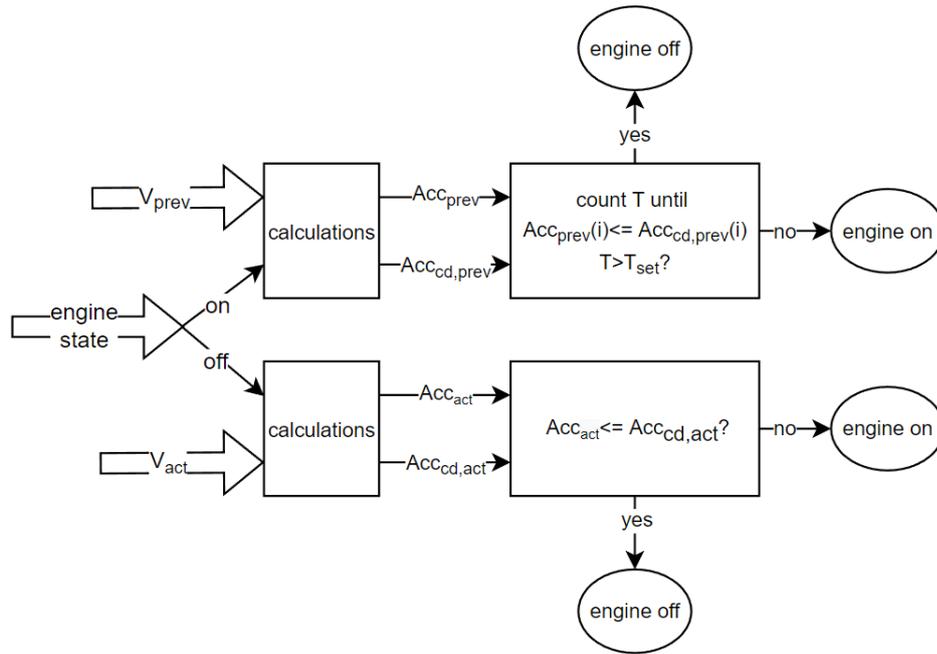


Figure 4.10: Logic scheme of stop prevision with velocity profile analysis.

Getting a value of T_{set} close to 8 seconds this strategy requires a considerable preview of the future that is not realistically achievable in a scenario where the velocity profile is predicted through an ACC. However, it is appropriate to think of obtaining a forecast of this magnitude in the case of a stop, considering that a stop sign, a traffic light or a stationary vehicle can be identified well in advance.

For this reason, the development was addressed to the second way of thinking, based on the perception of a stop instead of a long deceleration as before. The stop could be detected in many different ways, taking advantage of the ADAS sensors employed on the vehicle and machine learning algorithms for image recognition, a good overview of that is described in [34].

If a situation of arrest is detected, one may think that with the hardware configuration adopted, since the trend of the acceleration would probably be negative and in this situation only the electric motor is needed for regenerative braking, the combustion engine could be turned off.

Moreover, the employment of a stop velocity profiling, which will be explained in ??, could make sure to have only deceleration after a stop detection.

For those reasons the solution adopted triggers the stop whenever a stop is detected in the future, increasing by a lot the stop time of the engine without losing drivability.

As for the simulation, it was assumed to have a stop detection system that gave a Boolean signal *stop* as output. It was created by providing a Simulink block that identifies stops with a speed profile anticipated by an amount of time deemed appropriate for real applications, in this case, twenty seconds.

The anticipated profile is therefore analyzed to find the sections where the speed is zero and in this case, the *stop == 1* signal is issued, which identifies a stop in the future and which is maintained until the non-delayed reference speed profile also reaches the expected stop.

This stop sign is then used in the Start&Stop algorithm as a "stop imposed" signal in Figure 4.7 to take advantage of both the advance and the delay of the shutdown of the combustion engine.

The two algorithms converted into code are presented in section B.2

4.3 Results

The results were obtained by simulating the non-plug-in p2 hybrid vehicle performing a WLTP cycle.

The simulation setup involves the reference given by a velocity profile and the velocity feedback throttle control. For the last simulation, ADAS signals were simulated as an ideal source.

Table 4.1 shows the numerical result obtained in terms of reducing emissions and increasing stop time.

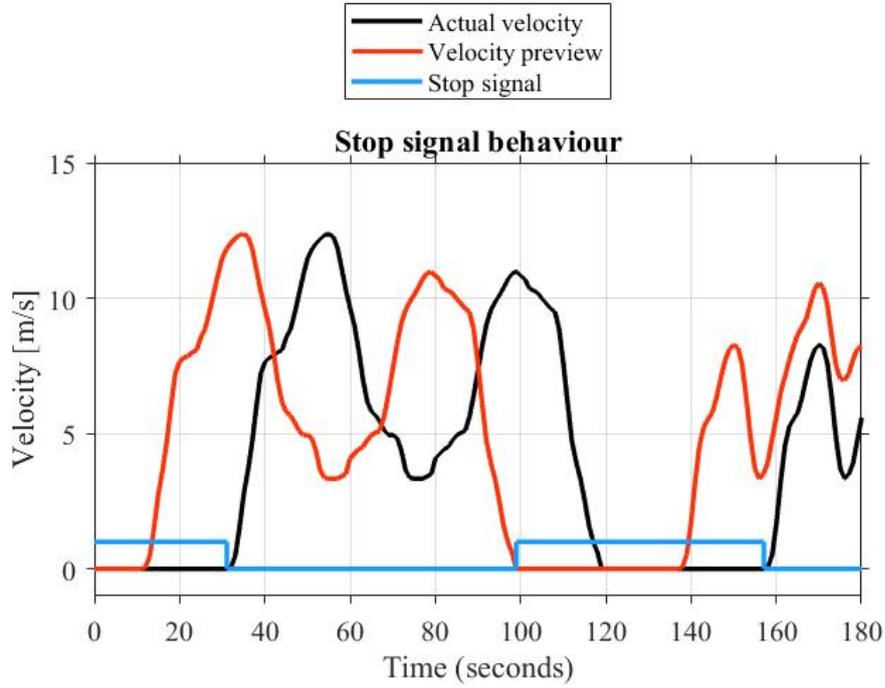


Figure 4.11: Adas-retrieved stop signal behavior with respect to the velocity preview and the actual one.

	Standard	ECMS based	ADAS based
CO_2 [kg/100km]	0.2721	0.2683	0.2663
Stop time	224.3	500.8	602.1

Table 4.1: Start&Stop algorithms performance comparison.

It can be observed how an increase in stop time involves a reduction in fuel consumption and consequently in pollutant emission. Comparing the results can be retrieved that adopting the ECMS-based Start&Stop strategy the stop time increases by the 123.27% with respect to the standard formulation causing a reduction in CO_2 emission of 1.4%. Focusing on the result gained through the ADAS-based strategy, the stop time reached a 168,44% improvement with respect to the standard formulation and 20.23% with respect to the ECMS-based strategy. This benefit caused a 2.13% emission reduction with respect to the standard formulation, which is a 0.75% reduction with respect to

the ECMS-based strategy.

In Figure 4.13 and Figure 4.12 are highlighted the points of the speed profile where the Stop is effective for the different start&stop algorithms.

In red one can see the behavior of the standard start and stop, which stops the combustion engine only when the vehicle is standing still and there's no torque request. This kind of logic can involve stops whose duration is under the time threshold treated in Equation 4.2, so that are not convenient being the effort to turn on and off the engine is greater than the one saved during the shutdown.

In yellow is represented the part of the velocity profile where the engine is turned off adopting the ECMS-based Start&Stop. One can notice how the engine turning on is delayed thanks to the employment of the new logic, causing the elimination of short engine stops and the stretching of every single stop in the cycle. Another thing to point out is the maintenance of the engine stop when the electric power train can handle the request, like in the first 150 seconds.

Finally in green is reported the behavior of the ADAS-based algorithm which involves also the anticipation of the engine's stop. This is evident when the stop is already effective during the breaking parts of the cycle. This feature contributes to avoiding short stops and increases stop time.

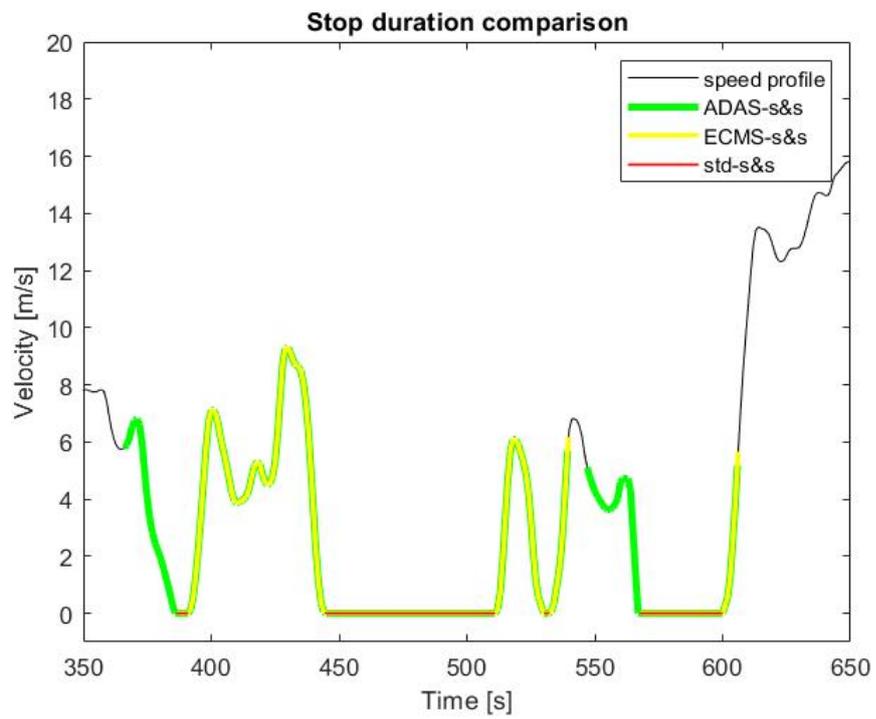


Figure 4.12: Particular of different Start&Stop algorithms short stop handling.

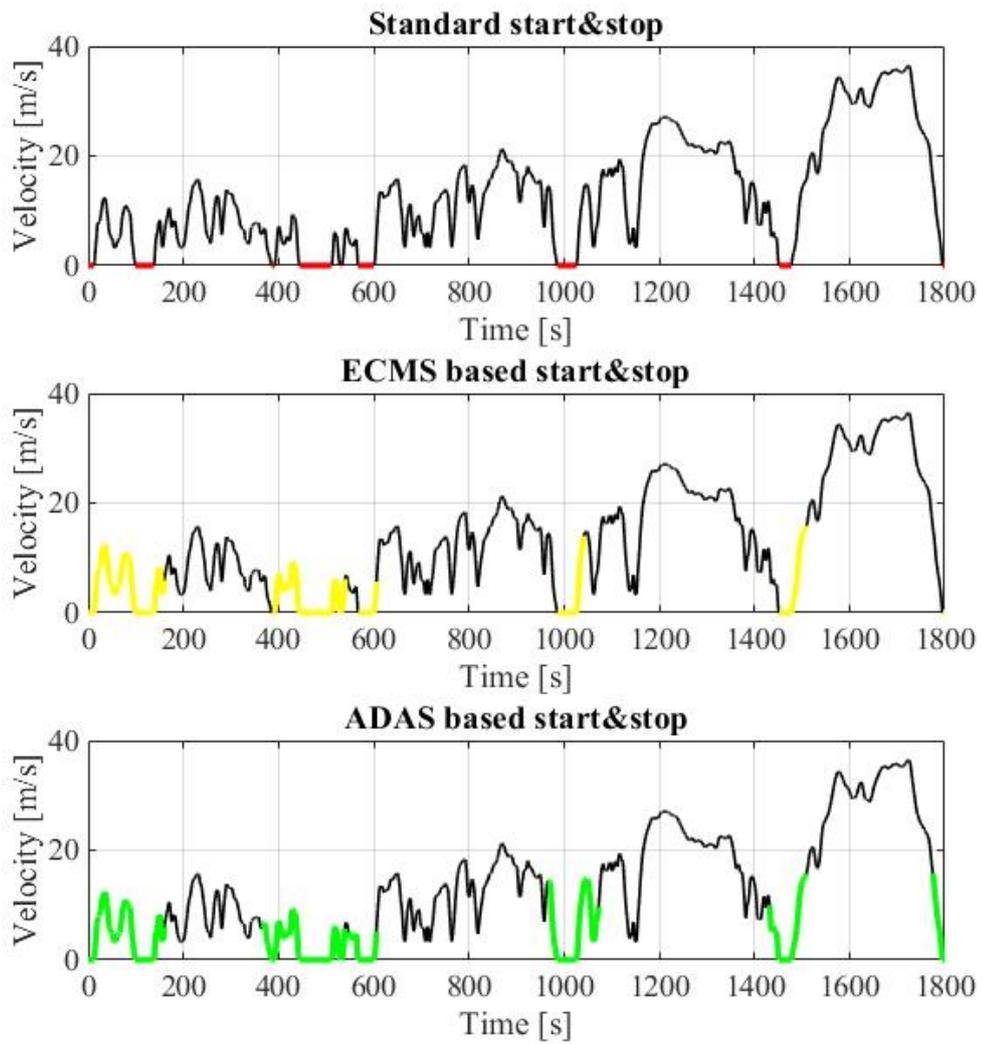


Figure 4.13: Engine stop behavior comparison for different Start&stop architectures.

Chapter 5

Conclusions

This thesis was aimed at reducing the consumption of a hybrid electric vehicle in P2 configuration by developing control logic that optimized the vehicle's behavior while saving fuel, designed based on a Matlab-Simulink model also used for the validation of the solutions found.

To pursue this goal, two algorithms have been developed, the first to improve the logic of power split between the engine and the electric motor, and the second to increase the gain in terms of emission reductions of the Start&Stop.

The power split logic, initially managed by the ECMS algorithm that in its standard version manages only the torque split required by the driver between the engine and electric motor, has been improved by increasing the control variables of the algorithm with the gear change. In this way, the system has an extra degree of freedom in deciding the optimal split that is chosen among all the possible configurations with which the power train is able to fill the driver's request. The realization of this algorithm took place in order to make it computationally light and feasible in real-time, increasing only the complexity of the cost variable that identifies the point of optimal and improving the exclusion of critical points through boundary conditions. To the cost function have been added components that regulate the gear change (weighing actuation and frequency), the use of the thermal engine (discouraging aggressive transients) and compliance with the required torque.

With the application of this algorithm, called ECMS-GC, an improvement in the vehicle's positions in the ability to follow the command at the same time as a reduction in emissions of 0.73% was observed simulating a WLTP cycle.

The algorithm in question can be the subject of future studies to improve it in several aspects. First, the performance of the algorithm is strongly influenced by the quality of the engine maps provided, so a more defined map would lead to better results. From a hardware point of view, a transmission that can reach more configurations and therefore more working points (for example with a multi-speed gearbox or ideally with a CVT), could lead the algorithm to make more optimal

choices having more points available to choose from. From a software point of view, an effective method could be studied for the choice of all the weights that characterize the cost function to obtain the ideal values for the reduction of consumption; this last study could also be based on the information coming from the ADAS so as to build a predictive-adaptive ECMS that is also based on the environment to determine the best power split.

The Start&Stop has been improved by trying to increase the stop time of the heat engine. To achieve this, two logics have been developed, one for the delay of the re-ignition, the other for the advance of the shutdown. The first makes sure that the re-ignition is implemented only when the ECMS, made aware of the state of the engine, on or off, and the weight that the re-ignition has, decides that it is necessary. The advance of the restart is based on ADAS signals from which it is possible to obtain a preview of the future speed trend or identify a stopping situation and therefore turn off the engine in advance. Thanks to these algorithms, the stop time of the thermal engine increases by 168%, causing a reduction in consumption of 2.13% traveling a WLTP cycle, all with the same vehicle performance.

Future studies for this system could focus on implementing other information about the external environment from, for example, maps or CV2X communication to refine future speed profile prediction and stop recognition. For example, a map where the location of intersections and traffic lights is known, combined with the information coming from the traffic light itself of its state, would let the algorithm understand if there will actually be a stop in the future or not with more precision.

Appendix A

Model images

A.1 Plant

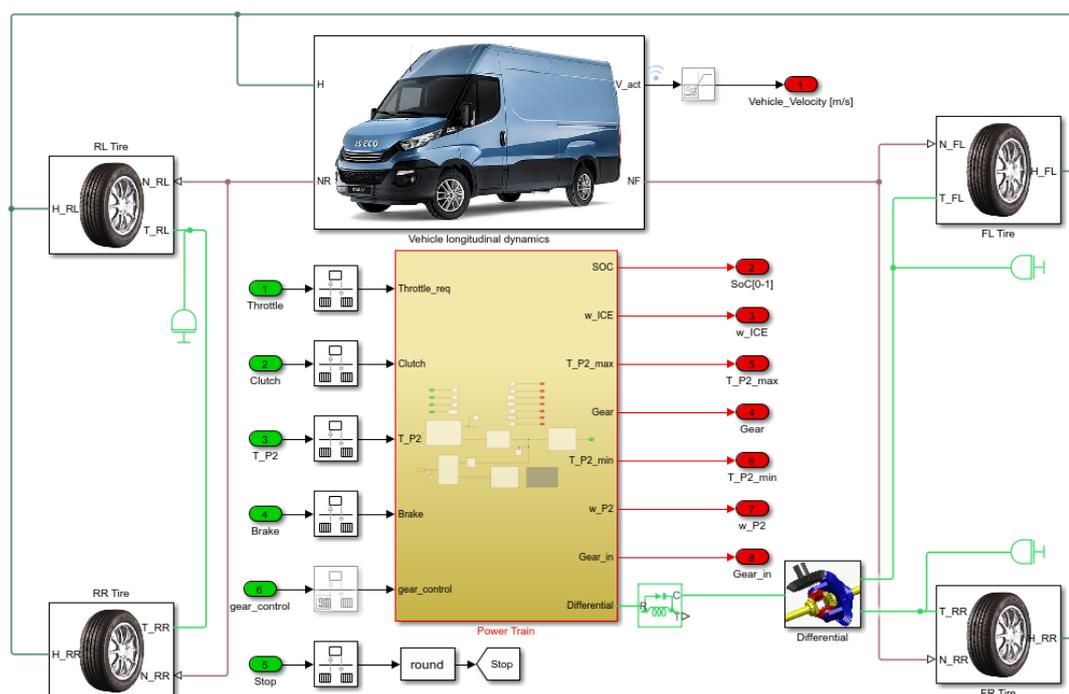


Figure A.1: Plant dynamics Simulink model.

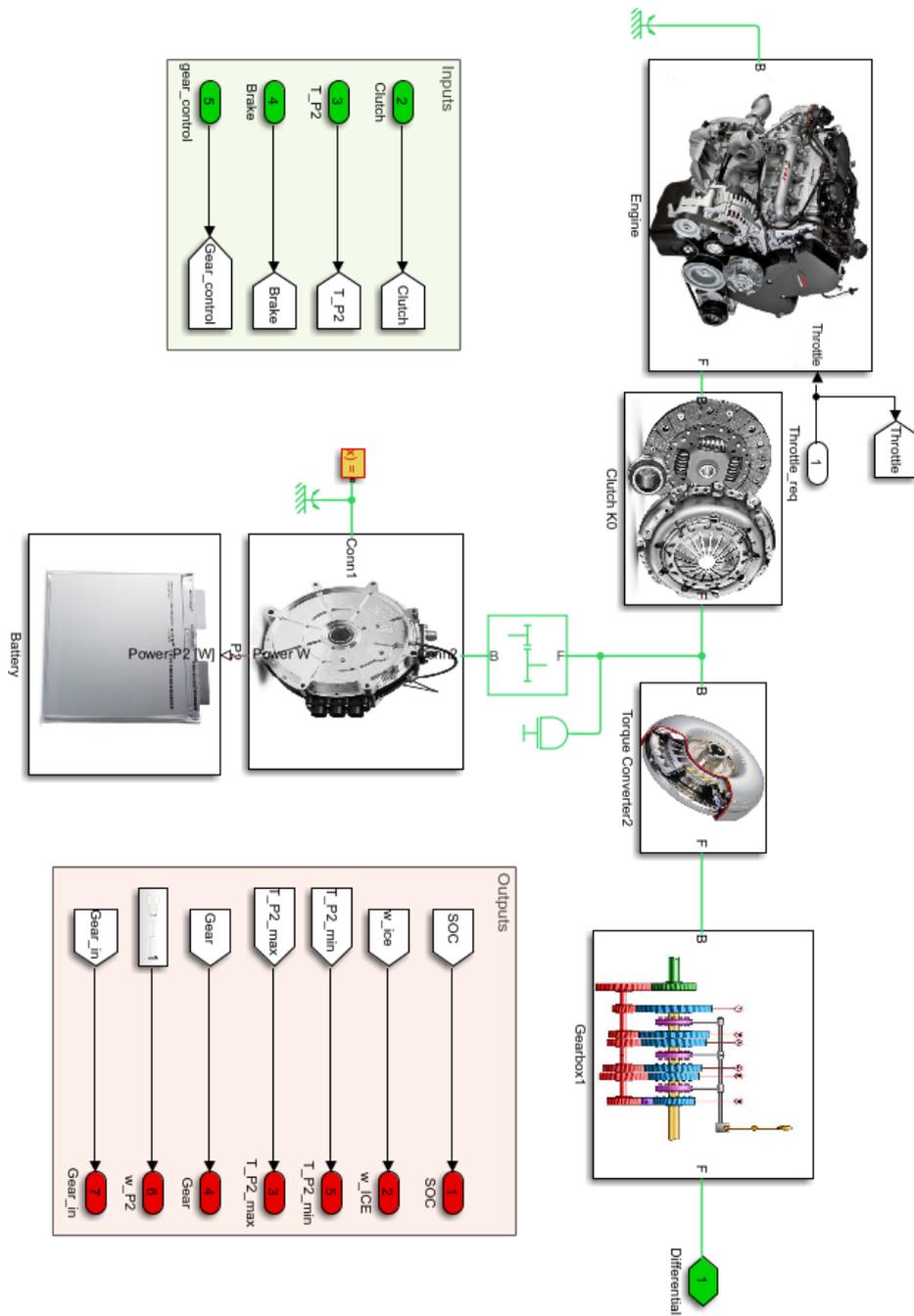


Figure A.2: Plant power-train Simulink model.

A.3 Simulation Interface

Simulation configuration

Gear_velocity
 Gear_ECMS

Power split logic

ADAS

Throttle control

Reference

Start&Stop

initial SOC 

T_step

sim_time

Figure A.4: Simulation configuration interface view.

ECMS controller parameters

- weights concerning gear change :

GC_constant_weight_braking
GC_constant_weight_traction
GC_time_weight_duration
GC_time_weight_intensity

- weights concerning insufficient torque :

torque_insufficient_constant_weight
torque_insufficient_proportional_weight
Threshold_insufficient_torque

- weights concerning delta T_{ice} :

deltaTice_proportional_weight
Threshold_deltaTice

- f(SOC) :

f_SOC
P
I
offset

Figure A.5: ECMS configuration interface view.

Appendix B

Matlab codes

B.1 ECMS-GC

```
1 function [T_ICE_out,T_EM_out, T_brake_out,Gear_out] = fcn(...
    T_req_in,T_mot_avl_vec,T_gen_avl_vec, w_ice_in, w_em_in, ...
    SoC_act,T_ice_avl_vcc,sss,gear_in,gear_act,T_ice_prec,GC_tw,...
    clck,STOP,stop_tr)
2
3 %% parameters initialization
4
5 T_EM_out = 0;
6 T_ICE_out = 0;
7 T_brake_out = 0;
8 Gear_out=0;
9 gear_act=round(gear_act);
10 T_ice_mo=0;
11 T_em_mo=0;
12
13 persistent Gear ice_data w_ice_map T_ice_map Q_ice_map em_data ...
    eff_motor w_em_gen T_em_motor Q_lhv ts eta m_dot_ice ...
    m_dot_em m_dot_eq u P_ratio s_data eta_ice_max s_map ...
    SoC_map s_ch T_ice eff_gen w_em_motor T_em_gen K T_em ...
    m_eq_GC_t m_eq_GC_b m_eq_dT_req_prop m_eq_dT_req_cons ...
    dT_req_max m_eq_dT_ice dT_ice_max cp restart_fuel fuel_data
14
15 if isempty(m_eq_GC_t) % initialize only once in a simulation
16
17     % initialization and fixed parameters
18
```

```

19 cp=load('controller_parameters.mat'); % recover parameters ...
    from HSI
20
21 ice_data = load('ICE_DATA_120422.mat'); %ICE map data
22 w_ice_map = ice_data.w_ice_map*pi/30;
23 T_ice_map = ice_data.T_ice_map;
24 Q_ice_map = ice_data.Q_ice_map;
25
26 % Electric motor
27 em_data = load('Physis350V300Nm.mat'); %EM map data
28 eff_motor = em_data.eff_motor;
29 eff_gen = em_data.eff_gen;
30 w_em_gen=em_data.w_em_gen.*pi/30;
31 w_em_motor = em_data.w_em_motor.*pi/30;
32 T_em_motor = em_data.T_em_motor;
33 T_em_gen = em_data.T_em_gen;
34
35 Q_lhv = 44; % calorific value of the fuel [MW/Kg]
36 eta_ice_max = 0.4; % efficiency of the motor
37
38 K= 1/(Q_lhv*1e3*eta_ice_max);
39 ts = 100; % number of tested possible splits
40 Gear=[ 4.55 , 2.96 , 2.07 , 1.69 , 1.27 , 1 , 0.85 , 0.65 ]; % Gear...
    ratios
41 fuel_data=load('restart.mat');
42 restart_fuel=fuel_data.fuel_consumed_hot*1e-3; %fuel needed...
    to restart the engine
43 end
44
45 % initialize every time step
46 m_eq_GC_t=cp.GC_constant_weight_traction;
47 m_eq_GC_b=cp.GC_constant_weight_braking;
48
49 dT_ice_max=cp.Threshold_deltaTice;
50
51 m_eq_dT_req_prop=cp.torque_insufficient_proportional_weight;
52 dT_req_max= cp.Threshold_insufficient_torque;
53
54 eta = 0;
55 m_dot_eq = inf;
56 m_dot_ice = inf;
57 m_dot_em = inf;
58 T_ice = 0;
59 T_em= 0;
60 T_ICE = 0;
61 T_EM = 0;
62 T_brake = 0;
63 u = zeros(1,ts);
64 m_dot_min=+inf;

```

```

65
66 if gear_act==0
67     gear_act=1;
68 end
69 if gear_in==0
70     gear_in=1;
71 end
72
73 m_dot_min_g=inf*ones(1,8);
74
75
76 %% computations
77
78 s_ch=sss;      % SOC weight factor
79
80 T_wheel = T_req_in; % requested torque after gear box
81 w_wheel = w_em_in/Gear(gear_act); % requested angular velocity...
    after the gearbox
82
83 if T_wheel > 2      % 2 is a threshold for algorithm tolerance
84     %traction situation
85     for g=1:8 % cycling through the gears
86         w_em=w_wheel*Gear(g); % requested velocity at the ...
            motor with the considered gear
87         w_ice=w_em;
88         T_ice_avail=T_ice_avl_vcc(g); % available torques with...
            the considered gear
89         T_mot_avail=T_mot_avl_vec(g);
90         T_gen_avail=T_gen_avl_vec(g);
91         T_req=min(T_ice_avail+T_mot_avail,T_wheel/Gear(g)); %...
            requested torque at the motor with the considered gear
92
93         if w_em<=7000*pi/30
94
95             u_con =[max((T_gen_avail)/(T_req),-1) min((...
            T_mot_avail)/(T_req), 1)]; % bounds of possible splits
96             u =sort( [ linspace(u_con(1),0, ts/2),linspace(0,...
            u_con(2), ts/2)]); %possible split conditions(percentage ...
            of electric torque)
97
98             for i = 1:ts
99                 m_dot_ice = inf;
100
101
102                 if w_em>4000*pi/30 % only electric available
103                     m_dot_ice =interpn(T_ice_map,w_ice_map,...
            Q_ice_map,0,750*pi/30,'linear',-1);
104                     T_ice_avail=0;

```

```

105         u(i)=1; % impose all electric split and ...
no torque available from ICE motor
106         end
107
108         if w_ice<=4000*pi/30 && w_ice>=750*pi/30 % ...
ICE working conditions
109
110             T_ice = min(T_req*(1-u(i)), T_ice_avail); %...
ICE torque for each possible split
111             m_dot_ice = interpn(T_ice_map, w_ice_map, ...
Q_ice_map, T_ice, w_ice, 'linear', -1);
112
113         else % ICE cannot work
114             T_ice=0;%min(max(0,T_req*(1-u(I))) ,...
T_ice_avail); % ICE torque
115             m_dot_ice = interpn(T_ice_map, w_ice_map, ...
Q_ice_map, T_ice, 750*pi/30, 'linear', -1); %idle consumption
116
117         end
118         if (T_ice)>(T_ice_avail) %&& (u(i)<0) % ...
discard impossible torque requests
119             m_dot_ice = 10000000;
120         end
121         if (m_dot_ice< 0) || (isnan(m_dot_ice)) % ...
discard impossible m_dot_ice cases
122             m_dot_ice = 10000;
123
124         end
125
126
127         if T_ice<0.1 % idle consumption for small ...
torques
128             m_dot_ice = interpn(T_ice_map, w_ice_map, ...
Q_ice_map, T_ice, 750*pi/30, 'linear', -1);
129         end
130
131         if stop_tr==1 %engine off
132             m_dot_ice =0;
133             T_ice=0;
134             u(i)=1; % impose all electric split and...
no torque available from ICE motor
135         end
136
137         if u(i)>=0
138             s_ch=sss;
139             T_en=min(max(T_req*u(i), T_req-T_ice), ...
T_mot_avail); % EM torque computation
140         else
141             s_ch=0.5; % discourage negative splits

```

```

142         T_em=max(max(min(T_req-T_ice,0) ,T_req*u(i)...
143     ),T_gen_avail);
144         end
145         if T_em>=0
146             eta = interpn(T_em_motor,w_em_motor, ...
147     eff_motor, T_em,w_em,'linear',-1); % pick efficiency of ...
148     motor with the coinsidered Torque and velocity
149         else
150             eta = interpn(-T_em_gen,w_em_motor, eff_gen...
151     , -T_em,w_em,'linear',-1); % pick efficiency of generator...
152     with the coinsidered Torque and velocity
153         end
154         if (eta <= 0) || (isnan(eta))
155             % discard impossible efficiency cases
156             if T_em>=0
157                 eta = 2.2440e-04;
158             else
159                 eta = -2.2440e04;
160             end
161         end
162         if T_em>=0
163             m_dot_em = K*T_em*w_em/eta; % computation ...
164     electric equivalent fuel mass
165         else
166             m_dot_em = K*T_em*w_em*eta;
167         end
168         if T_wheel-(T_ice+T_em)*Gear(g)> dT_req_max %...
169     weight concerning a supplying of insufficient torque
170         m_eq_dT_req_cons=...
171     cp.torque_insufficient_constant_weight;
172         else
173             m_eq_dT_req_cons=0;
174         end
175         if T_ice-T_ice_prec>dT_ice_max % weight ...
176     concerning quick variation in ICE torque
177         m_eq_dT_ice=...
178     cp.deltaTice_proportional_weight;
179         else
180             m_eq_dT_ice=0;
181         end
182         if u(i)==1 % if only electric configuration

```

```

179             m_restart=0;    % no fuel addition for ...
restart
180             else
181             m_restart=STOP*restart_fuel;    %fuel ...
addition only if the engine was off
182             end
183
184             % equivalent fuel
185
186             m_dot_eq = m_dot_ice + s_ch*m_dot_em + ...
m_eq_GC_t*abs(g-gear_act)*GC_tw + m_eq_dT_req_cons+(...
m_eq_dT_req_prop/s_ch)*abs((T_ice+T_em)*Gear(g)-T_wheel) + ...
m_eq_dT_ice*abs(T_ice-T_ice_prec)+m_restart;
187
188             if m_dot_eq <= m_dot_min    % condition of best ...
split combination
189                 m_dot_min_g(g)=m_dot_eq;
190                 m_dot_min=m_dot_eq;
191                 T_ICE_out = T_ice;
192                 T_EM_out=T_em;
193                 Gear_out=g;
194                 w_out=w_ice;
195                 T_ice_mo=T_ice_avail;
196                 T_em_mo=T_mot_avail;
197             end
198         end
199     end
200 end
201
202
203 elseif T_wheel<=-2    % braking situation    ...
204
205     T_ICE_out=0;    % use only EM
206
207     for g=1:8 % cycling through the gears
208
209         T_req=T_wheel/Gear(g);
210         w_em=w_wheel*Gear(g);
211         T_gen_avail=T_gen_avl_vec(g);
212
213         if w_em<=7000*pi/30
214
215             T_em=max(T_gen_avail, T_req);
216
217             eta = interpn(-T_em_gen,w_em_motor, eff_gen, -T_em,...
w_em, 'linear',-1);    % pick efficiency of generator with ...
the considered Torque and velocity
218

```

```
219         if (eta <= 0) || (isnan(eta))    % discard ...
impossible efficiency cases
220
221             eta = -2.2440e04;
222
223         end
224
225         m_dot_em = K*T_em*w_em*eta;
226         % equivalent fuel
227         m_dot_eq = m_dot_em+ m_eq_GC_b*abs(g-gear_act)*...
GC_tw;
228
229         if m_dot_eq <= m_dot_min    % condition of best ...
split combination
230             m_dot_min=m_dot_eq;
231             T_ICE_out = 0;
232             T_EM_out=T_em;
233             T_brake_out=T_req-T_em;
234             Gear_out=g;
235             w_out=w_em;
236         end
237     end
238 end
239
240
241 elseif abs(w_wheel)<0.5    % small torque, no velocity
242     m_dot_min=0;
243     T_ICE_out = 0;
244     T_EM_out=0;
245     Gear_out=0;
246     w_out=w_wheel;
247 else    % small torque, vehicle moving
248     Gear_out=gear_act;
249     T_ICE_out = 0;
250     T_EM_out=0;
251     m_dot_min=0;
252     w_out=w_wheel;
253 end
254
255 end
```

B.2 ADAS based Start&Stop

B.2.1 Velocity prediction analysis triggered version

```

1 function engine_off = fcn(t,T_pre, engine_state)
2 %t is the actual time
3 %T_pre is the prediction horizon
4
5 persistent V_z T_z A_z T_cutoff a b c
6 if isempty(V_z)
7 %initialize once in a simulation
8     profile=load('V_z_h.mat')
9     V_z=profile.V_z; %velocity profile
10    A_z=profile.A_z; %acceleration profile
11    T_z=profile.T_z;
12    T_cutoff=8; %threshold time to be worth to shut the engine...
13    coast_down=load('coast_down.mat') %coast down parameters
14    a=coast_down.a;
15    b=coast_down.b;
16    c=coast_down.c;
17 end
18 ind_in=round(1+t/0.05); %get the index on the velocity profile ...
19 ind_fin=min(length(T_z),round(ind_in+T_pre/0.05)); %get the ...
20 ind_fina=min(length(T_z)-1,round(ind_in+T_pre/0.05));
21 vec_v=V_z(ind_in(1):ind_fin(1)); %velocity prediction vector
22 vec_a=A_z(ind_in(1):ind_fina(1)); %acceleration prediction ...
23
24 T_off=0;
25
26 a_cd=polyval([c b a],vec_v); % coast down acceleration for ...
27
28 if engine_state == 0 %engine off
29     for i=1:length(vec_a)
30         if vec_a(i)<= a_cd(i) || vec_a(i)==0 && vec_v(i)==0
31             %no traction needed
32             T_off=T_off+0.05; %increase possible stop time
33         else
34             %need traction
35             break %stop count

```

```

36     end
37 end
38 if T_off >= T_cutoff
39     %it worth to stop the engine
40     engine_off=1;
41 else
42     engine_off=0;
43 end
44 else %engine is already stop
45     %consider only the actual condition.
46     if isempty(vec_a)==1
47         vec_a=0;
48     end
49     if vec_a(1) < a_cd(1) || vec_v (1) < 0.1
50         %no traction needed
51         engine_off=1;
52     else
53         %traction needed
54         engine_off=0;
55     end
56 end
57 end

```

B.2.2 Stop-triggered version

```

1 function STOP = fcn(STOP_p,T_ice_ECMS,Trig)
2 %STOP_p is the previous state
3 %T_ice_ECMS is the request of torque from ECMS
4 %Trig is the stop detected signal
5 %STOP is the command
6
7
8 if Trig==1
9     STOP=1;
10 else
11     if STOP_p==1 && T_ice_ECMS<=1
12         STOP=1;
13     else
14         STOP=0;
15     end
16 end
17 end

```

Bibliography

- [1] C. Bauer R. Sacchi et al. «When, where and how can the electrification of passenger cars reduce greenhouse gas emissions?» In: *Renewable and Sustainable Energy Reviews* (2022). DOI: <https://doi.org/https://doi.org/10.1016/j.rser.2022.112475> (cit. on p. 1).
- [2] J. Gessat. «Electrically Powered Hydraulic Steering Systems for Light Commercial vehicles». In: *SAE Technical Paper* (2007). DOI: <https://doi.org/https://doi.org/10.4271/2007-01-4197> (cit. on p. 1).
- [3] Lorenzo Serrao Simona Onori and Giorgio Rizzon. «Hybrid Electric Vehicles Energy Management Strategies». In: *Springer* (2015). DOI: <https://doi.org/https://doi.org/10.4271/2007-01-4197> (cit. on p. 1).
- [4] A. Sciarretta and L.Guzzella. «Control of Hybrid Electric Vehicles». In: *IEEE control system magazine* 27.2 (2007) (cit. on p. 1).
- [5] P. Pisu and G. Rizzoni. «A supervisory control strategy for series hybrid electric vehicles with two energy storage systems». In: *2005 IEEE Vehicle Power and Propulsion Conference* (2005). DOI: <https://doi.org/10.1109/VPPC.2005.1554534> (cit. on p. 2).
- [6] E. Galvagno G. R. Guercioni et al. «Adaptive Equivalent Consumption Minimization Strategy With Rule-Based Gear Selection for the Energy Management of Hybrid Electric Vehicles Equipped With Dual Clutch Transmissions». In: *IEEE Access* 8 (2020). DOI: <https://doi.org/10.1109/ACCESS.2020.3032044> (cit. on p. 2).
- [7] J.W. Grizzle Chan-Chiao Lin Huei Peng et al. «Power management strategy for a parallel hybrid electric truck». In: *IEEE Transactions on Control Systems Technology* (2003). DOI: <https://doi.org/10.1109/TCST.2003.815606> (cit. on p. 2).
- [8] J.W. Grizzle Chan-Chiao Lin Huei Peng. «A stochastic control strategy for hybrid electric vehicles». In: *Proceedings of the 2004 American Control Conference* (2003). DOI: <https://doi.org/10.23919/ACC.2004.1384056> (cit. on p. 2).

-
- [9] R. Cupek A. Ziebinski et al. «Review of advanced driver assistance systems (ADAS)». In: *AIP Conference Proceedings 1906* (2017). DOI: <https://doi.org/10.1063/1.5012394> (cit. on p. 2).
- [10] «Piattaforma tecnologica di Filiera PiTeF». In: (). URL: <https://bandi.regione.piemonte.it/contributi-finanziamenti/piattaforma-tecnologica-filiera-pitef> (cit. on p. 3).
- [11] «Dayco Europe S.r.l.» In: (). URL: <https://www.dayco.com/it/azienda/> (cit. on p. 3).
- [12] «Tecno System S.p.a.» In: (). URL: <https://tecnosystem.it/about/> (cit. on p. 3).
- [13] «Italtecnica S.r.l.» In: (). URL: <https://www.italtecnica.com/it/content/azienda> (cit. on p. 3).
- [14] «Podium Engineering S.r.l.» In: (). URL: <https://www.podium-tech.com/company/> (cit. on p. 3).
- [15] «Politecnico di Torino». In: (). URL: <https://www.polito.it> (cit. on p. 3).
- [16] E. Lavenia. «Energy flow management of P2 hybrid vehicle based on ADAS sensors». In: *Politecnico di Torino* (2022) (cit. on pp. 19, 20, 27).
- [17] «Adaptive Cruise Control with Sensor Fusion». In: (). URL: <https://it.mathworks.com/help/mpc/ug/adaptive-cruise-control-with-sensor-fusion.html> (cit. on pp. 22, 27).
- [18] «What Is a Live Script or Function?» In: (). URL: https://it.mathworks.com/help/matlab/matlab_prog/what-is-a-live-script-or-function.html (cit. on p. 23).
- [19] «EU: Cars and Light Trucks». In: (). URL: <https://dieselnet.com/standards/eu/ld.php> (cit. on p. 24).
- [20] «Worldwide Harmonized Light Vehicles Test Cycle (WLTC)». In: (). URL: <https://dieselnet.com/standards/cycles/wltp.php#hev> (cit. on pp. 25, 39).
- [21] Guoqiang Li and Daniel Goerges. «Optimal integrated energy management and shift control in parallel hybrid electric vehicles with dual-clutch transmission». In: *Sage Journal* 234 (June 2019), pp. 599–609 (cit. on p. 43).
- [22] Guido Ricardo Guercioni et al. «Adaptive Equivalent Consumption Minimization Strategy With Rule-Based Gear Selection for the Energy Management of Hybrid Electric Vehicles Equipped With Dual Clutch Transmissions». In: *IEEE* 8 (Oct. 2020), pp. 190017–190038 (cit. on p. 43).

- [23] Jan Wikander Mohammad Khodabakhshian Lei Feng. «Optimization of Gear Shifting and Torque Split for Improved Fuel Efficiency and Drivability of HEVs». In: *SAE International* (Aug. 2013) (cit. on p. 43).
- [24] Jan Wikander Mohammad Khodabakhshian Lei Feng. «One-Step Prediction for Improving Gear Changing Control of HEVs». In: *Journal of Robotics and Mechatronics* 26.6 (2014) (cit. on p. 43).
- [25] Yakhshilikova G. et al. «Development of Optimization Based Control Strategy for P2 Hybrid Electric Vehicle including Transient Characteristics of Engine». In: *APPLIED SCIENCES. - ISSN 2076-3417. - ELETTRONICO* (2022), p. 2852 (cit. on p. 52).
- [26] Li Yinghao Sheng Qiao Yang Yanding et al. «Application of Engine intelligent start-stop system in technology of vehicle fuel saving». In: *Sixth International Conference on Measuring Technology and Mechatronics Automation* (2014). DOI: 10.1109/ICMTMA.2014.35 (cit. on p. 63).
- [27] A. Emadi R. Jayabalan. «Acceleration support by integrated starter/alternator for automotive applications». In: *Journal of Automobile Engineering* 218.9 (2004), pp. 987–993 (cit. on p. 63).
- [28] Y. Su F. Xie W. Hong et al. «Direct start of a gasoline direct-injection engine without a starter». In: *Journal of Automobile Engineering* 230.4 (2016), pp. 491–502 (cit. on p. 63).
- [29] N. Chalhoub N. A. Henein D. Taraza et al. «Ex-ploration of the contribution of the start/stop transients in hev operation and emissions». In: *SAE Technical Paper* (2000) (cit. on p. 63).
- [30] Steve Yurkovich Marcello Canova Yann Guezennec. «On the Control of Engine Start/Stop Dynamics in a Hybrid Electric Vehicle». In: *Journal of Dynamic Systems, Measurement, and Control* 131 (2009) (cit. on p. 63).
- [31] Joseph Paul Porembski. «Control of Start and Stop events in a HEV». In: *The Ohio State University* (2007) (cit. on p. 63).
- [32] HaiLin Kui Cuizhu Bao Chen Chen et al. «SSVisual: Intelligent Start-Stop System». In: *20th IEEE International Conference on Mobile Data Management* (2019). DOI: 10.1109/MDM.2019.00–30 (cit. on p. 63).
- [33] Gheorghe Ciolan Ion Preda Dinu Covaciu. «COAST DOWN TEST – THEORETICAL AND EXPERIMENTAL APPROACH». In: *CONAT20104030* (2010). DOI: 10.13140/RG.2.1.4048.5925 (cit. on p. 69).
- [34] Liu H Zhang X et al. «The intelligent engine start-stop trigger system based on the actual road running status». In: *PLoS ONE* 16(6): e0253201 (2021). DOI: <https://doi.org/10.1371/journal.pone.0253201> (cit. on p. 71).