Co-simulation Methodologies for Hybrid and Electric Vehicle Dynamics

FINAL DEGREE PROJECT

ADVISOR: Francisco Javier García de Jalón de la Fuente
AUTHOR: Julia Veintimilla Porlán

Madrid, February 2016
Acknowledgements

This final degree project represents many hours of work, great effort and an enlightening learning experience that would not have been possible without the major support of many people.

First of all, I would like to thank particularly my parents, Jorge and Julia, for supporting me all these years of my degree studies and encouraging me in all my decisions.

Then, I would like to thank my advisor Francisco Javier García de Jalón de la Fuente, for his tremendous academic support, that was key to the completion of this project.

I also thank María Dolores Gutiérrez, who helped me get started with MBS3D software and with co-simulation methodologies, among other things, and dedicated many hours to my project showing great interest.

I am very grateful to all the friends that have shared with me these past years. Especially, I would like to thank Irene, who has always been available and has given me very valuable advice; Belén, for sharing with me countless hours of work and great moments; Bea, for her great encouragement; Ellen, for her unconditional support and for the time spent proofreading my work. Thank you for everything I have learned from you and with you.
Summary

In recent decades, full electric and hybrid electric vehicles have emerged as an alternative to conventional cars due to a range of factors, including environmental and economic aspects. These vehicles are the result of considerable efforts to seek ways of reducing the use of fossil fuel for vehicle propulsion. Sophisticated technologies such as hybrid and electric powertrains require careful study and optimization. Mathematical models play a key role at this point.

Currently, many advanced mathematical analysis tools, as well as computer applications have been built for vehicle simulation purposes. Given the great interest of hybrid and electric powertrains, along with the increasing importance of reliable computer-based models, the author decided to integrate both aspects in the research purpose of this work. Furthermore, this is one of the first final degree projects held at the ETSII (Higher Technical School of Industrial Engineers) that covers the study of hybrid and electric propulsion systems.

The present project is based on MBS3D 2.0, a specialized software for the dynamic simulation of multibody systems developed at the UPM Institute of Automobile Research (INSIA). Automobiles are a clear example of complex multibody systems, which are present in nearly every field of engineering. The work presented here benefits from the availability of MBS3D software. This program has proven to be a very efficient tool, with a highly developed underlying mathematical formulation.

On this basis, the focus of this project is the extension of MBS3D features in order to be able to perform dynamic simulations of hybrid and electric vehicle models. This requires the joint simulation of the mechanical model of the vehicle, together with the model of the hybrid or electric powertrain. These sub-models belong to completely different physical domains. In fact the powertrain consists of energy storage systems, electrical machines and power electronics, connected to purely mechanical components (wheels, suspension, transmission, clutch...). The challenge today is to create a global vehicle model that is valid for computer simulation. Therefore, the main goal of this project is to apply co-simulation methodologies to a comprehensive model of an electric vehicle, where sub-models from different areas of engineering are coupled. The created electric vehicle (EV) model consists of a separately excited DC electric motor, a Li-ion battery pack, a DC/DC chopper converter and a multibody vehicle model.

Co-simulation techniques allow car designers to simulate complex vehicle architectures and behaviors, which are usually difficult to implement in a real environment due to safety and/or economic reasons. In addition, multi-domain computational models help to detect the effects of different driving patterns and parameters and improve the models in a fast and effective way. Automotive designers can greatly benefit from a multidisciplinary approach of new hybrid and electric vehicles.

In this case, the global electric vehicle model includes an electrical subsystem and a mechanical subsystem. The electrical subsystem consists of three basic components: electric motor, battery pack and power converter. A modular representation is used for building the dynamic model of the vehicle drivetrain. This means that every component of the drivetrain (submodule) is modeled separately and has its own general dynamic model, with clearly defined inputs and outputs. Then, all the particular submodules are assembled according to the drivetrain configuration and, in this way, the power flow across the components is completely determined. Dynamic models of electrical components are often based on equivalent circuits, where Kirchhoff's voltage and current laws are applied to draw the algebraic and differential equations. Here, Randles circuit is used for dynamic modeling
of the battery and the electric motor is modeled through the analysis of the equivalent circuit of a separately excited DC motor, where the power converter is included.

The mechanical subsystem is defined by MBS3D equations. These equations consider the position, velocity and acceleration of all the bodies comprising the vehicle multibody system. MBS3D 2.0 is entirely written in MATLAB and the structure of the program has been thoroughly studied and understood by the author. MBS3D software is adapted according to the requirements of the applied co-simulation method. Some of the core functions are modified, such as integrator and graphics, and several auxiliary functions are added in order to compute the mathematical model of the electrical components.

By coupling and co-simulating both subsystems, it is possible to evaluate the dynamic interaction among all the components of the drivetrain. ‘Tight-coupling’ method is used to co-simulate the sub-models. This approach integrates all subsystems simultaneously and the results of the integration are exchanged by function-call. This means that the integration is done jointly for the mechanical and the electrical subsystem, under a single integrator and then, the speed of integration is determined by the slower subsystem.

Simulations are then used to show the performance of the developed EV model. However, this project focuses more on the validation of the computational and mathematical tool for electric and hybrid vehicle simulation. For this purpose, a detailed study and comparison of different integrators within the MATLAB environment is done. Consequently, the main efforts are directed towards the implementation of co-simulation techniques in MBS3D software. In this regard, it is not intended to create an extremely precise EV model in terms of real vehicle performance, although an acceptable level of accuracy is achieved.

The gap between the EV model and the real system is filled, in a way, by introducing the gas and brake pedals input, which reflects the actual driver behavior. This input is included directly in the differential equations of the model, and determines the amount of current provided to the electric motor. For a separately excited DC motor, the rotor current is proportional to the traction torque delivered to the car wheels. Therefore, as it occurs in the case of real vehicle models, the propulsion torque in the mathematical model is controlled through acceleration and brake pedal commands. The designed transmission system also includes a reduction gear that adapts the torque coming for the motor drive and transfers it to the driven wheels by means of a differential.

The main contribution of this project is, therefore, the implementation of a new calculation path for the wheel torques, based on performance characteristics and outputs of the electric powertrain model. Originally, the wheel traction and braking torques were input to MBS3D through a vector directly computed by the user in a MATLAB script. Now, they are calculated as a function of the motor current which, in turn, depends on the current provided by the battery pack across the DC/DC chopper converter. The motor and battery currents and voltages are the solutions of the electrical ODE (Ordinary Differential Equation) system coupled to the multibody system. Simultaneously, the outputs of MBS3D model are the position, velocity and acceleration of the vehicle at all times. The motor shaft speed is computed from the output vehicle speed considering the wheel radius, the gear reduction ratio and the transmission efficiency. This motor shaft speed, somehow available from MBS3D model, is then introduced in the differential equations corresponding to the electrical subsystem. In this way, MBS3D and the electrical powertrain model are interconnected and both subsystems exchange values resulting as expected with tight-coupling approach. This process is shown in Figure 0.1.
When programming mathematical models of complex systems, code optimization is a key step in the process. A way to improve the overall performance of the integration, making use of C/C++ as an alternative programming language, is described and implemented. Although this entails a higher computational burden, it leads to important advantages regarding co-simulation speed and stability. In order to do this, it is necessary to integrate MATLAB with another integrated development environment (IDE), where C/C++ code can be generated and executed. In this project, C/C++ files are programmed in Microsoft Visual Studio and the interface between both IDEs is created by building C/C++ MEX file functions. These programs contain functions or subroutines that can be dynamically linked and executed from MATLAB. This process achieves reductions in simulation time up to two orders of magnitude.

The tests performed with different integrators, also reveal the stiff character of the differential equations corresponding to the electrical subsystem, and allow the improvement of the co-simulation process. When varying the parameters of the integration and/or the initial conditions of the problem, the solutions of the system of equations show better dynamic response and stability, depending on the integrator used. Several integrators, with variable and non-variable step-size, and for stiff and non-stiff problems are applied to the coupled ODE system. Then, the results are analyzed, compared and discussed.

From all the above, the project can be divided into four main parts: 1. Creation of the equation-based electric vehicle model; 2. Programming, simulation and adjustment of the electric vehicle model; 3. Application of co-simulation methodologies to MBS3D and the electric powertrain subsystem; and 4. Code optimization and study of different integrators. Additionally, in order to deeply understand the context of the project, the first chapters include an introduction to basic vehicle dynamics, current classification of hybrid and electric vehicles and an explanation of the involved technologies such as brake energy regeneration, electric and non-electric propulsion systems for EVs and HEVs (hybrid electric vehicles) and their control strategies. Later, the problem of dynamic modeling of hybrid and electric vehicles is discussed. The integrated development environment and the simulation tool are also briefly described. The core chapters include an explanation of the major co-simulation methodologies and how they have been programmed and applied to the electric powertrain model together with the multibody system dynamic model. Finally, the last chapters summarize the main results and conclusions of the project and propose further research topics.

In conclusion, co-simulation methodologies are applicable within the integrated development environments MATLAB and Visual Studio, and the simulation tool MBS3D 2.0, where equation-based models of multidisciplinary subsystems, consisting of mechanical and electrical components, are coupled and integrated in a very efficient way.

**Key words:** co-simulation methodologies, multibody system, full electric vehicle, hybrid electric vehicle, equation-based dynamic model.

**UNESCO codes:** 1203.09 (Computer Assisted Design), 1203.26 Simulation, 1206 (Numerical Analysis), 1206.12 (Differential Equations), 3313 (Mechanical Engineering and Technology), 3317 (Motor Vehicle Technology), 3317.02 (Automobiles), 3327 (Transportation Systems Technology), 3306.03 (Electric Motors).

---

**Figure 0.1. Connection and exchange of values between MBS3D and the electrical subsystem**

When programming mathematical models of complex systems, code optimization is a key step in the process. A way to improve the overall performance of the integration, making use of C/C++ as an alternative programming language, is described and implemented. Although this entails a higher computational burden, it leads to important advantages regarding co-simulation speed and stability. In order to do this, it is necessary to integrate MATLAB with another integrated development environment (IDE), where C/C++ code can be generated and executed. In this project, C/C++ files are programmed in Microsoft Visual Studio and the interface between both IDEs is created by building C/C++ MEX file functions. These programs contain functions or subroutines that can be dynamically linked and executed from MATLAB. This process achieves reductions in simulation time up to two orders of magnitude.

The tests performed with different integrators, also reveal the stiff character of the differential equations corresponding to the electrical subsystem, and allow the improvement of the co-simulation process. When varying the parameters of the integration and/or the initial conditions of the problem, the solutions of the system of equations show better dynamic response and stability, depending on the integrator used. Several integrators, with variable and non-variable step-size, and for stiff and non-stiff problems are applied to the coupled ODE system. Then, the results are analyzed, compared and discussed.

From all the above, the project can be divided into four main parts: 1. Creation of the equation-based electric vehicle model; 2. Programming, simulation and adjustment of the electric vehicle model; 3. Application of co-simulation methodologies to MBS3D and the electric powertrain subsystem; and 4. Code optimization and study of different integrators. Additionally, in order to deeply understand the context of the project, the first chapters include an introduction to basic vehicle dynamics, current classification of hybrid and electric vehicles and an explanation of the involved technologies such as brake energy regeneration, electric and non-electric propulsion systems for EVs and HEVs (hybrid electric vehicles) and their control strategies. Later, the problem of dynamic modeling of hybrid and electric vehicles is discussed. The integrated development environment and the simulation tool are also briefly described. The core chapters include an explanation of the major co-simulation methodologies and how they have been programmed and applied to the electric powertrain model together with the multibody system dynamic model. Finally, the last chapters summarize the main results and conclusions of the project and propose further research topics.

In conclusion, co-simulation methodologies are applicable within the integrated development environments MATLAB and Visual Studio, and the simulation tool MBS3D 2.0, where equation-based models of multidisciplinary subsystems, consisting of mechanical and electrical components, are coupled and integrated in a very efficient way.

**Key words:** co-simulation methodologies, multibody system, full electric vehicle, hybrid electric vehicle, equation-based dynamic model.

**UNESCO codes:** 1203.09 (Computer Assisted Design), 1203.26 Simulation, 1206 (Numerical Analysis), 1206.12 (Differential Equations), 3313 (Mechanical Engineering and Technology), 3317 (Motor Vehicle Technology), 3317.02 (Automobiles), 3327 (Transportation Systems Technology), 3306.03 (Electric Motors).
Table of Contents

1. Introduction .......................................................................................................................... 5
   1.1. State of the Art ............................................................................................................... 5
   1.2. Comparison of EVs, HEVs and Conventional Vehicles ............................................... 6
   1.3. Hybrid and Electric Vehicle Modeling and Control ....................................................... 7
2. Objectives ............................................................................................................................. 9
   2.1. Main Objective ............................................................................................................. 9
   2.2. Secondary Objectives ................................................................................................ 9
3. Methodology ....................................................................................................................... 11
   3.1. Creation of the Equation-Based Electric Vehicle Model ............................................... 11
       3.1.1. Study of Hybrid and Electric Vehicle Technologies ............................................... 11
       3.1.2. Study of Existing Modeling Approaches ................................................................ 12
       3.1.3. Selection of the Electric Vehicle Equations ............................................................ 12
   3.2. Study of MBS3D 2.0 Simulation Tool ......................................................................... 13
   3.3. Programming of the Electric Vehicle Model in MATLAB ........................................... 13
   3.4. Co-simulation of MBS3D and Electric Vehicle Model ................................................ 14
   3.5. Mex File Function Implementation ............................................................................. 14
   3.6. Note Regarding Programming Code ........................................................................... 14
4. Basic Longitudinal Dynamics of Vehicles .......................................................................... 17
   4.1. Traction Force ............................................................................................................ 17
       4.1.1. Drive System-generated Tractive Effort ................................................................. 18
       4.1.2. Maximum Tractive Effort Based on Adherence Limit ............................................ 20
   4.2. Motion Resistance Forces .......................................................................................... 20
       4.2.1. Aerodynamic Resistance ....................................................................................... 21
       4.2.2. Rolling Resistance ................................................................................................ 22
       4.2.3. Grading Resistance ............................................................................................... 23
   4.3. Vehicle Power Requirements ...................................................................................... 23
5. Electric and Hybrid Electric Vehicles: Classification, Control Strategies and Brake Energy
   Regeneration ......................................................................................................................... 25
   5.1. Full Electric Vehicles .................................................................................................. 25
   5.2. Hybrid Electric Vehicles ............................................................................................. 26
       5.2.1. According to Topology .......................................................................................... 26
       5.2.2. According to Degree of Hybridization ................................................................. 28
   5.3. Energy Management and Control Strategies for EVs and HEVs .................................. 30
       5.3.1. Series HEV Control Strategies .............................................................................. 30
       5.3.2. Parallel HEV Control Strategies .......................................................................... 32
       5.3.3. Full EV Control Strategies ................................................................................... 32
Table of Contents

5.4. Brake Energy Regeneration ..................................................33
   5.4.1. Types of Regenerative Braking Systems (RBS) ..................33

6. Electric and Non-electric Propulsion Technologies for EVs and HEVs ...37
   6.1. Electric Propulsion Systems ............................................37
      6.1.1. EV-HEV Motor Requirements ...................................37
      6.1.2. Brushless DC Motor .............................................38
      6.1.3. Permanent Magnet Synchronous Motor .......................38
      6.1.4. Switched Reluctance Motor ...................................39
      6.1.5. AC Induction Motor .............................................39
   6.2. Non-electric Propulsion Systems .....................................40
      6.2.1. Internal Combustion Engines ...................................40
      6.2.2. Fuel Cells .......................................................44

7. Energy Storage Devices ..........................................................49
   7.1. Rechargeable Electrochemical Batteries ..............................50
      7.1.1. Lead-acid .......................................................51
      7.1.2. Nickel-cadmium ..................................................51
      7.1.3. Nickel-metal-hydride ..........................................51
      7.1.4. Lithium-ion .....................................................52
      7.1.5. Lithium-polymer ...............................................52
   7.2. Ultracapacitors or Supercapacitors ................................53
   7.3. Ultra-high-speed Flywheels ............................................54
   7.4. Battery Management System ..........................................55
      7.4.1. State of Charge Estimation ....................................57

8. Dynamic Modeling of Electric and Hybrid-Electric Propulsion Systems ....59
   8.1. Dynamic Modeling of Electric Motors ................................60
   8.2. Dynamic Modeling of Batteries ......................................61
   8.3. Dynamic Modeling of Electric Power Links .........................61
   8.4. Dynamic Modeling of Internal Combustion Engines ...............61
   8.5. Mathematical Model of Hybrid and Electric Vehicles ............62

9. Electric Vehicle Model Description ..........................................63
   9.1. Vehicle Parameters ...................................................63
   9.2. Electric Motor ................................................................64
   9.3. Battery Pack ................................................................66
      9.3.1. State of Charge ..................................................67
   9.4. Electrical Coupling Device ............................................68

10. Integrated Development Environment and Simulation Tool .................71
    10.1. MATLAB ..............................................................71
    10.2. Microsoft Visual Studio ..............................................72
    10.3. Simulation tool ........................................................73
10.3.1. Introduction to Multibody Systems Dynamic Simulation ........................................73
10.3.2. MBS3D 2.0 ........................................................................................................73
10.3.3. MEX File Functions .........................................................................................74
11. Co-simulation Process ...........................................................................................75
  11.1. Coupling of Models ............................................................................................75
  11.2. Introduction to MATLAB ODE Suite .................................................................76
  11.3. Coupling of MBS3D Software and Electrical Subsystem ....................................78
      11.3.1. Study and Reformulation of Electrical Powertrain Equations ......................78
      11.3.2. First Integration of Electric Vehicle Equations .............................................80
      11.3.3. Electrical Parameter Setting and Gear Ratio Adjustment .........................81
      11.3.4. 'Tight Coupling' Applied to MBS3D and Electrical Subsystem ..................84
  11.4. Complete Electric Vehicle Model .......................................................................91
  11.5. Translation into C/C++ Code ..............................................................................92
12. Results ....................................................................................................................95
  12.1. Comparison of Integration Time MATLAB - C/C++ ..........................................95
  12.2. Study and Comparison of MATLAB Integrators ...............................................96
  12.3. Study of Vehicle Performance Characteristics ................................................101
  12.4. Driving Cycle Implementation ........................................................................109
  12.5. Simulation with MATLAB Profiler ...................................................................113
13. Conclusions .............................................................................................................115
14. Further Research ...................................................................................................117
15. References .............................................................................................................119
16. Time Planning and Budget ....................................................................................121
  16.1. Time Planning ....................................................................................................121
  16.2. Budget ...............................................................................................................123
17. List of Figures ........................................................................................................125
18. List of Tables ..........................................................................................................129
19. List of Abbreviations .............................................................................................131
1. Introduction

The current concern about the environment as well as the constant search for driving comfort and efficiency, have led to a greater development of alternative vehicle propulsion systems (Thanh-Son Dao, Aden Seaman, John McPhee, 2010). The new propulsion ways for vehicles have been studied with a view to replacing the conventional internal combustion engine with more environmentally-friendly power sources. Throughout this process, many solutions have been proposed, all of them based on clean and sustainable forms of energy.

The electric vehicle was invented in 1834 and, up until now, it has been considered the most efficient vehicle model. However, its limited autonomy and high cost, due to the long charging time and great battery price, do not allow the widespread use of this type of vehicles. To overcome these disadvantages, hybrid vehicles were developed and, nowadays, they are recognized as the most feasible technology currently available.

Initially, hybrid vehicles appeared as a transitory solution to global warming, air pollution, and petroleum resource depletion, until electric vehicles (EVs) were fully implemented. Nevertheless, the discovery of the important advantages of these vehicles steered car manufacturers to look for ways of improving hybrid architectures. Consequently, in the late 90s, they launched the first mass-production hybrid vehicle prototypes. (Costas, 2009).

As their name suggests, hybrid vehicles are those that combine two or more different propulsion systems. Ordinarily, these power sources are an internal combustion engine (ICE) and one or more electric motors, properly interconnected for greater energy efficiency of the vehicle.

1.1. State of the Art

During the last decade, hybrid and full electric vehicles have become a feasible option for a growing number of consumers. This is mainly due to the greater concern about emissions and fuel saving advantages, although many drivers are more attracted by performance and fuel economy than by environmental issues. The creation of new design techniques, along with more efficient control strategies and progress in power electronics make hybrid and electric vehicles reliable and commercially viable.

Nowadays, the main vehicles involving electrical propulsion are full electric vehicles, hybrid electric vehicles and fuel cell vehicles. These types of vehicles require different control strategies and are now facing different challenges as well, since their development is not in the same stage.

From an environmental point of view, full electric vehicles offer zero-emission operation and efficient energy use, as well as the possibility to use alternative energy sources. Therefore, a widespread use of these vehicles can greatly reduce global air pollution and produce economical, technological and industrial growth. Two main issues affecting electric vehicle feasibility are range and cost. In order to make EVs affordable, the range has to be wide enough to fulfill the driver requirements and guarantee a reasonable level of autonomy. For this purpose, experts are studying and testing new battery technologies such as lithium-ion (Li-Ion) and nickel-metal hydride (Ni-MH), and other energy storage devices such as ultracapacitors. Regarding the cost, the individual components of the electric power train (batteries, power converters, electric motors, etc.) have to be enhanced and simplified, so that their price can be reduced (Chan, 2002). Additionally, EVs are becoming cheaper and extending their range thanks to advances in electric propulsion technologies, especially AC
induction machines and permanent magnet (PM) brushless motors. Electric vehicles are now commercialized mainly in the USA, Japan, Europe and India.

For HEVs, the major challenges are the design and optimization of the electric and non-electric propulsion systems, finding efficient control strategies based on the driving cycles, battery management and cost. In order to face these challenges, modern energy management strategies are arising such as fuzzy logic control, fuel consumption minimization strategy, stochastic methods, and others based on battery behavior. Despite their higher complexity, hybrid vehicle performance characteristics satisfy driver expectations to a greater extent than full electric vehicles.

EVs and HEVs are coexistent in the market, although they target different consumer groups. EVs are well accepted by public transport authorities, in places where electricity is accessible and cheap, and in countries with strict environmental regulations. On the other hand, HEVs are more suitable for drivers who expect high vehicle performance characteristics such as long driving ranges.

Figure 1.1 shows different directions for hybrid and electric vehicle development during the last decades.

![Figure 1.1. Technological trends of electric and hybrid vehicles (Chan, 2002)](image)

1.2. Comparison of EVs, HEVs and Conventional Vehicles

In regard to air pollution and fuel economy, hybrid and electric vehicles present clear advantages over conventional vehicles. Generally, HEVs consume less fuel than normal cars, since they combine the internal combustion engine with an electric propulsion source to increase efficiency. Plug-in hybrid electric vehicles and full electric vehicles have the ability to use electricity produced off-board, which completely eliminates the need for investing in petrol fuel. Although the electricity provided to these vehicles has a cost, numerous studies show that the price is considerably lower than that of conventional gasoline. On the other hand, purchase costs of hybrid and electric vehicles can be significantly higher, since they bring technologies that are in their early stages of development and their production is not
yet regular. Furthermore, the price of electric vehicles is increased by the cost of battery replacement during the vehicle life cycle.

From an environmental point of view, hybrid vehicles are always less pollutant than conventional cars. The environmental benefit of hybrid vehicles depends on the design of the drive train and the control strategy, and it greatly improves when both factors are optimized. Since they can also operate in pure-electric mode, avoiding CO\textsubscript{2}, NO\textsubscript{x} and SO\textsubscript{x} emissions derived from the combustion cycle, they can be as clean as battery-electric vehicles. Purely electric vehicles and plug-in hybrids operating in all-electric mode produce zero emissions, which represents a great environmental advantage. Even though the process of electricity generation in power plants produces gas emissions, it is clear that urban emissions are significantly reduced in comparison with conventional vehicles.

Hybrid vehicles have longer driving ranges than conventional vehicles due to the combination of both propulsion sources. However, this makes hybrid power trains more complex and expensive. Electric vehicles are now in the last position with regard to driving range, due to the limited capacity of the existing energy storage devices for EV applications. Nevertheless, the electric drive train architecture is simple and the power electronics technology is very advanced and reliable.

The important environmental and economic advantages of hybrid and electric vehicles, as well as their superior electromechanical power train, have recently increased research and innovation efforts towards their development. If the achievements continue at this rate, the use of these hi-tech vehicles will soon become widespread, thus leading to a very important growth of electric, automotive and electronic industries and bringing multiple benefits to these fields.

In this text, the abbreviation EV-HEV hereafter will include both purely electric and hybrid electric vehicles.

1.3. Hybrid and Electric Vehicle Modeling and Control

Apart from technical improvements, modeling and control are crucial for hybrid and electric vehicle production, from powertrain efficiency analysis to an estimate of the launch costs. Recently, both tasks have been the focus of research for engineers and car manufacturers and many papers have been published on these topics (Chan, 2002; Salmasi, 2007; Mikhail Granovskii, Marc A. Rosen, Ibrahim Dincer, 2006). There are quasistatic and dynamic mathematical models of every type of electric and hybrid vehicle, which have been run and validated using different approaches and simulation softwares such as MATLAB/Simulink, ADVISOR, AVL CRUISE, ITI-SimulationX, etc.

Mathematical modeling for simulation purposes is a key point in the automotive industry, where manufacturing times and costs are of great importance. The existing models highlight different aspects of the power train dynamics and use different methodologies for the study of the vehicle. Usually, all mathematical models include the particular model of every component of the power train and overall performance analysis. Multi-domain models are required for vehicle simulation, since the equations defining the components of the power train belong to different fields of science and technology. From this point of view, three core modeling philosophies can be distinguished: physical modeling, signal modeling and equation-based modeling (Wenyong Li, A. Abel, K. Todtermuschke, Tong Zhang, 2007). More and more mathematical models are trying to combine the three aforementioned techniques simultaneously, and many modeling environments and simulation tools have been created and improved to this end.
Particularly important are co-simulation methodologies which aim to couple two or more subsystems that compose the complete vehicle model and belong to different fields of engineering. This work focuses on modeling and control of hybrid and electric vehicles and, more specifically, on the equation-based dynamic model of an electric vehicle coupled with an existing mechanical model of the drive train based on multibody system dynamics, which has been designed and simulated through MATLAB and Visual Studio C/C++ software.
2. Objectives

The aim of this work is to continue the development of MBS3D 2.0, an open-source program for the dynamic simulation of multibody systems developed at INSIA-UPM. This simulation tool has been the subject of several studies, for example, a sensitivity analysis of the underlying mathematical formulation, and some features have been added during the last years such as 3D graphics and a real-time driving simulator.

In this project, a further development of MBS3D has been carried out. In particular, it is oriented to the extension of the software in order to be able to simulate hybrid and electric powertrains. Therefore, this work is characterized by the study of MBS3D from a new point of view: the co-simulation of models of different areas of expertise. The model for this research consists of a mechanical subsystem, corresponding to the multibody dynamic model of the complete car, and an electrical subsystem, corresponding to the vehicle powertrain.

2.1. Main Objective

The main objective of this project focuses on the application of co-simulation methodologies to the multi-domain dynamic model of an electric vehicle. For this purpose, it is intended to design and program the mathematical model of the electric powertrain and add it to the multibody system equations available in MBS3D. The numerical integration of the differential equations from the mechanical and electrical components will be done using MATLAB and, subsequently, a detailed analysis of the results will be performed. The implementation of co-simulation methodologies to the complete vehicle model will allow the study of the interaction between sub-models from different physical domains and experiment new techniques for the optimization of the integration process.

2.2. Secondary Objectives

The secondary objectives of this work are derived from both the mathematical tools and the electric vehicle model characteristics. These objectives are:

1. To carry out a detailed study of the state of the art, in order to understand the existing hybrid and electric vehicle technologies, the electric and non-electric propulsion systems, the control strategies and the modeling tools and techniques used for EVs-HEVs.
2. To develop and validate a mathematical model of an electric drivetrain with an equation-based approach.
3. To program all the necessary MATLAB and C/C++ functions for modeling all the electrical components of the powertrain: electric motor, battery and power converter.
4. To obtain real vehicle performance by adjusting the parameters of the electric powertrain components and simulating the driver behavior through steering, brake and gas pedal input control.
5. To evaluate the response of electric differential equations when they are solved with MATLAB integrators.
6. To optimize the programming code and improve the performance by introducing programs written in C/C++ language, through MEX file function implementation.
7. To test and compare different numerical ODE integrators by varying the initial conditions of the problem, as well as the parameters for the integration such as absolute and relative numerical errors, simulation time frame or number of integration steps.
8. To assess vehicle performance characteristics under different values of the battery state of charge or variable power requirement; and to design and simulate a driving cycle in an urban environment by varying the brake and gas pedal inputs.
3. Methodology

The methodology followed in developing the present work includes many different stages, from the theoretical study of hybrid and electric vehicle technologies, modeling approaches and simulation tools, to the mathematical analysis of differential equations, as well as programming and simulation tasks.

3.1. Creation of the Equation-Based Electric Vehicle Model

3.1.1. Study of Hybrid and Electric Vehicle Technologies

To build the vehicle model, a previous research of the existing hybrid and electric technologies was done. A large amount of bibliography was consulted and carefully studied, as well as some specialized articles and scientific journals available at INSIA. The main conclusions from this study were the classification of electric and hybrid power trains and their control strategies, which will be explained in later chapters. The major types of hybrid and electric vehicles (full electric, series and parallel hybrid electric vehicles), their energy management strategies and modes of operation (hybrid mode, electric mode and regenerative braking) were analyzed.

The original idea was to model a series hybrid power train, since it included elements from very diverse domains (internal combustion engine, battery, electric motor and mechanical transmission) and, at the same time, it is the simplest hybrid model with easy to implement control strategies. However, the inclusion of the internal combustion engine presented some difficulties, for it was necessary to insert efficiency maps in MATLAB in addition to the differential equations of the electric components. Therefore, the possibility of modeling a full electric drive train was considered and this option was finally chosen. After clearly understanding the operation of every component of the electric power train, it was found that an equation-based model of the battery pack, electric motor and power converter, would be broad enough to meet the main goal of this project. The co-simulation methodologies could be applied to two sub-models from different physical domains: the electric model of the power train components and the mechanical approach of the vehicle components given by MBS3D equations.

From the existing electric vehicle configurations, one of the most common arrangements consists of one electric motor drive with clutch-less single-gear transmission and differential. It is represented in Figure 3.1. The motor is powered by a battery pack which can be recharged through regenerative braking and with electricity provided from the grid. In addition, a power converter is needed for balancing the energy exchange between the battery and the electric motor. This configuration has been chosen for this work due to the simplicity of a single-gear transmission instead of a multi-gear one, and the clear power flow connections of the electrical components. The battery, by means of the power converter, delivers electric current to the motor drive which, in turn, provides mechanical torque to the transmission and the wheels.

![Figure 3.1. Electric vehicle power train configuration](image-url)
3.1.2. Study of Existing Modeling Approaches

As it has already been mentioned, there are many approaches for modeling hybrid and electric power trains. However, from the point of view of the traction power flow, two major calculation methods must be highlighted when considering a numerical simulation of the system: backward-facing or quasistatic approach and forward-facing or dynamics approach.

In the quasistatic approach, the driving cycle is an input to the model, as well as the vehicle acceleration and speed. Based on this information, the force required to meet the speed profile is directly calculated. This force is then translated into a torque that must be provided by the motor. The motor will deliver the torque at a certain rotational speed which is calculated by means of its efficiency map. In this way, component by component, the energy required from the battery is computed, carrying out the calculation against the traction power stream of the drivetrain (Keith B. Wipke, Matthew R. Cuddy, and Steven D. Burch, 1999). This modeling method simplifies the mathematical equations by considering quasi-stationary behavior of the electrical components. However, it was not chosen for the present work, mainly because it does not respect the physical causality of the power flow, which is an essential requirement for coupling the electric model with the multibody dynamic system in MBS3D.

The dynamic approach uses a forward-facing calculation or a “correct” physical description of the powertrain model. The inputs to the model are the acceleration and brake commands from the driver, which are translated into a torque delivered by the motor. The torque is input to the transmission and passed forward to the wheels in the direction of the traction power flow. Usually, the vehicle is modeled with a set of ordinary differential equations or even differential-algebraic equations (DAE). This modeling approach involves an important computational burden but is much closer to the actual vehicle behavior than the quasistatic method, since the inputs to the model are the same as those in the real system. Besides, in order to do a mathematical study of the complete electric vehicle model and a compare different co-simulation methods and integrators, both sub-models (electric and mechanical) need to have similar structures and even the same calculation approach. Therefore, since MBS3D software offers an ODE-based model of a mechanical multibody system, a dynamic modeling technique had to be followed for the electrical model too.

The dynamic approach was implemented by dividing the powertrain into independent modules, which were modeled separately. Each component of the drive train is represented by a submodule and has inputs and outputs from the upstream and downstream submodules. The transient behavior of every sub-model is defined by its own ODE or DAE system. Once all the components were modeled, the submodules were arranged and joined so that the complete vehicle model could be analyzed and simulated.

3.1.3. Selection of the Electric Vehicle Equations

In the bibliography, a number of equation-based models were found that had been implemented for simulation purposes. However, the main goal was to find an ordinary differential equation system or a set of differential-algebraic equations that could be easily transformed into an ODE one. After reviewing the literature, an optimal model for coupling with MBS3D was selected.

For the electric motor, two dynamic modeling options were available: a DAE/ODE model or one based in steady-state equations. The second option was rejected, since it focuses on control strategies such as direct torque control and field oriented control, which are outside the scope of this project. Therefore, a DAE/ODE model of the electric motor was considered. Usually, the dynamic model of electrical components is based on an equivalent circuit, and the differential equations are obtained by applying Kirchhoff voltage and current laws, as well
as power balances across the circuit components. In addition, the dynamic equations of the power converter could also be drawn from the electric motor equivalent circuit.

Two options were also available for building the dynamic model of the battery: an electrochemical model and an electric equivalent circuit model. The first is based on the physical behavior of the battery and considers the chemical processes that take place inside the battery cells. The second is an empirical approach, where the values of the circuit components such as resistance and inductance are obtained through experimental tests. This empirical data is usually provided by the manufacturer. The second option was selected, since a set of differential equations could be easily deduced from the analysis of the equivalent circuit.

Following this methodology, mainly based on equivalent electrical circuits, the physical causality representation of the dynamic sub-models is simple and adaptable to MBS3D software requirements.

The selected equations were later verified with professors from the electrical machines laboratory and from specialized master’s programs in hybrid vehicles offered at INSIA.

### 3.2. Study of MBS3D 2.0 Simulation Tool

In order to be able to couple the electric vehicle equations with the existing mechanical equations in MBS3D, it was necessary to achieve a thorough understanding of the simulation tool and its theoretical basis. For this purpose, a detailed study of MBS3D was done.

MBS3D 2.0 software is based on a semi-recursive dynamic formulation for the simulation of multibody system dynamics. The formulation, originally applied to a DAE system, results in a set of ordinary differential equations in matrix form which describe the dynamic behavior of the bodies constituting the multibody system. The solutions to this system of equations are the positions, velocities and accelerations of each body. A detailed description of MBS3D underlying mathematical formulation was published in “Improved Semi-Recursive Dynamic Formulation for the Dynamic Simulation of Multibody Systems” (J. García de Jalón, A. Hidalgo and A. Callejo, 2011).

Later, the MBS3D program was installed on the computer and the main functions were explored. Additionally, some examples of multibody systems are available with the program and they were run and debugged in order to evaluate and understand MBS3D properties. During this process, an intermediate ability with MATLAB and a basic knowledge of its integrators were acquired. After learning the working method of MBS3D 2.0, the author decided which functions should be modified and/or added to this software to achieve the goals of the current project.

Once the topology of the MBS3D car model was clearly understood, the differential-algebraic equations corresponding to the electric motor, the battery and the power converter were transformed into a set of ordinary differential equations and then adapted to the form of MBS3D equations. To do this, the known and unknown variables in the electrical system of equations were identified and classified as constant or time varying parameters. Next, it was verified that the number of unknowns was equal to the number of equations. In this way, the electrical system could be studied and then coupled with the MBS3D mechanical equations.

### 3.3. Programming of the Electric Vehicle Model in MATLAB

Before coupling the electric powertrain with the mechanical drivetrain, it was decided to test the electrical differential equations alone. The aim at this stage was to obtain a vehicle model
whose performance characteristics were as close as possible to the real system. This would save a lot of time and programming efforts when coupling both models.

To this end, the equations of the electric motor, the battery and the power converter were programmed in MATLAB code. For the integration, standard MATLAB integrators were used. After an initial assessment of the integration results, the parameters of the differential equations were adjusted until the integration results were similar to the real power train performance.

3.4. Co-simulation of MBS3D and Electric Vehicle Model

After testing and adjusting the electric vehicle subsystem, it was necessary to find a way to co-simulate both the electrical and the mechanical models in MATLAB programming environment.

Initially, two main co-simulation methodologies were considered: weak-coupling and tight-coupling. In the first method, two or more integrators are used to solve the different subsystems and values are exchanged once every subsystem has been integrated. In tight coupling, one single integrator is used to solve simultaneously all the subsystems and values are exchanged by function-call during the integration. Although both methods were applicable in MATLAB, tight-coupling approach was selected, since the integration of the equations under the same integrator would allow a comparison between different MATLAB integrators more easily.

Tight-coupling method implementation involved an extension of MBS3D structure to include the electrical differential equations. The libraries corresponding to the electric vehicle model were added to MBS3D libraries and both codes were coupled. New functions created by the user were added to the original program and some were modified and adapted to the new integration requirements.

3.5. Mex File Function Implementation

Efficiency is one of the essential features of mathematical models for simulation purposes. Therefore, the last stage of this work was to find ways to improve the integration process and speed up the simulation of the electric vehicle model.

MBS3D software includes a professional version called MBS3D 2.0 Pro, where some of the core functions are written in C/C++ code. When these functions are built as MEX-functions, they can be called from MATLAB in spite of being programmed in C/C++. It has been proven that, through this technique, the simulation can be up to two orders of magnitude faster. Consequently, the most direct method for improving the efficiency of the program consisted of translating the created MATLAB files into C++ code.

For a detailed explanation of the process described above, refer to “MEX File Functions” and “Translation into C++ Code” in later chapters.

3.6. Note Regarding Programming Code

The present project has an important computational burden, which can create some difficulties. The author intended to write this document in a language easily understood by a wider audience. Nevertheless, given the nature of this work, a deep understanding of the main topics and tasks requires basic programming knowledge. An effort has been made in order to avoid the inclusion of programming code in this document. However, sometimes it has been necessary for illustrating and clarifying the explanation. In these cases, MATLAB
source code fragments are written in the Courier-type font and, for C++ source code, the Consolas-type font is used. Some examples are given below:

- MATLAB code:

```matlab
function tau = wheelTorquesHE (t, Tnode, TbodyTnode, yElec, mbsOptions)
```

- C++ code:

```c
void WheelTorquesC (double *tauW, double t, struct cnode *Cnode, int *TbodyTnode)
```
4. Basic Longitudinal Dynamics of Vehicles

Basic models for the study of longitudinal vehicle motion include two main elements: vehicle dynamics and powertrain dynamics. In this chapter, the external forces that influence vehicle longitudinal dynamics will be studied: aerodynamic drag, rolling resistance forces, gravitational forces and longitudinal tire forces.

Vehicle performance includes the calculation of maximum acceleration, maximum cruising speed and gradeability. To obtain these parameters it is essential to know the longitudinal forces affecting a vehicle (Figure 4.1). The study of longitudinal dynamics is based on a simple two-dimensional model where lateral forces and vertical acceleration are neglected for the time being. Longitudinal dynamic behavior includes both traction and braking forces. However, in this text only traction forces will be explained, as this section shall only be a quick review to help understand internal combustion engine basic requirements.

The base of longitudinal performance of a vehicle will be the fundamental equation of longitudinal motion, where the dynamic features of the vehicle and the impact of the external environment are quantified. This will involve the evaluation of resistances that counteract movement, as a way to calculate the actual performance parameters of the vehicle and the influence of adherence on tire-road contact surface.

For the calculation, Newton’s second law is applied in longitudinal direction:

\[ \sum F_x = ma_x \]  

(4.1)

where \( \sum F \) = Traction force – Resistance forces.

4.1. Traction Force

Tractive effort is the force applied on the wheels at the road surface in order to move the vehicle. The available traction force is limited by two factors:

a) Tractive effort generated by the drive system.
b) Weight distribution of the vehicle and adherence (\(\mu\)) tire-road.

In general the drive system limits tractive capacity at high speeds, while adherence imposes a limit when running at low speeds. The most restrictive condition will determine the maximum traction force.

### 4.1.1. Drive System-generated Tractive Effort

The drive system includes: vehicle engine, gearbox and power transmission system.

The ideal engine power-torque or traction curve is represented in Figure 4.2. However, internal combustion engines do not benefit this behavior and their performance is shown in Figure 4.3. With a view to adapt the performance of internal combustion engines to the ideal traction curve, a gearbox is added to the drive system. This allows ICEs to overlap their behavior, approaching the ideal curve.

*Figure 4.2. Ideal engine power-torque curve, typical of EV motors*

*Figure 4.3. Power-torque curve for a conventional IC engine*
The force generated by the engine is given by the formula:

\[ F_e = \frac{M_e \cdot \varepsilon_0 \cdot \eta_d}{r} \]  \hspace{1cm} (4.2)

where

- \( M_e \): engine torque
- \( \varepsilon_0 \): transmission ratio
- \( \eta_d \): overall driveline efficiency
- \( r \): tire rolling radius

Therefore, the available tractive effort based on engine torque is a function of the maximum torque that the engine can generate and that can be transmitted to the wheels by the transmission system (Figure 4.4).

![Figure 4.4. Tractive effort of a conventional IC engine](Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010)

Although internal combustion engine characteristics are far from the ideal performance, their use is widespread. This is, among others, because of the following reasons (Pablo Luque Rodríguez, 2003):

- Good power/weight ratio.
- Acceptable fuel consumption.
- Low price.
- Economic fuel and, thus far, relatively abundant.
- Wide autonomy with volume reduced fuel tanks.
- Possibility to control operating regime easily.
- Low maintenance.

Despite the advantages, the negative environmental impact of vehicles equipped with internal combustion engines (noise, vibrations and gas emissions) is an essential point to consider. In addition, the oil crisis has triggered intense research with two principal purposes:
improving current internal combustion engines, making them more efficient and less polluting, and developing new propulsion technologies for vehicles.

4.1.2. Maximum Tractive Effort Based on Adherence Limit

The maximum permissible tractive effort that can be applied on the wheels is limited by friction between road and tire. This force is quantified using a maximum coefficient of adhesion $\mu_{\max}$.

Also, the limit imposed by adherence on vehicle traction varies depending on vehicle transmission system. It is therefore different in front wheel drive, rear wheel drive and four wheel drive vehicles.

![Typical model for evaluation of traction force capacity](image)

A typical model for the study of tractive effort capacity is shown in Figure 4.5.

Now, pitching moments and variations in the suspension system are not taken into account. After applying the equilibrium condition in longitudinal direction, the maximum tractive effort limited by adherence is (Francisco Aparicio Izquierdo, Carlos Vera, Vicente Díaz López, 1995):

- Front wheel drive vehicles: $F_{T_{\max}} = \frac{\mu \cdot P \cdot (l_2 + h \cdot f_r)}{B + \mu \cdot h}$ \hspace{1cm} (4.3)
- Rear wheel drive vehicles: $F_{T_{\max}} = \frac{\mu \cdot P \cdot (l_1 + h \cdot f_r)}{B + \mu \cdot h}$ \hspace{1cm} (4.4)
- Four wheel drive vehicles: $F_{T_{\max}} = \mu \cdot P \cdot \cos \theta$ \hspace{1cm} (4.5)

4.2. Motion Resistance Forces

There are three types of vehicle forward motion resistance forces:

- Aerodynamic resistance ($R_a$)
- Rolling resistance ($R_r$)
- Grading resistance ($R_g$)

Then, the total motion resistance is:

$$R_T = R_a + R_r + R_g$$ \hspace{1cm} (4.6)
4.2.1. Aerodynamic Resistance

Aerodynamic drag or resistance is produced when the vehicle runs through a fluid medium. It depends on the external air flow around the vehicle body and on the inside air circulation. Both flows create friction and pressure resistance.

\[ R_a = \frac{1}{2} \cdot \rho \cdot C_x \cdot A_f \cdot V^2 \]  

(4.7)

where

- \( C_x \): coefficient of aerodynamic resistance (depends on the shape of the vehicle body)
- \( A_f \): vehicle frontal area
- \( \rho \): air density
- \( V \): vehicle speed

### Table 4.1. Typical values of the coefficient of aerodynamic resistance for different types of vehicle

<table>
<thead>
<tr>
<th>Type of vehicle</th>
<th>Drag coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trucks</td>
<td>0.050</td>
</tr>
<tr>
<td>Articulated vehicles and road trains</td>
<td>0.095</td>
</tr>
<tr>
<td>Buses</td>
<td>0.038</td>
</tr>
<tr>
<td>Buses with an aerodynamic profile</td>
<td>0.019</td>
</tr>
<tr>
<td>Conventional passenger cars</td>
<td>0.022 – 0.035</td>
</tr>
<tr>
<td>Passenger cars with an aerodynamic profile</td>
<td>0.010 – 0.019</td>
</tr>
</tbody>
</table>

The density of air in normal temperature and pressure conditions (25°C and 1.074 Pa) is \( \rho = 1.225 \text{ kg/m}^3 \). This value varies with altitude and kinematic viscosity. \( A_f \) is calculated according to vehicle dimensions (Figure 4.6): \( A_f = f \cdot b \cdot h \), where \( f \) has a value between 0.8 and 0.85.

![Figure 4.6. Model for the calculation of vehicle frontal area](image)
4.2.2. Rolling Resistance

Rolling resistance forces are due to hysteresis in the tires. This resistance counteracts thrust force and its value depends on vehicle mass, steering geometry, profile, type and pressure of tires, driving speed, surface and state of the road.

It is calculated by multiplying the normal load on the center of each wheel \( P \) by the rolling resistance coefficient \( f_r \), whose value depends on the material and environmental factors.

\[
R_r = (f_0 + f_v V^n) \cdot P = f_r \cdot P \tag{4.8}
\]

where
- \( f_0 \) and \( f_v \): are parameters that depend fundamentally on the inflation pressure
- \( n \): is an empirical factor that varies between 2 and 2.5
- \( P \) is the vehicle weight

Figure 4.7 represents the ideal rolling resistance condition and a non-ideal situation, where the wheel is deformed by the surface.

Rolling resistance will increase with tire flexing work, air resistance and friction effects on the tire tread. Figure 4.8 shows the variation of rolling resistance coefficient on different surfaces.

Figure 4.7. a) Soft wheel rolling on a hard surface. The surface deforms the wheel. b) Ideal rolling resistance condition (Ortega, 2011)

Figure 4.8. Rolling resistance coefficient for various materials and different inflation pressure (Wong, 1993)
4.2.3. Grading Resistance

Grading resistance is due to the component of the weight opposing the movement of the vehicle when driving on a surface with a certain degree of steepness. The gravitational force acting on the vehicle is:

\[ R_g = P \cdot \sin \theta \]  \hspace{1cm} (4.9)

where
- \( P \): weight of the vehicle
- \( \theta \): inclination angle to the horizontal. If \( \theta > 0 \), the force counteracts vehicle motion. If \( \theta < 0 \) it is a driving force.

\( \theta \) is usually \( \leq 10^\circ \) in normal roads. Then, \( \sin \theta = \tan \theta \cdot \), “\( j \)” being the slope expressed in parts per unit.

4.3. Vehicle Power Requirements

In order to overcome the mentioned motion resistances, the necessary power for the vehicle will be:

\[ H = V \cdot (P \cdot \sin \theta + f_r \cdot P \cdot \cos \theta + \frac{1}{2} \rho \cdot C_x \cdot A_f \cdot V^2) \]  \hspace{1cm} (4.10)

This enables the circulation at a constant velocity. From the equation above it can be concluded that, the vehicle will reach the maximum velocity when the engine works at full load or full throttle.
Basic Longitudinal Dynamics of Vehicles
5. Electric and Hybrid Electric Vehicles: Classification, Control Strategies and Brake Energy Regeneration

5.1. Full Electric Vehicles

Hybrid electric vehicles are partially or totally propelled by an electric motor powered by batteries. In full electric vehicles, an electric motor alone delivers traction power to the wheels. The motor is provided with electrical current from a battery pack, which is charged through an external electrical outlet. Therefore, this type of vehicle does not use an internal combustion engine.

However, some of these cars include an internal combustion engine which generates electric power for charging the battery. They are known as range extended electric vehicles (RE-EV), since the engine simply extends battery autonomy and does not produce traction power. As in normal full electric vehicles, the electric motor is the only propulsion source.

Additionally there are many possible configurations of the electric vehicle drive train, depending on the propulsion technology, storage device and transmission system. The main configurations are represented in Figure 5.1.

![Figure 5.1. Possible configurations of EVs: (a) multi-gear transmission and clutch, (b) single-gear transmission without clutch, (c) integrated fixed gearing and differential, (d) two electric motors and fixed gearing, (e) direct drive with two separate motors and fixed gearing, and (f) two in-wheel motor drives (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010)
5.2. Hybrid Electric Vehicles

Hybrid electric vehicles are often classified according to two main criteria: topology and hybridization degree. Topology refers to the configuration of the ICE and the electric motor along the vehicle powertrain. The hybridization degree is associated with the powertrain ability to produce electric power for vehicle traction.

5.2.1. According to Topology

Traditionally, hybrid electric vehicles have been classified into two basic configurations, series and parallel, depending on the type of arrangement and connections between the drive train components.

Series-parallel drivetrain combinations have shown further advantages in terms of operation modes and efficiency, although the architecture is apparently more complicated.

**Series HEVs:**

A series HEV is one in which the electric motor alone propels the vehicle. Figure 5.2 shows a typical configuration for a series hybrid. A fuel tank is the main energy source and it is connected to the internal combustion engine (ICE). The ICE drives a generator that, in turn, delivers energy to the battery pack or energy storage device through a rectifier and a DC bus. These last two components form an electrical coupler between the ICE and the traction motor (electric motor). The bidirectional DC/DC converter is needed due to the presence in the drive train of more than one electric source: battery pack and generator. DC/DC converters and other types of multi-input converters, as well as dual-voltage and multi-level systems have been developed to control the performance of several electric sources working at diverse voltage levels (Lino Guzzella, Antonio Sciarretta, 2007).

The traction motor can function either as a motor or a generator. When it acts as a generator, the power flows from the electric motor to the batteries and in the opposite direction when it is controlled as a motor.

![Figure 5.2. Series hybrid electric drivetrain (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010)](image)
The battery charger is installed in case the battery pack needs to be charged from the power grid and, therefore, the vehicle is also given a plug-in capability.

Vehicles with a series hybrid drive train offer the possibility to function according to the following modes (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010):

1. **Pure electric traction mode**: The propulsion power required to move the vehicle flows from the battery pack. Then, the vehicle only uses electric power and the ICE is turned off.

2. **Pure engine traction mode**: The internal combustion engine together with the generator provides the propulsion power and no operation is carried out by the batteries. The electric machine serves as an electric transmission from the engine to the driven wheels.

3. **Hybrid traction mode**: The electrical coupler blends the power provided from both the engine–generator and the batteries.

4. **Engine traction with battery charging mode**: The internal combustion engine and the generator deliver power to charge the batteries and to drive the vehicle concurrently. In this case, the electrical coupler splits the power coming from the ICE–generator.

5. **Regenerative braking mode**: The ICE–generator is turned off and the traction motor is operated as a generator powered by the vehicle kinetic or potential energy. The power generated is charged to the batteries and reused in later propelling.

6. **Battery charging mode**: The power generated by the ICE-generator is all used to charge the batteries. No power is supplied to the electric motor.

7. **Hybrid battery charging mode**: When braking, both the ICE–generator and the electric motor function as generators to charge the batteries.

**Parallel HEVs:**

A parallel HEV is one in which both the electric motor and the IC engine propel the vehicle. As it is shown in Figure 5.3, the internal combustion engine and the electric motor are linked through a mechanical coupling.

![Figure 5.3. Parallel hybrid electric drivetrain (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010)]
The mechanical coupler is responsible for blending the traction torque or the speed generated in the engine and the motor, which is supplied to the driven wheels by means of a mechanical transmission.

Parallel architecture enables the two power sources to function either simultaneously or individually. This means that a parallel hybrid electric drive train can also operate according to any of the modes explained above for series hybrid drive trains.

- **Series-parallel combination HEVs:**

Series-parallel hybrid electric vehicles maintain the parallel drive train architecture plus a power split device (Figure 5.4). Depending on the driving condition, the power split device delivers IC engine torque to the front wheels by first separating a calculated percentage and supplying it to the batteries by way of the generator.

A control unit is needed to allocate the energy flow through the different drive train components, thus achieving the highest efficiency level.

As in the case of parallel hybrid electric drive trains, the electric motor and the internal combustion engine can function together or individually. During the vehicle launch or when it moves at low speeds, the electric motor alone drives the wheels, providing power through the drive-shaft. The combustion engine turns on when the speed increases and, consequently, the required power to move the vehicle is higher.

The battery is charged via the generator, with the power flowing from the IC engine, or by recovering the kinetic energy produced when braking.

**5.2.2. According to Degree of Hybridization**

Another criteria used to classify HEVs is one based on the degree of hybridization. Degree of hybridization is defined as the ratio between the electric power of the vehicle and the total traction torque. Accordingly, the following types of hybrid electric vehicles are described:

**Micro HEVs:**

Micro-Hybrid vehicles are usually provided with a technology known as “Start-Stop”, which enables the engine to turn off when the vehicle stops and to restart when the accelerator is pressed or the braking pedal is released. The “Start-Stop” operation is powered by the energy stored in the battery during braking. This represents a fuel-saving way of driving as well as a reduction of CO₂ emissions. It is not rigorously classified as a hybrid electric vehicle.
vehicle, since the power supplied by the electric motor is not used for the wheel traction, but for starting the IC engine again after a shutdown.

**Mild HEVs:**

In this model, the electric motor can be also used to provide propulsion power to the vehicle, although it only acts as a supplement for the engine during acceleration. For that reason, it is not possible for the vehicle to move with the traction torque coming from the electric motor alone. Generally, this kind of vehicles incorporate a regenerative braking system, where the electric motor is in charge of recovering the waste energy while braking and allows the battery recharge. Series and parallel HEVs are comprised in this category. Parallel mild HEVs maintain the “Start-Stop” technology.

**Full HEVs:**

In this case, the vehicle can operate in full electric mode during launch and when the engine runs at low speeds. The IC engine can also propel the vehicle alone. Furthermore, the architecture includes the regenerative braking system for battery recharging, which reclaims much more energy in a full HEV than in a mild HEV. A control unit chooses the most efficient and appropriate propulsion mode. This technology achieves a significant reduction in CO₂ emissions and important fuel savings, especially in urban environments, where the vehicle speed does not usually exceed 50 km/h.

Figure 5.5 shows a possible distribution of traction power between the electric motor and the internal combustion engine in HEVs. In this text, a hybrid electric vehicle will be understood as one with full hybridization degree.

---

**Figure 5.5. Distribution of motive power portions in HEVs (Raf Catthoor, Freek Beeckman, 2011)**
5.3. Energy Management and Control Strategies for EVs and HEVs

A control strategy is a preset rule in the vehicle controller that predicts and determines the operation of every component of the drive train. The control strategy is crucial for hybrid vehicles performance, since it decides what operation modes are proper and safe depending on the driver commands and the feedback from the drive train. The design of the control strategy is mostly influenced by two parameters: energy consumption and emissions. The goal is to balance an efficient energy usage with lower emissions, by finding the ideal operating point of the vehicle's engine and motor.

In order to achieve all the above, several control strategies have been tested and implemented. These strategies are different depending on the configuration of the drive train components, since they do not have the same mission requirements. Therefore, control strategies for series and parallel hybrid electric vehicles will be distinguished.

5.3.1. Series HEV Control Strategies

Two typical control strategies for series hybrid electric vehicles are: 1) Maximum state of charge of the peaking power source (PPS) and 2) Engine on/off or Thermostat control strategies.

Maximum State of Charge of PPS

In this strategy, the engine/generator becomes the primary power source, providing propulsion power to the vehicle and recharging the battery pack when needed. The main goal is to maintain the PPS at its maximum level of charge in order to guarantee high vehicle performance characteristics at any time.

With this control strategy, the drive train operates in four different modes depending on the commanded traction power (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010):

A. Hybrid traction mode: When the traction power command is greater than the maximum engine power, the PPS also delivers energy in order to fulfill the power requirement.

B. Engine/generator alone: When the traction power command is less than the power the engine/generator can produce, two different modes of operation can be defined. In the first mode of operation, if the state of charge is below its maximum level, the engine operates at full load and the power that is not provided for vehicle traction is used to charge the PPS. In the second mode, the engine/generator is controlled to operate in its optimal region, producing only the commanded traction power. In this case the PPS does not accept or deliver electrical current.

C. Hybrid braking mode: When the required braking power is greater than the power that the motor can produce, the mechanical brake provides the remaining power.

D. Regenerative braking mode: When the required braking power is less than the maximum braking power of the motor, it operates alone and only regenerative braking is used.

This strategy becomes a very effective option when driving in urban environments with frequent stop-go driving cycles. The different modes of operation are represented in Figure 5.6.
Engine On-Off or Thermostat Control Strategy

This control strategy determines the operation of the engine/generator depending on the state of charge of the PPS (Figure 5.7). When it is below a certain limit, the engine/generator turns on, part of its power goes to recharge the PPS and the rest is used for vehicle propulsion. When the state of charge reaches a predefined top level, the PPS alone propels the vehicle and the engine is set at idle.

In this way, the engine/generator always produces less power than the optimum so the strategy is only efficient if the PPS is recharged to its top level every time the engine turns on. Therefore, the thermostat control strategy should be applied on vehicles with frequent highway driving patterns, where the vehicle travels at a constant speed for long distances and there is enough time to fully charge the PPS.
5.3.2. Parallel HEV Control Strategies

The two control strategies explained above are also applicable to parallel hybrid electric vehicles. Additionally, two more control strategies have been developed for controlling parallel hybrid drive train performance: 1) Fuzzy logic control technique and 2) Dynamic programming technique.

**Fuzzy Logic Control Strategy**

This control strategy is based on fuzzy logic rules that determine the operation of the engine and the motor depending on the state of charge of the PPS and the commanded traction torque. The value of these two parameters is quantified and expressed with a linguistic sign. For the SOC, the letters used are: high (H), medium (M), low (L), and zero (Z). For the required traction torque the values are: positive high (PH), positive medium (PM), positive low (PL), zero (Z), negative high (NH), negative medium (NM) and negative low (NL). Fuzzy logic rule base is represented in Figure 5.8. The block diagram also shows how the motor torque is obtained from the commanded torque and the torque produced by the engine.

![Figure 5.8. Fuzzy logic control rule base (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010)](image)

**Dynamic Programming Control Strategy**

Dynamic programming technique is based on an analytical or numerical model of the drive train. This model can be solved through a dynamic optimization problem in order to reach minimum fuel consumption for a given driving cycle.

The problem formulation and constraints will not be described in this text.

5.3.3. Full EV Control Strategies

Electric vehicle control basically includes the control of the electric motor. Consequently, the control strategy is different for every type of electric motor. These strategies are applied by adapting the driving power circuit of the electric motor (motor controller) and implemented in a microprocessor. The main control strategies used in electric motors are listed below:

- Pulse Width Modulation (PWM)
- Variable-Voltage Variable-Frequency (VVVF)
- Field Oriented Control (FOC)
- Direct Torque Control (DTC)

The first control strategy is typically used for DC motor speed regulation and the rest are usually implemented in induction motors. A detailed explanation of these electric motor control strategies lies outside the scope of this text.
Furthermore, new high-performance control techniques are being applied for electric vehicle control: fuzzy logic control, expert system and artificial neuro networks.

5.4. Brake Energy Regeneration

One of the most widespread energy-saving technologies in the industry consists of converting the power flow from a certain application into an alternative form of energy that can be useful for the functioning of other systems. This principle is the basis for brake energy regeneration. During conventional vehicle braking, the engine turns kinetic energy into thermal energy, which is dissipated and therefore wasted. However, with an appropriate conversion, the kinetic energy could be absorbed in the form of electrical power that might otherwise be wasted.

In order to achieve an efficient energy use, the regenerative braking system has been introduced in hybrid and electric vehicles. The regenerative brake is a device that permits the conversion of kinetic energy into electric energy during vehicle deceleration.

In the brake system, an alternating current motor is installed, which acts as a generator when deceleration is detected and provides direct current to the battery pack by means of an inverter or a general power converter. The generated electrical energy is stored in batteries or capacitors for later use. Usually, this energy will be returned to the electric motor for vehicle launching or accelerating, or for the operation of auxiliary systems. This allows the electric motor to replace the combustion engine in certain driving conditions (typically launching and low speeds), provided that there is enough power in the batteries.

The energy efficiency of an EV-HEV depends to a large extent on the regenerative braking system at its disposal. The suitable type of regenerative braking is chosen according to vehicle energy needs, that is to say, to the required energy recovery level.

The ideal features seek in regenerative braking systems are:

- High and efficient energy conversion rate.
- Simple power electronics for the connection with the vehicle drive train.
- Rapid response to control inputs during braking.
- Smooth operation and power delivery for propulsion.
- Include an energy storage device with sufficient capacity and low volume.

5.4.1. Types of Regenerative Braking Systems (RBS)

Series Regenerative Braking Systems:

In series braking systems, the brake pedal position determines how the service and regenerative braking systems act. During deceleration, if the pressure applied by the driver on the brake pedal is light enough, the braking torque will be supplied by the electric generator alone. When the pressure increases, the service braking generates at the same time a friction torque in order to support the regenerative braking.

In the case of a sudden braking maneuver or when, due to any other reason the brake pedal is depressed steeply or non-progressively, the anti-lock braking system manages the braking process. Then, the braking torque comes exclusively from the friction brake.

In order to safeguard vehicle longitudinal stability, the process above will involve independent management of the four wheels. To do so, it is necessary to install an actuator or electrohydraulic braking unit. This unit is in charge of maintaining the required braking torque, combining conventional friction with energy recovery technique. The electrohydraulic
system interacts with the brake pedal, carrying out two tasks: balancing braking torque between the vehicle axles and handling brake cylinders pressure. This ensures an adjusted performance of the braking torque in both front and rear axles.

As well, an adequate coordination of the force provided by the braking system enables the reduction of stopping distance and guarantees tire grip at all times.

![Graph showing Series RBS performance](image)

*Figure 5.9. Series RBS performance (Miller, 2008)*

Figure 5.9 represents the performance of a series RBS. At the beginning of the braking process, the vehicle experiments deceleration in proportion to engine compression braking. Later the electric motor (M/G) starts recovering kinetic energy in generating mode. When the required braking torque exceeds some pre-defined point, the service brake engages and adds to regeneration torque. When M/G is no longer efficient for regeneration it will turn off and only service brake will be left (Miller, 2008).

**Parallel Regenerative Braking Systems:**

In parallel regenerative braking technology, service and regenerative brakes always act simultaneously. The fraction of braking torque provided by each system needs to be controlled by means of the generator electronic circuit.

Parallel RBS also requires coordination of the braking torque applied on the front and rear axles so as to guarantee vehicle longitudinal stability.

One of the advantages of this type of layout lies in the possibility of functioning without a full electrohydraulic brake (EHB), since the service brake is activated every time the driver depresses the brake pedal. Figure 5.10 shows this behavior in parallel RBS.
Since synchronization between regenerative and service brakes is not essential, the electronic control algorithm becomes simpler and, therefore, EHB system architecture is less complex. However, due to the tandem operation of both braking technologies, it may not always be possible to recover the heat dissipated in order to charge the vehicle battery pack.

Finally, regenerative braking systems in hybrid and electric vehicles have important advantages and some disadvantages which are described below.

Advantages:
- Regenerative braking systems remove most of the power losses produced in conventional vehicles.
- The brake system has, in turn, the ability to regenerate the absorbed power, so the efficiency losses are also reduced.
- The fuel economy is greatly improved, which means less CO₂ emissions.
- Higher vehicle operating range, since it directly depends on the fuel available in the tank.
- High and uniform efficiency, even at low temperatures.
- The batteries do not need to be charged externally.
- Fuel tank downsizing.
- Reduction of engine and brake wear.

Disadvantages:
- Higher weight than a conventional vehicle.
- More complexity of the power train, what makes maintenance difficult.
- High cost of the equipment, components and accessories.
Electric and Hybrid Electric Vehicles: Classification, Control Strategies and Brake Energy Regeneration
6. Electric and Non-electric Propulsion Technologies for EVs and HEVs

6.1. Electric Propulsion Systems

6.1.1. EV-HEV Motor Requirements

In EVs-HEVs, the electric traction motor is in charge of driving the wheels of the vehicle. The motor is, therefore, the main car propulsion unit. Low acoustic noise, driving comfort and control, low maintenance, reasonable cost and high efficiency are some of the characteristics that drivers would expect from a hybrid powertrain. Along with these features, the main technical requirements for electric motors in EVs-HEVs may be summarized in the following points:

- Flexible management of source voltage fluctuations.
- Fault tolerance.
- Ability to proportionate full torque at low speeds.
- High power-to-weight ratio.
- Rugged structure.
- Good acceleration performance.
- Prolonged constant power region of operation.

For EV-HEV applications, the electric motor is designed in such a way that it can also work as a generator. This enables kinetic energy recovery during braking.

The ideal characteristics for a traction motor are shown in Figure 6.1.

![Figure 6.1. Ideal motor force-speed profile of an electric motor (Z. Rahman, M. Ehsani and K. L. Butler, 2000)](image)

Two different areas can be observed: constant power region and constant force region. Initial acceleration will determine the total traction power of a vehicle. This acceleration value should be reached with minimum traction power.
Since overall vehicle operating point is not completely defined in EV-HEV applications, selection of electric motors for this purpose is a complex decision. From EV-HEV motor requirements explained above, it can be concluded that the choice must be based on energy efficiency, which determines fuel economy, and acceleration performance, usually dictated by the driver’s expectations.

The major types of traction motors so far are: brushless DC motor (BLDC), permanent magnet (PM) synchronous motor, switched reluctance (SR) motor and AC induction motor (IM). Their peak power for EV applications is usually in the range from 20 kW to 50 kW.

6.1.2. Brushless DC Motor

As its very name indicates, a brushless DC motor does not use brushes to invert the polarity in the rotor. The structure of a BLDC motor includes a rotor, typically a permanent magnet, and a stator, consisting of a coil arrangement. The rotor of a BLDC motor is a permanent magnet and the stator consist of a coil arrangement. When electric current is applied to the coil, it electrifies and becomes an electromagnet.

The operation of a BLDC is based on a simple force interaction between the permanent magnet and the electromagnet.

Brushless DC motor operation is reflected in Figure 6.2. When current is first applied to coil A, a magnetic field is created between the opposite poles of rotor and stator. In this condition, the attraction between opposite poles causes the rotor movement. As the rotor approaches the energized coil A, coil B is thrilled. Then, current energizes coil C similarly. After this, the first coil is energized again with the inverse polarity. The process is repeated and the rotor continues its rotary motion.

The position of the rotor is detected by a sensor and, based on this information the controller decides which coils should be energized. Typically, a Hall Effect sensor is used for this purpose.

As additional advantages, BLDC motors have a constant torque nature and are lighter than brushed conventional motors (Learn Engineering Website, 2015).

![Brushless DC motor](image)

Figure 6.2. Brushless DC motor (Parker, 2012)

6.1.3. Permanent Magnet Synchronous Motor

PMSM are synchronous electric motors that base their operation on the combination of permanent nature magnetic fields, that is to say, the electromagnetic field is produced by permanent magnets.
The most important disadvantage of PMS motors is the great cost of the permanent magnets, which also present bad fatigue behavior at high temperatures. They are mostly used in applications with moderate power requirements, being possible to find them in some industries with higher energy demands.

PMS machines have a stator built with a three-phase winding distribution around the circumference of the machine in a sinusoidal arrangement (120° phase displacement between each winding). The rotor is provided with permanent magnets. When current is applied in one of the stator phases, a linear magnetic flux is created. As a result of the phase displacement between each winding phase and the rotary motion of the rotor, the resulting magneto motive force vector in the air gap will be a rotating one.

Torque is caused by the combination of stator and rotor magnetic fields and is transmitted by a shaft. In order to ensure synchronism, the amplitude of stator currents are controlled and oriented on the basis of rotor position information. This technique is known as “field oriented control”.

6.1.4. Switched Reluctance Motor

SR motors are classified under stepper motors. The structure of switched reluctance motors consists of rotor and stator; both parts are made of magnetic steel layers and provided with salient poles. Additionally, the stator has concentrated windings, while there is no permanent magnet or coil on the rotor.

Its operating principle is based on variable reluctance between rotor and stator poles. This is due to its doubly-salient structure, which causes fluctuations in the flux path.

The rotor motion is produced as a result of its tendency to stay in a position with minimum reluctance. In SRM, it is the rotor tendency to rotate towards the energized stator poles what produces torque. With a correct synchronization of winding phases and rotor poles it is possible to create a continuous torque profile.

6.1.5. AC Induction Motor

AC induction motors are the most commonly used electrical machines. They are robust, economic and low maintenance motors.

The stator is built piling up thin iron laminations mechanized with slots. The pile is placed inside a steel cylinder. An arrangement of coils or winding is rolled through the slots. The rotor is a squirrel-cage.

When power is applied to the coils, a rotating magnetic field is created. The current is induced by the squirrel-cage rotor due to electromagnetic induction. One of the advantages of this type of motors is that they have inherent automatic start-up.

The mechanic rotary power is transmitted by means of a power axle. Energy losses produced during motor operation are dissipated in the form of heat. Therefore, induction motors require a fan to evacuate the thermal energy.

Figure 6.3 represents the electric motor technologies explained above.
6.2. Non-electric Propulsion Systems

In regard to the non-electric energy source, the most common systems used for HEVs are internal combustion engines and fuel cells.

6.2.1. Internal Combustion Engines

The ICE is a spark ignition engine (gasoline) or compression ignition direct injection (diesel) engine.

a) Spark-ignition engine (SI): in this sort of engines, petrol or gasoline is used as fuel. The working cycle consists of either four (Otto cycle) or two strokes. A spark plug is needed to ignite the fuel. Compression ratios of spark-ignition engines are lower than those in compression-ignition engines.

b) Compression-ignition engine (CI): this type of engines use diesel as their fuel. The diesel fuel self-ignites due to the high temperatures and air compression ratios reached. Therefore, the spark plug is not needed.

The operation of an internal combustion engine (ICE) is based on burning a mixture of air and gasoline. The mixture is compressed inside a chamber or a cylinder increasing pressure and combusting, thus providing linear alternative motion to the piston.

The four-stroke engine is the most common type of ICE. Figure 6.4 shows the main components of a spark-ignition 4S internal combustion engine.

*Figure 6.3. Electric motor technologies for EVs-HEVs (Lino Guzzella, Antonio Sciarretta, 2007)*
The intake valves open to let gases in during intake stroke and once the gas is burnt, the exhaust valve opens to let the exhaust fumes out. The cams on the camshaft are shaped in such a way that, as the shaft turns they push to open the valves enabling the process above.

The valves are attached to springs that return them to their starting position.

In SI engines, the distributor takes electricity from the battery of the vehicle and conducts it to the spark plugs using wires. The spark plugs are pieces of metal with a conductor material or electrode placed in the middle. When the electricity reaches the end of the electrode it produces a spark igniting the fuel above the piston.

Then the piston acquires a linear motion which depends directly on the combustion.

The crankshaft converts the linear motion of the piston into a rotary motion to turn the wheels of the vehicle.

The alternator charges the battery of the vehicle and it is connected to the crankshaft by a belt, so that it rotates simultaneously.

The timing belt that connects the alternator and crankshaft ensures that all the components run in unison.

A typical working cycle for a four-stroke ICE consists of the following steps (Figure 6.5):

1. **Intake stroke**: The intake valve opens to allow fuel and air in (only air in CI engines), using the cam/follower mechanism.

2. **Compression stroke**: As the piston goes up it compresses the air and fuel mixture. When it is completely compressed, a spark plug produces a spark igniting the gases. Ignition pushes the piston down. During this stroke both the intake and outlet valves remain closed. In CI engines, the fuel is injected directly into the cylinder just before the piston moves up to the top dead center.
3. **Expansion or power stroke:** In SI engines the spark burns the mixture and it combusts. However, in CI engines the fuel ignites due to high temperatures in the compressed air. The explosion forces the piston back down and turns the crankshaft at the bottom.

4. **Exhaust stroke:** During the exhaust stroke, the exhaust valve opens using the cam arrangement, allowing the exhaust gases to escape out of the engine chamber. The piston goes back up pushing all the exhaust gases out.

![Image of 5 cycles: induction, compression, ignition, expansion, and exhaust]

*Figure 6.5. Basic 4S cycle (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010)*

**ICE in Conventional Vehicles**

In conventional vehicles, the ICE is designed in order to fulfill performance requirements: maximum cruising speed, acceleration and gradeability.

a) **Maximum speed of a vehicle:**

The cruising speed of a vehicle refers to the maximum speed that it can develop with minimum fuel consumption. When a vehicle is running on a flat road, its maximum speed can be defined as the constant cruising speed that the vehicle can reach with full throttle of engine.

As it has already been exposed in previous chapters, the maximum speed of a vehicle is determined by the equilibrium between the traction force from the engine and the resistance forces. The maximum traction force is limited by the maximum engine speed, which will be determined by the minimum gear ratio.

Assuming vehicle circulation on a horizontal surface ($\Theta = 0$), the maximum speed will be obtained for the maximum power regime. For this speed value, the available power in the driving wheels must be enough to overcome motion resistances:

$$P_{max,engine} \cdot \eta_c = V \cdot (f_r \cdot P + \frac{1}{2} \rho \cdot C_x \cdot A_f \cdot V^2)$$ (6.1)

To obtain the maximum velocity using the minimum gear ratio, the following expression can be applied:

$$V_{max} = \frac{\pi \cdot n_{m1} \cdot r_c}{30 \cdot \varepsilon_q}$$ (6.2)

where
- $n_{m}$: engine speed
- $r_c$: dynamic tire rolling radius
- $\varepsilon_q$: minimum gear ratio

b) Acceleration performance:

The acceleration is usually calculated between 0 and 100 km/h. Vehicle acceleration affects both the travelling speed of its masses ($m$) and the increase of rotation speed of its rotating masses. This effect is often evaluated using a standardized empirical formula in order to calculate an equivalent mass factor:

$$\gamma_m = 1.04 + 0.0025 \cdot \varepsilon_j^2$$  \hspace{1cm} (6.3)

The mass factor accounts for inertia of vehicle’s rotating parts.

This leads to:

$$F_{da} = \gamma_m \cdot m \cdot a$$  \hspace{1cm} (6.4)

$$a(V, \varepsilon_j, \theta) = \frac{F_{da}(V, \varepsilon_j, \theta)}{\gamma_m \cdot m}$$  \hspace{1cm} (6.5)

Table 6.1 shows typical values of the mass factor according to vehicle size and gear transmission ratio.

<table>
<thead>
<tr>
<th>Gear ratios</th>
<th>High</th>
<th>Second gear</th>
<th>First gear</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big passenger car</td>
<td>1.09</td>
<td>1.14</td>
<td>1.30</td>
<td>-</td>
</tr>
<tr>
<td>Small passenger car</td>
<td>1.11</td>
<td>1.20</td>
<td>1.50</td>
<td>2.40</td>
</tr>
<tr>
<td>Truck</td>
<td>1.09</td>
<td>1.20</td>
<td>1.60</td>
<td>2.50</td>
</tr>
</tbody>
</table>

Table 6.1. Averaged values of equivalent mass factor

c) Gradeability:

Gradeability is defined as the grade that a vehicle can overcome assuming that it ascends at a constant speed. Due to the low speed, the aerodynamic resistance is not considered.

$$F_{T_{max}} = R_T = P \cdot \sin \theta + P \cdot f_r$$  \hspace{1cm} (6.6)

$$\theta = \sin^{-1} \frac{F_{T_{max}} - P \cdot f_r}{P}$$  \hspace{1cm} (6.7)
Simplifying:

\[ j = \frac{F_{\text{max}} - P \cdot f_r}{P} \]  

(6.8)

**ICE in Hybrid and Electric Vehicles**

In hybrid and electric vehicles, the combustion engine is not the primary propulsion force, since what delivers traction power to the wheels is the electric motor. The ICE only acts as a supplementary power source.

As in full hybrid vehicles, the ICE can be combined with a battery pack or peaking power source in order to provide enhanced range, lower emission and improved fuel consumption. As it has already been explained, the ICE is arranged in the power train in a series or parallel configuration with the electric motor. Furthermore, the ICE is typically used for recharging the battery. Then, no external electrical supply is required in these vehicles.

Full electric vehicles are not provided with an internal combustion engine, so the battery pack or energy storage device needs to be charged from the power grid.

**6.2.2. Fuel Cells**

A fuel cell is an electrochemical device that produces electricity through a controlled chemical reaction. The electricity is provided directly to an external circuit. Fuel cells are similar to batteries. However, they are designed to provide electricity continuously and their power supply is not limited to a certain storage capacity, as is the case in batteries. This means that fuel cells can generate energy as long as fuel is supplied, while batteries need to be recharged.

In addition, battery electrodes react and change depending on the state of charge, while in fuel cells the electrodes are catalytic and relatively stable.

Nowadays, fuel cell-powered EV are acquiring higher interest, since they present longer driving times without requiring as frequent recharges as battery-powered EV. Furthermore, the chemical process inside a fuel cell has high efficiency and a minimum impact on the environment.

**Basic Operation of a Fuel Cell**

The basic architecture of a fuel cell (Figure 6.6) consists of an electrolytic membrane and two electrodes: the anode or positive electrode, where the oxidation takes places, and the cathode or negative electrode, where the oxidizing agent is reduced.

The chemical reagents present in the reaction are oxygen and hydrogen. Hydrogen acts as fuel while oxygen is directly obtained from air. Other types of fuel such as methane, methanol, ethanol or diesel can also be used with this purpose, since they contain hydrogen in their formulation.
Fuel is supplied to the anode and it flows to the cathode. The electrolyte ionizes the hydrogen letting only the positive ions pass through the membrane. Since the electrons cannot go through the electrolyte, they are forced to leave the anode by means of an external circuit in the form of electrical current.

After, as oxygen flows inside the cathode, it combines with the positive ions to form water undergoing an exothermic reaction. Therefore, the chemical reaction products are water and heat.

The reaction takes place inside the cell itself. Water production happens in different parts of the cell depending on the electrolyte used.

**Fuel Supply**

The major challenge for fuel cell vehicle applications is hydrogen production and storage. Water electrolysis and steam reforming are currently the most considered technologies in commercial hydrogen production.

In steam reforming, hydrocarbon fuels and water steam undergo a chemical reaction at high temperature thus producing hydrogen.

As it has already been described, electrolysis of water takes place when an electric current passing through it induces decomposition into oxygen ($O_2$) and hydrogen ($H_2$). This technique is used to obtain hydrogen fuel.

Compressed hydrogen in a container at ambient temperature, cryogenic liquid hydrogen at low temperature and metal hydride method are three recent technologies of on-board hydrogen storage (Mehrdad Ehsani, Yimin Gao, Ali Emadi, 2010).

All these technologies have several advantages and disadvantages. Nonetheless, these details are beyond the scope of this text.

**Fuel Cell Technologies**

There are six main types of fuel cell technologies that differ in the electrolyte used as well as in their operation temperature.

a) Phosphoric acid fuel cells (PAFCs):
Operation temperature: -200ºC. This sort of fuel cell is the most developed on a commercial level. It is already been used for very diverse applications such as hospitals, buildings, offices, schools, electric power plants and airport terminals. They can be used in large size vehicles such as buses or locomotives. They reach 40% of efficiency, developing around 85% when the steam produced is used for cogeneration.

b) Proton exchange membrane (PEM):

Operation temperature: 50-100ºC. They have high power density. Their output can vary according to power requirements and they are suitable for applications where high initial power demands are needed, such as vehicles.

c) Molten carbonate fuel cells (MCFCs):

Operation temperature: 600ºC. These cells have important fuel/electricity ratios and use carbon as combustible. They make use of heat and electricity.

d) Solid oxide fuel cells (SOFCs):

Operation temperature: 500-1000ºC. It is potentially useful for high power applications including stations for electric energy production on an industrial scale. A solid oxide fuel cell often uses a hard ceramic material instead of a liquid electrolyte, enabling operation temperature to reach very high values. Efficiency can increase to 60% with SOFCs.

e) Alkaline fuel cells (AFCs):

Operation temperature: 50-250ºC. These cells use potassium hydroxide as electrolyte. Their electric generation is about 70% efficient. Consequently, they are used for special missions at NASA.

f) Direct methanol fuel cells (DMFCs):

Operation temperature: 50-100ºC. Methanol is used as fuel for these fuel cells, which is a liquid that can be easily obtained and stored.

Advantages of Fuel Cells

Fuel cells are considered one of the most advantageous alternative ways of energy production. The main benefits of this kind of electric energy production are:

- Since electric energy generation is a direct process, fuel cells can reach high efficiency degrees bordering 80% when also heat is recovered.
- They do not have any part in motion so their operation is completely noiseless.
- They do not use combustion as their energy generation mechanism, what makes them practically pollution-free.
- Fuel cells can be connected in parallel in order to fulfil any energetic requirement. Individual fuel cells can be arranged in such a way as to create more powerful motors.
- Attaching fuel cells to a processor enables energy production from ordinary combustibles such as alcohols, natural gas and fossil fuels, as well as from biomass. Nevertheless, the most convenient fuel is hydrogen, since it delivers more energy per mass unit than any other fuel. Additionally, Hydrogen can be easily produced via water electrolysis. Electrolysis equipment is often powered by solar panels or wind turbines.
- Fuel cells have minimum maintenance costs.

Apart from all the above mentioned, fuel cells offer important independence in terms of installation and use, since it opens a new electricity market very different from the traditional one.
7. Energy Storage Devices

Energy storage devices are units designed to store a certain amount of energy in any form, accepting it from a primary source and being able to liberate it when required. The stored power can be delivered in the same form or it can be converted into another form of energy.

Currently, there are many types of energy storage devices that have been discussed for EV-HEV applications. The main technologies include: electrochemical rechargeable batteries, ultracapacitors or supercapacitors and ultra-high-speed flywheels.

It appears certain that one of the most important equipment in electric and hybrid electric vehicles is the portable source of electrical energy. As a fundamental requirement, the electrical source should provide on-board electrical supply in order to be transformed into mechanical energy for propelling the vehicle.

Generally, the desired characteristics of energy storage devices in EVs-HEVs comprise:

- High specific power.
- High energy density.
- Good charge acceptance capability during regenerative braking and charging operation and extended cycle life.
- High efficiency and low cost (Marangoni, 2010).

Some of these parameters are defined below:

*Specific power:* available power per unit mass from the energy source.

*Power density:* amount of power per unit volume.

*Cycle life:* number of discharge/charge cycles reached during its life time before lowering 80% its initial capacity.

In addition to these features, other characteristics related to driver satisfaction should be emphasized such as low maintenance, safety and green operation.

In an EV-HEV, it is expected that the energy source delivers peak power during discharging. The opposite behavior should be noted during regenerative braking and recharging, when the portable source operates in storing or charge acceptance mode.
7.1. Rechargeable Electrochemical Batteries

![Figure 7.1. Typical battery cell](image)

Rechargeable electrochemical batteries have been the most popular choice of electrical energy source since the first EV-HEV models were launched.

A typical battery structure consists of two electrochemical cells or electrodes connected by means of an electrolyte. As it is shown in Figure 7.1, one of the electrodes is positive and the other is negative.

The basic principle of chemical batteries operation is the conversion of electrochemical energy into electrical energy through a controlled chemical reaction. The two electrodes, positive and negative, undergo this chemical reaction, thus producing electric power. It seems clear that the reaction will be different during battery charging and discharging operations. When the battery is accepting power, it will store the electrical current in the form of electrochemical power. During discharging, the electrochemical power will be converted to electrical energy and released.

The main requirements of batteries for EV-HEV applications are summarized as follows:

- In order to reach good acceleration performance, batteries are required very high specific energy and specific power.
- To guarantee a minimum degree of electrical autonomy, batteries are designed with very low maintenance requirements and extended cycle life.
- Since the braking process is repeated and continuous while driving, batteries must have high energy acceptance from the regenerative braking system.
- From an economic point of view, the cost of energy storage devices is expected to be reasonable.

All these features increasingly benefit the environment, as EVs-HEVs have less pollutant emissions and result in lower fossil fuel consumption.

An important parameter that will be later described is the battery state of charge (SOC), which determines the electrical performance in regard to the hybrid drive train.

During the last years, research has led to the implementation of very efficient, safe and economic electrochemical batteries in EVs-HEVs. The main types of electrochemical rechargeable batteries considered for EVs-HEVs are described below.
7.1.1. Lead-acid

Lead-acid batteries are a very common choice in conventional vehicles, as a start-up battery, and it is also used in EV-HEV for producing traction power.

The generation of propulsion power takes place when power from the battery is provided to the vehicle (discharging). During cell charge operation, electrical energy from an external source is provided to the energy storage device (charging).

While basic cell operation occurs, the chemical reactions in both electrodes are:

\[
PbO_2(s) + 4H^+(aq) + SO_4^{2-}(aq) + 2e \rightleftharpoons PbSO_4 + 2H_2O(l) \]

Lead-acid batteries are usually low-priced, secure and have high specific power. Nevertheless, their short cycle life, inferior specific energy and low temperature operation limit their use in EV-HEV applications. (Husain, 2003)

7.1.2. Nickel-cadmium

Nickel-cadmium batteries are also rechargeable batteries for industrial applications. They are less used than NiMH batteries due to their “memory effect” and the polluting power of cadmium.

It is an alkaline battery where the chemical reaction of a metal oxide inside an alkaline electrolyte (potassium hydroxide) produces electrical energy. The negative electrode is cadmium metal and the positive electrode consists of a nickel oxide.

The chemical reaction that takes place during cell operation is:

\[
Cd + 2NiOOH + 2H_2O \rightleftharpoons 2Ni(OH) \_2 + Cd(OH) \_2
\]

7.1.3. Nickel-metal-hydride

NiMH batteries are also alkaline batteries, where the negative electrode is a metal hydride and the positive electrode is nickel oxide, as in Ni-Cd cells. The negative electrode stores hydrogen.

One of the most important advantages of NiMH batteries is the fact that metal hydrides are able to catch and liberate hydrogen repeatedly before lowering the battery initial performance. During battery operation, the chemical reactions in both electrodes are:

Positive electrode,

\[
NiOOH + H_2O + e^- \rightleftharpoons Ni(OH) \_2 + OH^-
\]
Negative electrode,

\[ MH_{x} + OH^- \xrightarrow{\text{Charge}} MH_{x-1} + H_2O + e^- \]

In recent years, many EV and HEV models have applied this type of energy source to their systems. Of particular interest among these are: the Chrysler electric minivan “Epic”, Toyota EV RAV-EV and Toyota HEV Prius.

In comparison with Ni-Cd batteries, NiMH batteries are more expensive, less efficient and present a lower charge acceptance rate. However, NiMH batteries are safer and have a more extended cycle life. (Husain, 2003)

7.1.4. Lithium-ion

In Li-ion batteries, the positive electrode is cobalt oxide or a nickel based oxide and the negative electrode consists of lithium intercalated carbons (Li_xC). An organic material is used as electrolyte.

During discharge operation, positive lithium ions transit from the negative electrode to the positive electrode passing through the electrolyte. The protons (Li^+) are absorbed by cobalt or nickel at the positive electrode forming LiCoO_2.

When discharging the cell, the inverse process takes place and ions travel from the positive electrode to the negative electrode, reversing the aforementioned operation.

Due to their many advantages, Li-ion batteries are becoming increasingly common in EV-HEV, as they fulfill specific performance requirements for these applications. In general, their main features are: low self-discharge rate, high specific power, high specific energy and good high-temperature performance.

Li-ion batteries also have low environmental impact and less cost since their components are all recyclable.

7.1.5. Lithium-polymer

Lithium-polymer batteries incorporate polymer electrolytes in their structure. Therefore, the electrolyte is a solid state one. The negative electrode is usually lithium metal and they use a transition metal intercalation oxide (M_xO_y) as positive electrode.

The polymer electrolyte is typically polyethylene (PE) or polypropylene (PP) in the form of a microporous film, which guarantees flexible and safe cell architecture.

The main advantage of Lithium-polymer batteries is their security in the event of accidents, due to their solid state electrolytes. However, they have very weak-low temperature operation range (80-120ºC), limited by the low polymer conductivity.

Other key features are: high specific power, good calendar and cycle life, what make them very suitable for EV and HEV applications.
Table 7.1. Battery characteristics for different types of HEV (Pistoia, Gianfranco, 2010)

<table>
<thead>
<tr>
<th>Vehicle category</th>
<th>Battery voltage (V)</th>
<th>Battery energy content (kWh)</th>
<th>Chemistry and optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEV</td>
<td>12, 24, 36, 48</td>
<td>1–2</td>
<td>Pb, Ni – Energy</td>
</tr>
<tr>
<td>Industrial</td>
<td>12, 24, 36, 48</td>
<td>5–50</td>
<td>Pb – Energy</td>
</tr>
<tr>
<td>BEV</td>
<td>200–500</td>
<td>25–75</td>
<td>Ni, Li – Energy</td>
</tr>
<tr>
<td>FCV</td>
<td>300–400</td>
<td>1–3</td>
<td>Ni – Power</td>
</tr>
<tr>
<td>HEV – Micro</td>
<td>12</td>
<td>0.50</td>
<td>Pb – Power</td>
</tr>
<tr>
<td>HEV – Mild</td>
<td>42–200</td>
<td>1</td>
<td>Ni, Li – Power</td>
</tr>
<tr>
<td>HEV – Full</td>
<td>300–500</td>
<td>2–5</td>
<td>Ni, Li – Power</td>
</tr>
<tr>
<td>PHEV</td>
<td>300–500</td>
<td>5–20</td>
<td>Li, Ni – Energy</td>
</tr>
</tbody>
</table>

7.2. Ultracapacitors or Supercapacitors

Capacitors are devices designed with the ability to store electrical energy sustaining an electrostatic field. Their architecture consists of two conducting surfaces, generally plates or films, with a dielectric or vacuum between them (Figure 7.2). When potential difference is applied in both plates, they acquire electrical positive or negative charge respectively, while the total charge is zero. Capacitors store certain amount of energy that will be later delivered completely.

Supercapacitors or ultracapacitors are electrochemical devices derived from conventional capacitors, with an enhanced capacity to sustain unusually high energy density. This improvement enables supercapacitors to develop a performance similar to batteries. In addition, ultracapacitors do not involve chemical reactions.

In conventional capacitors, the capacitance is defined in relation to energy storage between the plates due to the insulator. When referring to ultracapacitors the concept pseudo-capacitance is more appropriate. The term pseudo-capacitance relates to the manner in which supercapacitors store electrostatic charge. In this case, storage of electrical charge is done in the form of metallic ions, thanks to the electrolyte used. The electrical charge stored in the form of ions adds up to the traditional energy storage in electrostatic charges.

Therefore, the nature of the electrolyte, the electrodes and the insulator is crucial for determining the pseudo-capacitance phenomenon. The best results are obtained when the electrodes are built with a highly porous and homogenous material combined with a carbon interphase. This configuration allows very high energy density and great capacitance.

Although supercapacitors have not been developed yet as primary electrical energy sources for EV-HEV applications, they are being considered as support for batteries or any other energy storage device in the vehicle.

Their use nowadays is limited to short and transitory power demands, since pseudo-capacitance is variable with voltage. For larger power demands, supercapacitors might reach high sizes, becoming very heavy, expensive and unsuitable for that application.

Nevertheless, ultracapacitors appear to be very efficient in energy recovery operations during regenerative braking of EVs-HEVs. Researchers are trying to improve supercapacitors characteristics, especially specific energy density, in order to fill the gap between ordinary capacitors and batteries.
7.3. Ultra-high-speed Flywheels

Flywheels are devices that enable a mechanical system to store kinetic energy due to its additional rotating inertia. The energy produced can be stored through a generator connected directly to the flywheel, which is similar to a disk or a rotor (Figure 7.3). Then, the electrical generator can be used to ride a motor engaging its output adequately.

Although flywheels efficiency depends on the vehicle acceleration performance, they are powerful enough to be used as primary energy sources in HEVs and EVs, unlike ultracapacitors. Flywheels can also serve as support energy sources for electrochemical batteries.

Another application for flywheels is kinetic energy recovery during braking. For this purpose, the flywheel needs to be connected to the vehicle wheels, thus matching their speed and storing kinetic energy during deceleration, performing a very efficient operation (James Larminie, 2003).

Since ultra-high-speed flywheels are mechanical devices, their operation do not involve chemical reactions affected by temperature conditions, which is a major advantage over batteries. Consequently, they are 100% environmentally-friendly.

Flywheels are also simple and reliable from a structural point of view, knowing its state of charge by measuring the rotational speed at any moment. Another advantages derived from its mechanical nature are very long service life with low-maintenance requirements. They have very short recharging times too. Among flywheels main drawbacks are: extra weight in the vehicle, power electronics needed to match the power output from the generator, less maneuverability due to rotating forces during flywheel motion and very quick power release in case of failure which can provoke serious instability and damage in the vehicle (Husain, 2003).

These are the main reasons that have determined flywheels limited implementation in electric and hybrid electric vehicles.

Common materials used for flywheels include composite materials such as carbon fiber, which reduces overall weight and responds better in the event of a system failure.
Figure 7.3. Connection with generator in flywheel

7.4. Battery Management System

In previous chapters, batteries have been presented as the first choice when selecting electrical energy sources for electric and hybrid electric vehicles. Since voltages and currents implied in HEV operation are much higher than those in conventional vehicles, the management of these variables deserves accurate consideration and study.

In HEVs, batteries do not need recharging from an external source and they should be designed to receive frequent power inputs from the regenerative braking system. In order to guarantee an optimum vehicle performance under these operation conditions, the battery management system (BMS) establishes appropriate power limits and control tasks.

When discussing hybrid and electric vehicles, batteries, or energy storage devices in general, are a small but highly important part of a complex structure. Therefore, the battery management system plays a crucial role in these applications, ensuring safety and reliability while driving. This does not mean that the BMS controls the whole power electronics of a hybrid electric vehicle, since a more powerful and extended controller is required for that purpose. Nonetheless, the BMS is in charge of granting real time and precise information about battery current state and certain battery parameters, such as the state of charge, which will be later defined.

Typically, the functions of the battery management system in HEVs are classified under six main tasks (Marangoni, 2010):

1. **Measuring.** Commonly, the BMS measures specific observable data from various systems such as battery cells, packs or management systems. Usually, the quantified parameters refer are voltages, currents, cell impedance and temperatures. The measurements can be done at battery cell or pack level, and they are not compared with a limit value since this will imply more complex calculation.

2. **Calculating.** The measurements from the first step are treated and introduced in calculation in order to obtain useful values that cannot be directly read from the battery systems. Typical estimated values are: state of charge (SOC), state of power (SOP) and state of health (SOH). SOC and SOP refer to the available energy and power in the battery, while SOH reflects the battery lowered performance over time.

3. **Monitoring.** This function, as measuring, refers to quantifying certain observable characteristics of the battery. The measurements are done by checking the levels of
the quantified parameters. Additionally, monitoring compares the checked levels against threshold values generating, if necessary, waning messages in order to perform a control action.

4. **Communicating**. Communicating is the ability to transfer the information obtained from measuring, monitoring and calculating to calling subsystems of the vehicle. The communication can be done by simple LED devices or more complex data buses.

5. **Control**. Control defines the direct management tasks of the BMS over the battery components. The BMS can determine direct control of subsystems and processes such as thermal management system or charge/discharge operations. Most often, control is separated from the battery pack so that its elements do not influence directly the decision routine. BMS control must be synchronized with the vehicle general control algorithm in order to avoid inconsistencies that could jeopardize the safety of the vehicle and its occupants.

6. **Balancing**. This task is targeted to find the balance between the cells composing the battery pack. Due to slight differences in self-discharge rates or cell impedances, the battery cells can perform differently over time, thus lowering the efficiency. An appropriate equalization technique should be applied so as to extend the cycle life of the battery and maintain its original capacity. Several technologies have been developed, including “bypass” and “active” equalization.

It should also be emphasized that the structure, components and tasks of the battery management system depend on the vehicle application requirements as well as on the battery operation conditions and electrochemistry. For this reason, the BMS architecture is not unique and it is more complex and arranged differently in each HEV, thus not requiring the implementation of all the above-mentioned BMS functions.

Table 7.2 shows a comparison between different BMS architectures and functions depending on the vehicle category:

<table>
<thead>
<tr>
<th>Vehicle category</th>
<th>Typical functions</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEV</td>
<td>Monitor, control</td>
<td>Centralized</td>
</tr>
<tr>
<td>Industrial</td>
<td>Monitor, measure, communicate</td>
<td>Centralized</td>
</tr>
<tr>
<td>BEV</td>
<td>Monitor, measure, calculate, communicate, balance</td>
<td>Distributed</td>
</tr>
<tr>
<td>FCV</td>
<td>Monitor, measure, calculate, communicate, balance</td>
<td>Centralized or Distributed</td>
</tr>
<tr>
<td>HEV – Micro</td>
<td>Monitor, measure, calculate, communicate</td>
<td>Centralized</td>
</tr>
<tr>
<td>HEV – Mild</td>
<td>Monitor, measure, calculate, communicate, balance</td>
<td>Centralized or Distributed</td>
</tr>
<tr>
<td>HEV – Full</td>
<td>Monitor, measure, calculate, communicate, balance</td>
<td>Centralized or Distributed</td>
</tr>
<tr>
<td>PHEV</td>
<td>Monitor, measure, calculate, communicate, balance</td>
<td>Distributed</td>
</tr>
</tbody>
</table>

Centralized architecture refers to a single management system that includes every BMS function. Most often, the BMS is located together with the battery pack. Therefore, to be implemented and physically feasible, this architecture requires that the battery pack size is medium or small.
In a distributed architecture, the BMS is formed by separated modules which are then assembled to each other. BMS functions are arranged among the different subsystems and occasionally repeated in some of the modules. Distributed architecture allows choosing the number of BMS components for different battery pack sizes and therefore, the design of the battery system is somehow versatile.

### 7.4.1. State of Charge Estimation

State of charge is defined as the parameter indicating the actual stored energy at a given time in relation to the original storage capacity of the battery pack.

In hybrid and electric vehicles the SOC is determined by the input power coming from the internal combustion engine (or fuel cell) and from the regenerative braking system. This implies that the battery pack suffers continuous charge and discharge cycles from both systems. Therefore, the SOC is not meant to be neither 100% (battery completely charged) nor 0% (battery completely discharged). Consequently, the SOC will be usually fluctuating between certain percentage limits defined beforehand by the battery pack designer. For EVs-HEVs, these limits usually determine a slender SOC range. This range could be further restricted in order to extend battery calendar and cycle life.

It is clear that batteries under deep discharge and charge series have shorter calendar and cycle life, than those operating with a SOC value far from those limits. This results from preventing changes in the battery chemistry due to overcharge and overdischarge states. In addition, the state of charge must be enough for delivering peak power during discharge operation, as well as for incoming large power rates from the regenerative braking system.

For the reasons set out above, the definition of the state of charge is crucial for hybrid vehicle performance, staking safety and efficiency. Therefore, it is essential certain degree of accuracy and precision on SOC readings. Since SOC is a parameter that cannot be directly quantified, its calculation will require a suitable estimation method. The complexity of this process will differ depending on the battery pack technology and architecture.

The simplest SOC estimation process is based on the measurement of the terminal current and it is known as "Coulomb counting". The accuracy of SOC readings is determined by current measurement precision. More sophisticated techniques include analysis of battery self-discharge and loss of original capacity over battery aging. Finally Kalman filtering procedure compares available values of the battery pack with outputs of a battery model, being the most advanced method so far (Lino Guzzella, Antonio Sciarretta, 2007).

Since SOC is an essential point in electric and hybrid and electric vehicles safety, researchers are designing and testing more advanced BMS technologies that integrate improved SOC estimation features. Realizing the great benefits of undertaking this challenge, designers try to fulfill safety requirements combining realistic battery models with technologies adapted to battery aging and operation.
8. Dynamic Modeling of Electric and Hybrid-Electric Propulsion Systems

When modeling electric and hybrid electric vehicles, there are two possible approaches: quasistatic and dynamic. Both approaches split the complete vehicle model into submodules corresponding to the different components of the drivetrain. Each module has well defined inputs and outputs which determine the power flow across the whole drivetrain of the vehicle. In this way, it is easy to study each component separately, define its particular dynamic model and then assemble all the submodules according to the full vehicle design. This provides general dynamic models for every component of the powertrain which can then be used for modular representations of electric and hybrid propulsion systems with different configurations.

![Figure 8.1. Basic series hybrid vehicle configuration. B: battery, E: engine, G: generator, M: motor, P: power converter, T: transmission (including clutch and gears), V: axles and vehicle. Bold lines: mechanical link, solid lines: electrical link (Lino Guzzella, Antonio Sciarretta, 2007)](image)

(a) Quasistatic approach

In the quasistatic approach, the main inputs are desired vehicle speed and acceleration. With the acceleration, the force acting on the wheels is calculated and therefore the torque demanded to the motor and/or the engine can be computed. After this step, it is possible to estimate the power requirement from the battery or peaking power source.

The main input variable for the dynamic approach is the gas pedal position, which determines vehicle speed. This approach yields as its main output the torque or force acting on the wheels. As an example, Figure 8.1 and Figure 8.2 represent the basic configuration and the power flow modeling of a series hybrid vehicle for both quasistatic and dynamic approaches.

In the present work, the dynamic approach has been implemented. This choice was made given the fact that MBS3D software is designed to receive as an input the torque acting on the car wheels and yields as outputs vehicle position, speed and acceleration at each moment. MBS3D block is represented in Figure 8.3. Therefore, dynamic approach was very appropriate for the aim of this project and enables the coupling of MBS3D software and multidisciplinary dynamic models such as electric car powertrains.

8.1. Dynamic Modeling of Electric Motors

The input variables for an electric motor/generator dynamic model are the motor shaft rotational speed and the power link voltage. The model outputs are the current transferred to the power converter (Figure 8.5). When the machine operates as a motor, the current $I_1(t)$ should be positive and when it operates as a generator $I_1(t)$ is negative and delivered by the machine.
Co-simulation Methodologies for Hybrid and Electric Vehicle Dynamics

8.2. Dynamic Modeling of Batteries

The battery model input is the terminal current $I_2(t)$. The model outputs are the battery charge $Q(t)$ and the terminal voltage $U_2(t)$ (Figure 8.6). When $I_2(t)$ is negative, the battery is in charging mode and a positive $I_2(t)$ charges it.

8.3. Dynamic Modeling of Electric Power Links

The main purpose of the electric power link or electrical coupling device, placed between the battery and the electric motor, is to manage the voltage levels of the different electric sources. In the case of an electric or hybrid vehicle, the electric power link consists of a bi-directional DC/DC converter, which boosts or bucks the voltage depending on the level needed. The dynamic model of power links (Figure 8.7) has as inputs the battery voltage $U_1(t)$ and the current at the other ports (in this case, only the electric motor current $I_2(t)$). The model outputs are the current at the battery port $I_1(t)$ and the voltage at the electric motor port $U_2(t)$.

8.4. Dynamic Modeling of Internal Combustion Engines

Since the vehicle model developed in this work is an electric one, the dynamic model of ICEs is not included among the differential equations. However, it is interesting to show the dynamic inputs and outputs of ICE modular representation sketched in Figure 8.8. For the ICE module, the inputs are the engine shaft rotational speed $\omega_e$ and the enthalpy flow $P_c$, related to the fuel mass flow by means of the expression

$$\dot{m}_f = P_c / H_i \quad (8.1)$$

where $H_i$ is the lower heating value.

The model output is the engine torque $T_e$. 
8.5. Mathematical Model of Hybrid and Electric Vehicles

In order to predict the dynamic behavior of a hybrid or electric drive train, an appropriate mathematical model is needed. The model must be, moreover, valid for post-simulation. The benefits of mathematical modeling lie on the great help that this represents for understanding complex structures such as vehicles. In addition, during vehicle development, mathematical analysis leads to considerably reduced time and costs. Consequently, from the perspective of industrial production, the translation to mathematical language is a key stage in the process.

Once the model has been established, simulation programs and codes are developed with the purpose of creating an accurate forecast for the engine behavior, the battery usage and the fuel consumption. Taking all this as a starting point, it is possible to design an efficient control strategy. Such a strategy is especially important, since it directly influences fuel-saving and determines the correct electrical power management in the vehicle.

In general, the sprung and unsprung mass dynamics, as well as the pneumatic performance are studied separately in order to construct the mathematical model of a vehicle. Besides, to define a complete hybrid vehicle model, it will be necessary to characterize the mathematical description of the electric drive train components.

The present work develops the mathematical model of a full electric vehicle and its validation through co-simulation methods. To that end, the equations corresponding to the operation of the electric powertrain have been introduced in MBS3D simulation tool and integrated with the dynamic equations of the vehicle model considered as a multibody system. MBS3D software will be described in a later chapter. The formulation of the electric drivetrain equations includes the currents and voltages needed to meet the power requirement. These currents and voltages are in turn the traction or braking torque acting on the wheels when the vehicle moves.

Currently there are two possible ways to simulate hybrid and electric vehicles:

- **Backward-Facing Approach:** a predefined driving cycle is the input to the vehicle model, which determines speed and power requirements. With this as the starting point and once the vehicle model is complete, the necessary force at any given time can be obtained and, therefore, the power and torque demand in order to fulfill the driving cycle. In this way, it is possible to calculate fuel and/or energy consumption. The calculation path of this approach is opposite to the energy flow.

- **Forward-Facing Approach:** the driver behavior is introduced as the main input, including parameters such as the gas pedal position. In this alternative, the gas pedal position determines the torque delivered by the motor and/or the engine, as a function of the system configuration and control strategy. The torque is provided to the wheels through the transmission system. Consequently, power losses and transmission efficiency must be taken into account.
9. Electric Vehicle Model Description

Figure 9.1 shows the powertrain configuration of the electric vehicle. As it can be seen, a battery pack provides current to the electric motor through a DC/DC power converter which levels the voltage between the two electrical sources. The electric motor shaft is connected to a reduction gear which lowers the speed of the motor so that the wheels rotate at a moderate speed. Finally a differential allows the wheels to rotate at different speeds in turns.

9.1. Vehicle Parameters

The vehicle data was taken from an existing MBS3D model called car2 which was then renamed as car2HE to indicate the latter is the extended version for hybrid and electric vehicles.

Relevant vehicle data and parameters of its geometry are shown in Table 9.1:

<table>
<thead>
<tr>
<th>Vehicle parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle mass</td>
<td>$M = 1148 \text{ kg}$</td>
</tr>
<tr>
<td>Frontal area</td>
<td>$A_f = 2.488 \text{ m}^2$</td>
</tr>
<tr>
<td>Centre of gravity height</td>
<td>$h_{\text{cog}} = 0.5373 \text{ m}$</td>
</tr>
<tr>
<td>Wheel radius</td>
<td>$r_w = 0.467264 \text{ m}$</td>
</tr>
<tr>
<td>Wheel stiffness</td>
<td>$k_w = 2 \times 10^5 \text{ N/m}$</td>
</tr>
<tr>
<td>Gear transmission ratio</td>
<td>$i_g = 8.25$</td>
</tr>
<tr>
<td>Transmission efficiency</td>
<td>$t_{\text{eff}} = 98%$</td>
</tr>
</tbody>
</table>

Nowadays, most electric vehicles do not have a gear-box, since electric motors produce their maximum torque at very low speeds and their maximum power at high speeds. In the case of a DC motor, full torque is generated near stall (zero rpm), what provides very good acceleration performance. For these reasons, multi-speed transmissions are not necessary.
in electric cars but they would rather add extra weight and complexity. Single-speed transmissions or reduction gears are used instead.

The gear ratio was designed so that when the motor reaches its maximum speed, the vehicle also moves at its maximum speed according to the expression:

\[ i_g = \frac{\pi n_{m,\text{max}} r_w}{30 V_{\text{max}}} \]  \hspace{1cm} (9.1)

where \( n_{m,\text{max}} \) is the maximum motor rotational speed in rpm, \( r_w \) is the tire radius and \( V_{\text{max}} \) is the vehicle maximum speed in m/s. In order to reach reasonable motor power rating, these values were optimized resulting in the following \( V_{\text{max}} = 160 \text{ km/h} \), \( n_{m,\text{max}} = 7500 \text{ rpm} \) and \( i_g = 8.25 \).

9.2. Electric Motor

The selected electric propulsion device is a separately excited DC motor. Although this type of DC motor is not commonly used for electric vehicle propulsion, it is very suitable for dynamic simulation within the MBS3D software due to the form of its dynamic equations.

In separately excited DC motors, the rotor (armature) winding is independent of the field winding, which produces the excitation, and they have different supplies. The equivalent circuit of the dynamic model of a separately excited DC motor is represented in Figure 9.2.

![Dynamic circuit of a separately excited DC motor](image)

**Figure 9.2. Dynamic circuit of a separately excited DC motor (Lino Guzzella, Antonio Sciarretta, 2007)**

The voltage equations are obtained applying Kirchhoff’s law. For the armature circuit:

\[ U_a(t) = I_a \frac{dI_a(t)}{dt} + R_a I_a(t) + U_i(t) \]  \hspace{1cm} (9.2)

where \( I_a(t) \) is the armature current, \( U_i(t) \) is the induced voltage or back emf, \( R_a \) is the armature resistance, \( L_a \) is the armature inductance, and \( U_a(t) \) is the armature voltage. The armature inductance and resistance are directly measured and provided by the manufacturer. The DC voltage applied to the armature windings is usually controlled by means of a chopper converter which regulates the supply voltage \( U_f(t) \) depending on the power requirements.

For the field circuit:

\[ U_f(t) = L_f \frac{dI_f(t)}{dt} + R_f I_f(t) \]  \hspace{1cm} (9.3)
with analogous meanings of the variables. The field voltage can also be regulated with a chopper converter as a function of the DC input voltage.

The relationship between motor torque and speed is obtained applying Newton’s second law to the motor shaft:

$$\frac{dw_2(t)}{dt} = \frac{T_a(t) - T_2(t)}{\Theta_m}$$  \hspace{1cm} (9.4)

where $w_2(t)$ is the motor speed, $T_a$ is the torque delivered by the motor, $T_2(t)$ is the load torque on the motor shaft and $\Theta_m$ is the motor rotational inertia.

Additionally, in separately excited DC motors, the motor torque is proportional to the field and rotor currents and the back emf or induced voltage is proportional to the field current:

$$T_a(t) = L_m I_f(t) I_a(t)$$  \hspace{1cm} (9.5)

$$U_f(t) = I_m I_f(t) w_2(t)$$  \hspace{1cm} (9.6)

The field current will be kept constant for speeds below the base speed ($w_b = 1785$ rpm in this case). For speeds above the base speed, field weakening control (FWC) will be applied. In the present work, FWC is done by reducing the current flowing through the field circuit with a chopper converter which will regulate the field voltage as a function of the DC supply:

$$U_f(t) = \beta(t) U_a(t)$$  \hspace{1cm} (9.7)

where $\beta(t)$ can be varied according to the field current output requirements.

The electric motor parameters are summed up in Table 9.2.

<table>
<thead>
<tr>
<th>$R_a$</th>
<th>0.516 Ω</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_a$</td>
<td>0.02 H</td>
</tr>
<tr>
<td>$R_f$</td>
<td>150 Ω</td>
</tr>
<tr>
<td>$L_f$</td>
<td>100 H</td>
</tr>
<tr>
<td>$L_m$</td>
<td>1.5 H</td>
</tr>
<tr>
<td>$V_a$</td>
<td>320 V</td>
</tr>
<tr>
<td>$V_f$</td>
<td>220 V</td>
</tr>
<tr>
<td>$\omega_n$</td>
<td>2800 rpm</td>
</tr>
<tr>
<td>$\omega_{max}$</td>
<td>7500 rpm</td>
</tr>
<tr>
<td>$T_n$</td>
<td>240 Nm</td>
</tr>
<tr>
<td>$\Theta_m$</td>
<td>$10^{-4}$ kg*m$^2$</td>
</tr>
</tbody>
</table>
9.3. Battery Pack

The battery pack of the electric vehicle model consists of 176 power-oriented Lithium-ion cells of the LiNi$_x$Mn$_y$Co$_z$O$_2$ (NMC) chemistry. The cells are grouped in 44 modules with 4 cells in each module in a 2x2 configuration. This results in a total nominal voltage of 320 V, and a total battery capacity of $Q_0=62$ Ah, taking into account the characteristics of the individual cells listed in Table 9.3.

There are two different approaches for the dynamic modeling of EV batteries: dynamic equivalent circuit and lumped-parameter electrochemical models. Here, a dynamic equivalent circuit will be used since variations in the concentration of the different electrochemical species, which are given by electrochemical models, are not particularly relevant for this work.

The equivalent circuit for dynamic battery modeling is the Randles model represented in Figure 9.3.

![Randles circuit for dynamic modeling of batteries](Lino Guzzella, Antonio Sciarretta, 2007)

The Kirchhoff’s equations for the circuit above are:

$$U_2(t) = U_{oc} - R_o I_2(t) - U_0(t)$$  \hspace{1cm} (9.8)

$$R_o C_{dl} \frac{dU_0(t)}{dt} = U_{oc} - U_2(t) - U_0(t) \left(1 + \frac{R_0}{R_d + R_{ct}} \right)$$  \hspace{1cm} (9.9)

$R_o$ represents the ohmic resistance; $R_d$ and $R_{ct}$ are the diffusion and charge-transfer resistances respectively; $C_{dl}$ is a double-layer capacitance which describes the capacitive effects of the charge accumulation/separation at the interface between the electrolyte and the electrodes.

$U_{oc}$ represents the open circuit voltage and $U_0$ is the non-ohmic overpotential caused by the current flow across the parallel branches.

In this model, the circuit parameters of the battery are SOC dependent assuming that the operating temperature is $T=40$ºC and that it is kept constant during the whole driving process. With this assumption, the estimated values for the battery cells are:
Since $U_{oc}$ has a stronger SOC dependency, it will be implemented as a function of the latter according to the following expression:

$$U_0 = \begin{cases} 
0.4 \times SOC + 3.5, & \text{if } SOC < 0.5 \\
SOC + 3.2, & \text{if } SOC \geq 0.5 
\end{cases}$$  \hspace{1cm} (9.10)$$

Figure 9.4 represents this behavior.

### Table 9.3. Lithium-ion cell characteristics (T=40°C)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_0$</td>
<td>0.009 Ω</td>
</tr>
<tr>
<td>$R_1$</td>
<td>0.0015 Ω</td>
</tr>
<tr>
<td>$C_{dl}$</td>
<td>$3.6 \times 10^4$ F</td>
</tr>
<tr>
<td>$Q_{cell}$</td>
<td>31 Ah</td>
</tr>
<tr>
<td>$V_{n,cell}$</td>
<td>3.7 V</td>
</tr>
</tbody>
</table>

9.3.1. State of Charge

Dynamic battery models are meant to be validated through simulations of the order of a few seconds and, therefore, variables such as the state of charge will not change significantly during short integration intervals. This is due to the long charge and discharge times of this type of energy storage device. For this reason, in the present work the state of charge is an input variable for the model and it is considered constant over time, although graphics will show slight variations in its value.

The state of charge estimation is done by “Coulomb counting” method and the equations applied are shown below:
$SOC(t) = \frac{Q(t)}{Q_0}$  \hspace{1cm} (9.11)

where $Q(t)$ is the battery charge and $Q_0$ is the battery nominal capacity. According to Coulomb counting method, the battery charge is calculated by integrating the current drawn from the battery, what means:

$$Q(t) = -I_{\text{batt}}(t)$$  \hspace{1cm} (9.12)

in case of discharge, and:

$$Q(t) = -\eta_c I_{\text{batt}}(t)$$  \hspace{1cm} (9.13)

where $\eta_c = 98.5\%$ is the charging efficiency.

### 9.4. Electrical Coupling Device

A very simple model of electrical coupling device has been implemented. As it has already been mentioned, it consists of a DC/DC chopper converter that regulates the armature voltage by “chopping” the DC supply voltage $U_i(t)$. Chopper converters are usually constituted by a free-wheeling diode and a fast semiconductor switch (Figure 9.5). The switch is in charge of cutting the current into segments according to its duty cycle (ratio of the time periods during which it is off or on).

![Figure 9.5. Basic chopper converter. S: switch. D: free-wheeling diode (Lino Guzzella, Antonio Sciarretta, 2007)](image)

All the above determines the relationship between the armature voltage and the DC link voltage:

$$U_a(t) = \alpha(t)U_i(t)$$  \hspace{1cm} (9.14)

where $\alpha(t)$ is the duty cycle of the chopper converter which will be changed linearly with the accelerator pedal travel distance.
Finally, a power balance across the motor and the DC/DC chopper will be needed for building the dynamic differential equations:

\[ P_l(t) = U_a(t)I_a(t) + P_{cc}(t) = U_1(t)I_1(t) \]  

(9.15)

where \( P_{cc}(t) \) are the power losses in the chopper converter.
10. Integrated Development Environment and Simulation Tool

The integrated development environment of this project includes two main programs: MATLAB and Microsoft Visual Studio. A brief description of each platform is given below.

10.1. MATLAB

The name MATLAB stands for Matrix Laboratory. It is a numerical computing environment which offers an integrated development environment with its own programming language (M code). The present work has been mainly developed in MATLAB environment.

Some of its basic features are: vector and matrix handling, algorithm implementation, plotting of data and functions, creation of graphical user interfaces (GUI) and communication with other computing codes and hardware devices.

MATLAB is a great program for scientific and technical computation. It allows maximum programmer productivity and minimum development time. For certain operations, such as executing functions written in its native code and with an appropriate size for applying its vectorization ability, it becomes very fast. However, as it uses an interpreted code instead of a compiled code, in other applications it is slower than the equivalent code written in C/C++ or Fortran. In any case, MATLAB is always an efficient high-level environment and easy-to-use technical computing language and it significantly enhances productivity in comparison with other development environments. Apart from the basic code, MATLAB also has several specialized libraries known as toolboxes.

The working environment in MATLAB is very graphical and user-friendly, similar to other professional Windows applications. The most important components of the working environment are (Javier García de Jalón, José Ignacio Rodríguez, Jesús Vidal, 2005):

- **MATLAB Desktop**: it is the main window where the rest of the components are located.
- **Individual components** oriented to particular tasks:
  a. Command Window: it is the most important window of the program. Here, MATLAB commands are executed interactively and the corresponding results are displayed.
  b. Command History: it enables access to the commands that have been previously executed in the command window.
  c. Workspace: it is the set of user defined variables that are available in the program memory at any given time.
  d. Launch Pad.
  e. Current Directory: when MATLAB is commanded to run a certain file, the first place where the program searches for the script is the current directory. Therefore, for an M-file to be executed, it is necessary to locate it in the current directory or add it to one of the directories included in MATLAB path.
  f. Help: this option offers information on everything related to MATLAB software.
  g. Editor & Debugger: MATLAB scripts are created in the editor window. It shows the elements of the commands with different colors. The debugger enables to run a simulation step by step, use breakpoints to stop the simulation at a certain line, step in and out the functions, etc.
h. Array Editor: it displays the values of all the vectors and matrices defined in the program in a worksheet style. It also allows the modification of these values by clicking on the corresponding cell.

i. Profiler: this component enables the study of the CPU time distribution while running a MATLAB program.

Some of these components are shown in Figure 10.1.

![Figure 10.1. MATLAB working environment](image)

### 10.2. Microsoft Visual Studio

All the C/C++ source files included in this project have been created in Microsoft Visual Studio environment. The main applications of Microsoft Visual Studio include the creation of computer programs in different codes such as C++, C#, Visual Basic .NET, etc., the design of computer applications for Windows, Android and iOS, web applications and cloud services.

Visual Studio also has an integrated debugger which supports nearly any programming language. Among the main features of this IDE are (Visual Studio Website, 2015):

- **Code editor**: similar to MATLAB editor, it supports syntax highlighting and code completion, using different colors for functions, variables, etc.

- **Debugger**: it can be used for debugging applications written in different languages, as long as they are supported by Visual Studio. It can also debug and attach to processes that are being run. This characteristic has been essential for carrying out the present work, since debugging process is always a key point when understanding computing code and finding programming errors.

- **Designer**: it includes a set of visual designers in order to create applications easily.

- **Other tools**: such as solution explorer, open tab browser, object browser, team explorer, data explorer, etc.

- **Extensibility**: it allows the user to develop extensions for extending Visual Studio functionalities.
Visual Studio working environment is shown in Figure 10.2.

![Visual Studio working environment](image)

**Figure 10.2. Microsoft Visual Studio working environment**

### 10.3. Simulation tool

#### 10.3.1. Introduction to Multibody Systems Dynamic Simulation

Multibody systems are sets of rigid or deformable solids whose relative motion is restricted by kinematic joints or pairs among them. Kinematic and dynamic simulation of multibody systems is part of what is known as computational mechanics. The motion differential equations that govern multibody system dynamic behavior were developed mainly by Newton, Lagrange and Euler. However, it was not possible to integrate these equations effectively until 1960, when digital computers provided the necessary tools for solving precisely numerical applications such as robotics and space problems.

Practical applications of multibody system dynamics are present in many areas of engineering: vehicles, biomechanics, machinery, robotics, etc. These applications often require different levels of accuracy and elapsed time during integration. Therefore, many integrators have been tested and implemented, thus developing multiple integration methods with certain features related to stability, convergence and precision.

Multibody system problems typically imply large displacements, both relative and absolute, being characterized by a strong non-linear behavior. Then, constraint equations are required for their resolution. Depending on the algebraic or differential nature of these equations, the set of equations that determines the movement will be either ODE or DAE and different integration methods are needed. Nevertheless, since DAE direct integration is much more complex than ODE integration, algebraic equations are often used to transform DAE in ODE, reducing the index of the equations system. In the particular case of multibody systems, for achieving ODE systems it is enough to apply the second derivative of the constraint equations (accelerations) together with the motion differential equations.

#### 10.3.2. MBS3D 2.0

MBS3D 2.0 is an open source general-purpose program for the dynamic simulation of multibody systems. It is entirely programmed in MATLAB and it uses a very efficient and tested mathematical semi-recursive formulation. MBS3D 2.0 Pro is an enhanced or
“professional” version where some of the most critical functions have been written in C/C++ in order to improve integration speed. The professional version has been proven to be two orders of magnitude faster than MBS3D 2.0. These C/C++ functions have been built later as MEX functions so that they could be implemented directly in MATLAB. MEX files enable to call C, C++ or Fortran programs from the MATLAB command and/or from MATLAB functions line as if they were built-in functions.

The simulation carried out in this project is MBS3D based and its main goal is to include the simulation of hybrid and electric vehicle dynamics in the scope of this software. For this purpose some of the main functions have been modified and extended. Also a completely new electrical model has been programmed and added to the original MBS3D scripts through co-simulation techniques.

**10.3.3. MEX File Functions**

The term MEX stands for “MATLAB EXecutable”. MEX files provide an interface between MATLAB and functions written in other programming languages such as C++, C and Fortran. In the framework of this project, MEX functions are used to run C/C++ source files in the MATLAB environment in order to increase simulation speed and efficiency.

Once the C/C++ source file is created, it needs to be compiled and linked to MATLAB by creating an associated MEX file function. To build a C/C++ MEX file, the `mex` function in MATLAB should be used as follows:

```
mex filename.cpp
```

where `filename` is the name of the C/C++ source file. To create MEX files, a MATLAB supported compiler must be installed first on the computer. In this work, Microsoft Visual C/C++ 2010 compiler has been used. To call a MEX file from MATLAB, the name of the file should be typed on the MATLAB workspace, without the file extension. The MEX file, at the same time, calls and executes the C/C++ subroutine.

The process described above was been applied to the MATLAB files involved in this project. First of all, the original files written in MATLAB code were translated into C/C++ language. Then, these C/C++ source files were compiled and their corresponding MEX files were built. Once these steps were successfully completed, it was possible to run the .cpp files in the MATLAB environment obtaining the same results as with the MATLAB files but in a significantly shorter simulation time.
11. Co-simulation Process

Generally, engineering prototypes such as vehicles require a multidisciplinary approach, which involves different modeling, simulation and control methodologies for each physical domain. The main model is often divided into more simple models which are solved separately with specialized software or simulation tools. Then, these sub-models are coupled within a co-simulation environment. The main advantage of splitting the model is that currently there are well tested and available software tools for modeling and simulating systems from one specific physical domain.

11.1. Coupling of Models

The models can be coupled according to different levels. Coupling levels can be established considering the general classification for CAE (Computer-Aided Engineering) software tools represented in Figure 11.1.

![Figure 11.1. Coupling of models (levels) (Martin Arnold, Werner Schiehlen, 2008)](image)

According to the figure above, the first distinction within coupling CAE software is between design tools and design models. Design tool refers to the computation of general design parameters and it does not consider the model of the whole system. Design models consider a whole system and its entire functionality. The next classification is between operational models, such as running computer codes or simulation models, and descriptive models, such as programming code or symbolic equations of motion (Martin Arnold, Werner Schiehlen, 2008).

On the level of operational design models (e.g. running computer codes), there are two main approaches depending on the number of integrators which are communicating and exchanging values during the simulation. From this point of view, two co-simulation techniques can be distinguished: one integrator and two or more integrators. When there is only one integrator the simulation method is known as function-call or 'tight coupling' (Figure 11.2). Two or more integrators generate the 'weak coupling' or co-simulation method (Figure
11.3. In tight-coupling scheme all subsystems are integrated jointly and values are exchanged by function-call. In weak-coupling method each subsystem is integrated separately at different integration rates and information is exchanged from one integrated subsystem to another (Martin Arnold, Werner Schiehlen, 2008).

![Tight Coupling](image1)

**Figure 11.2. Tight Coupling**

![Weak Coupling](image2)

**Figure 11.3. Weak Coupling**

Weak-coupling of models may have better co-simulation efficiency because of the different integration rates for different subsystems which results in lower integration time. However, in this work tight coupling method has been chosen due to its simplicity and adaptability to MBS3D requirements. In this way, all the equations (mechanical and electrical) are integrated simultaneously by the same integrator and under a single integration rate.

11.2. Introduction to MATLAB ODE Suite

MATLAB has a set of integrators for solving ordinary differential equations called “MATLAB ODE Suite”. This suite was created for the solution of initial value problems

\[ y' = F(t, y) \]  \hspace*{1cm} (11.1)

on a time interval \([t_0, t_f]\) and with initial values \(y(t_0) = y_0\). These solvers allow the general form

\[ M(t)y' = f(t, y) \]  \hspace*{1cm} (11.2)

where \(M(t)\) is a non-singular mass matrix and usually sparse.

These integrators have been developed by worldwide experts and have a common or unified user interface. They all have a variable step size in order to adapt to problem difficulties and some of them are able to solve stiff differential equations. A stiff system consists of a set of ordinary differential equations where the main solution varies slowly over time but there are also proximate solutions that are changing rapidly, so the integrator needs a small step size for being efficient and obtaining reliable results.

Some of these integrators are based in Runge-Kutta and Adams-Bashforth-Moulton numerical methods. Runge-Kutta methods are a family of implicit and explicit methods for ODE numerical analysis and solving. ABM method is a linear multistep method which in each
integration step, bases a new solution point on a linear combination of all the previous solution points and derivatives thus improving efficiency.

A brief description of each integrator is provided below:

**ode45:** 4/5 order Runge-Kutta integrator. Non-stiff systems.

**ode23:** 2/3 order Runge-Kutta integrator. Non-stiff systems.

**ode113:** Integrator based on Adams-Bashforth-Moulton method. Variable order. Non-stiff systems.

**ode15s:** Variable order multistage methods. DAEs and stiff problems.

**ode23s:** Low order integrator. Provides an alternative to ode15s for the solution of stiff problems.

**ode23t:** Low order integrator, based on the trapezoidal rule; provides a solution without numerical damping. Moderately stiff problems.

The code interface of ODE Suite integrators was written so that they all could be use in exactly the same way. The general code for solving an initial value problem is shown below, where van der Pol equation (vdp) has been used as an example.

```
[T, Y] = odeX(@deriv, tspan, y0, options)
```

```matlab
function dy = vdp(t,y)
dy = zeros(2,1); %preallocate column vector dy
dy(1) = y(2);
dy(2) = (1-y(1)^2)*y(2)-y(1);
```

In the code above, `@deriv` is a reference to function where the differential equations are defined, `tspan = [t0,tfinal]` is the interval of integration and `y0` is the vector that contains the initial conditions. The length of vector `y0` determines the number of equations. It is also usual to include an `options` structure. It is built by calling `odeset` function and is often used to establish tolerances associated with the error control (Lawrence F. Shampine, Mark W. Reichelt, 1997).

For example, in order to solve a stiff problem with a constant Jacobian, an absolute error tolerance of $10^{-08}$, and a maximum step size of 3500, the `options` code would be written as follows:

```
options = odeset('Jacobian','AbsTol',1e-08,'hmax',3500);
```

In this way, all codes in the suite are simple and have the same appearance to the user. In addition, throughout this work `rk4b` integrator has been used. It is not a MATLAB integrator and it is a straightforward implementation of a fourth order Runge-Kutta method. Unlike MATLAB integrators, `rk4b` has a constant integration step size.

Several tests and simulations have been carried out in order to study and compare the behavior of the different ODE Suite integrators when solving the electric vehicle model equations. These simulations include:

- Comparison of the integration time when running MATLAB and C/C++ files.
- Study of stability and comparison of the integration time between different integrators when varying integration output step, integration interval, relative and absolute error tolerances and initial conditions of the problem.
Co-simulation Process

- Study of vehicle performance characteristics under different values of the battery state of charge, what will determine maximum available power from the peaking power source.
- Short driving cycle implementation by varying the brake and/or gas pedal inputs.
- Comparison of integration times between the original MBS3D car model and the electric vehicle model, including the new electrical differential equations.

The results of all the above will be discussed in the next chapter.

11.3. Coupling of MBS3D Software and Electrical Subsystem

As it has already been mentioned, a tight coupling method has been used to co-simulate both mechanical and electrical subsystems. In order to couple all the equations within MBS3D simulation environment, four main stages were followed: study and reformulation of the electrical powertrain equations, a first integration of the electrical equations separately from multibody dynamic equations, electrical parameter setting and gear ratio adjustment, and finally coupling of both subsystems.

11.3.1. Study and Reformulation of Electrical Powertrain Equations

After studying the dynamic modeling of electrical powertrains, a set of differential-algebraic equations was available for the integration. However, in order to include these equations in MBS3D software, they needed to be reformulated and transformed into a set of ordinary differential equations. For this purpose, the nomenclature of all the dynamic electric equations was unified and algebraic equations were inserted into their corresponding differential equations. All the model equations were analyzed and transformed as follows:

- Electric motor:

  \[ U_a(t) = L_a \frac{dI_a(t)}{dt} + R_a I_a(t) + U_i(t) \]  
  \[ U_f(t) = L_f \frac{dI_f(t)}{dt} + R_f I_f(t) \]  
  \[ \frac{dw(t)}{dt} = \frac{T_a(t) - T(t)}{\theta_m} \]  
  \[ U_i(t) = L_a I_a(t) w_2(t) \]  
  \[ T_a(t) = L_m I_f(t) I_a(t) \]  
  \[ P(t) = U_a(t) I_a(t) + P_{i,d}(t) = U(t) I(t) \]

- Battery:

  \[ R_0 C \frac{dU_0(t)}{dt} = U_{oc} - U(t) - U_0(t) \left(1 + \frac{R_0}{R_d + R_f}\right) \]  
  \[ U(t) = U_{oc} - R_d I(t) - U_0(t) \]
- DC/DC chopper converter:

\[ U_a(t) = \alpha(t)U(t) \quad (11.11) \]

\[ U_f(t) = \beta(t)U(t) \quad (11.12) \]

First, in equations (11.3) and (11.5) \( U_i(t) \) and \( T_a(t) \) were replaced with equations (11.6) and (11.7) respectively.

The same was done in equations (11.3) and (11.4), where \( U_a(t) \) and \( U_f(t) \) were replaced with equations (11.11) and (11.12) respectively. Equation (11.8) was transformed by replacing \( U_a(t) \) with equation (11.11).

\[ \alpha(t)U(t) = L_a \frac{dI_a(t)}{dt} + R_a I_a(t) + L_m I_f(t)w_2(t) \quad (11.13) \]

\[ \beta(t)U(t) = L_f \frac{dI_f(t)}{dt} + R_f I_f(t) \quad (11.14) \]

\[ \frac{dw(t)}{dt} = \frac{L_m I_f(t)I_a(t) - T(t)}{\theta_m} \quad (11.15) \]

\[ \alpha(t)U(t)I_a(t) + P_{c(t)} = U(t)I(t) \quad (11.16) \]

Then, \( U(t) \) was replaced with equation (11.10) in equations (11.9), (11.13), (11.14) and (11.16):

\[ R_i C_{di} \frac{dU_0(t)}{dt} = R_i I(t) + U_0(t) - U_0(t) \left( 1 + \frac{R_0}{R_d + R_{ci}} \right) \quad (11.17) \]

\[ \alpha(t)[U_0 - R_i I(t) - U_0(t)] = L_a \frac{dI_a(t)}{dt} + R_a I_a(t) + L_m I_f(t)w_2(t) \quad (11.18) \]

\[ \beta(t)[U_0 - R_i I(t) - U_0(t)] = L_f \frac{dI_f(t)}{dt} + R_f I_f(t) \quad (11.19) \]

\[ R_i I^2(t) + [U_0(t) - U_0 - \alpha(t)R_i I_a(t)]I(t) + P_{c(t)} + \alpha(t)[U_0 - U_0(t)]I_a(t) = 0 \quad (11.20) \]

Finally, leaving the unknown derivatives on the left hand side of the equations and adding the mechanical equation corresponding to the derivative of the front axle rotation angle (angular speed), the final system is:

\[ \frac{dI_a(t)}{dt} = \frac{\alpha(t)}{L_a} [U_0 - R_i I(t) - U_0(t)] - \frac{R_a}{L_a} I_a(t) - \frac{L_m}{L_a} I_f(t)w_2(t) \quad (11.21) \]

\[ \frac{dI_f(t)}{dt} = \frac{\beta(t)}{L_f} [U_0 - R_i I(t) - U_0(t)] - \frac{R_f}{L_f} I_f(t) \quad (11.22) \]
\[
\frac{dU_0(t)}{dt} = \frac{1}{C_{dl}} \left( I(t) - \frac{U_0(t)}{R_i} \right) \tag{11.23}
\]
\[
\frac{dw_m(t)}{dt} = \frac{L_s I_f(t) I_u(t) - T(t)}{\theta_m} \tag{11.24}
\]
\[
\frac{d\varphi_m(t)}{dt} = w_m(t) \tag{11.25}
\]

\(I(t)\) is obtained by means of the quadratic expression (11.20).

### 11.3.2. First Integration of Electric Vehicle Equations

Before coupling both subsystems, a preliminary mathematical model of the electric powertrain alone was programmed and tested in MATLAB.

For this purpose, the system of equations above was implemented in MATLAB code and simulated with real motor and battery parameters. In this first approach, the equivalent inertia force needed for acceleration was taken into account as an additional resistance torque applied on the motor shaft, different from rolling resistance, aerodynamic drag and grading resistance. Then, the load torque equation has the following summands:

\[
T(t) = R_a + R_r + R_g + F_{ie} \tag{11.26}
\]

where \(R_a\) is the aerodynamic resistance, \(R_r\) is the tire rolling resistance, \(R_g\) is the resistance due to road slope and \(F_{ie}\) is the equivalent inertia force during acceleration. This expression was later included in equation (11.24).

The MATLAB scripts created for the electric system integration are listed below:

a) Main program:

- sistemaElectrico.m

b) Main functions:

- deriv.m
- loadTorque.m
- wheelTorques.m

c) Auxiliary functions:

- alpha.m
- beta.m
- I.m
- U0.m

For integrating the equations, the main program sistemaElectrico was created in MATLAB. In his script the electrical parameters are set, the initial state vector
\[ y_0 = [I_a, I_f, U_0, w_0, \varphi_0] \] is computed as well as the integration interval, and the call to integrator ode45 is done:

\[ [T, Y] = \text{ode45}(\text{deriv}, tspan, y0, \text{options}, Ra, La, Lm, Rf, Lf, R0, Rl, Cd1, Thm, Ncells, Q0, r, m, i0, eff); \]

Then, function deriv was built with the electrical differential equations and their respective parameters, which were computed by programming and calling auxiliary functions such as alpha, beta, I, and U0. The auxiliary functions will be referred and explained further on in the text.

Function loadTorque returns the value of \( T(t) \) computed by means of equation (11.26), where \( R_a, R_r \) and \( R_g \) are calculated with basic vehicle longitudinal dynamic equations (see Section 4.2), and \( F_i \) has an approximate value of 2000 kg*1.4 m/s².

Finally, function wheelTorques returns the value of the torque provided by the electric DC motor calculated with equation (11.7):

\[
\text{function } \tau = \text{wheelTorques}(t, T_{node}, T_{bodyTnode}, I_a, I_f, L_m, w, r, m) \\
\text{parmotor} = I_a*I_f*L_m; \\
\tau = \text{[parmotor, parmotor, 0, 0]'}; \\
\]

A few simulations were run in order to achieve good integration performance and stable variable response. Once the results were verified, the electrical subsystem was ready to be coupled to MBS3D software and integrated together with the mechanical equations.

11.3.3. Electrical Parameter Setting and Gear Ratio Adjustment

A deep research was done in order to find suitable parameter values for the electric motor and the battery pack. After studying and comparing many different prototypes, the values were set and adjusted in such a way that the response of the powertrain model was adequate and close to real vehicle behavior.

Electric Motor Parameters

The original data of the electric motor model was maintained within its order of magnitude, although the parameter values were changed after coupling both subsystems so that the torque and speed profiles were appropriate for vehicle performance characteristics.

DC/DC Chopper Field Weakening Control Technique

As it has already been explained, the DC/DC chopper converter output voltage was modeled as a linear function of the gas pedal input. Also, it was designed to control the field current by means of a field weakening control technique (FWC). In this case, the chopper duty cycle is varied depending on the motor shaft speed. The FWC technique was designed after running several simulations and adjusting the duty cycle in order to obtain good power output rating. The implemented FWC is shown below:

\[
\beta(t) = \frac{I_f R_f}{U_\infty - R_n I - U_0}, \text{ if } \omega_m \leq 2800 \text{ rpm} \\
(11.27)
\]

\[
\beta(t) = \frac{I_f R_f}{U_\infty - R_n I - U_0} - 0.01, \text{ if } 4000 \text{ rpm} \geq \omega_m > 2800 \text{ rpm} \\
(11.28)
\]
\[
\beta(t) = \frac{I_f R_f}{U_{oc} - R_0 I - U_0}, \quad \text{if } 4300 \geq \omega_m > 4000 \text{ rpm} \quad (11.29)
\]

\[
\beta(t) = \frac{I_f R_f}{U_{oc} - R_0 I - U_0} - 0.005, \quad \text{if } \omega_m > 4300 \text{ rpm} \quad (11.30)
\]

With this field weakening control law, if the car speed was maintained between 2800 rpm and 4000 rpm or above 4300 rpm during long periods of time, \( \beta(t) \) could reach very low values and even become negative. Consequently, the field current \( I_f \) would also be negative causing a sudden braking maneuver. Therefore, in order to avoid undesired braking torque, it was necessary to maintain \( I_f \) above a minimum positive value. This condition was added to equations (11.28) and (11.30) and it is expressed as follows:

\[
\beta(t) = (11.27) \text{ or } (11.30), \quad \text{if } I_f > 0.1 \quad (11.31)
\]

\[
\beta(t) = \frac{I_f R_f}{U_{oc} - R_0 I - U_0}, \quad \text{if } I_f \leq 0.1 \quad (11.32)
\]

**Battery Pack Circuit Parameters**

The battery pack was designed according to electric motor characteristics since both devices battery and electric motor need to have similar values of nominal voltage. Battery cell characteristics correspond to those of a typical high power Lithium-ion cell. Cell parameters such as resistance and capacity are usually temperature and state of charge dependent. However, in this model the operating temperature is considered constant \( T=40^\circ\text{C} \). As it is shown in Figure 11.4, Figure 11.5 and Figure 11.6 (T. Huria, M. Ceraolo, J. Gazzarri, R. Jackey, 2012), at this temperature the variation of cell characteristics with the state of charge is not significant, so in the present design cell parameters have a constant value.
The battery pack should have a nominal voltage of 320 V to suit electric motor characteristics. Since each battery cell has a nominal voltage of 3.7 V, 320 V/3.7 V = 86.49 cells connected in series are needed at least to fulfill this specification. Besides, each cell has a nominal capacity of 31 Ah. This means that two cells connected in parallel will provide 31 Ah x 2 = 62 Ah (20.2 kWh), which is enough capacity for an electric vehicle peaking power source.

On the basis of all the above, it was decided to arrange 176 Lithium-ion cells in 44 modules with 4 cells in each module in a 2 x 2 configuration (Figure 11.7).

Then the values of the complete battery pack parameters are calculated according to the cell arrangement as follows:

\[
\frac{1}{R_{0T}} = \frac{1}{R_0} + \frac{1}{88R_0} \quad \text{or} \quad R_{0T} = 44R_0 \tag{11.33}
\]

\[
\frac{1}{R_{1T}} = \frac{1}{88R_1} + \frac{1}{88R_1} \quad \text{or} \quad R_{1T} = 44R_{1T} \tag{11.34}
\]
\[ C_{dir} = \frac{C_{dl}}{88} + \frac{C_{dt}}{88}; \quad C_{dir} = \frac{C_{dt}}{44} \] (11.35)

\[ Q_0 = 2Q_{cell} \] (11.36)

\[ V_n = 88V_{n,cell} \] (11.37)

The battery pack specifications are summed up in Table 11.1.

**Table 11.1. Battery pack specifications**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{0T} )</td>
<td>0.396 Ω</td>
</tr>
<tr>
<td>( R_{1T} )</td>
<td>0.066 Ω</td>
</tr>
<tr>
<td>( C_{dtT} )</td>
<td>818.182 F</td>
</tr>
<tr>
<td>( Q_0 )</td>
<td>62 Ah</td>
</tr>
<tr>
<td>( V_n )</td>
<td>325.6 V</td>
</tr>
<tr>
<td>( P_{max} )</td>
<td>85 kW</td>
</tr>
</tbody>
</table>

**Gear Ratio Adjustment**

The gear reduction ratio was also adjusted by calculating and testing different values in MBS3D. After several simulations, the graphics showed that the gear ratio which gave the best vehicle performance would be between 8 and 9. Therefore, using the gear ratio equation

\[ i_g = \frac{\pi n_{m,\text{max}} r_w}{30V_{\text{max}}} \] (11.38)

with the wheel radius \( r_w = 0.437264 \) m, with a gear reduction ratio of \( i_g = 8.25 \), the vehicle will reach its maximum speed \( V_{\text{max}} = 160 \) km/h when the motor shaft rotates at \( \omega_{\text{max}} = 5000 \) rpm.

**11.3.4. 'Tight Coupling' Applied to MBS3D and Electrical Subsystem**

After integrating the electrical subsystem under different conditions and adjusting its parameters for better vehicle performance, the coupling of equations was done. As it has already been mentioned, tight coupling method was applied for this purpose.

The main steps taken to apply tight coupling were: extension of the state vector with the new electrical variables, programming of necessary additional functions and modification of the existing functions in MBS3D.
**Modification of the State Vector**

The original state vector for the car model in MBS3D had 48 mechanical variables corresponding to the relative coordinates of the different parts of the vehicle body, such as the chassis or the wheels. Initially, all the electrical differential equations were coupled to MBS3D model, what added 5 new variables to the state vector: \( I_a, I_f, U_0, \omega_m \) and \( \phi_m \). Then, these variables were reduced to 3 (\( I_a, I_f \) and \( U_0 \)), since the motor rotational speed could be calculated from the wheels speed, which is an output value from MBS3D software. The latter was done considering the relationship between the front wheels speed and the motor shaft speed given by the equation of the car differential:

\[
\omega_m = \frac{\omega_{FRW} + \omega_{FLW}}{2} \quad (11.39)
\]

where \( \omega_{FRW} \) and \( \omega_{FLW} \) are the front right and front left wheel speeds in rad/s.

The expression above was inserted in equation (11.21) resulting in:

\[
\frac{dI_a(t)}{dt} = \frac{\alpha(t)}{L_a} [U_{ac} - R_0 I(t) - U_0(t)] - \frac{R_m}{L_a} I_a(t) - \frac{L_m}{L_a} I_f(t) \left[ \frac{w_{FRW}(t) + w_{FLW}(t)}{2} \right] \quad (11.40)
\]

Furthermore, MBS3D already considers the load torque acting on the motor shaft, including rolling and aerodynamic resistance and vehicle linear and rotational inertias, due to vehicle mass, rotating components of the drive train and the wheels. Consequently, equations (11.24) and (11.25) disappear when both subsystems are coupled. All this means that the final hybrid state vector contains 48+3 = 51 variables, since 3 more differential equations were added to the original ODE system: electrical equations (11.21), (11.22) and (11.23).

The new state vector for hybrid and electric vehicle dynamic modeling and simulation has the form:

\[
y_{HE}(t) = \begin{bmatrix} \dot{z} \\ z \\ y_{elec} \end{bmatrix} = \begin{bmatrix} \dot{z} \\ z \\ I_a \\ I_f \\ U_0 \end{bmatrix} \quad (11.41)
\]

where \( z \) and \( \dot{z} \) are the vector of relative coordinates of the vehicle bodies and its first order time derivative respectively. Therefore, the whole ODE system to be solved is:

\[
\begin{bmatrix} M(t) \\ M_{elec}(t) \end{bmatrix} y_{HE}(t) = f(t, y) \quad (11.42)
\]
Programming of Auxiliary Functions

Once the full state vector was computed, it was necessary to program additional functions, in order to calculate the factors involved in the electrical differential equations that were not obtained directly from the integration. These factors are: DC/DC chopper converter duty cycles $\alpha(t)$ and $\beta(t)$, battery open circuit voltage $U_{oc}$, battery state of charge $SOC(t)$ and battery output current $I(t)$. To do this, the electric motor, battery and drive train parameters needed first to be included in mbsOptions structure. This structure has the following basic parameters and fields:

**Basic:** contains all the basic parameters related to integration time and number of time steps, which determines the number of output points for the integrator. In the case of a constant step-size integrator, such as rk4, these points will coincide with the state vector computation times. In the case of variable step-size integrators, such as ode113, some of the output points will be computed by interpolation, since the times at which the state vector is computed will not coincide with these outputs.

**Analysis:** the multibody analysis can be either dynamic or kinematic. The parameter solver stores the integrator selected by the user. Other options are positionCorrection, which controls the constraint enforcement during simulation and odeOptions, which determines the relative and absolute integration error tolerances and stores the name of the output function. Additional fields that need to be defined are: extForces (external forces), fncTorques (applied torques), fncDrivenCoor (driven coordinates).

**Post-process:** when the option energy is enabled, the program plots an energy balance in a separate window. If the options reactions is set to yes, the joint reaction forces will be calculated.

**Display:** this includes the animation characteristics animationStep, which can be controlled to decrease the rate of output representation, animation, which controls the simulation playback and the option replay, which can be set to replay several times the animation at the end of the simulation.

**Storage:** if the option storeResults is enabled, the program will save the final position of the simulation in the file specified in storeFile. This position may be the starting position for a later simulation.

To the original structure, the following electrical information was added: battery pack and electric motor data, gear reduction ratio and initial values for the electrical variables $I_a, I_f$ and $z$.

\[
\begin{bmatrix}
M(t) \\ 1 \ 0 \ 0 \\ 0 \ 1 \ 0 \\ 0 \ 0 \ 1
\end{bmatrix}
\begin{bmatrix}
z^2 \\ z^2 I_a \\ I_f U_0
\end{bmatrix}
= \begin{bmatrix}
M(t) \\ 1 \ 0 \ 0 \\ 0 \ 1 \ 0 \\ 0 \ 0 \ 1
\end{bmatrix}
\begin{bmatrix}
z^2 \\ z^2 \\ L_a I_a(t) - L_a I_f(t) \\ 2
\end{bmatrix}
\begin{bmatrix}
\frac{\alpha(t)}{L_a} [U_{oc} - R_0 I(t) - U_0(t)] - R_a I_a(t) - L_a I_f(t) \\ \frac{\beta(t)}{L_f} [U_{oc} - R_0 I(t) - U_0(t)] - \frac{R_f}{L_f} I_f(t) \\ \frac{1}{C_{dl}} [I(t) - \frac{U_{oc}(t)}{R_i}]
\end{bmatrix}
\]

(11.43)
This was done by creating two new fields in mbsOptions: electricMotor and battery, where the new data was stored. An example of data storage in mbsOptions is shown below:

```plaintext
mbsOptions.electricMotor.Ra=0.516; % Motor armature resistance
mbsOptions.electricMotor.La=0.02; % Motor armature inductance
mbsOptions.battery.Nmod=44; % Number of modules of the battery
mbsOptions.battery.R0=0.009*Nmod; % Battery ohmic resistance
```

Finally, another field containing the initial values for the electrical variables was added to mbsOptions structure:

```plaintext
Ia0=0; If0=0.4; U00=0;
mbsOptions.yiniHE = [Ia0, If0, U00]';
```

The complete mbsOptions structure is shown below.

```plaintext
mbsOptions =
    name: 'Car2'
    animationStep: 2
    analysis: 'dynamic'
    kinInput: @car2kinInput1
    solver: 'ode113'
    deriv: @RTDyn4mHE
    extForces: @car2UserForcesHE
    fncDrivenCoor: @drCoorPos2
    fncTorques: @wheelTorquesHE
    wheel: [1x1 struct]
    positionCorrection: 'no'
    odeOptions: [1x1 struct]
    energy: 'no'
    reactions: 'no'
    stats: 'yes'
    animation: 'no'
    replay: 'no'
    viewDir: [38 25]
    scaleWF: 1.000000000000000e-03
    graphicRefPoint: [6 1]
    unitVectScale: 0.300000000000000
    pauseVal: 1.000000000000000e-05
    storeResults: 'no'
    storeFile: 'data2.mat'
    sprdmp: [1x2 struct]
    battery: [1x1 struct]
    electricMotor: [1x1 struct]
    yiniHE: [3x1 double]
```

After enlarging mbsOptions structure with the electrical parameters, six new functions were programmed and added to MBS3D folders. These functions are summarized and briefly explained in Table 11.2.
### Table 11.2. Auxiliary functions programmed for completing the electrical vehicle model

<table>
<thead>
<tr>
<th>MATLAB function</th>
<th>Function call &amp; Returned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>alfa</td>
<td>(a = \alpha(t)) (\text{Returns the value of } \alpha(t) \text{ which is the chopper duty cycle calculated as a linear function of the gas pedal input. Its value can vary between -1 and 1, since it also depends on the brake pedal position. It is a non-dimensional number.})</td>
</tr>
<tr>
<td>beta</td>
<td>(b = \beta(t, y, \text{OCV, I2, mbsOptions})) (\text{Returns the value of } \beta(t) \text{, which is the chopper duty cycle calculated as a function of the motor field parameters. It depends on the field weakening control technique and it varies according to the motor shaft rotational speed. It is a non-dimensional number.})</td>
</tr>
<tr>
<td>Uoc</td>
<td>(\text{OCV} = \text{Uoc}(t, \text{Ncells, mbsOptions})) (\text{Returns the value of } U_{oc} \text{ in volts, which is the open circuit voltage of the battery calculated as a function of the initial state of charge.})</td>
</tr>
<tr>
<td>(\text{I})</td>
<td>([\text{I1, I2}] = \text{I}(\text{Ia, U0, OCV, R0, PLC, a})) (\text{Returns the value of } I(t) \text{ in amperes, which is the battery current calculated by means of a quadratic equation as a function of } U_0(t), U_{oc}, \alpha(t), I_a(t), P_{dc}(t)\text{ and } R_0. \text{ It returns two values, but only one is valid and it has been determined through model simulations.})</td>
</tr>
<tr>
<td>StateOfCharge</td>
<td>(\text{SOC} = \text{StateOfCharge}(t, \text{SOC0, I2, Q0})) (\text{Returns the value of } \text{SOC}(t) \text{ which is the current state of charge of the battery calculated approximately as a function of the battery initial state of charge } \text{SOC0}, \text{ the battery current output } I(t), \text{ the battery capacity } Q_0, \text{ and the charging efficiency } \eta_c. \text{ It is a non-dimensional number.})</td>
</tr>
<tr>
<td>powerLossesC</td>
<td>(\text{PLC} = \text{powerLossesC}(t)) (\text{Returns the value of } P_{dc}(t) \text{ in watts, which represents the power losses in the chopper converter. In this work it is considered constant, although a more precise model could be done by taking into account input and output currents to the chopper converter.})</td>
</tr>
</tbody>
</table>

### Modification of Existing Functions in MBS3D

Since the state vector was changed, some of the functions of MBS3D software needed to be modified as well, in order to integrate correctly all the equations. The main changes in these functions are explained below:

- **car2HE.m** is the main program where the call to mbs3dHE is done.
  - Extension of \text{mbsOptions} structure with the data corresponding to the electric traction motor, the battery pack and the drivetrain components.
  - Call to MBS3D program through the modified function mbs3dHE.
mb3dHE.m is the main function and it carries out the simulation. Here the initial state vector is computed and passed as an input argument to the integrators.

- Extension of the initial state vector \( y_0 \). This vector originally contained the initial positions and independent velocities of the vehicle bodies:
  \[ y_0 = [\text{zpos}; \text{zvel}(\text{izInd})] \]. Later, the initial electrical variables stored in mbsOptions were added and the new state vector was computed:
  \[ y_{0\text{HE}} = [\text{zpos}; \text{zvel}(\text{izInd}); \text{mbsOptions.yiniHE}] \].
  The old vector \( y_0 \), which is an input argument to the integrators, was replaced by the new one \( y_{0\text{HE}} \) in the calls to the integrators.

- Add of calls to other integrators than the included originally: \( \text{rk4bHE}, \text{ode45}, \text{ode23}, \text{ode23s} \) and \( \text{ode23t} \).

- Creation of new figures showing the electric powertrain results: front wheels speed, electric motor armature and field currents, electric motor power and torque outputs, battery voltage and current, battery power output, battery state of charge gas pedal command and field weakening control command.

rk4bHE.m receives the initial state vector \( y_{0\text{HE}} \) as an input argument and calls RTDyn4mHE in order to integrate the differential equations. It uses Runge-Kutta fourth order integration method. No significant modifications have been done in this function.

RTDyn4mHE.m receives the state vector \( y \) as an input argument and computes its first time order derivative.

- Extension of the state vector \( y \) by adding the electrical components after the mechanical components at the end of the original vector:
  \[ y_{\text{Elec}} = y(\text{nocJoints+nzInd+1:nocJoints+nzInd+nElec}), \]
  where \( n_{\text{elec}} = 3 \) is the number of electrical variables:
  \[ n_{\text{Elec}} = \text{length(mbsOptions.yiniHE)} \].

- Computation of the necessary function calls and initialization of electrical unknowns in order to obtain the factors needed for the integration of the electrical differential equations.

- Computation of the electrical differential equations:
  \[
  \begin{align*}
  y_{\text{Elecp}}(1) &= (OCV-R_0*\text{Ibatt-U}_0)*a/L_a - I_a*R_a/L_a - I_f*L_m*rt*(y(39)+y(41))/(2*L_a); \\
  y_{\text{Elecp}}(2) &= (OCV-R_0*\text{Ibatt-U}_0)*b/L_f - I_f*R_f/L_f; \\
  y_{\text{Elecp}}(3) &= (1/C_{dl})*(\text{Ibatt-U}_0/R_1); \\
  \end{align*}
  \]
  where \( y_{\text{Elecp}} = [\frac{dI_a}{dt}, \frac{dI_f}{dt}, \frac{dU_0}{dt}] \) is a vector containing the electrical system derivatives.

- Computation of the new state vector derivative by adding the electrical system derivatives to the mechanical state vector derivative:
  \[
  \begin{align*}
  dy_{\text{HE}} &= \text{zeros(nocJoints+nzInd+nElec,1)}; \\
  dy_{\text{HE}}(1:\text{nocJoints}) &= y_{\text{full}}(\text{nocJoints+1:2*nocJoints}); \\
  dy_{\text{HE}}(\text{nocJoints+1:nocJoints+nzInd}) &= z_{\text{AccInd}(1:nzInd)}; \\
  dy_{\text{HE}}(\text{nocJoints+nzInd+1:nocJoints+nzInd+nElec}) &= y_{\text{Elecp}}'; \\
  \end{align*}
  \]
The new state vector derivative now contains all the dependent open-chain relative velocities, the independent relative accelerations and the electrical state vector derivatives.

*wheelTorquesHE.m* receives as input arguments some of the values obtained from the electrical equations integration, \( I_a \) and \( I_f \), and computes the torques applied on the front wheels of the vehicle.

- Initially, in *wheelTorques.m* the torque was computed as a vector with four components in which the first two had a positive constant value and the last two were zero. This resulted in constant torque applied to the front wheels as follows:
  \[
  \text{tau} = [300, 300, 0, 0]'.
  \]
  In *wheelTorquesHE.m*, the torque provided by the DC is computed according to the separately excited DC motor equation:
  \[
  T_m(t) = I_a(t)I_f(t)L_m,
  \]
  where \( L_m \) is the armature-field mutual inductance. Furthermore, the gear reduction ratio and transmission efficiency are taken into account for this calculation, what yields:

  \[
  \begin{align*}
  \text{motorTorque} & = I_a*I_f*L_m; \\
  \text{totalTorque} & = \text{motorTorque}*\text{teff}*\text{rt}; \\
  \text{tau} & = [\text{totalTorque}, \text{totalTorque}, 0, 0]';
  \end{align*}
  \]

  where, \( \text{teff} = 0.98 \) is the transmission efficiency and \( r_t = 8.25 \) is the gear reduction ratio, which is stored in \text{mbsOptions} as \( i_0 \):

  \[
  \text{rt} = \text{mbsOptions.electricMotor.i0}.
  \]

*tireForcesHE.m* computes the external normal force acting on the car wheels, as well as wheel center velocities.

- The state vector of electrical variables \( \gamma_{elec} = [I_a, I_f, U_0] \) was added as an input argument for it would be used when calling *wheelTorquesHE* inside this function.

*car2UserForcesHE.m* computes the suspension force acting on the vehicle chassis and the contact forces and moments acting on the wheels.

- The state vector of electrical variables \( \gamma_{elec} = [I_a, I_f, U_0] \) was added as an input argument for it would be used when calling *wheelTorquesHE* inside this function.

*evaluateExternalForcesHE.m* evaluates the external forces in the joints due to springs and dampers and the forces on the rods due to weight.

- The state vector of electrical variables \( \gamma_{elec} = [I_a, I_f, U_0] \) was added as an input argument for it would be used when calling *car2UserForcesHE* inside this function.

*animateRec3HE.m* plots the system position and velocity during the integration.

- The state vector of electrical variables \( \gamma_{elec} = [I_a, I_f, U_0] \) is computed again in this function, since it will be an input argument to *tireForcesHE*, which is called inside *animateRec3HE* during the integration.
11.4. Complete Electric Vehicle Model

![Diagram](diagram.png)

*Figure 11.8. Approach of MBS3D function-call process, main input and output arguments and tasks of the main functions*
Figure 11.8 shows the complete computational approach of the electric vehicle model. It reflects how the function-call is carried out and the main input arguments and return values of all the functions.

11.5. Translation into C/C++ Code

After verifying the complete electric vehicle model, it was observed that the integration times were too long for vehicle simulation purposes (the real integration time was approximately ten times longer than the simulation time frame). Therefore, it was necessary to think of a way to improve the overall model efficiency. The most immediate existing tool was to extend the software MBS3D 2.0 Pro, where some of the scripts were written in C/C++ code. This would imply the translation of all the new MATLAB functions into C/C++ language, as well as the modification of the existing files.

Microsoft Visual Studio was the IDE for programming the functions in C/C++ code and the same steps as outlined for MATLAB model implementation applied: extension of the state vector with the new electrical variables, programming of necessary additional functions and modification of the existing functions in MBS3D 2.0 Pro. Additionally, mbsOptions structure in C/C++ gets the data directly from the main program in MATLAB code and transforms it in a .cpp file called transformMbsOptions_mbsOptionsCHE. This program was extended in the same way as new fields were added to mbsOptions structure, but the transformation and storage of the variables of mbsOptions to C/C++ variables in the new structure mbsOptionsCHE is done simultaneously. The example below shows the C/C++ code used for getting the battery data from the original structure mbsOptionsHE, transforming and storing it in the new C/C++ structure mbsOptionsCHE.

```c++
if ( (tmp=mxGetField(mbsOptionsHE, 0, "battery")) != NULL ) {
    tmp2 = mxGetField(tmp, 0, "Nmod");
    (*mbsOptionsCHE).Battery.Nmod = mxGetScalar(tmp2);
    tmp2 = mxGetField(tmp, 0, "Ncells");
    (*mbsOptionsCHE).Battery.Ncells = mxGetScalar(tmp2);
    tmp2 = mxGetField(tmp, 0, "R0");
    (*mbsOptionsCHE).Battery.R0 = mxGetScalar(tmp2);
    tmp2 = mxGetField(tmp, 0, "R1");
    (*mbsOptionsCHE).Battery.R1 = mxGetScalar(tmp2);
    tmp2 = mxGetField(tmp, 0, "Cdl");
    (*mbsOptionsCHE).Battery.Cdl = mxGetScalar(tmp2);
    tmp2 = mxGetField(tmp, 0, "Q0");
    (*mbsOptionsCHE).Battery.Q0 = mxGetScalar(tmp2);
    tmp2 = mxGetField(tmp, 0, "SOC0");
    (*mbsOptionsCHE).Battery.SOC0 = mxGetScalar(tmp2);
}
```

Basic study and research on C/C++ code was done before starting the translation of the scripts. The statement of functions and variables in C/C++ was reviewed and especially, pointer handling and operations. Since MBS3D software had already been studied in depth, it was quite simple to understand the same software in C/C++ language. The main difficulty was to verify the programs every time a change was introduced, for it is necessary to compile C/C++ files before being able to run/debug them and most of the times the errors were not easily detected.
After completing the translation into C/C++ language, a slight change was introduced in the main program: the option deriv was changed from @RTDyn4mHE, where the MATLAB integration files are called, to @RTDyn_car2HE so that the integration was carried out by the programs in C/C++ code.

```matlab
%mbsOptions.deriv = @RTDyn4mHE;
mbsOptions.deriv = @RTDyn_car2HE;
```

As it will be shown in the results, the translation of MATLAB code into C/C++ code reduced the integration times by two orders of magnitude. This represents a great benefit and fulfills one of the main goals of this work: to save manufacturing costs and simulation time of real vehicle prototypes with an efficient mathematical model.
Co-simulation Process
12. Results

The following simulations have been performed with different numerical integrators, integration parameters (integration output step and interval, graphics on/off, C/C++ or MATLAB derivative function), and initial conditions of the problem (initial velocity and initial state of charge). All these data can be modified by the user in the main program `car2HE.m`. The specific characteristics of the numerical integrators, such as absolute and relative error tolerances, must be modified generally in the function that carries out the simulation `mbs3dHE.m`.

12.1. Comparison of Integration Time MATLAB - C/C++

First, a short simulation has been run in order to show the difference between MATLAB code and C/C++ code integration time. This has been done by calling the corresponding differential equations programmed either in MATLAB code or C/C++ code. The integrators used during these tests are `rk4bHE` and `ode113`.

The main parameters for the integration are:

- Integration step: $1\cdot10^{-3}$
- Integration interval: 2 s

<table>
<thead>
<tr>
<th></th>
<th>Graphics ON</th>
<th>Graphics OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MATLAB</td>
<td>C/C++</td>
</tr>
<tr>
<td><strong>rk4bHE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>231.989 s</td>
<td>62.645 s</td>
</tr>
<tr>
<td></td>
<td>204.390 s</td>
<td>56.152 s</td>
</tr>
<tr>
<td></td>
<td>216.091 s</td>
<td>60.620 s</td>
</tr>
<tr>
<td><strong>Average integration time</strong></td>
<td><strong>217.5 s</strong></td>
<td><strong>59.8 s</strong></td>
</tr>
<tr>
<td><strong>ode113</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>228.685 s</td>
<td>121.387 s</td>
</tr>
<tr>
<td></td>
<td>206.051 s</td>
<td>107.963 s</td>
</tr>
<tr>
<td></td>
<td>222.420 s</td>
<td>119.073 s</td>
</tr>
<tr>
<td><strong>Average integration time</strong></td>
<td><strong>219.1 s</strong></td>
<td><strong>116.1 s</strong></td>
</tr>
</tbody>
</table>

Table 12.1 shows a reduction of up to two orders of magnitude in the integration time between the simulations with the core functions written in MATLAB code or C/C++ code. The integration time also decreases significantly when the function in charge of plotting the system position, `animateRec3HE.m`, is disabled (graphics off).
12.2. Study and Comparison of MATLAB Integrators

In the following simulations, the concept of stability refers to the oscillatory behavior of the integrated variables. This means that the dynamic response of a determined variable has been considered stable when it does not show undesired oscillations, peaks or fluctuations. The simulations which have shown a stable behavior with a determined integrator and under the first integration output step \(1 \times 10^{-3}\) have not been run again and will be marked with a hyphen "-" in the respective tables.

It is also important to clarify the difference between integration step and integration output step. The integration step determines the times at which the integrator will compute the state vector. The integration output step defines the times at which the numerical integrator computes the outputs of the integration. The output step can be calculated by means of the following equation:

\[
\frac{\text{Integration time}}{\text{Number of time steps}} = \frac{t_{\text{final}} - t_0}{n_{\text{Int}}} \quad (12.1)
\]

In constant step-size integrators, such as rk4 integrator, both the integration step and the integration output step are equal. On the contrary, in variable step-size integrators these parameters do not coincide and some of the output points will be computed by interpolation in order to fulfill the \(n_{\text{Int}}\) output requirement. The number \(n_{\text{Int}}\) is set by the user in the main program car2HE.m.

a) Initial velocity = 0 m/s, integration interval = 0.5 s, C/C++ files:

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Integration time (s)</th>
<th>Stability</th>
<th>Integration time (s)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>rk4bHE</td>
<td>29.346</td>
<td>NO</td>
<td>281.9</td>
<td>NO</td>
</tr>
<tr>
<td>ode113</td>
<td>68.775</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode45</td>
<td>70.619</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode23</td>
<td>54.505</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode15s</td>
<td>41.354</td>
<td>NO</td>
<td>41.627</td>
<td>NO</td>
</tr>
<tr>
<td>ode23s</td>
<td>96.885</td>
<td>YES(^{(1)})</td>
<td>320.973</td>
<td>YES(^{(2)})</td>
</tr>
<tr>
<td>ode23t</td>
<td>30.181</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1 \times 10^{-3})</td>
<td>(1 \times 10^{-4})</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^{(1)}\) The graphics show a stable behavior for all variables except for the field current \(I_f\), which has frequent fluctuations that indicate certain instability.

\(^{(2)}\) \(I_f\) presents less fluctuations by reducing the integration output step.

---

\(^{i}\) All the simulations in this and the following sections have been done with an initial battery state of charge of SOC = 1 and under full throttle condition (gas pedal position = 100%).
Figure 12.1. Fluctuations in the field current during the integration with ode23s

Figure 12.2. Improved dynamic response of $I_f$ with ode23s when reducing the integration output step

Figure 12.1 and Figure 12.2 show the improvement in the stability of $I_f$ when reducing the integration output step size from $1 \times 10^{-2}$ to $1 \times 10^{-3}$.

The graphics also show that the solution with rk4bHE numerical integrator is not stable and the front wheels speed has great oscillations with strongly marked peaks.

Furthermore, in this case it is not possible to complete the integration with ode15s due to the use of wrong absolute and relative error tolerances. The value given originally to the error tolerances is not appropriate for those that the integrated variables reach during the simulation and, consequently, the integration process cannot be performed. These parameters will be later modified and adjusted properly.

b) Initial velocity = 0.4 m/s, integration interval = 0.5 s, C/C++ files:

Table 12.3. Integration time and stability of the response with different integrators under the conditions specified in b)

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Integration time (s)</th>
<th>Stability</th>
<th>Integration time (s)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>rk4bHE</td>
<td>29.780</td>
<td>NO</td>
<td>273.443</td>
<td>NO$^{(4)}$</td>
</tr>
<tr>
<td>ode113</td>
<td>37.896</td>
<td>NO$^{(3)}$</td>
<td>273.440</td>
<td>NO</td>
</tr>
<tr>
<td>ode45</td>
<td>39.941</td>
<td>NO</td>
<td>297.462</td>
<td>NO</td>
</tr>
<tr>
<td>ode23</td>
<td>33.459</td>
<td>NO</td>
<td>268.286</td>
<td>NO</td>
</tr>
<tr>
<td>ode15s</td>
<td>29.190</td>
<td>NO</td>
<td>252.051</td>
<td>NO</td>
</tr>
<tr>
<td>ode23s</td>
<td>33.309</td>
<td>NO</td>
<td>255.646</td>
<td>NO</td>
</tr>
<tr>
<td>ode23t</td>
<td>28.982</td>
<td>NO</td>
<td>264.489</td>
<td>NO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Integration output step</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 \times 10^{-3}$</td>
</tr>
<tr>
<td>$1 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Julia Carolina Veintimilla Porlán
(3) All the integrators, including rk4bHE, show great oscillations between 0.12 s and 0.22 s, probably due to a very high peak value of some of the integrated variables, reflected in Figure 12.3.

(4) The accuracy of the response with rk4bHE integrator improves greatly when the integration step is reduced, although it still presents a great fluctuation between 0.12 s and 0.22 s.

The oscillation represented in Figure 12.3 is observed in the simulation with all the integrators listed in the table above. It is caused by the peak values that the integrated variables reach during the simulation, as a consequence of the computed initial vehicle velocity 0.4 m/s. Therefore, it is a problem associated to the values of the system derivatives rather than to the numerical integrator. This means that the problem is mostly related to the developed physical model and not to the mathematical tool.

c) Initial velocity = 0.8 m/s, integration interval = 0.5 s, C/C++ files:

Table 12.4. Integration time and stability of the response with different integrators under the conditions specified in c)

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Integration time (s)</th>
<th>Stability</th>
<th>Integration time (s)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>rk4bHE</td>
<td>27.446</td>
<td>NO</td>
<td>270.291</td>
<td>YES</td>
</tr>
<tr>
<td>ode113</td>
<td>35.916</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode45</td>
<td>37.035</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode23</td>
<td>31.983</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode15s</td>
<td>28.434</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode23s</td>
<td>32.676</td>
<td>YES(5)</td>
<td>246.046</td>
<td>YES(6)</td>
</tr>
<tr>
<td>ode23t</td>
<td>28.853</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Integration output step

\[1 \times 10^{-3}\]

\[1 \times 10^{-4}\]
The field current $I_f$ shows again an unstable behavior.

The stability of the field current $I_f$ is significantly improved by reducing the integration output step.

The integration with rk4bHE yields a stable response when the integration step is reduced from $1 \times 10^{-3}$ to $1 \times 10^{-4}$. This means that, from a minimum value of the initial velocity and with an appropriate integration step-size, it is possible to reach the stability of the solution with rk4bHE integrator, however, affecting the integration time which increases by one order of magnitude. This type of instability is associated with the numerical integrator. Figure 12.4 and Figure 12.5 show this behavior.

d) Initial velocity = 1.0 m/s, integration interval = 0.5 s, C/C++ files:

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Integration time (s)</th>
<th>Stability</th>
<th>Integration time (s)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>rk4bHE</td>
<td>27.128</td>
<td>NO</td>
<td>262.913</td>
<td>YES</td>
</tr>
<tr>
<td>ode113</td>
<td>31.039</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode45</td>
<td>34.060</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode23</td>
<td>29.129</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode15s</td>
<td>27.53</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ode23s</td>
<td>30.969</td>
<td>YES(7)</td>
<td>241.357</td>
<td>YES(8)</td>
</tr>
<tr>
<td>ode23t</td>
<td>30.004</td>
<td>YES</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

$1 \times 10^{-3}$ $1 \times 10^{-4}$

Integration output step

(7) The field current $I_f$ shows again an unstable behavior.
The stability of the field current $I_f$ is again significantly improved by decreasing the integration output step.

As in the previous case, the solution with rk4bHE becomes completely stable when reducing the integration step size by one order of magnitude.

e) Initial velocity = 1.0 m/s, integration interval = 5 s, C/C++ files:

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Integration time (s)</th>
<th>Stability</th>
<th>Integration time (s)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>rk4bHE</td>
<td>264.710</td>
<td>NO(9)</td>
<td>2892.807</td>
<td>YES</td>
</tr>
<tr>
<td>ode113</td>
<td>275.809</td>
<td>YES</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ode45</td>
<td>294.726</td>
<td>YES</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ode23</td>
<td>264.164</td>
<td>YES</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ode15s</td>
<td>253.177</td>
<td>YES</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ode23s</td>
<td>408.286</td>
<td>YES</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ode23t</td>
<td>318.809</td>
<td>YES</td>
<td>1.10^{-3}</td>
<td>1.10^{-4}</td>
</tr>
</tbody>
</table>

Integration output step

(9) The graphics show unstable behavior only between 0 s and 1.5 s which is again corrected by reducing the integration step.

f) Initial velocity = 0 m/s, integration interval = 10 s, C/C++ files:

In this case, only ode113, ode45, ode23s and ode23t have been tested, since these integrators show stability starting from zero initial velocity, during long simulation times and with a large integration output step. Therefore, it is possible to obtain an overall view of the electric variables, such as motor and battery currents and voltages, within a reasonable integration time due to the high integration output step supported by these integrators.

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Integration time (s)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ode113</td>
<td>107.068</td>
<td>YES</td>
</tr>
<tr>
<td>ode45</td>
<td>136.144</td>
<td>YES</td>
</tr>
<tr>
<td>ode23s</td>
<td>693.510</td>
<td>YES(10)</td>
</tr>
<tr>
<td>ode15s</td>
<td>65.402</td>
<td>YES</td>
</tr>
<tr>
<td>ode23t</td>
<td>70.608</td>
<td>YES</td>
</tr>
</tbody>
</table>

Integration output step

(10) The graphics show unstable behavior only between 0 s and 1.5 s which is again corrected by reducing the integration step.
The graphics show that all the variables have a stable behavior although the integrator may find some difficulties between 4 s and 6.5 s, resulting in longer integration time than ode113 and ode45.

g) Corrections and adjustments:

Originally, after reducing the output step size in simulations a) and b), integrators ode15s and ode23s did not experience improved stability. Therefore, the possibility of reaching a better response by modifying relative and absolute error tolerance was considered. Some of the simulations above were applied again to these integrators testing different error tolerance values, obtaining enhanced results in some cases:

- Integration with ode15s became completely stable in cases a), b), c) and d) when relative tolerance was reduced from $1 \cdot 10^{-6}$ to $1 \cdot 10^{-7}$ and absolute tolerance was increased from $1 \cdot 10^{-6}$ to $1 \cdot 10^{-4}$.
- Integration with ode23s was more stable in case a) when relative and absolute tolerances were changed from $1 \cdot 10^{-4}$ and $1 \cdot 10^{-5}$ to $1 \cdot 10^{-8}$ and $1 \cdot 10^{-3}$ respectively, although it still presented some oscillations and other signs of instability. In the remaining cases, no significant improvements were achieved.

This may occur because the values of the solution components never get bigger than the error tolerance and, therefore, it becomes necessary to find an appropriate scale for the values of absolute and relative error tolerances.

12.3. Study of Vehicle Performance Characteristics

The simulations above have been run in order to test and compare different integrators for the multi-domain set of ordinary differential equations. In general, good and significant results can be achieved by varying the initial value of some of the unknowns (the initial velocity in all cases) and also changing the integration output step.

From the perspective of electric car modeling, the variables most affecting vehicle performance characteristics are gas/brake pedal position and initial state of charge of the battery. Several simulations have been executed with the aim of showing differences between performance parameters of the vehicle such as maximum car speed, maximum output power and maximum motor torque. The first three tests are carried out under full throttle condition and different values of initial state of charge of the battery for each simulation. The MATLAB integrator ode113 has been used for all the vehicle performance tests.

The following table reflects the main results of the simulations performed in this section.

<table>
<thead>
<tr>
<th>SOC</th>
<th>$V_{\text{max}}$ (km/h)</th>
<th>$P_{\text{max, EM}}$ (kW)</th>
<th>$T_{\text{max}}$ (Nm)</th>
<th>$P_{\text{max, batt}}$ (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80.880</td>
<td>36.5</td>
<td>240</td>
<td>84</td>
</tr>
<tr>
<td>0.7</td>
<td>75.115</td>
<td>32</td>
<td>220</td>
<td>72</td>
</tr>
<tr>
<td>0.4</td>
<td>70.796</td>
<td>28</td>
<td>205</td>
<td>64</td>
</tr>
</tbody>
</table>
The common parameters for these simulations are:

- Initial velocity = 0 m/s
- Integration interval = 10 s
- Integration output step = $1 \cdot 10^{-2}$
- C/C++ files are used
- Gas pedal position = 1

Figure 12.6 shows the multibody system position and velocity at the end of the simulations. These graphics reflect that the maximum car speed decreases with the initial state of charge.
Figure 12.6. Multibody system position and velocity at the end of the simulations
The graphics above show that, when reducing the initial state of charge of the battery from 1 to 0.7, the electric motor maximum output torque drops from 240 Nm to 220 Nm (Figure 12.7 and Figure 12.8), and the maximum power output decreases from 36.5 kW to 32 kW (Figure 12.9).

The drop is less significant when reducing the initial state of charge from 0.7 to 0.4: the maximum motor torque decreases from 220 Nm to 205 Nm, and the peak power output drops from 32 kW to 28 kW.
The graphics above (see Figure 12.10, Figure 12.11 and Figure 12.12) show that the initial state of charge also affects the peak values of the electric motor operating currents. In particular, the maximum value of the armature current decreases with the initial state of charge. However, the value of state of charge does not change the behavior of the field current since, as it has already been explained, it mainly depends on the applied field weakening control technique.
The graphics above show the expected behavior of the battery current and voltage: both operating parameters reduce their peak values when the initial state of charge is decreased. However, they have similar evolution over time in the three cases (see Figure 12.13, Figure 12.14 and Figure 12.15).

Table 12.9 summarizes the peak values of these battery parameters depending on the initial state of charge.

**Table 12.9. Approximate battery maximum currents and voltages with different values of SOC₀**

<table>
<thead>
<tr>
<th>SOC₀</th>
<th>I&lt;sub&gt;max_batt&lt;/sub&gt; (A)</th>
<th>V&lt;sub&gt;max_batt&lt;/sub&gt; (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>400</td>
<td>280</td>
</tr>
<tr>
<td>0.7</td>
<td>370</td>
<td>270</td>
</tr>
<tr>
<td>0.4</td>
<td>350</td>
<td>260</td>
</tr>
</tbody>
</table>
The battery peak power output is reduced in approximately 20 kW, from 84 kW (Figure 12.16) to 64 kW (Figure 12.18) when the initial state of charge decreases from 100% to 40%. The decline is smaller in the case of an initial state of charge of 70% (Figure 12.17), where the power output drops from 84 kW to 72 kW. The evolution of the battery power output is similar in all cases.
Figure 12.19. State of charge evolution during simulations performed with different values of $SOC_0$. 
The results show that the battery is discharged in a similar way in the three cases represented in Figure 12.19.

The best performance characteristics are achieved when the initial state of charge of the battery is 100%. This is expected because the power provided by the battery pack depends directly on the energy stored. Therefore, the maximum power output from the battery will decrease with the state of charge and it will also affect the maximum torque and power outputs of the electric motor and the maximum vehicle speed.

In percentage terms, when reducing the initial state of charge from 100% to 70%, the maximum battery power output is reduced by 14.28%, the electric motor power output by 12.33%, and the maximum motor torque output by 8.33%. In the case of an initial state of charge of 40%, the reductions are 23.81% for the maximum battery power output, 23.29% for the electric motor peak power and 14.58% for the maximum output torque, with regard to a 100% initial state of charge.

The results of these simulations are shown in the graphics above and summarized in Table 12.8.

12.4. Driving Cycle Implementation

The last test consists of a short driving cycle in an urban environment. This means that the average vehicle speed is around 50 km/h and it is not usually above 60 km/h. The battery state of charge remains practically constant due to frequent regenerative braking, since on a standard urban driving cycle the vehicle has often stops and starts.

The driving cycle implementation in this work has not been performed in a conventional way, where a designed driving cycle is an input to the model. In this case, the vehicle speed is controlled with the gas/brake pedal position, which determines directly the amount of current provided to the electric motor and, therefore, the output traction torque. Then, the inputs to the vehicle model are the driver acceleration and brake commands. As it has already been explained, the gas/brake pedal position is computed in the file alpha.m, where the gas pedal command has been assigned different values over time.

The integrator used for the driving cycle was ode113 and the integration parameters are:

- Integration interval: 60 s
- Integration output step: $1 \cdot 10^{-2}$
- C/C++ files are used

The simulation is run with an initial vehicle velocity of 0 m/s and the initial battery state of charge is 100%.

The final position of the system and the final vehicle speed at the end of the driving cycle simulation are represented in Figure 12.20. In addition Figure 12.21 shows the evolution of the front wheels speed during the driving cycle.
Results

Figure 12.20. Final multibody system position and velocity

Figure 12.21. Front wheels speed during the driving cycle
Figure 12.22. Motor torque and power requirements during the driving cycle

Figure 12.23. Gas/brake pedal power command during the driving cycle

Figure 12.24. Evolution of the motor armature and field currents during the driving cycle

Figure 12.25. Field weakening control command $\beta(t)$

Figure 12.22 and Figure 12.24 plot the variation of the operating parameters of the electric motor according to the gas/brake pedal inputs, computed by the user in alpha.m. The motor torque and power outputs, as well as the armature current show direct proportionality to the power requirement, according to the gas/brake pedal position. The field current behavior responds to the applied field weakening control technique.

The gas/brake pedal command is represented in Figure 12.23. As it can be seen, the driving cycle consists of short and successive acceleration and deceleration processes.

Figure 12.25 reflects the evolution of the field weakening command, which is also a result of the designed control technique in function beta.m.
The battery power output during the driving cycle is also proportional to the power commanded by the driver (Figure 12.26). The position of gas and brake pedals determines the amount of current that needs to be drawn from the battery. Besides, the current and voltage of the battery are related by means of a quadratic expression, as it was explained in Section 11.3.1. The variation of both variables (battery current and voltage) during the driving cycle is represented in Figure 12.28.

The state of charge of the battery decreases according to the amount of power provided by the battery to the electric motor for vehicle propulsion, and it increases during the braking process, due to brake energy regeneration. Figure 12.27 plots the evolution of the state of charge, estimated by “Coulomb counting method” (see Section 9.3.1). As it can be seen, the battery is discharged when the gas is further depressed (acceleration), and charged when this pedal is released or the driver operates the brake pedal (deceleration or braking). Finally, Figure 12.29 shows how the value of the battery ohmic-overpotential changes during the simulation of the driving cycle.
12.5. Simulation with MATLAB Profiler

Finally a test with MATLAB Profiler function has been performed. This function allows tracking execution time when running MATLAB programs, which is very helpful in order to debug and optimize the code. First, a simulation with the original MBS3D car model, without including the electrical equations, was done. Then, the same simulation was run in the electric vehicle model. In this way, it is possible to visualize the CPU time distribution among the different functions and compare the integration time between the original car model equations and the new electric vehicle model equations. This function is enabled by selecting the “Run and Time” button in the MATLAB Editor. At the end of the simulation, a Profile Summary of the execution will be displayed in a new window. The summary shows the total time spent in executing each function of the program. By clicking on the desired function, the user can also see the execution time of each code line, and the most critical lines will be highlighted.

The integrator used for these simulations are rk4b and rk4bHE. The integration parameters are:

- Integration interval: 5 s
- Integration step: $1 \cdot 10^{-3}$
- All the programs are written in MATLAB code

The total integration time was 834.845 s for the original car model, and 829.073 s for the electric vehicle model.

Table 12.10 summarizes the total CPU time spent in the execution of the main car model original and modified functions. These data was generated in the Profile Summary of both simulations.

<table>
<thead>
<tr>
<th>Original Function</th>
<th>Total Time (s)</th>
<th>Modified Function</th>
<th>Total Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>car2.m</td>
<td>837.121</td>
<td>car2HE.m</td>
<td>835.409</td>
</tr>
<tr>
<td>mbs3d.m</td>
<td>836.039</td>
<td>mbs3dHE.m</td>
<td>834.490</td>
</tr>
<tr>
<td>rk4b.m</td>
<td>834.836</td>
<td>rk4bHE.m</td>
<td>829.057</td>
</tr>
<tr>
<td>RTDyn4m.m</td>
<td>496.827</td>
<td>RTDyn4mHE.m</td>
<td>494.838</td>
</tr>
<tr>
<td>wheelTorques.m</td>
<td>0.284</td>
<td>wheelTorquesHE.m</td>
<td>0.741</td>
</tr>
<tr>
<td>tireForcesHE.m</td>
<td>19.385</td>
<td>tireForcesHE.m</td>
<td>19.588</td>
</tr>
<tr>
<td>car2UserForces.m</td>
<td>20.791</td>
<td>car2UserForcesHE.m</td>
<td>20.984</td>
</tr>
<tr>
<td>evaluateExternalForces.m</td>
<td>24.432</td>
<td>evaluateExternalForcesHE.m</td>
<td>24.547</td>
</tr>
<tr>
<td>animateRec3.m</td>
<td>336.956</td>
<td>animateRec3HE.m</td>
<td>333.120</td>
</tr>
</tbody>
</table>

The most important differences are in \texttt{rk4b.m/rk4bHE.m} (the integrator function), \texttt{RTDyn4m.m/RTDyn4mHE.m} (which computes the derivatives), and \texttt{animateRec3.m/animateRec3HE.m} (the graphics function).
Results

However, as it can be seen, there is no significant difference between the total time spent in the execution of the functions in the original car model and the electric vehicle model.

Table 12.11 shows the execution time of the auxiliary functions programmed in the electric vehicle model, and the total additional integration time of these functions.

Table 12.11. CPU time distribution for the new auxiliary functions of the EV model

<table>
<thead>
<tr>
<th>New Function</th>
<th>Total Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>alfa.m</td>
<td>0.027</td>
</tr>
<tr>
<td>beta.m</td>
<td>0.366</td>
</tr>
<tr>
<td>Uoc.m</td>
<td>0.262</td>
</tr>
<tr>
<td>I.m</td>
<td>0.387</td>
</tr>
<tr>
<td>StateOfCharge.m</td>
<td>0.009</td>
</tr>
<tr>
<td>powerLossesC.m</td>
<td>0.034</td>
</tr>
<tr>
<td>Total additional time (s)</td>
<td><strong>1.085</strong></td>
</tr>
</tbody>
</table>

The results of these simulations show that the addition of the electrical differential equations does not affect the overall efficiency of the vehicle model in terms of total integration time.
13. Conclusions

The different tests and analysis of the results obtained throughout this project have led to a number of conclusions that will be explained below.

First of all, as a result of the new features introduced in MBS3D car model, the software is now able to support and perform the co-simulation of models from two different physical domains: electrical and mechanical. This has enabled a detailed study and comparison of the dynamic response of the differential equations to several numerical integrators such as rk4bHE and MATLAB ODE Suite integrators: ode45, ode23, ode113, ode15s, ode23s and ode23t. In general, when both subsystems are coupled, the complete vehicle model reacts more appropriately to those integrators specifically designed to solve stiff problems ode15s, ode23s and ode23t. This phenomenon is clearly seen in the integration speed and dynamic stability of the solution with the aforementioned integrators. It might be because of the need to use a small integration output step size, in order to adapt to the requirements of two submodels whose solutions evolve very differently over time. However, although the efficiency might be lower with integrators for non-stiff systems such as ode113, the model also shows reliable results in this case. This has been verified, for instance, in the driving cycle implementation.

The created electric vehicle model presents adequate performance characteristics that have been achieved through a careful design of the battery pack and the adjustment of the electric motor field and armature parameters. For this purpose, it has been necessary to perform numerous simulations which, in turn, required high programming time and multiple analyses of the graphics.

The introduction of the gas and brake pedal commands through the equations corresponding to the electric motor model, has allowed a more practical approach to driver behavior and to real vehicle performance. The driving cycle implementation has demonstrated that the system fulfills a basic requirement for vehicle models: the ability to perform a simulation in a real environment and under typical driving conditions.

Furthermore, the graphics displayed the influence of parameters such as the state of charge of the battery on powertrain performance characteristics, especially on peak power of the battery and the electric motor and maximum output torque of the electric motor. The application of field weakening control technique for managing the current of the electric motor field has also required many simulations in order to reach a well-designed strategy. The results show that FWC enhances the characteristics of the electric motor model and brings it closer to real motor performance. In addition, the state of charge estimation by “Coulomb counting” method has enabled to some extent the introduction of brake energy regeneration in the EV model. These data demonstrate an interesting approximation of the mathematical model to the real system.

Simulations also show that the efficiency of the electric vehicle model improves significantly with the translation of MATLAB code into C/C++ code. Although this process entailed more computational effort and added complexity to the project, the reduction achieved in simulation time was very valuable and marked a significant step forward in optimizing the simulation tool.

The available car model in MBS3D has proven to be consistent and flexible enough to be extended with differential equations that belong to the electrical domain. Although the real performance of electrical components is very different from that of the mechanical components, for example in terms of speed of response, the ODE electric vehicle model has no disadvantages compared to the conventional vehicle model. In order to determine the
Influence of the electric powertrain equations on the overall integration process, a simulation with the Profiler function has been run in MATLAB. The results of this simulation show that, in regard to integration time, the electrical differential equations do not add significant time to the integration process of the mechanical equations. Therefore, the efficiency of the multibody system is not compromised when the electrical subsystem is coupled.

In conclusion, a multidisciplinary vehicle model has been developed by applying co-simulation methodologies to two different sub-models: a mechanical multibody system dynamic model and an equation-based electrical model. These subsystems have been efficiently coupled in the MATLAB/Visual Studio environment and have given a favorable response to several numerical integrators. At the same time, a preliminary full electric vehicle model has been created, which is a reliable base for building other similar EV and even HEV computational models within MBS3D 2.0 software.
14. Further Research

Hybrid and electric vehicle technology offers a wide range of possibilities. During this project, many modeling options raised that could not be implemented because of time constraints. Furthermore, the main conclusions of the project show that there is still room for improvement in the electric vehicle mathematical model as well as in the application of co-simulation methodologies. Some of these options are now considered as opportunities for further research.

Regarding the simulation tool, MBS3D 2.0, it could be further extended to cover the study of hybrid electric powertrains. This implies the inclusion of the internal combustion engine model. Usually, dynamic ICE models are based on efficiency maps, where the operation point of the engine is established by introducing the gas pedal position. In addition, the vehicle controller model becomes more complex and requires the design of energy management techniques and control strategies. The control strategy could be implemented in MATLAB as a programming algorithm which will determine the performance of the electric motor drive and the engine, depending on parameters such as the engine temperature or the state of charge of the peaking power source. Table 14.1 shows an example of control algorithm, where the hybrid vehicle is operating in electric mode and the ICE operating point is established depending on the SOC of the battery and the engine temperature (Gaspart, 2013):

Table 14.1. Example of control strategy for hybrid electric vehicles

<table>
<thead>
<tr>
<th>SOC (%)</th>
<th>ΔSOC</th>
<th>ICE temperature (°C)</th>
<th>¿Engine ON?</th>
<th>ICE operating point</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOC ≥ 40</td>
<td>-</td>
<td>-</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>20 &lt; SOC ≤ 40</td>
<td>&lt; 0</td>
<td>-</td>
<td>NO</td>
<td>-</td>
</tr>
<tr>
<td>20 &lt; SOC ≤ 40</td>
<td>&gt; 0</td>
<td>T &lt; 90</td>
<td>YES</td>
<td>Cold-load: 2500 rpm, 500 Nm</td>
</tr>
<tr>
<td>20 &lt; SOC ≤ 40</td>
<td>&gt; 0</td>
<td>T &gt; 90</td>
<td>YES</td>
<td>Medium-load: 500 rpm, 80 Nm</td>
</tr>
<tr>
<td>SOC ≤ 20</td>
<td>-</td>
<td>T &lt; 90</td>
<td>YES</td>
<td>Cold-load: 2500 rpm, 500 Nm</td>
</tr>
<tr>
<td>SOC ≤ 20</td>
<td>-</td>
<td>T &lt; 90</td>
<td>YES</td>
<td>Quick-load: 2500 rpm, 100 Nm</td>
</tr>
</tbody>
</table>

The mathematical model of the battery that has been presented in this project is also worthy of further study. In particular, it is interesting to design longer driving cycles, where variations in the state of charge are significant. These variations affect the performance of the cell components and should therefore be taken into account when programming the functions corresponding to the parameters of the electrical equivalent circuit of the battery. Charge and discharge processes can be redesigned and studied through a more accurate mathematical description.

Another option is to replace the equations of the separately excited DC motor by others corresponding to different types of electric motors used for EV applications: AC induction
motor, brushless DC motor or switched reluctance motor. Obviously, the results would be completely different for each type of electric motor and it would also affect the battery model.

In relation to the co-simulation technique, weak-coupling method can be tested on the electric vehicle model by using two different integrators for the different subsystems. In this way, it is possible to adjust the integration rate to the needs of each sub-model and increase the efficiency of the simulation. Runge-Kutta integrator has proven successful in solving the multibody system equations. On the other hand, the results show that the electric system equations are integrated more efficiently with MATLAB integrators and, in some cases, especially with those designed to solve stiff problems. Therefore, both types of integrators could be applied simultaneously. This might result in shorter integration times and could simplify the process of adding systems of equations from more than two different domains. Using weak-coupling approach, a comparison between the results of the mechanical and the electrical equations can be done more easily, since both sub-models are integrated separately. Moreover, within weak-coupling there are three possible schemes of co-simulation integration process: direct, alternated and iterated (Figure 14.1). In the first scheme, values are exchanged once between the subsystems and then they are integrated independently. In alternated co-simulation, the integration and exchange of values of both subsystems takes place consecutively. In the iterated scheme both subsystems exchange information at the beginning of the integration and then they are integrated independently and simultaneously (Martin Arnold, Werner Schiehlen, 2008). The three options could be tested in the MATLAB environment.

Finally, regarding the powertrain model, other configurations could be considered. For instance, recently emerged technologies suggest the implementation of in-wheel electric motors. The adoption of this structure implies an independent control of each motor drive and, therefore, of each wheel. The control techniques for vehicles equipped with in-wheel motors are usually based on torque vectoring control (TVC). TVC strategies improve overall vehicle dynamic performance through the modulation of the output torque of the electric motor. Therefore, the use of TVC for the simulation of the multibody system together with the electrical model represents significant progress in the study of vehicle dynamics. The mathematical model of the in-wheel motor configuration would entail important changes in the design of MBS3D, since the software currently supports front-wheel drive, rear-wheel drive and four-wheel drive propulsion systems. Nevertheless, appropriate mathematical and programming tools are available and it is possible to introduce variations in the existing functions in order to adapt the created electric vehicle model to the new configuration.
15. References


Chan, C. C. (2002). The state of the art of electric and hybrid vehicles.


Marangoni, G. (2010, December 7). Battery Management System for Li-ion Batteries in Hybrid Electric Vehicles. Padova, Italy.

References


16. Time Planning and Budget

16.1. Time Planning

The time planning that has been followed while developing this final degree project is explained below. For this purpose, a Gantt chart has been built which includes the developed activities and the time required to complete them. The project started in June 2015 and was concluded in February 2016.

Table 16.1 summarizes the research, design and programming activities, the days spent on each task and how they are related to each other.

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. General bibliographic review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1. State of the art of EVs and HEVs</td>
<td>6 days</td>
<td></td>
</tr>
<tr>
<td>1.2. Study of current modeling techniques and software</td>
<td>11 days</td>
<td>1.1</td>
</tr>
<tr>
<td>1.3. Review of MATLAB and C/C++ programs and codes</td>
<td>4 days</td>
<td>1.3</td>
</tr>
<tr>
<td>1.4. Study of MBS3D simulation tool</td>
<td>7 days</td>
<td>1.3</td>
</tr>
<tr>
<td>2. Software installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1. Installation of MBS3D 2.0/2.0 Pro</td>
<td>7 days</td>
<td></td>
</tr>
<tr>
<td>2.2. Installation of Visual Studio</td>
<td>4 days</td>
<td></td>
</tr>
<tr>
<td>3. EV model definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1. Electric motor modeling</td>
<td>9 days</td>
<td>1.1, 1.2</td>
</tr>
<tr>
<td>3.2. Battery modeling</td>
<td>16 days</td>
<td>3.1</td>
</tr>
<tr>
<td>3.3. Power converter modeling</td>
<td>8 days</td>
<td>3.1, 3.2</td>
</tr>
<tr>
<td>3.4. Electric powertrain equation-based modeling</td>
<td>8 days</td>
<td>3.1, 3.2, 3.3</td>
</tr>
<tr>
<td>4. Programming</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1. Preliminary EV model in MATLAB</td>
<td>64 days</td>
<td>3.4</td>
</tr>
<tr>
<td>4.2. Coupling of MBS3D and EV models</td>
<td>17 days</td>
<td>4.1</td>
</tr>
<tr>
<td>4.3. Programming of graphics</td>
<td>2 days</td>
<td>4.2</td>
</tr>
<tr>
<td>4.4. Translation into C/C++ code</td>
<td>16 days</td>
<td>4.2</td>
</tr>
<tr>
<td>5. Model study and validation through simulations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1. Adjustment of the model parameters</td>
<td>6 days</td>
<td>4.3</td>
</tr>
<tr>
<td>5.2. Study and comparison of MATLAB ODE Suite integrators</td>
<td>11 days</td>
<td>4.4</td>
</tr>
<tr>
<td>5.3. Study of vehicle performance characteristics</td>
<td>5 days</td>
<td>4.4, 5.1</td>
</tr>
<tr>
<td>5.4. Driving cycle implementation</td>
<td>20 days</td>
<td>4.4, 5.1</td>
</tr>
<tr>
<td>5.5. MATLAB profiling</td>
<td>6 days</td>
<td>4.4, 5.1</td>
</tr>
<tr>
<td>6. Project report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.1. Theoretical framework</td>
<td>140 days</td>
<td>1.2, 1.4</td>
</tr>
<tr>
<td>6.2. Introduction</td>
<td>10 days</td>
<td>1.2</td>
</tr>
<tr>
<td>6.3. Methodology</td>
<td>6 days</td>
<td>3.4, 4.2, 6.2</td>
</tr>
<tr>
<td>6.4. Results</td>
<td>19 days</td>
<td>5.2, 5.3, 5.4, 5.5</td>
</tr>
<tr>
<td>6.5. Final review</td>
<td>5 days</td>
<td>6.1, 6.2, 6.3, 6.4</td>
</tr>
</tbody>
</table>

In addition, the Gantt chart (Figure 16.1) is enclosed and the specific execution dates for each task are also reflected in Table 16.2.
Figure 16.1. Gantt chart of the project
16.2. Budget

In this section, the analysis of the costs associated to the implementation of the present project will be carried out. For the budget estimation, the working hours, the staff and the equipment used will be taken into account.

**Table 16.3. Budget estimation for the final degree project implementation**

<table>
<thead>
<tr>
<th>Payment Description</th>
<th>Units</th>
<th>Unit Price</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours spent on the project development</td>
<td>360 hours</td>
<td>20 €</td>
<td>7,200 €</td>
</tr>
<tr>
<td>Computers</td>
<td>2</td>
<td>780 €</td>
<td>1,560 €</td>
</tr>
<tr>
<td>Depreciation of computer equipment</td>
<td>-</td>
<td>-</td>
<td>400 €</td>
</tr>
<tr>
<td>Technical computer staff</td>
<td>-</td>
<td>-</td>
<td>900 €</td>
</tr>
<tr>
<td><strong>Subtotal</strong></td>
<td></td>
<td></td>
<td>10,060 €</td>
</tr>
<tr>
<td><strong>Indirect Costs (12%)</strong></td>
<td></td>
<td></td>
<td>1,207.20 €</td>
</tr>
<tr>
<td><strong>Total Cost</strong></td>
<td></td>
<td></td>
<td>11,267.2 €</td>
</tr>
</tbody>
</table>
The purchase of two computers has been included, as well as the depreciation of this equipment. For this calculation, an annual depreciation of 26% of the computer equipment has been considered, which results in 19.5% over the price of each computer, taking into account that the project has been developed during 9 months. The labor has a cost of 20 € per hour and the cost of technical staff has been estimated in 900 €.

After adding the different items of the budget represented in Table 16.3, a total amount of 11,267.2 € is obtained.
17. List of Figures

Figure 0.1. Connection and exchange of values between MBS3D and the electrical subsystem ................................................................. 2
Figure 1.1. Technological trends of electric and hybrid vehicles ................................................................. 6
Figure 3.1. Electric vehicle power train configuration .................................................................................. 11
Figure 4.1. Forces affecting a vehicle ............................................................................................................... 17
Figure 4.2. Ideal engine power-torque curve, typical of EV motors ............................................................ 18
Figure 4.3. Power-torque curve for a conventional IC engine ................................................................. 18
Figure 4.4. Ttractive effort of a conventional IC engine ................................................................................... 19
Figure 4.5. Typical model for the evaluation of traction force capacity ......................................................... 20
Figure 4.6. Model for the calculation of vehicle frontal area ............................................................................. 21
Figure 4.7. a) Soft wheel rolling on a hard surface. The surface deforms the wheel. b) Ideal rolling resistance condition .................................................................................................................... 22
Figure 4.8. Rolling resistance coefficient for various materials and different inflation pressure ...................... 22
Figure 5.1. Possible configurations of EVs ....................................................................................................... 25
Figure 5.2. Series hybrid electric drivetrain .................................................................................................... 26
Figure 5.3. Parallel hybrid electric drivetrain .................................................................................................. 27
Figure 5.4. Series-parallel hybrid drivetrain .................................................................................................... 28
Figure 5.5. Distribution of motive power portions in HEVs .............................................................................. 29
Figure 5.6. Maximum state of charge of PPS control strategy ........................................................................... 31
Figure 5.7. Thermostat or engine on-off control strategy .................................................................................. 31
Figure 5.8. Fuzzy logic control rule base ........................................................................................................... 32
Figure 5.9. Series RBS performance ............................................................................................................... 34
Figure 5.10. Parallel RBS performance ........................................................................................................... 35
Figure 6.1. Ideal motor force-speed profile of an electric motor ......................................................................... 37
Figure 6.2. Brushless DC motor ..................................................................................................................... 38
Figure 6.3. Electric motor technologies for EVs-HEVs .................................................................................... 40
Figure 6.4. Main parts of a spark-ignition internal combustion engine ......................................................... 41
Figure 6.5. Basic 4S cycle ............................................................................................................................... 42
Figure 6.6. Normal operation of a fuel cell ...................................................................................................... 45
Figure 7.1. Typical battery cell ....................................................................................................................... 50
Figure 7.2. Basic configuration of a capacitor .................................................................................................. 54
Figure 7.3. Connection with generator in flywheel .......................................................................................... 55
Figure 8.1. Basic series hybrid vehicle configuration ....................................................................................... 59
Figure 8.2. Power flow variables for a series hybrid drivetrain ......................................................................... 59
Figure 8.3. Modular representation of MBS3D software .................................................................................. 60
List of Figures

Figure 8.4. Flow of power factors for an electric car drivetrain ........................................ 60
Figure 8.5. Motor/generator modular representation for dynamic modeling ........................ 61
Figure 8.6. Battery modular representation for dynamic modeling ................................. 61
Figure 8.7. Electric power link modular representation for dynamic modeling .................. 61
Figure 8.8. ICE modular representation for dynamic modeling ........................................ 62
Figure 9.1. Drivetrain configuration of the electric vehicle model ...................................... 63
Figure 9.2. Dynamic circuit of a separately excited DC motor ........................................ 64
Figure 9.3. Randles circuit for dynamic modeling of batteries ......................................... 66
Figure 9.4. Open circuit voltage as a function of the state of charge .............................. 67
Figure 9.5. Basic chopper converter ................................................................................. 68
Figure 10.1. MATLAB working environment ..................................................................... 72
Figure 10.2. Microsoft Visual Studio working environment ............................................ 73
Figure 11.1. Coupling of models (levels) ......................................................................... 75
Figure 11.2. Tight Coupling ............................................................................................ 76
Figure 11.3. Weak Coupling ............................................................................................ 76
Figure 11.4. Ohmic internal resistance of a Lithium-ion cell ........................................... 82
Figure 11.5. Charge-transfer and diffusion resistance of a Lithium-ion cell ...................... 82
Figure 11.6. Double layer capacitance of a Lithium-ion cell .......................................... 83
Figure 11.7. Battery pack configuration ........................................................................... 83
Figure 11.8. Approach of MBS3D function-call process, main input and output arguments and tasks of the main functions ................................................................................. 91
Figure 12.1. Fluctuations in the field current during the integration with ode23s .............. 97
Figure 12.2. Improved dynamic response of \( I_f \) with ode23s when reducing the integration output step ............................................................................................................................. 97
Figure 12.3. Front wheels speed plot with ode113 integrator. All the integrators show similar oscillatory behavior in the same time interval ......................................................... 98
Figure 12.4. Unstable behavior of the front wheels speed during the simulation with rk4bHE .................................................................................................................................................. 99
Figure 12.5. The dynamic response with rk4bHE becomes stable when the integration step is reduced .................................................................................................................................. 99
Figure 12.6. Multibody system position and velocity at the end of the simulations .......... 103
Figure 12.7. Motor torque and power outputs with \( SO\) = 1 ............................................ 104
Figure 12.8. Motor torque and power outputs with \( SO\) = 0.7 ........................................ 104
Figure 12.9. Motor torque and power outputs with \( SO\) = 0.4 .......................................... 104
Figure 12.10. Electric motor armature and field currents with \( SO\) = 1 .......................... 105
Figure 12.11. Electric motor armature and field currents with \( SO\) = 0.7 ....................... 105
Figure 12.12. Electric motor armature and field currents with \( SO\) = 0.4 ..................... 105
Figure 12.13. Battery current and voltage with \( SO\) = 1 ............................................. 106
Figure 12.14. Battery current and voltage with \( SO\) = 0.7 ........................................... 106
Figure 12.15. Battery current and voltage with $SOC_0 = 0.4$ .................................................106
Figure 12.16. Battery power output with $SOC_0 = 1$ .................................................................107
Figure 12.17. Battery power output with $SOC_0 = 0.7$ .............................................................107
Figure 12.18. Battery power output with $SOC_0 = 0.4$ .............................................................107
Figure 12.19. State of charge evolution during simulations performed with different values of $SOC_0$ .................................................................108
Figure 12.20. Final multibody system position and velocity ........................................................110
Figure 12.21. Front wheels speed during the driving cycle ..........................................................110
Figure 12.22. Motor torque and power requirements during the driving cycle .........................111
Figure 12.23. Gas/brake pedal power command during the driving cycle .................................111
Figure 12.24. Evolution of the motor armature and field currents during the driving cycle...111
Figure 12.25. Field weakening control command $\beta(t)$ .............................................................111
Figure 12.26. Battery power output during the driving cycle .......................................................112
Figure 12.27. Variation in the battery state of charge during the driving cycle .........................112
Figure 12.28. Battery current and voltage requirements during the driving cycle ....................112
Figure 12.29. Evolution of $U_0$ during the driving cycle ............................................................112
Figure 14.1. Direct, alternated and iterated weak-coupling schemes .........................................118
Figure 16.1. Gantt chart of the project .........................................................................................122
18. List of Tables

Table 4.1. Typical values of the coefficient of aerodynamic resistance for different types of vehicle ................................................................. 21
Table 6.1. Averaged values of equivalent mass factor ............................................. 43
Table 7.1. Battery characteristics for different types of HEV (Pistoia, Gianfranco, 2010) ...... 53
Table 7.2. Different BMS architectures and functions according to vehicle category (Pistoia, Gianfranco, 2010) .......................................................... 56
Table 9.1. Vehicle parameters ............................................................................. 63
Table 9.2. DC motor parameters and specifications ................................................. 65
Table 9.3. Lithium-ion cell characteristics (T=40°C) ............................................. 67
Table 11.1. Battery pack specifications .................................................................. 84
Table 11.2. Auxiliary functions programmed for completing the electrical vehicle model ..... 88
Table 12.1. Different integration times using MATLAB or C/C++ files .................. 95
Table 12.2. Integration time and stability of the response with different integrators under the conditions specified in a) ............................................................. 96
Table 12.3. Integration time and stability of the response with different integrators under the conditions specified in b) ............................................................. 97
Table 12.4. Integration time and stability of the response with different integrators under the conditions specified in c) ............................................................. 98
Table 12.5. Integration time and stability of the response with different integrators under the conditions specified in d) ............................................................. 99
Table 12.6. Integration time and stability of the response with different integrators under the conditions specified in e) ............................................................. 100
Table 12.7. Integration time and stability of the response with different integrators under the conditions specified in f) ............................................................. 100
Table 12.8. Comparison of vehicle performance characteristics with different values of the battery state of charge ............................................................... 101
Table 12.9. Approximate battery maximum currents and voltages with different values of \( SOC_0 \) ........................................................................................................ 106
Table 12.10. Comparison of the MATLAB Profile Summary of the original and modified functions in MBS3D car model ......................................................... 113
Table 12.11. CPU time distribution for the new auxiliary functions of the EV model .... 114
Table 14.1. Example of control strategy for hybrid electric vehicles ....................... 117
Table 16.1. Task description, duration and predecessors ......................................... 121
Table 16.2. Project task list and date of implementation .......................................... 123
Table 16.3. Budget estimation for the final degree project implementation ............... 123
19. List of Abbreviations

MBS  Multi-body System
ODE  Ordinary Differential Equation
DAE  Differential-algebraic Equation
EV   Electric Vehicle
HEV  Hybrid Electric Vehicle
SOC  State of Charge
FWC  Field Weakening Control