Alma Mater Studiorum - University of Bologna

School of Engineering

Department of Industrial Engineering
Second-cycle Degree in Mechanical Engineering

Master degree thesis
in
MODELING AND CONTROL OF INTERNAL COMBUSTION ENGINES AND HYBRID PROPULSION SYSTEMS M

Development of a speed profile prediction algorithm based on navigation data for energy management optimization

Candidate
Mattia Boccolini

Advisor
Prof. Nicolò Cavina

Co-Advisors
Ing. Alessandro Perazzo
Ing. Lorenzo Brunelli
Ing. Alessandro Capancioni
Prof. Davide Moro
Prof. Enrico Corti

Academic Year 2020-2021
Session III

Alma Mater Studiorum - Università di Bologna

Scuola di Ingegneria

Dipartimento di Ingegneria Industriale
Corso di Laurea Magistrale in Ingegneria Meccanica

Tesi di laurea

in
MODELING AND CONTROL OF INTERNAL COMBUSTION ENGINES AND HYBRID PROPULSION SYSTEMS M

Development of a speed profile prediction algorithm based on navigation data for energy management optimization

Candidato
Mattia Boccolini

Relatore
Prof. Nicolò Cavina

Correlatori
Ing. Lorenzo Brunelli
Ing. Alessandro Capancioni
Ing. Alessandro Perazzo
Prof. Davide Moro
Prof. Enrico Corti

Anno Accademico 2020-2021
Sessione III

Agli Essenziali

La somma delle radici quadrate di qualunque
due lati di un triangolo isoscele
è uguale alla radice quadrata del rimanente.

[SCARECROW
THE WONDERFUL WIZARD OF OZ]

Abstract

During the last decades the number of vehicles on the roads progressively rises together with the relative emissions of pollutants, such as CO_2 and other greenhouse gases, worsening air quality and traffic conditions especially in metropolitan areas. The OEM (Original Equipment Manufacturer) are investing a lot of money in alternative solutions, progressively abandoning the Internal Combustion Energy which can no longer satisfy the more and more stringent regulations as well as the market requests. The general tendency is to introduce an alternative source of energy flanking the conventional engine, allowing its downsizing and helping it during the less efficient operating points. Moreover, the new kind of sensor systems (named Advanced Driver-Assistance Systems) are starting to be implemented on-board in addition to connectivity devices. From this union, a connected vehicle arise, able to exchange data with the surrounding environment and give the driver new kind of assistance functionality, like the Lane Departure Warning, Adaptive Cruise Control and Parking Assistance, avoiding dangerous or inefficient decisions. In parallel, these functions can recreate an electronic horizon, based on the path selected by the driver, supplying to the Control Unit a detailed preview. The natural tendency is to gradually limiting the driver control on the vehicle up to definitely excluding him from driving decisions. Nevertheless, the implementation of fully working and safe ADAS functions implies millions of kilometres of road test validations, to safely introduce them on the market. Thus, the OEM have to develop new methodologies to substitute the road tests with specific simulations, where tools (like Model-in-the-Loop, Software-in-the-loop, Hardware-in-the-Loop) are used to verify its reliability.

This master thesis work wants to develop and implement an on-board algorithm for the speed prediction of a connected vehicle along a given route, by elaborating real-time navigation data. In particular, information about the legal speed limits, the traffic density and the presence of the so-called "stop events", such as traffic lights and roundabouts, are sent to the vehicle by the map service provider and they constitute the input for the algorithm. The algorithm allows you to select one out of three different driver type (quiet, normal, aggressive): this choice, together with the performance parameters of the vehicle, influences the acceleration and braking phases of the prediction. Once the prediction is generated, it constitutes an input for the predictive ADAS functions and energy management functions. The work has been split in three main phases: development, calibration end validation. During the first phase, the logics of the algorithm have been implemented by means of a Simulink block in order to be included into the Hybrid Control Unit at the HiL. Subsequently, the calibration of the key parameters took place by means of real speed profile analysis. In the end, the behaviour of the algorithm has been investigated studying its response in different scenarios.

The master thesis activity has been carried out at Green Mobility Research Laboratory, born by the collaboration between the University of Bologna and FEV Italia s.r.l, internationally recognized leader in design and development of advanced gasoline, diesel and hybrid powertrains and vehicle systems.

Abstract in lingua italiana

Negli ultimi decenni si è registrato un progressivo aumento dell'utilizzo di mezzi di trasporto individuali e delle relative emissioni di agenti inquinanti, come CO_2 e altri gas serra, peggiorando la qualità dell'aria e la viabilità, specialmente nelle aree metropolitane. Le case costruttrici stanno investendo molte risorse in soluzioni alternative, riconoscendo come i motori a combustione interna, per quanto efficienti questi possano essere, non riescano più a soddisfare le richieste di mercato e rispettare la legislazione sempre piu stringente. L'obiettivo dichiarato è l'abbandono graduale dei combustibili fossili in favore di una progressiva elettrificazione dei veicoli. L'implementazione di una fonte di energia alternativa, in particolare quella elettrica, permette una riduzione delle dimensioni del motore termico e un aiuto nei punti operativi meno efficienti ma richiede una nuova concezione della sua gestione a bordo. In più, l'istallazione nel veicolo di nuovi sistemi di sensoristica avanzata (in gergo Advanced Driver-Assistance Systems) e di moderni dispositivi di connectivity, lo rendono capace di comunicare con l'ambiente circostante, aprendo all'introduzione di nuove funzionalità di supporto al guidatore, come il Lane Departure Warning, Adaptive Cruise Control e Parking Assistance, per evitare che esso compia scelte pericolose o inefficienti. Parallelamente queste funzioni possono permettere la ricostruzione di un orizzonte elettronico, basato sul percorso deciso dal guidatore, fornendone alla centralina un'anteprima dettagliata. La conseguenza naturale sara quella di limitare progressivamente il controllo del guidatore sul veicolo fino ad escluderlo definitivamente. Tuttavia, l'implementazione di funzionalita ADAS pienamente funzionanti e sicure prevedrebbe milioni di chilometri di test su strada, prima di introdurle sul mercato. Ciò obbliga i costruttori a sperimentare nuove tipologie di validazione per soddisfare questa esigenza sostituendo i test stradali con specifiche simulazioni, all’interno delle quali vengono utilizzati strumenti (Model-in-the-Loop, Software-in-the-Loop, Hardware-in-the-Loop) per verificarne l’affidabilita. Questo elaborato si pone l’obbiettivo di implementare un algoritmo on-board per la predizione della velocità di un veicolo connesso, lungo un dato percorso, per mezzo dell’elaborazione dei dati di navigazione real-time. In particolare, le informazioni riguardanti i limiti di velocità, la densità di traffico e la presenza di eventuali ”stop events” come semafori e rotonde, vengono inviate al veicolo dal map service provider e costituiscono gli input dell’algoritmo. L’algoritmo permette di scegliere tra tre diversi tipi di driver (prudente, normale, aggressivo): questa selezione, insieme ai parametri di prestazioni del veicolo pre-impostati, andrà ad influenzare le fasi di accelerazione e decelarazione nella predizione. Una volta generata, tale predizione costituisce l’input delle funzioni ADAS predittive e di quelle addette all’energy management. Il lavoro è stato suddiviso in tre fasi principali: sviluppo, calibrazione e validazione. Durante la fase di sviluppo, le logiche dell’algoritmo sono state implementate in Simulink in modo da poter essere inserite nell’ Hybrid Control Unit (HCU) del Software-in-the-Loop. Successivamente, è stata effettuata la calibrazione dei paramentri principali dell’algoritmo, basata sull’analisi di profili di velocità acquisiti su strada. Infine, è stata investigata la risposta dell’algoritmo a scenari diversificati.

Il lavoro di tesi è stato svolto presso il Green Mobility Research Laboratory, nato dalla collaborazione tra l’Università di Bologna e FEV Italia s.r.l., compagnia internazionale leader nella progettazione e nello sviluppo di powertrain e sistemi veicolo.

Acknowledgments

My heartfelt thanks to Professor Nicolò Cavina, who has given me the opportunity to start this formative project, but above all for letting me understand what I want to do when i grow up. Another special thanks to Ing. Alessandro Capancioni and Lorenzo Brunelli for guiding me during the whole project. I cannot omit to thank all the FEV Bologna team, for welcoming me and for making me feel part of it from the first day. Lust but not least, I have to say thank you to Denise and Rebecca for giving me a ride to the laboratory every day, and to Giorgio for being my ”driver type: normal” during the measurements.

Contents

Abstract ix

Abstract in lingua italiana xi

Acknowledgments xiii

List of figures xvii

List of tables xix

Nomenclature 1

1 Introduction 1
- 1.1 Motivations, challenges and targets . . . 1
- 1.2 ADAS . . . 8
- 1.3 Connectivity . . . 10
  - 1.3.1 Predictive driving (eHorizon) . . . 11

2 Simulation environment 15
- 2.1 Simulation environment . . . 15
- 2.2 Model-in-the-Loop . . . 16
- 2.3 Software-in-the-Loop . . . 17
- 2.4 Connected Hardware-in-the-Loop . . . 19

3 Speed Profile Prediction 21
- 3.1 State of the art . . . 21
- 3.2 Previous version . . . 22

3.3 Speed Profile Prediction algorithm 23
    3.3.1 Traffic density 24
    3.3.2 Stop events 25
    3.3.3 Space based driver model 27
    3.3.4 Oscillation around constant speed segments 29

4 Calibration procedure **33**
4.1 Methodology 33
    4.1.1 Scenarios 33
    4.1.2 Measurements 35
    4.1.3 Key Performance Indicators 37
4.2 Calibration 39
    4.2.1 Acceleration and braking maneuvers 40
    4.2.2 Traffic: influence on average speed and oscillations 40

5 Results and validation **45**
5.1 Scenario 1 45
5.2 Scenario 2 47
5.3 Scenario 3 50

6 Conclusions and future works **53**
6.1 Initial and final speed 53
6.2 Oscillation energy 54
6.3 Road curvature radius and weather conditions 55
6.4 Stop event duration, slope and eco-Driving 56

Bibliography **59**

List of figures

1.1 Overview of the Hybrid Electric Vehicle depending on Hybridization and CO_2 reduction 4
1.2 Fuel specific energy in function of their volumetric density 5
1.3 Parallel hybrid power characteristics 5
1.4 Series/Parallel hybrid power characteristics 6
1.5 Series hybrid power characteristics 6
1.6 Parallel hybrid driveline architecture 7
1.7 On-board sensing equipment 9
1.8 Automation levels according to SAE 10
1.9 V2X connectivity 11
1.10 Ideal predictive strategy 12
1.11 Zero Emission Zone 13

2.1 V-Model methodology layout 16
2.2 Software-in-the-Loop scheme 18

3.1 Previous version of the speed profile prediction 23
3.2 Green traffic light probability depending on traffic code. 26
3.3 Verified conditions for acceleration. 28
3.4 Effects of different $\tau = \frac{1}{T}$ in a first order system undergoing to a step stimulus. 29
3.5 Measured speed profile Vs. "zero-noise" predicted speed profile 30
3.6 Measured speed profile Vs. Predicted speed profile 31

4.1 1st scenario: 0-50-0 km/h acceleration and deceleration 34

4.2 2nd scenario: from Via Umberto Terracini to Piazza Grigoris Lambrakis, Bologna 35
4.3 Measurement chain 36
4.4 u-center GUI: speed over time diagram view 36
4.5 u-center GUI: table view with time, speed, longitude and latitude . . 36
4.6 Acceleration and braking: before(left) and after(right) calibration . . 40
4.7 Example of FFT applied to a distance-based measured signal 41
4.8 Frequency and magnitude's scatter in case of traffic code 'orange' . . 43

5.1 Scenario 1: speed 46
5.2 Scenario 1: energy 47
5.3 Scenario 2: speed 48
5.4 Scenario 2: energy 49
5.5 Scenario 3: speed 51
5.6 Scenario 3: energy 51

6.1 Effects of oscillations on KPI 54
6.2 Limit of adherence C 56
6.3 Predictive Functions layout 57

List of tables

3.1 SPP input 24
3.2 Stop event typologies 25

4.1 Calibration parameters 39

5.1 Scenario 1:speed KPI 45
5.2 Scenario 1:energy KPI 46
5.3 Scenario 2:speed KPI 47
5.4 Scenario 2:energy KPI 48
5.5 Scenario 3:speed KPI 50
5.6 Scenario 2:energy KPI 50

Chapter 1

Introduction

1.1 Motivations, challenges and targets

Starting from the twentieth century, the population growth, together with the technological development, has lead the industrialization process to completely new scenarios: the birth of more and more factories and the consequent movement of the people from the countryside to the city centres. This period saw an exponential growth of production, giving the people technologies, once prohibitive for the price and now more accessible. As a natural consequence of the economic boom, in parallel with a general carelessness and incompetence about environment, the emissions of carbon dioxide (CO_2) and other greenhouse gases (GHG) rapidly increased over the year, becoming one of the most challenging issues of the present time. An important role is played by the automotive industries, in fact, cars are used throughout the world and they have become the most adopted solution for people transportation in many countries. Transportation was responsible for 24% of direct CO_2 emissions in 2017. The 77% of both global final energy demand and CO_2 emissions are accountable to the transport sector as a whole, comprehending cars, trucks, buses and two-wheelers. Moreover, car buyers continue to choose bigger, heavier vehicle and this has lead to a rise in the average new car CO_2 emissions in 2017 [1]. Therefore, the European Union has made substantial efforts tightening the CO_2 maximum limit on a Worldwide harmonized light vehicles test procedure from 130 ${}^{g_{CO_2}}/_{km}$ of 2015 to 95 ${}^{g_{CO_2}}/_{km}$ by 2021 [2], aiming at the fixed target of 60

${}^{g_{CO_2}}/_{km}$ by 2030 [10]. Hence, automobile manufacturers and engineers have spent the last decade trying to develop innovative solutions with the double purpose of satisfying the market request and complying to the regulations, increasingly stringent. The result of these years of research is the decision to adopt other form of energy supporting the conventional engine. In the so-called Hybrid Vehicles the primary energy source is generally an Internal Combustion Engine (ICE); depending on the nature of the secondary source, ”Hybrid” can mean

- **Hydraulic Hybrid** that kind of vehicles have a hydraulic pump as secondary mover or generator, which stores the energy in an auxiliary hydraulic accumulator where oil is used as operator fluid. For their weight and their characteristics, this powertrain is particularly indicated for heavy-duty vehicles;

- **Kinetic Hybrid** kinetic hybrid powertrain means a driveline with a highspeed flywheel as auxiliary mover, with the possibility of storing kinetic energy, especially during regenerative braking; [6]

- **Compressed-air Hybrid** these vehicle are powered by motors which produce power with the compressed-air expansion in a similar way of the steam engine. As a non-flammable fluid, the compressed-air can be stored in pressurized tank at 30 MPa;

- **Electric Hybrid** here, the auxiliary energy source is the electro-chemical energy provided by Electric Motors and the batteries are the storage system which can be recharged during breaking or with the ICE.

The more promising technology in term of CO_2 reduction is the Hybrid Electric Vehicle (HEVs). Focusing on them, the level of hybridization depends on the range of action of the electric motor which is indicated by the *Hybridization Degree* (HD) and described by the eq.(1.1)

$$HD = \frac{P_{S,max}}{P_{S,max} + P_{ICE,max}} \tag{1.1}$$

where $P_{S,max}$ is the maximum power of the secondary source of energy while $P_{ICE,max}$

is the maximum power deliverable by the ICE. Consequently, it is possible to distinguish the following typology of HEVs:

- **Micro Hybrid** with a $HD \sim 5\%$, it's a vehicle equipped with an Electric Motor (EM) linked to the ICE and it can only have Start and Stop functionality. Most of the them have also some sort of Energy Management function, which optimizes the consumption of the low voltage (12 V) battery energy [26];
- **MHEV** (Mild Hybrid EV) with a HD $\sim$ 15%, these types generally use a compact electric motor (usually $< 20kW$) to provide auto-stop/start features, extra power assist during the acceleration and to work as a generator on the deceleration phase (regenerative braking). The battery is a Low Voltage Battery of 48V, whose purpose is to actuate an Energy Management Strategy (EMS) and it allows a minimum range of full-electric drive[1].
- **FHEV** (Full Hybrid Electric Vehicle) where the $HD \sim 35\%$, the Electric Machines and batteries are increased in size, allowing an extended full-electric drive. The recharging of the batteries can happen only with breaking recuperation and with the ICE, because it isn't possible to do from external sources;
- **PHEV** (Plug-in HEV) is usually a general fuel-electric Off-Vehicle Charging (OVC) hybrid vehicle with increased energy storage capacity and a $HD \sim 50\%$. This allows the vehicle to drive on all-electric mode a distance that depends on the battery size and its mechanical layout (series or parallel). At the end of the journey, it may be connected to mains electricity supply through a socket to avoid recharging using the on-board internal combustion engine. This concept is attractive to those seeking to minimize on-road emissions by avoiding - or at least minimizing - the use of ICE during daily driving. As with pure electric vehicles, the total emissions saving, for example in CO_2 terms, is dependent upon the source of the energy produced by the provider company;
- **BEV** (Battery EV) are vehicles where there isn't an ICE and the traction is granted only by electric motors powered by batteries. Properly, they're not

---

[1] Exceptions of full-electric drive vehicle equipped with a 48V battery are the MEET model developed by MAHLE[18] or the one designed by Valeo[22]

hybrid vehicle, because the energy source is only one, but it will be the arrival point of the transaction where the hybrid vehicles are only intermediate step. At the moment, the main problem of BEV is the capacity of the battery cells, so how the energy is stored [26]. In Fig. 1.2, the fuel (gaseous and liquid) and batteries specific energy is represented in function of their volumetric density, there the problematics of the batteries compared to the other fuels are clearer. To make a more practical example, the same energy needed for a drive of about 500km is stocked in 46 litres ($\sim$ 43kg) of gasoline but in more than 700kg of batteries. Nevertheless, from the dawn of the batteries for automotive purpose, thanks to the improvement in technology their cost becomes cheaper and cheaper, while their energy density increases [21].

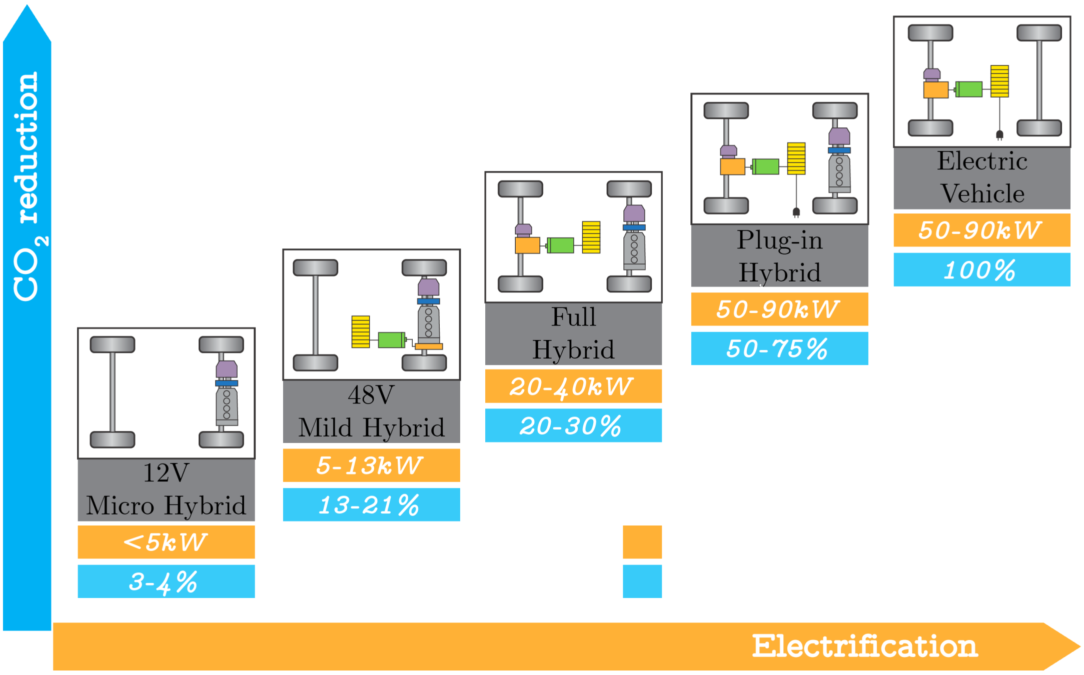

CO2 reduction
12V
Micro Hybrid
<5kW
3-4%
48V
Mild Hybrid
5-13kW
13-21%
Full
Hybrid
20-40kW
20-30%
Plug-in
Hybrid
50-90kW
50-75%
Electric
Vehicle
50-90kW
100%
Electrification

Figure 1.1: Overview of the Hybrid Electric Vehicle depending on Hybridization and CO_2 reduction

Once the typology has been defined, it is possible to describe how the energy flow is transferred from the energy storage (tank for ICE or battery for the EMs) to the wheels. Three paths are possible:

- **Parallel** the engine is the main power source while the electric motor provides

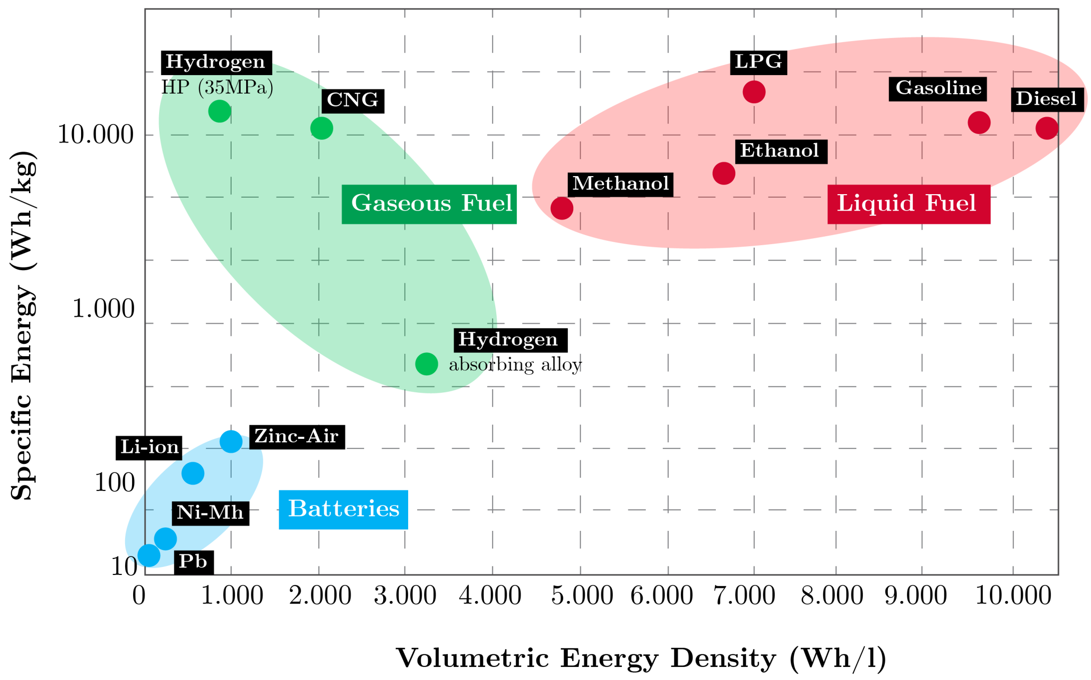


Figure 1.2: Fuel specific energy in function of their volumetric density

assistance as needed, delivering torque from zero rpm during standing starts and acceleration. This cooperation consent to avoid engine working points where the specific fuel consumption is high. The powertrain can be adapted simply by adding an electric motor and batteries to an existing vehicle, as in Fig. 1.3.

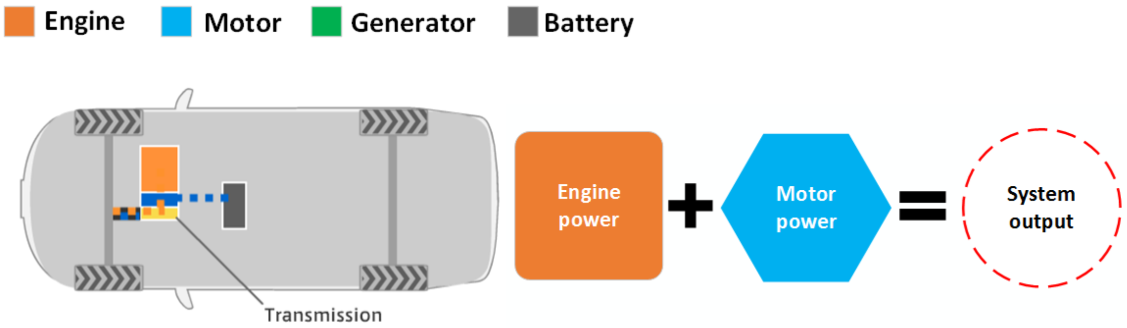

Engine
Motor
Generator
Battery
Transmission
Engine power
Motor power
System output

Figure 1.3: Parallel hybrid power characteristics

- **Series/parallel** combined hybrid vehicles share characteristics of series and parallel layouts. In particular, the EM powers the vehicle from a standing start and at low speed whereas, as the speed increases, ICE and EM work together to efficiently provide the power required. As can be expected, the system is more complex featuring a power split device and a generator. An exemplification is shown in Fig. 1.4.

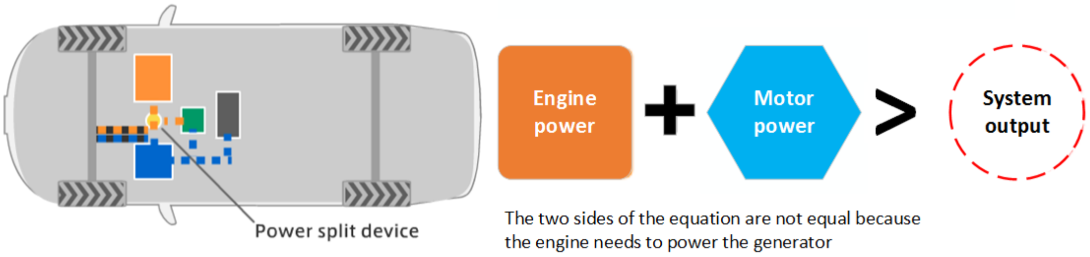

Power split device
Engine power
+
Motor power
>
System output
The two sides of the equation are not equal because the engine needs to power the generator

Figure 1.4: Series/Parallel hybrid power characteristics

- **Series** the series layout provides torque solely by using electric motors, like electric vehicles, and the aim of ICE is to recharge the battery with the generator. The powertrain is equivalent to an EVs, but because the vehicle also includes an engine, it is considered a hybrid (Fig. 1.5) [8].

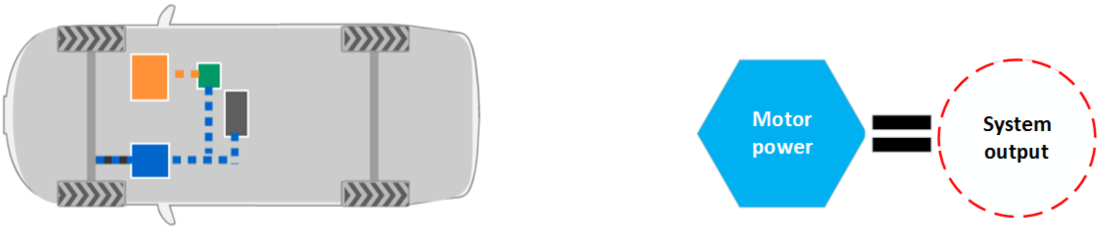

Motor power
=
System output

Figure 1.5: Series hybrid power characteristics

For what concern the HEVs parallel topology, several architectures are possible differing from each other for the position of the electric machines within the driveline. As shown in Fig. 1.6, they are as follow:

- **P0** the engine is coupled to the motor through a belt, so the electric machines is called Belt-driven Starter Generator (BSG);

- **P1** the EM is directly mounted on the crankshaft, upstream of the clutch, and it is named Integrated Starter Generator (ISG);
- **P2** the EM is separated from the engine by a clutch, that allows the pure electric drive;
- **P3** the EM is mounted on the secondary shaft out of the gearbox;
- **P4** the EM is connected to the front or rear wheels by means of a transmission ratio;

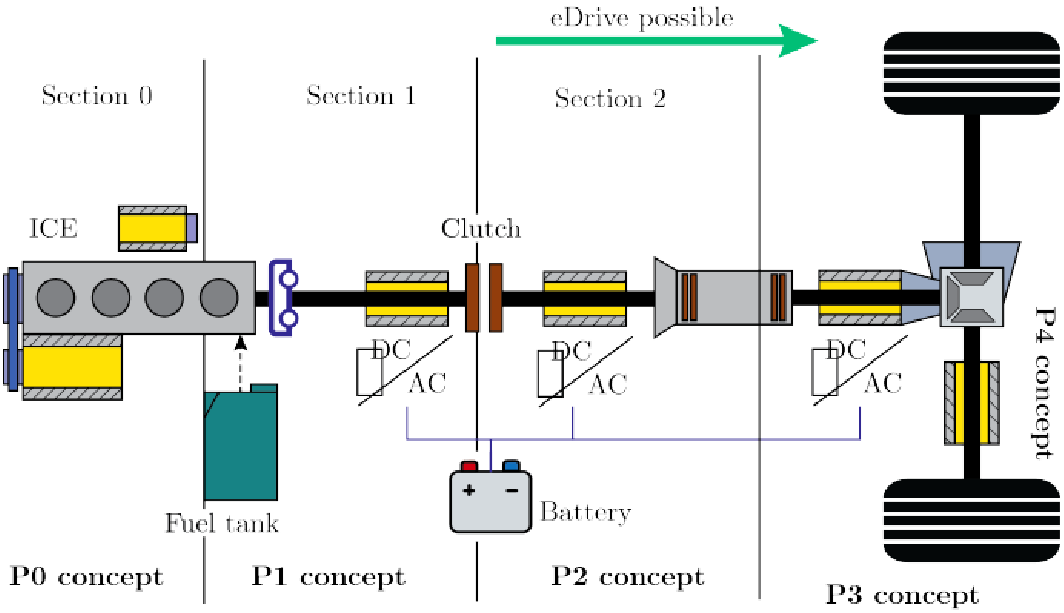

eDrive possible
Section 0
Section 1
Section 2
ICE
Clutch
DC
AC
DC
AC
DC
AC
Fuel tank
Battery
+
-
P0 concept
P1 concept
P2 concept
P3 concept
P4 concept

Figure 1.6: Parallel hybrid driveline architecture

Introducing a different type of energy flow (electrical energy) additional to the chemical one, engineers have to face new challenging problems. In fact, while the available space remains the same, the components rise in number: one or more electric motors, a bigger battery, a more powerful control unit and the inverters have to be rationally placed inside the vehicle. Adding new components doesn't imply only a different spacing configuration but it also means a more complex control at system level and also regarding the safety. On one hand, it's possible to achieve similar performance to standard vehicle with internal combustion engine while greatly improving fuel efficiency and tailpipe emission, recovering the energy from braking. On the other hand the torque split (so how the torque request is fulfilled) becomes the new control variable and it is complicated to handle. The challenge is to find the more efficient split that covers the torque request among the possible solutions. As a matter of fact, the computational effort of the control unit becomes heavier. Finding the optimal or sub-optimal solution is a part of the so called Energy Management Strategy which tries to minimize a two-variables function, where the fuel consumption is no longer the only parameter to keep under observation, but it's flanked by a new one: the state of charge of the Battery Storage System, shortened SoC. The state of charge represents the actual capacity of the battery over its maximum capacity and it's expressed in percentage. To better understand its meaning, it could be compared to the physical level of the liquid fuel in the tank. So then, the OEMs have invested money and time to develop new energy optimization strategies with the purpose of minimizing the overall energy consumption.

1.2 ADAS

The general tendency is moving toward a vehicle efficient and clean, but a non-negligible limit to that goal it will always be the driver, the less predictable variable in the system. The innovative Advance Driver-Assistance Systems (ADAS) come to help limiting the driver actions but they require a reliable detection of the vehicle and the surrounding environment. That virtual reconstruction permits the Hybrid Control Unit (HCU) to make more efficient choices both regarding the road safety as well the torque management. The new generation of on-board sensors and control strategies assist a common driver during acceleration and braking and they help him avoiding inefficient decisions such as during the gear shift, stop & start system, increasing the driving comfort and safety. A short explanation of that kind of equipment is given below, and it is shown in Fig. 1.7.

- **LIDAR** (LIght Detection And Ranging)
  remote sensing method that uses light in the form of a pulsed laser to measure ranges (variable distances) [20].

- **RADAR** (RAdio Detection And Ranging) detection system that uses radio waves to determine the range, angle, or velocity of objects. In particular it is distinguished in long range (LRR) for Adaptive Cruise Control, medium range (MRR)for cross traffic alert and lane change assist, short-range (SRR) for parking aid, obstacle/pedestrian detection [12].

- **CAMERAS**
  a video sensor used to perceive the environment around the vehicle.

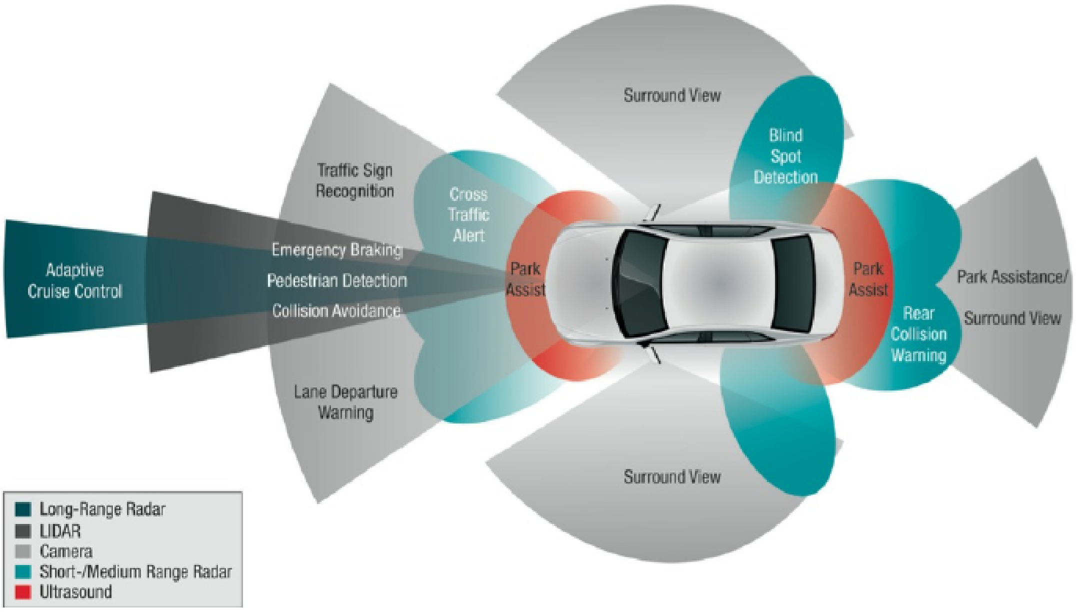
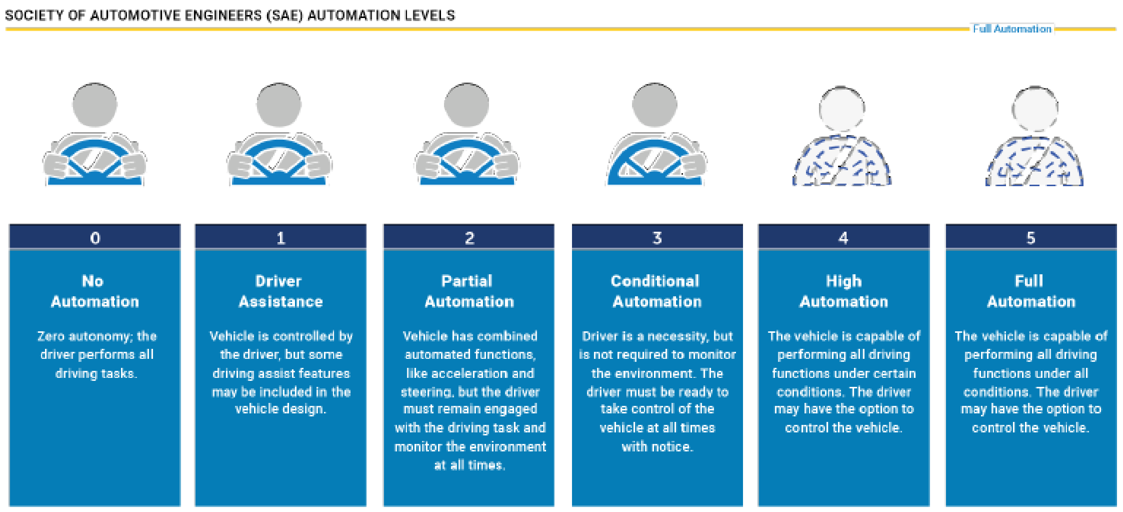

Surround View
Blind
Spot
Detection
Traffic Sign
Recognition
Cross
Traffic
Alert
Adaptive
Cruise Control
Emergency Braking
Pedestrian Detection
Collision Avoidance
Park
Assist
Park
Assist
Park Assistance/
Surround View
Rear
Collision
Warning
Lane Departure
Warning
Surround View
Long-Range Radar
LIDAR
Camera
Short-/Medium Range Radar
Ultrasound

Figure 1.7: On-board sensing equipment

All these efforts are made with the aim of designing and producing an autonomous vehicle capable of driving safely and efficiently on the road. This, from one hand will erase or at least reduce mortal accidents, and on the other hand will gave to the people a less polluting way of transportation. Obviously, to reach that goal, some gradual steps have to be fulfilled. The Society of Automotive Engineers (SAE) has defined different automation levels, which span from Level 0, without automation systems, to Level 5, where the car is completely self-driving [14]. In Fig.1.8 there is represented a schematic description of each level. In the future, the ADAS will intervene in the driving process more intensively and autonomously, for example influencing braking and steering maneuvers (with traffic jam chaffeur or motorway autopilot functionalities) [9].



0 **No Automation** Zero autonomy; the driver performs all driving tasks.

1 **Driver Assistance** Vehicle is controlled by the driver, but some driving assist features may be included in the vehicle design.

2 **Partial Automation** Vehicle has combined automated functions, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times.

3 **Conditional Automation** Driver is a necessity, but is not required to monitor the environment. The driver must be ready to take control of the vehicle at all times with notice.

4 **High Automation** The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.

5 **Full Automation** The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle.

Figure 1.8: Automation levels according to SAE

1.3 Connectivity

The way toward the autonomous driving is strictly linked with the progresses made by the telecommunication industries, in a on-going development of the Vehicle-to-Everything (V2X) connectivity technologies, with the aim of passing information from a vehicle to any entity that may affect the vehicle and vice versa:

- Vehicle-to-Vehicle (V2V)
- Vehicle-to-Infrastructure (V2I)
- Vehicle-to-Cloud (V2C)
- Vehicle-to-Pedestrian (V2P)

The amount of information work together in order to achieve road safety, traffic efficiency and energy savings. Some of the functionality given by the connectivity could influence the vehicle as a warning (for forward collisions, lane change and blind spots) or directly acting on it. Not only they will be at the base for the optimal safety and effective autonomous driving of tomorrow, but they constitute a powerful instrument that can already be used to fill the gap introduced by the human factor. The focus of this essay, for example, is on a PHEV vehicle made connected (enabled to the V2X communication) so that it is able to exchange data with a Map Service Provider (MSP) and to use this information to forecast the driving cycle performed by the human driver. Then, this prediction named eHorizon, can be used not only by the ego-vehicle for the energy management, but it can also be shared with other users to actuate strategies of intelligent mobility.

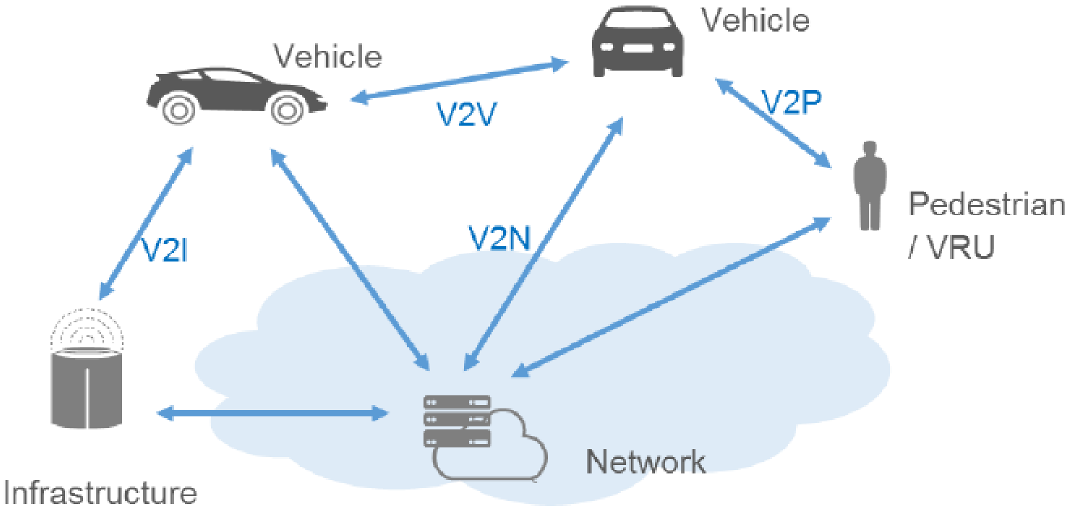


Figure 1.9: V2X connectivity

1.3.1 Predictive driving (eHorizon)

In term of efficiency, the optimal solution will be reached with the complete knowledge of the future, because it would permit to elaborate a strategy suitable for the specific route. In the following paragraph there is a very didactic example, but also very clear, of how the predictive drive will help: Fig. 1.10 shows a possible path of a commercial vehicle that approaches a climb. In normal conditions, climbing the hill, the internal combustion engine will provide the requested torque, but doing so it recharges, or doesn't discharge the battery. Once the car reaches the top of the climb, there is a downhill in front of it where it could perform regenerative breaking but the battery capacity could be already at the maximum limit. So the potential energy of the slope is useless. But if the presence of this difference in height were known, the control unit will allow the EMs to help the ICE during the climb, making it working at more efficient points, and at the top of the hill the battery could be recharged. This is only a little example of the enormous potential of a predictive strategy, in fact, it will be possible to avoid traffic congestions, accidents, dangerous situations and so on.

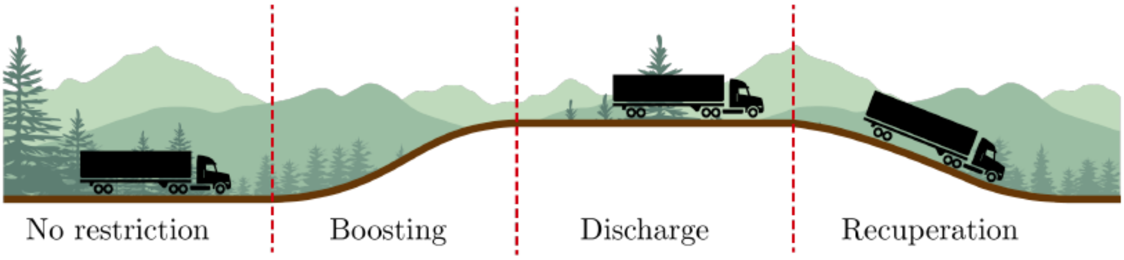


Figure 1.10: Ideal predictive strategy

This master thesis can be considered as a branch of a wider previous project developed by Green Mobility Research Laboratory (GMRL) [3] regarding predictive functions for energy management. The aim of the study was the development of these functions followed by their implementation in a parallel PHEV model (MiL) by means of Simulink [4]. In the first stage, torque management strategies have been implemented in the Control Unit in order to determine the optimal *(Discrete Dynamic Programming)* (DDP) or sub-optimal *(Equivalent Consumption Minimization Strategy)* (ECMS) torque split, as the solution of a cost function minimization problem. In this case, the cost function is the equivalent consumption of fuel (both chemical and electric). In parallel, two specific Predictive Functions (PF) have been implemented: the *Zero Emissions Zone* (ZEZ) and the *Predictive Thermal Management* (PTM)[5]. Both the functions are based on the knowledge of the driving cycle at priori. In particular, they are focused on the urban zone where the usage of the conventional engine is forbidden, so the vehicle has to switch to full electric mode. Firstly, the connectivity check if any part of the selected route belongs to a ZEZ and, in case of positive answer, the two functions are called. The Zero Emissions Zone function gives to the torque strategies a target to fulfill, in order to guarantee enough battery state of charge at the beginning of the event [fig.**??**]. On the other side, the Predictive Thermal Management decides if there is the possibility of managing the battery cooling circuit more efficiently and saving energy. In the end, the simulations results are made compliant with the newest regulations of the laboratory tests (Worldwide harmonized Light vehicle Test Cycle) and of the road tests (Real Driving Emissions).

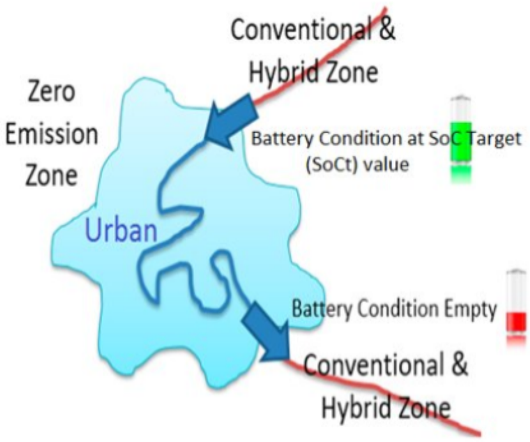

Conventional &
Hybrid Zone
Zero
Emission
Zone
Battery Condition at SoC Target
(SoCt) value
Urban
Battery Condition Empty
Conventional &
Hybrid Zone

Figure 1.11: Zero Emission Zone

Chapter 2

Simulation environment

2.1 Simulation environment

This chapter wants to provide an overview on the main steps that characterized the development of the Speed Profile Prediction algorithm (SPP), starting from the definition of the specifications and concluding with the implementation at the connected HiL. In addition, a brief description of the various adopted software/hardware architectures is presented.
It is important to stress how the whole procedure has been driven by the V-Model methodology of software development. In fact, while on the one hand the usage of connected vehicles represents a solution to crucial issues such as pollution and congested urban centers, on the other hand, the development of such a complex framework collides with the common goal of shortening the vehicle time-to-market. The old consolidated methodologies (reliable road tests above all) does not fit anymore because the safety validation of a current driver assistance system alone requires up to 2 millions test kilometers. In addition, most of the situations under control regard dangerous and highly-variable test conditions. Thus, the tendency is to move all the driven kilometers from the roads to advanced simulation environment in a process called "from-road-to-rig-to-desktop"[3].

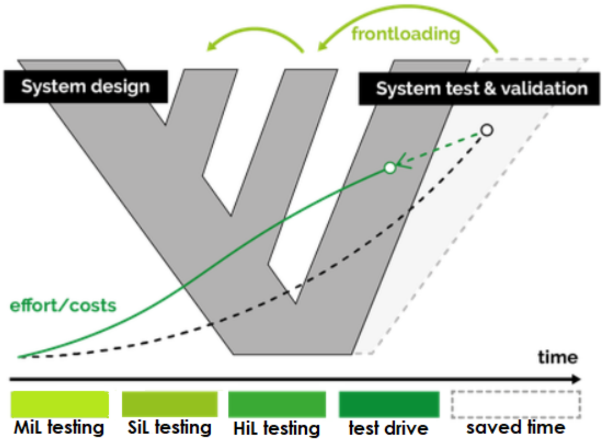


Figure 2.1: V-Model methodology layout

2.2 Model-in-the-Loop

The project started with the qualitative outlining of the information that could have been received from the MSP. In parallel, a bibliographic research has been conducted in order to define the state of the art and to assume adequate specifications as target for the logics to be implemented. Once the specifications have been clarified, the code writing phase has begun. The V-Model expects a testing procedure for each development step of the algorithm, that's why the first version of the algorithm has been written as a script in MATLAB[1]: this solution allows a quick implementation of the basic logics followed by a preliminary check of their congruence and validity. Once the algorithm has passed this first test, the second step was the conversion of the script into a Simulink[2] model by means of a Stateflow[3] chart. The need to switch into Simulink environment stems from the fact that, this algorithm is going to be flashed into a Simulink-programmable HCU. Until this moment, the Speed

---

[1] MATLAB (an abbreviation of "matrix laboratory") is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages[23].

[2] Simulink is a MATLAB-based graphical programming environment for modeling, simulating and analyzing multidomain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. Simulink is widely used for multidomain simulation and model-based design[24].

[3] Stateflow (developed by MathWorks) is a control logic tool used to model reactive systems via state machines and flow charts within a Simulink model[25].

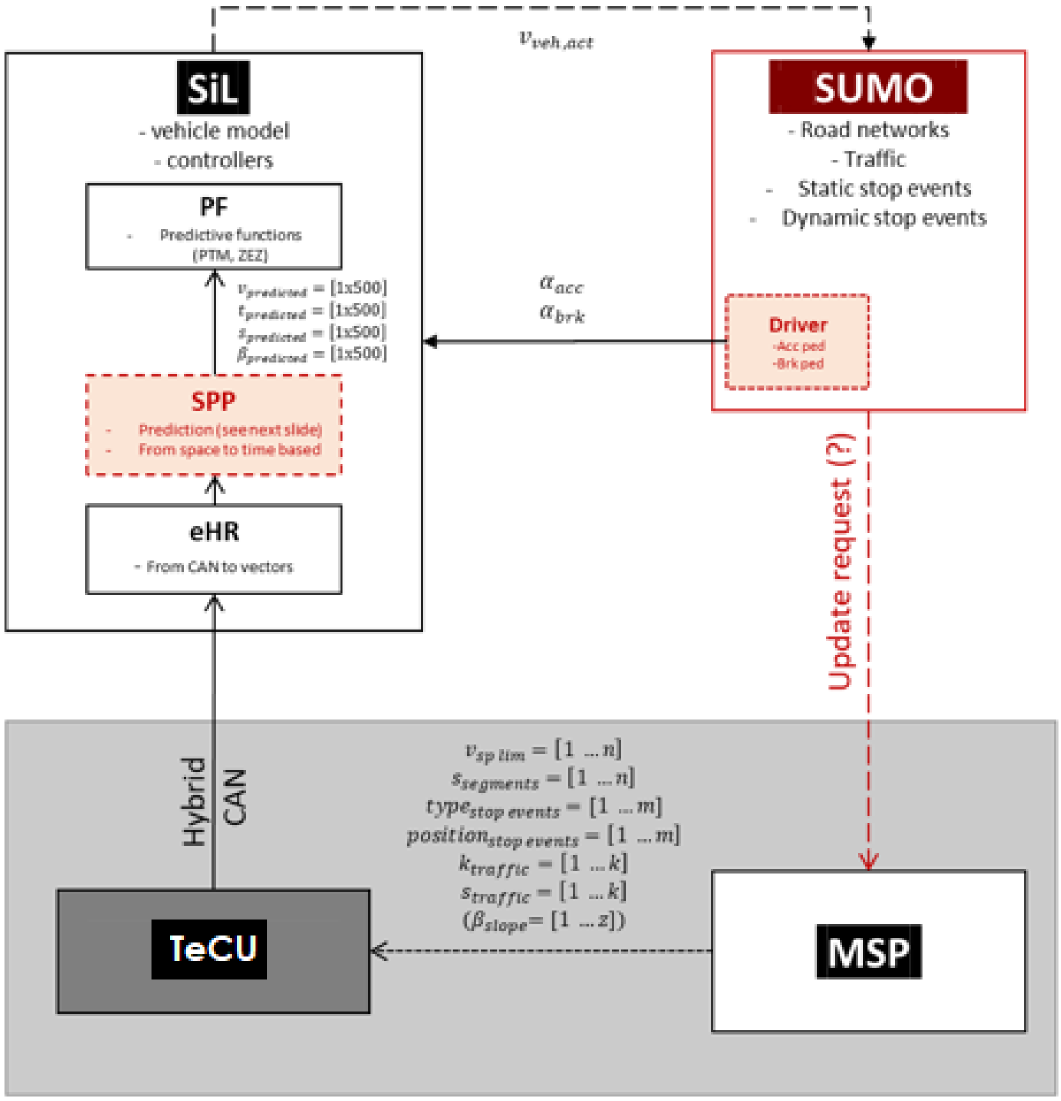
Profile Prediction Algorithm (SPP) as been treated as a stand alone model, without the slightest thought for the specifications imposed by the on-board implementation, but just considering his results in terms of prediction. Nevertheless, thanks to this transition to Simulink, the inclusion of the SPP in the Model-in-the-Loop has become possible. MiL testing and simulation is a technique used to abstract the behaviour of a system or sub-system in a way that this model can be used to test, simulate and verify one or more component while they are still under development. In other words, referring to this project, it means that also in this early phase, an off-line co-simulation of the SPP coupled with the PF has been already possible, even if compatibility between software and hardware in the HCU was still to be verified.

2.3 Software-in-the-Loop

Once the logic have been implemented in Simulink, one step beyond along the development process can be done passing from a Model-in-the-Loop to a Software-in-the-Loop simulation. SiL makes it possible to test software prior to the initialization of the hardware prototyping phase, significantly accelerating the development cycle. It represents an environment where the under analysis hardware component (HCU in this case) is replaced by its digital twin, containing exactly the same software as the real one. In this way, SiL enables the earliest detection of system-level defects or bugs, reducing the costs of later stage troubleshooting, when the number and complexity of component interactions is greater. A schematic view of SiL architecture employed in this project is shown in fig. 2.2: Here below, the steps of the data flow are briefly depicted:

- The virtual MSP sends navigation to the Telecommunication Control Unit (TeCU), in a format that is exactly the same as the one sent by the real MSP.

- TeCU elaborates navigation data in order to make them compliance with the CAN line and sends them to the eHorizon reconstructor (eHR) inside the HCU. In particular, the vectors containing navigation data are fragmented in vector of lower size.


$v_{veh,act}$
SiL
- vehicle model
- controllers
PF
- Predictive functions (PTM, ZEZ)
$v_{predicted}$ = [1x500]
$t_{predicted}$ = [1x500]
$s_{predicted}$ = [1x500]
$\beta_{predicted}$ = [1x500]
SPP
- Prediction (see next slide)
- From space to time based
eHR
- From CAN to vectors
SUMO
- Road networks
- Traffic
- Static stop events
- Dynamic stop events
α_{acc}
α_{brk}
Driver
-Acc ped
-Brk ped
Update request (?)
Hybrid CAN
$v_{sp\ lim}$ = [1 ... n]
$s_{segments}$ = [1 ... n]
$type_{stop\ events}$ = [1 ... m]
$position_{stop\ events}$ = [1 ... m]
$k_{traffic}$ = [1 ... k]
$s_{traffic}$ = [1 ... k]
(β_{slope}= [1 ... z])
TeCU
MSP

eHR – eHorizon Reconstructor; SPP – Speed Profile Prediction; MSP – Map Service Provider

Figure 2.2: Software-in-the-Loop scheme

- eHR reads data from CAN, re-builds the original information from fragments and generates input to feed the SPP.
- SPP generates the predicted speed profile to be sent as input to the PF.
- PF apply the optimization strategies to the predicted speed profile, computing

a SOC target and the thermal management strategy for the path. All the energies involved in calculations are retrieved by PF thanks to a Simulink model that reproduces the vehicle dynamics and its control system.

- SUMO (Simulator of Urban MObility) provides the virtual scenarios where PF are tested off-line. The need for SUMO to be introduced in the loop comes from the fact that in real driving, the predicted cycle will never perfectly match the real one. Simulating scenarios in SUMO makes already possible at this stage the evaluation of the response of the PF to this error, by assuming the cycle driven by SUMO's driver as the real one, avoiding costs and risks linked to road testing.

2.4 Connected Hardware-in-the-Loop

HiL, as stated by the name itself, provides more advanced tests on the control system, thanks to a detailed and already validated vehicle model implemented on a Real-Time PC that reproduce the physical signals of the environment. It includes sub-models for the vehicle longitudinal dynamics, the hybrid powertrain components, and their controllers. The Real-Time PC is connected to a rapid-prototyping HCU running an energy management supervisory controller where SSP and PF are included. Thus, while at the SiL it is possible to prevent failures due to programming incompatibility, the HiL testing allows in addition to avoid hardware-related problems such as overrunning or signal disturbances. Before the introduction of connectivity devices, HiL included only the hardware related to a conventional vehicle but, thanks to the work made by previous master thesis [7][11], it is possible today to execute on-line co-simulations, where the HiL receives real-time data by the MSP. Furthermore, to simulate at C-HiL that the vehicle under test is actually traveling along roads in a urban environment, the GPS signal is generated locally by a host PC in which SUMO is running.

Chapter 3

Speed Profile Prediction

3.1 State of the art

In the last four decades several studies about predicting the future speed of the ego-vehicle have been conducted since it is a necessary component of many Intelligent Transportation Systems (ITS) applications, in particular for safety and energy management systems. The speed prediction models state of the art can be grouped into two main categories: parametric and non-parametric. In the first category, we find algorithms where the driving task is modeled as a stimulus-response system, so, as a control problem where driver's goal is to keep a safe distance with the vehicle in front or to pursue a target speed according to some imposed constrains (e.g. speed limits). The nomenclature is justified by the fact that in this kind of algorithms a model for the ego-vehicle and/or for the driver needs to be included by means of parameters: an example of the most advanced parameters model can be found in the ones developed to be implemented in traffic simulators (e.g. SUMO). More recently some non-parametric models have been proposed: this kind of algorithms are based on probabilistic theories such as Artificial Neural Network, Markov chains or Monte Carlo methods. The main advantages related to this class consist in allowing a greater flexibility in the representation of the dynamics and in being suitable for machine learning and artificial intelligence systems. The counterpart is inherent in the probabilistic nature of these methods since they need time to acquire big data in order to generate consistent predictions. For a more detailed comparison please refer to the study conducted by Lefèvre et al. *Comparison of Parametric and Non-Parametric Approaches for Vehicle Speed Prediction* [16].

3.2 Previous version

A basic version of a speed profile prediction algorithm was previously implemented [7] giving a chance for the predictive functions to be tested at the connected-HiL. This trivial version was able to receive only the information about the legal speed limits by MSP and then, assuming the vehicle speed to be coincident with the speed limit value, the transitions between different speed values were modulated following a constant acceleration/deceleration law. In addition, stop events and relative duration were supposed (3.1). Although these predictions turned out to be inconsistent from an energetic point of view, this first version of the algorithm had allowed the telecommunication chain to be tested, marking the passage from HiL to a c-HiL, which was the main target at the time [7][11].

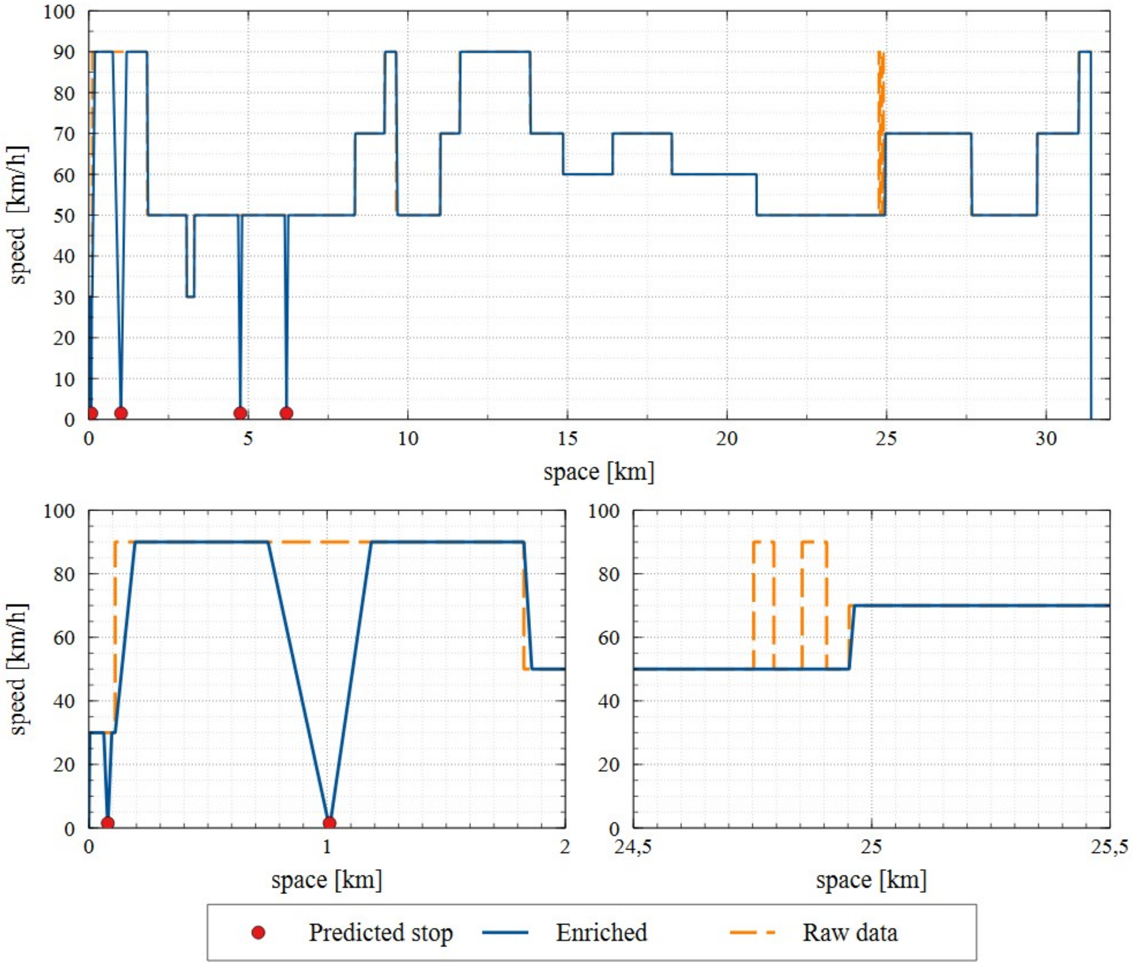


Figure 3.1: Previous version of the speed profile prediction

3.3 Speed Profile Prediction algorithm

This thesis work is based on the idea that the generation of a speed profile for energy management purposes might be a smart way to maximize the exploitation of navigation data that nowadays are commonly requested by the cars. Selecting the parametric approach has made possible to take this route in a relative short time and being free from the necessity to handle a large amount of previously recorded driving cycles. At first, when an enrichment target with respect to the previous version had to be defined, the algorithm described in [2] has been assumed as benchmark, despite of differences in the information supplied by the eHorizon provider. The logic at its base is to use routing information to feed a space based driver model who, segment-by-segment, will decide between accelerating, decelerating and holding the current speed in order to reach the maximum speed allowed by boundary conditions. In the following subsections it will be shown in details how the boundary conditions are retrieved starting from the input (eHorizon) with a deepening on the driver model. Here below, for the sake of clarity, the list of the input is provided:

Speed limits	values [1 x n] lengths [1 x n]
Traffic density	codes [1 x m] lengths [1 x m]
Stop events	type [1 x p] position [1 x p]

Table 3.1: SPP input

3.3.1 Traffic density

Naturally, the first limitation to the speed achievable by the driver along the path is enforced by speed limits. The information regarding speed limits is sent to the algorithm in the form of two equal sized vectors, one containing the speed limit values and the other containing the lengths of the relative range of validity. A furthermore limitation to the speed domain is imposed thanks to the information about the traffic density. This information consist likewise in two coupled equal sized vectors, respectively containing the traffic codes and their validity range lengths. A traffic code is a commonly way used by MSPs to describe the traffic density relative to a road segment by means of a colour. In this case, the colours provided by the MSP are four: starting from *green* in free flow traffic condition, coming to *black* in heavily congested traffic condition. The algorithm manages this information decreasing the speed limits by a factor, the code weight (CW), depending on the traffic code c, according to the fact that the more a street segment is congested, the more the segment average speed is low with respect to the speed value allowed by the limit. Thus, the maximum allowed speed (MAS) at the $j-$th road segment is defined by the 3.1 as:

$$MAS_j = V_{lim,j} \cdot CW_j(c) \tag{3.1}$$

Where $V_{lim,j}$ is the legal speed limit.

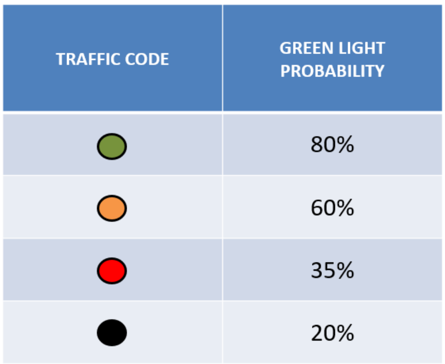
3.3.2 Stop events

In the end, there is the information for what concerns the stop events. It is commonly referred to a stop event as an event located along the path, whose presence implies that the speed of the vehicle in that position must be partially (e.g. bump) or totally (e.g. red traffic light) decreased. Just like the previous ones, this information is split in two vectors containing the typology of the stop events and their position. At the moment, four stop event typologies can be recognized by the algorithm, but, thanks to its modular architecture, the implementation of new typologies is straightforward. This is just an example of how the modular architecture makes the algorithm able to easily keep pace with the advisable improvement of the available information over time. The current list of stop events typologies is composed by:

Static	Stop signs Bumps
Dynamic	Traffic lights Roundabouts (yield signs)

Table 3.2: Stop event typologies

Stop events can be grouped into two main families: static and dynamic. Bumps and stop signals belongs to the static one, since, once their position is known, the ego-vehicle speed can be certainly assumed to be equal to zero (stop signals) or equal to a reference value (bumps). Prediction becomes harder in case of dynamic stop events, such as traffic lights or roundabouts. The term "dynamic" is due to the fact that they may not affect the current speed even if their position is known: to figure it out, it is sufficient to think about a driver approaching a green traffic light keeping the speed constant. To consider all the dynamic stop events taking place would mean making the predictive driving cycle far-fetched, introducing an error that becomes greater when free-flow traffic conditions are verified. On the opposite, neglecting the dynamic stop events at all, the predicted driving cycle would be still implausible, worsening in case of congested traffic conditions. For this reason, the necessity to take into account a *stop-over probability* has come to light. In real world, traffic lights' timing is influenced by several static and real-time variables, such as: time, traffic density, road type, presence of special vehicles (i.e. emergency or public transport). As a first approximation, the stop-over probability has been modeled by means of binomial probability, where the chances of green light are related to the traffic code assigned to the segment of interest [3.2]. This choice is supported by the fact that the traffic code is influenced by most of the mentioned variables. Thus, when green light is drawn, the evaluation of the speed in that position is performed without caring about the presence of the stop event, otherwise, speed is imposed to be equal to zero. The choice to not introduce the yellow light comes from the fact that, according to traffic laws, it can affects driver's decision in two ways, both considered by the SPP inside red and green light cases. In fact, a yellow light warns you that the red signal is about to appear. When you see the yellow light, you should stop, if you can do so safely (red light), otherwise, it is allowed to leave the intersection (green light). When the presence of a roundabout is detected, the algorithm acts in a similar way, drawing a symbolic green (which stands for "nobody is coming") or red light, with the only difference that in case of green light the speed is imposed to be $\leq 30\,\text{km/h}$ as attempt to simulate the behavior of a real driver approaching to a roundabout.



TRAFFIC CODE	GREEN LIGHT PROBABILITY
🟢	80%
🟠	60%
🔴	35%
⚫	20%

Figure 3.2: Green traffic light probability depending on traffic code.

3.3.3 Space based driver model

Usually, the target for the average driver during his routine is to arrive to destination in the shortest time possible, respecting the law and avoiding accidents. Assuming this as fundamental rule, the driver model has been conceived for chasing the MAS, evaluating at every discretization space interval the opportunity to accelerate. At first, the path is divided by J segments whose endpoints are named *nodes*. Nodes are points where boundary conditions are imposed, since they represent coordinates where changes in traffic code and/or speed limit take place, or where the stop events are located. Then, each $j-$th segment is ulteriorly divided by N points where the speed will be evaluated point-by-point. To understand how it works, let's suppose the driver proceeding at the speed V_i on the generic $i-$th point along the $j-$th segment. First of all, the algorithm looks at the current speed V_i and verifies one out of two conditions:

1. $V_i = MAS$
2. $V_i < MAS$

If condition 1 is verified, the driver will decide whether to keep the speed constant or to start braking in order to match a lower speed imposed by the closest stop event or speed limit change.
Otherwise, the driver also considers whether to accelerate and two variables are computed:

- V: the speed at the point *i+1* assuming that the acceleration is performed;
- Q: it represents the maximum speed at the point *i+1* that still allows you to brake in time to match the speed imposed by the next stop event or speed limit change.

Thus, the speed at the next point is imposed by means of three double `if` conditions:

1. `if` $V \leq Q$ `&&` $V \leq MAS \Rightarrow V_{i+1} = V$;
2. `if` $V \leq Q$ `&&` $V \geq MAS \Rightarrow V_{i+1} = MAS$;
3. `if` $V \geq Q$ `&&` $V \leq MAS \Rightarrow V_{i+1} = Q$.

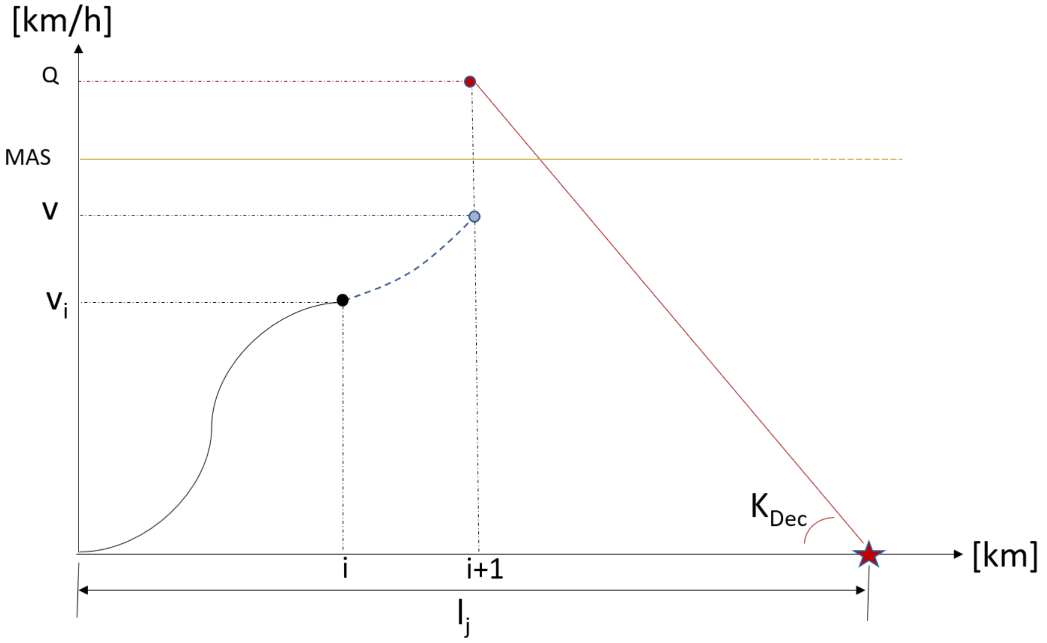


Figure 3.3: Verified conditions for acceleration.

Q and V values are calculated in accordance with braking and acceleration laws that takes into account the ego-vehicle model thanks to two parameters: *Maximum Longitudinal Acceleration (MaxLongAcc)* and *Maximum Longitudinal Deceleration (MaxLongDec)*. Before going into deep with this topic, it is appropriate to cite the fact that the algorithm allows you to choose one out of three different driver types (*quite, normal, aggressive*), since both the acceleration and deceleration laws are affected by this choice. This feature has been inspired by the fact that, the energy consumption related to a given route can be largely influenced by driver's behaviour. The braking phases, like in the previous version of the algorithm, are modeled my means of a constant deceleration law. In this case, a constant K_{dec} has been introduced and its value, ranging between 0 and 1, depends on the selected driver type: it represents the percentage of *MaxLongDec* used by the driver.
For what concern accelerations, an exponential law has been introduced instead of the constant acceleration of the previous version. In this way, longitudinal accelerations have been modeled as a first order system undergoing to a step stimulus,

described by the equation [3.2]:

$$a(x) = K_{acc}(1 - e^{-(\tau x)}) \tag{3.2}$$

Both parameters in the equation K_{acc} and τ depend on the selected driver type. Like for the deceleration, K_{acc} determines the percentage of ego-vehicle's *MaxLongAcc* requested by the driver or, in other words, the step stimulus value. In this kind of systems, the time constant τ defines the time employed to achieve the step value [fig.3.4]: the lower is τ, the slower is the transient. The purpose of this approach to the modeling of the acceleration manoeuvre is to consider different ways to act on the pedal that are specific to different drivers.

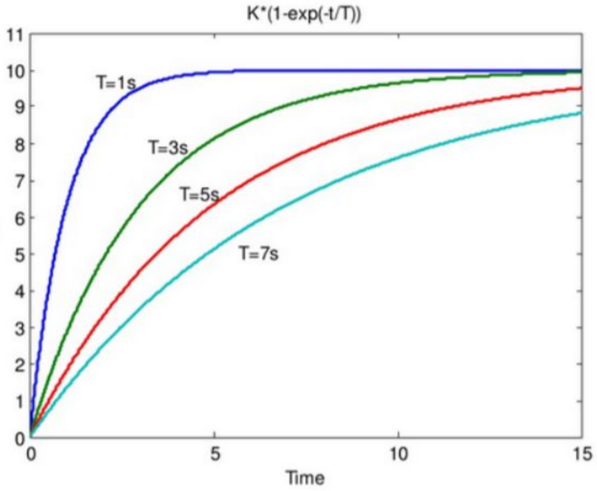


Figure 3.4: Effects of different $\tau = \frac{1}{T}$ in a first order system undergoing to a step stimulus.

3.3.4 Oscillation around constant speed segments

At this point, in accordance with all is said above, there would be a virtual driver who would keep the speed perfectly constant at the maximum allowed value for as long as possible. As could be easily imagined, this behaviour is far from reality [fig.3.5], since the velocity often exhibits oscillations around the speed limit due to inharmonic traffic flow. Furthermore, different drivers are expected to react in different ways to traffic variation: for instance, aggressive drivers tends to delay as much as possible braking until they are very close to the vehicle beyond. To account

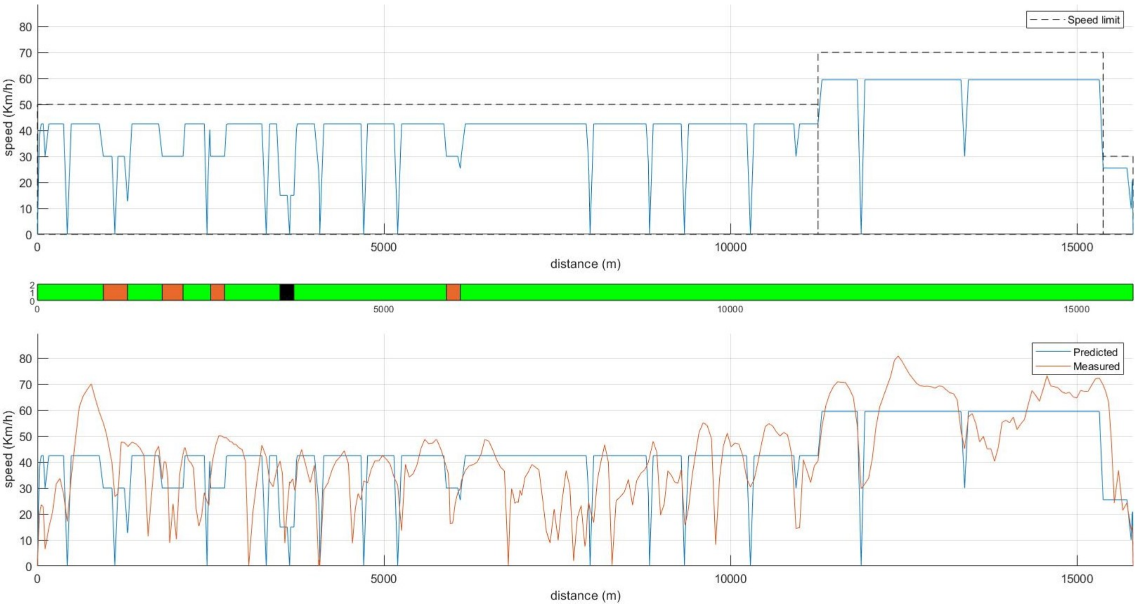


Figure 3.5: Measured speed profile Vs. "zero-noise" predicted speed profile

for these oscillation, the sum of l-cosines characterized by different amplitude A_r and frequencies ω_r is added to the MAS, so that the reference signal followed by the driver is no more constant but governed by the equation:

$$MAS_{noise} = MAS + \sum_{r=1}^{l} A_r cos(2\pi f_r x) \tag{3.3}$$

All the parameters in the [3.3] are functions of the selected driver type and of the traffic code. For further explanations on how they have been determined please refer to the chapter *Calibration and Validation*. The oscillation around constant speed segments constitutes the last space-based enrichment with respect to the previous version of the algorithm and an example of the final output is shown in fig.3.6:

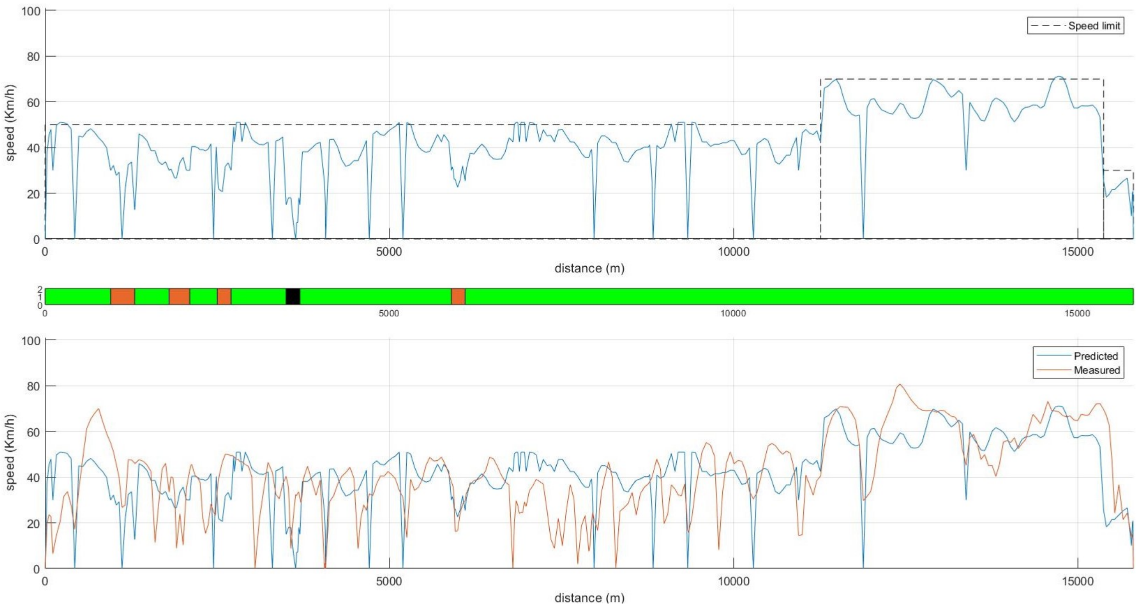


Figure 3.6: Measured speed profile Vs. Predicted speed profile

Chapter 4

Calibration procedure

4.1 Methodology

This chapter represents an overview on how possible values for the parameters which mostly affect the prediction have been determined. The whole procedure is based on the acquisition of real data by an on-board recording across predefined scenarios. In particular, this section wants to provide a detailed description of the methodologies adopted during the measurements phase. Let it be clear that a rigorous calibration requires an amount of time and resources that are beyond the availability of this thesis work. Nevertheless, the procedures described in this chapter have allowed to supply the algorithm with real-based foundations and can be considered an effective starting point for future works.

4.1.1 Scenarios

Two scenarios have been introduced so that the first could be used to tune the phases of acceleration and deceleration, while the second to evaluate the influence of the environment in terms of traffic, speed limits and stop events:

1. This scenario consists of an ad hoc maneuver where the driver performs a 0-50 km/h acceleration followed by a 50-0 km/h deceleration on a straight road. The range of speed has been assumed considering the use-case this algorithm has been developed for: the ZEZ is a urban contest where the most of the

maneuvers are included in this interval. Looking at fig4.1, it should be notice how the speed of 50 km/h is not achieved by the real driver: that is because of car OEMs are used to program cockpits where the displayed speed is slightly higher than the actual one.

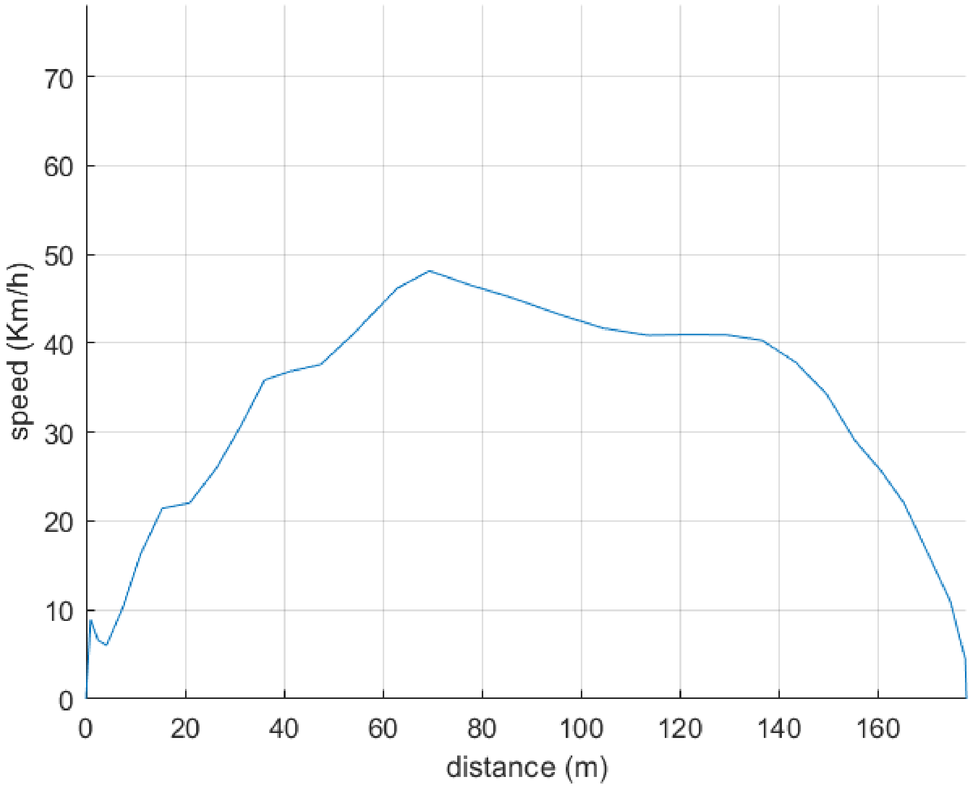


Figure 4.1: 1st scenario: 0-50-0 km/h acceleration and deceleration

2. In this case, a route across the city center of Bologna has been identified to be driven at different time-slots. This route in particular has been chosen for its peculiarity to present different speed limits and a high variation of traffic condition during the day. Thus, selecting adequate time-slots, the effects of every traffic code can be investigated.

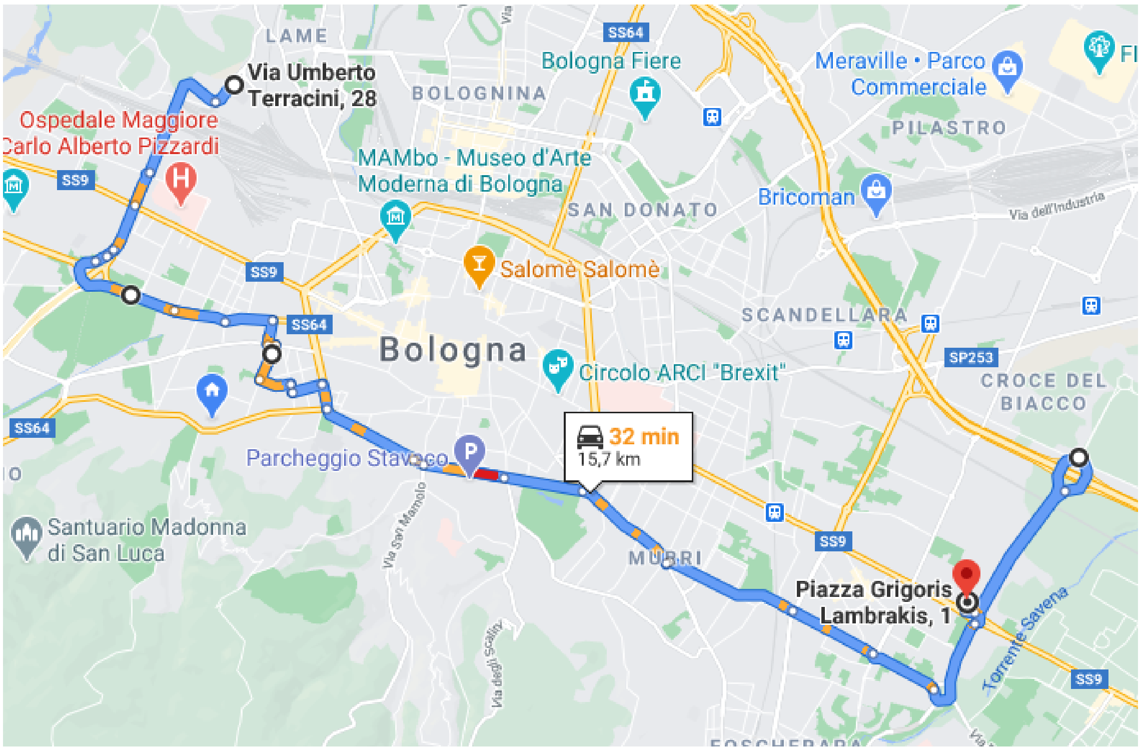

LAME
Via Umberto Terracini, 28
BOLOGNINA
SS64
Bologna Fiere
Meraville • Parco Commerciale
PILASTRO
Ospedale Maggiore Carlo Alberto Pizzardi
SS9
MAMbo - Museo d'Arte Moderna di Bologna
SAN DONATO
Bricoman
Via dell'Industria
Salomè Salomè
SCANDELLARA
Bologna
SS64
Circolo ARCI "Brexit"
SP253
CROCE DEL BIACCO
SS64
32 min
15,7 km
Parcheggio Staveco
Santuario Madonna di San Luca
Via San Mamolo
MURRI
SS9
Piazza Grigoris Lambrakis, 1
Torrente Savena
Via degli Scalini
SS9

Figure 4.2: 2nd scenario: from Via Umberto Terracini to Piazza Grigoris Lambrakis, Bologna

4.1.2 Measurements

The acquisition of the real speed profiles through the scenarios has been done by means of a GPS receiver (model: Navilock nl-8002u [19]) connected via USB to a laptop hosted in the passenger compartment. All the measures about vehicle speed and position are recorded with a 1 Hz sampling frequence and can be saved on PC thanks to the GNSS evaluation software developed by u-blox, named u-center. The Graphical User Interface (GUI) of the software offers the possibility for the users to select the variables of interest out of a long list and use them to feed real-time charts, like in fig. 4.4. Furthermore, data can also be grouped into tabs so that they can be easily copied into an Excel data sheet or into a Matlab workspace [fig. 4.5].

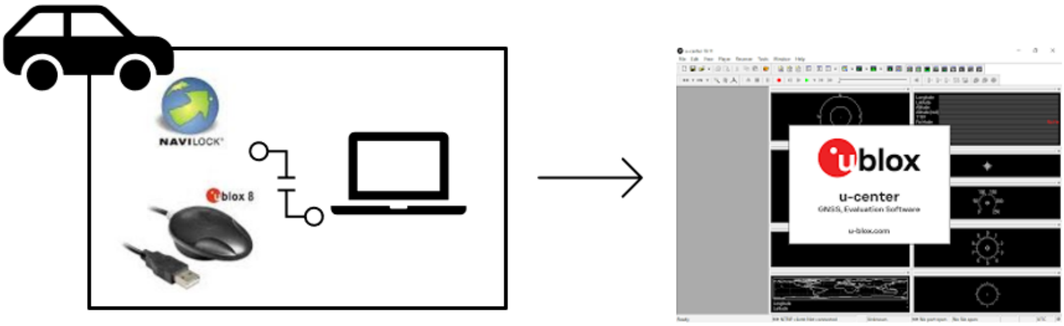

NAVILOCK
blox 8
blox
u-center
GNSS Evaluation Software
u-blox.com

Figure 4.3: Measurement chain

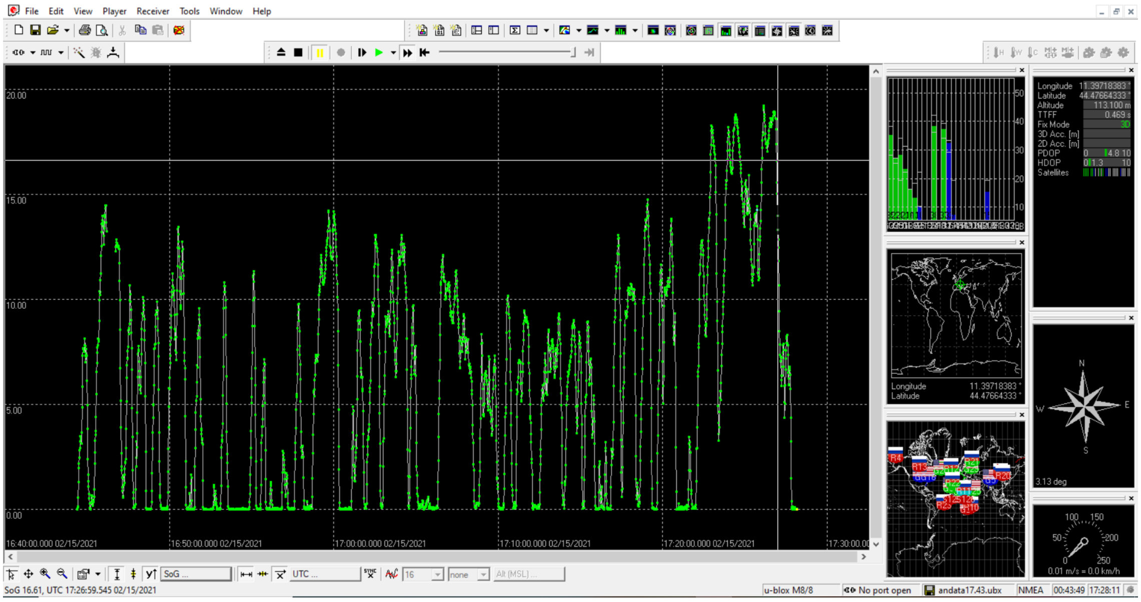
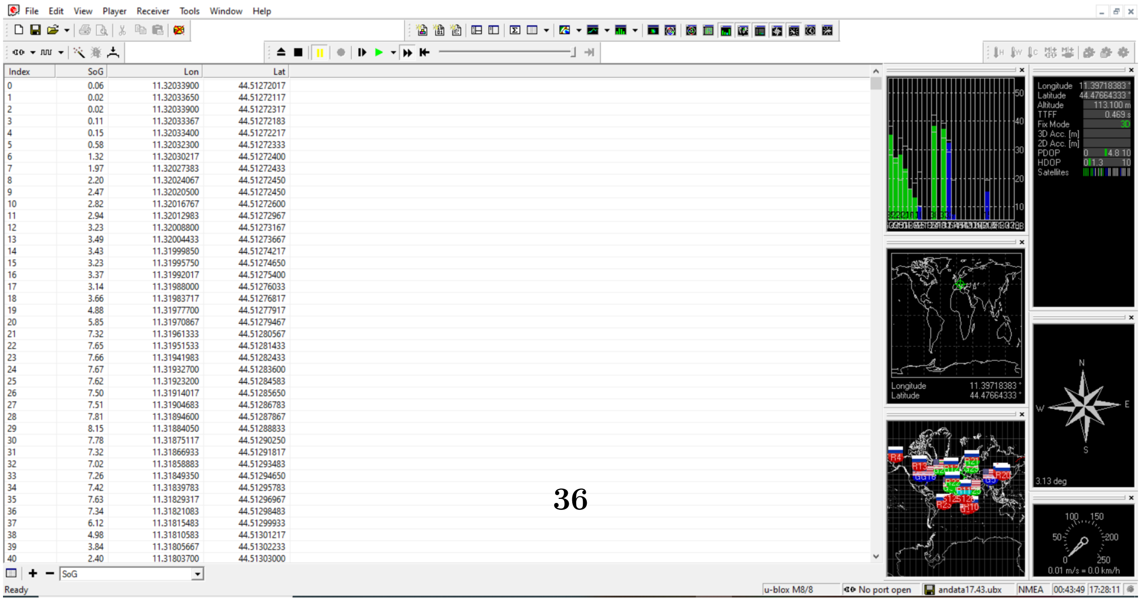

File
Edit
View
Player
Receiver
Tools
Window
Help
20.00
15.00
10.00
5.00
0.00
16:40:00.000 02/15/2021
16:50:00.000 02/15/2021
17:00:00.000 02/15/2021
17:10:00.000 02/15/2021
17:20:00.000 02/15/2021
17:30:00.00
SoG 16.61, UTC 17:26:59.545 02/15/2021
u-blox M8/8
No port open
andata17.43.ubx
NMEA
00:43:49
17:28:11
Longitude 11.39718383 °
Latitude 44.47664333 °
Altitude 113.100 m
TTFF 0.469 s
Fix Mode 3D
3D Acc. [m]
2D Acc. [m]
PDOP 14.8 10
HDOP 1.3 10
Satellites
N
E
S
W
3.13 deg
0.01 m/s = 0.0 km/h

Figure 4.4: u-center GUI: speed over time diagram view



Index	SoG	Lon	Lat
0	0.06	11.32033900	44.51272017
1	0.02	11.32033650	44.51272117
2	0.02	11.32033900	44.51272317
3	0.11	11.32033367	44.51272183
4	0.15	11.32033400	44.51272217
5	0.58	11.32032300	44.51272333
6	1.32	11.32030217	44.51272400
7	1.97	11.32027383	44.51272433
8	2.20	11.32024067	44.51272450
9	2.47	11.32020500	44.51272450
10	2.82	11.32016767	44.51272600
11	2.94	11.32012983	44.51272967
12	3.23	11.32008800	44.51273167
13	3.49	11.32004433	44.51273667
14	3.43	11.31999850	44.51274217
15	3.23	11.31995750	44.51274650
16	3.37	11.31992017	44.51275400
17	3.14	11.31988000	44.51276033
18	3.66	11.31983717	44.51276817
19	4.88	11.31977700	44.51277917
20	5.85	11.31970867	44.51279467
21	7.32	11.31961333	44.51280567
22	7.65	11.31951533	44.51281433
23	7.66	11.31941983	44.51282433
24	7.67	11.31932700	44.51283600
25	7.62	11.31923200	44.51284583
26	7.50	11.31914017	44.51285650
27	7.51	11.31904683	44.51286783
28	7.81	11.31894600	44.51287867
29	8.15	11.31884050	44.51288833
30	7.78	11.31875117	44.51290250
31	7.32	11.31866933	44.51291817
32	7.02	11.31858883	44.51293483
33	7.26	11.31849350	44.51294650
34	7.42	11.31839783	44.51295783
35	7.63	11.31829317	44.51296967
36	7.34	11.31821083	44.51298483
37	6.12	11.31815483	44.51299933
38	4.98	11.31810583	44.51301217
39	3.84	11.31805667	44.51302233
40	2.40	11.31803700	44.51303000

File
Edit
View
Player
Receiver
Tools
Window
Help
SoG
Ready
u-blox M8/8
No port open
andata17.43.ubx
NMEA
00:43:49
17:28:11
Longitude 11.39718383 °
Latitude 44.47664333 °
Altitude 113.100 m
TTFF 0.469 s
Fix Mode 3D
PDOP 14.8 10
HDOP 1.3 10
Satellites
N
E
S
W
3.13 deg
0.01 m/s = 0.0 km/h

Figure 4.5: u-center GUI: table view with time, speed, longitude and latitude

4.1.3 Key Performance Indicators

In order to evaluate the overall goodness of predictions as well as the way they are affected by single parameters, key performance indicators (KPI) have been defined. Naturally, since the output generated by the algorithm is a speed profile prediction, an evaluation of the error in terms of speed had to be done, but always minding the energy management purpose. For this reason, two class of indicators have been introduced:

- Speed KPI
- Energy KPI

The identification of speed KPI has been entrusted to a bibliography search in order to be aligned to the many in the past who have experimented the development of algorithms in this field [16][17][15]. This search revealed how the error in these kind of predictions are commonly evaluated by means of *Mean absolute error* (MAE) and *BIAS*, both expressed in km/h and defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |S_{p,i} - S_{r,i}| \tag{4.1}$$

$$BIAS = \frac{1}{n}\sum_{i=1}^{n} S_{p,i} - S_{r,i} \tag{4.2}$$

where $S_{p,i}$ and $S_{r,i}$ are respectively the predicted and the measured speed at point i and n is the total amount of point where the differences are calculated. MAE, not caring about the algebraic signs of the errors, expresses the mean distance between the prediction and real data. BIAS, instead, operates an algebraic sum so that self-compensation of the error can occur, resulting in possible low values of BIAS even in presence of high punctual errors. For this reason, BIAS on its own is not a reliable index to evaluate the goodness of the predictive model, but represents a powerful instrument to identify eventual issues related to a systematic under/overestimation of the speed.

Then, to define effective energetic indicators, [13] has been assumed as reference. In this document, SAE defines the parameters used to evaluate the goodness of tests where a reference cycle must be followed by a human operator acting manually on pedals. First of all, three force components are calculated for both the predicted and the measured cycle:

- **Road load** it represents the force required to win rolling resistance and drag force. It is expressed by the equation 4.3, where $F_0[N]$, $F_1[\frac{N}{km/h}]$, $F_2[\frac{N}{(km/h)^2}]$ are the cost down coefficients of the vehicle:

$$F_{RL} = F_0 + v \cdot F_1 + v^2 \cdot F_2 \tag{4.3}$$

- **Positive inertial force** it is associated only to positive values of acceleration (a^+) and represents the force required by the vehicle mass to be accelerated. It is expressed by the equation 4.4, where M is the vehicle mass:

$$F_{I^+} = M \cdot a^+ \tag{4.4}$$

- **Negative inertial force** it is associated only to negative values of acceleration (a^-) and represents the force required by the vehicle mass to be decelerated:

$$F_{I^-} = M \cdot a^- \tag{4.5}$$

Once the three forces have been defined, the relative energies can be retrieved by integrating the forces along the path length L, obtaining:

$$E_{RL} = \int_0^L F_{RL} dl \tag{4.6}$$

$$E_{I^+} = \int_0^L F_{I^+} dl \tag{4.7}$$

$$E_{I^-} = \int_0^L F_{I^-} dl \tag{4.8}$$

Now, three energy KPI can be introduced in form of energy rate, where subscripts

p and r are respectively referred to the predicted and measured (real) speed profile:

$$\%E_{RL} = \frac{E_{RL_p} - E_{RL_r}}{E_{RL_r}} \cdot 100 \tag{4.9}$$

$$\%E_{I^+} = \frac{E_{I_p^+} - E_{I_r^+}}{E_{I_r^+}} \cdot 100 \tag{4.10}$$

$$\%E_{I^-} = \frac{E_{I_p^-} - E_{I_r^-}}{E_{I_r^-}} \cdot 100 \tag{4.11}$$

While in case of an ICE vehicle differences in the negative inertial energy could be neglected, they have a huge impact in case of PHEV since they represent the amount of energy that can be potentially restored while braking. From this, the necessity for 4.11 to be introduced.

4.2 Calibration

After the acquisition of the real speed profiles, the data analysis could start. The target of the analysis was to tune the parameters related to acceleration, deceleration and traffic. It is important to consider that one series of these parameters needs to be determined for each driver type. Same approaches can be used for calibration in all the three cases, for example, by instructing the real driver on behaving differently during the measurements, or alternatively, recurring to three different real drivers properly recruited. In the next paragraphs, the procedures in case of driver type *normal* are shown for each parameter.

Acceleration	K_{acc}, τ
Deceleration	K_{dec}
Traffic	CW, A_r, f_r

Table 4.1: Calibration parameters

4.2.1 Acceleration and braking maneuvers

As anticipated in section 4.1.1, an ad hoc maneuver has been recorded as reference for the calibration of acceleration and braking. The approach has consisted in a trial and error sequence where different combinations of K_{acc}, τ and K_{dec} have been tested, in order to make the profile generated by SPP matching with the measured one. The grade of matching has been evaluated on the base of MAE, starting from $MAE = 6.37km/h$ to get to $MAE = 2.42km/h$ [fig. 4.6].

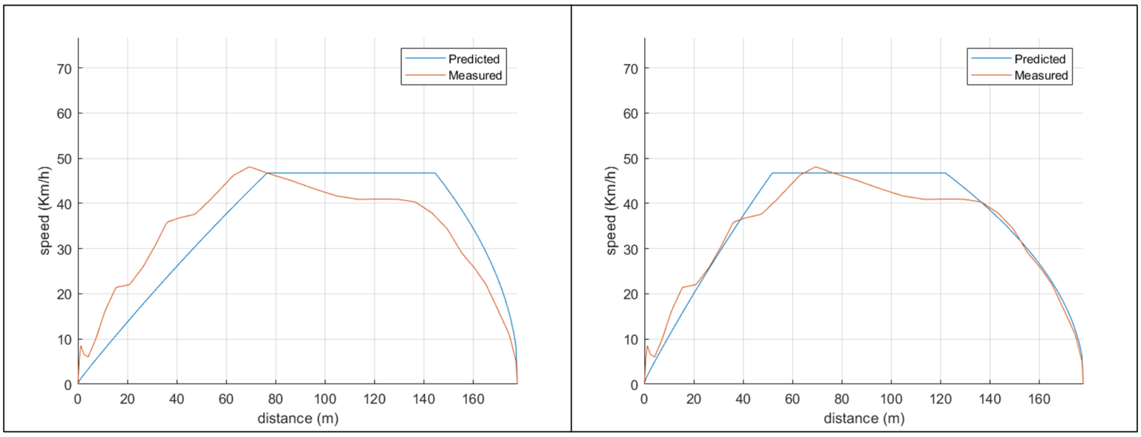


Figure 4.6: Acceleration and braking: before(left) and after(right) calibration

4.2.2 Traffic: influence on average speed and oscillations

All the parameters related to traffic have been retrieved starting from measured speed profiles by way of frequence analysis. At first, portions of signal coming from different measured speed profiles, have been grouped by traffic code. Then, the Fast Fourier Transform (FFT) has been applied in Matlab to each signal obtaining the distribution of the amplitudes over frequencies. Since the SPP generates a space-based speed profile, also the frequencies must be expressed in terms of space (1/m) to be used in the algorithm. For this reason, a pre-processing is required by the measured signals before the FFT can be applied. In particular, a re-sampling of the signal have to be performed to make it sampled every meter instead of every second. At this point, applying the FFT, distance based frequencies and relative modules are obtained as shown by the example in fig.4.7. As expected, the amplitudes of these

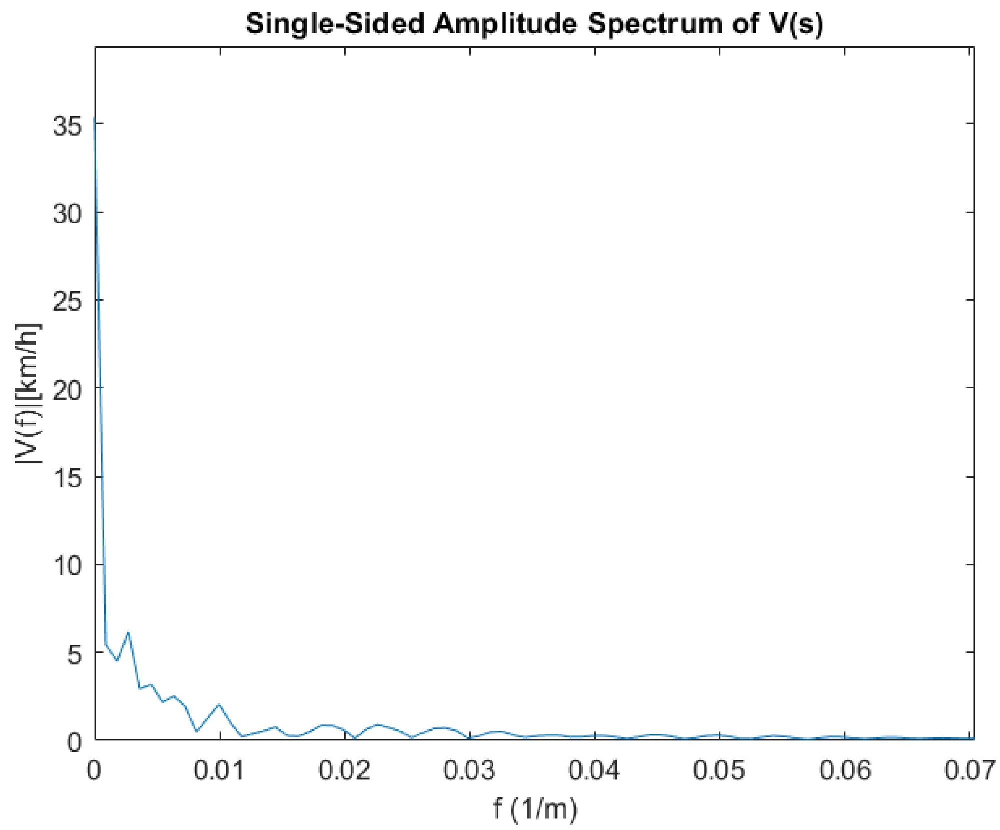


Figure 4.7: Example of FFT applied to a distance-based measured signal

signals are characterized by a continue distribution over the frequencies, since infinite frequencies are required to perfectly approximate a non-periodic signal. Despite of it, to be represented in Matlab as couple of vectors, a signal must be discretized and this allow to easily identify frequencies with a high value of magnitude. The first consideration has been made looking at the amplitudes at $f = 0$: since they represent the average speed along the analyzed segments, they have been used to assign a CW value to every traffic code. Let's assume:

- the FFT be performed on J signals corresponding to J segments denoted by the same traffic code c;
- l_j be the length of the j-th segment;
- $V_{lim,j}$ be the legal speed limit at the j-th segment;

- $A_{0,j}$ be the magnitude at $f = 0$ of the j-th segment;

The *CW* for the traffic code c has been retrieved as:

$$CW_c = \frac{\sum_{j=1}^{J} \frac{A_{0,j}}{V_{lim,j}} \cdot l_j}{\sum_{j=1}^{J} l_j} \tag{4.12}$$

Consequently, once all the CW have been defined, a constant MAS can be assigned to each segment, as defined in the equation 3.1. After the MAS has been assigned, r cosines of magnitudes A_r and frequencies f_r are added, as shown by the equation 3.3. This results in oscillation around the MAS value whenever it is reached by the driver. Here below, the procedure followed to retrieve A_r and f_r as functions of the traffic code c is reported. Let's assume:

- the FFT be performed on J signals corresponding to J segments denoted by the same traffic code c;
- four ranges of frequencies can be defined, denoted by r = 1,...,4;
- N_j be the number of magnitudes $A_{n,r,j}$ corresponding to the $f_{n,r,j}$ frequencies included in the range r, of the j-th signal;

Amplitudes A_r and frequencies f_r for the traffic code c have been retrieved as:

$$A_{r,c} = \frac{\sum_{j=1}^{J} \left(\frac{1}{N_j} \sum_{n=1}^{N_j} A_{n,r,j}\right) \cdot l_j}{\sum_{j=1}^{J} l_j} \tag{4.13}$$

$$f_{r,c} = \frac{\sum_{j=1}^{J} \left(\frac{1}{N_j} \sum_{n=1}^{N_j} f_{n,r,j}\right) \cdot l_j}{\sum_{j=1}^{J} l_j} \tag{4.14}$$

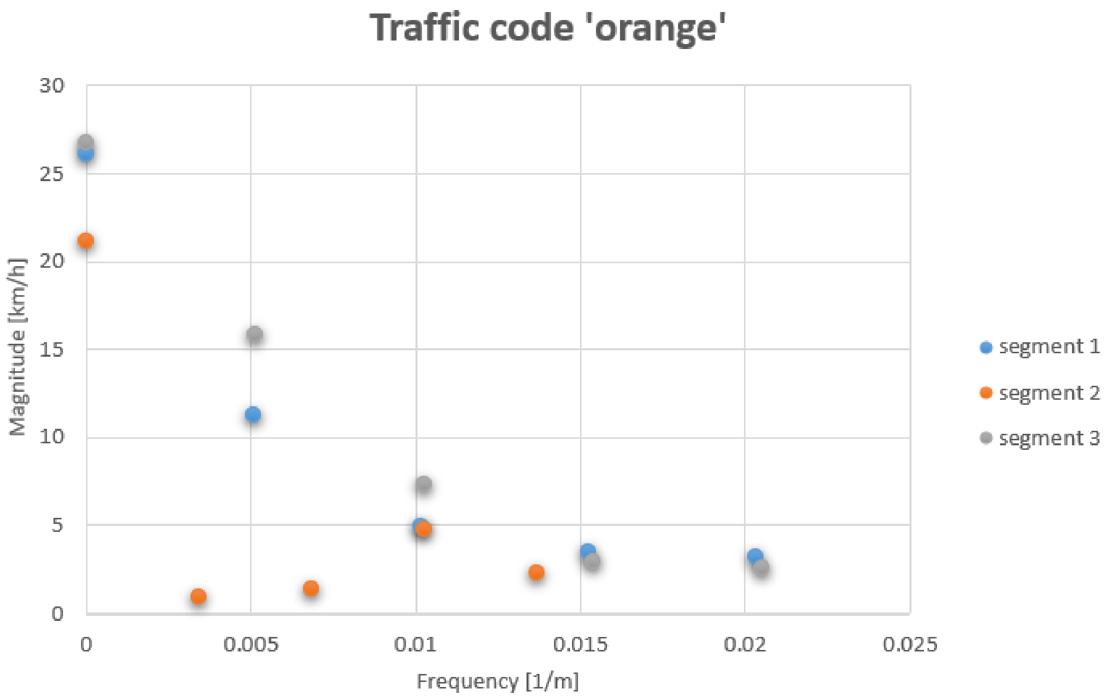


Figure 4.8: Frequency and magnitude's scatter in case of traffic code 'orange'

Chapter 5

Results and validation

Consequently to the calibration phase, the SPP has been applied to three different scenarios to get a first evaluation of the performances at this progress of development. In the next sections, the results for each cycle will be shown as a comparison between measured and predicted speed profile, expressed in terms of speed KPI and energy KPI.

5.1 Scenario 1

This cycle is the result of a 15.8 km route driven at 15.45 and governed by the three typical speed limit of a city center: 50 km/h, 30 km/h (residential areas), 70 km/h (ring road). Three traffic codes have been experimented (green, orange, black) whose lengths, in this particular case, have been well forecast by the MSP: the model, in fact, resulted to be perfectly centered (BIAS $= -0.10\,\text{km/h}$) despite of the real driver more than once has exceeded the speed limit by a lot (m = 805 and m = 1243)[fig.5.1, 5.2].

Table 5.1: Scenario 1:speed KPI

	KPI $[km/h]$
MAE	12.41
BIAS	-0.10

Table 5.2: Scenario 1:energy KPI

	Abs	Rel
	[kWh]	[%]
E_{RL}	0.005	-0.4
E_{I+}	0.032	-2.0
E_{I-}	0.032	-2.0

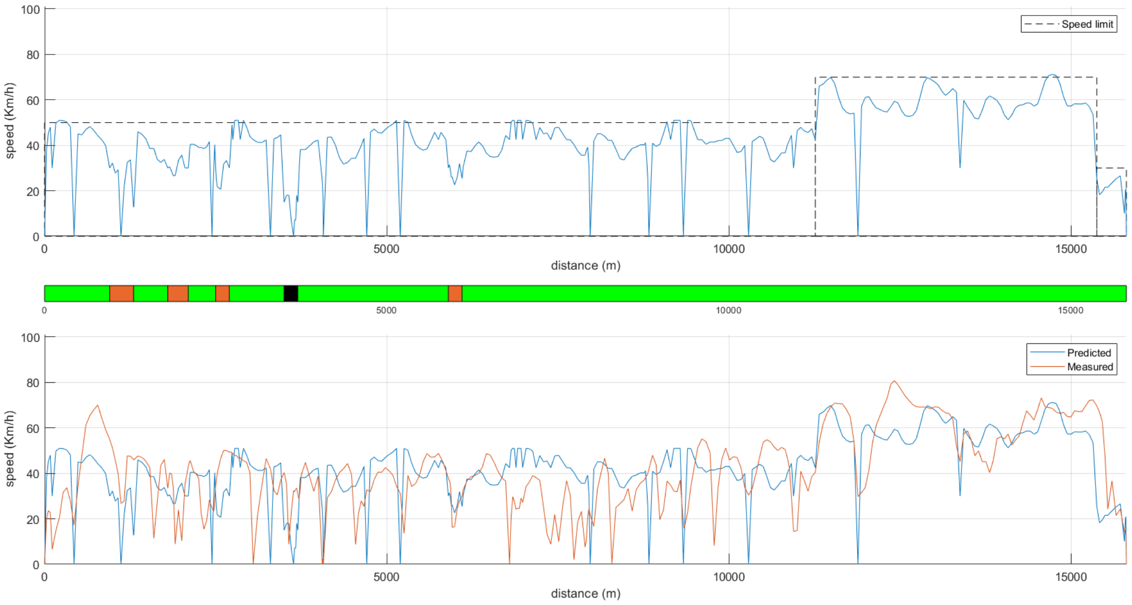


Figure 5.1: Scenario 1: speed

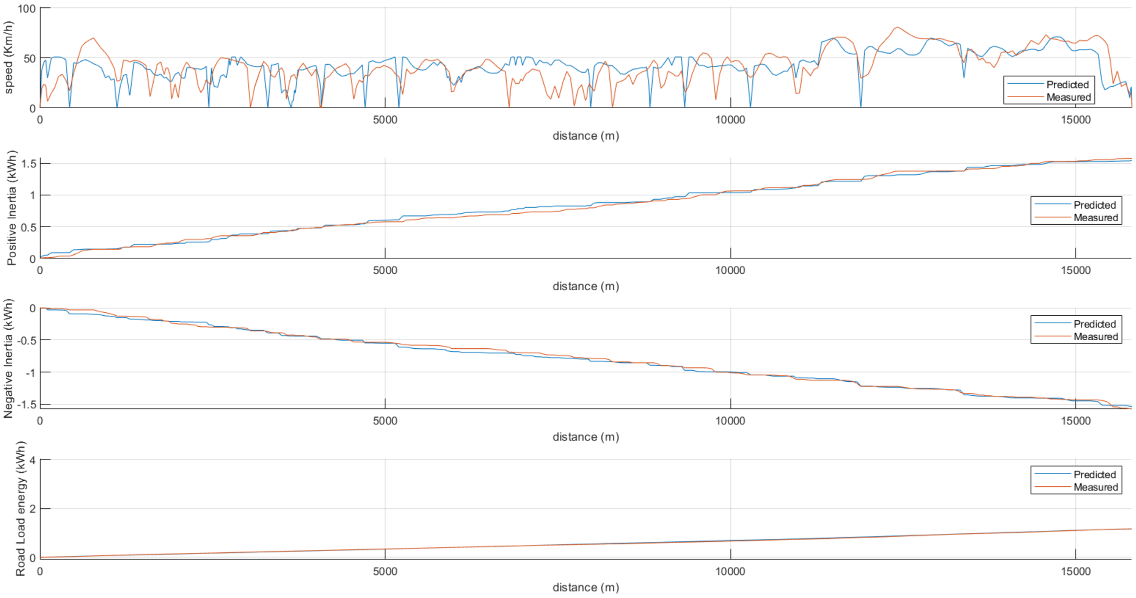


Figure 5.2: Scenario 1: energy

5.2 Scenario 2

This scenario represents the one with the longest length (17831 m) and the most populated traffic codes vector (55 elements). Such a variety has been obtained doing the measurements at the rush hour (17.43). In this case, the predicted speed tends to be higher than the actual one, especially in the central area (from m 7000 to m 12000), but still keeping a narrow value of BIAS (3.31 $-3.31\,\mathrm{km/h}$)[fig.5.3]. Looking at the energy KPI, it is possible to see how this scenario is the only one where the inertial energies are overestimated[fig.5.4]. Possible reasons for this behavior will be discussed in detail in the next chapter, where possible solutions are introduced as future works.

Table 5.3: Scenario 2:speed KPI

	KPI $[km/h]$
MAE	11.07
BIAS	3.31

Table 5.4: Scenario 2:energy KPI

	Abs $[kWh]$	Rel $[\%]$
E_{RL}	0.004	3.55
E_{I^+}	0.036	23.47
E_{I^-}	0.036	23.47

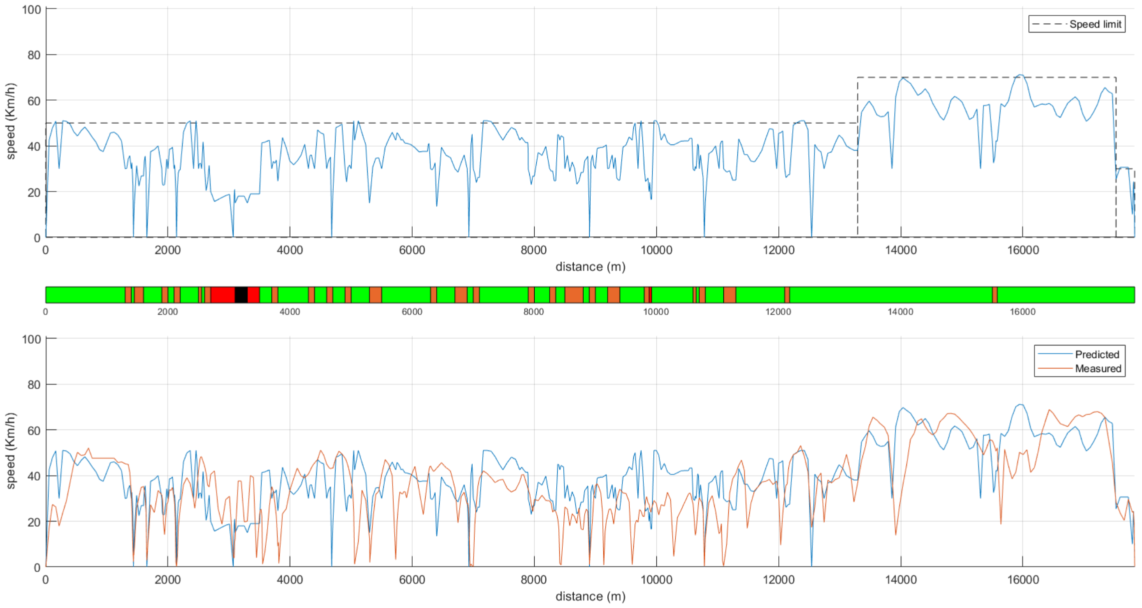


Figure 5.3: Scenario 2: speed

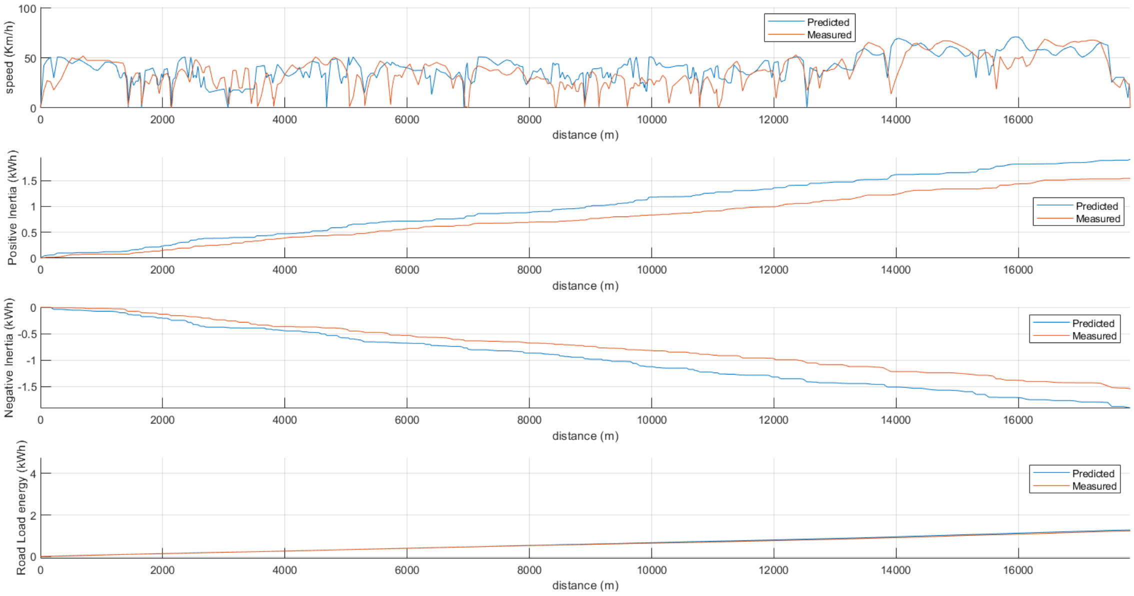


Figure 5.4: Scenario 2: energy

5.3 Scenario 3

This scenario has been introduced to test the robustness of the algorithm in case of a short length prediction (3,186 km). In this case it can be easily noticed how the oscillations of the predicted signal seems to have a totally different behavior with respect to the measured signal. This condition is largely due to the fact that, in this case, the traffic codes information has not been so reliable like in the other two previous cases. In general, has been observed how the predicted and the measured road load energy have the tendency to be coincident: this is in accordance with the low values of BIAS encountered and with the fact that all the three scenarios are denoted by a low value of average speed, resulting in small contributes of the term $v^2 \cdot F_2$ [fig.5.5,5.6].

Table 5.5: Scenario 3:speed KPI

	KPI $[km/h]$
MAE	8.27
BIAS	1.88

Table 5.6: Scenario 2:energy KPI

	Abs $[kWh]$	Rel [%]
E_{RL}	0.003	1.76
E_{I^+}	0.046	-9.82
E_{I^-}	0.056	-11.72

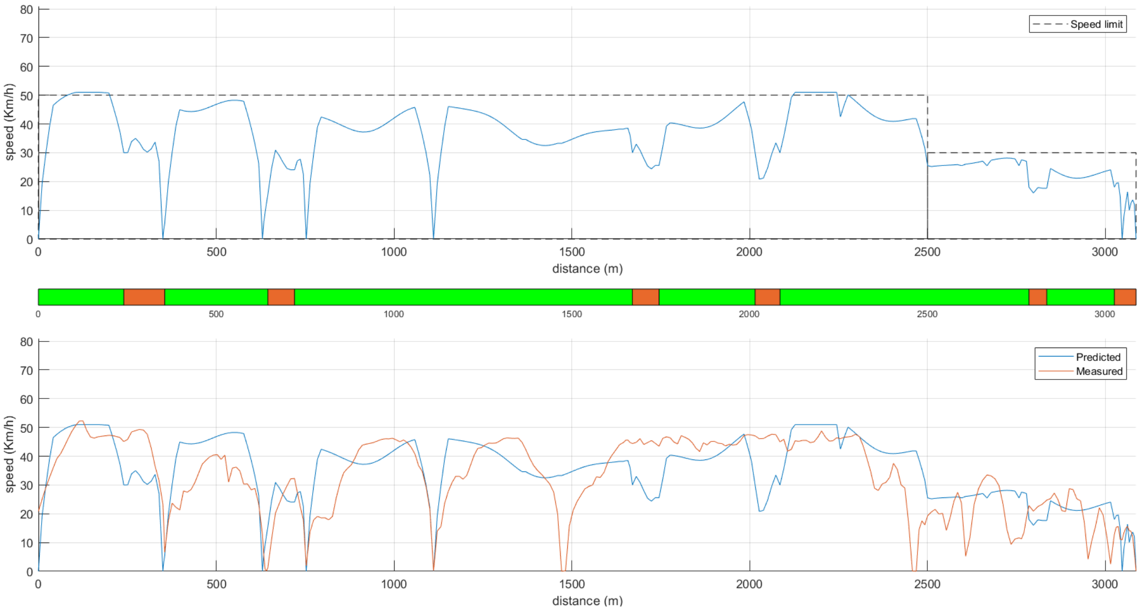


Figure 5.5: Scenario 3: speed

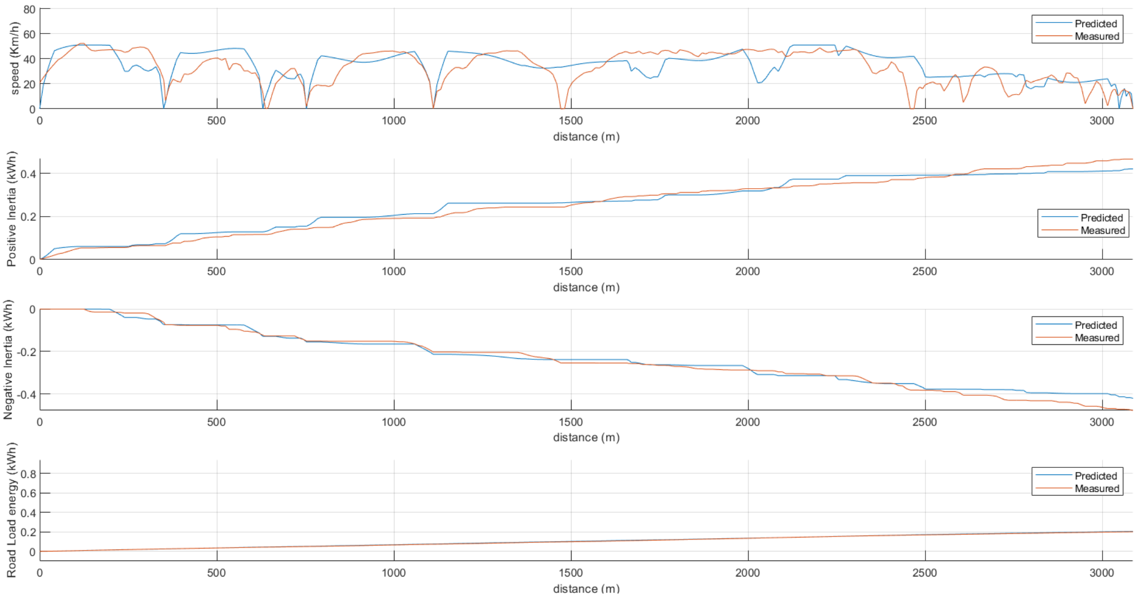


Figure 5.6: Scenario 3: energy

Chapter 6

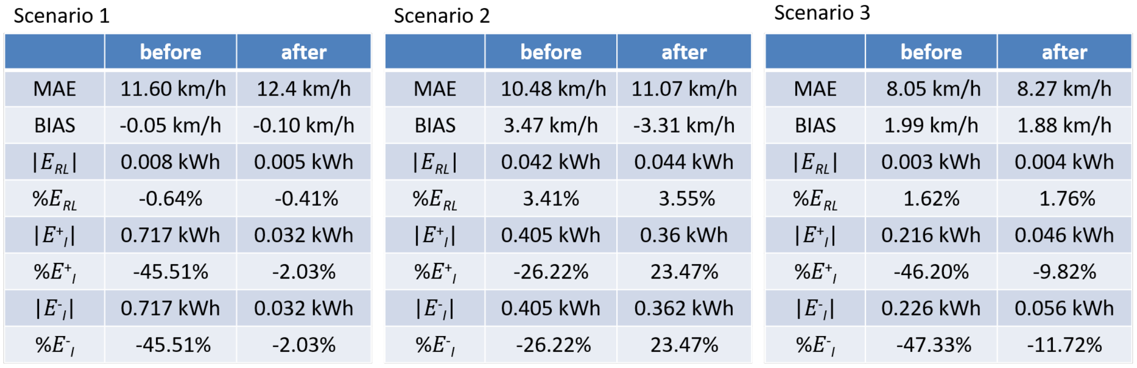
Conclusions and future works

6.1 Initial and final speed

Looking at the energy KPI in Scenario 1 [fig.5.2] and Scenario 2 [fig.5.4], it can be noticed that both of them present coincident errors for positive and negative inertia ($\%E_{I^+} = \%E_{I^-}$). On the contrary, this condition is not found in Scenario 3 [fig.5.6]. This behavior is coherent with the way inertial forces are defined: they represent the energy requested by the vehicle to be accelerated and decelerated, including the work done by non-conservative forces. Consequently, the cumulative overall inertial energy, for a cycle whose initial and final speed are coincident, is equal to zero. So, the coincident errors in first two cases are due to the fact that both the predicted and the measured initial and final speeds are equal to zero. Instead,in Scenario 3, both the predicted and the measured final speeds are equal to zero, while the initial speeds are not coincident. This observation could appear as trivial, but it highlights how the prediction accuracy can be increased when initial and final speed are well-imposed. Since in the current version of the algorithm these values are imposed manually, a possible improvement could be the implementation of a further algorithm branch that generates them by basing on navigation data in the vicinity of ZEZ-in and ZEZ-out gates.

6.2 Oscillation energy

The comparison of the energy KPI brings to an additional observation: the narrow underestimation of the inertial energies in Scenario 1 ($\approx -2\%$) and 3($\approx -10\%$) are quite in contrast with the overestimation that took place in Scenario 2 ($> +20\%$). A possible cause for this behavior has been investigated in the oscillations introduced to reproduce the effects of the traffic flows, since their reliability is still to be verified. In fact, despite of magnitudes and frequencies of these oscillations have been retrieved starting from the frequency analysis of real data, as done by many before, the procedure described in chapter 5 is an empirical approach that can not count on bibliographic references. Thus, to make a first evaluation of this methodology, predictions for all the three scenarios described in chapter 5 have been executed before and after the oscillation were added to the MAS. The results of this series of comparative simulations are reported in fig.6.1. As can be easily seen, MAS, BIAS and road load energy are quite unaffected by the introduction of the oscillations. The situation is totally different for the inertial energies: underestimation in Scenario 1 e 3 is effectively reduced by the introduction of the oscillations, while is way too overcompensated in Scenario 2. Unfortunately, the current information availability on traffic codes and measured cycles is not sufficient for an accurate identification of the causes for this misalignment. Nevertheless, one general conclusion can be retrieved: in three out of three cases the inertial energies would be heavily underestimated without the addition of the oscillations. This trend confirms how the oscillation constitute an essential part for a robust prediction and how this topic deserves to be



Scenario 1

	before	after
MAE	11.60 km/h	12.4 km/h
BIAS	-0.05 km/h	-0.10 km/h
$\lvert E_{RL}\rvert$	0.008 kWh	0.005 kWh
$\%E_{RL}$	-0.64%	-0.41%
$\lvert E^+_I\rvert$	0.717 kWh	0.032 kWh
$\%E^+_I$	-45.51%	-2.03%
$\lvert E^-_I\rvert$	0.717 kWh	0.032 kWh
$\%E^-_I$	-45.51%	-2.03%

Scenario 2

	before	after
MAE	10.48 km/h	11.07 km/h
BIAS	3.47 km/h	-3.31 km/h
$\lvert E_{RL}\rvert$	0.042 kWh	0.044 kWh
$\%E_{RL}$	3.41%	3.55%
$\lvert E^+_I\rvert$	0.405 kWh	0.36 kWh
$\%E^+_I$	-26.22%	23.47%
$\lvert E^-_I\rvert$	0.405 kWh	0.362 kWh
$\%E^-_I$	-26.22%	23.47%

Scenario 3

	before	after
MAE	8.05 km/h	8.27 km/h
BIAS	1.99 km/h	1.88 km/h
$\lvert E_{RL}\rvert$	0.003 kWh	0.004 kWh
$\%E_{RL}$	1.62%	1.76%
$\lvert E^+_I\rvert$	0.216 kWh	0.046 kWh
$\%E^+_I$	-46.20%	-9.82%
$\lvert E^-_I\rvert$	0.226 kWh	0.056 kWh
$\%E^-_I$	-47.33%	-11.72%

Figure 6.1: Effects of oscillations on KPI

further explored. Anyway, a possible workaround to energy estimations issues could be provided by supplying the SPP with a feedback signal of the vehicle speed. In this way, the algorithm could be programmed for a real-time KPI evaluation, that would allow it to autonomously select another driver type and refresh the prediction in case a certain error threshold is overcome. Naturally, a full calibration of all the three driver types it is the necessary condition for this solution.

6.3 Road curvature radius and weather conditions

In the real world, traffic and stop events are not the only reasons that can induce a driver to reduce the vehicle speed with respect to the legal speed limit. For example, sometimes the road shape or condition is that proceeding according to the speed limit value would be unsafe. Even if they are not between the current available input, the SPP is prearranged to operate a furthermore limitation to the assignable speeds domain by elaborating the information regarding weather conditions (WeCon) and the road curvature radius R. This enrichment would be easily implemented by flanking V and Q [fig.6.2] with another upper limit condition: the speed C. C represents the maximum speed allowed by the limit of adherence and can be retrieved by equaling the centrifugal force and the centripetal friction force:

$$M \cdot \frac{C^2}{R} = M \cdot g \cdot \mu_{WeCon} \tag{6.1}$$

Obtaining:

$$C = \sqrt{R \cdot g \cdot \mu_{WeCon}} \tag{6.2}$$

where:

- M is the mass of the vehicle;
- R is the road curvature radius;
- μ_{WeCon} is the friction coefficient expressed as function of the weather conditions;

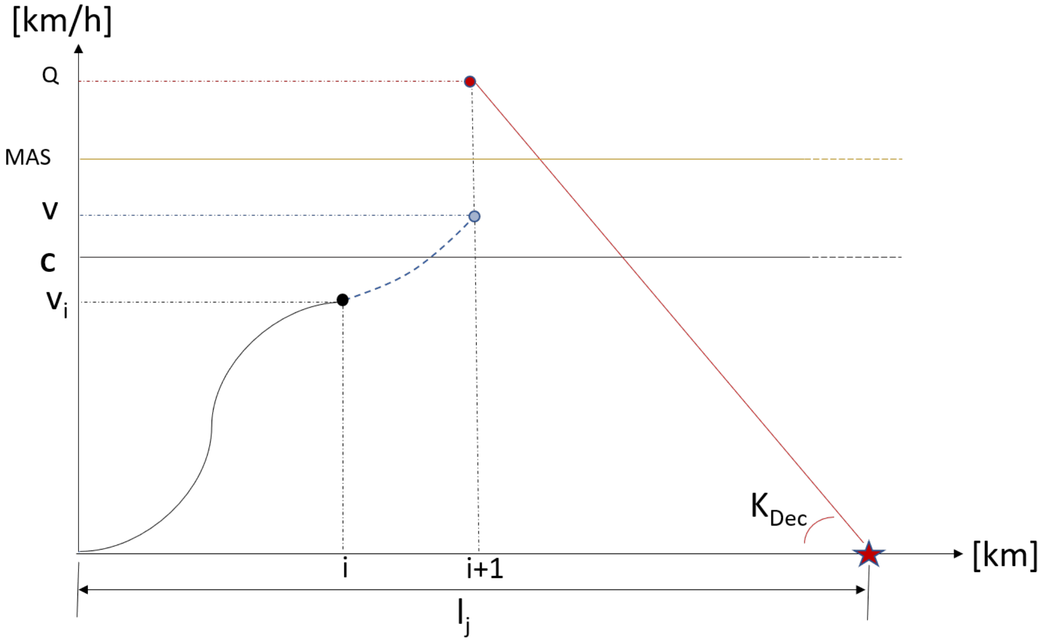


Figure 6.2: Limit of adherence C

6.4 Stop event duration, slope and eco-Driving

The space-based speed profile generated by the SPP does not represents on its own the totality of the information needed by the PF to actuate the energy management optimization strategies. Before feeding the PF, the space-based profile needs to converted into a time-based profile. This process introduces a new prediction variable that only affects the time-based model: the *stop events duration* vector. The length of this vector depends on how many times the predicted speed had touched the zero value, and its elements represent the duration of the single stop events expressed in seconds. A concept for the logic that assigns the duration values to each element is currently defined but is still to be implemented. This concept starts from the acquisition of real stop event duration that has already took place during the recordings previously discussed. The idea consists in a random assignment where the assignable values have a distribution of probability derived from the real data. At the moment, to execute the simulations, all the duration are imposed to be equal to zero. The implementation of this SPP's branch has the priority between the next enrichment, since the idle time can have a huge impact on the overall efficiency of a cycle. Once the time-based speed profile has been generated, it is finally sent to the PF with the last of the information generated by the eHr: the slope. Thanks to the slope, the PF are able to include the potential energy gradient of the path in their calculations, with the aim of maximizing the amount of energy produced by the regenerative braking system. The implementation of all the so far discussed future works finds its natural evolution in the so-called *eco-Driving* functions. When the driver sends a navigation request for a certain destination, the eco-Driving asks for all the predictions to be repeated for several routes alternative to the one chosen by the driver, then it operates the minimization of a cost function, identifying the most efficient route that is immediately suggested to the driver.

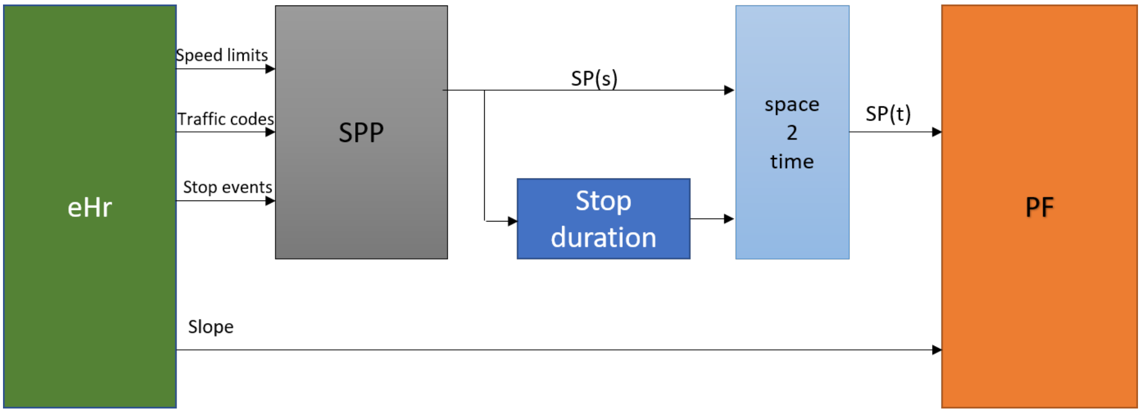


Figure 6.3: Predictive Functions layout

Bibliography

[1] IEA - International Energy Agency. *Transport. Tracking Clean Energy Progress.* 2018.

[2] T. J. Böhme and B. Frank. “Hybrid vehicles. Theory, Methods and Applications”. 1st ed. Advances in Industrial Control Series. Springer International Publishing, **2017**. ISBN: 978-3-319-51315-7.

[3] Università di Bologna. *LABORATORY. GREEN MOBILITY RESEARCH LAB.* URL: `https://site.unibo.it/gmrl/en/laboratory`.

[4] L. Brunelli. “Design of a user-friendly Plug-in HEV simulation platform for energy management strategies and e-Horizon functions performance evaluation”. Master degree. Università di Bologna, 2018.

[5] A. Capancioni. “Development of a Predictive Thermal Management Function for Plug-in Hybrid Electric Vehicles”. Master degree. Università di Bologna, 2018.

[6] G. Caramia. “Development of an innvative not rule-based energy management strategy for a hybrid electric vehicle”. Master degree. Università di Bologna, 2016.

[7] R. Casadio. “Development of interface algorithms to let a connected vehiclemanage and use smart city services and eHorizon functionalities”. Master degree. Università di Bologna, 2020.

[8] Honda Motor CO. *The Honda Hybrid System.* 2018. URL: `https://world.honda.com/automobile-technology/IMA/detail/`.

[9] dSPACE. “Developments on the Electronic Horizon”. *dSPACE magazine* (2010).

[10] ICCT EU. *2020-2030 CO_2 standards for new cars and light-commercial vehicles.* 2018.

[11] P. Gonnella. “Development of a V2X platform for predictive driving functionvalidation”. Master degree. Università di Bologna, 2019.

[12] M. Greco. *Automotive Radar. IEEE Radar Conference.* IEEE. 2012.

[13] SAE International. *SURFACE VEHICLE RECOMMENDED PRACTICE. Drive Quality Evaluation for Chassis Dynamometer Testing.* 2014.

[14] SAE International. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.* 2016. DOI: `10.4271/J3016_201609`.

[15] Lin Yang Jingni Yuan. "Predictive energy management strategy for con-nected 48V hybrid electric vehicles". *Volume 187.* Energy. URL: `https://doi.org/10.1016/j.energy.2019.115952`.

[16] Stephanie Lefevre et al. "Comparison of parametric and non-parametric approaches for vehicle speed prediction". June 2014. DOI: `10.1109/ACC.2014.6858871`.

[17] Kuan Liu et al. "Vehicle Velocity Prediction and Energy Management Strategy Part 1: Deterministic and Stochastic Vehicle Velocity Prediction Using Machine Learning". *SAE Technical Paper.* SAE International, Apr. 2019. DOI: `10.4271/2019-01-1051`. URL: `https://doi.org/10.4271/2019-01-1051`.

[18] MAHLE. *Experience urban mobility in a new way.* 2017. URL: `https://www.meet.mahle.com`.

[19] NAVILOCK. *Navilock NL-8002U USB 2.0 Multi GNSS Receiver u-blox 8 1.5 m.* URL: `https://www.navilock.de/produkte/G_62523/dokumente.html?`.

[20] NOAA. *What is LIDAR?* 6/25/18. URL: `https://oceanservice.noaa.gov/facts/lidar.html`.

[21] N. Prof. Cavina. *Hybrid propulsion systems for automotive applications.* Università di Bologna, 2017.

[22] Valeo. *Smart Technology for Smarter Cars.* 2017. URL: `https://www.valeo.com/en/who-we-are/`.

[23] the free encyclopedia Wikipedia. *MATLAB.* URL: `https://en.wikipedia.org/wiki/MATLAB`.

[24] the free encyclopedia Wikipedia. *Simulink.* URL: `https://en.wikipedia.org/wiki/Simulink`.

[25] the free encyclopedia Wikipedia. *Stateflow.* URL: `https://en.wikipedia.org/wiki/Stateflow`.

[26] X-Engineer.org. *Understanding micro, mild, full and plug-in hybrid electric vehicles.* 2017. URL: `https://x-engineer.org/automotive-engineering/vehicle/hybrid/micro-mild-full-hybrid-electric-vehicle/`.