Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Energetics (XXXIV Cycle)

Hybrid and Electric Vehicles Optimal Design and Real-time Control based on Artificial Intelligence

Claudio Maino

******

Supervisor(s):
Prof. Daniela Anna Misul
Prof. Ezio Spessa
Prof. Nicola Amati

Doctoral Examination Committee:
Prof. Aldo Sorniotti, University of Surrey, Great Britain
Prof. Giansalvo Cirrincione, Université de Picardie Jules Verne, France
Prof. Giorgio Rizzoni, Ohio State University, OH, USA
Prof. Nicolò Cavina, Università degli Studi di Bologna, Italy
Prof. Zoran Filipi, Clemson University, SC, USA

Politecnico di Torino
2022

Declaration

I hereby declare that the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Claudio Maino

Torino, July 27, 2022

To those who fight everyday

Summary

Electrification of on-road vehicles is one of the most recognized technology transformations worldwide and is nowadays pushing the world towards new mobility scenarios. New national and international regulations have born which introduce restrictions for conventional vehicles and set goals for green mobility. The infrastructure is evolving to welcome charging devices, zero-emissions are required to drive within specific city centers, stops are occasionally imposed to "old" vehicles in case of high pollution periods. Many resources are being worldwide invested into the research and development of innovative technologies for electrified vehicles, similar to what happened for fuel-based vehicles.

An outstanding progress has occurred also to computer systems during the last years. Specifically, the improvement of the computational power achieved by newest processors has promoted the possibility of establishing new connections between engineering and informatics. Amongst all, techniques based on Artificial Intelligence (AI) have seen a spread in their application to very different engineering fields, such as robotics, computer vision, semantics, etc.

In the present dissertation, a journey into the optimization of the design and the real-time control of electrified vehicles through computer-assisted simulation is made which aims at demonstrating some of the potentials enclosed in AI-based approaches. Different techniques have been analyzed and their performances have been thoroughly discussed to provide the reader with a clear view of the integration of AI into the research about electrified vehicle technologies. Hybrid configurations have been taken into account among the variety of electrified powertrains to be possibly analyzed (hybrid, fuel cell, full electric). The availability of different and multiple power sources within the same driveline introduces complexity both when the optimal design and the optimal real-time control are investigated. A demanding test case has hence been selected to stress AI systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem overview

Global warming is nothing new. Especially, the contribution of carbon dioxide (CO_2) to the greenhouse effect is nothing new. One of the first scientific papers that addressed the problem dates back to 1896, in which Arrhenius ([1]) introduced a model for the estimation of the effects produced by the presence of heat-absorbing gases in the atmosphere. Following his first article, in [2] Arrhenius suggested the possibility of global warming to show up sooner than it was thought. In 1938, Callendar ([3]) claimed that doubling the CO_2 level could have turned into an Earth's warm of roughly 2°C. Forty years later, Keeling et al. ([4]) published the results of a statistical analysis carried over atmospheric CO2 concentrations of the air near Hawaiian Islands and proved that the average seasonal oscillation could

---

Part of the present section has been extracted from:

Misul, D. A. et al. (2020). Development and assessment of a random-forest model for real-time prediction of diesel engine-out particulate matter emissions. International Journal of Mechanics and Control. Levrotto & Bella Co., 21(02), pp. 95–110.

Maino, C. *et al.* (2021). A deep neural network based model for the prediction of hybrid electric vehicles carbon dioxide emissions. *Energy and AI*. Elsevier Ltd, 5, p. 100073. Doi: 10.1016/j.egyai.2021.100073.

Maino, C. *et al.* (2022). Project and Development of a Reinforcement Learning Based Control Algorithm for Hybrid Electric Vehicles. *Applied Sciences*, 12, p. 812. Doi: https://doi.org/10.3390/app12020812.

have been described by an increasing function.

In 1988, Hansen et al. ([5]) presented the results of a model forecasting an almost immediate increase of the global mean surface air temperature and demonstrated the necessity of introducing urgent measurements to mitigate the climate change process and deal with it. From that moment onwards, more and more attention has been paid to the problem. The International Panel on Climate Change (IPCC) has been established in December 1988 ([6]), many international meetings have been held and two important documents have been subscribed by multiple countries in the world: the Kyoto Protocol in 1997 ([7]) and the Paris Agreement in 2015 ([8]). Nevertheless, today's world condition is still of suffering. In Figure 1, the Earth's ground temperature variation measured from 1896 to 2020 is reported ([9]). A worryingly increasing trend can be appreciated along with the absence of a curve slope modification in correspondence of the international agreements period. According to data presented in [10]-[13], the CO_2 emissions level has risen from 315 parts per million (ppm) to 400 ppm between 1958 and 2015. At the actual global emission rate, projections show the level could exceed 900 ppm. As demonstrated by Parmesan et al. in [14], such a scenario involves the possibility of irreversible alterations to the Earth's natural systems.

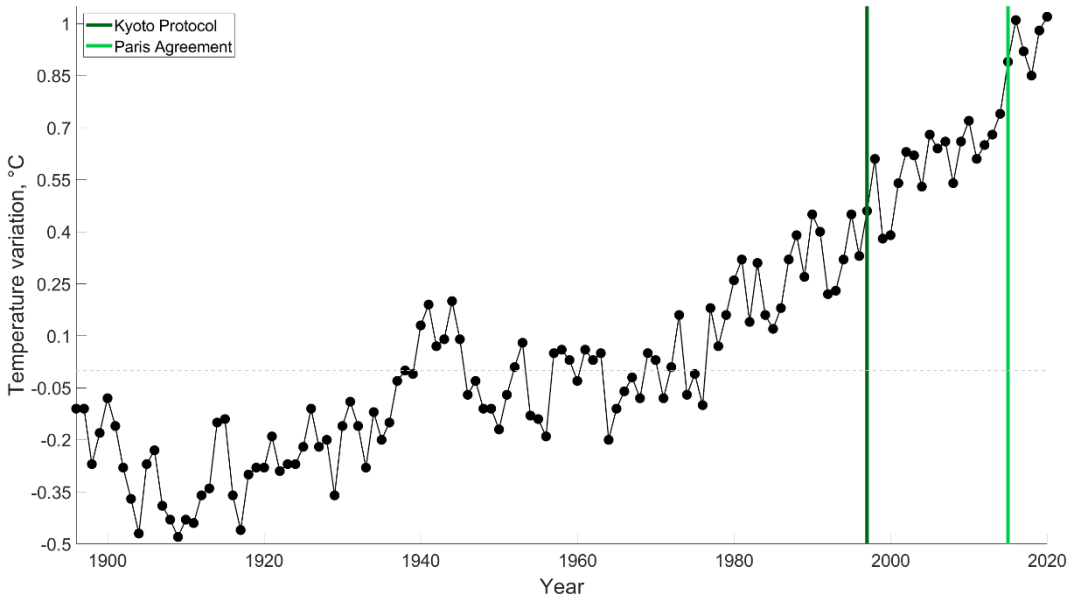

Figure 1. Global warming from 1896 to 2020.

According to the IPCC report of 2018 ([15]), the rise of temperatures should not overcome 1.5°C. Flattening the curve of temperature variation has hence become a priority all over the world. It is therefore evident that a careful planning of short-term and long-term actions is required to the majority of the countries.

The transportation sector holds a key role within the context of CO_2 production. In 2019, 24% of European CO_2 emissions have been produced by the transportation sector ([16]), while roughly 29% in U.S. ([17]). Still, on-road vehicles (including passenger cars, light-duty vehicles, heady-duty vehicles, two and three wheelers and buses) have contributed for roughly 77% of European CO_2 emissions ([18]), while for 82% in U.S. ([19]). In 2020, the shares of conventional (petrol or gasoline based) passenger cars (*pc*), vans and heavy-duty vehicles (*hdv*) sold in Europe with respect to the total sales of the relative vehicular segments have reached 75.5%, 95.8% and 96.6%, respectively ([20]-[22]). A similar situation has occurred in U.S. and China, as the share of new non-conventional vehicles sold has nearly achieved 10% ([23]).

The World harmonized Light-duty Test Procedure (WLTP) ([24]) and the EPA Federal Test Procedure (FTP) ([25]) are the actual test procedures for Europe and U.S., respectively. However, new regulations have begun their course promoting a shift towards "greener" solutions for the on-road mobility and a consequent reduction of the emissions related to fuel combustion. "Fit for 55 Package" ([26]), new limits of passenger cars CO_2 emissions ([27]) and proposals for Euro 7 standards ([28]) represent some important examples for Europe; similarly, an update of the regulatory framework for light-duty vehicles (*ldv*) will impact the U.S. market in the next few years ([29]). Consistently, electrification of vehicles has accelerated and it clearly represents one of the most promising approaches identified by the scientific community to face the problem of on-road emissions. Electrifying vehicles can assume different meanings. The replacement of the conventional power source, i.e. the internal combustion engine (ICE), or the integration with an electric assist are two different solutions. In other words, full electric vehicles and hybrid electric vehicles (HEV) are nowadays considered as the two major solutions. In Figure 2, the typical Usage of Electricity (UoE) is reported for different electrified powertrains ([30]-[32]), which is representative of the common share of electricity consumed in an electrified vehicle with respect to the fuel. Black lines are used to highlight the minimum and maximum UoE boundaries for each technology. The green area can hence be considered as the electric operational region of the vehicles. Conventional vehicles (CVs) are the common liquid fuel-based vehicles and do not imply the usage of electricity. Micro hybrids

(mHEVs), mild hybrids (MHEVs) and full hybrids (FHEVs) represent the ICE-based hybrid architectures with a progressively increasing electric contribution (from 5-10% to 40-50%) without the possibility of being recharged from an external source. On the other hand, plug-in hybrids (PHEVs) are the ICE-based hybrid architectures with the possibility of being recharged from the grid; larger battery packs and oversized motor-generators are typically involved in these architectures and wider electric ranges can be exploited, further reducing the fuel-related emissions. Finally, battery electric vehicles (BEVs) and fuel cells electric vehicles (FCEVs) represent the on-road emissions-free technologies since any polluting component is removed from the driveline and the whole driving experience takes advantage of electricity. For a deeper dissertation about (P)HEVs (i.e. hybrid and plug-in hybrid systems) and electric vehicles, please refer to [33] and [34], respectively.

Given the differences between the electrified vehicle technologies, different are the related advantages and drawbacks. Full electric powertrains are characterized by a simpler design as the conventional energy source (and any other related component) is completely neglected. Consequently, a green driving experience can be accomplished with a complete abatement of on-road emissions. In turn, the costs related to the electric components, the necessity of an efficient recharging infrastructure as well as the limited driving ranges are important barriers to be overcome ([35]). In fact, several factors, such as the environmental temperature, the efficiency of the battery thermal management system (BTMS) or the driving conditions should be taken into consideration for a quantification of the real benefits brought by a complete vehicle electrification ([36]). On the other hand, (P)HEVs are featured by the combination of multiple energy sources and thus are capable of reducing range anxiety and recharging issues ([37]). Furthermore, users' requirement about vehicle drivability and performance (e.g. maximum acceleration) can be met displacing undersized electric components with respect to BEVs while adding an ICE ([38]). Obviously, the presence of the thermal propulsion system constrains the possibility of minimizing the on-road emissions. To this end, two demanding operations have to be carried out. First, a design optimization aimed at defining the hybrid optimal layouts (i.e. configurations of the electric components in terms of number and location in the driveline) for different driving scenarios. Second, a complex control problem aimed at identifying a smart usage of the on-board power components with the capability of maximizing the system efficiency.

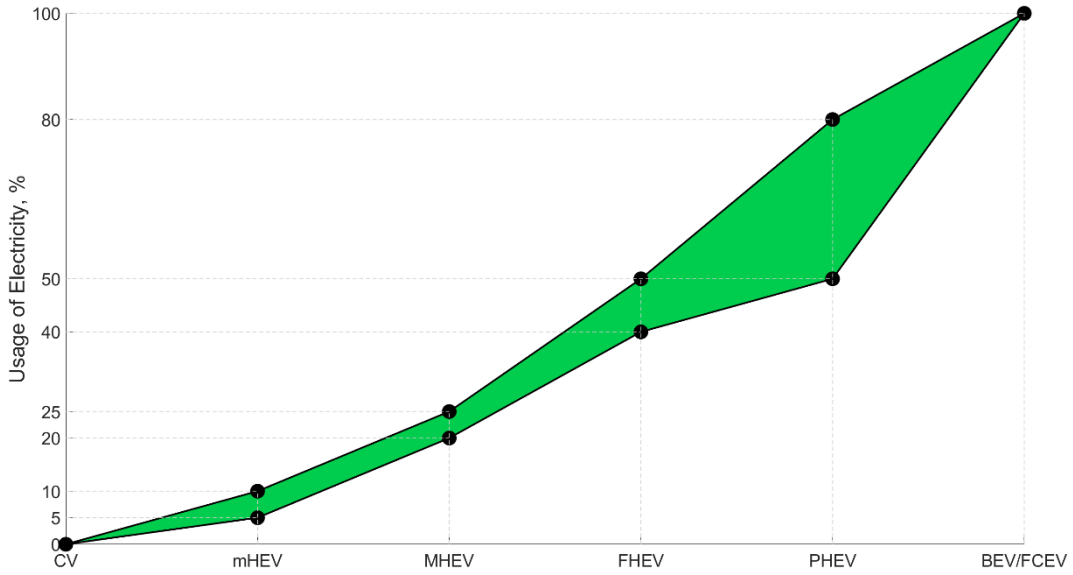

Figure 2. Typical usage of electricity for different electrified vehicles.

Supporting the shift towards new powertrains by means of innovative methodologies with the capability of solving the problems introduced by electrification is mandatory. Exploiting the maximum performances of both hybrid and electric vehicles can be obtained only by defining the optimal design (mainly for (P)HEVs) and the optimal control (BEVs, (P)HEVs and FCEVs).

Within this framework, several research works have been presented in the literature focused on solving the problem with various approaches. Nevertheless, few steps have been taken in the direction of developing and applying intelligent systems for design and real-time control of electrified vehicles. As experienced in very different research fields, such as Internet of Things (IoT) or medicine, the progress in processors and numerical computation can play a key role in the evolution of the tools and the maximization of their accuracy. In fact, even if intelligent approaches had been introduced already in 1950s ([39]-[41]), the computational power needed to solve complex problems was unreachable at that time. Thanks to the research in computer systems and to their actual performances, learning-based approaches have nowadays become a widely used support for the development of breakthrough methodologies. The latter are typically classified with the famous signature of Artificial Intelligence (AI). Among the possible definitions given to AI, the following is considered for this discussion ([42]):

"Artificial intelligence (AI) systems are software (and possibly also hardware) systems designed by humans that, given a complex goal, act in the physical or digital dimension by perceiving their environment through data acquisition, interpreting the collected structured or unstructured data, reasoning on the knowledge, or processing the information, derived from this data and deciding the best action(s) to take to achieve the given goal. AI systems can either use symbolic rules or learn a numeric model, and they can also adapt their behaviour by analysing how the environment is affected by their previous actions."

According to the present definition, the usage of AI techniques can be extended to any engineering problem featured by complex optimizations and/or predictions in presence of tons of data. The algorithmic structures are generally referred to as Machine Learning (ML), recalling the algorithms capability of operating through a learning process. Three macro-categories are typically defined for ML algorithms: supervised learning (SL), unsupervised learning (UL) and reinforcement learning (RL). A SL algorithm is trained on labeled data (i.e. data with a known class or value) so as that predictions about future and unknown data could be generated ([43], [44]). An UL algorithm is instead employed when unlabeled data (i.e. data without a known class or value) are available and is generally employed to catch patterns underneath the data as well as intercepting particular trends ([45]). Finally, a RL algorithm is characterized by a learning process based on numerical rewards and/or punishments aimed at maximizing the numerical reward value ([46]-[47]). A pre-determined link between ML categories and engineering applications cannot be identified; indeed, a preliminary accurate analysis of the task to be solved is needed.

In the present dissertation, two of most debated aspects in the literature about electrified vehicles have been considered as test cases: optimization of design and optimization of real-time control for HEVs. Showing the potentials of intelligent systems and allowing the research on vehicular technologies for taking a step forward in the utilization of AI-based techniques have been set as main goals of this work.

1.2 Optimal design and real-time control of hybrid and electric vehicles: a literature review

In the last years, attention has been paid in the literature about the topics of optimal design and optimal real-time control of electrified vehicles. The most complex operations of design and energy management occur when (P)HEVs are analyzed since a wide set of architectures and layouts can be realized (with respect to full electric vehicles) and their relative performances can strongly differ according to the change in the driving conditions (e.g. driving style, driving mission, traffic, etc.). Therefore, the methodologies with the capability of solving design and control problems of (P)HEVs can be exploited also when simpler electrified powertrains have to be focused. In this section, the main outcomes of past research activities about design and energy management optimizations are going to be discussed separately based on the findings about the most complex electrified architectures, i.e. (P)HEVs. For the sake of clarity, "hybrid architecture" refers to the configuration of the traction system, such as parallel, series, power-split or complex series/parallel ([48]). On the other hand, "hybrid layout" refers to the number, the location and the sizing of the electric components embedded in a given architecture.

A typical (P)HEV optimal design problem can be represented through the scheme illustrated in Figure 3. A set of design variables (e.g. ICE sizing, battery sizing, position of the electric motors, gear ratios, etc.) has to be defined for a given hybrid architecture. Each set of design variables is representative of the specific hybrid layout to be analyzed. Then, the driving conditions have to be identified, including the vehicle velocity trajectory (possibly including information about average traffic conditions) and the road slope. Still, an optimization target has to be defined according to the specific design objective, such as minimizing the fuel consumption and/or the pollutant emissions rather than maximizing the electric range. Finally, an energy management optimizer (or simply "optimizer") has to be adopted for solving the control problem related to the optimal management of the main powertrain components considered for a specific configuration (i.e. hybrid architecture and layout, driving conditions and optimization target) from an energetic perspective. The target of the optimizer corresponds to the design-related optimization target. In fact, given the need of defining a method for comparing different powertrains, the problem of a (P)HEV optimal design has to be necessarily linked to the relative problem of optimal control.

Within the framework of (P)HEV optimal control, several approaches have been developed and their relative performances have been thoroughly analyzed in

the literature ([49], [50]). A first distinction can be made between real-time capable energy management systems (EMSs) and global optimizers (GOs). The first are typically featured by a sub-optimal fast-running algorithmic structure, whereas the latter comprise computationally heavier procedures aimed at evaluating the optimal solution to a control problem ([51]). Thanks to the knowledge of the entire driving mission, GOs have proved to outperform other approaches (such as heuristics, rule-based or static optimizers) in the evaluation of the maximum performances of a (P)HEV. The assessment of the maximum performances obtained by a specific hybrid powertrain is a fundamental milestone to be achieved for comparing different configurations. Therefore, the definition of a proper control strategy for (P)HEVs assumes a relevant role among the four macro-inputs of Figure 3. Since the optimization of the real-time control is performed only after the identification of the vehicle design, GOs are generally preferred for design optimization operations.

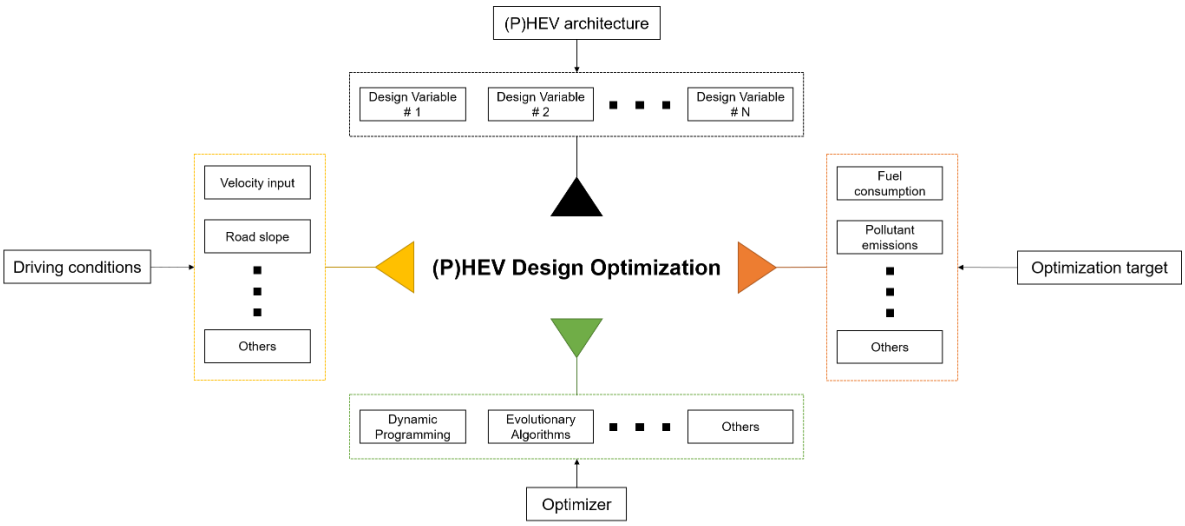

Figure 3. General framework of hybrid electric vehicles design optimization.

A typical distinction about GOs is made between dynamic optimization and static optimization methods ([52]). Dynamic optimization methods include methods based on optimal control theory and are capable of obtaining the optimal solution to a control problem. One of the most used and popular GOs is represented by Dynamic Programming (DP) ([53]). The standard DP formulation is characterized by a multi-stage decisional process aimed at providing the solution to

the Bellman optimality equations ([54]) in a time-discretized environment. A large number of DP applications to (P)HEV control has been explored in the literature. Sundström and Guzzella ([55]) presented a famous first DP implementation in 2009, which paved the way for several other studies during the following years. In [56], the design of a complex parallel HEV equipped with two motor-generators (MGs) has been optimized through DP to minimize the fuel consumption (i.e. on-road CO_2 emissions) and the pollutant emissions of the vehicle over several driving missions. In [57], the same operation has been performed for a parallel through-the-road architecture. Few years later, Finesso *et al.* ([58]) have refined the optimization methodology to design a dual-mode HEV architecture with the capability of minimizing fuel consumption and pollutant emissions while considering battery ageing effects. In this case, an enhanced formulation of the DP objective functional has been taken into account. Other approaches have been exploited through the years. Hybrid formulations of DP have also been exploited in the last years. As example, Nüesch ([59]) and Gissing ([60]) have shown the potentials of coupling DP with Convex Optimization (CO) for controlling a parallel HEV and with heuristics for controlling a series HEV, respectively. A different dynamic optimization method has been presented in [61], in which the Pontryagin's Minimum Principle (PMP) has been used to solve the problem of minimizing the fuel consumption of a parallel HEV while cutting the computational time needed to identify the solution.

On the other hand, static optimization methods include derivative free and gradient-based methods ([52]). Unlike dynamic optimization, both derivative free and gradient-based methods are typically capable of identifying sub-optimal solutions of a control problem with smaller computational effort. In [62] and [63], the optimal design of a parallel HEV has been determined after the identification of the optimal control trajectory through two different derivative free algorithms, specifically Multi-Objective Genetic Algorithm (MOGA) and Particle Swarm Optimization (PSO). The MOGA has been employed to maximize the fuel economy (and hence the CO_2 emissions) while minimizing pollutant emissions, whereas the PSO has focused on the minimization of the production costs along with the fuel economy maximization. As far as gradient-based methods are concerned, a sequential quadratic programming (SPQ) has been presented in [64], in which the optimization of the offline control of a parallel HEV has been investigated. Few others methodologies have shown the capability of identifying an offline optimal solution to a (P)HEV control problem without being explicitly oriented to offline optimal control. Specifically, the Equivalent Consumption Minimization Strategy

(ECMS) and its adaptive formulation, A-ECMS, have been presented in [65], [66] and [67], and represent two of the most promising EMS for HEVs in terms of local optimization of the hybrid systems operations. Both ECMS and A-ECMS are inspired by PMP optimization logics and are capable of locally optimizing the performance of a (P)HEV by means of a proper tuning of equivalence factors (ECMS) as well as of adaptation rules (A-ECMS). Thanks to an accurate tuning of the main parameters based on the driving mission, the ECMS-related control strategies proved to be capable of producing comparison results with respect to DP ([67]).

For the sake of clarity, similar considerations can be made about design optimizations of BEVs or FCEVs. Certainly, three of the four macro-inputs (vehicle architecture, design variables and optimization target) would differ in the design optimization framework consistent with the change in the electrified powertrain; nevertheless, no modifications are introduced in the general approach. Some examples have been presented in [68]-[71].

A relevant research effort has hence been given to the development and application of different optimizers with the capability of assisting in the identification of the optimal design of one or few electrified powertrains. Regardless the electrification level of the powertrains, design optimization operations might also be intended to identify solutions within huge design spaces. Unfortunately, seeking for optimal solutions to a design problem by means of GOs has to be paid through time ([72]). In fact, independently from the specific combination of design variables, driving conditions and optimization target, the selection of a GO, such as DP, introduces a relevant increase in the amount of calculations required to solve the control problem of a single architecture ([73]). Therefore, the possibility of identifying the optimal solution(s) within huge spaces can hardly be pursued. In addition, if the same operation were intended for multiple electrified powertrains of a vehicles' fleet, it can turn into an impossible process. Since an a priori knowledge of the driving environment is available, a shift towards non-optimal optimizers would not represent a smart solution as a sub-optimal solution would be guaranteed. As far as learning algorithms are considered, a lack in the literature exists about methodologies with the capability of predicting the results of a specific optimizer for new and unknown testing conditions after a proper training on previously simulated data. Recalling the definition of AI stressed in the Introduction, supervised AI-based methodologies could perfectly fit the problem and can be hence considered as potentially promising tools.

Once the optimal design of an electrified vehicle has been identified (or a given configuration has been selected), the problem of controlling the vehicle under real-time conditions arises. The complexity of identifying real-time capable, light, adaptive and accurate control methodologies explodes in case of (P)HEVs, for which the variability of real-world driving conditions becomes a significant obstacle to the definition of a real-time optimal management of the main power and energy components. Concerning real-time control of hybrid electric powertrains (for simplicity "hybrid powertrains"), an extensive amount of researches can be found in the literature ([74]-[77]). The simplest non-optimal approaches date back to the 1990s, when rule-based controllers (RBC) have been developed starting from dynamometer test and road test data ([74]). Indeed, the first optimum-oriented real-time controllers were presented in the 2000s, specifically RBCs with rules based on the optimal control policies defined by GOs, such as DP ([78]) or genetic algorithm (GA) ([79]). Since an offline tuning of the main parameters is necessary for a proper selection of the control rules, a significant limitation in terms of optimization accuracy and adaptability highlights when the real driving conditions differ from the calibration ones. From that moment onwards, the potentials of several other methodologies have been exploited. Examples of the literature about ECMS-oriented real-time controllers have been presented in ([80]-[83]). In [80], a merge between RBC and ECMS approaches has been performed to tackle the limitations of RBC and ECMS as stand-alone techniques; in [81] and [82], information obtained through offline simulations with DP have been used to calibrate the ECMS main parameters; in [83], a similar operation has been carried out considering GA instead of DP. Feeding real-time controllers with additional information about the driving environment has also proved to be a promising solution. Around 2010, the potentials of real-time control strategies based on model predictive control (MPC) have been presented ([84], [85]). In MPC, an optimizer can be employed to solve a simpler control problem over a future time horizon thanks to the prediction of a user-defined set of variables of the actual driving mission. Since the control policies of MPC-based EMSs strongly rely upon the accuracy of future driving conditions accuracy, the identification of the optimal solution to (P)HEVs real-time control is not guaranteed.

Within the framework of (P)HEV optimal real-time control, the fit of learning-based controllers has been exploited in the literature. Specifically, SL and RL techniques have been mostly investigated thanks to their dual capability of predicting driving conditions and controlling the vehicle, whereas UL approaches have not seen a spread ([86]). In fact, UL algorithms are typically trained at

understanding patterns within the input data when any specific target has to be directly forecasted. Therefore, UL-based control agents cannot directly fulfill the requirements of a complete EMS for an electrified vehicle. In Figure 4, a schematic overview of the main ML-based approaches for real-time control of electrified powertrains.

Regarding SL for the control of electrified powertrains, Neural Networks (NNs) surely represent one of the most performing tools to be embedded in a SL-based EMS ([87]). Several examples can be found in the literature. In [88], NNs have been accounted to make predictions about road type and traffic congestion to be fed into a second-level controller. In this case, a conventional vehicle had been chosen for testing the technology even if the approach was intended also for hybrid powertrains. In [89], Murphey *et al.* expanded the usage of NNs to both long-term and short-term power control of a HEV through the prediction of the driving mission category (NN_RT&TC) and the typical driver response on the pedal (NN_DT), respectively. Afterwards, Murphey *et al.* completed a two-paper series by presenting an additional couple of NNs responsible of predicting the optimal power-split between ICE and battery based on the outputs of NN_RT&TC and NN_DT as well as on the offline control solutions defined by a DP-based optimizer ([90]). In [91], the results produced by two distinct NNs with responsibility of controlling the power-split of a PHEV have been presented. Specifically, both the NNs were trained upon offline optimizations through DP, but one NN had been designed to control the vehicle in case of battery state of charge (SOC) lower than 30% while the other NN in case of battery SOC higher than 30%, knowledge of the future driving mission and residual trip duration larger than the all-electric range (AER). Differently from the EMS structure of [89] and [90], the NNs of [91] have been trained to learn the optimal control policy in a single step. A similar approach has been presented in [92], in which the performance of an EMS based on three different NNs coupled with an MPC controller have been exploited for a multi-mode PHEV. In this case, two NNs have been trained to predict the operating mode and the PMP co-state based on the optimal control trajectories defined by DP, whereas the third NN has been trained to predict the vehicle velocity. The outputs of the three NNs have hence been passed to a real-time capable MPC controller. Among the different types of NNs, recurrent NNs (RNNs) have proved to outperform other NNs, such as deep NNs (DNNs) or convolution NNs (CNNs), when the prediction of time histories is targeted ([93]). Regarding (P)HEV control, in [94] a RNN-based EMS has been trained to predict the optimal battery SOC trajectory of a PHEV relying upon a database of DP-based battery SOC profiles for

different driving conditions. The battery SOC prediction produced by the RNN were hence used for a controlling the vehicle through a hybrid PMP-PSO algorithm. A different utilization of a RNN is presented in [95], in which the ML-agent is directly trained to learn the optimal power-split between ICE and MG for a mild HEV. Despite simpler testing conditions linked to the prediction of a low-voltage HEV energy management, the RNN has proved the capability of completely replacing other control approaches (e.g. ECMS, PMP, etc.). Long-short term memory (LSTM) algorithms have also been exploited is the literature of (P)HEV real-time control as advanced implementation of RNNs, with the capability of mitigating vanishing gradients problems ([96]). Two examples can be found in [97] and [98], in which a LSTM has been integrated in the EMSs of two different HEVs as a vehicle velocity prediction tool, responsible of providing useful information to an MPC controller. Similarly, in [99] a LSTM network has been used to predict the vehicle velocity of a PHEV bus and pass the information to a hybrid GA-DP optimizer in the cloud. To the author's knowledge, a direct implementation of LSTM for the prediction of the optimal power-split between the components of an electrified powertrain lacks in the literature.

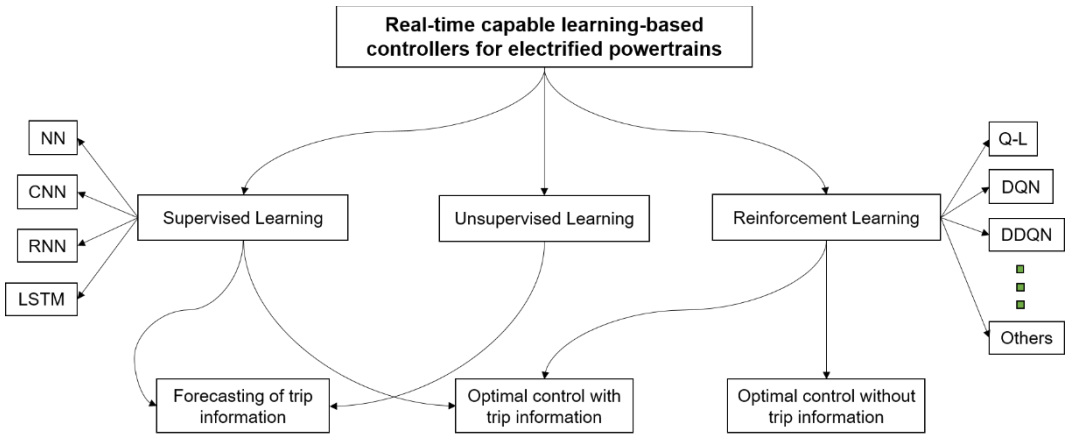

Figure 4. Learning-based control strategies for the real-time control of electrified vehicles.

As a final recap, controlling an electrified vehicle by means of a SL-based EMS can be performed by means of two main approaches: using SL algorithms to predict future driving conditions to be inputted into an EMS controller ([100]-[104]) or feeding SL controllers with information about the entire driving environment

([105]-[108]). According to the second case, the real-world utilization of a complete SL-based controller should involve the knowledge of future driving conditions. For example, information about future velocity trajectory or road altitudes could directly be fed into the ML-agent (NNs, RNNs, LSTMs, etc.) by means of GPS or GIS, respectively. More generally, vehicle-to-everything (V2X) communication represents an essential tool. Nevertheless, bounding the EMS with information about the future can turn into a strong limitation for any electrified application in which future data cannot be gathered (e.g. the trip destination is unknown).

In the very last years, a strong effort has also been paid by researchers to the analysis of RL algorithms performance when applied to the real-time control of electrified vehicles ([86], [109]). Differently from SL, the learning agents are not typically trained on a pre-constituted dataset whereas they are built to directly solve the optimization problem by means of a complex trial-and-error operation. Within this framework, vehicle state, controller actions and agent rewards have to be accurately designed to provide the RL agent with meaningful information about the powertrain behaviour. Thanks to a proper tuning of the agent, the limitation of embedding signals about future driving conditions into the EMS can theoretically be overcome. As far as the RL agents are concerned, several algorithmic structures can be taken into account, both excluding or including the usage of NNs. Concerning the RL-based EMSs excluding NNs, few research activities have focused on the development and application of RL control agents for (P)HEVs with the support of information about the future driving conditions. In [110], the information about the remaining travel distance is given to the controller. In [111] and [112], Liu et al. demonstrated the performance of a RL controller for HEVs assisted by a transition probability matrix (TPM) of the road power demand based on an a priori knowledge of the driving mission and on a vehicle velocity prediction, respectively. On the contrary, a larger share of research works has focused on real-time control of electrified vehicle through RL without future trip information. In [113]-[115], the capability of a Q-learning controller to minimize the fuel consumption throughout the driving mission has been assessed for a power-split FHEV, a MHEV and a range-extender electric vehicle (REEV), respectively. In [116], a similar Q-learning agent has been tested for the online control of a parallel HEV when modifications are applied to learning experience selection, state and action spaces discretization, number of system states and some hyper-parameters values. Moving to different electrified powertrains, the real-time control optimization of a FCEV through Q-learning has been presented in [117], in which

the control trajectory considers the usage of a fuel cell instead of a thermal engine as non-electric propeller.

One of the main drawbacks of RL techniques is represented by the difficulty of defining a unique setup of the agents when specific environment are faced. Unfortunately, only a small portion of the literature has focused on the analysis of the effects produced by changes in the configuration of the algorithms. This can be principally referred to the computational effort required in the training process of the agents. Specifically for the research field of electrified vehicles, more effort is hence needed to define approaches and methodologies aimed at assessing the real performances achieved by RL-based tools when tested on very different test cases, both in terms of driving conditions and algorithmic setups.

1.3 Research objective and main contributions

In the present work, a dissertation about optimal design and real-time control of hybrid and electric powertrains based on AI approaches is presented. Demonstrating the potentials of ML-related techniques when applied to the field of vehicle electrification is to be considered as the main research objective. Specifically, attention has been paid on the capabilities of SL and RL algorithms when applied to different optimization tasks about electrified vehicles. The methodologies have been developed and tested specifically for hybrid powertrains. Two motivations can be identified beyond this choice: first, data related to (P)HEVs have been available for modeling the vehicle and, hence, assessing for the effects of AI agents; second, (P)HEVs' optimal design and control problems can assume a significant complexity and, hence, they can represent an important proving ground for testing the potentials of ML approaches. Nevertheless, the proposed AI-based approaches are intended to be re-designed and possibly transferred to different testing cases, including different types of electrified vehicles.

As a preliminary step, fast-running low-throughput models for (P)HEVs have first been developed so as that the performances of ML approaches could be exploited for different (P)HEV test cases. Within this phase, the performances of different electrified powertrains have been studied by means of a deterministic DP (DDP) global optimizer. Based on the necessity of handling such a tool in a proper way, a relevant research effort has been devoted to the development of methodologies for a smart usage of DDP for (P)HEVs.

Thanks to a consolidated utilization of the DDP, the results obtained by the latter have been thoroughly analyzed with respect to the identification of optimal design and off-line optimal energy management of hybrid powertrains. The two operations have been conducted by introducing supervised ML algorithms with the aim of understanding the patterns behind the optimal decisions taken by the global optimizer. As an outcome, the development of supervised approaches for the identification of optimal design and real-time capable control policy has been performed based on direct results of DDP.

Finally, with the aim of demonstrating the possibility of ML to overcome the DDP limitation of identifying optimal control policies under real-time conditions, RL approaches have been analyzed. Specifically, the performances of a Q-learning based EMS have been evaluated related to a HEV application. A detailed study has

been about carried out about the effects produced by introducing variations into the main parameters of a Q-learning based EMS for real-world driving.

Considering the entire course of the research, the main contributions to the literature can be briefly introduced as follows:

- A methodology for the identification of the optimal mesh refinement of a DDP algorithm when considering the battery SOC as a state variable. According to an optimized number of battery SOC levels, a drastic reduction of machine time can be obtained while ensuring the maximum accuracy of the DDP calculations.
- The demonstration of the performances achieved by a pipeline of NNs when applied to the prediction of the results obtained by a DDP over a massive set of hybrid layouts. The accuracy of the predictions realized by the pipeline could allow for the identification of optimal design regions with a complete replacement of the DDP. Consistently, computationally heavy DDP optimizations could be neglected thanks to proper training operations of the pipeline.
- The assessment of the capability of LSTMs to learn and predict the optimal control choices obtained by a DDP off-line optimizer for a HEV. Differently from other utilizations of LSTMs, the latter have not been adopted to forecast information about the driving environment whereas they have been trained to directly learn the optimal utilization of the powertrain in terms of both gear and power-flow. A completely supervised controller has hence realized that can be tested on various driving conditions.
- A methodology to project and develop a Q-Learning agent for real-time control of HEVs, which could be useful to realize experiments aimed at defining the maximum performances achieved by RL-based EMSs. The results obtained during the research have proved the importance of accurately tuning the main parameters of Q-Learning to avoid drawing wrong conclusions about the RL agent behaviour. Specifically, attention has been paid in demonstrating the effects of changes in the discount factor, reward function, learning rate and exploration strategy.

1.4 Outline of the dissertation

The outline of the dissertation follows a precise structure that postpones the discussion of the capabilities of AI-based methodologies for electrified vehicles after the presentation of benchmark models and algorithms. The development of the latter has, in fact, been considered as a fundamental milestone to be achieved before taking a step towards SL and RL techniques.

In Chapter 2, a backward-facing modeling approach for HEVs is first introduced to the reader that will be used for modeling any other hybrid powertrain considered throughout the dissertation. Therefore, a DDP-based methodology for the identification of the optimal design of HEVs is presented and the results obtained for a *hdv* are discussed.

Attention has been paid during the research to the optimization of the DDP algorithm used in Chapter 2. In Chapter 3, a methodology for the selection of the best discretization level of the DDP state space is thoroughly described in the case of battery SOC as one of the state variables. The results presented in Chapter 3 have oriented the setup of the DDP grids used in the following research activities.

From Chapter 4 onwards, the capabilities of AI-based techniques are analyzed considering very different tasks. Specifically, the problem of optimal design of HEV fleets has been considered and the potentials of DNNs to predict the tank-to-wheel CO_2 emissions of different HEV parallel architectures have been discussed in Chapter 4. To this end, the algorithmic structure of the SL-based predictor has firstly been illustrated and then the main results obtained by its application have been presented to the reader.

Moving to Chapter 5 and Chapter 6, the opportunity of adopting ML for the real-time control of HEVs is investigated by presenting the potentials of SL and RL approaches, respectively. In Chapter 5, the capability of a particular family of RNNs (LSTMs) to directly control a hybrid powertrain under real-world conditions is discussed. Specifically, a methodology for building an efficient learning of a LSTM is described based on the optimal control trajectories obtained by a DDP algorithm. On the other hand, the project and development of a Q-Learning agent for HEVs' real-time control is presented in Chapter 6. In this case, a modular software framework is first described that embeds RL agent, environment of the experiment and simulator (HEV model). Therefore, a study of the effects produced by

variations in the most influencing parameters of the agent on the performances of the entire control system is conducted.

Chapter 2

Optimal design of hybrid powertrains through dynamic programming

In this Chapter, the topic of (P)HEV design optimization is disserted. Nevertheless, an exhaustive presentation of the (P)HEV model employed for the analyses is carried out. Starting from scratch, a "model" of a vehicle can be considered as a virtual representation of a vehicle that allows for conducting experiments (simulations) that might prove something about that specific system ([118]). Among the different types of models, physics-based mathematical models have been used in the research activity ([119]) and hence they are discussed in the present dissertation. Before stepping into the specific (P)HEV driveline, two additional clarifications can be made about the methodology and the assumptions made in the considered vehicle models. A "backward-facing" (or kinematic) model features the capability of the vehicle model to fulfill the power request coming from the driving missions ([120]).

Part of the present section has been extracted from:

Maino, C. *et al.* (2021). Optimal mesh discretization of the dynamic programming for hybrid electric vehicles. *Applied Energy*. Elsevier Ltd, 292(March), p. 116920. Doi: 10.1016/j.apenergy.2021.116920.

Anselma, P. G. *et al*. (2019). Comparing parallel hybrid electric vehicle powertrains for real-world driving. 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2019. AEIT, pp. 1–6. Doi: 10.23919/EETA.2019.8804609.

Such an approach is hence followed by evaluating the power required at the wheels level, then back propagating the torque and the speed request to the entire driveline (such as final drive, transmission, etc.) and finally achieving the propellers (ICE, MG). Thanks to this workflow, the adoption of backward-facing models prevails in the design optimization operations since the capability of the vehicle to run over a specific velocity trace can be immediately tested by modifying the sizing of the main powertrain components. On the contrary, a "forward-facing" (or dynamic) model embeds the adoption of dynamic models of the powertrain components as well as specific model of the driver ([120]). Thanks to the latter, a non-ideal (and more realistic) response of the vehicle to a reference acceleration/deceleration ramp can be simulated. Moreover, the utilization of dynamic model allows for a more accurate estimation of the effects produced by a given power request on a component. In this case, the performance of the propellers (e.g. ICE) is linked to the power request produced by the driver model. Finally, the power is propagated to the other powertrain components until the wheels are achieved and a given vehicle velocity output is produced. Differently from backward-facing models, a difference between the reference and the real vehicle velocity can be obtained. For this reason, forward-facing models are not typically considered for design optimization analyses while they are used when specific controllers have to be tested over a specific velocity trajectory ([67]).

As a first research objective, an optimization of the main components of hybrid powertrains has been targeted. Therefore, a backward-facing vehicle modeling approach has been identified as a valid choice and employed for the development of parallel (P)HEV models. For the sake of clarity, a similar approach could be considered for modeling other hybrid architectures, such as series, power-split or multi-mode ([121]). The selection of parallel configurations has been bounded to two main reasons: data availability and possibility of exploring of the most interesting architectures for FHEVs and PHEVs. In fact, parallel (P)HEVs have proved to be capable of achieving significant improvements in terms of fuel efficiency and pollutant emissions reductions without introducing drastic modifications to the conventional powertrains ([122]).

In the following sections, the vehicle model of a pre-transmission parallel (P)HEV architecture is presented.

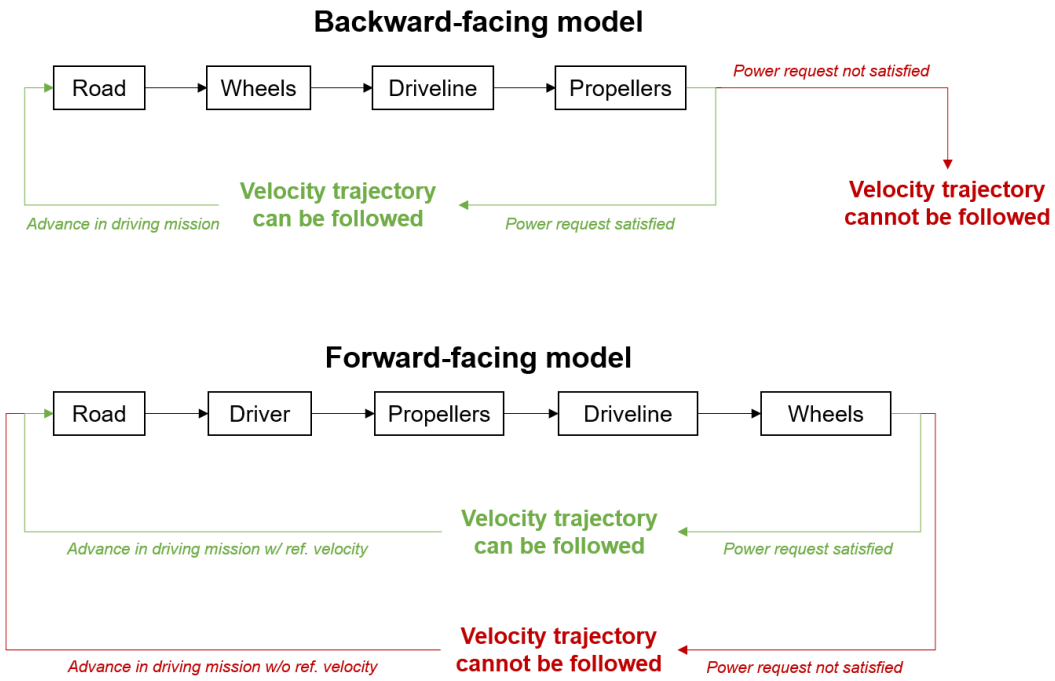

Figure 5. Workflows of backward-facing vs. forward-facing models.

2.1 Vehicle model of a pre-transmission hybrid electric powertrain

According to the backward-facing modeling approach, the velocity input of the road can be used to evaluate the power required at the wheels of the vehicle. The relative torque and speed demanded to the mechanical power sources (i.e. ICE and MG) can hence be calculated starting from the wheels and passing through any modeled component. Therefore, regardless the selection of a given (P)HEV architectures, the first step refers to the evaluation of the resistant forces of the vehicle.

According to [123] and [124], the resistance forces seen by a vehicle when driven longitudinally over a specific driving mission mainly depend upon aerodynamic friction losses, rolling friction losses and gravity-related forces. The total force to be generated at the wheels level F_t to win the resistant forces at a generic time instant t can be calculated by ([124]):

$$F_t(t) = F_a(t) + F_r(t) + F_g(t) + F_v(t) + m_v \frac{dv(t)}{dt} \quad (1)$$

in which F_a, F_r, F_g and F_v represent the aerodynamic friction, the rolling friction, the force produced by the gravity on non-flat road paths and the additional force produced by various non specified effects, respectively, m_v is the vehicle mass and $dv(t)/dt$ is the acceleration of the vehicle. The contribution brought by aerodynamic friction losses ("drag resistance") to the total resistance force is calculated through the following equation:

$$F_a(t) = \frac{1}{2} \cdot \rho_a \cdot A_f \cdot c_d \cdot v(t)^2 \quad (2)$$

in which ρ_a is the density of the ambient air, A_f is the frontal section of the vehicle and c_d is the drag coefficient specifically calculated for the considered vehicle ([125]). The effects of the rolling frictions ("rolling resistance") on the total resistance force are calculated by:

$$F_r(t) = c_r(v(t), p, r) \cdot m_v \cdot g \cdot \cos \alpha(t) \quad (3)$$

in which c_r is the rolling coefficient, g is the acceleration due to gravity and α is angle between the actual road slope and a flat trajectory. The rolling coefficient c_r can depend upon multiple variables, such as the tire pressure p and the road surface conditions r ([126]). Therefore, it could be subject to variations within the same driving scenario. In this research, a constant rolling coefficient has been considered throughout the entire driving mission assuming negligible variations of the tire pressure as well as of the road surface. Considering non-flat roads, the effects of the force produced by the gravity ("grade resistance") have to be taken into account:

$$F_g(t) = m_v \cdot g \cdot \sin\alpha(t) \tag{4}$$

The contribution of the grade resistance would clearly be nullified if a flat road is driven (i.e. $\alpha = 1,\ \sin\alpha = 0$); on the contrary, positive or negative grade resistances should be considered in case of uphill or downhill, respectively. Regarding F_v, the possibility of involving unexpected resistance forces has been neglected.

The final equation used to calculate the resistance force in case of non-flat roads can hence be written as:

$$\begin{aligned} F_t(t) = \frac{1}{2} \cdot \rho_a \cdot A_f \cdot c_d \cdot v(t)^2 + c_r \cdot m_v \cdot g \cdot \cos\alpha(t) + \\ + m_v \cdot g \cdot \sin\alpha(t) + m_v \frac{dv(t)}{dt} \end{aligned} \tag{5}$$

while for flat roads:

$$F_t(t) = \frac{1}{2} \cdot \rho_a \cdot A_f \cdot c_d \cdot v(t)^2 + c_r \cdot m_v \cdot g + m_v \frac{dv(t)}{dt} \tag{6}$$

In the present model, the time history is not considered as a continuous variable whereas it is discretized by a given number of time steps based on the length of the driving mission to be simulated. An adaptive width of the time steps could be considered; nevertheless, a default value of 1 second has been implemented according to the typical assumption of the main type-approval tests for on-road

vehicles (e.g. WLTP, FTP, etc.). Within a discretized environment, finite differences have been considered for approximating the derivatives, such as the derivative of the vehicle velocity in time (i.e. vehicle acceleration). Among the possible methods, backward differences have been considered for the approximation of the derivatives ([127]). Therefore, the following equation has been implemented for calculating the vehicle acceleration at a given time step $a(t)$:

$$a(t) = \frac{dv(t)}{dt} = \frac{v(t) - v(t-1)}{\Delta t} \tag{7}$$

in which Δt=1s. For the sake of simplicity, the same approach will have to be considered for calculation of any other derivative.

As far as design optimization are concerned, GOs are typically employed for identifying the optimal solution to an off-line control problem in which the entire driving profile is known before of the optimization. The usage of central difference could hence be exploited as approximation method of the same derivative; nevertheless, the knowledge of future conditions about the driving mission has to be assumed. In the perspective of employing a backward-facing model for multiple tasks (such as optimization of the real-time capable controllers), assuming the complete knowledge of the driving missions cannot always be a valid hypothesis. Therefore, backward differences turn to be a more flexible methodology to be employed for any simulation setup either including or excluding the availability of future trip information.

Consistent with the prerequisites of a backward-facing modeling approach, the vehicle velocity $v(t)$ is to be set as a reference velocity imposed by the road. Therefore, a generic driving mission has necessarily to be modeled at least by means of a temporal information of the desired velocity. Moreover, data about the road slope (i.e. the variation in the road altitude) could also be used to include non-null values of F_g in the calculation of the total resistant force to the motion (see (4)). The velocity and road slope signals can hence be employed as inputs of the vehicle model so as to proceed with the calculation of the resistance forces according to (5) and (6).

Once the total resistance force F_t has been defined, the force to be outputted by the entire powertrain at the wheels level is known. The losses and the inertial terms of any rotating included in the driveline can hence be estimated starting from the value of F_t. Consequently, the load request of the propeller(s) can be calculated.

For the sake of simplicity, the following equations will refer to the vehicle longitudinal dynamics in case of flat road. The total power required to traction the vehicle P_t can be calculated as:

$$P_t(t) = F_t(t) \cdot v(t) \tag{8}$$

from this stage, moving backwardly until the power energy sources, the model equations can differ depending upon the specific powertrain to be modeled. In the present section, the entire set of equations considered for a pre-transmission parallel (P)HEV is presented. For insights about the models considered for different parallel (P)HEV architectures, please refer to Appendix A. The hybrid powertrain reported in Figure 6 has been modeled in the MATLAB® software environment.

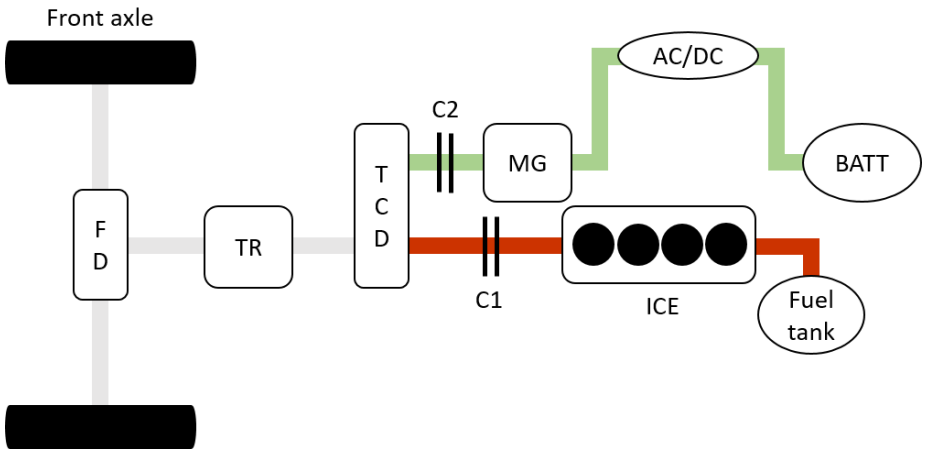

Figure 6. Driveline of a pre-transmission parallel hybrid electric powertrain.

According to a backward-facing model, the electric path (green) of a pre-transmission (P)HEV is encountered after the transmission (TR). A final drive (FD) is modeled as a speed multiplier between the main shaft (grey) and TR. The latter has been modeled as a discrete multi-gears transmission. Each gear has been considered a speed multiplier with a given efficiency. Such an assumption has been forced by the availability of data about a manual gearbox for conventional vehicles. Nevertheless, increasing the number of gears rather than changing the type of

transmission (e.g. CVT) would not imply any change in the presented model. Moving backwardly from that stage, a torque-coupling device (TCD) is considered as an additional speed multiplier, which connects the electric power sources to the main shaft. Beyond the TCD level, two clutches (C1 and C2) are positioned for engaging/disengaging the engines from the wheels. Finally, the powertrain propellers (i.e. ICE and MG) are achieved.

The set of equations used to calculate the power required at the propellers starts with the evaluation of the power at the wheels P_{wh}:

$$\begin{gathered} P_{wh}(t) = (P_t(t) + I_{wh} \cdot \dot{\omega}_{wh}(t) \cdot \omega_{wh}(t)) \cdot \eta_{wh}^{k} \\ P_{wh}(t) = (P_t(t) + P_{in,wh}(t)) \cdot \eta_{wh}^{k} \end{gathered} \tag{9}$$

where $P_{in,wh}$ represents the inertial power of the wheels as a function of their inertia coefficient I_{wh}, rotational acceleration $\dot{\omega}_{wh}$ and rotational speed ω_{wh}, η_{wh} represents the efficiency of the wheels and k is an index of traction/braking conditions. Specifically, k is equal to -1 during traction and 1 during braking phases. In fact, losses have to be accounted additional terms to the power demanded to the propellers during traction phases; on the contrary, the kinetic energy available for regeneration is lowered by the same losses when the vehicle is braking ([34]). Regarding the sub-model of the vehicle tires, a detailed model of the real forces applied to the tires (such as those presented in [128]) has not been implemented due to a lack of data. Therefore, the wheels have been modeled as rigid bodies rotating at a given speed with a given inertia.

Consistent with the driveline of Figure 6, the power right after the TCD level is equal to the power at the wheels P_{wh} at net of the efficiencies of FD, TR and TCD. Therefore, if any loss is considered in the driveline until TCD:

$$P_{wh}(t) = P_{fd}(t) = P_{tr}(t) = P_{tcd}(t) \tag{10}$$

where P_{fd}, P_{tr}, P_{tcd} represent the power at the FD, TR and TCD, respectively. On the contrary, if non-null losses are considered throughout the driveline, a set of equations is applied to compute the power at the TCD:

$$P_{fd}(t) = (P_{wh}(t) + P_{in,fd}(t)) \cdot \eta_{fd}^{k} \quad (11)$$

$$P_{tr}(t) = (P_{fd}(t) + P_{in,tr}(t)) \cdot \eta_{tr,i}^{k} \quad (12)$$

$$P_{tcd}(t) = (P_{tr}(t) + P_{in,tcd}(t)) \cdot \eta_{tcd}^{k} \quad (13)$$

in which $P_{in,fd}$, $P_{in,tr}(t)$ and $P_{in,tcd}(t)$ are the inertial powers of FD, TR and TCD, respectively, while η_{fd}^{k}, $\eta_{tr,i}^{k}$ and η_{tcd}^{k} are the efficiencies of FD, TR and TCD, respectively. Similar to (9), the inertial powers of the three rotating components have been calculated based on their relative inertial coefficients. As far as TR is concerned, a given efficiency has been considered for each i-th activated gear. Once the power at the TCD is known, the power demanded to the propellers during traction phases can be calculated according to four different operating modes ([56]):

- Pure electric mode (*pe*): the MG is used as only propeller of the vehicle (C1 disengaged, C2 engaged);
- Pure thermal mode (*pt*): the ICE is used as only propeller of the vehicle (C1 engaged, C2 disengaged);
- Power-split mode (*ps*): the MG and the ICE are used simultaneously to propel the vehicle (C1 and C2 engaged);
- Battery charging mode (*bc*): the MG is used as generator while the ICE provides enough power to both propel the vehicle and recharge the battery.

About braking conditions, regenerative braking is considered as only operating mode (i.e. C1 disengaged, C2 engaged and MG working as generator to recharge the battery). Assuming P_{req} to be the power at the TCD including the inertial terms of both the propellers ($P_{req} = P_{tcd} + P_{in,ICE} + P_{in,MG}$), the powers of MG ($P_{MG}$) and ICE ($P_{ICE}$) can be calculated as:

$$P_{MG}(t) = \alpha_{om} \cdot P_{req}(t) \quad (14)$$

$$P_{ICE}(t) = (1 - \alpha_{om}) \cdot (P_{req}(t) + P_{aux,ICE}) \quad (15)$$

in which α_{om} ("power-split ratio") is an index that allows for the determination of the power demands to the propellers according to the different operating modes of the (P)HEV architecture while $P_{aux,ICE}$ is a fixed power load related to the auxiliaries of the ICE. A similar power term about the auxiliaries of the electric components will be introduced in the evaluation of the battery power. Depending on the value assumed by the power-split ratio, the operating mode of the hybrid system can be indirectly identified:

- $\alpha_{om} = 1$ (*pe*) $\rightarrow$ $P_{ICE} = 0, P_{MG} = P_{req}$;
- $\alpha_{om} = 0$ (*pt*) $\rightarrow$ $P_{MG} = 0, P_{ICE} = P_{req}$;
- $0 < \alpha_{om} < 1$ (*ps*) $\rightarrow$ $0 < P_{MG} < P_{req}, 0 < P_{ICE} < P_{req}$;
- $\alpha_{om} < 0$ (*bc*) $\rightarrow$ $P_{MG} < 0, P_{ICE} > 1$

In case of *bc* mode, the ICE propels the vehicle with a higher power with respect to the one required to traction the vehicle ($P_{ICE} > 1$) while the MG operates as generator to take advantage of the exceeding power ($P_{MG} < 0$). For the sake of clarity, a positive sign is used to define a power outgoing from a component, whereas a negative sign is used to define a power entering a component.

In Section 2.2, the problem of defining an offline policy with the aim of optimizing the usage of the different power components is analyzed. Nevertheless, the sub-models of the main components included in the driveline are first presented, specifically ICE, MG and battery.

2.1.1 Sub-models: Internal Combustion Engine

The ICE has been modeled according to experimentally derived 2-D look-up tables in which the fuel consumption (FC) can be calculated based on the ICE torque and speed. Accordingly, engine efficiency and other indexes (e.g. brake specific fuel consumption) can be evaluated. Examples of the ICE operating maps featured by FC and efficiency are reported for a generic application in Figure 7 and Figure 8, respectively. The wide open throttle (wot) and the motoring (mot) curves have also been modeled ([129]). The wot has been employed to determine the maximum load achieved at the different engine rotational speed, whereas the mot has been considered to discriminate between ICE on/off operations with negative requested torques (non-traction conditions). Specifically, the ICE has been modeled to be

switched off in case of operating points falling below the mot. According to this approach, the clutch C1 (Figure 6) is disengaged and the powertrain is capable of maximizing the power entering the generator to recharge the battery. Without any further dynamics regarding ICE transient conditions, only an additional fixed FC has been considered during each time step with engine switching on.

For the sake of clarity, the ICE operating maps have been associated to few but important ICE characteristics: displacement, peak power, mass and inertia coefficient. Based on the ICE characteristics, different ICE sizing can be simulated through a scaling operation over the operating maps. In Section 2.2, the scaling operation is thoroughly described.

Consistent with the presented backward-facing model and the ICE-sub model, the actual FC can be estimated at each time step of a generic driving mission by means of an interpolation in the operating maps. Once the final step of the mission has been performed, the evaluation of the cumulative FC can be realized and the related tank-to-wheel CO_2 emissions can be estimated.

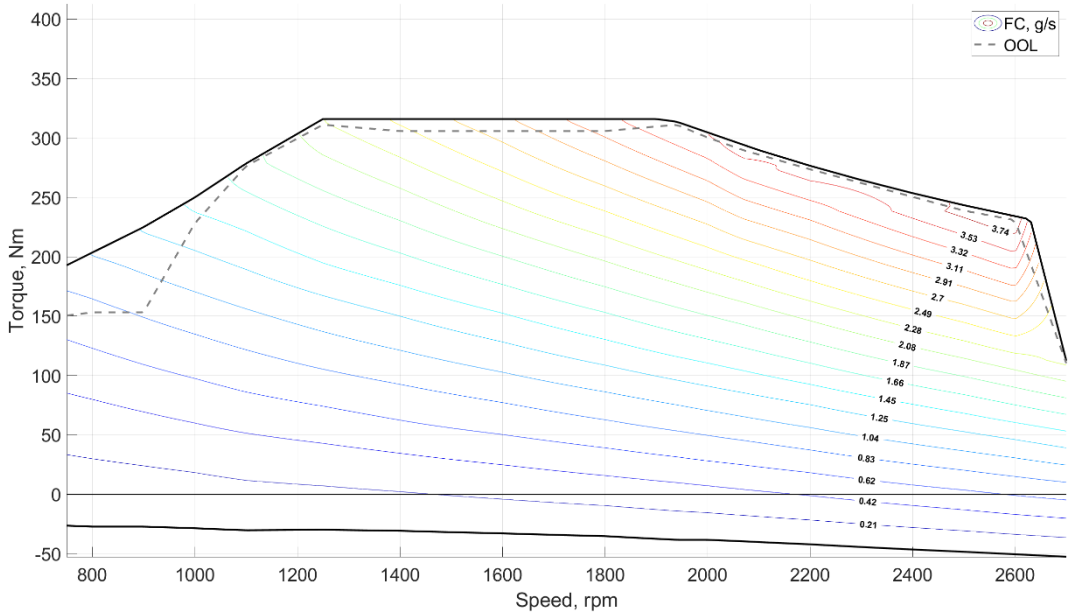

Figure 7. Example of a 2-D operating map of an internal combustion considering the engine fuel consumption.

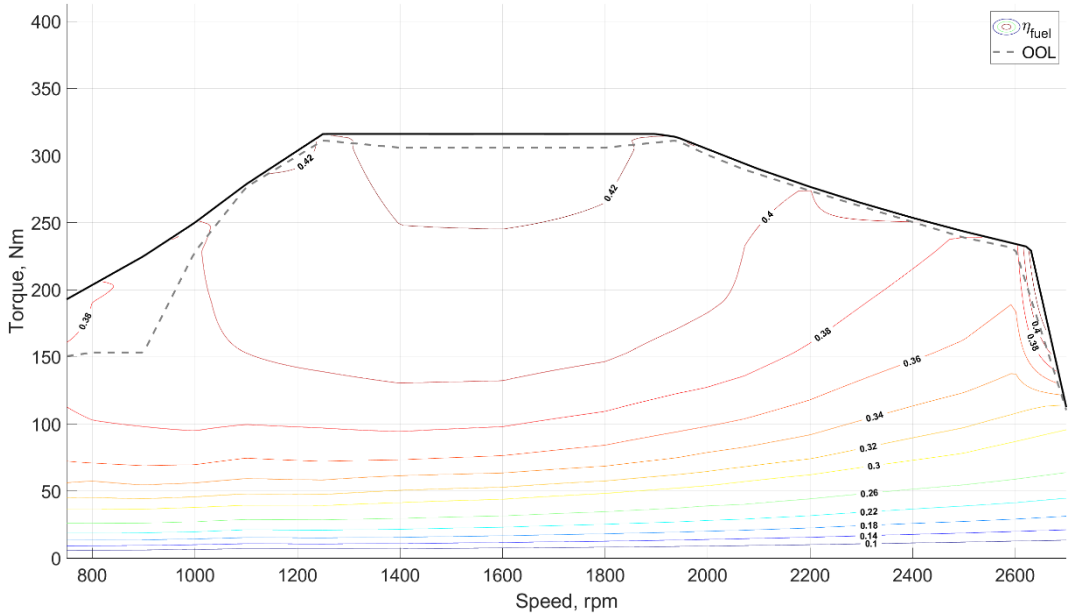

Figure 8. Example of a 2-D operating map of an internal combustion engine considering the engine efficiency.

A similar approach could be employed if other ICE-based effects should be assessed, such as NO_x and PM emissions for Diesel engines ([130]) or CO and PN for gasoline engines ([131]). Within this dissertation, only the CO_2 emissions have been accounted for any optimization procedure and hence no additional ICE maps have been considered in the studies.

2.1.2 Sub-models: Motor-Generator

In a pre-parallel (P)HEV architecture, the conventional engine is typically coupled with an electric machine with the capability of working as a generator (i.e. motor-generator) ([34]). The electric power coming from an electric energy source can be converted into mechanical power to propel the vehicle when the MG is working as a motor, whereas the opposite condition can be realized during braking phases when the MG works as a generator ([132]). The possibility of minimizing the number of additional components to be integrated within the conventional powertrain and realize a parallel architecture can consistently be achieved.

According to the available data, brushless permanent magnet MGs ([133]) have been considered for the entire set of analyses presented in the dissertation. As for the ICE, the operating conditions of MGs have been evaluated by means of 2-D

look-up tables in which the MG efficiency is defined as a function of torque and speed. In Figure 9, an example of a MG operating map representative of the iso-efficiency curves is charted. In this case, the operating conditions of the MG working as a generator (negative torques) have been considered to be equal to those working as a motor (positive torques) due to a lack of data. On the contrary, two distinct areas could be identified in case of data availability.

As for the ICE sub-model, the operating map of the MG has been associated to some characteristic variables: peak power in traction, peak power in regeneration, mass and inertia coefficient. Based upon these, a scaling operation which can be performed on a MG operating map is presented in Section 2.2.

The relation between mechanical and electric powers outputted by the MG at a given time step of a generic driving mission is considered as:

$$P_{MG,el}(t) = P_{MG,mech}(t) \cdot \eta_{MG}^{k}(T_{MG,mech}, \omega_{MG}) \tag{16}$$

in which $P_{MG,mech}$ and $P_{MG,el}$ represent the MG mechanical and electric powers, respectively, while η_{MG}^{k} is the MG efficiency related to the specific working point as a function of MG mechanical torque $T_{MG,mech}$ and speed ω_{MG}. Consistent with the power conversion of a MG, $P_{MG,mech} < P_{MG,el}$ during traction conditions (k=-1) whereas $P_{MG,mech} > P_{MG,el}$ during braking conditions (k=1). No distinction between peak and continuous power is modeled for the MG, hence only the maximum effort achieved by the MG on a generic driving mission have been estimated.

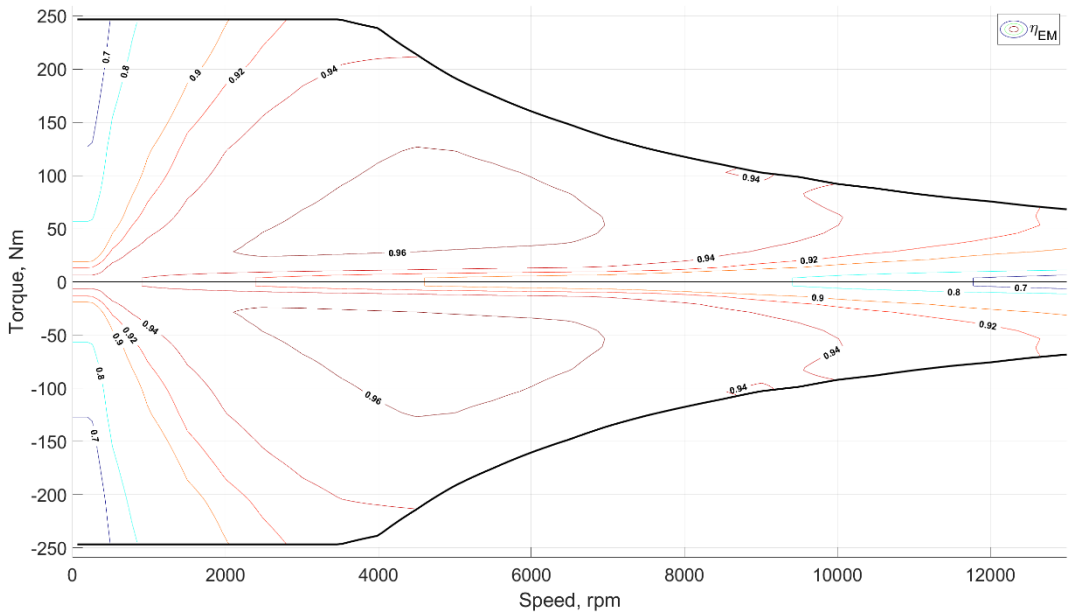

Figure 9. Example of a 2-D operating map of a motor-generator considering the motor-generator efficiency.

2.1.3 Sub-models: battery

The calculation of the MG electric power allows for the determination of the electric power load demanded to a rechargeable energy storage system (REESS). An AC/DC converter has been considered between the MG and the REESS with the capability of converting the DC current outputted by the REESS into an AC current to be read by the MG, and vice versa ([134]).

The AC/DC has been modeled by means of a fixed efficiency ($\eta_{AC/DC}$) which leads to the calculation of the REESS electric power demand $P_{REESS,el}$ according to:

$$P_{REESS,el}(t) = P_{MG,el}(t) \cdot \eta_{AC/DC}^{k} + P_{aux,REESS} \quad (17)$$

in which $P_{aux,REESS}$ represents the load of electric auxiliaries to be constantly considered when the electric sources are operated. Among the different REESS technologies and models ([135]), a low-throughput simplified model of a Li-Ion battery has been developed and used to estimate the energy lost or recuperated during traction or braking phases, respectively. Specifically, the internal resistance

(R_{int}) equivalent model ([136]) selected for the entire research activity is presented in Figure 10. The open-circuit voltage (V_{oc}) and the internal resistance (R_i) of the battery have been considered according to experimental data as a function of the battery SOC. Example trends of the two characteristics are charted for a single cell in Figure 11.

According to the actual battery SOC and the relative V_{oc} and R_i, the possibility of the battery to provide the vehicle with a power contribution during traction $P_{b,el}^{tr}$ can be evaluated through:

$$P_{b,el}^{tr}(t) = P_{REEES,el}^{tr}(t) < \min(P_{b,max,1}^{tr}(t), P_{b,max,2}^{tr}(t)) \tag{18}$$

with:

$$P_{b,max,1}^{tr}(t) = V_{oc}(SOC(t)) \cdot I_{dis,max} - R_i(SOC(t)) \cdot I_{dis,max}^2 \tag{19}$$

$$P_{b,max,2}^{tr}(t) = \frac{V_{oc}^2(SOC(t))}{4 \cdot R_i(SOC(t))} \tag{20}$$

in which $P_{b,max,1}^{tr}$ is the result of a power balance applied to the R_{int} model circuit, whereas $P_{b,max,2}^{tr}$ stems from the value of the maximum current admitted in the battery during discharge mode $I_{dis,max}$. The latter is evaluated by means of the availability of data about the battery capacity C_b and the maximum C-rate during traction $C_{dis,max}$ ([137]).

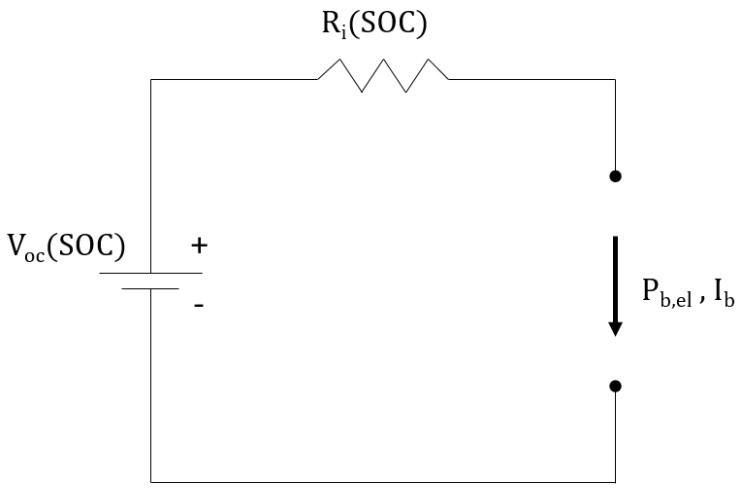

Figure 10. Internal resistance equivalent model of a Li-Ion battery.

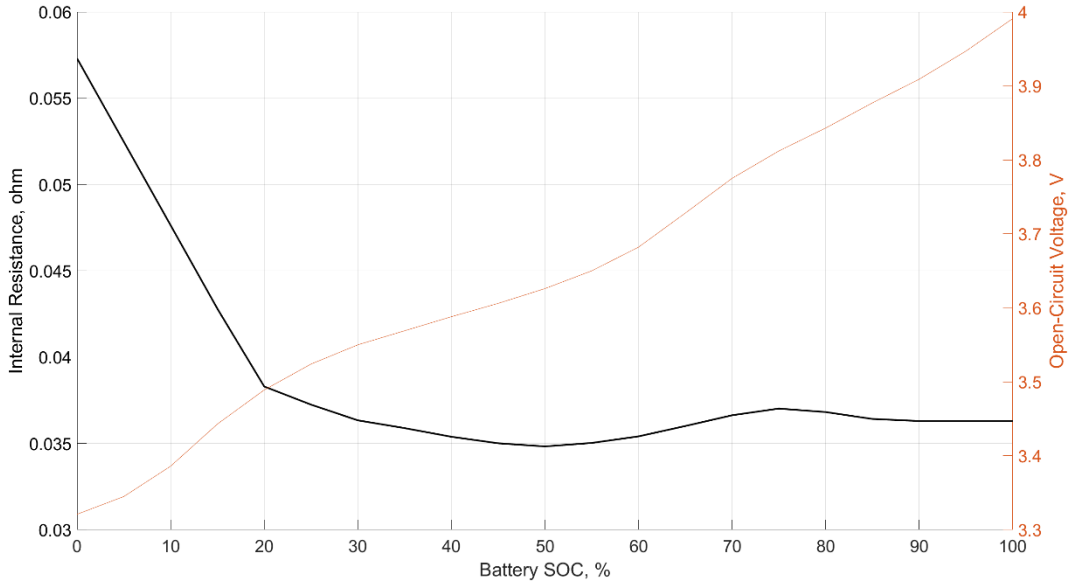

Figure 11. Example of the internal resistance and open-circuit voltage trends as a function of battery SOC.

According to these values, the maximum current in traction can be calculated as:

$$I_{dis,max} = \frac{C_b}{1h} \cdot C_{dis,max} \tag{21}$$

As far as braking events (i.e. battery regeneration phases) are concerned, the power available for a battery recharge $P_{b,el}^{br}$ can be calculated as:

$$P_{b,el}^{br}(t) = P_{REEES,el}^{br}(t) > P_{b,max}^{br}(t) \tag{22}$$

with:

$$P_{b,max}^{br}(t) = -V_{oc}(SOC(t)) \cdot I_{ch,max} - R_i(SOC(t)) \cdot I_{ch,max}^2 \tag{23}$$

in which $P_{b,max}^{br}$ is the maximum power to be inputted in the battery during charging conditions considering the maximum current admitted during regeneration $I_{ch,max}$. Similar to (21), $I_{ch,max}$ has been calculated as:

$$I_{ch,max} = \frac{C_b}{1h} \cdot C_{ch,max} \tag{24}$$

where $C_{ch,max}$ is representative of the maximum C-rate during regeneration ([137]). Consistent with the evaluation of the battery power in traction $P_{b,el}^{tr}$ or braking $P_{b,el}^{br}$, the current flowing into the battery I_b can be obtained for each time instant and used to calculate the battery SOC variation according to a Coulomb Counting (CC) method ([138]):

$$SOC(t) = SOC(t-1) - \int_{t-1}^{t} \frac{\eta_c \cdot I_b}{C_b} dt \tag{25}$$

which turns into the following similar formulation for a time-discretized environment:

$$SOC(t) = SOC(t-1) - \frac{\eta_c \cdot I_b \cdot \Delta t}{C_b} \tag{26}$$

where η_c represents the coulombic efficiency ([139]). The latter has been set to the unity for the entire research activity. A positive sign has been considered for I_b flowing out of the battery, whereas a negative sign has been defined for currents entering in the battery. According to (26), the evolution of the battery SOC trajectory can be computed during a generic driving mission. As far as electrified vehicles are concerned, the trend of the battery SOC plays a fundamental role in the identification of the vehicle state as well as in the definition of good or bad control policies. In the following section, an approach for the identification of the optimal control policy is presented for (P)HEV applications.

2.2 Identification of a benchmark control policy

Once the backward-facing model of a (P)HEV has been completed, the objective of establishing a methodology for the optimization of a hybrid powertrain design has been targeted. To this end, the definition of an approach for the identification of a benchmark control policy is a key step to be carried out. In this Section, the approach used for analyzing the optimal performance of a pre-transmission (P)HEV architecture in a discretized environment is presented. Apparently, a customization operation has to be performed if the same approach is intended to be used on any other parallel hybrid architecture reported in Appendix A.

A DDP GO has been selected as the most promising method for the identification of the optimal control policy of a (P)HEV when simulated on a generic driving mission according to [140]. A control policy is here defined as a combination of decisions taken at each time step throughout the entire driving mission. The DDP algorithm has proved the capability of solving complex control problems in time-discretized environments while ensuring the convergence to a global optimum. Thanks to the identification of the latter, the best performances achieved by (P)HEV on a specific driving mission can be evaluated. To this end, a "stage cost" (or "running cost") has been identified as a function of the state of the system X and the control decision U:

$$g(X,U) = \dot{m}_{fc} \tag{27}$$

in which $\dot{m}_{fc}$ is the actual FC evaluated at each time step of the driving mission between the initial (t_0) and final (t_N) time instants. The optimization process consists of two phases, namely backward and forward phases. In the backward phase (from t_N to t_0), the optimal "cost-to-go" function J^* is calculated for each stage of the driving mission, whereas the final control trajectory is recreated in the forward phase (from t_0 to t_N). For the sake of simplicity, the stages of the driving mission have been considered to correspond to the time steps of the driving mission. During the backward phase, the cost-to-go function and the optimal control decision can be evaluated based on following mathematical formulation of the problem:

$$J_t^*(X_t) = g_N(X_N) + \min_{U_t}(g_t(X_t,U_t) + J_{t+1}^*(X_{t+1})) \qquad with\ t = 1, \dots N-1 \tag{28}$$

$$U_t^*(X_t) = g_N(X_N) + \underset{U_t}{argmin}(g_t(X_t, U_t) + J_{t+1}^*(X_{t+1}))\ \ with\ t = 1, \dots N-1$$

According to (28), the optimal control policy $U = [U_1^*, \dots, U_{N-1}^*]$ can hence be defined. The evolution of the system (i.e. the vehicle) in a time-discretized environment can be written as:

$$X_{t+1} = f(X_t, U_t, W_t)\ \ with\ t = t_0, \dots, t_N \tag{29}$$

where $X = \{x_1, x_2, \dots, x_n\}$ represents the state of the vehicle with n state variables, $U = \{u_1, u_2, \dots, u_m\}$ is the set of control decisions defined by the single decisions made upon u_m sub-control variables and $W = \{w_1, w_2, \dots, w_z\}$ is the set of w_z exogenous inputs ([141]). The DDP algorithm has hence been adopted to find the optimal set of decisions U which leads to the minimization of J while ensuring a feasible evolution of the state X while complying with the exogenous inputs W of the problem. The state space (i.e. the set of state variables), the control space (i.e. the set of sub-control variables) as well as the exogenous inputs can be different according to the specific optimization problem. Moreover, the GO could involve different boundary conditions about the vehicle state feasibility if different hybrid architectures are studied, such as HEVs or PHEVs. For instance, a DDP algorithm can be employed to optimize the control policy in case of a charge-sustaining (CS) condition for a HEV ([142]) or in case of a mixed charge-depleting/charge-sustaining (CD/CS) strategy for a PHEV ([143]).

Since the DDP configurations could significantly vary depending upon the specific application, a particular setup of the algorithm had to be chosen. In the following sections, the definition of the state and control variables as well as of the exogenous inputs and boundary conditions considered in the entire research activity involving DDP are detailed.

2.2.1 DDP: state space

The setup of the DDP state space considered in the present dissertation is hereafter presented. The battery SOC has been considered as a discrete variable with values comprised within the minimum and the maximum thresholds of a user-defined SCO window. Moreover, the ICE state has been included in the DDP state space as a Boolean variable referred to ICE on and off operations. The final set of state variables appears as:

$$X = \{x_1, x_2\} \quad with \begin{cases} x_1 = [SOC_1, \dots, SOC_D] \\ x_2 = [0, 1] \end{cases} \tag{30}$$

in which x_1 and x_2 are the state variables featuring battery SOC and ICE on/off, respectively, while SOC_1 and SOC_D are the lowest and highest discrete values achieved by the battery SOC, respectively.

2.2.2 DDP: control space

Recalling the four macro operating modes of a pre-transmission hybrid powertrain (i.e. *pe*, *pt*, *ps*, *bc*) defined in Section 2.1, one of the DPP control variables employed in the literature typically refers to the torque-split between the ICE and the MG ([144]). For the present research, the torque-split has been considered according to the configuration set in [35]: a sub-control variable α has been connected to user-defined values of torque-split between ICE and MG. Based on (14) and (15), multiple values of α have been selected and passed to the DDP as first control variable. The indexes related to each α value will be referred as power-flow index (PFI). Moreover, an additional sub-control variable τ has been considered to account for any gear to be possibly inserted according to the modeled transmission presented in Section 2.1. The gear associated to each τ will be referred as gear number index (GNI).

The final set of control variables can be written as:

$$U = \{u_1, u_2\} \quad with \begin{cases} u_1 = [\alpha_1, \dots, \alpha_M] \\ u_2 = [\tau_1, \dots, \tau_Z] \end{cases} \tag{31}$$

in which M is the user-defined number of torque-split values (i.e. the amount of PFIs) and Z is the number of gears available in the modeled transmission.

2.2.3 DDP: exogenous inputs and boundary conditions

A couple of exogenous inputs has been considered in the study, featuring the vehicle velocity v and the road grade g. The set of exogenous inputs can be defined as:

$$W = \{w_1, w_2\} \quad with \begin{cases} w_1 = [v(1), \dots, v(T)] \\ w_2 = [g(1), \dots, g(T)] \end{cases} \tag{32}$$

where T is the terminal time step of the driving mission. In case of flat roads, w_2 has been neglected since comprised only of null values.

Finally, three boundary conditions have been considered in the research activity:

$$SOC(T) = SOC^* \tag{33}$$

$$SOC_{min} < SOC(t) < SOC_{max} \quad for\ t = 1, \dots, T \tag{34}$$

$$v_{req}(t) = v(t) \quad for\ t = 1, \dots, T \tag{35}$$

where $SOC(T)$ is representative of the battery SOC value at the final time step of the driving mission, SOC^* is a user-defined battery SOC value to be a identified before the optimization process begins and v_{req} is the vehicle velocity required by the road at each time step (exogenous input). Considering the first boundary condition expressed by (33), an optimal CS trajectory for a HEV could be exploited by setting $SOC^* = SOC(1)$ whereas a CD trajectory for a PHEV could be optimized imposing $SOC^* = SOC^{CD} < SOC(1)$. Then, the second condition forces the battery SOC to not exceed over or under the maximum and minimum battery SOC values, respectively. The feasible battery SOC range will be referred as "SOC window". Finally, the last condition of (35) directly refers to the workflow of a backward-facing vehicle model (Figure 5) in which the velocity profile is used as an external signal to be strictly followed at each time step of the driving mission.

2.2.4 Feasibility and admissibility

With the aim of simplifying the readability of the results reported in the next sections of the dissertation, a distinction between the terms "feasible/unfeasible" and "admissible/inadmissible" is hereafter clarified. A feasible control policy is obtained when the decisional chain produced by the controller (e.g. DDP) can contemporary satisfy the conditions imposed by (33), (34) and (35). Therefore, the terminal battery SOC is equal to the user-defined SOC value, the battery SOC is always comprised within the SOC window throughout the entire driving mission and at least one action can be selected at each time instant to propel the vehicle at the required velocity. Related to feasible control policies, feasible control variables are realized throughout the driving mission. On the, if at least of the three boundary conditions defined in Section 2.2.3 is not satisfied, the control policy is considered

as unfeasible. A feasible layout is hence intended as a layout which produced a feasible control policy, vice versa for an unfeasible layout.

In the following sections, admissible or inadmissible hybrid layouts will also be cited. Specifically, an admissible (P)HEV layout is intended as a layout which proves the capability to satisfy minimum drivability requirements. The latter have been evaluated according to specific tests of maximum velocities, maximum accelerations, gradeability and pure electric mode driving. For a deeper insight about performance tests for (P)HEV, please refer to [35] and [145]. On the contrary, an inadmissible layout has been identified when at least of the drivability tests has not been completed.

As far as the correlation between admissible layouts and DDP is concerned, an admissibility study has been performed on each (P)HEV layout before the beginning of a specific optimization with DDP. Thanks to this approach, only admissible layouts could be considered in the wider and more generic (P)HEV design optimization procedure presented in the following Section.

2.3 A procedure for the design optimization of hybrid electric vehicles considering scaling of the components

As far as (P)HEVs are concerned, the identification of the optimal design is one of the most exploited topics in the literature ([146]). Examples can be shown, such as [147], in which the optimal design of a parallel hybrid is identified by comparing the results produced by four different control strategies on a given set of (P)HEV layouts. Similarly, the design of a parallel architecture has been optimized according to the results of a DP algorithm in [148] and a sizing rule for the definition of the optimal degree of hybridization. In [149], the optimal design is searched with DP and hence an ECMS is applied under real-world conditions considering the optimal sizing of the powertrain components.

Regardless the specific application, defining the optimal sizing of the main powertrain components is necessary for the assessment of the best energetic performances achieved by different (P)HEV architectures and layouts. The vehicle model and the DDP configuration presented in Section 2.1 and Section 2.2, respectively, have been integrated within a complete design optimization procedure for (P)HEVs. The workflow of Figure 12 is representative of the different steps considered in the design procedure optimization procedure for the pre-transmission parallel (P)HEV (Section 2.1).

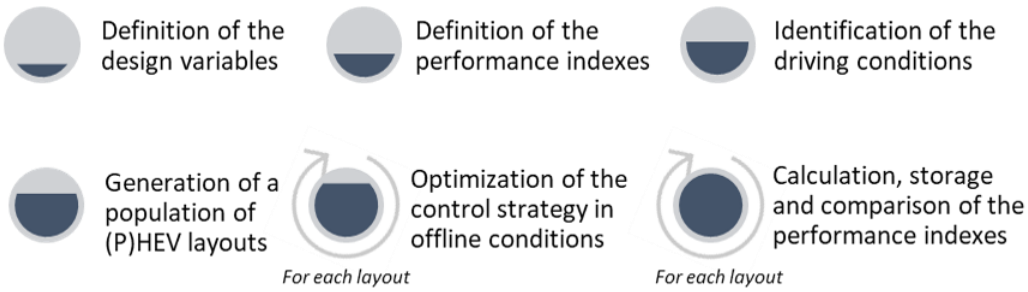

Figure 12. Steps of a complete design optimization procedure for hybrid electric powertrains.

The initial stage of the procedure consists of the definition of the "design variables", i.e. the variables to be optimized that are representative of the (P)HEV

design. The set of design variables considered in this study is reported in Table 1, specifically the ICE displacement V_{ICE}, the MG peak power $P_{max,MG}$ (considering the same efficiency maps with MG working as motor and generator), a battery power-to-energy ratio PE representative of the fraction between $P_{max,MG}$ and the energy of the battery in kWh, the battery maximum C-rates during discharge $C_{dis,max}$ and charge $C_{ch,max}$, the FD speed ratio sr_{FD} and the TCD speed ratio sr_{TCD}.

Table 1. Design variables considered for a pre-transmission parallel hybrid electric vehicle.

Powertrain component	Design Variable
Internal Combustion Engine	Displacement
Motor-Generator	Peak power
Battery	Power-to-energy ratio
Battery	Maximum C-rate in discharge
Battery	Maximum C-rate in charge
Final drive	Speed ratio
Torque-coupling device	Speed ratio

For the sake of clarity, a modification of the powertrain architecture could lead to a modification to the set of design variables. For a deeper insight about the design variables considered for other parallel architectures, please refer to Appendix B. This said, a single set of performance indexes has to be defined so as that a fair comparison between the different (P)HEV layouts could be carried out. Several indexes could be used depending on the specific application, such as the tank-to-wheel CO_2 ($CO_{2,TTW}$) for HEVs ([35]), the total cost of ownership ([145]) or the

temperature of the ICE ([150]). Similarly, the driving conditions have to be identified so to guide a specific design optimization. Considering a backward-facing model of the vehicle (Section 2.1), the velocity (and road grade) trajectories could be extracted from type-approval tests rather than real-world driving missions. Once the first three steps of the procedure have been completed, a set of (P)HEV layouts have to be generated for the real optimization to begin. Therefore, the values to be assigned at each design variable of the layout have to be defined. Considering the design variables of Table 1, two scenarios might occur:

1. The values of each design variable are known for multiple layouts according to experimental data or other sources;
2. The values of at least one design variable are unknown for multiple layouts.

In the case of complete knowledge about the (P)HEV designs to be compared, the step of generating the population of (P)HEV layouts can immediately be solved. On the contrary, the problem has to be faced when the values of at least one design variable are not available. Several techniques could be used to generate a population of layouts ([151]); nevertheless, an analysis of the relative pros and drawbacks is beyond the scope of this dissertation. Within this research, a space-filling design of experiment (DoE) approach ([151]) has been employed to generate the values of the design variables for a relevant number of (P)HEV layouts. Considering the seven design variables reported in Table 1, a 7-D population of points has been generated by means of a Sobol sequence algorithm ([152]). Consistently, a set of layouts featured by design variables with DoE-based quasi-random values can hence be obtained without a manual operation. Now, an additional consideration is to be made. If the design of the vehicle is completely unknown (i.e. no results have ever been obtained about the performance of a specific (P)HEV architecture on a given driving mission) and a design optimization has to be carried out, the widest possible design domain should be analyzed. Therefore, the DoE approach could be useful in the generation of points in a large but rough space. On the contrary, smaller and finer grids could be generated in case of reduced problems with a particular focus on unknown design regions, such as those featured by a partial knowledge about the vehicle. Once the values of the design variables are generated, the sub-models of the (P)HEV components have to be adapted. To this end, a linear scaling approach has been considered for the sub-models of ICE and MG, whereas a complete resizing procedure has been developed for the battery. About the other design variables (i.e. final drive and torque-coupling device speed ratios), a change in their value did not produce a change in the model. The last stages of the design

optimization procedure can finally be achieved, in which a GO is first used to identify the optimal control policy and then the performance indexes are computed and stored for each tested layout, respectively. A final comparison can hence be carried out between the feasible layouts and one optimal design region (or more sub-optimal) could be identified.

In the following sections, the three approaches used to re-size the main powertrain components are first described and then the results of a complete design optimization operation for a heavy-duty HEV are reported.

2.3.1 Linear scaling of the Internal Combustion Engine

A scaling operation is intended to be employed when data are possessed only about a reference component and the necessity of generating new data about the same component with a different sizing arises. Such a consideration might be extended to any component of a generic driveline; nevertheless, experimental data are needed to prove the reliability of applying such a methodology.

Specifically for the ICE and considering the ICE sub-model presented in Section 2.1.1, a linear scaling operation has been applied to the FC map of a reference ICE based upon the ICE-related design variable shown in Table 1 (i.e. ICE displacement). A scaling factor for the ICE s_{ICE} is calculated according to:

$$s_{ICE} = \frac{V_{ICE}}{V_{ICE}^*} \tag{36}$$

in which V_{ICE}^* is the displacement of a reference ICE. Assuming the wide-open throttle power curve to scale consistent with s_{ICE}:

$$\frac{P_{ICE}}{P_{ICE}^*} = s_{ICE} = \frac{V_{ICE}}{V_{ICE}^*} \tag{37}$$

Now, the reference ICE power P_{ICE}^* can be calculated in any point of the ICE operating map as:

$$P_{ICE}^* = T_{ICE}^* \cdot \omega_{ICE}^* = \dot{m}_{fc}^* \cdot H_i \cdot \eta_g^* \tag{38}$$

in which $\dot{m}_{fc}^{*}$ is the actual FC based on the reference ICE map, H_i is the lower-heating value related to the fuel type (e.g. gasoline) and η_g is the ICE overall efficiency ([129]). If data are available about a realistic variation of the ICE speed range and overall efficiency, the inverse formulation of (38) could be employed to calculate the new FC of a downsized (or upsized) engine. On the contrary, the speed range and the ICE efficiency curves can be considered to be unchanged from different maps according to [153] when not enough data are available. Such an assumption is considered to be reliable only in case of restrained re-sizing operations. In fact, a linear scaling operation over extremely different sizing could introduce unrealistic approximations into the final ICE maps.

The new FC map turns into a compressed (or expanded) version of the reference map considering:

$$\dot{m}_{fc} = \dot{m}_{fc}^{*} \cdot s_{ICE} \tag{39}$$

The new ICE mass m_{ICE} and the inertia coefficient I_{ICE} can also be calculated considering the scaling factor:

$$m_{ICE} = m_{ICE}^{*} \cdot s_{ICE} \tag{40}$$

$$I_{ICE} = I_{ICE}^{*} \cdot s_{ICE} \tag{41}$$

An example of the ICE wot curve scaling for a downsized ICE in case of s_{ICE}=0.58 is reported in Figure 13.

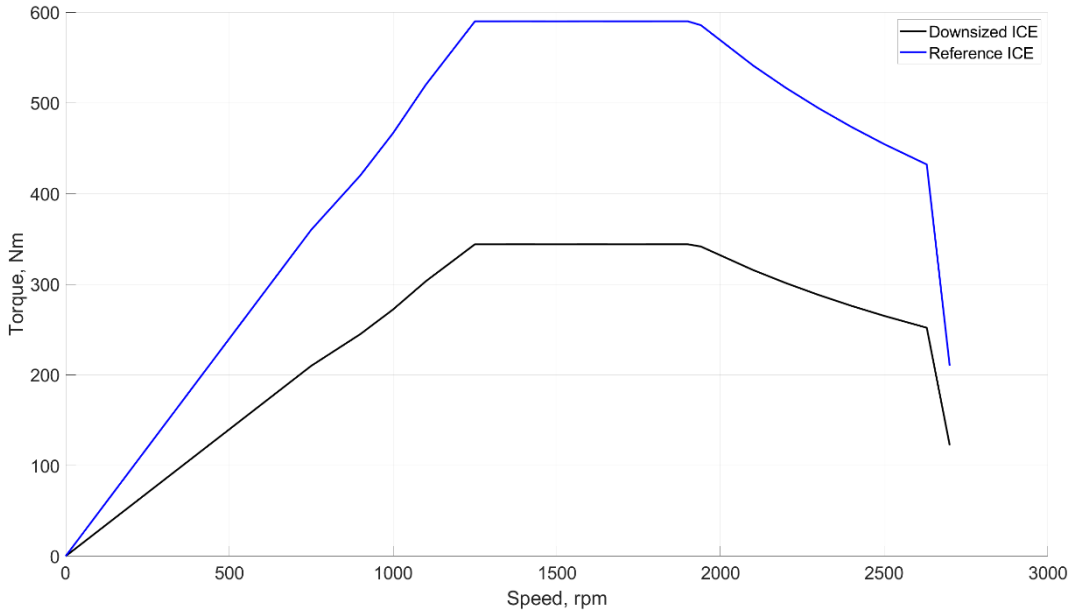

Figure 13. Example of a scaling operation over the wide-open throttle curve of an internal combustion engine.

Similar to the linear scaling operated over the ICE operating map, the MG speed range has been considered to be unchanged between new and reference MG maps due to a lack of data. Furthermore, an unchanged shape of the MG efficiency curves has been assumed. Even for the MG, the application of a linear scaling approach has not been intended for extreme MG re-sizing so as that the generation of unrealistic data could be avoided. The new MG masses and inertia coefficients can be computed through:

$$m_{MG} = m^*_{MG} \cdot s_{MG} \tag{43}$$

$$I_{MG} = I^*_{MG} \cdot s_{MG} \tag{44}$$

The result of the MG maximum and minimum torque curves obtained when a downscaling operation is performed with s_{MG}=0.85 are shown in Figure 14.

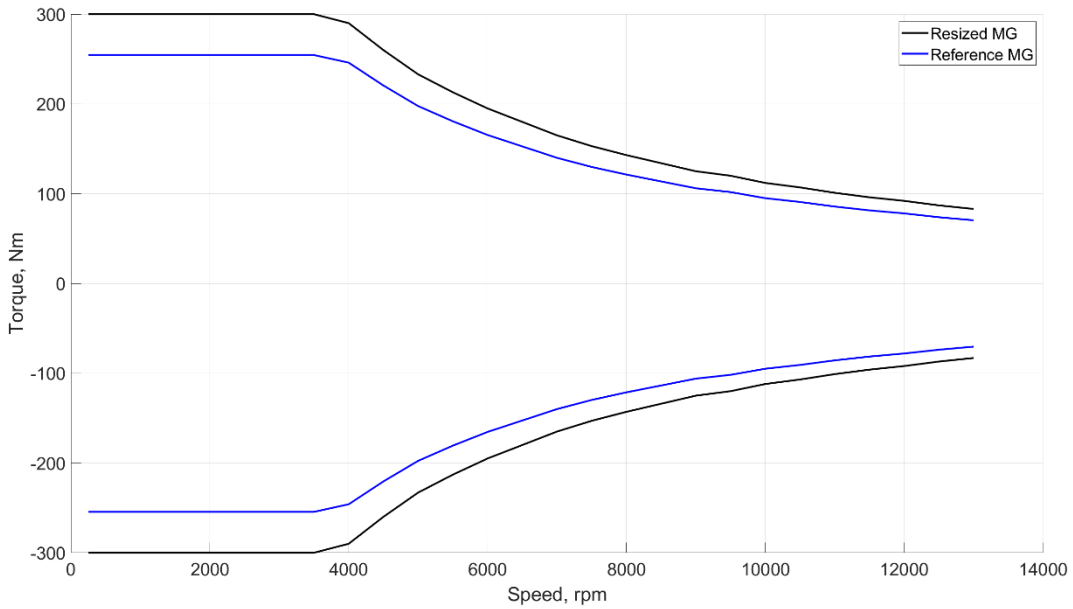

Figure 14. Example of a scaling operation over the torque limit curves of motor-generator.

2.3.3 Battery sizing

The battery sizing procedure presented in this Section refers to the Li-Ion battery architecture presented in [154]. In Figure 15, a representation of the considered battery architecture in reported. Starting from the lowest level, a set of cells in series N_c is comprised within a "unit"; then, a set of units is placed in parallel N_{pu} within a "module"; finally, a set of modules in series N_{sm} constitutes the battery. Few but necessary characteristics about a reference battery have to be available at the beginning of the sizing operation, specifically the number of cells in a unit, the cell capacity, the cell V_{oc} and the nominal voltage of the battery.

The battery sizing tool presented in [154] is hereafter detailed. The total number of cells of a generic battery N_{tot} can be calculated as:

$$N_{tot} = \frac{P_{max,MG}}{PE \cdot V_c^* \cdot C_c^*} \tag{45}$$

in which V_c^* and C_c^* represent the V_{oc} and the capacity of a reference battery cell.

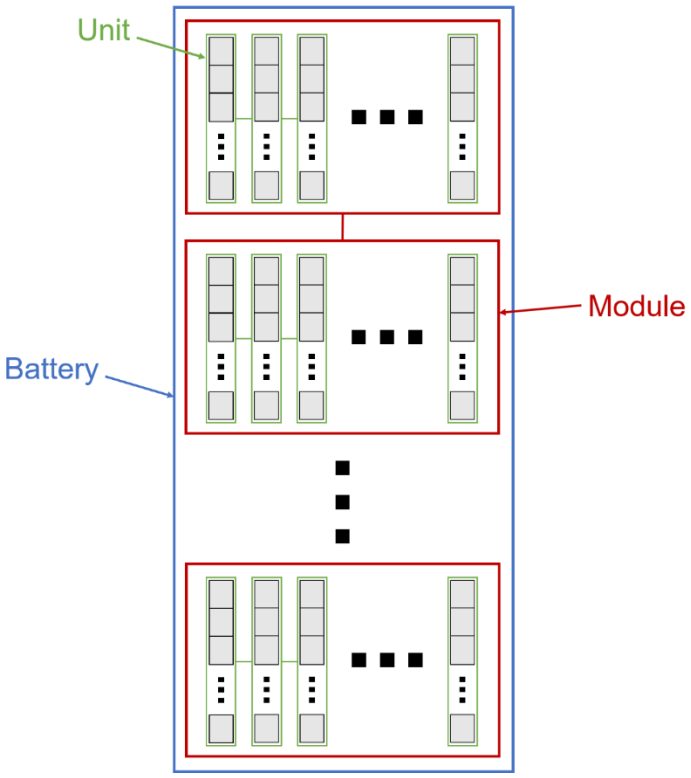

Figure 15. Architecture of the considered Li-Ion battery.

For the sake of clarity, the same type of battery cells has to be assumed for both the reference and the resized batteries. The total number of cells N_{tot} can hence be used to calculated the number of parallel units N_{pu} and series modules N_{sm} as:

$$N_{pu} = \frac{N_{tot} \cdot V_c^*}{V_{nom}{}^*} \tag{46}$$

$$N_{sm} = \frac{N_{tot}}{N_{pu} \cdot N_c{}^*} \tag{47}$$

in which $V_{nom}{}^*$ is the nominal voltage of the reference battery. The new battery capacity C_b can be computed as:

$$C_b = N_{pu} \cdot C_c^* \tag{48}$$

Moreover, a linear scaling operation over the battery mass can be conducted if the total number of cells (${N_{tot}}^*$) and the mass (${m_b}^*$) of the reference battery are available. In this case, the new battery mass m_b can be computed considering the updated number of total cells N_{tot}:

$$m_b = {m_b}^* \cdot \frac{N_{tot}}{{N_{tot}}^*} = {m_b}^* \cdot \frac{N_{pu} \cdot N_{sm}}{{N_{pu}}^* \cdot {N_{sm}}^*} \tag{49}$$

Finally, the variation of the open-circuit voltage and the internal resistance of the new battery can be calculated as:

$$V_{oc}(SOC) = N_{sm} \cdot {N_c}^* \cdot V_c^*(SOC) \tag{50}$$

$$R_i(SOC) = \frac{N_{sm} \cdot {N_c}^*}{N_{pu}} \cdot R_c^*(SOC) \tag{51}$$

in which R_c^* is the internal resistance of the reference battery cell as a function of battery SOC.

2.3.4 Results of a design optimization procedure

The setup of a complete optimization procedure over a hybrid truck are presented in this Section along with some of the most interesting results . The approach shown in Section 2.1 has been considered to model a pre-transmission parallel hybrid powertrain for a truck. The World Harmonized Vehicle Cycle (WHVC) has been identified as a proper driving mission for a design optimization in case of a heavy-duty application ([155]). The parameters of Table 1 have been employed to design the hybrid powertrain and their relative values have been modified within the optimization procedure. The variables reported in Table 2 refer to the conventional powertrain version of the electrified truck considered in this study. Considering a downsizing operation over the ICE (and its related limit power curve reduction) and the introduction of any other design parameter for a HEV version, a variation list (VL) of 5120 HEV layouts has been generated through a DoE-based Sobol sequence approach. The ranges of variation of the seven design variables are reported in Table 3. Qualitative 2-D views of a portion of the points generated for each combination of design parameters have been charted in Figure 16 (only 256 elements).

Table 2. Reference values considered for a conventional powertrain of a truck.

Variable	Value
ICE Displacement	5.4 *l*
ICE maximum Power	142 *kW*
Final Drive speed ratio	3.6

Table 3. Range of variation of the design variables considered for the design optimization procedure of a hybrid truck.

Design Variable	Range of variation
ICE Displacement	[2.4 – 4.5] *l*
Motor-Generator peak power	[90 - 300] *kW*
Battery power-to-energy ratio	[2 – 20] *kW/kWh*
Battery maximum C-rate in discharge	[2 – 12] *A/Ah*
Battery maximum C-rate in charge	[2 – 12] *A/Ah*
Final drive speed ratio	[2 – 5]
Torque-coupling device Speed ratio	[2 – 5.5]

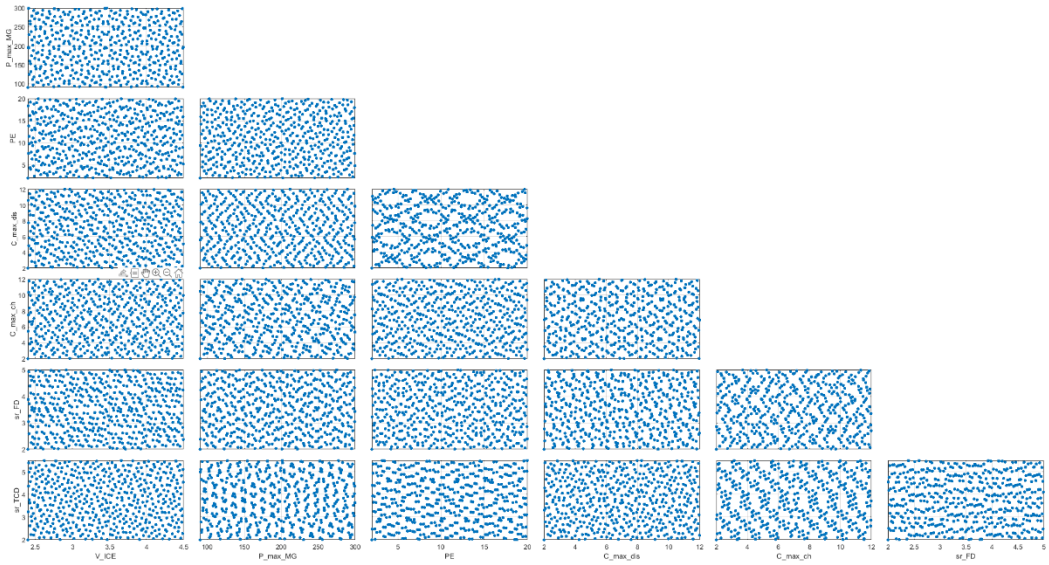

Figure 16. 2-D views of a 256-elements portion of the variation list considered for the design optimization of a hybrid truck.

As far as the optimization target is concerned, the cumulative FC at the end of the driving mission has been considered as objective functional of the problem (see (27)) and the DDP has been adopted to solve the optimization in case of a charge-sustaining trajectory. The DDP setup (i.e. number and values of the state and sub-control variables) is reported in Table 4. Specifically, 10 *bc* and 5 *ps* levels have been considered for the control variable u_1 along with a *pt* and *pe* operating modes, whereas six gears have been considered in the transmission of the vehicle as secondary control variable u_2.

As far as the state space is concerned, the battery SOC has been accounted as x_1 considering a SOC window in [0.4-0.8] for CS and a minimum SOC variation of $2e^{-4}$ (i.e. a SOC grid discretized in 2001 levels, N_{SOC}). Such a value has been selected after several DDP simulations considering a set of different SOC levels. Specifically, both the accuracy of the CO_2 final value and the computational time required for the optimization have been studied for a fixed HEV layout. The final choice has been made by empirically considering the number of SOC value which could produce realistic CO_2 emissions while limiting the computational time. Section 3 is completely devoted to a detailed dissertation about the topic of DDP grid optimization for HEVs. Finally, the ICE on-off condition has been considered as secondary state variable x_2 by means of a boolean value.

Table 4. Setup of the Dynamic Programming for the design optimization of a hybrid truck.

Control variables	
Power-flow (u_1)	$\alpha_{om,bc}$= [-2.5, -2, -1.75, -1.5, -1.25, -1, -0.75, -0.5, -0.25, -0.1] $\alpha_{om,pt}$= 0 $\alpha_{om,ps}$= [0.1, 0.25, 0.5, 0.75, 0.9] $\alpha_{om,pe}$ = 1
Gear number (u_2)	$\tau_{1,\dots,Z}$= [1, 2, …, 6]
State variables	
Battery SOC (x_1)	SOC_{min}= 0.4 SOC_{max}= 0.8 min $\dot{SOC}$= 2e^{-4}
ICE state (x_2)	$ICE_{on/off}$= [0, 1]

A reduction of the FC has been clearly expected as result of the feasible HEV layouts based upon the theoretical opportunity of exploiting CO_2-free regions under *pe* operating mode. Consistent with the linear correlation between FC and tank-to-wheel CO_2 ([156]), the latter has been considered as a meaningful metric to compare the tested HEV layouts. In Table 5, a bunch of the best and the worst feasible HEV layouts detected in the design optimization are ranked with respect to the CO_2 emissions (in g/km) attained with a DDP-based optimal control policy. Specifically, the values of the design variables (DVs, Table 5, Ref, first row) are reported for the reference conventional architecture (Ref, Table 5, second row) and for the top and worst five HEV layouts (Table 5, from third row to the last). For the sake of clarity, 2598 layouts have produced feasible control policies with respect to a design space

of 5120 layouts and hence the worst case of Table 5 refers to the 2598th layout. As far as the $CO_{2,TTW}$ is concerned, the window of emissions reduction falls in the range 16.7-2.7%. A view of the CO_2 results is reported in Figure 17 considering both the ICE displacement (upper plot) and the MG peak power (lower plot). The colors are representative of the identification number of the HEV tested layouts of Table 4; hence, they rely upon the ranking produced by the DDP. A denser design region highlights for the lower emissions while the number of points reduces as the CO_2 emissions grow. Such a trend will be important in the discussion of the results presented in the Chapter 4.

Once the optimal design regions (i.e. the values of the DVs that lead to the minimization of the optimization target) have been defined, the optimization procedure can be iteratively operated considering smaller and smaller variation ranges of the DVs (Table 3). Thanks to such a progressive modification in the design space, an accurate identification of the optimal HEV layout(s) can be performed for the studied application. Evidently, the same approach could be extended to PHEVs or to any other electrified application in which the optimal design of the vehicle can be defined through DVs and a global optimization algorithm can be implemented.

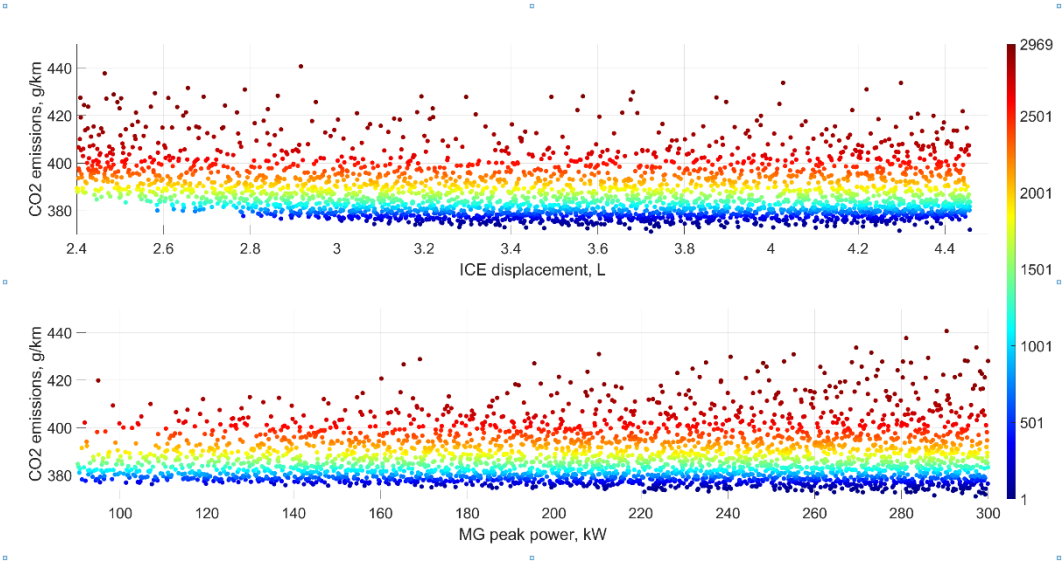

Figure 17. Minimum CO_2 emissions obtained by Dynamic Programming when applied to the hybrid layouts considered in the Variation List of the design optimization.

Table 5. Ranking of the best and worst feasible for an example design optimization of hybrid layouts based on the tank-to-wheel CO_2 emissions computed by Dynamic Programming.

DV	V_{ICE}	$P_{max,MG}$	PE	$C_{dis,max}$	$C_{ch,max}$	sr_{FD}	sr_{TCD}	$CO_{2,TTW}$
Ref.	5.4	-	-	-	-	3.6	-	445.9
1	3.7	297.0	18.0	7.8	9.6	3.2	4.4	371.2
2	4.3	287.5	18.7	6.6	10.2	3.1	4.9	371.6
3	4.4	299.2	17.8	8.5	6.3	3.0	5.1	371.9
4	3.7	260.5	17.0	8.3	8.7	3.3	4.7	372.4
5	3.6	293.9	19.8	9.4	9.2	3.1	3.7	372.5
…	…	…	…	…	…	…	…	…
2594	3.5	299.9	2.5	9.0	8.2	4.6	5.4	428.0
2595	3.6	240.6	2.0	11.0	2.8	4.8	2.2	429.8
2596	4.2	210.3	2.1	11.7	5.3	4.9	3.8	430.9
2597	4.3	269.5	2.3	8.7	10.7	4.8	3.5	433.6
2598	4.0	297.2	2.1	7.1	4.0	4.6	4.5	433.7

2.4 Conclusions

The analyses and the results presented so far have been a solid base for any other topic covered in the following sections. Specifically, some research questions have arouse:

- A fixed number of DDP state levels has been considered for each HEV layout involved in the design optimization based upon an empirical choice aimed at minimizing the simulation time. Can this approach be ameliorated? Are there any possibilities to include an automatic definition of the optimal number of DDP state levels which guarantees the accuracy of the results while drastically reducing the machine time? This topic has been covered in Chapter 3.
- A massive testing of thousands (P)HEV layouts could be necessary in case of design optimizations without information about the optimal set of DVs. Nevertheless, such an operation could become computationally too expensive when few (or even a single) modifications to the simulation setup are introduced. For instance, a comparison between the (P)HEV optimal design could be realized with respect to different hybrid architectures, driving missions, optimization targets, etc. Could an approach be developed with the capability of learning the patterns between the values of the DVs and the DDP-based CO_2 results of the feasible (P)HEV layouts? Such a question has lead to the development of a learning-based algorithm which is presented in Chapter 4.
- The optimal control policies of a DP algorithm cannot be defined during under real-time conditions due to a clear limit of the DP theoretical background. Nevertheless, an optimal solution to a control problem could be identified in offline conditions and hence be applied under real-world driving scenarios if an intelligent algorithm could be trained at reproducing the decisional process of the DP. The development of two different AI agents and the results of their applications to (P)HEVs is discussed in Chapter 5 and Chapter 6.

Chapter 3

Mesh optimization of a Deterministic Dynamic Programming algorithm for hybrid powertrains

In the present Chapter, the topic of defining a method for the identification of the proper discretization of a DDP computational grid with the capability of minimizing the simulation time is discussed. Specifically, the approach is intended to be used when a DDP algorithm is applied to the control problem of (P)HEVs considering the battery SOC as a state variable and the curse of dimensionality effect has to be limited ([154]). Evidently, excluding the possibility of using unmanageable DDP configurations could provide the user with a certainty about the identification of the optimal solution to a control problem.

Several approaches have been presented in the literature about the optimization of the computation grid of DDP algorithms, all aimed at strongly reducing the simulation times. In [157], the optimization of the computational grid has been obtained by adopting two different methods.

Part of the present section has been extracted from:

Maino, C. *et al.* (2021). Optimal mesh discretization of the dynamic programming for hybrid electric vehicles. *Applied Energy*. Elsevier Ltd, 292(March), p. 116920. Doi: 10.1016/j.apenergy.2021.116920.

The first focuses on the elimination of the nodes corresponding to redundant controls whereas the second is oriented to reduce the amount of interpolations required for the calculation of the cost function on the redundant nodes and on the nodes where the system turns out to be unfeasible. The problem of control redundancy has already been assessed for in [158]. Although the approach has been proved to be valid for dynamic programming formulations which have to manage redundant controls, the effectiveness of the method has not been demonstrated for other DP formulations in which non-redundant controls are involved. Instead, [159] proposes an iterative approach for the DDP grid refinement where the SOC range is reduced at each iteration by adopting a manually tuned calibration factor. Despite the encouraging outcomes obtained on a single HEV architecture, the consistency of the results has to be proved when different architectures are taken into account. Enhancements to the approach could hence be achieved by developing an automatic methodology for the selection of appropriate contraction factors for different HEV architectures.

To solve the limitations imposed by the abovementioned approaches, a statistical approach has been developed with the capability of determining the optimal discretization of the state space when the battery SOC is selected as one of the DDP state variables and in the condition that no redundant controls are admitted. Before the presentation of the methodology as well as of the main results obtained on various testing conditions, the vehicular application is presented along with the DDP configuration.

3.1 Vehicle model and problem formulation

Two different pre-transmission HEVs have been investigated in the present study, specifically a *pc* and an *hdv*. For each vehicle, a reference version (i.e. a conventional vehicle equipped with ICE only) has also been considered and the main specifications are summarized in Table 6. About the reference vehicles, experimentally derived ICE maps reporting FC as a function of ICE torque and speed, a fixed final drive speed ratio and a multi-gear manual transmission have been considered for both *pc* and *hdv*.

As far as the HEV powertrains are concerned, the backward-facing vehicle modeling approach presented in Chapter 2 has been employed for generating the models of the two hybrid powertrains.

Table 6. Reference vehicles main specs.

Parameter	Passenger car	Heavy-duty vehicle
ICE: type / displacement / maximum power	Diesel / 1.3 *l* / 70 *kW*	Diesel / 5 *l* / 132 *kW*
Total mass	1250 *kg*	7360 *kg*
Frontal area	2.19 m^2	6.5 m^2
Drag coefficient	0.270	0.703
Tire radius	0.3007 *m*	0.3658 *m*

In Table 7, the values assumed by the DVs considered in this study are reported for the two pre-transmission HEVs. Starting from the nominal values of Table 6, the linear downscaling presented in Section 2.3.1 has been applied to the FC map of the ICE. For each operating point, the ICE brake power of the reference map has been multiplied by the ratio between the ICE displacements of the considered HEV layout and of the conventional vehicle. ICE efficiencies and speeds have been

assumed to be remain unchanged ([153]). It is anyhow worth noticing that a variation into the scaling operation of the ICE would only turn into a possible variation of the optimal control trajectories determined by the DDP, whereas no difference in the outcomes of the methodology presented in this Chapter would be appreciated. The MG considered for the analyses belongs to the brushless permanent magnet electric motor/generator family and the relative data have been extracted from catalogues for web ([160]). When necessary, the linear MG efficiency map scaling presented in Section 2.3.2 has been carried out. Finally, the models of a Li-Ion battery presented in Section 2.1.3 and Section 2.3.3 have been considered both for the calculation of battery SOC variation and for battery design, respectively. Nevertheless, a difference in the calculation of the maximum current admitted in the battery highlights. The current flowing into a single cell I_c has been evaluated as:

$$I_c = \frac{I_b}{N_{pu}} \tag{52}$$

and the possibility of generating such a current has been verified at each time step thanks to data about the maximum admissible cell current $I_{c,max}$. Evidently, feasible control decisions have been considered only in case of $I_c < I_{c,max}$. Consistent with this check, $C_{dis,max}$ and $C_{ch,max}$ could be removed from the set of considered DVs (Table 1). In fact, the adoption of $I_{c,max}$ allows for an immediate calculation of the output/input power limit of the battery. The maximum C-rates admitted could hence be removed from the set of DVs.

Table 7. Values of the design variables considered for the hybrid powertrains of both *pc* and *hdv*.

Vehicle	V_{ICE}	$P_{max,MG}$	PE	sr_{TCD}	sr_{FD}
	l	*kW*	*kW/kWh*	-	-
pc	0.75	44	10	3.5	3
hdv	3	79.2	10	3.5	3.58

Regarding the formulation of the offline control problem, a DDP algorithm has been employed to optimize the control strategy of the HEVs under offline conditions. Specifically, the problem formulation presented in Section 2.2 has been examined, i.e. the identification of the control policy with the capability of minimizing the cumulative fuel FC while obtaining a perfect CS trajectory and satisfying the road power requests. The state and control variables considered in this study are those presented in sections 2.2.1 and 2.2.2, respectively. About the control space, the actual values assumed by each sub-control variable are reported in Table 8. A lower number of sub-control variables α_{om} has been considered with respect to Table 4. Such a choice has been taken relying upon the necessity of performing a large set of simulations with non-negligible machine times. Nevertheless, the methodology presented in this Chapter is intended to define the optimal mesh discretization of a DDP algorithm even when larger control spaces are defined.

As far as the state space is concerned, critic issues about the possibility of employing DDP can arise when the state space is expanded without a real control of the discretization levels. In the following sections, the problem is thoroughly motivated and the method for solving such a problem is discussed.

Table 8. Values of the sub-control variables considered by the Deterministic Dynamic Programming.

PFI	***pt***		***pe***	***ps***		***bc***
α_{om}	0		1	0.25; 0.5; 0.75		-0.5;-1
GNI	**1**	**2**	**3**	**4**	**5**	**6**
τ_{pc}	4.17	2.13	1.32	0.95	0.75	0.62
τ_{hdv}	6.02	3.32	2.07	1.4	1	0.79

3.2 State space: discretization of the battery state of charge

The DDP algorithm solves the optimization problem by calculating the optimal cost-to-go function, starting from the last mission time step T and going backwards until the first stage is reached (backward phase). Considering a generic time stage of the mission k, the algorithm calculates the cost-to-go functions for all the possible combinations (N_x) of state and control variables at t_k which can be computed as:

$$N_x = N_{\alpha_{om}} \times N_\tau \times N_{SOC} \times N_{ICE_{on/off}} \tag{53}$$

where $N_{\alpha_{om}}$ and N_τ represent the number of operating modes (α_{om}) and gears (τ) reported in Table 8, respectively, whereas N_{SOC} is the number of battery SOC discrete levels and $N_{ICE_{on/off}}$ is the number of engine states (i.e. on/off). In other words, N_x represents the product between the number of discrete values assumed by *cv* and *sv* to be performed at each time step of the driving mission. The overall number of computations is hence equal to:

$$N_{x,tot} = N_x \times T \tag{54}$$

This said, the values of $N_{\alpha_{om}}$, N_τ and $N_{ICE_{on/off}}$ are fixed to the specific setup of α_{om}, τ and $ICE_{on/off}$ considered for the final application. As an example, the number of gears cannot be different from six if a 6-gears transmission is included in the study. From a more general perspective, a modification of the values considered in the control space could allow the DDP for the identification of a different control policy without affecting the accuracy of the calculations. On the contrary, the selection of an inappropriate value of N_{SOC} could lead to unrealistic calculations. The motivation lies in the classic workflow of the DDP.

Considering the set of *sv*, the engine state (ES) is a Boolean number connected to the value of α_{om} (e.g. ES would be null for α_{om}=1 whereas it would be equal to 1 if α_{om}=0), whereas the battery SOC values are estimated by the battery model based upon the battery power request. Hence, the battery SOC estimates for t_{k+1} are not necessarily equal to the SOC values represented on the DDP grid and an approximation to the closest node has to be made. A coarser battery SOC grid induces larger approximations, and, consequently, larger numerical errors on the

simulation results. On the other hand, if a very fine battery SOC grid is considered, N_x could dramatically explode ([53], [73] and [161]) leading to unrealizable simulation times. A trade-off between the battery SOC variations (ΔSOC) calculated at each time step and the simulation times has hence to be defined.

At this stage, it is worthwhile recalling that ΔSOC computed between two generic time steps is directly linked to the battery energy variation (ΔE_{bat}) in the same time-interval as follows:

$$\Delta E_{bat} = \Delta SOC \cdot V_c^* \cdot C_c^* \cdot N_{tot} = \Delta SOC \cdot \frac{P_{max,MG}}{PE} \tag{55}$$

Therefore, the proper number of battery SOC levels to be set for the DPP backward phase depends upon the ΔSOC (or ΔE_{bat}) variations realized through the mission, which in turns depend upon time step size, driving mission characteristics, control state, vehicle layout and architecture. A robust and accurate methodology to select the proper SOC "grid" for DDP estimation is of fundamental importance to obtain an unbiased comparison of performance for different HEVs along different driving missions.

The proposed methodology for determining the battery SOC discretization level of the DDP grid which guarantees the optimal trade-off between the accuracy of the DDP calculations and the computational effort relies on a three-step procedure:

- Estimation of the battery energy variation for each mission time step;
- Application of a statistical analysis on the estimated battery energy variations;
- Definition of the SOC grid optimal size.

3.3 Battery energy variations

The first step of the procedure for the determination of the proper number of battery SOC levels for a DDP algorithm consists in the estimation of the expected battery energy throughout the considered mission for a given layout, architecture and set of control variables. Since the optimal control strategy is not known at this stage of the process, the battery electric power can be evaluated thanks to the results of the simulations already indexed and stored in the Configuration Matrix described in [56] and [57]. Starting from the vehicle velocity and applying the equations of the vehicle dynamics, the power required by any powertrain component for each time step and combination of *cv* can be obtained. Therefore, once the unrealizable powers demands are discarded (i.e. a specific combination of PFI and GNI that leads to a cell current higher than admissible values, see (52)), the battery energy variations are estimated by numerically integrating the battery power occurring for any feasible condition.

3.4 Mesh optimal size

Once the mapping of the battery energy variations has been completed, the frequency distribution of the estimated values can be worked out. Figure 18 reports the frequency distribution of the energy variations and shows a definitely non-Gaussian distribution. Clearly, any inference method ([162]) can be taken into account at this stage. By the way, a statistical manipulation of the data through distribution transformations can lead to the generation of a new Gaussian distribution. Once the transformed is performed, the results of any parametric approach can be processed and hence sent back to the original domain through the inverse transformation. To do so, an iterative multi-step algorithm is hereafter proposed.

Assuming the "skewness" s to be the index of a distribution asymmetry ([163]), a Gaussian shape (perfectly symmetric) would produce a null skewness value according to:

$$s = \frac{m_3}{m_2\sqrt{m_2}} \quad with \begin{cases} m_3 = \dfrac{\sum(x_i - \bar{x})^3}{n} \\ m_2 = \dfrac{\sum(x_i - \bar{x})^2}{n} \end{cases} \tag{56}$$

in which $\bar{x}$ is the mean value of the distribution, x_i is the actual value of the i-th sample and n is the number of values of the distribution. Indeed, negative skewness indices are representative of a negative asymmetry whereas positive ones stem when a positive asymmetry occurs ([164]). A slightly asymmetric distribution ($s \approx 0$) can be studied as a Gaussian one only if the following criterion is satisfied ([164]):

$$\left| \frac{s}{\sqrt{6/n}} \right| < Z_a \tag{57}$$

where Z_a represents a Gaussian distribution standardized variable referred to the tolerance level a ([165]). In this study, the results obtained when a is set to 95% ($Z_a = 1.96$), 90% ($Z_a = 1.65$) and 68% ($Z_a = 1$) are presented.

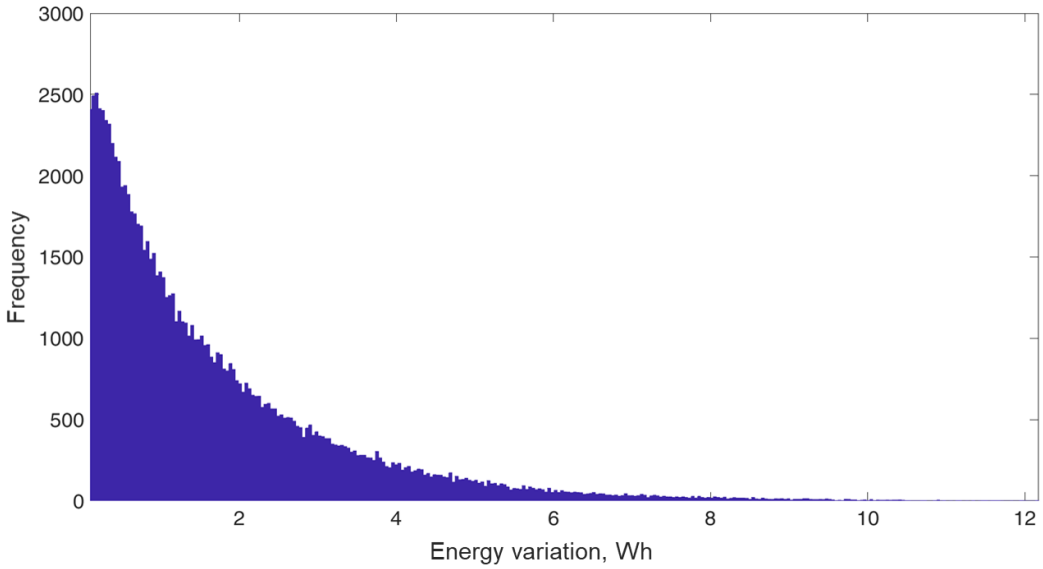

Figure 18. Distribution of the battery energy variations obtained on the World harmonized Light-duty Test Cycle with *pc*.

The logic of the proposed procedure is hereafter outlined:

1. Once the mapping of the battery energy variations is available (see Section 3.3) and the related frequency distribution is generated, the skewness index is computed through (56) and the condition of (57) is checked upon.
2. If the condition of (57) is verified, the procedure jumps to points 6, otherwise it continues through point 3.
3. According to the skewness value, a negative or positive asymmetry is found and a transformation to the distribution is performed as follows:

$$\begin{cases} x_{tras} = x^{m} & with\ m > 1\ and\ s < 0 \\ x_{tras} = x^{\frac{1}{m}} & with\ m > 1\ and\ s > 1 \end{cases} \quad (58)$$

in which m is a coefficient to be tuned during the iterations progress. In this study, the initial value has been set to 1.1 and it has been gradually increased along the iterations.

4. The skewness index of the transformed distribution is computed and the condition of (57) is once more checked.
5. If the condition of (57) is verified, the procedure continues to Point 6, otherwise m is changed and the Point 3 is repeated. In this study, m has been incremented by 0.1 iteration after iteration.
6. The mean value μ and the standard deviation σ of the obtained transformed distribution (almost symmetric, see example in Figure 19, where the transformed shape of the distribution depicted in Figure 18 is shown) are computed and the tolerance level a is selected together with its relative Z_α. The confidence interval c of the transformed distribution is estimated through the following parametric approach:

$$c = [\mu - Z_\alpha \sigma \, , \mu + Z_\alpha \sigma \,] \tag{59}$$

7. The lower extreme of the transformed confidence interval is chosen as a statistically consistent minimum energy variation within the possessed batch of data and is identified in the original domain thanks to the inverse transformation of the distribution.

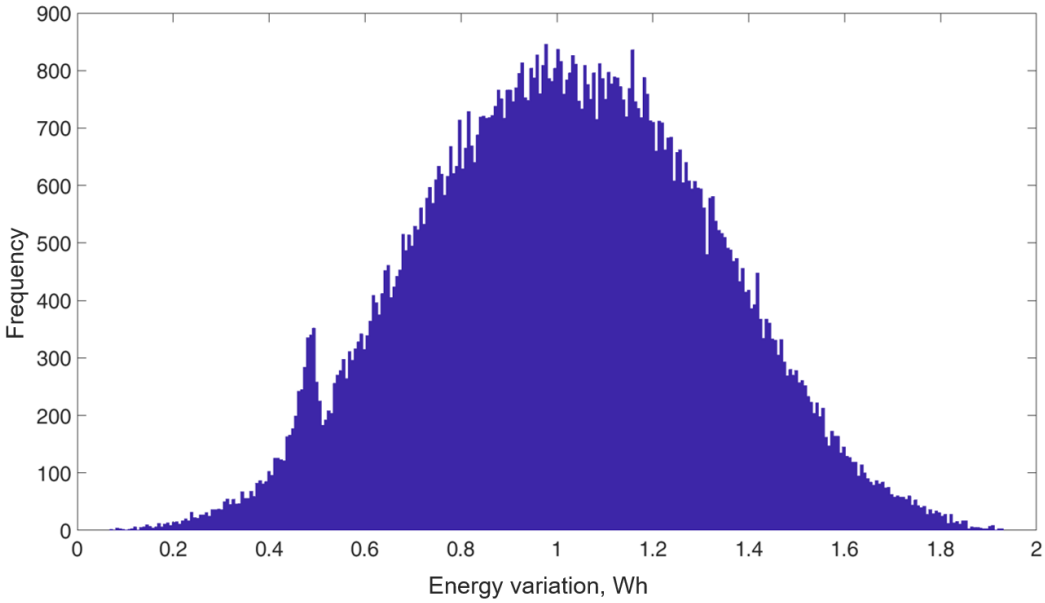

Figure 19. Transformed distribution of the battery energy variations obtained on the World harmonized Light-duty Test Cycle with *pc*.

From a more practical point of view, the usage of $a = 95\%$ means that the 5% of the distribution tails (2.5% per tail) of the energy variation transformed distribution are discarded. Therefore, the minimum energy step Δh identified on the transformed shape will be the one immediately after the after the 2.5% left-handed threshold. The same consideration can be made for a equal to 90% and 68%. The output of the procedure is hence exploited to set the value of the energy step Δh that is used to define the SOC discretization grid. It is worth observing that the method is devoted to describe the refinement of the battery SOC grid considering a fraction of the battery SOC window typically exploited during HEV operations. The selection and optimization of the SOC ranges are beyond the scope of the present research and their actual values have not to be considered as a potential issue for the proposed methodology.

The number of battery SOC discretization levels (N_{SOC}) to be set in the DDP configuration is finally computed through:

$$N_{SOC} = int\left(\frac{\Delta E_{SOC,win}}{\Delta h}\right) = int\left[(SOC_{max} - SOC_{min})\frac{P_{max,MG}}{\Delta h \cdot PE}\right] \tag{60}$$

in which $\Delta E_{SOC,win}$ is the portion of battery energy associated to the employable battery SOC range (from SOC_{min} to SOC_{max}). In the following sections, interesting results are provided with respect to the application of the proposed statistical procedure when extremely different test cases are taken into account.

3.5 Results

Three indices are used to evaluate the performances of the vehicles on a given driving mission: $CO_{2,TTW}$, total cost of ownership (TCO) and simulation time (or machine time). The tank-to-wheel CO_2 emissions are directly connected to the fuel consumption at the end of the driving mission considered the SOC-constrained min(FC)-based control policy identified by the DDP. Indeed, the TCO is obtained by summing the operating and fixed costs ([35]) of the vehicle for a 10-years' time span and is used to obtain an indication about the long-term economic advantages brought by a hybridization solution. As additional metric, the simulation time has been selected to assess for the computational effort required by the optimization tool.

A distinct set of driving missions has been employed for each vehicle segment addressed in this study. The driving mission set for *pc* is made up of Worldwide harmonized Light vehicles Test Cycle (WLTC), Artemis Urban Driving Cycle (AUDC), Artemis Rural Driving Cycle (ARDC) and Artemis Motorway Driving Cycle (AMDC). World Harmonized Vehicle Cycle (WHVC), City Suburban Cycle (CSC), Heavy Heavy-Duty Diesel Truck Transient mode (HHDDTT) and European Transient Cycle (ETC) have been selected for *hdv*. In Table 9, the main characteristics of the different missions are summarized in terms of average and maximum vehicle velocity, vehicle acceleration and power required for the traction. Moving from the third column (Table 9, left-hand side), T is the mission duration, D is the distance covered, $\bar{v}$ is the vehicle mean velocity, v_{max} is the vehicle maximum velocity, $a_{max,trac}$ is the maximum acceleration required during traction phases while $P_{max,trac}$ and $\bar{P}_{trac}$ are the maximum and average power required to the vehicle during traction phases, respectively. For the sake of clarity, the parameters of Table 9 going from third to sixth column (i.e. mission duration to vehicle maximum velocity) have been extracted from [166], whereas the remaining ones have been estimated for the two HEVs based on their vehicle specifications.

The effect of SOC discretization on the values of $CO_{2,TTW}$ and TCO achieved with DDP optimal control policy has been investigated. To this end, sweeps of the number of battery SOC levels have been carried out considering a range from 10^2 to 10^5 for both the simulated vehicles. The lowest threshold of the range (10^2) has been identified based on the minimum number of battery SOC levels allowing for a feasible simulation (i.e. the minimum number of SOC levels for which the DDP can still identify an optimal control policy).

Table 9. Main characteristics of the driving cycles used for testing *pc* and *hdv*.

Vehicle	Cycle	T	D	$\bar{v}$	v_{max}	$a_{max,trac}$	$P_{max,trac}$	$\bar{P}_{trac}$
		s	*km*	*km/h*	*km/h*	m/s^2	*kW*	*kW*
pc	WLTC	1800	23.26	46.49	131.3	1.75	36.23	9.19
	AUDC	993	4.87	17.63	57.7	2.86	26.65	6.51
	ARDC	1082	17.27	57.41	111.5	2.36	41.88	9.69
	AMDC	1068	29.54	99.49	150.4	1.91	59.51	19.85
hdv	WHVC	1800	20.07	40.12	87.8	1.59	116.2	34.66
	ETC	1800	29.47	58.94	91.1	3.83	120.55	39.11
	CSC	1700	10.74	22.73	70.49	1.14	89.99	24.86
	HHDDTT	667	4.58	24.72	76.42	1.29	116	21.69

On the other hand, the highest value (10^5) is representative of a battery energy variation that can be realistically realized in the batteries of the two vehicles during a single time-step. As an example, for a calculation time-step of one second, 10^5 SOC levels lead to ~0.015 *Wh* and ~0.03 *Wh* for *pc* and *hdv*, respectively. Therefore, the energy steps associated to 10^5 can be considered sufficiently small considering the typical energetic consumptions of a vehicle in a second. Furthermore, a maximum admissible machine time had to be identified. Such a value has been set to 2 hours for a single simulation. Consistently, maximum simulation times in the order of 90 minutes for *pc* and 60 minutes for *hdv* will be shown in the next charts.

In Figure 20, $CO_{2,TTW}$ (top chart), TCO (middle chart) and computational time (bottom chart) are charted as a function of the number of battery SOC levels for the investigated *pc*. A logarithmic scale has been employed for the battery SOC axis. The $CO_{2,TTW}$ and TCO increase as SOC level number increases until a knee is reached at about 10^3 SOC levels. From that stage onwards, their values remain almost constant even if the number of SOC level increases. On the other hand, the computational time exhibits a significant increase once the knee is overcome. Remarkably, the trend of the TCO is aligned with that of the $CO_{2,TTW}$. As a matter

of fact, the vehicle operating costs vary as the carbon dioxide emissions change whereas the fixed costs remain unchanged. Still, comparing the final emissions and TCO over the different missions, the AMDC shows worse performances if compared to the AUDC and ARDC. The type-approval WLTC cycle holds intermediate values for both $CO_{2,TTW}$ and TCO. From an optimization perspective, a realistic estimation of the $CO_{2,TTW}$ has to be realized, especially when regulatory driving missions are taken into account. This said, Figure 20 proves that an ineffective mesh refinement could lead to non-negligible errors (tens of grams per kilometre) in the estimation of the $CO_{2,TTW}$ for the WLTC. Unrealistic or optimistic results have to be clearly avoided as the compliancy to the regulation targets has to be assessed for ([24], [27]).

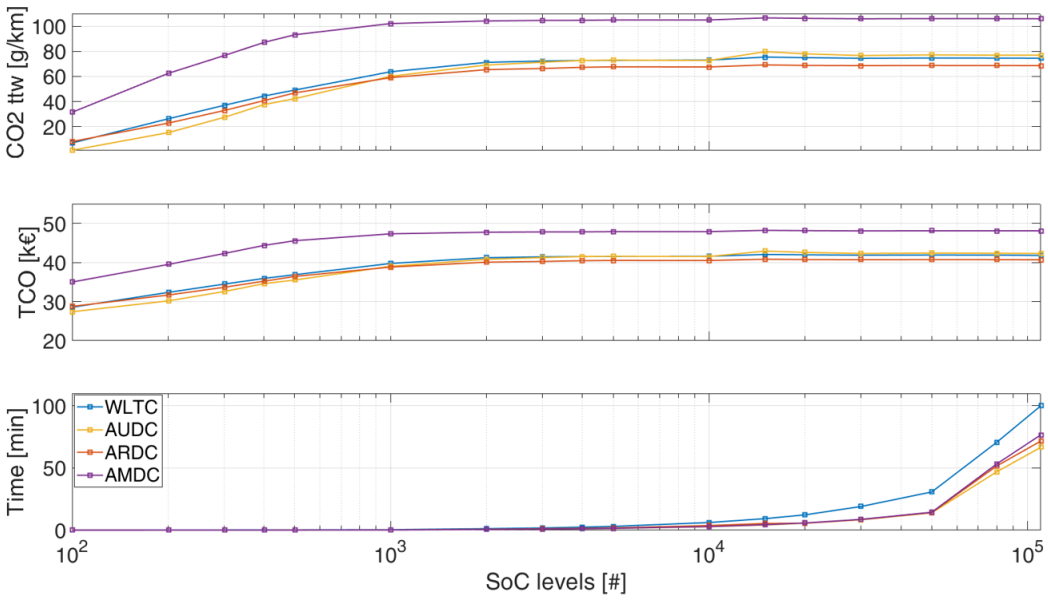

Figure 20. Results of the sweep operation over the number of battery SOC levels for *pc*.

As far as *hdv* is concerned, similar results to those obtained for *pc* are shown in Figure 21: $CO_{2,TTW}$ and TCO reach a constant value after the knee is reached (in this case at a number of SOC levels equal to about $2x10^3$).

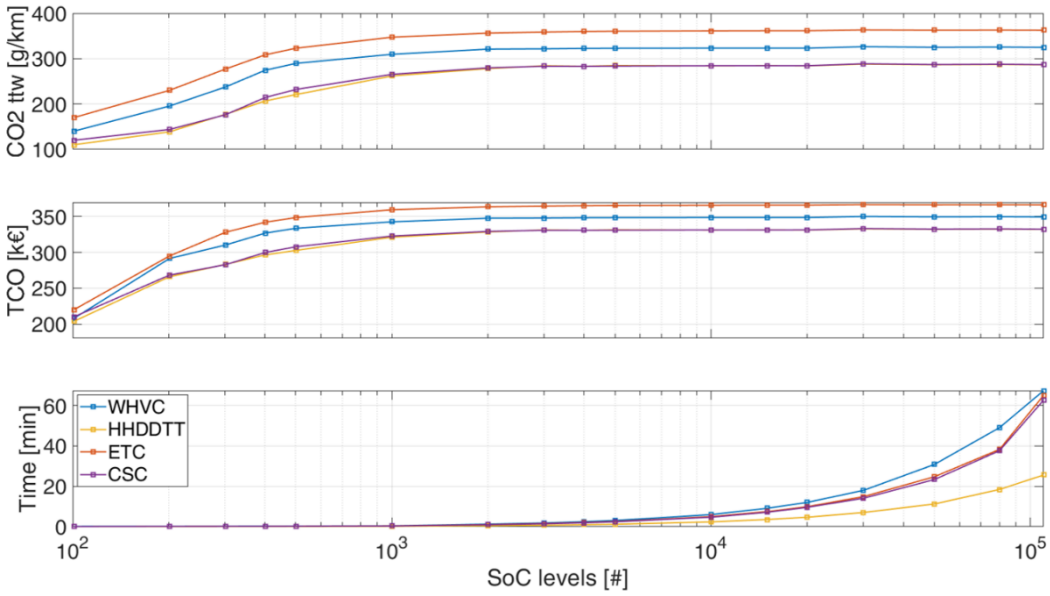

Figure 21. Results of the sweep operation over the number of battery SOC levels for *hdv*.

3.5.1 Outcomes for $N_{SOC,95}$

The results obtained when the statistical procedure presented in Section 3.4 with the tolerance level a equal to 95% is applied to the two test cases are hereafter shown. First, the energy variations and their frequency distributions are pre-processed for each driving scenario. Then, the statistical procedure is fully exploited and the minimum energy steps are found to identify the number of SOC levels to be used in the simulation. In Figure 22 and Figure 23, the results obtained when Z_α equal to 1.96 is used to define the confidence interval c (Section 3.4) are presented. In the charts, a circular marker represents the number of battery SOC levels evaluated at the end of the procedure for each driving mission. The performance indices obtained by the DDP fall on the stable portion of the sweep curve regardless the driving mission. Still, N_{SOC} never matches with the amount of battery SOC levels corresponding to the lowest energy step admissible (final abscissa of the curves): the assumption of a=95% is to be reliably considered as the most conservative configuration for the entire set of test cases. In Table 10, the performance indices are presented when the highest admissible number of battery SOC levels (r, fourth column) and N_{SOC} (fifth column) are accounted into the DDP

configuration; in the last column, the percentage difference between the two conditions $e_{rel,95}$ (%) exhibits for each observed variable and it is evaluated according to:

$$e_{rel,95} = \frac{x_{i,N_{SOC}} - x_{i,r}}{x_{i,r}} \cdot 100 \;\; with \; i = 1,2,3 \tag{61}$$

in which $x_{i,N_{SOC}}$ represents i-th variable estimated for the DDP optimal control policy when the number of SOC levels outputted by the statistical approach is used while $x_{i,r}$ represents the i-th variable obtained when the DDP is characterized by the maximum admissible SOC levels. Therefore, as far as the $CO_{2,ttw}$ (i=1) and the TCO (i=2) are concerned, $e_{rel,95}$ is to be considered as an approximation error whereas it should be accounted as a benefit for the simulation time (i=3). Promising results arise for both vehicles: a very slight approximation is introduced into the $CO_{2,ttw}$ and the TCO (lower than 1% in absolute value) whereas important reductions of simulation time are achieved. Referring to *pc* (Figure 22), the lower amount of SOC levels is identified on the AMDC (machine time $e_{rel,95}$~93% whereas the higher is related to the AUDC ($e_{rel,95}$~43%). As far as *hdv* is concerned (Figure 23), the rougher refinement can be exploited for the ETC ($e_{rel,95}$~81%) whereas the finer is on CSC ($e_{rel,95}$~9%).

The application of the statistical approach definitely allowed for achieving very accurate results in terms of emissions and costs while saving a relevant amount of simulation time for both *pc* and *hdv*.

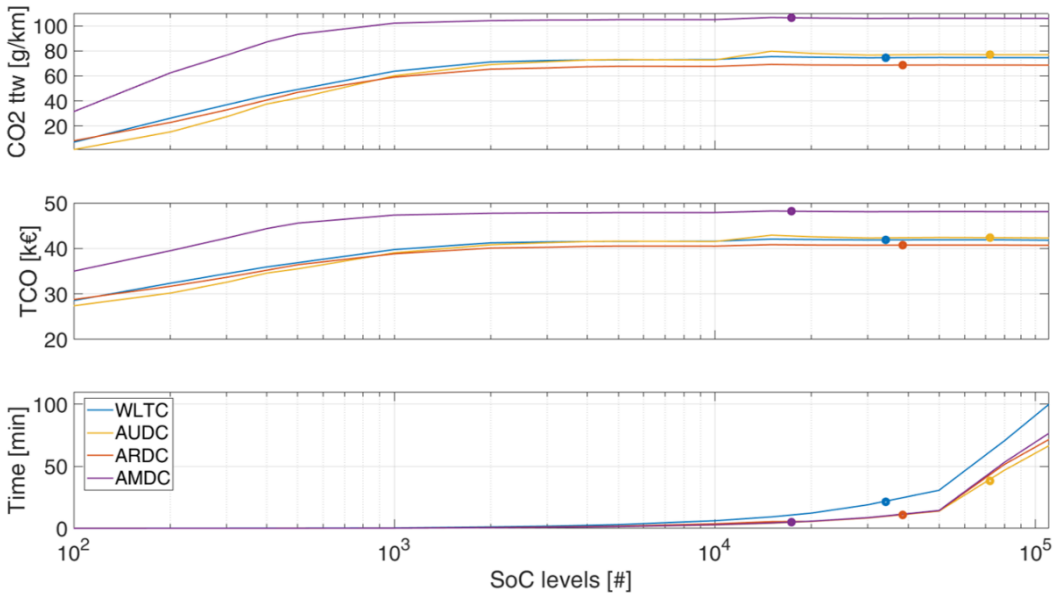

Figure 22. Positioning of the results obtained by the DDP on the sweep curves when $N_{SOC,95}$ is used for *pc*.

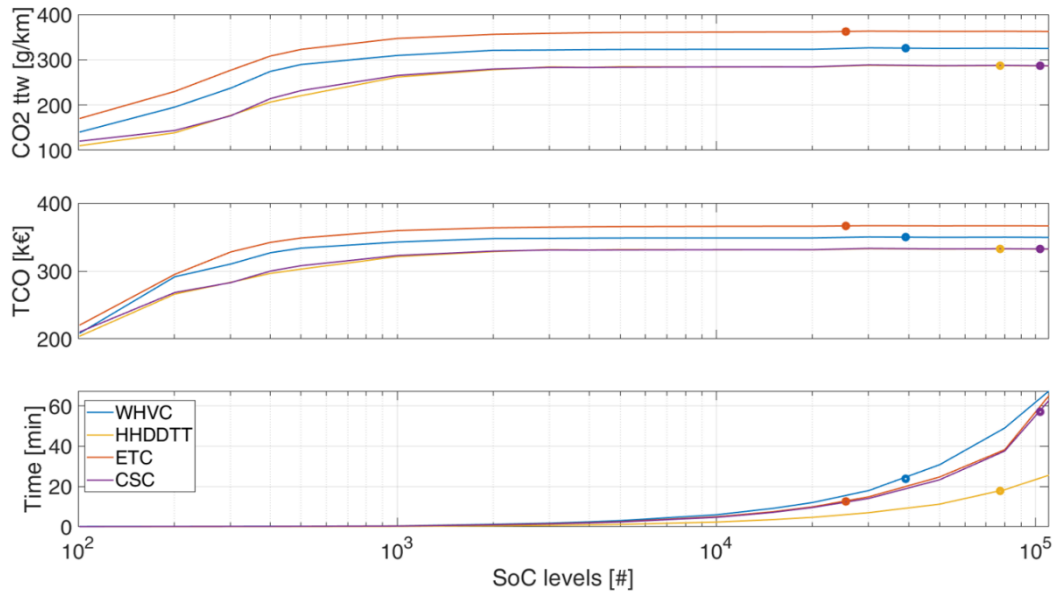

Figure 23. Positioning of the results obtained by the DDP on the sweep curves when $N_{SOC,95}$ is used for *hdv*.

Table 10. Results obtained by the DDP for each driving mission when the highest admissible SOC levels r and $Nsoc_{,95}$ are used for *pc* and *hdv*.

Vehicle	Index	Cycle	r	$N_{SOC,95}$	$e_{rel,95}$ [%]
pc	$CO_{2,ttw}$ *g/km*	WLTC	74.5	74.5	-0.01
		AUDC	76.9	77.0	+0.13
		ARDC	68.6	68.6	+0.06
		AMDC	105.9	106.4	+0.49
	TCO *k€*	WLTC	41.8	41.9	+0.17
		AUDC	42.3	42.4	+0.17
		ARDC	40.7	40.7	+0.07
		AMDC	48.1	48.2	+0.19
	Machine time *min*	WLTC	100.0	21.5	-78.53
		AUDC	66.7	38.3	-42.57
		ARDC	71.7	14.0	-80.41
		AMDC	76.7	5.1	-93.37
hdv	$CO_{2,ttw}$ *g/km*	WHVC	324.8	325.6	+0.27
		HHDDTT	286.3	287.0	+0.24
		CSC	286.6	286.8	+0.07
		ETC	362.5	362.4	-0.05
	TCO *k€*	WHVC	349.5	349.9	+0.11
		HHDDTT	332.2	332.6	+0.09
		CSC	332.4	332.5	+0.02
		ETC	366.5	366.4	-0.02
	Machine time *min*	WHVC	67.3	23.8	-64.60
		HHDDTT	25.6	17.8	-30.44
		CSC	62.6	57.1	-8.92
		ETC	65.0	12.6	-80.60

3.5.2 Definition of the optimal mesh refinement

The promising results highlighted in the previous Section have pushed the research to the identification of different settings for the statistical approach with the potential of shifting the discretization levels down to the first portion of the sweep curves in terms of trade-off between accuracy and simulation time saving. To this end, three different tolerance levels have been imposed as hyper-parameters of the statistical approach (Section 3.4). The amount of SOC levels $N_{SOC,95}$, $N_{SOC,90}$ and

$N_{SOC,68}$ (a equal to 95%, 90% and 68%, respectively) are represented in the following charts by circular, rhombic and squared markers, respectively. In Figure 24, a comparison of the results obtained for *pc* are charted. The amount of battery SOC levels identified by the statistical procedure is gradually reduced as the tolerance interval percentage reduces. Specifically, the squared markers for a=68% are positioned in an optimal location given their proximity to the end of the sweep curves knee: such a location could lead to a very large reduction of the simulation times with the introduction of a slight approximation into the DDP results. As from Table 11, an acceptance of a $CO_{2,TTW}$ approximation error ranging between ~2% and ~5% could provide simulation time reductions in the order of 97-99%. If such an accuracy is not satisfactory, $N_{SOC,90}$ can be taken into account given that an accuracy loss around 1% is introduced as simulation time reductions ranging between ~89% and ~97% are guaranteed. Consistent with the trend of $CO_{2,TTW}$, the approximation error over the TCO increases as N_{SOC} decreases.

A very similar behaviour is shown for the *hdv* test case (Figure 25). If a is lowered, N_{SOC} is reduced without introducing relevant accuracy errors (it never falls on the sweep curves knee). Moreover, as for the *pc*, a=68% allows for achieving an optimal trade-off between DDP accuracy and simulation times abatement. Promising results are shown when $e_{rel.68}$ is taken into account (Table 11). The worst case occurs for the ETC driving mission in which an accuracy loss around 2% is introduced into the $CO_{2,TTW}$ estimation, whereas the best case is found for the HHDDTT as the approximation error is limited to ~0.5%. On the other hand, the machine times are once again cut in a range between ~95% and ~98%. No strange behaviours stem for the TCO estimation as very restrained approximation errors lower than 1% arise for each testing mission.

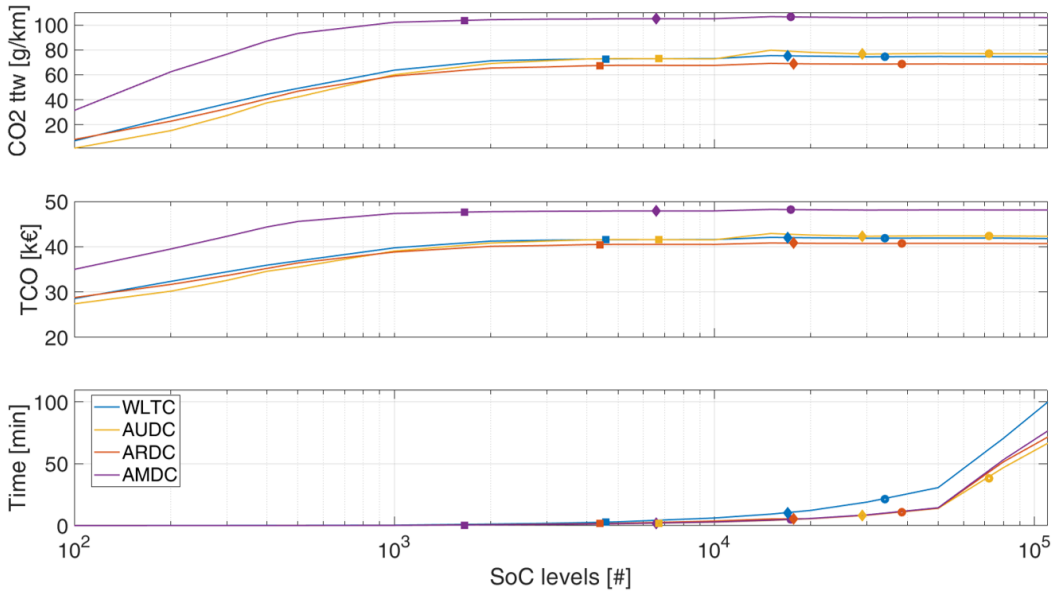

Figure 24. Comparison of the different positioning of the results obtained by the DDP on the sweep curves when $N_{SOC,95}$, $N_{SOC,90}$ and $N_{SOC,68}$ are used for *pc*.

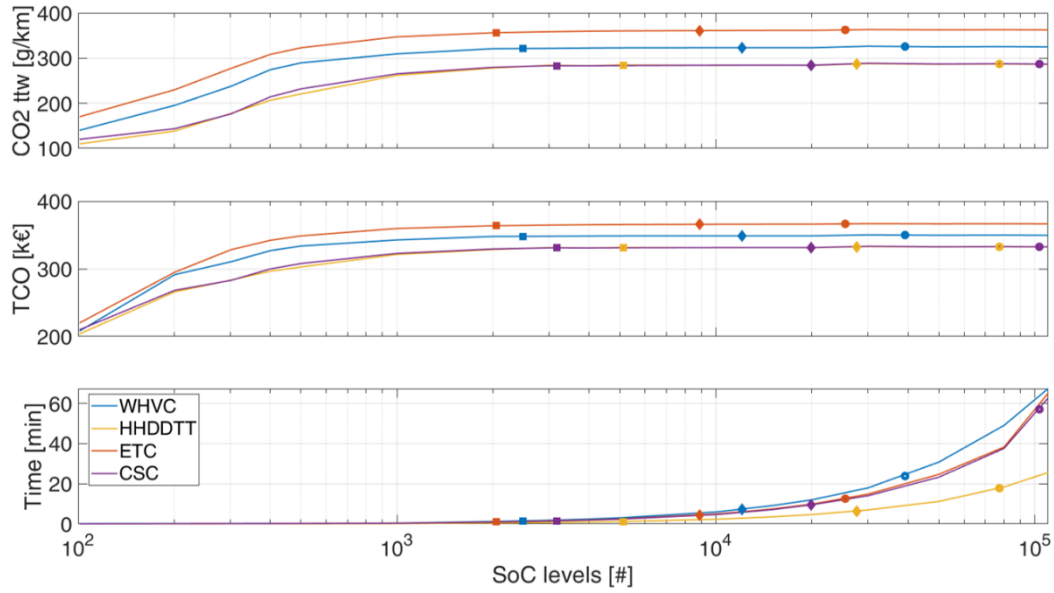

Figure 25. Comparison of the different positioning of the results obtained by the DDP on the sweep curves when $N_{SOC,95}$, $N_{SOC,90}$ and $N_{SOC,68}$ are used for *hdv*.

Table 11. Results obtained by the DDP for each driving mission when $N_{SOC,95}$, $N_{SOC,90}$ and $N_{SOC,68}$ are used for *pc* and *hdv*.

Vehicle	Index	Cycle	$N_{SOC,95}$	$N_{SOC,90}$	$N_{SOC,68}$	$e_{rel,95}$ [%]	$e_{rel,90}$ [%]	$e_{rel,68}$ [%]
pc	$CO_{2,ttw}$ *g/km*	WLTC	74.5	75.2	72.8	-0.01	+0.97	-2.34
		AUDC	77.0	76.8	72. 9	+0.13	-0.16	-5.20
		ARDC	68.6	68.9	67.4	+0.06	+0.50	-1.75
		AMDC	106.4	104.9	103.4	+0.49	-0.90	-2.31
	TCO *k€*	WLTC	41.9	42.0	41.5	+0.17	+0.53	-0.65
		AUDC	42.4	42.3	41.6	+0.17	+0.05	-1.75
		ARDC	40.7	40.8	40.5	+0.07	+0.22	-0.54
		AMDC	48.2	47.9	47.6	+0.19	-0.42	-1.04
	Machine time *min*	WLTC	21.5	10.5	2.8	-78.53	-89.49	-97.16
		AUDC	38.3	8.2	1.9	-42.57	-87.76	-97.12
		ARDC	14.0	6.4	1.7	-80.41	-91.07	-97.67
		AMDC	5.1	1.9	0.5	-93.37	-97.46	-99.37
hdv	$CO_{2,ttw}$ *g/km*	WHVC	325.6	322.9	321.1	+0.27	-0.55	-1.11
		HHDDTT	287.0	286.8	284.6	+0.24	+0.21	-0.59
		CSC	286.8	284.2	283.2	+0.07	-0.82	-1.21
		ETC	362.4	360.8	356.3	-0.05	-0.47	-1.72
	TCO *k€*	WHVC	349.9	348.7	347.9	+0.11	-0.23	-0.46
		HHDDTT	332.6	332.5	331.5	+0.09	+0.08	-0.23
		CSC	332.5	331.3	330.8	+0.02	-0.32	-0.47
		ETC	366.4	365.7	363.7	-0.02	-0.20	-0.76
	Machine time *min*	WHVC	23.8	7.3	1.5	-64.60	-89.22	-97.71
		HHDDTT	17.8	6.5	1.2	-30.44	-74.76	-95.15
		CSC	57.1	9.5	1.5	-8.92	-84.79	-97.54
		ETC	12.6	4.4	1.1	-80.60	-93.17	-98.34

3.5.3 Effect of the number of sub-control variables

The robustness of the method was also assessed for by varying the number of sub-control variables used by the DDP to search for the optimal control policy. For the sake of simplicity, *pc* has been selected as only test case of the analysis. Regarding the GN, the number of τ values is fixed to the number of gears of the considered transmission. On the other hand, a change into the number of PFs could be managed through the modification of the values of α_{om}. Specifically, the number of *ps* and *bc* levels might be increased or reduced based on the necessity of simulating larger or lower sets of admitted actions. The values assigned to α_{om} are listed in Table 12 considering three different cases, namely s_1, s_2 and s_3. The battery energy variations produced in the three different cases can be different, as demonstrated by the maps of Figure 26. As it can be noted, the change of the number of α_{om} levels leads to a reshape of the distribution of the power demanded to the battery. Consistent with an increased number of *ps* and *bc* levels, finer and finer battery energy variations can hence be observed as the number of PFIs is enlarged.

The setup of a different set of sub-control variable could reasonably lead to a modification of the optimal control policy obtained with DDP. The different battery SOC profiles related to the optimal control policies identified by the DDP in case of s_1, s_2 and s_3 have hence been studied. In Figure 27, the battery SOC trajectories are charted considering the highest amount of battery SOC levels admitted in the study (r). Different battery SOC trajectories are clearly produced by the DDP. The orange trajectory (s_3, 16 PFIs) exploits a smaller amount of the SOC window, whereas a larger amount of charge is used as the values of α_{om} are reduced to 7 and 4. The trends of Figure 27 proves that a change into the number of sub-control variables could introduce non-negligible variations into the results of the DDP. Therefore, the reliability of the statistical approach performances when a different set of control actions is used has to be ensured as a final analysis. In other words, the number of battery SOC levels identified at the end of the procedure does not have to introduce any issue into the results accuracy.

In Figure 28, the sweep curves realized with s_1, s_2 and s_3, are plotted for each performance index when *pc* is tested on the WLTC. Coherent to the comparison presented in Section 3.5.2, the results related to $N_{SOC,95}$ $N_{SOC,90}$ and $N_{SOC,68}$ are reported through circular, rhombic and squared markers. Despite the different sets of sub-control variables, the $CO_{2,TTW}$ and TCO estimates converge roughly on the same values while the calculation times significantly increase as the number of control variables is augmented (Figure 28).

Table 12. Three different sets of values for the discretization of the sub-control variable of the power-flow.

PFI	Set	*pt*	*pe*	*ps*	*bc*
α_{om}	s_1	0	1	0.5	-0.5
	s_2	0	1	[0.25, 0.5, 0.75]	[-0.5, -1]
	s_3	0	1	[0.1:0.1:0.9]	[-0.5:-0.5:-2.5]

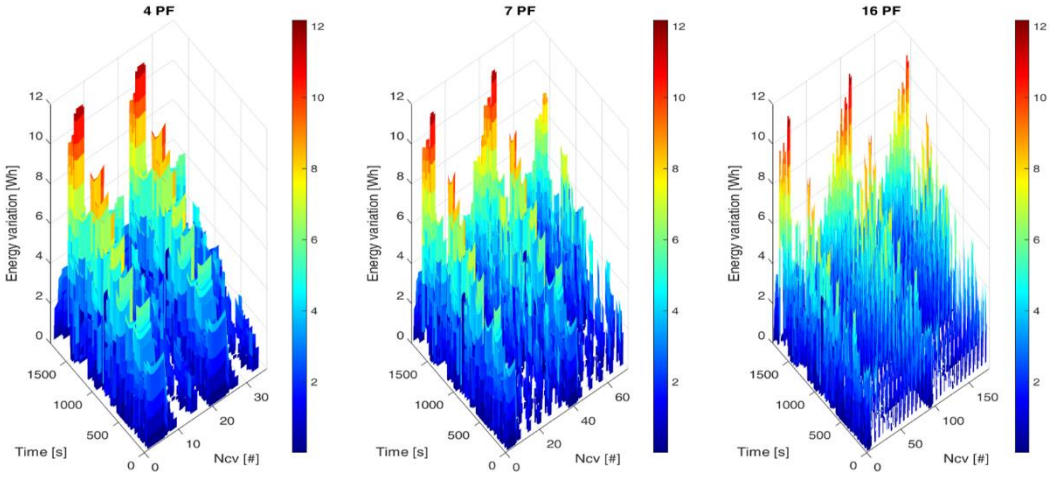

Figure 26. Battery energy variations throughout the driving mission as a function of the number of sub-control variables admitted in s_1, s_2 and s_3 when *pc* is tested on the WLTC.

Nevertheless, the $CO_{2,TTW}$ and TCO exhibit different increasing trends on the left-handed side of the sweep curves (smaller amount of battery SOC levels), whereas the simulation times overlap. A change into the sweep curves could have represented an obstacle for the application of an automatic approach for the selection of the DDP mesh setup. Considering the location of the markers on the sweep curves of Figure 28, the outputs of the statistical approach show to be robust even to a modification into the control space. In fact, the number of battery SOC calculated for each considered value of a fall into the stable portion of the sweep

curves. Once again, $N_{SOC,68}$ (squared markers) proves to be the optimal choice between the three different configurations.

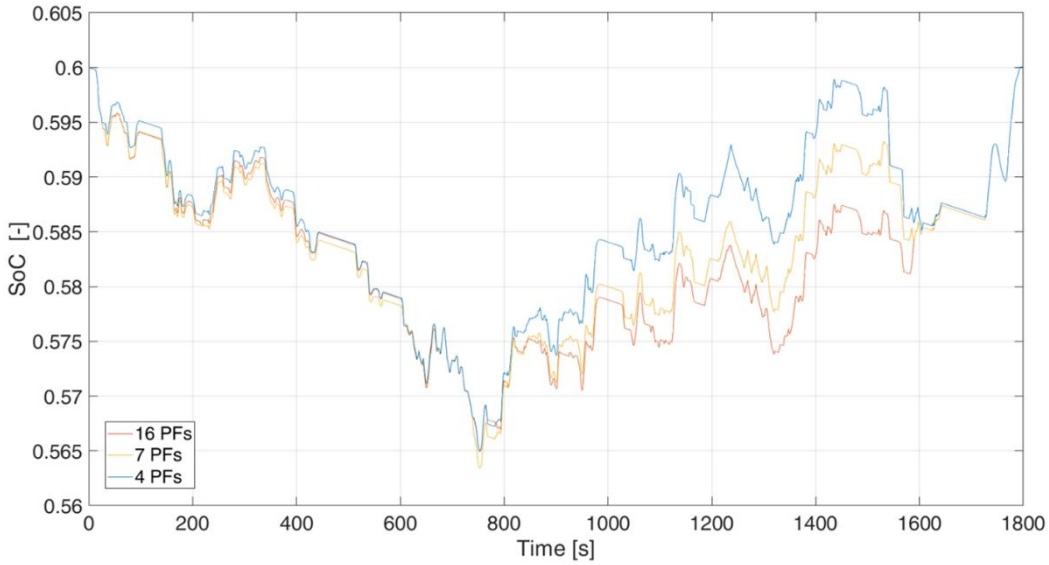

Figure 27. Battery state of charge trajectories obtained by the DDP when the number of sub-control variables admitted in s_1, s_2 and s_3 are used for *pc* on the WLTC with the maximum number of battery state of charge levels (r).

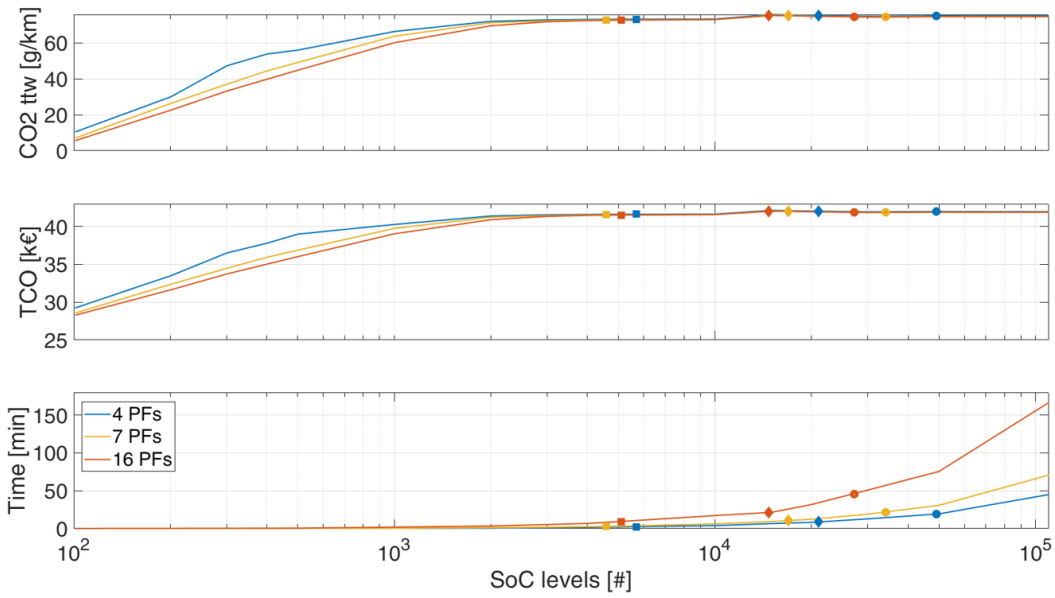

Figure 28. Comparison of the different positioning of the results obtained by the DDP on the sweep curves with $N_{SOC,95}$, $N_{SOC,90}$ and $N_{SOC,68}$ when the number of sub-control variables admitted in s_1, s_2 and s_3 are used for *pc* on the WLTC.

3.6 Conclusions

In this Chapter, a methodology for the identification of a battery SOC-based DDP mesh that guarantees the optimal trade-off between calculation accuracy and computational effort has been presented. The main contributions of the research can be summarized in:

- A method for pre-processing the distribution of the battery energy variations along a given driving mission based only on the road velocity trajectory;
- The development of a statistical approach with the capability of identifying the proper amount of SOC discretization levels of a DDP that guarantees an optimal trade-off between the accuracy of the DDP calculations and the computational effort required;
- The test of the approach on very different conditions, including different vehicles, HEV layouts and driving missions.

The entire methodology presented in this Chapter can actually be considered as a reliable tool for the identification of an optimal DDP setup in case of battery SOC as *sv*. In fact, the robustness of the method has been proved through a massive testing under very different configurations of the simulation scenario. Relying upon the results of the research, the curse of dimensionality phenomenon of the DDP can evidently be limited when the offline optimal control policy of a (P)HEV has to be identified. It is worth underlying that the presented methodology has been used for each topic presented in the following chapters.

At the present stage of the research, first a procedure for (P)HEV design optimizations and then a method for ensuring a proper configuration of the DDP algorithm have been presented. From this point onward, innovative methodologies with the capability of learning from DDP and reproducing its performance both in case of (P)HEV design and control problems are discussed.

Chapter 4

A predictive model for the CO_2 emissions of hybrid powertrains based on Deep Neural Networks

The (P)HEV design optimization based on DDP presented in Chapter 2 allows for the identification of the hybrid layouts with the best energetic performances in a specific testing condition (i.e. type of vehicle, hybrid architecture, driving mission, etc.). An optimal trade-off between the accuracy of the DDP calculations and the simulation time could also be obtained thanks to the approach presented in Chapter 3. Nevertheless, a non-negligible issue could arise when the design optimization has to focus on a very large design domain including multiple (P)HEV architectures and layouts as well as different simulation setups. In these cases, the machine time reductions permitted by the statistical approach might not be enough. Beyond the problem about the machine time, the computational effort required by a complete design optimization with multiple iterations in the design space could be too expensive. The latter could hence be forecasted when new and unknown inputs are fed into the same tool.

Part of the present section has been extracted from:

Maino, C. *et al.* (2021). A deep neural network based model for the prediction of hybrid electric vehicles carbon dioxide emissions. Energy and AI. Elsevier Ltd, 5, p. 100073. Doi: 10.1016/j.egyai.2021.100073.

Evidently, the forecasting operation on new data should be much faster with respect to the new simulations through DDP.

Machine Learning represent a promising family of techniques with the capability of learning from a dataset. Beyond the classification in SL, UL and RL, a typical macro-discrimination is made between classification and regression-aimed ML algorithms ([167]). Briefly, classification algorithms are employed to predict a class the processed inputs belong to, whereas regression algorithms are used to predict numerical values. Recalling the difference between SL and UL, "labelled data" (i.e. data for which the class/value to be predicted is a priori known) have to be fed into a SL algorithm, whereas "unlabelled data" can be comprised in the dataset of an UL algorithm with the aim of intercepting macro-trends in the input data. Considering the topic of CO_2 prediction for (P)HEVs, a global optimization through DDP (or any other global optimizer) would guarantee the calculation of optimal outputs for each of the tested layouts. Labelled data would be available and a SL problem would be generated. Specifically, a SL problem in which the values of the DVs of a specific (P)HEV architecture and the $CO_{2,TTW}$ represent the inputs and the prediction target, respectively.

Since the numerical correlations between the layouts' specifications (i.e. the inputs) and the global optimizer's results (i.e. the outputs) are expected to be highly non-linear ([168]), Neural Networks (NNs) have been selected as the most promising solution for the considered prediction task. NNs represent one the ML-related tools with the capability of solving complex prediction problems ([169]). Common features are the layered structure, a variable number of nodes for each computational layer and a recursive approach that is of paramount importance in the learning process ([170]). In case of multiple hidden layers (i.e. the layers comprised between the first and the last layers of the NN), Deep-NNs (DNNs) are generated, which have shown a greater learning potential due to their deep layered structure ([171], [172], [173]). In other words, a DNN is a multilayer perceptron (MLP), shallow if only one hidden layer is used or deep if more than one hidden layer are used.

An innovative utilization of DNN for (P)HEVs is presented in this Chapter which differs from the AI-based solutions typically presented in the literature. In fact, NNs for (P)HEVs are basically focused on monitoring the real-time vehicle performances as well as on dictating a proper control strategy. On the contrary, the methodology hereafter discussed is intended to be integrated within design

operations for (P)HEV fleets and could represent a fast and reliable tool to identify the optimal design regions related to specific driving conditions.

4.1 Vehicle models, database definition and problem formulation

Three different HEV parallel architectures featured by a single MG have been analyzed for the present study, namely P2 (pre-transmission), P3 (post-transmission) and P4 (through-the-road). The HEV pre-transmission driveline is reported in Figure 6, whereas post-transmission and through-the-road are illustrated in Figure 29 and Figure 30, respectively. In a post-transmission HEV, the MG is connected to the traction axle through a TCD before the TR (wheels-side), allowing for the MG speed to the independent from the ICE shaft speed (i.e. the transmission ratio does not affect the actual speed seen by the MG). On the other hand, the MG is connected to the non-traction axle and a complete decouple of the two propellers is realized.

A baseline HEV model has first been generated for the three HEV parallel architectures according to the modeling approach presented in Chapter 2. Please refer to Appendix A for insights about the power and speed calculations for P3 and P4 drivelines. The main specs about the baseline vehicle are reported in Table 13. Therefore, a DoE-based VL (Section 2.3) has been produced for each architecture and the related HEV layouts have been set up relying upon the map scaling operations presented in Section 2.3.1 and Section 2.3.2. The P2 and P3 architectures have been characterized by the same DVs (Table 1), whereas the P4 architecture has not included the TCD since the position of the MG on the rear-axle requires the exploitation of a rear final drive (FDR) instead of TCD ([174]). Finally, a database of HEV layouts has been generated based upon the values assumed the entire set of DVs, including the P2, P3 and P4 datasets. Overall, 1500 samples (i.e. different HEV layouts) have been considered for each HEV architecture (i.e. 4500 overall elements).

Considering the DDP algorithm proposed in Chapter 2 ([174]) featured by a number of battery SOC levels optimized through the statistical presented in Chapter 3 ([154]), the $CO_{2,TTW}$ emissions of the HEV layouts have been obtained for the WHVC. The database could hence be integrated with the CO_2 emissions estimated by the DDP. An extract of the database of the separate axle HEV architecture is reported in Table 14. It is constituted by 1500 entries (number of rows) and 8

features (number of columns), the latter comprising seven features related to the DVs (columns 2nd to 8th) and one feature for the DP-based $CO_{2,TTW}$ emissions (9th column).

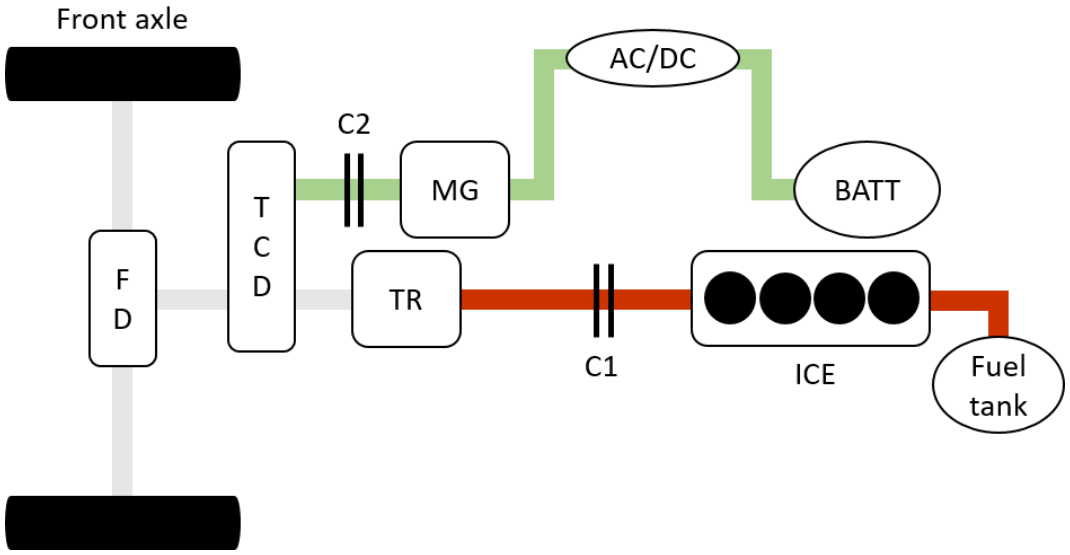

Figure 29. Driveline of a post-transmission parallel hybrid electric powertrain.

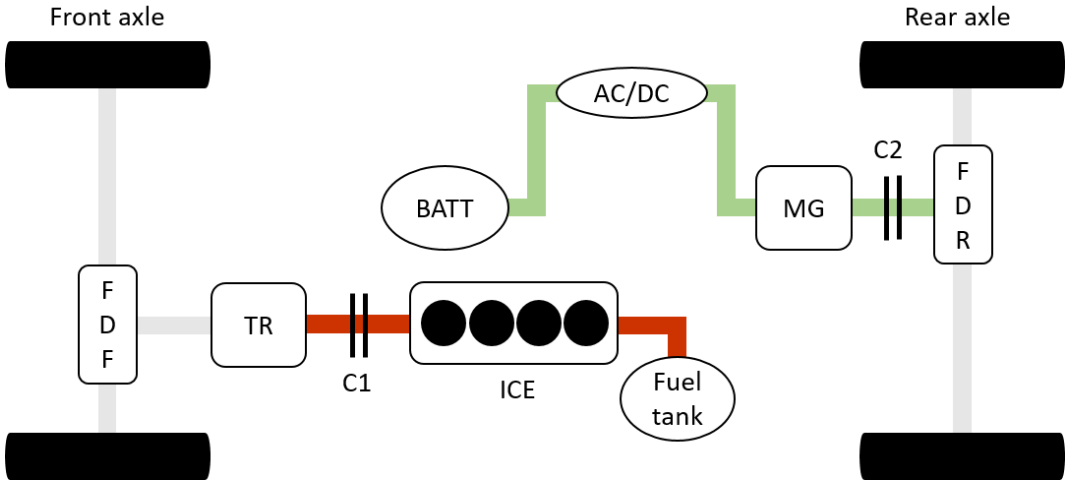

Figure 30. Driveline of a through-the-road parallel hybrid electric powertrain.

It is worth observing that the same structure would apply for the databases of the P2 and P3 architectures except for the sixth column which will be representative of the TCD speed ratio (sr_{TCD}) instead of the FDR speed ratio (sr_{FDR}).

Table 13. Main specs of the baseline vehicle and its propellers.

Vehicle	
Vehicle class	Heavy-duty
Vehicle total mass	7500
Number of wheels	6
Internal Combustion Engine	
Type	Diesel
Displacement	4.5 *l*
Maximum power	150 *kW*
Motor-generator	
Rated Power	125 *kW*
Maximum Torque in traction	300 *Nm*
Maximum Torque in braking	-300 *Nm*

Given the possibility of DDP to detect both feasible and unfeasible layouts, a numerical distinction between the CO_2 emissions produced in the two cases had to set up. To this end, a very large value equal to 10000 *g/km* has been connected to the CO_2 emissions of unfeasible layouts. This approach could have been replaced

by other similar approaches, such as including a Boolean variable (0 for unfeasible and 1 for feasible layouts). Nevertheless, a modification should have been applied to the original structure of the database and an additional choice about the $CO_{2,TTW}$ value of unfeasible layouts had to be taken. Given the CO_2 results of feasible layouts appear to be far below 10000 g/km, the choice of connecting unfeasible layouts to a very large and unrealistic CO_2 emissions value has guaranteed the possibility of numerically avoid problems in the training process of the NNs.

Table 14. An extract of the database related to the P4 architecture.

Sample	V_{ICE}	$P_{max,MG}$	PE	sr_{FDF}	sr_{FDR}	$C_{dis,max}$	$C_{ch,max}$	$CO_{2,TTW}$
	l	*kW*	*kW/kWh*					*g/km*
1	2.6	106	5.1	4.1	14.8	8.8	9.3	377.1
2	3.9	113	5.1	4.6	10.5	8.0	8.0	389.4
…	…	…	…	…	…	…	…	…
1500	3.9	52	29.3	4.7	10.4	6.3	11.5	10000.0

4.2 Deep Neural Networks for the prediction of CO_2 emissions of hybrid electric vehicles

Two different DNNs have been developed aimed at solving two different tasks: first, a classification task related to the distinction between feasible and unfeasible layouts, then a regression task about the prediction of the actual $CO_{2,TTW}$ emissions produced by feasible layouts. The classification and regression DNNs will be referred as cDNN and rDNN, respectively. A pipeline made of the two DNNs has hence been integrated within a larger model, namely DNNs-PM, along with other minor tools. The DNNs-PM is capable of producing a realistic prediction of the DP-based CO2 of HEVs by means of a two-steps procedure:

1. Assessment of the HEV layout feasibility: a cDNN predicts the capability of a specific HEV layout of concluding a driving mission while satisfying the boundary conditions of (33)-(35);
2. Prediction of the CO_2 emissions: an rDNN predicts the CO_2 emitted by the feasible layouts.

A general formulation of the inputs/outputs of the DNNs can be written as:

$$\begin{cases} F = f(DesPars, \theta_c) \\ CO_{2,TTW} = f(DesPars(F), \theta_r) \end{cases} \tag{62}$$

where F represents the set of feasible HEV layouts predicted by cDNN, $DesPars$ represents the set of design parameters considered for the HEV layouts, $CO_{2,TTW}$ represents the tank-to-wheel CO_2 emissions predicted by rDNN, while θ_c and θ_r represent the set of parameters for the cDNN and rDNN, respectively.

In Figure 31, the workflow of the DNNs-PM is reported. Once the database has been successfully generated, a set of pre-processing operations is carried out (targets definition and database normalization) and the cDNN training is performed. During the learning process, an Automatic Search Tool (AST) is employed to select the most effective combination of network's parameters. A validation of the model is performed so as that the real performances of the DNN could be evaluated on new and unknown data. Once the feasible layouts are identified and the unfeasible ones are discarded (i.e. filtering procedure), the updated database can be passed to the second DNN. The rDNN learning process can finally take place with the support of the AST and the CO_2 regression can be realized.

In the following sections, the steps taken for the development of the DNNs-PM are thoroughly discussed. Then, the results of the model are presented for a specific HEV application. For the sake of clarity, the typical formulation and operating principle of a NN are hereafter not remarked as they are taken for granted. Please refer to [175] if a deeper dissertation about NNs is needed. The DNNs have been developed using Python programming language ([176]).

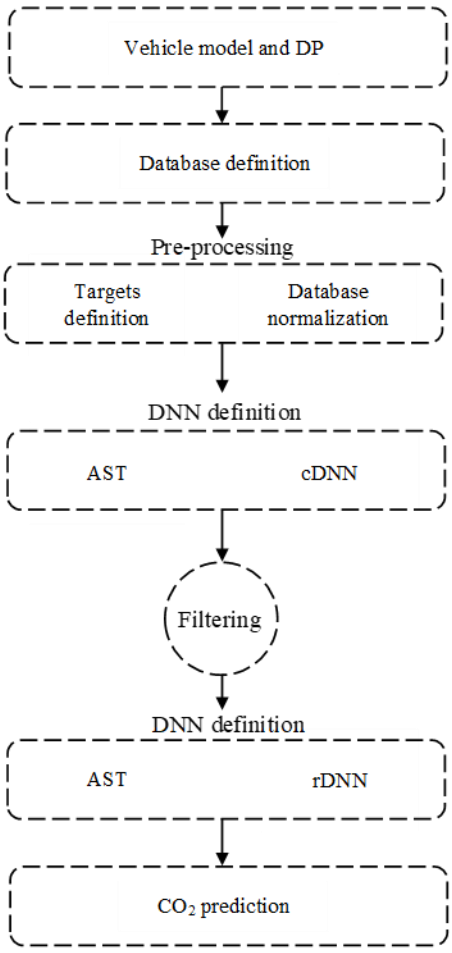

Figure 31. Workflow of the model based on Deep Neural Networks for the prediction of the CO_2 emissions of hybrid electric vehicles.

4.2.1 Data management

A single database has been passed to the DNNs-PM (Table 14), indeed it has been handled differently based on the specific type of DNN to be employed (i.e. classification or regression purposes). The rDNN is asked to predict the exact CO_2 estimates and hence any action has to be performed on the values of the CO_2 produced by DP and stored in the last column of Table 14. On the contrary, the cDNN aims at performing a classification and hence categorical targets are required instead of numerical targets. To this end, the CO_2 estimates ($CO_{2,TTW}$) of Table 14 have been replaced by:

$$F = \begin{cases} 1 & if\ CO_{2,TTW} \neq 10000 \\ 0 & otherwise \end{cases} \tag{63}$$

Another operation has been performed on the values assumed by the design parameters. Different order of magnitudes are highlighted in Table 14 for the seven input variables. That is consistent with the physical quantities represented by the design parameters. From a numerical perspective, such a condition can lead to an inefficient network training as the network predictions could be learnt based only on the features characterized by larger values ([177], [178]). An effective and fast solution is the normalization of the whole dataset. This procedure is typically realized by substituting each entry of the dataset with an equivalent value using a relation which is able to enforce the same range of variation for the entire set of features while preserving their informative potential ([179], [180]). In the present activity, a database standardization has been chosen to avoid an unbalanced distribution after the normalization data. The updated values of the input features have been calculated through:

$$\tilde{x}_i = \frac{x_i - \bar{x}}{std(x)}\ for\ i = 1, 2, \dots, N \tag{64}$$

where $\tilde{x}_i$ is the i-th layout's feature after standardization, x_i is the i-th layout's feature before standardization, $\bar{x}$ is the average of the considered features, $std(x)$ is the standard deviation of the considered feature and N is the number of design parameters. Based on the standardization operation of (64), the values assumed by the descriptive features have become numerically comparable.

A final operation has been performed over the database aimed at generating three different sub-datasets to be used as training, validation and test sets. The

number of entries of each sub-dataset have been defined based on two different indexes, namely train-to-test split (*t/t split*) and train-to-validation (*t/v split*). The *t/t split* is representative of the fraction between the number of entries in training and validation sets and to the total number of entries in the database. The *t/v split* is representative of the fraction between the number of entries in the training set and the entries in validation set. Since the depth of the training set can strongly affect the model performance ([172]), the larger share of data has been selected as training set. The latter has been characterized by labelled data and has been employed to train the DNNs to learn the correlations between the inputs features and the output targets. On the other hand, smaller validation and test sets have been generated: the validation set has been employed to assess for the DNN performances during validation procedures aimed at identifying the best networks' parameters; the test set has been considered when the validated DNN had to be tested on new unlabelled data. As final consideration, the entries of each sub-dataset been selected by means of a stochastic operation aimed at excluding the DNN behaviour to be biased by specific data splits.

4.2.2 Definition and tuning of the model parameters

Three different types of networks' parameters have been considered in the present activity: networks' weights, fixed-parameters and hyper-parameters. The networks' weights represent the parameters involved in the computation of the activation functions, which are autonomously tuned in the back-propagation phase of the training process ([43]). The fixed-parameters have been considered as the user-defined parameters to be identified before the training process of the networks begins. In the present work, the values of the fixed parameters have been chosen after several experiments aimed at exploiting the optimal trade-off between training time and networks' accuracy. The list of fixed-parameters is presented in Table 15. Evidently, fixed-parameters have been kept unchanged for the entire set of analyses (the description and investigation of these parameters is beyond the scope of this study and the related investigations are hereafter not presented). Finally, any other parameter related to the DNNs with the necessity of a tuning (and optimization) operation integrated in the training process has been identified as hyper-parameter. Given the different nature of the networks involved in the DNNs-PM pipeline, different sets of hyper-parameters have been considered.

In Table 16, the lists of hyper-parameters are reported both for cDNN and rDNN. As far as cDNN is concerned, the learning rate and the L2 regularization

parameter are characterized by logarithmic variations with base 10, whereas the batch size assumes a logarithmic variation with base 2. Indeed, the number of hidden layers and the number of neurons in the first hidden layers are featured by a variation in the whole numbers domain. About rDNN, the same considerations can be made for each hyper-parameter except for the L2 regularizer, which has been neglected and replaced by the weighs initialization technique.

Additional hyper-parameters related to the number of neurons in each hidden layer should have been included in the list of both cDNN and rDNN. Nevertheless, unbounding the number of neurons per layer could have introduced two relevant issues. First, the real number of hyper-parameters to be tuned would have been linked to the value assumed by a single hyper-parameter. In fact, a variation into the number of hidden layers would have led to a modification of the length of the hyper-parameters list. A kind of inconvenient loop of hyper-parameters would have generated. Second, the number of possible combinations to be analyzed could have exploded in case of networks with many hidden layers. Therefore, the following equation has been employed to calculated the number of neurons of each hidden layer while bounding the hyper-parameters' space:

$$L_j = \frac{L_1}{2^{(j-1)}} \tag{65}$$

in which L_j is the number of neurons of the j-th hidden layer while L_1 is the number of neurons of the first hidden layer. A 5-dimensional hyper-parameters' space can hence be studied both for cDNN and rDNN. Since the range of the values assumed by each hyper-parameter could vary in a very wide domain, an automatic procedure for the selection of meaningful hyper-parameters' combination is necessary. Consistent with the model workflow of Figure 31, a tool (AST) for an automatic on-training tuning of the networks' hyper-parameters has been developed. Through AST, several combinations of the values assumed by the DNN hyper-parameters ("trials") are randomly selected through a Random-Search (RS) operation ([181]). Therefore, the network performances are assessed for using a k-fold Cross-Validation (k-CV) operation ([182]). For each trial, an average task-specific performance index based on the k validation splits is computed. Evidently, considering an average value represents a more robust approach to assess for the real performances of a given trial for a specific DNN (e.g. compared to the evaluation of the maximum values).

Table 15. Fixed-parameters considered for the Deep Neural Networks.

Fixed-parameter	cDNN	rDNN
Number of epochs	250	100
Activation function (internal layers)	ReLU	ReLU
Activation function (output layer)	Sigmoid	Linear
Optimizationr	Adam	Adam
Normalization	Batch normalization	Batch normalization & Dropout

Table 16. Hyper-parameters considered for the Deep Neural Networks

cDNN	rDNN
Learning rate	Learning rate
Number of hidden layers	Number of hidden layers
Number of neurons in the first hidden layer	Number of neurons in the first hidden layer
L2 regularization parameter	Weights initialization
Batch size	Batch size

Finally, the performances obtained through different trails are compared using a task-specific algorithm and the most promising set of hyper-parameters is selected.

In the following sections, the performance indexes used for cDNN and rDNN are first presented and then the entire AST procedure is illustrated for both cDNN and rDNN.

4.2.3 Performance index for the Classification Deep Neural Network

As far as the classification task of the cDNN is concerned, a binary classification had to be performed based on (63). In the present work, the Matthews Correlation Coefficient (MCC) has represented the performance index used to assess for the cDNN performance. The MCC is considered as one of the best "single-number" performance indexes, deriving directly from the Confusion Matrix (CM) calculated for a specific trial ([183]). The MCC can be computed as:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{66}$$

where TP are the true positive predictions, TN are the true negative predictions, FP are the false positive predictions and FN are the false negative predictions. The MCC ranges from -1 (the model is in a complete opposition with respect to the observations) to 1 (perfect classification); in the case of null MCC, the model is considered to produce random predictions. Along with the MCC, the associated CM has also been considered for an even more exhaustive evaluation of the networks' performances.

4.2.4 Performance index for the Regression Deep Neural Network

Given the target of the rDNN to replicate the numerical result, the widely used Coefficient of Determination (CoD) R^2 has been selected as one promising index to track the fitting effectiveness of the DNNs-PM predictions with respect to DP-based targets ([184]). The CoD can be calculated through:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \; with \; i = 1, 2, \dots, M \tag{67}$$

in which y_i is the true value of the i-th sample, $\hat{y}_i$ is the predicted value of the i-th sample, $\bar{y}$ is the average of the target values and M is the number of samples. The CoD ranges from an infinitely negative number to 1: a baseline model predicting the average value of the distribution exhibits a null R^2; a perfect fit of the model leads to R^2=1, whereas negative values typically account for models worse than the baseline one ([185]).

Additionally to CoD, the Root Mean Squared Error (RMSE) has also been monitored throughout the experiments as loss-function of the rDNN ([186]). The RMSE can be calculated through:

$$RMSE = \sqrt[2]{\frac{\sum_i^M (\hat{y}_i - y_i)^2}{M}} \; with \; i = 1, 2, \dots, M \tag{68}$$

The values assumed by the RMSE range from 0 (perfect predictions) to infinitely large values (predictions infinitely distant from target values).

4.2.5 Automatic Search Tool

The steps performed by the AST have been adapted according to the set of networks' hyper-parameters. In case of cDNN, a user-defined range of variation ("space") for each hyper-parameter has been first selected and divided into 3 equal sectors ("sub-spaces"). Then, a 4-steps multi-level procedure has been employed to assess for the best combination of hyper-parameters:

1. A RS is performed in the whole hyper-parameters' space with a user-defined number of trials. For each RS output, the related sub-space is monitored (beginning of higher-level tuning procedure);
2. An 8-fold k-CV is performed for each trial. Considering the outputs of each CV fold, the hyper-parameters' combinations leading to the three best MCC results are selected for each trial. Therefore, the three sub-spaces related to the best results are identified (end of higher-level tuning procedure);
3. A RS is performed in each sub-space with the same number of trials of Step 1 (beginning of lower-level tuning procedure);
4. An 8-fold k-CV is performed for each trial. Considering the outputs of each CV fold, the hyper-parameters' combination leading to the best

MCC is identified and chosen for the final training and testing of the cDNN (end of lower-level tuning procedure).

Considering the differences between the performance indexes of cDNN and rDNN, the procedure of the AST have been modified in case of rDNN. Specifically, three operations have been conducted:

1. A RS is performed in the whole hyper-parameters' space with a user-defined number of trials;
2. An 8-fold k-CV is performed for each trial. Considering the outputs of each CV fold, the hyper-parameters' combinations leading to average CoD higher than 0.75 are stored;
3. If at least one trail with average CoD higher than 0.75 is identified, the the hyper-parameters' combination leading to the highest CoD and chosen for the final training and testing of the rDNN. If no trials with CoD higher than 0.75 are identified, the procedure is repeated starting from Step 1.

4.3 Results

The DNNs-PM has been applied to the prediction of both feasibility ($feas$, classification task) and tank-to-wheel CO_2 emissions ($CO_{2,TTW}$, regression task) identified by the DDP for several HEV parallel layouts of an *hdv*. The performance of the DNNs-PM has been evaluated following two steps: first, the evaluation of the model capability to effectively learn from a single dataset ("intensive training") and, then, the comparison of the results produced by the model when tested on multiple different datasets ("stress test"). This approach has first been applied to the cDNN as stand-alone DNN and, hence, it has been extended to the entire pipeline. The study of the results produced of the intensive training is thought to be useful for a deep comprehension of the DNN behaviour, whereas a comparison on different datasets is mandatory to highlight the real generalization capability of the model when tested under various conditions. In the following sections, the outcomes of 46 independent experiments (28 for the cDNN, 18 for the entire pipeline) are presented which aim at assessing for the realistic potentials of embedding the DNNs-PM into a (P)HEV design optimization operation. For the sake of completeness, the results presented in the following sections refer to experiments performed with a 2.4GHz CPU-12GB RAM personal computer.

4.3.1 cDNN: results of the intensive training

As first step, the consistency of the learning process of cDNN has been assessed for by analyzing the learning curves obtained during the training phase of a single experiment to ensure the absence of underfitting, overfitting or biasing phenomena. The dataset of the P4 HEV architecture has been considered for the present analysis. Recalling Section 4.2.1, a random selection of the training, validation and test sets is performed so as that any bias is introduced in the network operation due to a fixed split in the data. Twenty hyper-parameters' combinations have included in the tuning process carried by the AST. A limited number of 250 epochs has been considered for the training phase and the *t/t split* has been set to 80-20 (i.e. the fraction of training and validation samples with respect to the overall number of sample in the dataset is equal to 80%). Notice that the *t/v split* has been set to 7-1 since an 8-fold k-CV has been involved in the AST procedure. The complete configuration for the intensive training analysis over cDNN is resumed in Table 17.

Table 17. Configuration of cDNN for the intensive training analysis.

Hyper-parameter	**Range of variation**
Learning rate	0.0002 – 0.02
Number of hidden layers	1 – 4
Number of neurons in the first hidden layer	130 – 230
L2 regularizer	0.003 – 0.3
Batch size	16 – 128
Sub-dataset	**Number of samples**
Training	1050
Validation	150
Testing	300

Consistent with Section 4.2.3, two performance indexes have been taken into account for monitoring the cDNN performances, namely MCC and CM. In Figure 32 and Figure 33, the results are reported in terms of MCC and CM, respectively. The MCC (Figure 32) has been monitored throughout the training phase (red curve) and for each split considered in the k-CV procedure. Then, the average MCC trend has been calculated and monitored for the k-CV splits (green curve) and compared with the MCC trend of the best performing split (blue curve). The latter is represented by the data split leading to the highest MCC at the last training epoch. Promising results highlight as the MCC trajectory increases along with the training epochs. First, an increasing trend can be detected in both training and validation curves. Evidently, the average MCC values calculated considering the entire set of k-CV data splits are affected by ineffective hyper-parameters' combination. Second, considering red and blue curves, the distance between the MCC traces

during training and validation shrinks along with the progress in the epochs. Third, the final values of both training and testing are consistently above 0.8, which can be considered as a very good result in a classification task. It can hence be stated that cDNN is not undergoing a healthy learning process without underfitting, overfitting phenomena ([187]). Furthermore, no warnings are shown by the curves noise, which can be linked to the intrinsic stochastic approach of DNNs. Consequently, a lower number of samples in the data split can be connected to a lower possibility of incurring in smoothed responses of the DNN predictions. From this perspective, the MCC proves to be spikier in case of best validation split (150 samples) with respect to the training sub-dataset (1050 samples).

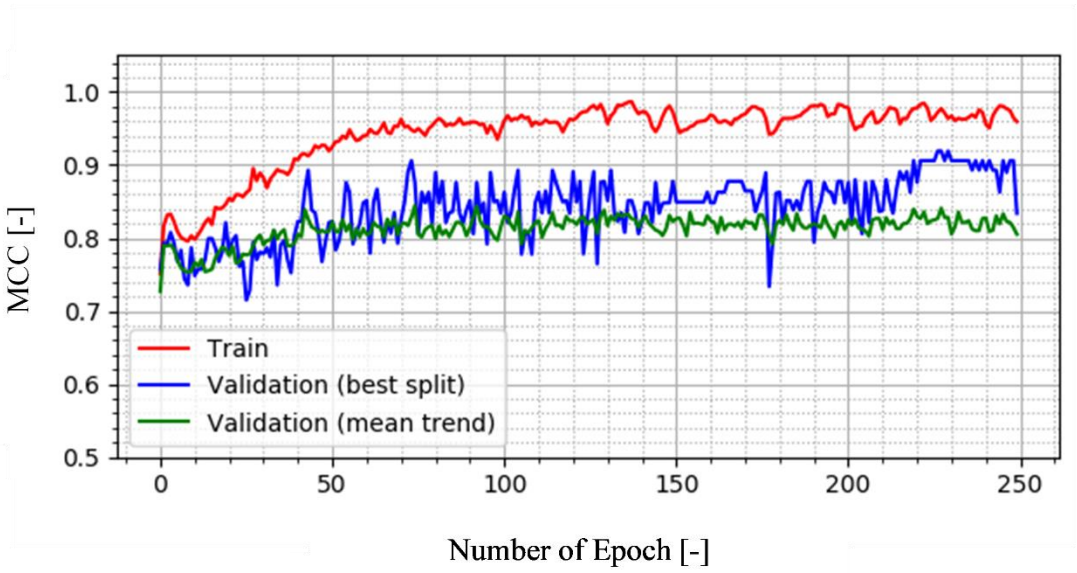

Figure 32. Trend of the MCC obtained by cDNN during training and validation phases of the intensive training analysis.

In addition, the complete CM obtained by cDNN when tested on the P4 dataset is shown in Figure 33, which reports the exact number of TP, TN, FP and FN. As it can be noted, the results of the main diagonal prove a strong capability of the DNN to predict the correct class. The number of effective prediction peaks at 86.9% while the MCC calculated through (66) results in 82.9%. Such a value is aligned

with the final value of the average MCC trend (Figure 32, green curve) proving the robustness of the outcomes at the end of the k-CV operations.

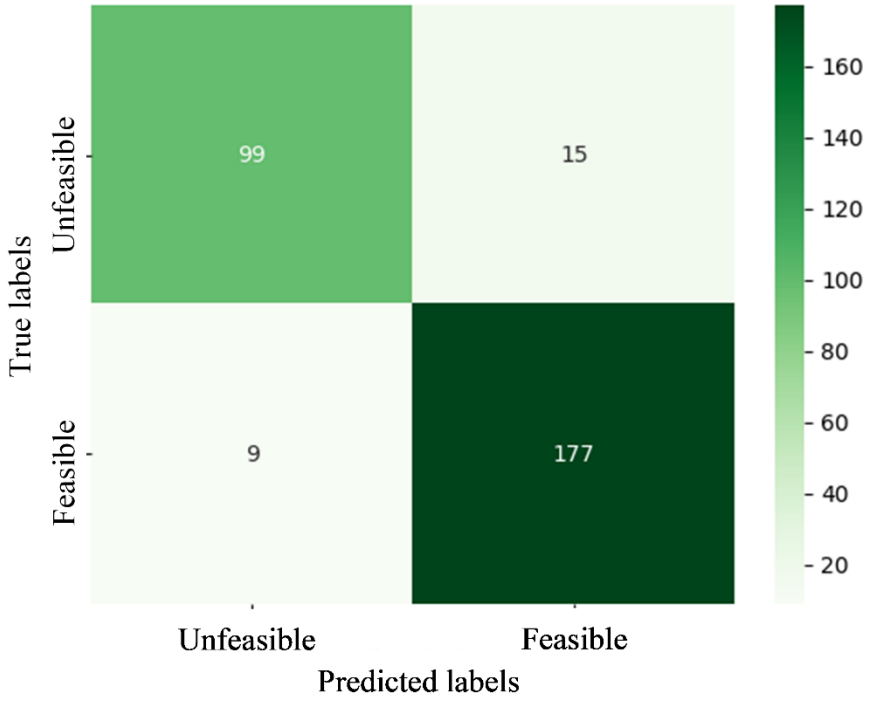

Figure 33. Configuration Matrix obtained by cDNN during testing phase of the intensive training analysis.

At this stage, the cDNN capability to produce stable performances has been evaluated based on the results obtained by 24 experiments in which changes into the input datasets are introduced. Such an analysis has aimed at excluding biasing phenomena related to the selection of a specific training, validation and testing sub-datasets. Specifically, six tests have been conducted for four different *t/t splits* (60-40, 70-30, 80-20 and 90-10). Consistently, a diversification of the sub-datasets is ensured and the assessment of an unbiased behaviour is promoted. In Table 18, a description of the new sub-datasets is reported for each *t/t split*. Any other parameter related to the setup of the experiments (e.g. number of epochs, hyper-parameters' space, etc.) has been kept unchanged with respect to the first intensive training analysis. Therefore, the *t/v split* has remained equal to 7-1.

Table 18. Updated setup of the sub-datasets considered for the intensive training analysis.

t/t split	Trials	Number of samples		
		Training	Validation	Test
60/40	6	788	112	600
70/30	6	919	131	450
80/20	6	1050	150	300
90/10	6	1181	169	150

The results obtained by cDNN on the updated testing sub-datasets are presented in Table 19 in terms of actual MCC for each trial (I, II, …, VI), average MCC (*avg*) and standard deviation of the results (*std*). A direct correlation arises between classification accuracy and training set size as the average MCC appears to increase along with the number of training samples. Such a result can be considered as the most coherent outcome of a robust learning process. Moreover, the experiments with 80/20 and 90/10 *t/t splits* are featured by both higher *avg* and *std*. As far as the *avg* is concerned, an amelioration of the DNN average performance could be expected coherently with the broaden number of training samples. On the other hand, reducing the number of testing samples can lead to an increase of the influence produced by the outcome of a single prediction with respect to the overall prediction. This leads to higher *std* values.

Table 19. Results obtained by cDNN considering the updated sub-datasets of the intensive training analysis

t/t split	*I*	*II*	*III*	*IV*	*V*	*VI*	*avg*	*std*
	%	%	%	%	%	%	%	%
60/40	81.9	82.1	76.5	82.2	79.7	76.4	79.8	2.33
70/30	82.1	86.3	81.6	76.8	81.6	81.6	81.7	2.55
80/20	82.9	82.2	87.2	85.1	84.4	75.0	82.8	3.55
90/10	85.8	78.8	80.4	84.5	88.8	85.8	84.0	3.16

4.3.2 cDNN: results of the stress test

Once the evaluation of the model robustness has been completed considering the dataset of a single HEV architecture (P4), a comparison of the results produced by the model when applied to three different datasets (P2, P3 and P4 alternatively) has been carried out under slightly different conditions. For the present stress test, the hyper-parameters' space has been enlarged and the number of trials in the AST procedure has been increased. In fact, no a priori assumptions can be made about the effectiveness of the optimal hyper-parameters' combination found for the P4 dataset when applied to different architectures. Therefore, the extension of the hyper-parameters' space can increase the probability of spotting an optimal (or near optimal) hyper-parameters' combination for each architecture. For the just mentioned reasons, an increase in the overall performance has also been expected for the P4 dataset.

For the present analysis, the *t/t split* has been set to 90/10 since it has proved to be beneficial to the cDNN performance (Table 19). The complete setup updated for the stress test of the cDNN is reported in Table 20. Given the share of training samples with respect to the overall number of samples in the datasets, a single trial has been conducted and the related results have been considered for the comparison of the cDNN performances.

The results obtained at the end of the stress test are reported in Table 21 considering the MCC over the testing sub-datasets of each HEV dataset. The

expected increase in the cDNN classification accuracy is evidently confirmed. Particularly for the P4 dataset, the MCC reaches 91.5% that represents an increase of more than 2.7% with respect to the maximum MCC registered in Table 19 for the same *t/t split*. Nevertheless, the cDNN performances on the classification of feasible P2 and P3 layouts are found to outperform the best performances found for the P4 dataset. In fact, the MCC obtained for the P2 and P3 datasets peaks roughly 97% and 92%, respectively. Such an outcome can evidently be linked to the correlations underlying between the DVs of pre/post-transmission HEV architectures and the DP-based tank-to-wheel CO_2 emissions. Nevertheless, such a hypothesis has not been further explored in the present study and its represents ground for future research.

Table 20. Configuration of cDNN for the stress test.

Hyper-parameter	**Range of variation**
Learning rate	0.00001 – 0.1
Hidden Layers	1 – 15
Number of neurons in the first hidden layer	20 – 300
L2 regularizer	0.0001 – 0.09
Batch size	16 – 516
Sub-dataset	**Number of samples**
Training	1181
Validation	169
Testing	150

Table 21. Results obtained by cDNN on the testing sub-datasets of each architecture considered in the stress test.

Dataset	MCC %
P2	96.9
P3	92.4
P4	91.5

4.3.3 rDNN: results of the intensive training

The rDNN is responsible of predicting the CO_2 emissions of the HEV layouts classified as feasible by the cDNN. The assessment of the performances achieved by the DNNs pipeline has been carried out through 18 experiments. Specifically, six experiments have been performed over the datasets of each considered architecture (P2, P3 and P4). Similar to cDNN, six experiments have first been performed with *t/t split* fixed at 90/10 over the P4-dataset; the hyper-parameters' configuration used for the rDNN is reported in Table 22. Given that the number of samples used by the rDNN depends upon the number of feasible layouts detected by the cDNN, the split between training, validation and testing sub-datasets is not reported in Table 22. For the sake of clarity, the small variations in the MCC produced by the cDNN (Table 19) have led to the identification of a very similar number of feasible layouts.

The performance indexes presented in Section 4.2.4 have been taken into account for monitoring the performances of the rDNN. The learning curves (i.e. the trends of the rDNN performance indexes) obtained with a generic experiment of the intensive training of the rDNN over the P4 dataset are reported in Figure 34. An increasing trend of the R^2 highlights both for training (red curve) and for validation (green curve) sub-datasets; contrarily, a decreasing trend of the loss function (i.e. RMSE) shows both for training (blue curve) and for validation (orange curve). These two outcomes can be considered as a promising sign of an effective overfitting-free learning process ([187]).

Table 22. Configuration of rDNN for the intensive training analysis.

Hyper-parameter	Range of variation
Learning rate	0.002 – 0.1
Hidden Layers	1 – 6
Neurons first hidden layer	30 – 80
Batch size	8 – 64
Dropout	0 – 0.5
Weights initialization	Xavier, random, truncated normal

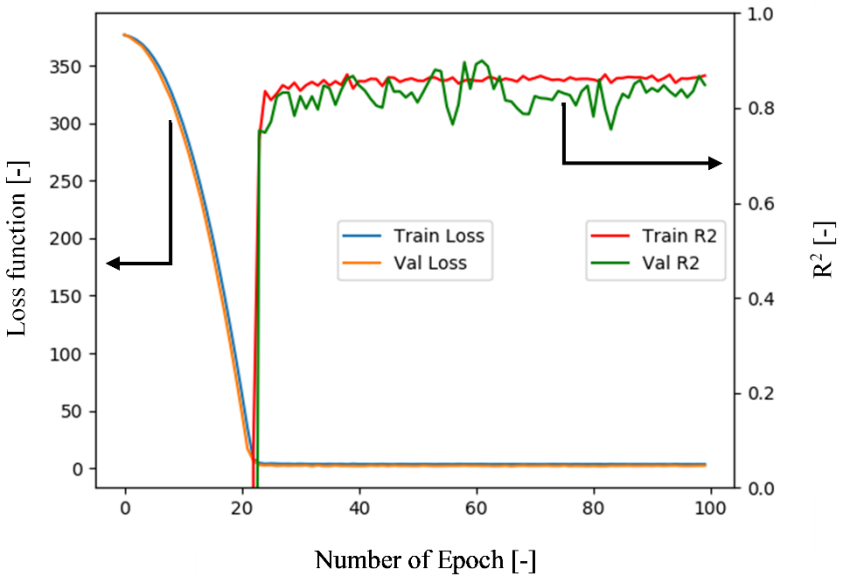

Figure 34. Trend of the performance indexes obtained by rDNN during training and validation phases of the intensive training analysis.

The comparison between the DP-generated CO_2 emissions (i.e. labels) and the values predicted (i.e. predictions) by the rDNN are reported in Figure 35 for the test sub-datasets considered for each experiment of the intensive training analysis.

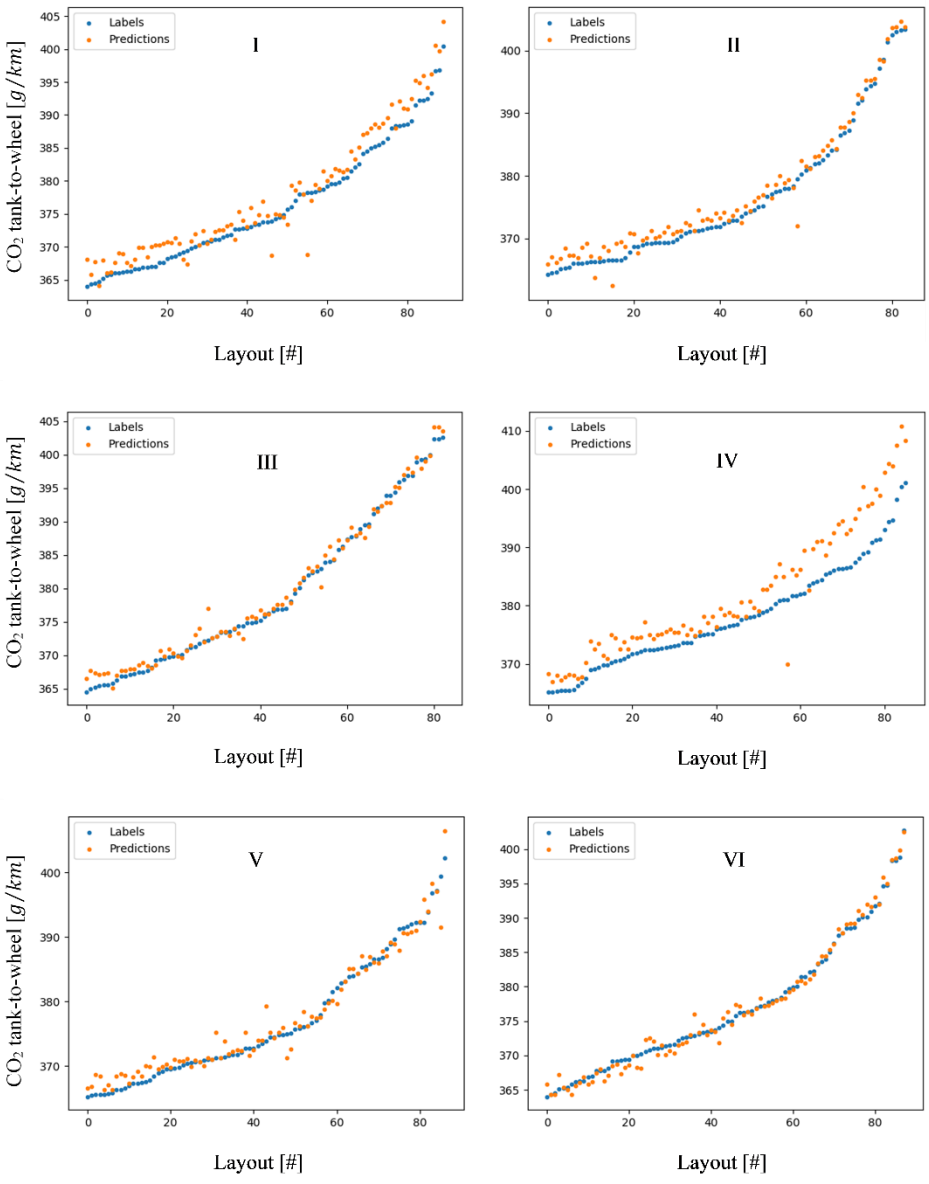

Figure 35. Comparison between the labelled CO_2 emissions and the CO_2 emissions predicted by the rDNN for the intensive training analysis.

The numerical results achieved at the end of each experiment have also been tracked so to quantify the rDNN capability of producing reliable predictions. In Table 23, the regression average errors ($\hat{E}$), maximum errors (*Max E*) and minimum errors (*min E*) are reported along with the values of CoD (R^2) and loss function (RMSE) obtained on the test sub-dataset. For the entire set of metrics, *avg* and *std* have been reported. Moreover, absolute (*Abs*) and relative (*Rel*) value have been accounted for quantifying the regression errors. Notice that relative errors of "<1" are referring to relative errors below 0.5%. Confirming the positive trend of Figure 35, very promising performances are shown by the pipeline as an overall $\hat{E}$ under 0.5% has been produced at the end of the intensive training analysis. Interestingly, even the worst rDNN performance (4th experiment) leads to a very small average error ($\hat{E}$=1%).

Table 23. Results obtained by the rDNN over the test samples of the P4 dataset considered for the intensive training analysis.

Experiment	$\hat{E}$		*Max E*		*min E*		R^2	*RMSE*
	Abs *g/km*	***Rel*** *%*	***Abs*** *g/km*	***Rel*** *%*	***Abs*** *g/km*	***Rel*** *%*	**-**	*g/km*
I	2.11	1	9.39	2	0.03	<1	0.923	2.50
II	0.92	<1	4.67	1	0.01	<1	0.988	1.25
III	1.49	<1	7.43	2	0.03	<1	0.972	1.77
IV	4.12	1	11.48	3	0.15	<1	0.672	4.61
V	1.28	<1	7.88	2	0.00	<1	0.963	1.93
VI	0.78	<1	3.13	1	0.01	<1	0.989	1.06
avg	1.78	<1	7.33	2	0.04	<1	0.918	2.19
std	1.13	<1	2.78	1	0.05	<1	0.112	1.18

As a final check, the dispersion of the relative errors of the CO_2 predictions is reported in Figure 36 for each test case. Very small errors highlight as a symptom

of the rDNN robustness to several tests. Particularly, the maximum error peaks at 3% in the 4th experiment, which can be considered as a very important result.

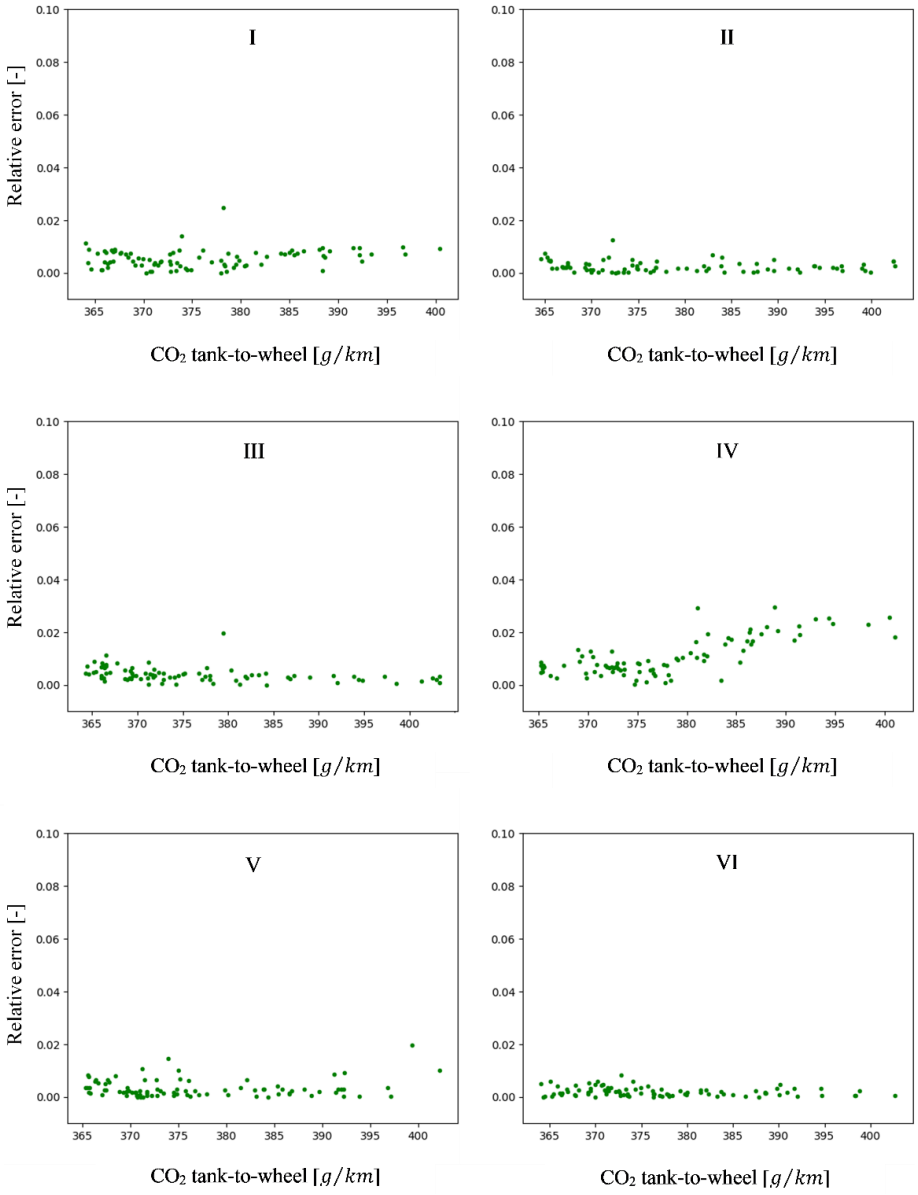

Figure 36. Dispersion of the relative errors obtained by the rDNN over the test samples of the P4 dataset considered for the intensive training analysis.

4.3.4 rDNN: results of the stress test

As final operation, a test has been conducted by applying the entire pipeline (i.e. cDNN and rDNN) to the P2 and P3 datasets considering the same *t/t split* (i.e. 90/10). Specifically, 12 experiments (six for each dataset) have been performed and the performance indexes of the rDNN have been tracked. The outcomes of each experiment are reported in Table 24 considering also the average, the maximum and the minimum errors (absolute – relative). Evidently, the results related to the P4 dataset are the same results presented in Table 23. The relative errors falling very close to zero are reported as "<1".

Robust performances can be observed for each of the tested HEV architecture. As far as average error ($\hat{E}$) is concerned, very contained errors can be appreciated regardless of the HEV architecture. The maximum relative discrepancy is detected at 1%. Furthermore, the maximum relative discrepancy (*Max E*) does not exceed 4% for P2 and P3 datasets even though a higher *std* is obtained. At the same time, the minimum relative errors (*min E*) also are very close to zero. In these cases, quasi-perfect predictions have been attained.

As far as the regression performance indexes are concerned, the RMSE is typically considered as the most reliable index to assess for the real discrepancies between the predictions and target values ([188]). Therefore, the performances of the rDNN over the three datasets have been compared based on the RMSE results. As an outcome, an improved response of the pipeline can be appreciated when applied to the P2 and P3 architectures leading to lower *avg* and *std.*

Table 24. Results obtained by the rDNN on the testing sub-datasets of each architecture considered in the stress test.

Dataset	R^2 -							
	I	***II***	***III***	***IV***	***V***	***VI***	***avg***	***std***
P2	0.911	0.536	0.788	0.806	0.879	0.716	0.773	0.114
P3	0.937	0.947	0.719	0.955	0.879	0.953	0.899	0.078
P4	0.923	0.988	0.972	0.672	0.963	0.989	0.918	0.112
	RMSE *g/km*							
	I	***II***	***III***	***IV***	***V***	***VI***	***avg***	***std***
P2	0.81	1.36	1.17	1.00	1.14	1.45	1.16	0.21
P3	0.84	0.93	1.33	0.75	1.11	0.77	0.95	0.21
P4	2.50	1.25	1.77	4.61	1.93	1.06	2.19	1.18
	$\hat{E}$ *g/km - %*							
	I	***II***	***III***	***IV***	***V***	***VI***	***avg***	***std***
P2	0.55 - <1	1.05 - <1	0.77 - <1	0.68 - <1	0.72 - <1	1.13 - <1	0.82 - <1	0.19 - <1
P3	0.59 - <1	0.64 - <1	0.99 - <1	0.56 - <1	0.92 - <1	0.54 - <1	0.71 - <1	0.17 - <1
P4	2.11 - 1	0.92 - <1	1.49 - <1	4.12 – 1	1.28 - <1	0.78 - <1	1.78 - <1	1.05 - <1
	Max E *g/km - %*							
	I	***II***	***III***	***IV***	***V***	***VI***	***avg***	***std***
P2	4.36 - 1	15.14 - 4	12.60 - 4	5.47 - 2	5.55 - 2	7.31 - 2	8.41 - 3	3.73 - 1
P3	2.83 - 1	4.72 - 1	13.79 - 4	2.83 - 1	2.38 - 1	3.20 - 1	4.96 - 2	3.72 - 1
P4	9.39 - 2	4.67 - 1	7.43 - 2	11.48 - 3	7.88 - 2	3.13 - 1	7.33 - 2	2.58 - 1
	min E *g/km - %*							
	I	***II***	***III***	***IV***	***V***	***VI***	***avg***	***std***
P2	0.00 - <1	0.01 - <1	0.00 - <1	0.01 - <1	0.01 - <1	0.00 - <1	0.01 - <1	0.01 - <1
P3	0.01 - <1	0.03 - <1	0.01 - <1	0.00 - <1	0.06 - <1	0.00 - <1	0.02 - <1	0.02 - <1
P4	0.03 - <1	0.01 - <1	0.03 - <1	0.15 - <1	0.00 - <1	0.01 - <1	0.04 - <1	0.05 - <1

4.3.5 Analysis on false positives

The perfect response of a DNN could be referred to the ideal scenario in which labeled data are perfectly predicted. Nevertheless, a small error is typically assumed even when very performing DNNs are tested in practice. In fact, the maximum accuracy achieved by a DNN operating in real-world conditions is considered when the prediction error is "small enough". The definition of the perfect response of a DNN can hence be rephrased as the response to the real scenario in which labeled data are predicted with an error tending towards zero. A fundamental step to be performed in the validation of a DNN behaviour stands in the quantification of the prediction error. In other words, is the prediction error small enough for avoiding critic malfunctioning of the DNN when tested in real-world conditions?

Specifically for classification tasks, a prediction error can turn into the generation of false positives (i.e. unfeasible layouts that are incorrectly classified as feasible) or false negatives (i.e. feasible layouts that are incorrectly classified as unfeasible). For the problem of HEV optimal design, false negatives would only lead to the avoidance of CO_2 predictions for few layouts. Such a scenario would not be dramatic from a design perspective since no misleading results could be produced and post-processed. Contrarily, the identification of false positives could strongly affect the operation of results post-processing. In fact, since an unfeasible layout is characterized by inexistent CO_2 labels (represented by 10000 *g/km* in Table 14), the generation of any CO_2 prediction would be completely unrealistic.

Therefore, a final analysis has been carried out in order to evaluate the rDNN behavior in case of a cDNN misclassification which leads to a given amount of false positive layouts. The cDNN false positive classifications resulting from the 18 experiments of Section 4.3.4 have been extracted and fed to the rDNN. The $CO_{2,ttw}$ emissions predicted by the rDNN for false positive layouts are reported in Table 25 (ascending order). In the table, the minimum predicted $CO_{2,TTW}$ values related to the true positive layouts are also reported for convenience ($CO_{2,TTW,min}$). Given that the layouts leading to the minimum emissions are typically considered as a guidance for a design optimization procedure, monitoring the $CO_{2,TTW}$ predicted for false positive layouts to not fall below the minimum of the true positive ones is a fundamental.

The results of the analysis on false positives show that only two layouts have shown a predicted CO_2 value under the minimum predicted one. In Table 25, the

two cases are highlighted in grey. Overall, the false positive layouts for which $CO_{2,TTW}$ values are lower than the minimum true positive-related CO_2 value represent the 0.07% of the total layouts analyzed (roughly the 0.22% of the total unfeasible layouts). Since the identification of the minimum emissions region is typically selected referring to the minimum emission values, the behaviour of the rDNN in case of false positive layouts can be considered as a promising outcome for the present application.

Table 25. Results of the analysis on false positives.

P2						
	I	***II***	***III***	***IV***	***V***	***VI***
$CO_{2,TTW,min}$ *g/km*	331.62	331.53	331.97	332.19	331.52	330.13
$CO_{2,TTW}$ *g/km*	338.93	-	338.28	346.66	345.11	342.37
	365.86	-	-	362.53	349.64	343.44
	-	-	-	441.07	377.57	349.68
	-	-	-	-	400.52	351.10
	-	-	-	-	-	357.13
	-	-	-	-	-	369.83
	-	-	-	-	-	396.27
P3						
	I	***II***	***III***	***IV***	***V***	***VI***
$CO_{2,TTW,min}$ *g/km*	333.81	334.09	332.89	333.38	334.91	333.57
$CO_{2,TTW}$ *g/km*	340.30	338.83	341.00	347.10	336.12	347.86
	343.99	355.30	344.03	347.47	338.42	-
	358.06	362.79	-	352.92	339.19	-
	387.31	-	-	-	342.44	-
P4						
	I	***II***	***III***	***IV***	***V***	***VI***
$CO_{2,TTW,min}$ *g/km*	364.07	365.14	362.41	366.93	366.32	364.30

	363.15	369.63	367.24	367.26	363.03	368.32
	367.19	382.43	373.92	367.37	367.97	369.95
	368.06	389.17	379.15	372.64	368.08	372.43
	372.25	-	382.74	377.32	368.20	374.76
$CO_{2,TTW}$	375.38	-	383.56	379.55	368.36	376.72
g/km	377.73	-	385.94	379.98	369.30	379.11
	379.25	-	388.03	380.59	370.53	380.38
	380.48	-	-	382.85	372.37	380.98
	381.08	-	-	391.06	383.40	382.41
	381.30	-	-	393.05	-	388.21

4.4 A pipeline of Deep Neural Networks for design optimization of hybrid electric fleets

As last step, the opportunity of integrating the DNNs-PM within a design optimization operation for HEV fleets has been analyzed and it is hereafter discussed. Specifically, the tool capability of spotting realistic low-CO_2 regions has been assessed for by comparing the pipeline predictions with respect to the results achieved by the DDP (i.e. labels). A reliable response of the DNNs-PM would turn into the confirmation of the model to be positively involved in a real-world energy-oriented design optimization for HEVs.

For the sake of conciseness, the outcomes produced by the DNNs-PM when applied to the P3 architecture have been taken into account. The CO_2 emissions related to the testing samples identified in Table 20 (*t/t split* 90/10) are reported in Figure 37. Specifically, Figure 37-a(1) and Figure 37-a(2) refer to the response surfaces obtained by DDP and DNNs-PM considering the CO_2 emissions with respect to V_{ICE} and $P_{max,MG}$, respectively. In Figure 37-b(1) and Figure 37-b(2), the CO_2 response surfaces are charted for DDP and DNNs-PM considering V_{ICE} *and* PE, respectively. It is worth highlighting that the choice of representing the CO_2 emissions with respect to V_{ICE}, $P_{max,MG}$ and PE is arbitrary and is only meant to exemplify the results. Two main considerations can be made looking at the results of Figure 37. First, the DNNs-PM effectively shows the capability of reproducing the shape of the CO_2 surfaces independently from the considered DVs. Second, the minimum CO_2 emissions region defined by the DNNs-PM correspond to the same region obtained by DP. Consistent with the physics behind the prediction task of HEVs-related emissions, the DNNs-PM generates response surfaces that privilege the influence of downsized ICEs and powerful MGs. According to both conditions, a HEV DP-based controller could contain the ICE utilization in favor of pure electric and power-split modes so to comply with the FC minimization target.

Finally yet importantly, the entire set of DNNs-PM predictions over the test sub-datasets occur instantaneously. In fact, only training and validation phases of the cDNN and rDNN require a relevant amount of time. Therefore, it could be deduced that a proper training of the DNNs-PM could be employed to realize multiple instantaneous tests over different datasets.

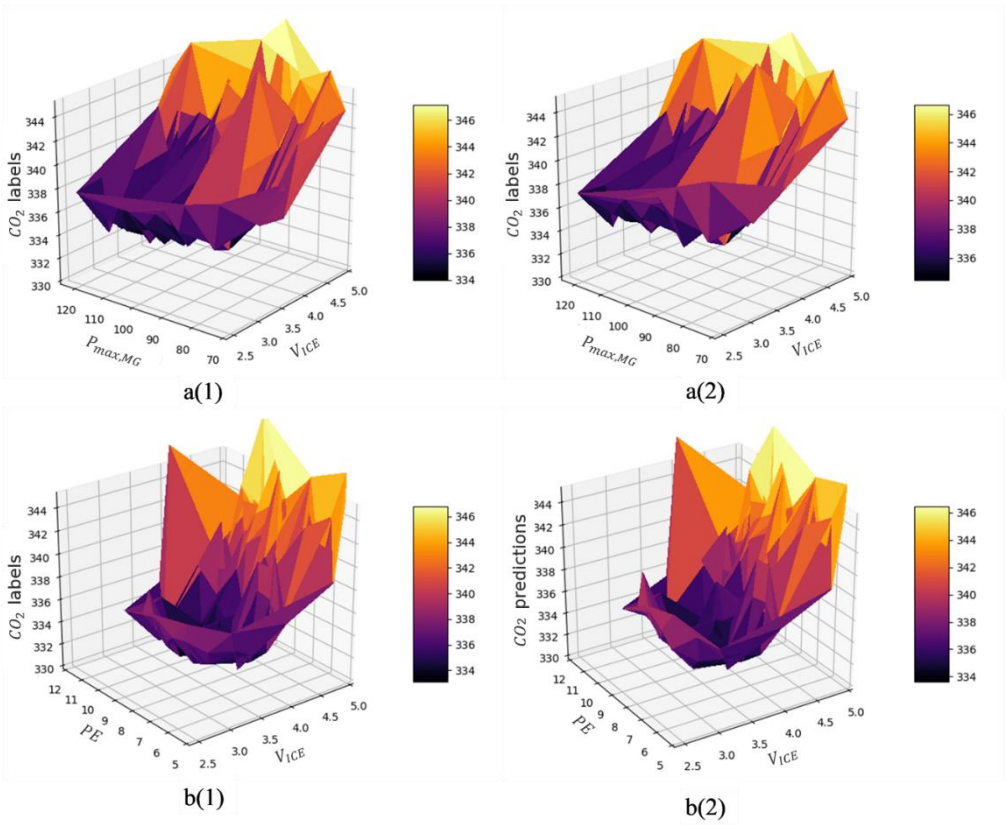

Figure 37. 3-D response surfaces obtained through DNNs-PM; a: CO_2 vs V_{ICE} and $P_{max,MG}$; b: CO_2 vs V_{ICE} and PE; (1): DP-based CO_2 labels; (2): DNNs-PM predictions.

4.5 Conclusions

In this Chapter, the capability of a pipeline made up of two DNNs has been proved by means of specific experiments aimed at assessing for the performances of both classification and a regression DNN, namely cDNN and rDNN, respectively. The main contributions of the study can be summarized as follows:

- The DNNs with the capability of catching the strongly non-linear correlations between the DVs of different HEV architectures and both the feasibility of the HEV layouts (classification task) and their related tank-to-wheel CO_2 emissions (regression task).
- Performance of the cDNN increasing with the number of layouts employed in the training procedure. The best results have been proved for the P2 dataset.
- The rDNN showing CO_2 predictions comparable to the target values (labelled data) with average errors lower than 0.5% for any feasible layout outputted by the cDNN.
- The capability of the model to limit the effect of false positive layouts. Only the 0.07% of the total layouts have been misclassified as false positive layouts by the cDNN. According to this very small percentage, the results obtained with the rDNN have not been affected.
- The demonstration of the opportunity to integrate the DNNs-PM within a design optimization operation for HEV fleets. In fact, it has shown the capability of instantaneously identifying low CO_2 emissions regions comparable with those obtained by a global optimization algorithm (such as DP).

The entire model (DNNs-PM) can hence be considered as an innovative solution for predicting both (P)HEV feasibility and emissions while drastically reducing the computational time required to complete an exhaustive design optimization procedure of (P)HEV fleets. Further research steps will involve the prediction of indicators about vehicle drivability and total cost of ownership. Furthermore, new hybrid architectures will be tested in order to enlarge the prediction possibility over a wider design domain and different driving missions will be integrated so that cycle-depending indexes could be analyzed.

Chapter 5

Learning from the decisions of dynamic programming through Long-Short Term Memory

In Chapter 4, the possibility of an AI-based approach to be embedded into the design optimization of (P)HEVs has been demonstrated with impressive results. From the present stage of the dissertation onwards, the performances of two different AI techniques for the real-time control of (P)HEVs are discussed.

Recalling the general formulation of the optimization problem proposed in Chapter 2 for (P)HEVs (see (27)-(29)), a robust EMS should be capable of minimizing the fuel consumed at the end of a given driving mission while guaranteeing a specific battery SOC trajectory. If the optimization is targeted under real-time condition, a real-time capable EMS turns into a necessary element (i.e. an EMS with the capability of being implemented on-board). To this end, AI-based controllers (see Figure 4) have proved to be promising solutions for in-vehicle implementation. In fact, immediate decisions could be performed online once a computationally expensive offline training process has been completed. The latter assumes a fundamental role in the design of an AI-based controller regardless of the specific type of algorithm (SL, UL, RL) but is not to be considered as a limitation for the application to real-world driving.

A clear choice can be made at this stage considering AI techniques for the real-time control of (P)HEVs. On one side, SL techniques could be taken into account

either for predicting the driving patterns ([88], [89]) or for learning and reproducing the decisions taken by another offline EMS in case of trip information ([91], [92], [94]-[99]). On the other, RL agents could be employed to directly learn the optimal control policy even without trip information through an intelligent trail-and-error approach ([109]-[117]).

In this Chapter, the potentials of a LSTM-based controller for HEVs to learn the optimal policy detected by a DDP algorithm and to reproduce it in a different testing environment are discussed. Indeed, a RL agent is examined later in Chapter 6.

5.1 From Neural Network to Long-Short Term Memory for hybrid electric vehicles real-time control

The design of a SL-based controller for hybrid powertrains generally involves the development and calibration of NNs ([87]). In fact, the effectiveness proved by NNs to perform accurate predictions in case of complex problems can be considered as an important opportunity to be explored even for (P)HEVs. Either as stressed in the introduction of the present Chapter, NNs have been exploited in the literature for the prediction of future trip conditions or actual control decisions only information about the driving mission. In fact, the a priori knowledge of the mission allows the NNs to be trained upon labeled data.

The most generic NNs exclude feedback connections in which the "model outputs are fed back into itself" ([44]). In the case of experimenting NNs for HEVs, any connection would hence be considered between the outputs of the prediction occurring at different stages (or time steps) of the mission. Therefore, classic NNs are typically trained to forecast future information about the trip based on a given set of variables representative of the driving conditions. When a direct connection is searched between the driving conditions and the optimal control decisions to be taken, additional considerations have to be made about the NNs. In fact, controlling a HEV can be clearly referred as a problem in which the output of a single decision might be influenced by a temporal sequence of inputs. Neglecting the actual control action not to be dependent from the past trajectory of the problem features is not coherent with real-world driving experience. For such a motivation, a RNN could be considered as a more consistent tool to face the real-time control problem since it has been designed properly for solving sequential data ([189]).

According to [190], a RNN is characterized by the capability of mapping a history of previous inputs to each output. A "memory" is created by means of a loop mechanism that allows the information ("hidden state") of a given time step to be passed to the following time step(s). For additional details about the workflow of RNNs, please refer to [190]. Nevertheless, RNNs are affected by the "vanishing gradient" phenomenon, for which temporally distant events appear to not be useful for the model training ([191]). Since the duration of real-world trips could significantly vary and long driving missions could be experimented, considering the RNN does not appear as the most promising solution for an HEV AI-based EMS.

Fortunately, variants of RNNs have been proposed in the literature to overcome the vanishing gradient problem, namely LSTMs ([192]) and Gated Recurrent Units (GRUs) ([193]). The latter can be considered as a specialized variant of LSTMs. The main contribution brought by both LSTMs and GRUs is represented by the possibility of obtaining information even from temporally distant stages of an episode thanks to specific mechanisms called "gates". For the sake of clarity, even if LSTMs and GRUs represent two similar approaches, only LSTMs are discussed in this Chapter since they have been applied to the test case of HEV real-time control.

In a LSTM, two different memories could be distinguished, namely "long-term memory" and "working memory". The long-term memory and the working memory are typically referred to the "cell state" and the "hidden state", respectively. Compared to RNNs, an internal self-loop is introduced in the LSTM cells by means of a smart update of the cell states. Then, the classic outer loop from one cell to the following is realized considering both the cell and hidden states ([44]). The structure of a LSTM cell is illustrated in Figure 38. Three different gates are introduced in a LSTM cell, namely "forget gate", "input gate" and "output gate", which are featured by neuron with a given activation function. First, the forget gate (also called "remember vector") is responsible of telling the cell state which information from the previous cell state (c_{t-1}, h_{t-1}) as well as from the actual input features X_t should be considered or forgotten. Then, the input gate defines which information from the actual input should be considered or forgotten. The outputs of forget gates f_t and input gates i_t are then combined to update the cell state output c_t to be passed to the next LSTM. Finally, the output gate sets the information h_t to be passed to the next LSTM hidden state by operating on c_t and o_t. For deeper insights about the calculations specifically performed in the cells, please refer to [44].

In the following sections, the most relevant elements considered for the application of an LSTM to the energy management of a HEV are first presented. Then, the main results obtained for a *hdv* are reported considering two different test cases.

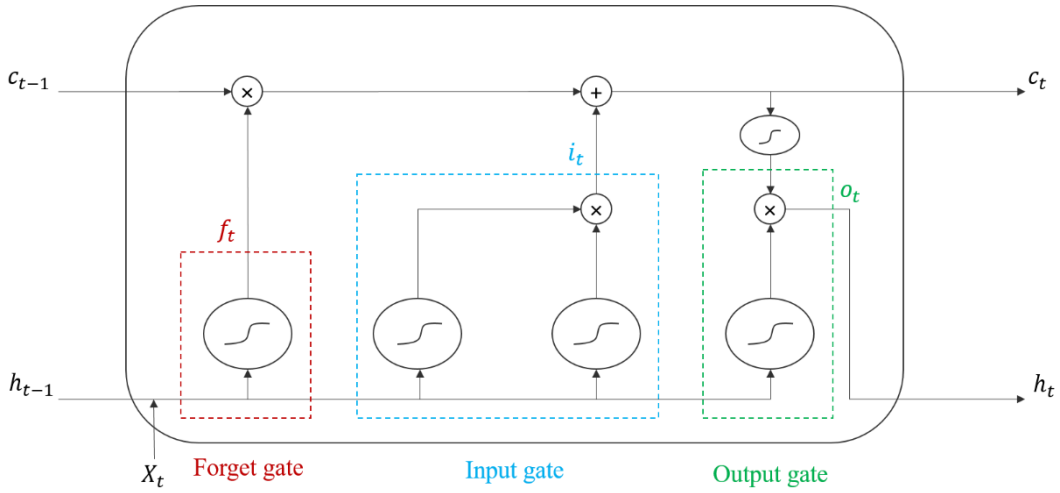

Figure 38. Internal structure of a LSTM cell.

5.1.1 Definition of the test cases

Two different studies are presented in the following sections, which highlight the performances of LSTM-based controllers when aimed at learning the decisions of benchmark optimizers. First, the GNI selection of a generic vehicle has been considered as a stand-alone control problem. In this case, the AI-based controller has been trained contemplating an optimized GNI schedule of a conventional powertrain. Such a reduced problem has been handled as a preliminary test to check the potentials of an LSTM-based EMS for application to vehicles. Then, the more complex control problem involving the combination of GNI and PFI for a HEV has been targeted. In this case, two LSTM-based controllers have been developed in series, the first responsible of predicting GNI while the second responsible of predicting PFI. Still, the modeling approach presented in Chapter 2 has been employed to generate the HEV model to be simulated and the DDP with an optimized mesh of Chapter 3 has been considered for the identification of the benchmark control policy.

For the sake of simplicity, the cases of GNI stand-alone control and combined GNI-PFI control will be referred to as “GN-Net” and “DP-Net”, respectively.

5.1.2 Feature extraction

The input features of the LSTM-based controllers have to be defined according to the information that could be acquired on-board under real-time conditions. Considering a backward-facing modeling approach, temporal and spatial coordinates could be used to gain information about the road requests (e.g. velocity, acceleration, etc.). Furthermore, in-vehicle available data (e.g. battery SOC, GNI, etc.) could be exploited for ensuring a wider knowledge to the EMS about the system to be controlled. In the present dissertation, the operation of obtaining information from the environment is referred as "feature extraction".

For the sake of clarity, a massive testing campaign has been performed considering different sets of features aimed at assessing for the best performances achieved by the LSTM-based controllers. Considering the two different test cases presented in Section 5.1.1, two distinct feature extractions have been designed. As far as GN-Net is concerned, the set of features has been identifying based on a wide set of variables presented in [194]. In the latter, the most influent signals for the identification of the driving pattern have been discussed. On the other hand, in-vehicle information have also been included for DP-Net.

The final sets of features considered for GN-Net and DP-Net are reported in Table 26. As it can be noted, the feature extraction of DP-Net stems from the same dataset of GN-Net while adding five other signals. The latter comprise the gear predicted by the first LSTM, information about the battery SOC and the percentage of time left to conclude the driving mission. It is worth observing that the same set of features used to train the AI-based EMS have to be considered in the testing phase. Therefore, the possibility of adopting the presented LSTM controller for (P)HEVs under real-time conditions embeds the necessity of having a sort of knowledge about the upcoming trip. In other words, a GPS should be involved in a real-world utilization of the tool.

Table 26. Feature extraction for GN-Net and DP-Net.

GN-Net	DP-Net
Actual velocity, *km/h*	Actual velocity, *km/h*
Actual velocity variation, *km/h*	Actual velocity variation, *km/h*
Traction/braking, *bool*	Traction/braking, *bool*
Share of idle time, *%*	Share of idle time, *%*
-	Predicted GNI, -
-	Actual battery SOC, -
-	Battery SOC variation from previous time step, -
-	Difference between actual battery SOC and SOC^*, -
-	Share of mission time left, *%*

5.1.3 Training and testing environments

The classic operation of every AI algorithm comprises two main phases, namely training and testing. In addition, a validation phase can be considered when an optimization of the main hyper-parameters has to be performed (see Chapter 4). In the research activity presented in this Chapter, a strong assumption has been made about the composition of the training and testing datasets seen by the AI controller. Given the more general operation of a HEV could involve the possibility of dealing with different environments, the test of a LSTM-based EMS on the same training data has been considered as a non-innovative study. Therefore, training and testing conditions have been clearly diversified so as that the real generalization capability of the controller could be assessed on new and unknown data. Such an operation relies upon the same considerations presented in Chapter 4 regarding the dataset split in training, validation and testing sub-datasets for a DNN.

In the results, the specific training and testing datasets used for GN-Net and DP-Net are clarified.

5.1.4 Architecture of the LSTM controllers

The architectures of GN-Net and DP-Net are reported in Figure 39 and Figure 40, respectively. Consistent with the test cases presented in Section 5.1.1, DP-Net embeds GN-Net for the prediction of GNI, which is hence used as an additional input of the network. Specifically, four inputs d_{1-4} and six outputs GNI_{1-6} are considered for GN-Net (Figure 39), whereas four additional inputs d_{5-8} and seven outputs PFI_{1-7} are considered for DP-Net (Figure 40). The set of inputs refers to the features listed in Table 26 while the outputs refer to GNI and PFI of Table 8 (Chapter 3).

Between the input and output layers, three hidden layers have been built. In both cases, a hierarchical structure ([195]) has been developed with a dense layer (i.e. the most typical layer of a NN made of classic neurons) following two LSTM layers. The dense layer has been introduced due to the necessity of coupling the dimensionality of the last hidden and the output layers. In fact, the dimensionality of the second hidden LSTM layer is not equal to the number of outputs, namely six and seven for GN-Net and DP-Net, respectively. Therefore, an additional dense layer with "softmax" activation functions ([196]) has been involved in the architecture.

Consistent with the networks' architectures, the configurations used for the two controllers are reported in Table 27. The hierarchical structure has been realized by means of a decreasing pyramidal number of elements in the hidden layers (128-64-32). The cross entropy has been considered as the most appropriate loss function of the multi-classification task ([197]). Adam and Dropout (0.2) have been considered as optimization and regularization methods, respectively. Finally, a different number of epochs has been set given the greater complexity of the problem to be solved by DP-Net. As noted in Chapter 4, please consult [175] for additional details about characteristics and specifications about NNs.

In the following Section, the results obtained by the application of both GN-Net and DP-Net to a specific vehicular application are presented.

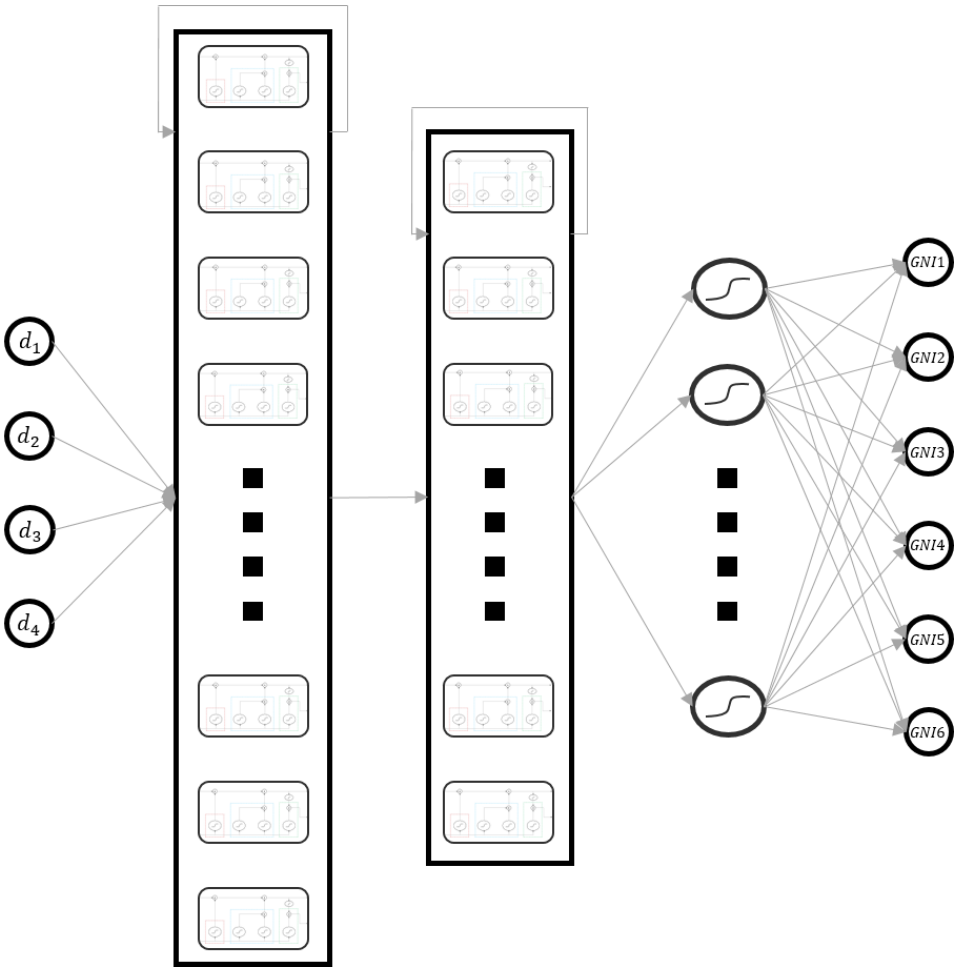

Figure 39. GN-Net architecture.

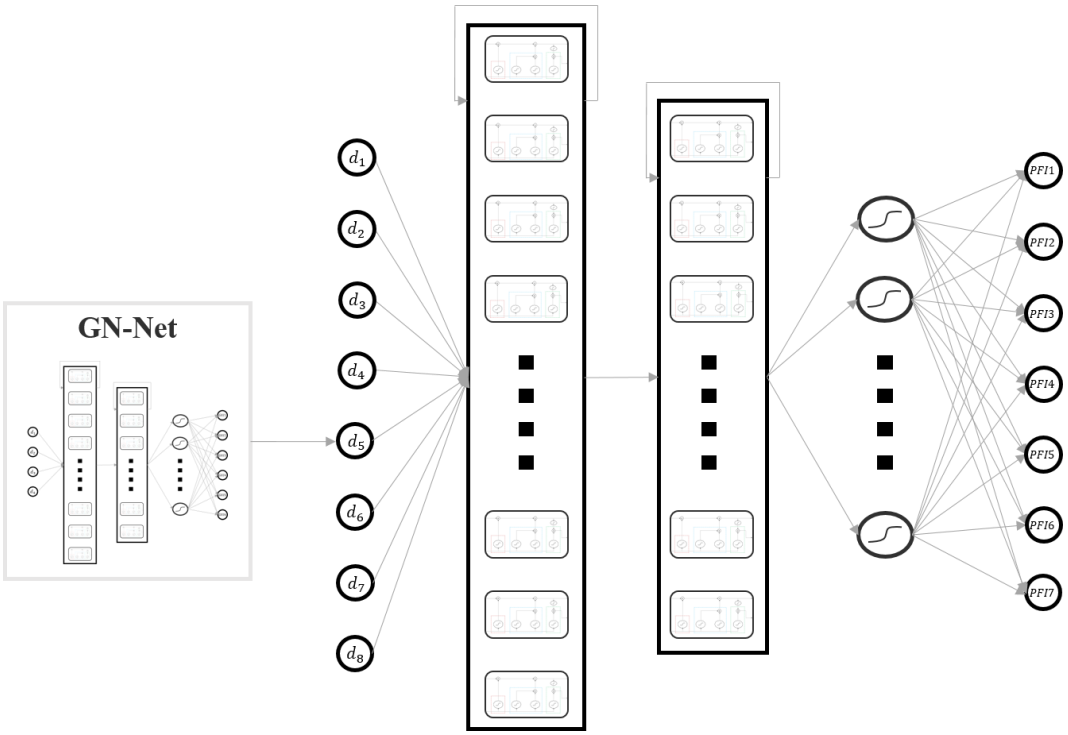

Figure 40. DP-Net architecture.

Table 27. Configurations of GN-Net and DP-Net.

Item	GN-Net	DP-Net
Number of cells in the first LSTM layer	128	128
Number of cells in the second LSTM layer	64	64
Number of neurons in the dense layer	32	32
Optimization	Adam	Adam
Regularization	Dropout 0.2	Dropout 0.2
Loss function	Cross entropy	Cross entropy
Number of epochs	100	300

5.2 Results

The LSTM-based controllers for (P)HEVs have been tested relying upon the considerations made in the previous sections and the main results are hereafter reported for a *hdv* application. Specifically, a post-transmission parallel heavy-duty HEV architecture has been modeled (Chapter 4, Figure 29) according to the the vehicle specs of the baseline vehicle reported in Table 13 and to the backward-facing modeling approach presented in Chapter 2. Specifically, the DVs presented in Table 1 have been considered for the HEV layout, including the ICE displacement V_{ICE}, the MG peak power $P_{max,MG}$, the battery power-to-energy ratio PE, the torque-coupling device and final drive speed ratio (sr_{TCD}, sr_{FD}) and the maximum C-rate during battery discharging and charging operations ($C_{dis,max}$, $C_{ch,max}$).

For the sake of clarity, the *hdv* layout is not representative of a real vehicle already placed on the market. In fact, a generic high-performance design has been accounted for testing the AI-based EMS that could minimize the insurgence of unfeasibilities related to a specific downsizing of the powertrain main components. In fact, a qualitative assessment of the performances achieved by LSTMs when applied to a HEV control problem has been targeted instead of an optimization of the controller for a given application.

Table 28. Design of the post-transmission parallel hybrid powertrain considered for a heavy-duty application.

V_{ICE}	$P_{max,MG}$	PE	sr_{TCD}	sr_{FD}	$C_{dis,max}$	$C_{ch,max}$
l	*kW*	*kW/kWh*	-	-	*A/Ah*	*A/Ah*
5	125	10	5	3.6	7.5	7.5

Distinct driving missions have been considered to build the training and testing datasets of the LSTM-based controllers (see Section 5.1.3). Particularly, type-approval and chassis dynamometer procedures have been used to build the training datasets whereas real-world driving missions have been considered to test controllers. The WHVC has been considered as stand-alone driving cycle to train GN-Net, whereas a miscellaneous training dataset has been considered for DP-Net,

which consists of WHVC, ETC and capped variants of ARDC and AUDC. In Figure 41 and Figure 42, the velocity trajectories of the capped ARDC and AMDC are reported, respectively. The saturation of the velocity profiles has been realized by checking the maximum velocity and acceleration achieved by the HEV *hdv* with the layout of Table 28. As far as the testing datasets are concerned, the velocity trajectories of the two experimentally derived real-world driving missions used to test the LSTM controllers, namely CLUSTa and CLUSTb, are charted in Figure 43 and Figure 44, respectively. Two very different driving missions highlight from the charts that are representative of mixed urban and extra-urban driving conditions with very different driving length and duration. Consistently, the diversity between the two driving mission has been considered as a reliable assumption to test the controllers under variable conditions.

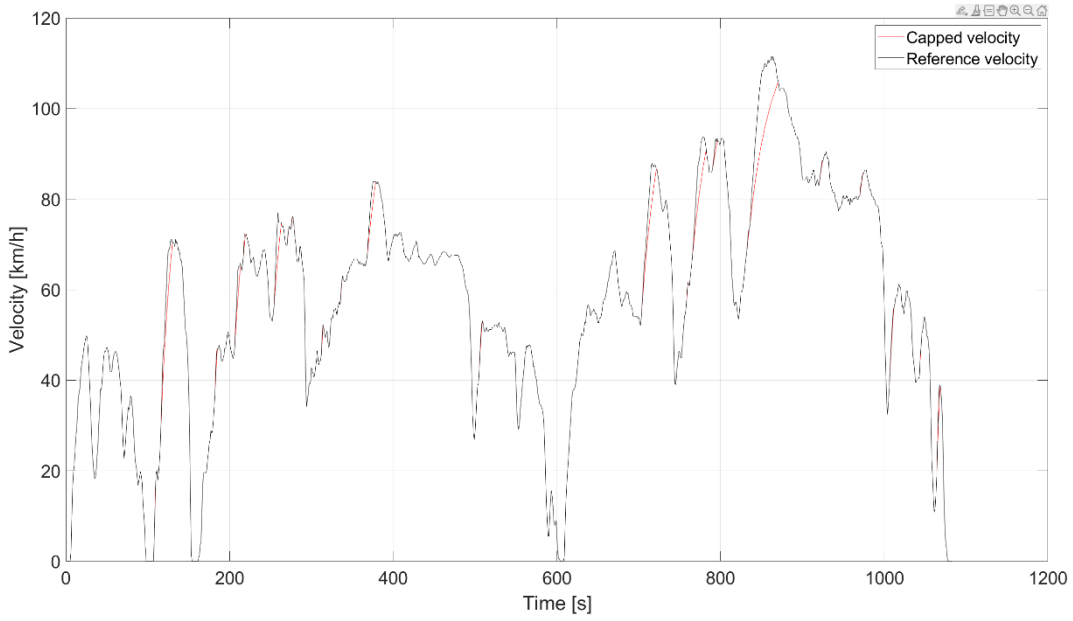

Figure 41. Comparison between reference and capped velocity profiles of the ARDC.

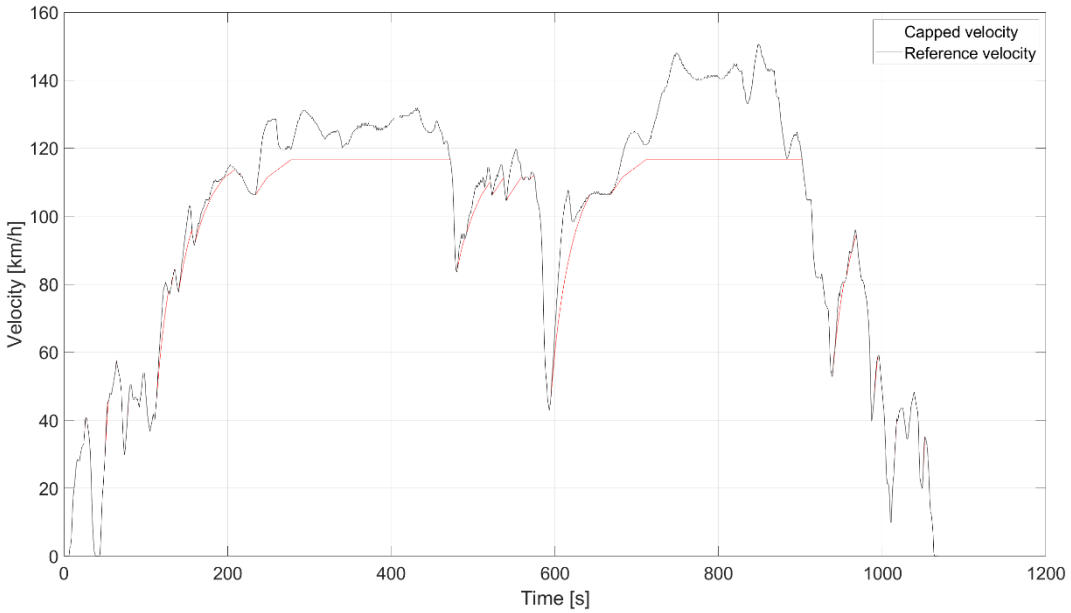

Figure 42. Comparison between reference and capped velocity profiles of the AMDC.

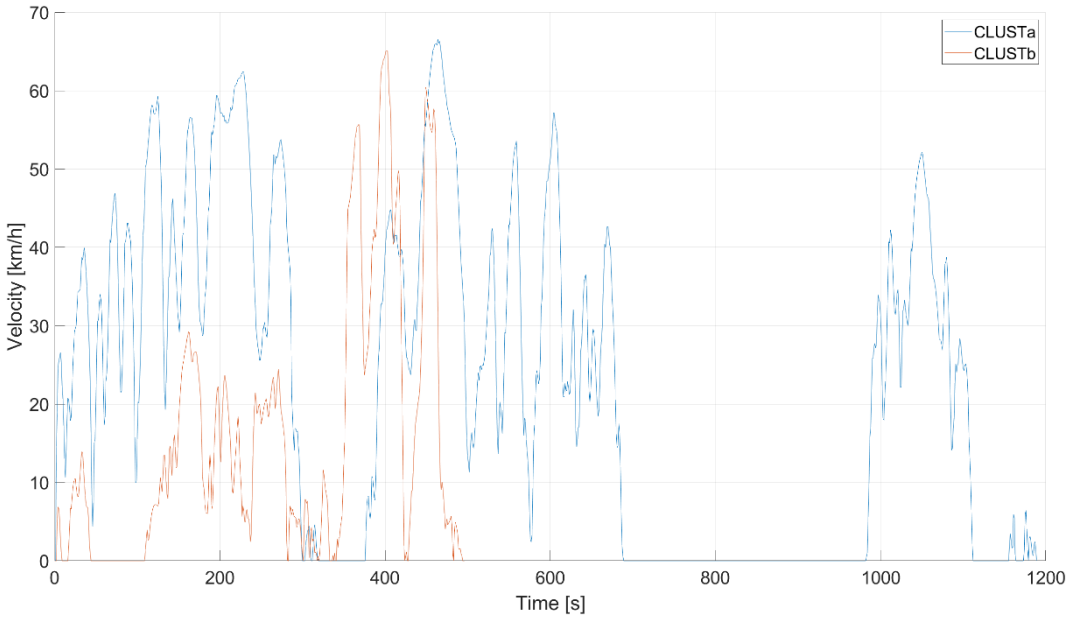

Figure 43. Velocity trajectories of CLUSTa and CLUSTb with respect to the driving mission time.

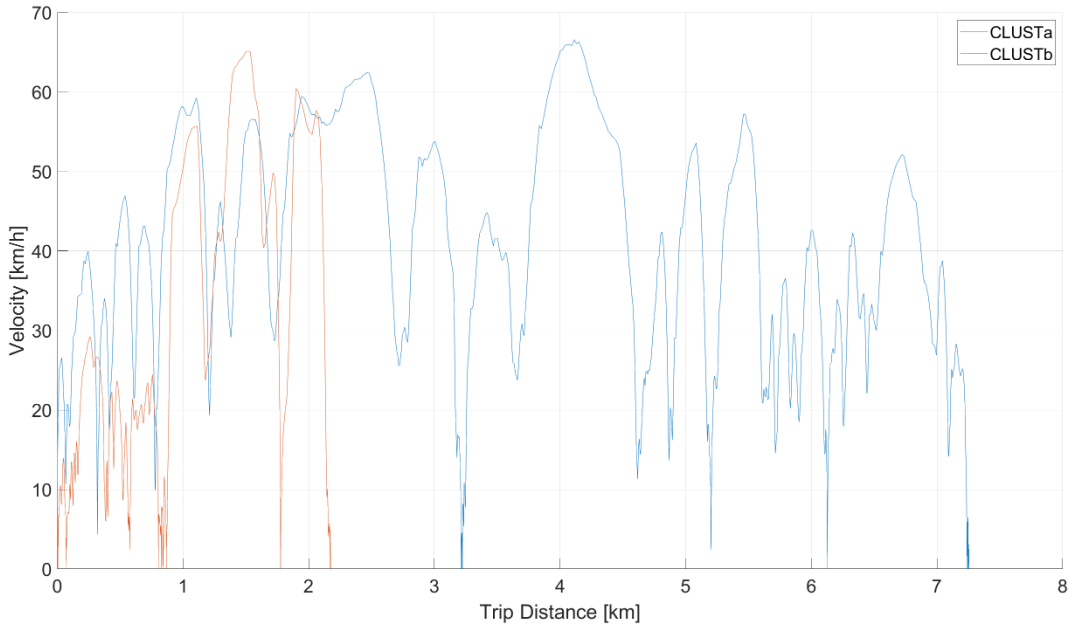

Figure 44. Velocity trajectories of CLUSTa and CLUSTb with respect to the trip distance.

5.2.1 GN-Net for conventional vehicles

As far as GN-Net is concerned, the performances of the controller have been assessed for by means of a specific set of experiments on a conventional version of the *hdv* presented in Section 5.2. Specifically, the powertrain has been designed considering the values of V_{ICE} and sr_{FD} reported in Table 28. Recalling Section 5.1.1, the benchmark control trajectory has been identified as optimized gear-shifting schedule obtained by an offline optimizer for conventional vehicle aimed at minimizing the cumulative fuel consumption at the end of the driving mission (WHVC). Once the optimal control chain has been identified, the training dataset for the LSTM controller has been generated combining the input features identified in Table 26 and the GNI outputs.

In Figure 45 and Figure 46, the GNI sequences obtained by the benchmark strategy (black circles) and by GN-Net (red crosses) on CLUSTa and CLUSTb are charted, respectively. Consistent with the testing process of AI algorithms, the outputs of GN-Net on the testing missions have been obtained without providing the network with any knowledge about the optimal GNIs selected by the benchmark strategy. The similarity between the GNI trajectories of the optimal gear-shifting

schedule and GN-Net (ΔGNI) are reported in Table 29 for both CLUSTa and CLUSTb considering four different cases: optimal choices ($\Delta GNI = 0$), near-to-optimal choices ($abs(\Delta GNI) = 1$), far-from-optimal choices ($abs(\Delta GNI) = 2$) and completely failed choices ($abs(\Delta GNI) > 2$). Optimal choices have been identified for the majority of the missions (68.4% and 72.2% over CLUSTa and CLUSTb, respectively), followed by near-to-optimal choices (26.1% and 26.4% over CLUSTa and CLUSTb, respectively). Promisingly, far-from-optimal and completely failed choices sum up only for 5.5% and 4.4% on CLUSTa and CLUSTb, respectively. To complete the comparison, the cumulative FCs related to the control decisions of optimal gear-shifting schedule and GN-Net on CLUSTa achieve 1.85 kg and 2.04 kg (+10.3%), respectively. On CLUSTb, the same controllers lead to 0.72 kg and 0.76 kg (+9.4%). Considering the conversion factor of The fuel consumed at the end of the missions based on the decisions of GN-Net is hence not extremely different from the fuel consumed with the benchmark strategy even if "only" an average 70% of optimal choices has been identified. Such an even can be referred to the large share of near-to-optimal choices with respect to the shares of far-from-optimal and completely failed choices. In fact, selecting a near-to-optimal gear might not lead to significant differences in the working operation of the propeller.

Given that a very small training dataset has been passed to GN-Net (only a single driving mission), the results obtained on the two testing environments can be considered as promising outcomes about the potentials of employing LSTMs for a real-time capable vehicular controller. In the following Section, the results obtained with DP-Net are discussed for the hybrid version of the *hdv* considered so far.

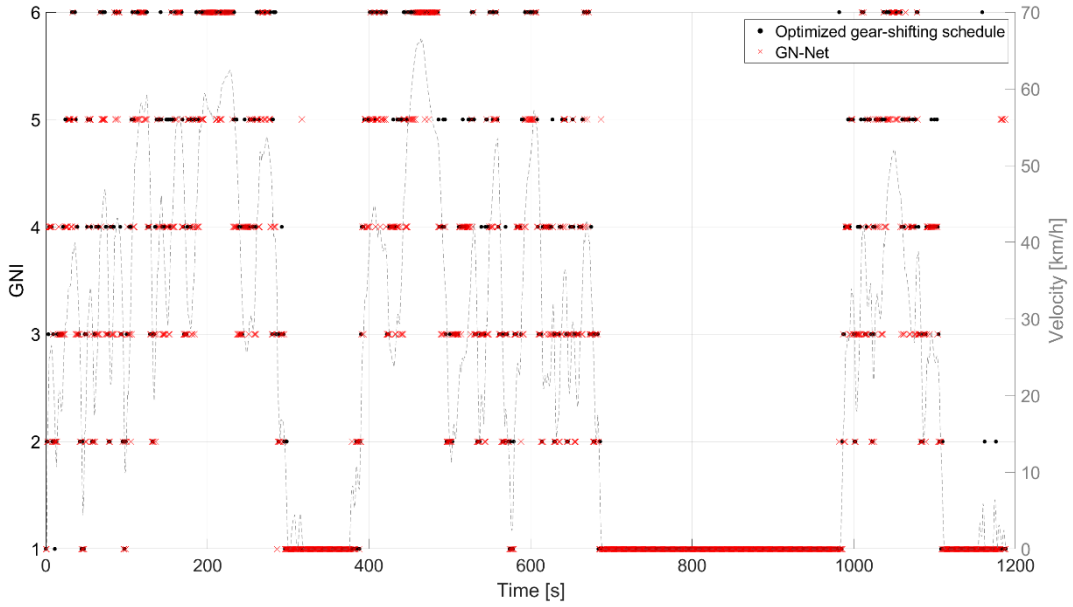

Figure 45. Comparison between the gears selected by the optimized gear-shifting schedule and by GN-Net on CLUSTa.

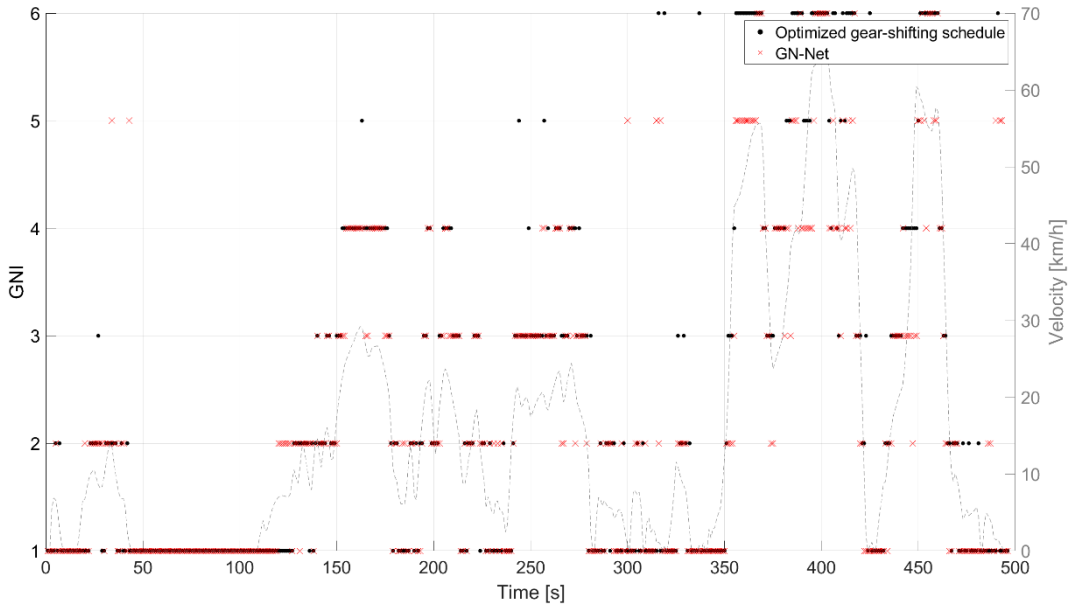

Figure 46. Comparison between the gears selected by the optimized gear-shifting schedule and by GN-Net on CLUSTb.

Table 29. Differences between the GNI outputs obtained by the optimized gear-shifting schedule and by GN-Net on the testing missions.

Case	Relative share – CLUSTa %	Relative share – CLUSTb %
$\Delta GNI = 0$	68.4	72.2
$abs(\Delta GNI) = 1$	26.1	22.4
$abs(\Delta GNI) = 2$	4.8	2.8
$abs(\Delta GNI) > 2$	0.7	2.6

5.2.2 DP-Net for hybrid powertrains

Consistent with the results presented for GN-Net, similar experiments have been carried out considering DP-Net for a HEV application. Nevertheless, a different set of input features (Table 26) and training driving missions (including WHVC, ETC, capped ARDC and capped AMDC) has been taken into account. Two motivations can be referred to this choice. First, a more complex problem is faced (combination of GNI-PFI). Second, the results obtained with GNI suggest that LSTM-based controller could take advantage from a wider training configuration. Recalling Section 5.1.1, the DDP formulated in Chapter 2 has been identified as benchmark strategy to train the network and compare the control trajectories over the testing missions.

In Figure 47 and Figure 48, the GNI and PFI trajectories obtained by DDP (black circles) and DP-Net (red crosses) over CLUSTa are charted, respectively. Moreover, the battery SOC obtained with the two controllers are presented in Figure 49. As it can be noted, very similar battery SOC trends highlight for both DDP (black curve) and DP-Net (red curve) based on the relative control chains. Consistent with the purpose of AI-based techniques to learn and reproduce the behavior of benchmark data, DP-Net is capable of sustaining the battery SOC even without an explicit formulation of the problem. In Table 30, the similarities between the choices performed by the benchmark and by DP-Net are reported in terms of GNI (ΔGNI) and PFI (ΔPFI). An immediate consideration can be made comparing the GNI results obtained by GN-Net and DP-Net: the latter proves to be relevantly more accurate in classifying the optimal GNI with respect to GN-Net. Given that the same input features have been involved in the training phase of the first LSTM

network of DP-Net, the results amelioration can be referred to the enlargement of the training dataset. In other words, increasing the number of driving missions in the training dataset allows for a more solid training of the AI algorithm.

An additional check has been carried out considering the multiple operating modes realized by a parallel HEV architecture. In fact, given the possibility of adopting multiple *ps* and *bc* levels (Table 8), the absolute difference between the PFIs selected by the algorithms could theoretically be misleading for the assessment of the real learning progress. Therefore, the relative share of utilization and the related energy consumption has been reported in Table 31 for both DDP and DP-Net on CLUSTa. Promising performances highlight for DP-Net as comparable share of utilization and energy consumption are achieved for any operating mode. In fact, the largest differences peak at 1.2% and 0.2 kWh for the relative share and the energy consumed, respectively. For the sake of clarity, the same values are obtained by DDP and DP-Net for *pe* during braking since no influence of the EMS is realized during the regenerative braking phases of the mission.

Regarding the cumulative FC realized through the control policies of the controllers, an additional FC should be considered for a fair comparison that account for the difference between the two final battery SOC values. Nevertheless, given the quasi-perfect CS trajectory attained with DP-Net in Figure 49 (final battery SOC equal to 0.598), a negligible fuel consumption should have been added. For the sake of simplicity, no adjustment has been considered for the final FC value. Particularly, 0.99 kg of fuel has been obtained at the end of CLUSTa with DP-Net while the DDP optimizer has realized 0.97 kg. Therefore, DP-Net has been capable of closing the FC gap from the benchmark strategy to roughly +2%. Considering the results of GN-Net for CLUSTa presented in the previous Section (+10.3%), a very interesting step ahead has been realized by DP-Net in learning the optimal control policy.

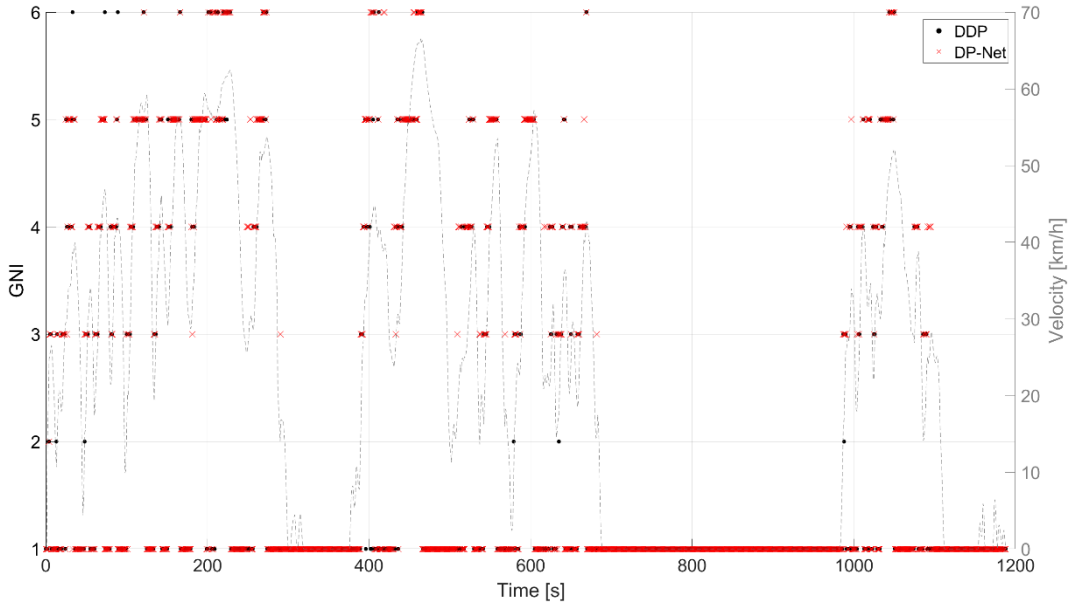

Figure 47. Comparison between the gears (GNI) selected by DDP and DP-Net on CLUSTa.

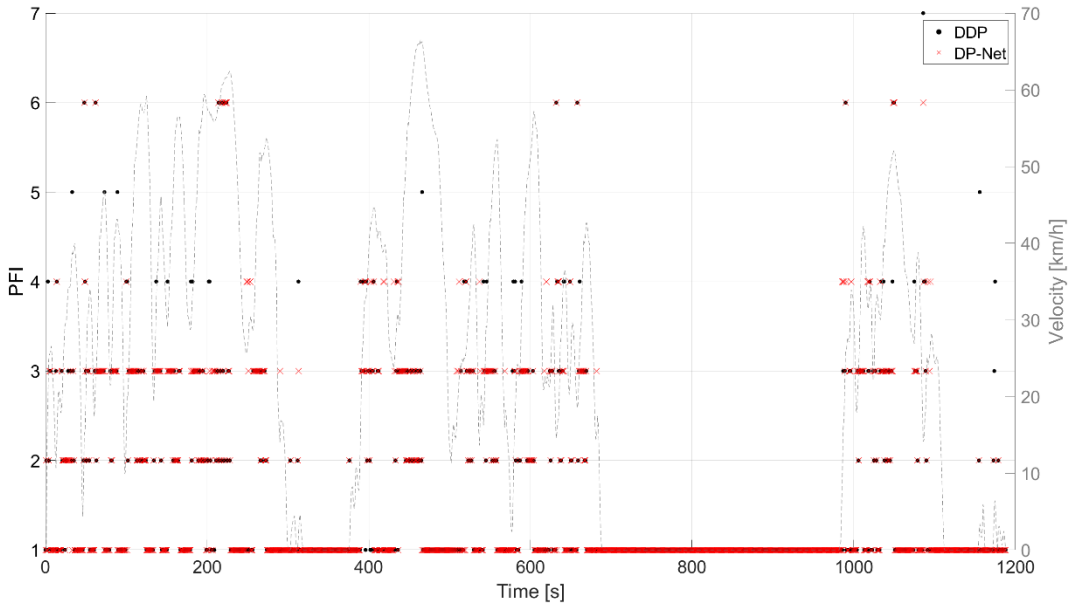

Figure 48. Comparison between the operating modes (PFI) selected by DDP and DP-Net on CLUSTa.

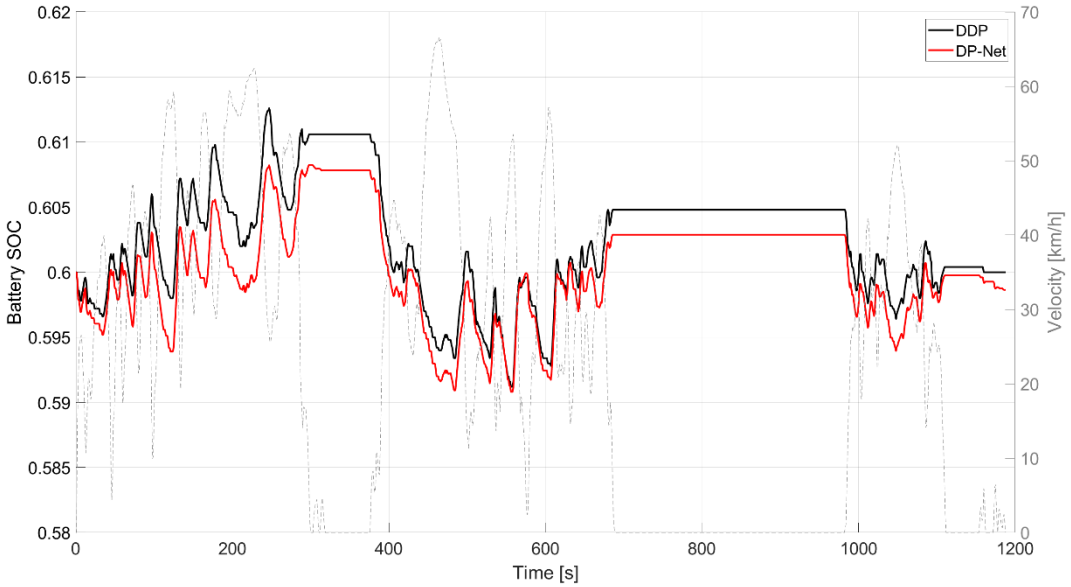

Figure 49. Comparison between the battery SOC trajectories related to the control decisions of DDP and DP-Net on CLUSTa.

Table 30. Differences between the GNI and PFI outputs obtained by DDP and DP-Net on CLUSTa.

GNI		PFI	
Case	**Relative share** $\%$	**Case**	**Relative share** $\%$
$\Delta GNI = 0$	92.7	$\Delta PFI = 0$	93.3
$abs(\Delta GNI) = 1$	3.8	$abs(\Delta PFI) = 1$	2.4
$abs(\Delta GNI) = 2$	2.6	$abs(\Delta PFI) = 2$	1.5
$abs(\Delta GNI) > 2$	0.9	$abs(\Delta PFI) > 2$	2.8

Table 31. Share of utilization and energy consumption related to each operating mode realized by DDP and DP-Net on CLUSTa.

Operating mode	DDP		DP-Net	
	Relative share *%*	Energy consumption *kWh*	Relative share *%*	Energy consumption *kWh*
pe (traction)	9.4	0.5	8.2	0.3
pe (braking)	60.6	-3.0	60.6	-3.0
pt	9.2	1.5	9.4	1.5
ps	19.5	3.1	20.6	3.3
bc	1.2	0.1	1.2	0.1

As far as CLUSTb is concerned, the same experiments have been carried out aimed at comparing the control sequences obtained by DDP and DP-Net as well as the relative battery SOC trends and cumulative FCs. The trajectories of GNI and PFI obtained on CLUSTb are charted in Figure 50 and Figure 51, respectively, while the battery SOC curves are compared in Figure 52. The numerical comparison between the decisions of the controllers are reported in Table 32 and Table 33 considering the GNI-PFI combination and the distinction between different operating modes, respectively. Two considerations can be made looking at charts and tables. First, the battery SOC curves appear to be more distanced one from the other with respect to the same curves obtained on CLUSTa (Figure 49). Nevertheless, a very interesting final battery SOC value is realized by DP-Net at the end of the driving mission (0.595). Second, a robust learning process is highlighted both by the actual differences between the GNI and PFI outputs (Table 32) and by the comparison of the share of utilizations and the energy consumptions related to the different operating modes (Table 33). In fact, the largest share of GNI and PFI is perfectly predicted by DP-Net (95.8% and 94.1%, respectively). Moreover, interesting performances are detected in the analysis of the operating modes as the same energy consumptions are obtained for each PFI with restrained errors in the relative share of utilization (peak at 1.6% for *pe* during traction).

As final comparison, the cumulative FC realized by the controllers has been analyzed without adding the negligible amount of fuel related to the quasi-perfect CS trajectory of DP-Net. For CLUSTb, 0.33 kg and 0.32 kg have been realized by DP-Net and DDP, respectively, resulting in an increase of +3.1% in case of DP-Net.

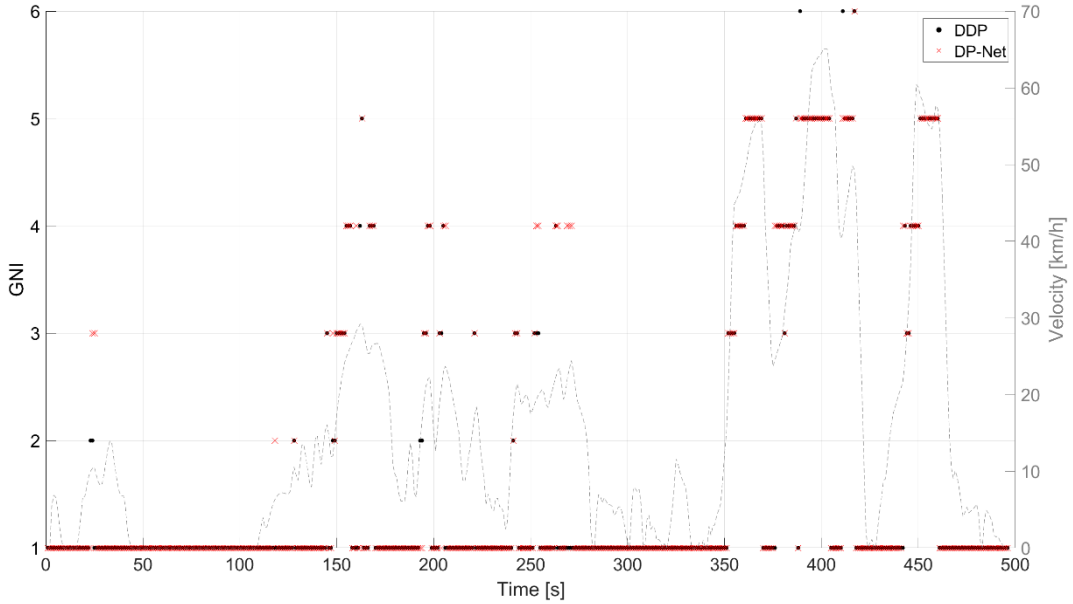

Figure 50. Comparison between the gears (GNI) selected by DDP and DP-Net on CLUSTb.

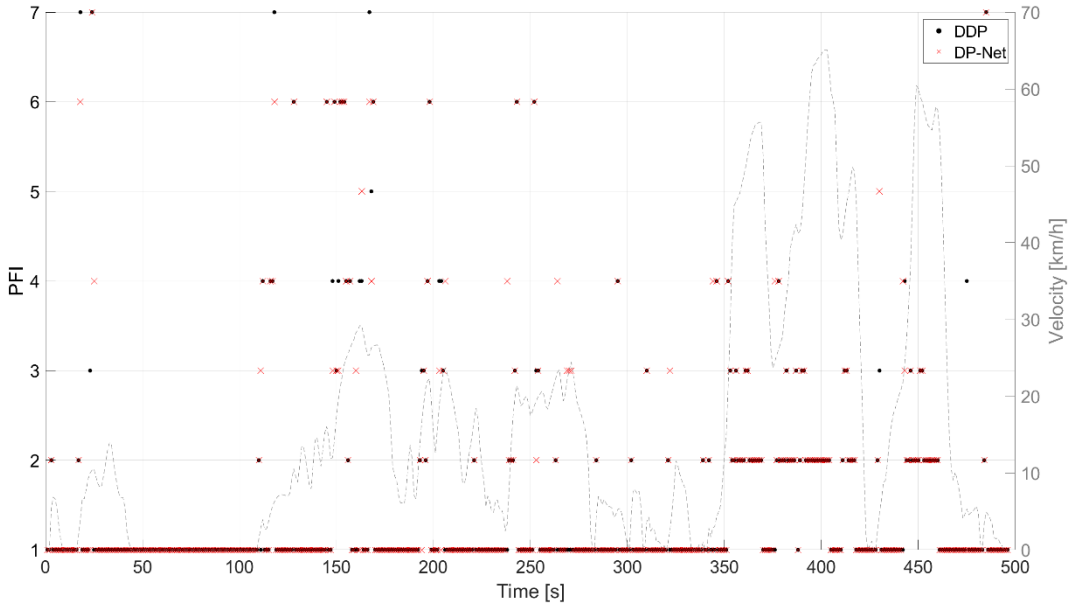

Figure 51. Comparison between the operating modes (PFI) selected by DDP and DP-Net on CLUSTb.

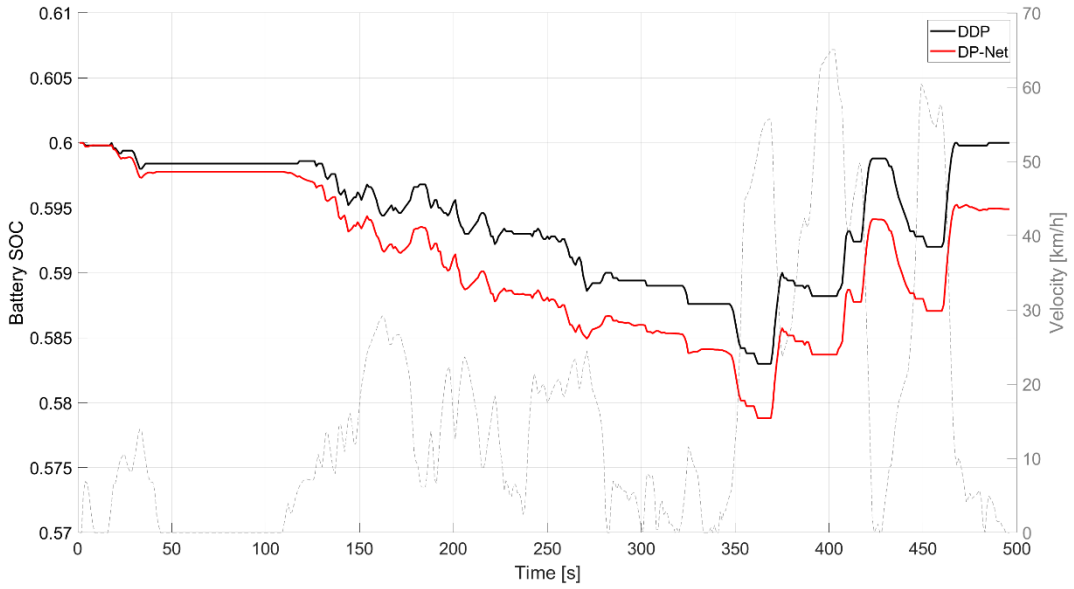

Figure 52. Comparison between the battery SOC trajectories related to the control decisions of DDP and DP-Net on CLUSTb.

Table 32. Differences between the GNI and PFI outputs obtained by DDP and DP-Net on CLUSTb.

GNI		PFI	
Case	**Relative share** *%*	**Case**	**Relative share** *%*
$\Delta GNI = 0$	95.8	$\Delta PFI = 0$	94.1
$abs(\Delta GNI) = 1$	2.0	$abs(\Delta PFI) = 1$	2.0
$abs(\Delta GNI) = 2$	0.4	$abs(\Delta PFI) = 2$	1.8
$abs(\Delta GNI) > 2$	1.8	$abs(\Delta PFI) > 2$	2.0

Table 33. Share of utilization and energy consumption related to each operating mode realized by DDP and DP-Net on CLUSTb.

Operating mode	**DDP**		**DP-Net**	
	Relative share *%*	**Energy consumption** *kWh*	**Relative share** *%*	**Energy consumption** *kWh*
pe (traction)	23.2	0.3	21.6	0.3
pe (braking)	50.8	-1.1	50.8	-1.1
pt	14.5	0.9	14.7	0.9
ps	8.5	0.4	9.9	0.4
bc	3.0	0.1	3	0.1

5.3 Conclusions

In this Chapter, the potentials of interpreting the real-time control problem of a *hdv* HEV by means of a SL problem involving a particular family of RNN, namely LSTM, have been thoroughly discussed. Preliminarily, a real-time capable gear-shifting controller based on a LSTM-based network, namely GN-Net, has been developed and trained upon a small dataset involving an optimized gear-shifting schedule of a conventional powertrain on the WHVC cycle. Four input features have been extracted from on-road available signals related to the road velocity and acceleration requests. Therefore, a real-time capable EMS for HEVs has been developed, namely DP-Net, assembling two different LSTM networks in series, the first predicting the optimal gear number (GN-Net) and the latter predicting the optimal operating mode. In this case, the training dataset has been enlarged to consider multiple driving missions, specifically WHVC, ETC and two capped variants of ARDC and AMDC. Moreover, four additional input features have been added to the training process of the LSTM network for the classification of the operating mode, including the gear predicted by GN-Net and on-board available information about the battery SOC.

The results of the application of both GN-Net and DP-Net have been realized considering two experimentally derived real-world driving missions as testing datasets, namely CLUSTa and CLUSTb. The main outcomes of the testing phase can be summarized as follows:

- The potentials of replicating a benchmark gear-shifting sequence of a conventional powertrain has been assessed by GN-Net. The latter has proved to be capable of realizing roughly 70% of 24% of optimal and near-to-optimal choices, respectively, averaged between CLUSTa and CLUSTb.
- The gear-shifting sequences of GN-Net have led to an average FC increase of roughly 9.8% with respect to the FC realized by the benchmark strategy. Given the very small dataset involved in the training process of GN-Net, the results have been considered as sufficiently promising.
- The capability of DP-Net to reproduce the behaviour of a DDP algorithm has been assessed considering a broader training configuration. The LSTM-based controller has proved to be capable of reproducing the optimal gears and operating modes identified by DDP with impressive results. In fact, optimal gear numbers and optimal

operating modes have been classified more than roughly 94% averaged between CLUSTa and CLUSTb. Moreover, DP-Net has proved to be capable of also accurately replicating share of utilization and energy consumption of each operating mode of a parallel HEV (pure electric, pure thermal, power-split and battery charging). In this case, the worse classification errors for the share of utilization and the energy consumption among the entire set of experiments has peaked at 1.6% and 0.2 kWh, respectively.

- The gear-shifting and operating mode sequences of DP-Net have led to an average FC of 2.55% with respect the optimal FC realized by the DDP. Considering the impossibility of employing the DDP under real-time conditions, the opportunity of DP-Net to replicate the optimal choices of a global optimizer introducing such a small FC penalty can be considered as a very interesting outcome of the research.

Open points have arisen as the application of LSTMs to the control problem of hybrid powertrains has been investigated. Consistently, the analysis and application of transformers ([198]) could be exploited as innovative AI-based approach with the capability of promisingly outperforming RNNs and LSTMs.

Chapter 6

Towards optimal real-time control of hybrid powertrains through Reinforcement Learning

The potentials of the application of a particular family of RNNs to the problem of optimal real-time control of hybrid powertrains have been discussed in Chapter 5. Indeed, some assumptions have been made to guarantee the possibility of using an LSTM-based EMS for HEVs:

1. The availability of labeled data, namely a set of pre-processed DP-based control trajectories;
2. The inclusion of temporal and/or spatial information about future driving conditions.

Unfortunately, no chance exists about the possibility of employing SL techniques for sequential control problems (such as RNNs and LSTMs) without considering a relevant a priori knowledge of the driving environment ([86]). In fact, larger input datasets for the training phase are typically associated to more accurate control sequences when the SL networks are operated in testing conditions.

Part of the present section has been extracted from:

Maino, C. *et al.* (2022). Project and Development of a Reinforcement Learning Based Control Algorithm for Hybrid Electric Vehicles. *Applied Sciences*, 12, p. 812. Doi: 10.3390/app12020812.

A typical utilization of unsupervised techniques refers to multi-level EMSs, in which the classification of driving patterns is performed through clustering algorithms in a higher level while an optimized low-level controller is used for the instantaneous management of the powertrain ([199]). In cases such as [200], a tentative of using UL has been made which complies with the possibility of generating an intelligent 3-D RBC map. To the author's knowledge, UL has still not proved the capability of being used as stand-alone algorithm in the EMS logic of (P)HEVs. On the contrary, a vast literature has been written focused on the potentials of applying RL techniques to the problem of (P)HEV real-time control ([201]). Several algorithms have been tested for different hybrid architectures, starting from well-established tabular algorithms (e.g. SARSA, Temporal Differences, Q-learning, etc.) and arriving to the latest findings in the integration of DNNs into the workflow of RL (Deep Q-network, Double Deep Q-network, Deep Deterministic Policy Gradient, etc.). Thanks to any of the above-mentioned RL approaches, the requirements of SL can be overcome thanks to an iterative interaction between an agent and an environment.

As far as Q-learning for (P)HEVs is concerned, a considerable number of articles has been presented in literature. Two of the leading implementations for real-time control of a HEV have been presented in [111] and [112], in which a transition probability matrix (TPM) has been used to provide the agent with some statistics about the driving mission to be experienced with a series and a parallel HEV, respectively. In [113], a Q-learning algorithm is applied to the real-time control problem of a range-extender HEV (REHEV) without any a priori information about the DM. Its capability of learning a control policy that reduces the cumulative FC has been demonstrated. Similar considerations about Q-learning are made in [114] about the performance of the control agent when applied to a power-split HEV architecture. In this case, a comparison between the best performances achieved at the end of the learning process and those produced by a benchmark DP is carried out. In [116], a wider set of experiments has been carried out testing a Q-learning for a parallel HEV with ICE and MG mounted on two different axles. Specifically, a change in the states and actions is performed about the number and the type of considered variables; moreover, the learning experience of the agent is discussed; finally, a sensitivity analysis over a constant-through-episodes value of the agent exploration rate. Regardless of the different applications and tests conducted over the Q-learning, any of the cited papers involves a detailed testing plan of the controller aimed at stressing its performances when applied to the optimal control of a hybrid powertrain on real-world driving environments. In

fact, a reliable utilization of ML-based techniques has to be achieved by means of detailed analyses focused on the assessment of the agent response when modifications are introduced into its most influencing parameters. Unexpected rather than counter-intuitive behaviors could be theoretically realized, and the results obtained over different driving environments could be overturned.

In this Chapter, a complete design of a Q-learning based EMS for hybrid powertrains is presented along with the main results obtained for a HEV *pc* application.

6.1 Reinforcement Learning for hybrid electric vehicles

Optimizing the energy management of a (P)HEV under real-time driving conditions is considered as one of the highest barriers to be overcome if consistent reductions in the overall emissions are targeted. In fact, the driving environment can change continuously throughout a given mission according to changes in the traffic, road conditions, weather conditions, driving behavior, etc. Moreover, FHEV and PHEV architectures are typically subject to different control policies due to the substantial differences in the powertrain. Therefore, the definition of an optimal control strategy in such a variable scenario can become a very complicated problem to deal with and several assumptions have to frequently be included in the analysis.

The general formulation of the problem accounts for the same objective functional J formulated in (27) (see Chapter 2), which is considered to be optimized under a control policy π^* and a set of boundary conditions. The general formulation of the problem can be hence defined through (28), whereas the boundary conditions can be referred to (33)-(35). Thanks to the knowledge of the entire velocity trajectory throughout a given mission (with or without the road altitude), the DDP has proved the capability to identify the optimal control policy in a completely discretized environment. Nevertheless, two main drawbacks are typically addressed by this approach when real-world conditions are considered ([174]):

- The entire trajectory of the vehicle velocity is not a priori known under real-time conditions;
- The DDP mesh discretization level could be too fine for the on-board electronics to solve practically a control problem under real-world unpredictable driving scenarios (i.e. the DDP cannot be implemented in the electronic control unit (ECU) under real-time conditions).

Evolving from DP to RL could potentially represent a solution to these two problems. Before presenting the main outcomes of the transition to RL-based approach, an insightful discussion about the possibilities of RL is presented the following sections.

6.1.1 From dynamic programming to Reinforcement Learning

Among the decision-making techniques based on ML, the family of algorithms featured by an agent interacting with an environment to reach a desired goal is typically addressed as RL ([47]). A feedback signal named “reward” is passed to

the agent at each time step of a training episode as an index of the effectiveness of the action taken. The agent-environment interaction can be formalized through a Markov decision process (MDP). A classic formulation of an MDP is completely described by a tuple <$\check{S}$, $\check{A}$, r, P, γ>, in which $\check{S}$ is the state space (i.e., set of states), $\check{A}$ is the action space (i.e., set of control actions available to the agent), r: $\check{S} \times \check{A} \rightarrow \mathbb{R}$ is the reward function, P: $\check{S} \times \check{S} \times \check{A} \rightarrow [0,1]$ is the state-action transition probability function and γ is the discount factor, a scalar value which expresses the agent's preference of gaining a future reward with respect to an immediate reward. Moreover, consistent with the experiments presented in the following sections, a further assumption of $\check{S}$ and $\check{A}$ being finite (and discrete) can be made.

The workflow of a generic RL approach can be easily explained starting from the MDP tuple. At each time step t, the agent receives a set of information about the current state of the environment, denoted as s_t. According to its control policy π, the agent chooses an action a_t. In response to the selected action a_t, the environment evolves its state from s_t into s_{t+1} based on the transition probability P. The reward feedback signal r_t is hence released. An iteration of the learning procedure is then summarized by the tuple <s_t, a_t, r_t, s_{t+1}>. Once the procedure for a time step is completed, the system moves to the following time step $t+1$ and the operations are repeated iteratively until a terminal state is reached:

$$e_T = (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \tag{69}$$

where e represents the trajectory of the agent-environment interactions within a generic episode while T represents the terminal time step. The training process can hence be satisfied by completing multiple episodes. The MDP formalism requires the states to satisfy the Markovian property: the transition from s_t to s_{t+1} depends only on s_t and a_t and not on the complete trajectory from s_0 to s_t ([47]). Therefore, the transition probability function P depends only on (s_t, a, s_{t+1}):

$$P[S_{t+1} = s_{t+1} \mid S_t = s_t, A_t = a_t] = P[S_{t+1} = s' \mid S_t = s_t, \dots, S_0 = s_0, A_t = a_t] \tag{70}$$

Stepping back to the comparison between DP and RL, both the approaches refer to iterative solutions of the Bellman optimality equation ([47]). Nevertheless, a difference between the two optimizations lies in the opportunity of assuming the knowledge of the transition probability function P. In fact, P is known in the DP algorithm and hence a direct solution of the optimal policy can be obtained under

offline conditions without interacting with the environment. On the contrary, no a priori knowledge about the environment is assumed in RL. Therefore, the state-action transition probability function is neglected from the optimization.

As a final consideration, a generic RL agent cannot always observe the entire state of the system under real-time conditions since a significant number of signals might not be available from the on-board system. Specifically for the problem of HEVs real-time control, the on-board instrumentation is typically not capable of providing an RL agent with the entire set of signals needed to produce a complete view of the environment (i.e., vehicle conditions, actual and future traffic, driver behavior, weather, etc.). In these cases, the agent-environment interactions cannot be modeled considering the formulation of (69) and a distinction between state and "observation" needs to be introduced. The state s_t that satisfies the Markovian property defined with (69) collects the entire set of signals representative of the environment, whereas the observation o_t is a subset of s_t constituted only by the set of signals which can be passed to the agent ([47]). In other words, the observation should be designed to best approximate the Markovian property about the system state. Even though theory about Partially Observable MDP has been developed in literature which accounts for the violation of the Markov property for o_t, the latter has been assumed to satisfy the Markovian property ([202]).

6.1.2 Discounted return and discount factor

For a generic RL agent, the objective functional J of DP can be stated in terms of "discounted return" G_t at each time step ([47]):

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k} \tag{71}$$

in which $\gamma \in [0, 1]$ is the discount factor. If γ=0 leads the agent to a single-step optimization, γ=1 permits the optimization to consider all the steps included in an episode. In other words, lower γ push the agent to simplify the problem by preferring the immediate reward, whereas higher γ would significantly increase the complexity of the problem faced by the agent by considering a look-ahead horizon. Evidently, the choice of the discount factor determines such an optimization horizon. A thorough discussion of the performance of an RL-based EMS for HEVs considering different discount factors is not typically made in the literature. As examples, no details about the value of the discount factor are provided to the reader

in [111] or [116], whereas γ=0.6 is selected in [113] apparently without a precise motivation. So to close this gap in the literature, the response of the RL agent to different discount factors is discussed in the following sections by means of a specific test plan.

6.1.3 Exploration vs exploitation

The RL agent possesses no a priori knowledge about the environment. Therefore, a policy should be determined that governs the actions selected by the agent. A critical challenge can be referred to the trade-off between exploration and exploitation ("exploration-exploitation dilemma") ([47]). Exploitation refers to the agent's decision of selecting the most effective action in a specific time step. On the other hand, exploration is the condition of the agent for which a random action among the set of feasible actions is chosen.

In the initial phases of the training process, the agent is typically asked to explore the environment and make experience ([47]). When the agent has acquired some knowledge about the environment, a trade-off between exploration and exploitation is imposed. In fact, the agent has to be pushed to define new ways to avoid feasible control strategies possibly conducting to local optima (i.e. sub-optimal control strategies). One common solution refers to the usage of a "ε-greedy" method ([47]). Consistent with this method, the agent mainly acts with a greed policy (i.e., exploitation is way preferred to exploration) but maintains a small probability ε to select a random action.

The choice of the exploration strategy could be crucial for the decisional process of a HEV RL-based EMS. Therefore, three different exploration laws have been developed and applied to the present research. The trends are charted in Figure 53 with respect to the training episodes. The speed of the exploration rate ε falling from a high initial value (0.8) to a very low final value (0.05) has been considered to properly label the exploration strategy. Specifically, the agent is allowed to explore the environment with a progressively decaying probability of selecting a random action in the first 5% and 75% of the training episodes in "boost" and "slow". On the contrary, a fixed exploration rate until 75% of the training process followed by an increasing percentage of exploitation is considered in the "inverse slow" strategy.

The share of exploration in a generic episode could practically lead to an unfair evaluation of the actual performance of an RL agent. This is due to the stochasticity of the exploration which causes the agent to modify the action to be applied.

Therefore, the proposed methodology for an exhaustive project of an RL agent for HEVs distinguishes between training and testing episodes. During training episodes, the agent is allowed to use the exploration strategy to collect the transition useful to gain more information about the environment. On the contrary, the exploration is not permitted during testing episodes and the agent is asked to choose greedy actions throughout the entire episode. This said, two final considerations can be made about testing episodes. First, at the end of the training process, a testing episode has to be performed to check on the real performances of the trained RL agent. Second, despite the difference in the exploration strategy, the agent could be tested with a user-defined frequency so as to track the learning progress of the agent. In the following sections, the usage of testing episodes will be presented both for in-training and out-of-training phases.

6.1.4 Reward function

The immediate effectiveness of a generic action taken by the agent can be expressed by means of the reward feedback. The identification of a proper definition for the reward function has to be made based upon the specific case study. In this research, the problem of minimizing the FC of a HEV under real-time conditions is targeted considering two boundary conditions, namely battery CS and compliance with the driving mission velocity profile. Three different reward functions have been identified. The number and typology of variables included in the reward functions have been defined considering the HEV control problem and the necessity of producing different reward orientations. The reward functions have been expressed as:

$$R = \begin{cases} R_1 = k_1 + k_2 \cdot (SoC - SoC^*)^2 + k_3 \cdot FC \\ R_2 = k_1 + k_3 \cdot FC + k_4 \cdot FC_{eq}^T \\ R_3 = k_1 + k_3 \cdot FC + k_5 \cdot FC_{eq} \end{cases} \tag{72}$$

in which k_{1-5} are numerical coefficients to be tuned, SoC^* is a reference battery SOC value, SoC is the actual battery SOC value and FC_{eq} is the equivalent FC calculated as:

$$FC_{eq} = -\frac{(SoC - SoC^*) \cdot E_{b,w}}{H_i \cdot \overline{\eta}_{ICE}} \tag{73}$$

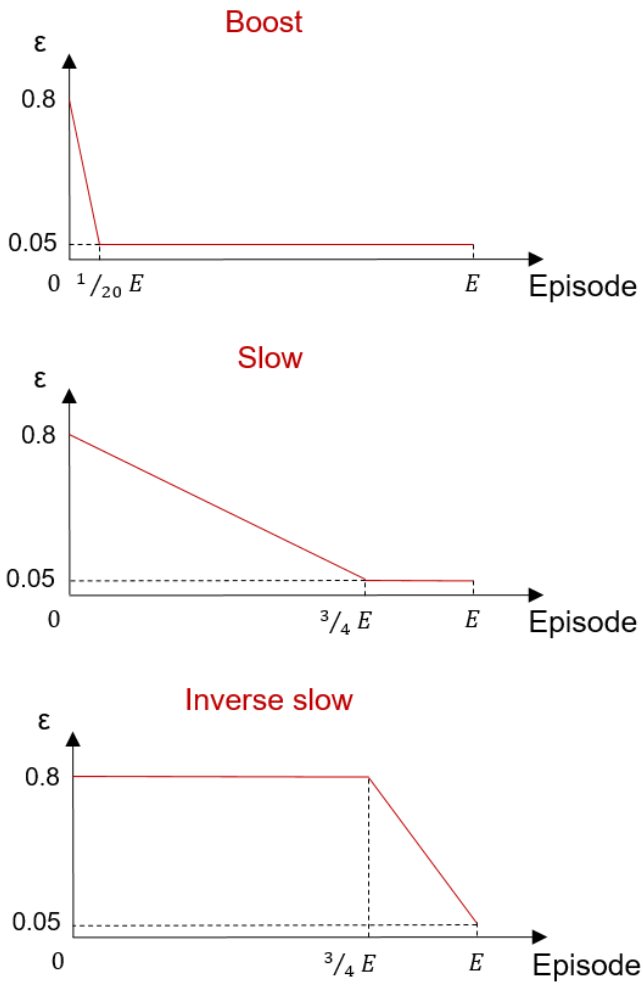

Figure 53. Exploration strategies considered for the Reinforcement Learning agent.

where $E_{b,w}$ is the battery energy content related to the admitted battery SOC window, H_i is the lower heating value of the fuel considered for the specific ICE application and $\overline{\eta}_{ICE}$ is a fixed average ICE efficiency value. Consistent with the WLTP, the equivalent FC has been calculated only in case of $SoC < SoC^*$. In (73), the difference between FC_{eq}^T and FC_{eq} is that FC_{eq}^T is calculated only at the final time step T, whereas FC_{eq} is calculated at each time step (in case of $SoC < SoC^*$).

As far as the rewards of (72) are concerned, R1 has been considered as a battery SOC-oriented reward function, whereas R2 and R3 have been progressively adapted to be more FC oriented. For this purpose, the coefficients k_{1-5} have been properly tuned for conditioning the reward orientation. For the sake of clarity, additional parameters as well as a procedure for the fine-tuning of the numerical coefficients could be considered for the optimization of the reward functions. Nevertheless, the research of the optimal performances obtained for a specific HEV application is beyond the scope of the present research. Therefore, the selection of the different reward functions presented in (72) has been considered as a reliable step for the assessment of the response generated by the RL-based EMS under very different conditions even without research of the optimal results. Considering R1, R2 and R3, a specific reward shaping analysis for HEVs is presented in the following sections.

6.1.5 The Q-learning algorithm

The core part of the RL agent is constituted by the algorithmic structure as well as by the transition function ("step function") used to stimulate the learning process. The Q-learning algorithm is one of the first and most studied RL algorithms proposed in the literature ([203]). It is based on the idea of the maximization of a q-value $q_\pi(s|a)$. The estimates of the optimal q-value (q^*) are iteratively obtained by means of the Temporal Difference (TD) principle, which provides the updates at each time step of an episode using the received reward ([47]). In a TD approach, the term $R_{t+1} + \gamma Q(s_{t+1}, s_{t+1})$ is introduced as a better approximation of the q-value for each of the state-action pairs. Such a relation is typically referred as TD-Target ([47]). The mathematical formulation for the TD-update can be written as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \qquad (74)$$

in which α is the learning rate, which quantifies the importance of the q-value update on the original value. The term that multiplies the learning rate is typically referred as TD-Error ([47]).

Based on the TD update expressed in (74), a Q-learning agent modifies the estimate of the TD-Error by including the maximum q-value which could be achieved in the next state for each action available at that stage. In other words, the q-value estimates for Q-learning involve the effect of a greedy action performed in

the next state. The mathematical formulation of the Q-learning step function becomes:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t)) \tag{75}$$

For each update of the q-value, the new estimates are stored for each state-action pair in a matrix named Q-table. In the latter, the evolution of the estimates of realizable state-action pairs can be tracked. For the sake of clarity, the real driving conditions of a HEV have been taken into account at this stage. Specifically, the entire set of actions is not always available to the agent for each time step of a driving mission due to the conditions (33)-(35). Evidently, the sub-sets of feasible and unfeasible actions are known for a generic state. Therefore, the unfeasible actions have been excluded both for the update of the q-value and for the policy π.

6.2 An Integrated Modular Software Framework for hybrid electric vehicles

Testing a RL-based EMS for a specific HEV application requires the development of a proper software platform. Therefore, the Integrated Modular Software Framework (IMSF) illustrated in Figure 54 has been realized. In the software, three main elements, namely Simulator, Environment and Agent, have been developed and integrated. In the Simulator, a HEV model has been included that is used to define the physical variation of the system when a generic action is selected by the RL agent and applied. The Environment Interface is responsible of establishing the links between the world (i.e. the vehicle) and the RL-based control logic. The latter has been realized in the Agent module. Thanks to the IMSF, an extensive and reproducible set of experiments has been performed.

In the following sections, the three core modules of the IMSF are described. For additional details, please refer to [204].

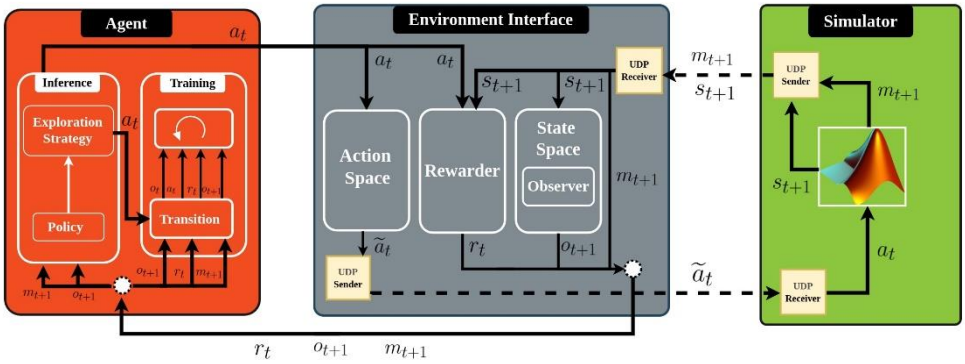

Figure 54. Integrated Modular Software Framework.

6.2.1 Simulator

In the Simulator (Figure 54, green box), the vehicle model of a pre-transmission parallel HEV *pc* has been generated. Specifically, the driveline presented in Figure 6 has been considered based on the modeling approach presented in Chapter 2. The main characteristics of both vehicle and main components are reported in Table 34.

The powertrain of a conventional pc with an 88 *kW* gasoline ICE has been integrated with a 70 *kW* MG and a 74 *kW*-6.1 *kWh* battery. The peak power of the MG has been considered for both traction and regenerative braking phases since a symmetrical operating map has been involved in the analyses. About the energy storage system, a sizing operation has been performed so that the peak power achieved by the battery could fulfill the power requirements of the MG, both during traction and regenerative braking conditions. For the sake of clarity, the HEV model has been developed using MATLAB®.

Table 34. Vehicle specs and characteristics of the main powertrain components.

Vehicle specs	
Vehicle class	Passenger car
Kerb weight	750 *kg*
Vehicle mass (including pwt components)	1200 *kg*
Transmission	6-gears
Internal Combustion Engine	
Fuel type	Gasoline
Maximum power	88 *kW* (@ 5500 *rpm*)
Maximum torque	180 *kW* (@ 1750-4000 *rpm*)
Rotational speed range	0-6250 *rpm*
Motor-generator	
Maximum power	70 *kW* (@ 6000 *rpm*)
Maximum torque	154 *kW* (@ 0-4000 *rpm*)
Rotational speed range	0-13500 *rpm*
Battery	
Peak power	74 *kW*
Energy content	6.1 *kWh*
AC/DC converter efficiency	0.95

As far as the workflow of the IMSF is concerned, the Simulator is responsible of generating the transition from a state s_t in a generic time step to the next state s_{t+1} in the following time step. The state variation can be computed based on the physical action $\widetilde{a_t}$ received by the Environment Interface (please refer to next section for the definition of physical action). During this process, the Simulator outputs also the action feasibility mask m_{t+1}, which represents the set of feasible actions at the following time step (which can be defined thanks to the calculation of s_{t+1}). Assuming a discrete action space, the feasibility mask can be represented as a Boolean vector that excludes the unfeasible actions from the feasible ones. The feasible actions have been represented by 1 whereas unfeasible actions have been represented by 0.

6.2.2 Environment Interface

As stated in the introduction of Section 6.2, the Environment Interface (Figure 54, grey box) works as bridge between the Agent and the Simulator. It comprises three sub-modules: the Action Space (AS), the State Space (SS) and the Rewarder (REW).

The AS sub-module represents the module responsible for handling and translating the logical action proposed by the Agent into a physical action for the Simulator. A physical action ($\widetilde{a_t}$) is the actual action applied to the HEV powertrain according to the control policy in a single time step. In the present research, the HEV operating mode (i.e., pure thermal, pure electric, power-split or battery charging) as well as the gear inserted in the transmission have been considered. Specifically, the *cv* reported for *pc* in Table 8 have been assumed as the set of physical actions. On the other hand, the logical action (a_t) is merely a logical representation of the physical action for the Agent. In the Environment Interface, the AS submodule is hence devoted to both map $\widetilde{a_t}$ into a_t and send the feasibility mask m_{t+1} from the Simulator to the Agent. For the sake of clarity, the AS could be capable of handling both discrete and continuous actions.

As far as the SS is concerned, the Agent does not receive information about the entire state of the system ("true state") but an observation that contains only partial information. The SS is hence responsible for the conversion of a state s_t produced by the Simulator into an observation o_{t+1} to be read by the Agent. The process that leads to the generation of the observation is divided into two main steps. An Observer is first asked to filter the state to maintain only the realistically available

information in a real-life context. Then, the Agent can receive the observation in a discrete or continuous space. If in the discrete space, the filtered state is further processed by means of a discretization; otherwise, if in continuous space, the filtered state is standardized to a [0, 1] range.

Finally, the REW is the sub-module responsible for computing the reward r_t based on a given state-action pair. The calculated reward is hence sent to the Agent as feedback for the taken action a_t.

The entire Environment Interface module has been written using the Python 3.8.5 programming language ([205]) considering the OpenAI gym standard.

6.2.3 Agent

In the Agent module (Figure 54, red module), the algorithmic structure needed to train a RL-based control logic with the capability of learning from the experience gained during the driving mission has been developed. Considering a very large number of training episodes, the optimal control policy could be achieved thanks to an accurate design of the Agent. In other words, the latter should be capable of choosing the logical action a_t at each time step for every received observation o_t based on the specific exploration–exploitation trade-off explained in Section 2.3. As for the Environment, three submodules are included in the Agent: Policy, Exploration Strategy and the Training.

The Policy sub-module is devoted to calculate the selection priority of each logical action available at a given time step. The result is hence sent to the Exploration Strategy sub-module that is asked to output the exploration rate given a specific exploration strategy. Based on the priority values as well as on the action feasibility mask m_t, the Exploration Strategy sub-module chooses an action between the feasible ones. Particularly for greedy strategies, the chosen action would result in the action with the maximum priority. Finally, the Training sub-module collects the transition (s_t, a_t, r_t, s_{t+1}) and updates the internal decision model of the agent according to the rules of Q-learning.

As for the Environment Interface module, the Agent module has been written using the Python 3.8.5 programming language ([205]) considering the OpenAI gym standard.

6.3 Results

In this Section, the results obtained through the application of a Q-learning based RL agent to the problem of the real-time energy management of a pc over real driving missions are presented.

According to the main hyper-parameters discussed in Section 6.1, the results are mainly focused on:

- Effect of the discount factor;
- Reward shaping;
- Sensitivity to the learning rate;
- Variation of the exploration strategy.

Other parameters have not been included in the test plan since an easier tuning operation could be identified, such as the maximum number of episodes in the experiment. As an example, if any limitation would have existed about computational time, a very large number of episodes could have been set so as to be confident about the possibility of the agent to converge to its best performance. The performances of the different algorithm configurations have assessed by means of the battery SOC trajectories realized during and at the end of the training and testing processes, the cumulative fuel consumption, the real fuel consumption and the shape of the discounted return.

The HEV performances have been evaluated on four different experimentally derived DMs ([206]), identified as CLUST1, CLUST2, CLUST3 and CLUST4. From Figure 55 to Figure 58, some of the main characteristics of each driving mission are compared, specifically the vehicle velocity as a function of time, the vehicle as a function of distance traveled, maximum vehicle acceleration and velocity and average vehicle acceleration and velocity. The selection of the driving missions has been based upon the necessity of testing the RL control agent for HEVs in a driving environment characterized by a miscellaneous set of driving conditions. As far as the vehicle velocity profiles are concerned, the longest driving mission among the four is CLUST1, whereas CLUST2, CLUST3 and CLUST4 are progressively shorter. Moreover, CLUST1 is featured by the highest maximum and average velocities. On the contrary, the shortest driving mission (CLUST4) involves the most demanding average acceleration during traction, as it can be seen from the very spiky velocity trajectory charted in Figure 56. The maximum

acceleration during traction is achieved on CLUST3, whereas the less demanding operations are required on CLUST2.

In order to provide a fair comparison between the performances of the different tested Q-learning configurations, benchmark results have been produced through the DDP algorithm presented in Chapter 2. Consistent with the three reward functions of Section 6.1.4, the cost function of the DDP has been formulated considering the FC minimization in case of complete CS of the battery SOC at 60%. Moreover, the state and control variables of the DDP have been selected according to the set of observations and actions considered for the Q-learning agent (i.e. battery SOC as observation, power flow and gear number as actions). As far as the state of the system is concerned, the approach presented in Chapter 3 to identify the optimal battery SOC mesh discretization has been applied to both the DDP computational grid and the Q-table ([154]). Specifically for the present case of study, 501 levels have been considered in a battery SOC window comprised between 55% and 65%. Regarding the control space, the same control decisions have been allowed to both the Q-learning and the DDP. Specifically, 42 possible actions have been permitted considering six GNs and seven PFIs.

A final consideration has to be made before stepping into the following sections. The target of the present research refers to the identification of the analyses to be performed when a robust design of a RL control algorithm for (P)HEVs is needed. Since any optimization of the Q-learning agent has been performed, the results of this activity should not be read as representative of the best performances that could be obtained by an optimized Q-learning version have been reported. Future works will relate to development and fine-tuning of optimized RL-based control agents.

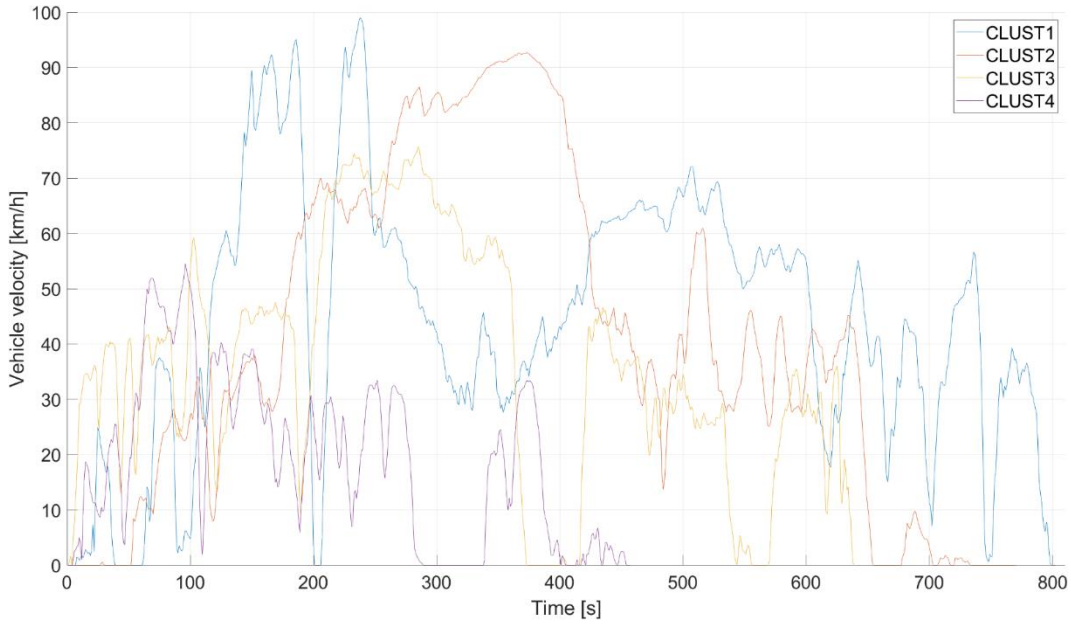

Figure 55. Vehicle velocity trajectories of the experimentally real-world driving missions with respect to the mission time.

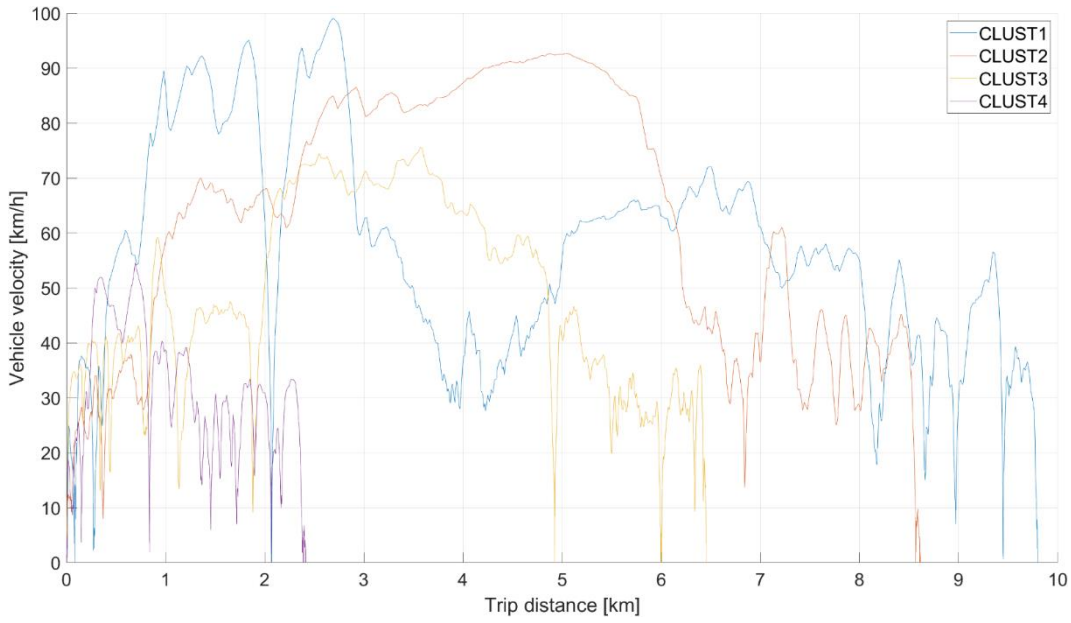

Figure 56. Vehicle velocity trajectories of the experimentally real-world driving missions with respect to the trip distance.

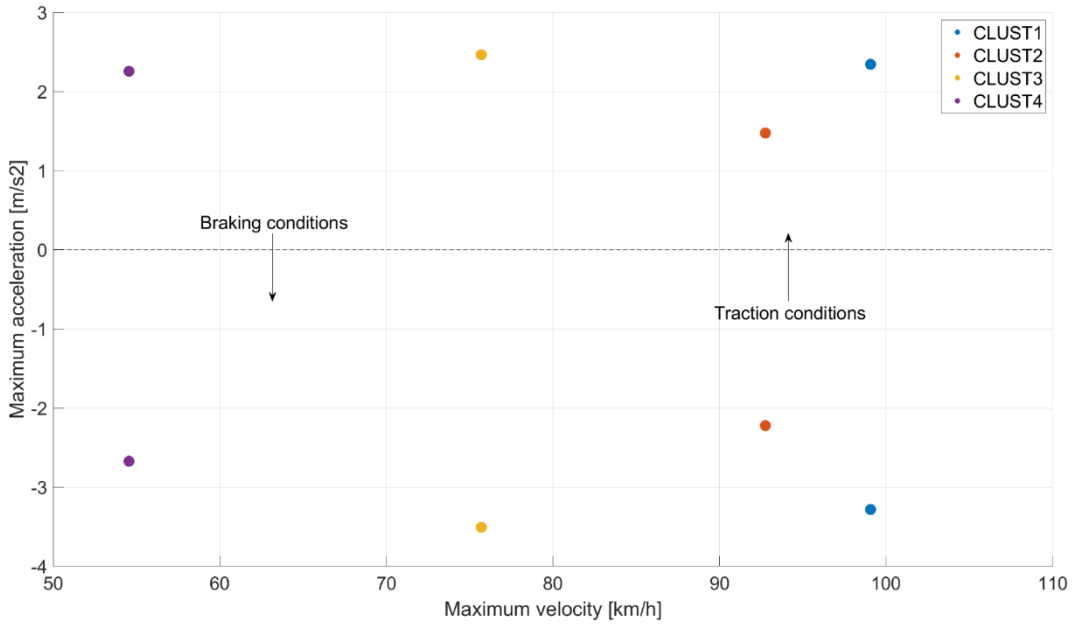

Figure 57. Comparison of the maximum acceleration obtained during traction and braking of the real-world driving missions with respect to the maximum vehicle velocity.

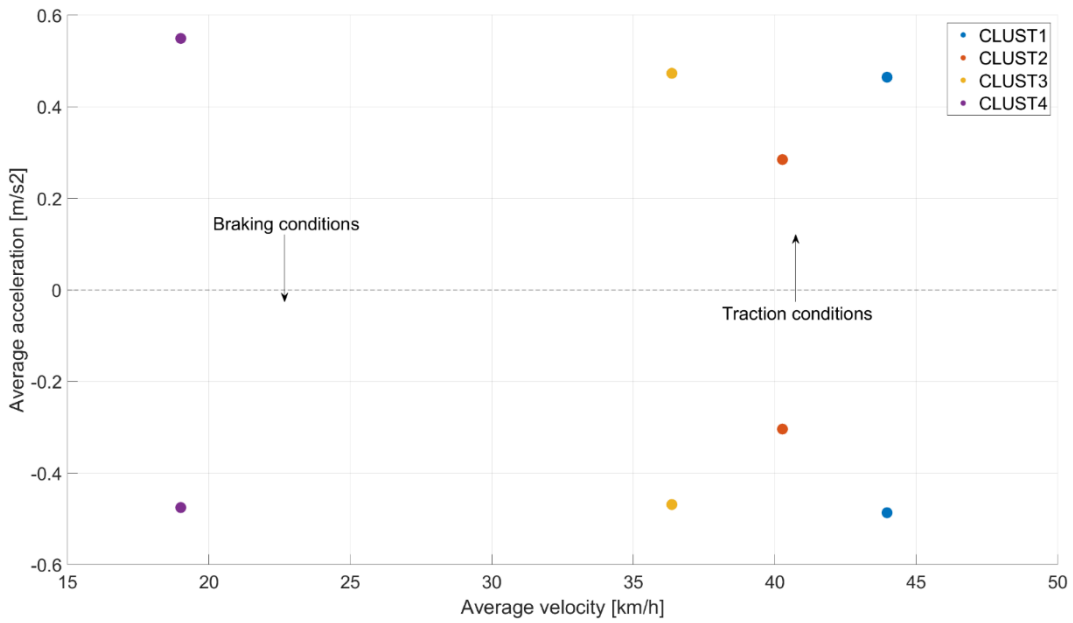

Figure 58. Comparison of the average acceleration obtained during traction and braking of the real-world driving missions with respect to the average vehicle velocity.

6.3.1 Effect of the discount factor

As first analysis, the performance of a Q-learning control agent for a parallel HEV on CLUST1 has been assessed for by studying the influence of the value considered for the discount factor (see (74) and (75)). Recalling Section 6.1.2, reducing the value of the discount factor encourages the agent to pick actions based on an immediate reward, whereas future rewards can be accounted only by increasing the value of the discount factor. Nevertheless, the real-time control of HEVs could require a trade-off between the CS of the battery SOC and the FC minimization. Therefore, an agent preferring actions based on immediate rewards might not necessarily lead to interesting control solutions. As an extreme experiment, a null value of the discount factor could demonstrate an expected inefficient setup of the agent. On the contrary, the value of the discount factor could be increased toward the unity to assess for better agent performances. In the following charts and tables, the results produced by a sensitivity of the Q-learning performances to a change in the value of the discount factor are presented.

As far as the whole agent configuration is concerned, reward function R1, α=0.9 and boost exploration strategy have been considered for this set of analyses over the effect of γ. Moreover, a plausible CS strategy has been considered only when a final battery SOC equal to 60±2.5% has been achieved at the end of the driving mission. Consistently, the final fuel consumed at the end of the driving mission has been considered only for the experiments featured by battery plausible CS trajectories. In fact, since the final fuel consumption is calculated relying on the equivalent fuel consumption calculated through (73), the accuracy of the estimation of the equivalent fuel consumption could play a significant role in the results. Therefore, a minimization of the impact of inaccurate estimations of the equivalent fuel consumption has been pursued. This phenomenon could evidently be obtained in case of a complete CS trajectory. On the contrary, if a CS trajectory is not obtained, introducing a threshold in the maximum and minimum battery SOCs could allow for excluding unrealistic calculations of the final fuel consumption.

The discounted return evaluated at the first time step of the CLUST1 is reported in Figure 59 considering a null discount factor. Neglecting the role of γ in (74) and (75) leads to the generation of a constant discounted return. The agent is pushed to learn a control policy only based on the actual reward, without accounting for the implication of the chosen action in the future. In Figure 60, the trends of the battery SOC are charted both for training (top chart) and for testing (bottom chart) episodes

between the first and the last episodes. As it can be clearly noted, the agent is not capable of sustaining the battery SOC during the training process, which turns into an unsatisfying result for the testing process. According to such a bad utilization of the electric path, the cumulative fuel consumption does not represent an interesting result and it is hence not considered. The results obtained by applying the same Q-learning configuration for CLUST2, CLUST3 and CLUST4 have shown the same behavior and, therefore, they are not reported for briefness.

Differently from the case of γ=0, increasing the value of the discount factor enhances the influence of the outcome of future time steps on the agent's decisional process (Section 6.1.2). The change in the performance of a Q-learning controller for HEVs has hence been assessed for higher discount factors. Specifically, the discount factor has been set to 0.9, 0.99 and 0.999. In Table 35, the results obtained when a variation of γ is introduced in the agent configuration are reported for each driving mission. The final battery SOC (SoC_f) always falls within the CS window except for the case of γ=0.999 over CLUST3. The latter represents a hard environment to be solved by the algorithm as it is characterized by a short but spiky velocity profile with strong accelerations and decelerations. Consistently, not the entire set of actions is feasible during several time steps of the driving mission. Such conditions can significantly reduce the number of feasible control trajectories to be explored during the training process, affecting the capability of the agent to complete a robust learning phase. For the sake of clarity, an optimization of the Q-learning configuration should have been carried out to provide the control agent with the optimal settings to properly solve the control problem over CLUST3. Nevertheless, discovering unsolved problems is a coherent result with respect to the target of the present activity.

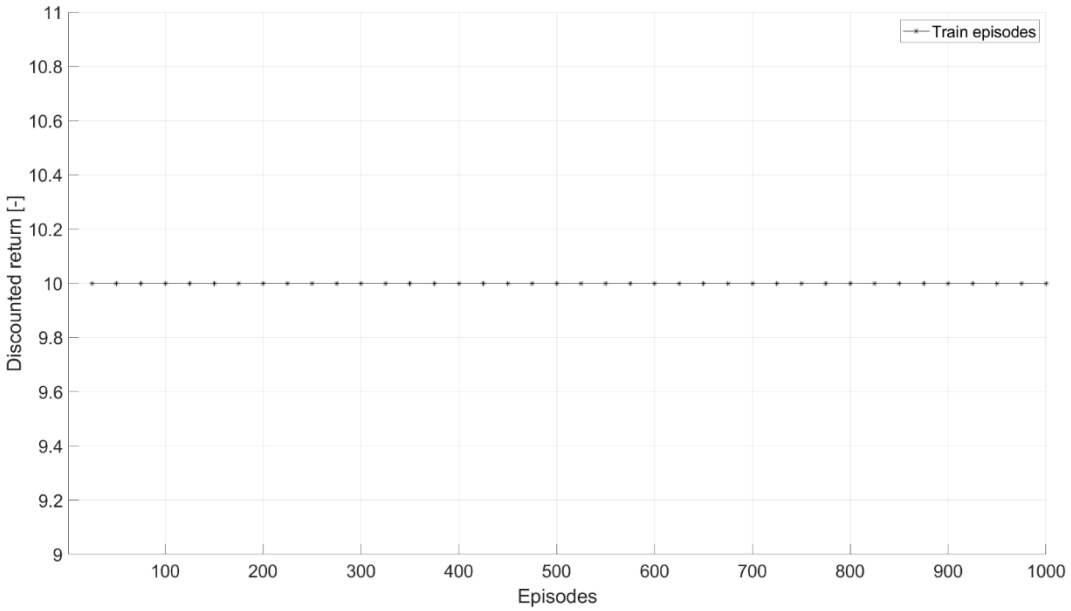

Figure 59. Discounted return for each episode evaluated at the beginning of the episode for CLUST1 in case of null discount factor.

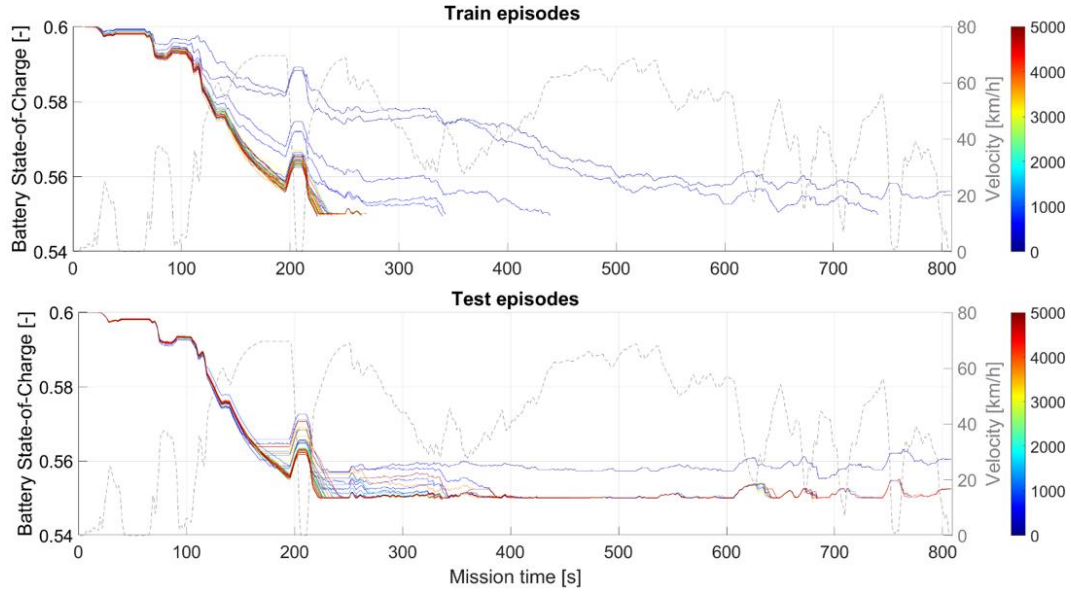

Figure 60. Trends of the battery SOC obtained during throughout the training (top chart) and testing (bottom chart) episodes.

Significant differences arise in the absolute values of both FC and real fuel consumption (rFC) achieved with the different γ values. The difference between FC and rFC consists of the additional equivalent fuel consumption (see (73)) included at the end of the experiment when the final battery SOC is lower than the reference one (0.6). Given the different durations and lengths of the four driving missions, distinct cumulative FCs could be expected at the end of the experiments. Specifically, CLUST1 and CLUST2 are longer driving missions and highlight larger fuel utilization; vice versa for CLUST3 and CLUST4.

A lower γ value (0.9) appears to outperform higher values for each driving environment in terms of rFC. On the contrary, a clear trend is not introduced by increasing γ to 0.99 or 0.999, whereas different behaviors can be obtained for the different driving missions. The intermediate value (0.99) outperforms higher values only on CLUST2 and CLUST3, whereas the highest tested value leads to a better result on CLUST1 and CLUST4. According to this non-trivial response of the Q-learning, testing the RL agent with multiple values of the discount factor appears to be a fundamental step when the identification of a specific reward function has been made.

The most interesting trends related to the best experiment over CLUST1 (γ=0.9) are reported in the next charts. The training discounted return evaluated at initial time step, 1/3 and 2/3 of the CLUST1 duration are reported in the charts from Figure 61 to Figure 63 considering the entire experiment. Differently from the result obtained with a null discount factor, different values can be noted throughout the episodes. Specifically, the discounted return appears to converge right after the first episodes (90.675) when calculated in the first time step of the driving mission, whereas a larger number of episodes is necessary for the discounted return to reach a stabilization farther in the driving mission. In fact, the discounted return of Figure 62 starts from a very low value (roughly 27), and it requires almost 500 episodes to achieve a sort of stabilization; furthermore, a negative discounted return occurs at the beginning of the experiment and almost 500 episodes are needed to converge at 2/3 of the driving mission duration (Figure 63). As an outcome of this study, the agent appears to be capable of learning the optimal decisions to be taken throughout the entire driving mission. Such a set of results based on the discounted return is fundamental if realistic information about the learning progress of the agent is to be gained. In fact, evaluating the discounted return in a single step of the driving mission does not provide the user with an exhaustive outcome about the capability of the RL agent to select the best actions during the entire mission.

Table 35. Results of the Q-learning agent over each driving mission considering different discount factors.

DM	SoC_f -				FC g			
	DDP	***Ql γ=0.9***	***Ql γ=0.99***	***Ql γ=0.999***	***DDP***	***Ql γ=0.9***	***Ql γ=0.99***	***Ql γ=0.999***
CLUST1	0.6000	0.5838	0.5939	0.5904	246.6	229.91	263.20	244.47
CLUST2	0.6000	0.5826	0.5962	0.6000	197.2	227.19	255.38	263.46
CLUST3	0.6000	0.5817	0.5754	-	152.6	173.02	169.26	-
CLUST4	0.6000	0.5819	0.6019	0.5974	53.9	52.07	106.56	93.81
	rFC g				ΔFC %			
CLUST1	246.6	253.21	272.02	258.27	-	2.68	10.31	4.73
CLUST2	197.2	252.28	260.9	263.46	-	27.93	32.30	33.60
CLUST3	152.6	199.33	204.7	-	-	30.62	34.14	-
CLUST4	53.9	78.21	106.56	97.59	-	45.10	97.70	81.06

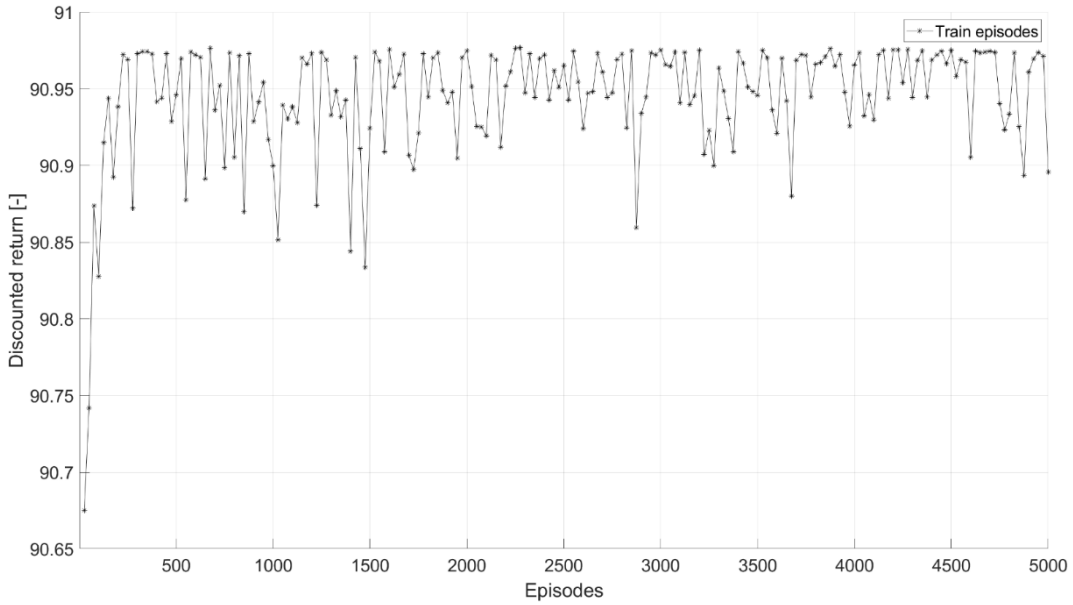

Figure 61. Discounted return for each episode evaluated at the beginning of the episode for CLUST1 in case of γ=0.9.

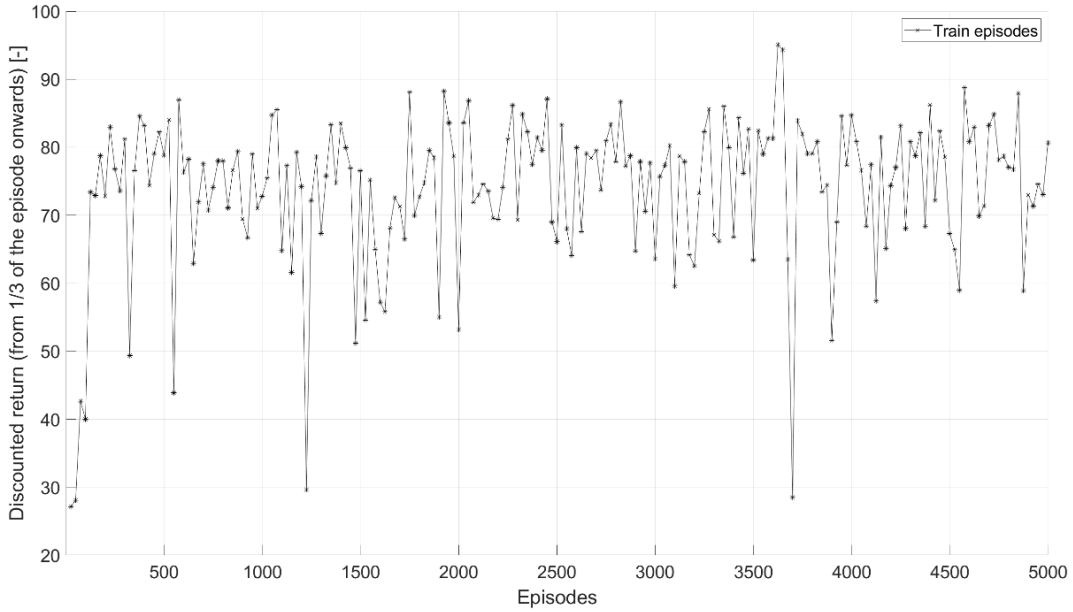

Figure 62. Discounted return for each episode evaluated from 1/3 of the episode for CLUST1 in case of γ=0.9.

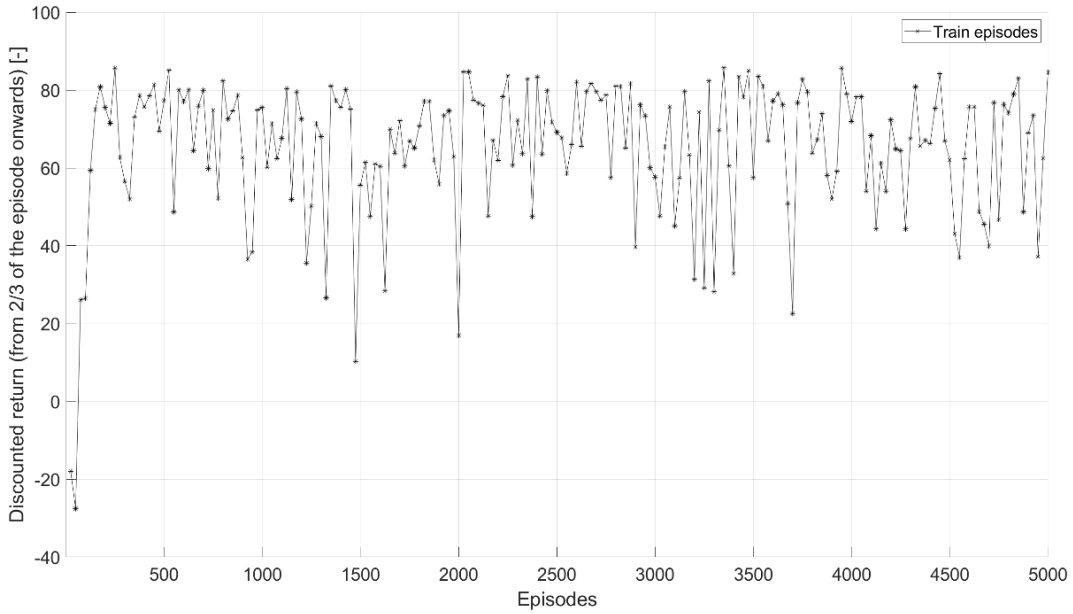

Figure 63. Discounted return for each episode evaluated from 2/3 of the episode for CLUST1 in case of γ=0.9.

Additional meaningful trajectories for the problem of HEVs real-time control are represented by the battery SOC and the cumulative FC realized with a RL-based control logic. In Figure 64 and Figure 65, the trends of the battery SOC evaluated in one of the first and the last episodes over CLUST1 are charted both for training and testing conditions, respectively. The agent proves the capability of realizing a CS battery SOC trajectory at the end of the experiment, whereas a very poor control policy is highlighted at the beginning of the experiment. An interesting outcome is also represented by the difference between training and testing episodes. In fact, the final battery SOC of the last testing episode (0.5838) is closer to the reference value (0.6) with respect to the one obtained in the last training episode (0.5760). As far as the cumulative FC is concerned, Figure 66 and Figure 67 highlight the agent capability of maintaining roughly the same FC during training and testing, respectively. This can be considered a consistent result of the training process since the achievement of improved CS trajectories is not affecting the final FC.

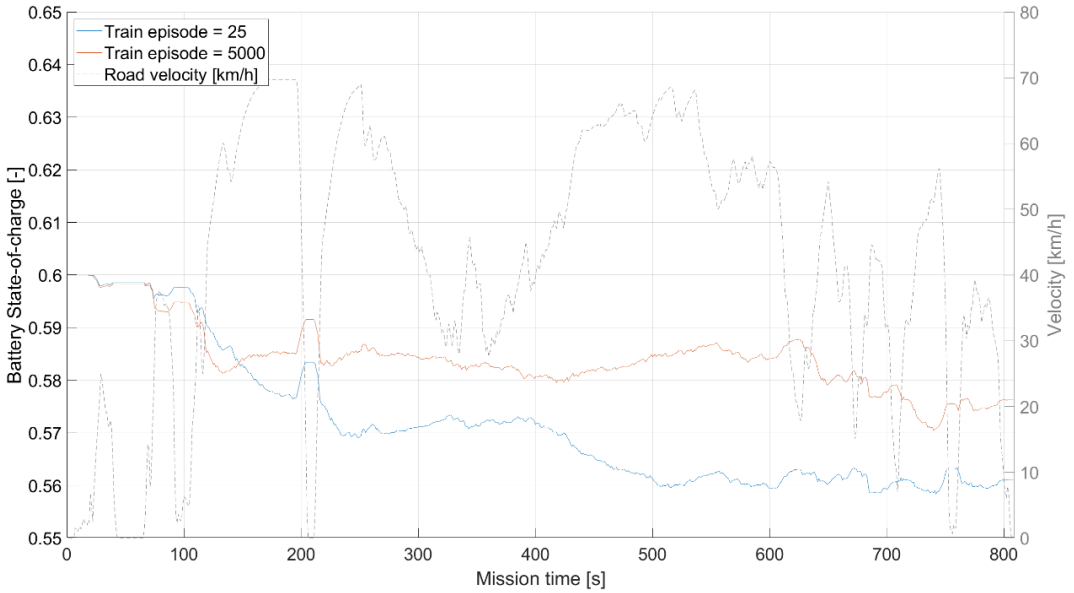

Figure 64. Battery SOC obtained at the end of the first and the last training episodes for CLUST1 in case of γ=0.9.

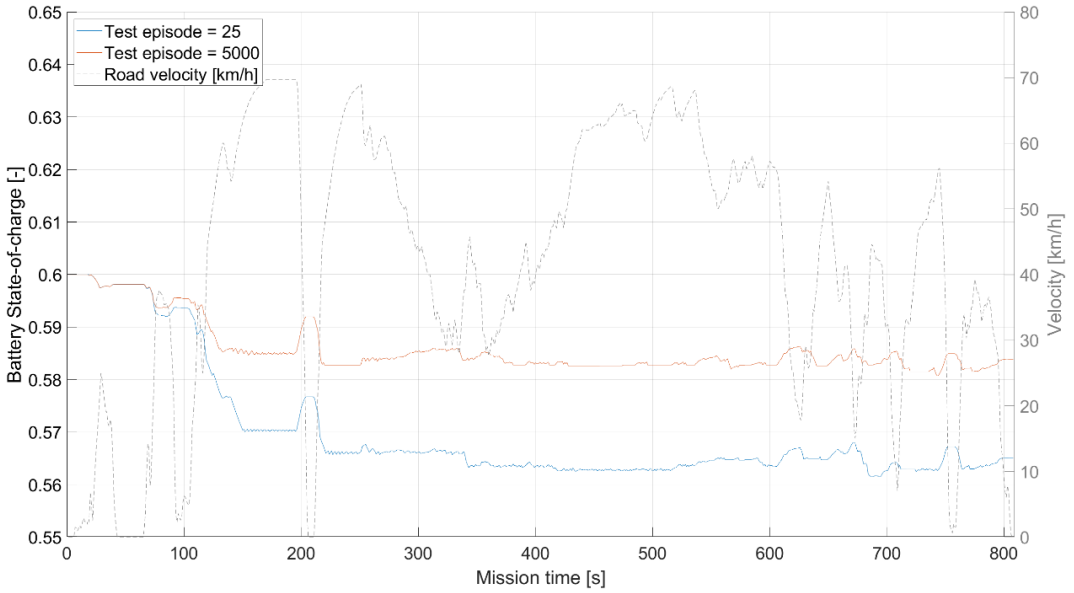

Figure 65. Battery SOC obtained at the end of the first and the last testing episodes for CLUST1 in case of γ=0.9.

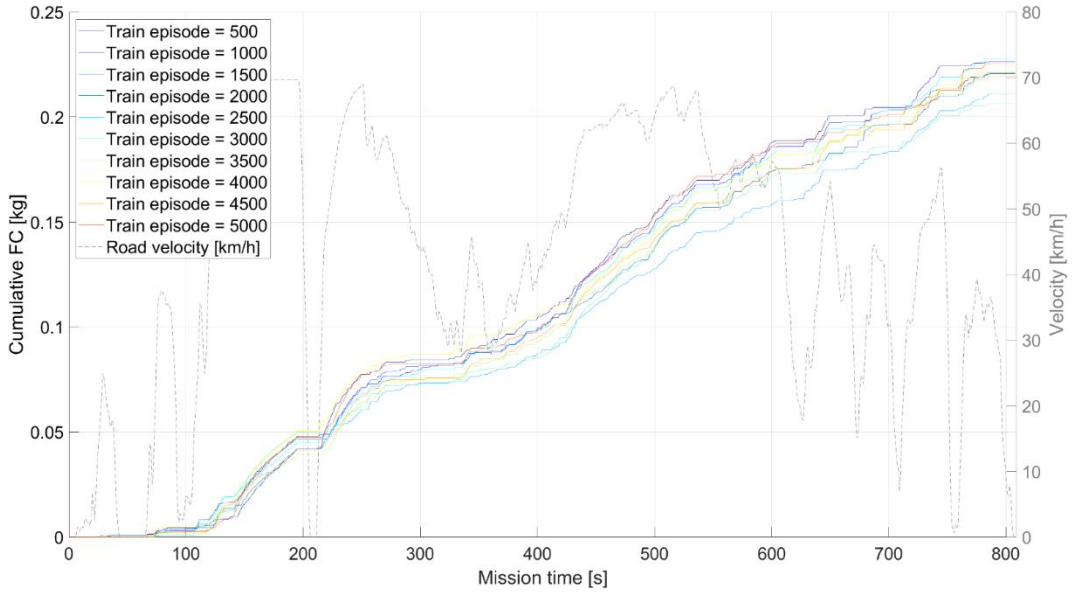

Figure 66. Cumulative FC obtained during training for CLUST1 in case of γ=0.9.

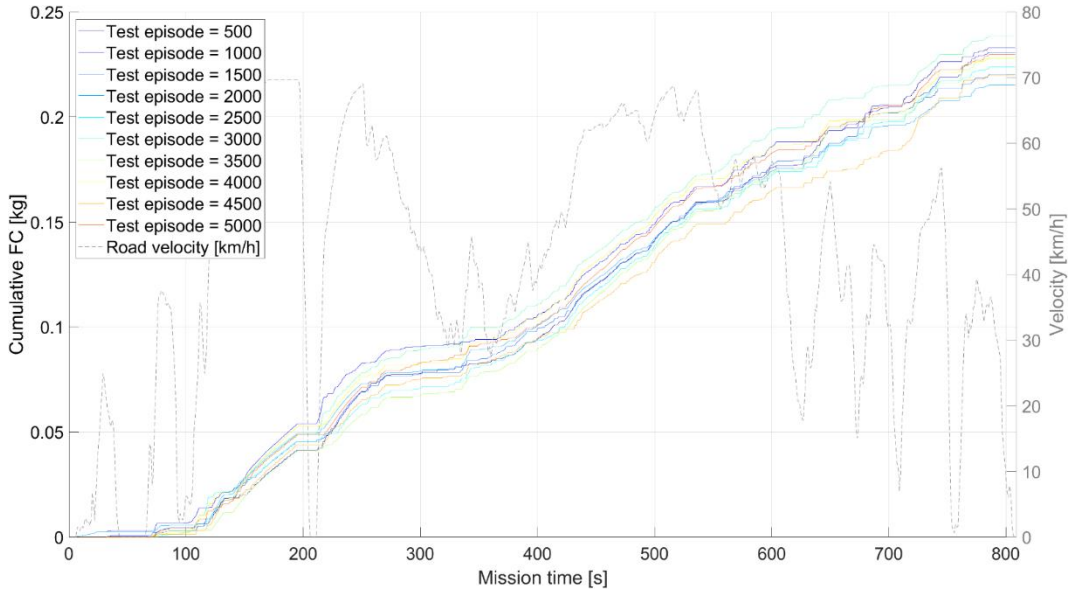

Figure 67. Cumulative FC obtained during testing for CLUST1 in case of γ=0.9.

6.3.2 Reward shaping

Within this section, the results produced by a reward shaping analysis over the Q-learning agent are reported considering the three values of the discount factor considered in Table 35. The best performing combination of reward function and discount factor has been identified (and highlighted in grey) for each driving mission.

In Table 36, the results obtained on each driving mission are reported for γ=0.9 considering SoC_f, FC, rFC and ΔFC. As for the analysis about the effect of the discount factor on the agent performances, reward function R1, α=0.9 and boost exploration strategy have been considered for the present analysis. The adoption of the lowest non-null value of γ shows very interesting results when matched with the reward function R1. The agent is capable of completing the experiments on each driving mission with final battery SOC values within the admitted CS window (SoC_f>0.58) and promising final fuel consumption with respect to DDP for CLUST1. On the contrary, matching γ=0.9 with R2 does not lead to a good response of the control agent. Regardless a change in the formulation of R2 (i.e., R2a, R2b and R2c), the agent is never capable of exploiting a complete CS, hence invalidating the entire set of experiment. Finally, the results produced by the adoption of the reward function R3 fall in between those obtained by R1 and R2. A CS trajectory is achieved for three out of four driving missions, with final battery SOC nearer to the reference value and real fuel consumptions comparable with those obtained by R1.

In Table 37 and Table 38, the same experiments have been carried out considering a discount factor equal to 0.99 and 0.999, respectively. As it can be seen, the response of the agent when considering R2 is always negative. Indeed, matching R1 and R3 with higher γ values leads to different and interesting results with respect to γ=0.9. First, the final battery SOCs are closer to the reference value for CLUST1, CLUST2 and CLUST4, which is a positive index about the agent capability of learning CS strategies. Second, no significant differences arise about the real fuel consumptions produced on CLUST1, CLUST2 and CLUST3 with respect to those obtained with γ=0.9. Such a result can be read as the demonstration that consistent Q-learning behaviors can be obtained by using different couples of discount factor and reward function. Particularly, R1 proves to be more effective on CLUST1 and CLUST3 in case of γ=0.99, whereas R3 is better for CLUST2 and CLUST4. For γ=0.999, R3 can outperform R1 on CLUST3 while producing very similar results for CLUST2 and CLUST4.

Recalling Section 6.1.4, the reward function R1 is a more SOC-oriented while R2 and R3 can be considered as more FC-oriented. Such a consideration is fundamental to justify the difficulty of the agent to solve the problem in case of R2 and R3. In fact, a more complex problem has to be faced when R2 and R3 are considered since a lower priority is given to the battery SOC trajectory. Therefore, unfeasibility conditions related to the battery SOC exceeding the limits imposed by the battery SOC window are more likely to be experienced. Therefore, higher discount factors (i.e. larger time horizons for the discounted return) could be considered as they have proved to allow for stronger agents in cases of very complex problems.

Consistent with the results of the reward shaping analysis, a strong interaction between discounted return and reward function has been demonstrated. A non-trivial agent response can be obtained when reward functions with different levels of complexity are coupled with the discount factors and tested on different driving conditions.

Table 36. Results of the reward shaping analysis considering γ=0.9.

γ	DM	SoC_f -						FC *g*					
		DDP	***R1***	***R2a***	***R2b***	***R2c***	***R3***	***DDP***	***R1***	***R2a***	***R2b***	***R2c***	***R3***
0.9	CLUST1	0.6000	0.5838	-	-	-	0.5977	246.6	229.91	-	-	-	259.89
	CLUST2	0.6000	0.5826	-	-	-	0.6000	197.2	227.19	-	-	-	282.63
	CLUST3	0.6000	0.5817	-	-	-	-	152.6	173.02	-	-	-	-
	CLUST4	0.6000	0.5819	-	-	-	0.5884	53.9	52.07	-	-	-	66.52
		rFC *g*						**ΔFC** *%*					
	CLUST1	246.6	253.21	-	-	-	263.13	-	2.68	-	-	-	6.70
	CLUST2	197.2	252.28	-	-	-	282.63	-	27.93	-	-	-	43.32
	CLUST3	152.6	199.33	-	-	-	-	-	30.62	-	-	-	-
	CLUST4	53.9	78.21	-	-	-	83.29	-	45.10	-	-	-	54.53

Table 37. Results of the reward shaping analysis considering γ=0.99.

γ	DM	SoC_f -						FC *g*					
		DDP	***R1***	***R2a***	***R2b***	***R2c***	***R3***	***DDP***	***R1***	***R2a***	***R2b***	***R2c***	***R3***
0.99	CLUST1	0.6000	0.5939	-	-	-	0.6018	246.6	263.20	-	-	-	280.80
	CLUST2	0.6000	0.5962	-	-	-	0.6031	197.2	255.38	-	-	-	254.28
	CLUST3	0.6000	0.5754	-	-	-	-	152.6	169.26	-	-	-	-
	CLUST4	0.6000	0.6019	-	-	-	0.5984	53.9	106.56	-	-	-	88.48
		rFC *g*						**ΔFC** *%*					
	CLUST1	246.6	272.02	-	-	-	280.83	-	10.31	-	-	-	13.88
	CLUST2	197.2	260.9	-	-	-	254.28	-	32.30	-	-	-	28.95
	CLUST3	152.6	204.7	-	-	-	-	-	34.14	-	-	-	-
	CLUST4	53.9	106.56	-	-	-	90.86	-	97.70	-	-	-	68.57

Table 38. Results of the reward shaping analysis considering γ=0.999.

γ	DM	SoC_f -						FC *g*					
		DDP	***R1***	***R2a***	***R2b***	***R2c***	***R3***	***DDP***	***R1***	***R2a***	***R2b***	***R2c***	***R3***
0.999	CLUST1	0.6000	0.5904	-	-	-	0.6011	246.6	244.47	-	-	-	293.15
	CLUST2	0.6000	0.6000	-	-	-	0.6049	197.2	263.46	-	-	-	266.82
	CLUST3	0.6000	-	-	-	-	0.5795	152.6	-	-	-	-	185.51
	CLUST4	0.6000	0.5974	-	-	-	0.5982	53.9	93.81	-	-	-	95.23
		rFC *g*						**ΔFC** *%*					
	CLUST1	246.6	258.27	-	-	-	293.15	-	4.73	-	-	-	18.88
	CLUST2	197.2	263.46	-	-	-	266.82	-	33.60	-	-	-	35.30
	CLUST3	152.6	-	-	-	-	214.99	-	-	-	-	-	40.88
	CLUST4	53.9	97.59	-	-	-	97.86	-	81.06	-	-	-	81.56

6.3.3 Sensitivity to the learning rate

Another important factor for the assessment of the performance of a Q-learning algorithm is represented by the learning rate (α). As for the discount factor, a sweep over three different values has been performed, specifically 0.1, 0.5 and 0.9. For the present analysis, the best combination between reward function and discount factor achieved in Section 6.3.2 has been selected: R1 with γ=0.9. The boost exploration strategy has also been maintained.

In Table 39, the results obtained by the Q-learning control agent are reported for each driving mission. As for the experiments shown in the previous sections, CLUST3 appears to be the hardest driving environment to be solved when moving from the original combination of R1, γ=0.9 and α=0.9. Moreover, unexpected results can be obtained on different driving missions in case of a drastic change to the agent configuration. As an example, a very low learning factor can be capable of outperforming even the most common values (e.g. 0.9) on specific driving missions. Particularly, the lowest rFC is exploited with α=0.1 on CLUST2 while a comparable rFC with respect to α=0.9 is obtained on CLUST1. As far as α=0.5 is concerned, such a configuration appears not to be optimal in minimizing the FC whereas it is capable of achieving the best CS results for CLUST1, CLUST2 and CLUST4.

Since the learning rate generally determines the speed of the Q-table update, a non-trivial response of the agent has been proved for throughout the present test cases. In fact, substantial variation can be obtained in the behavior of the agent when a reduction in the α value is introduced in the Q-learning setup. Nevertheless, as a general outcome of several research works in the literature, a higher value leads to more robust RL controllers for HEVs.

Table 39. Results of the sensitivity analysis over the learning rate.

DM	SoC_f -				FC g			
	DDP	***Ql α=0.1***	***Ql α=0.5***	***Ql α=0.9***	***DDP***	***Ql α=0.1***	***Ql α=0.5***	***Ql α=0.9***
CLUST1	0.6000	0.5778	0.6014	0.5838	246.6	226.51	298.62	229.91
CLUST2	0.6000	0.5832	0.6021	0.5826	197.2	207.42	280.22	227.19
CLUST3	0.6000	-	-	0.5817	152.6	-	-	173.02
CLUST4	0.6000	-	0.6003	0.5819	53.9	-	101.89	52.07
	rFC g				ΔFC %			
CLUST1	246.6	258.65	298.62	253.21	-	4.89	21.09	2.68
CLUST2	197.2	231.65	280.22	252.28	-	17.47	42.10	27.93
CLUST3	152.6	-	-	199.33	-	-	-	30.62
CLUST4	53.9	-	101.89	78.21	-	-	89.04	45.10

6.3.4 Variation of the exploration strategy

As a final step, the exploration strategy of the agent has been varied according to the strategies shown in Section 6.1.3. The considered exploration strategies are featured by a linear decay of the exploration rate followed or preceded by a constant trace. Specifically, "boost" and "slow" are characterized by two decays with different slopes followed by a horizontal phase with ε=0.05. On the contrary, a horizontal phase with ε=0.8 is maintained for the majority of the experiment and then followed by the linear decrease in "inverse slow". The three exploration strategies will be hereafter referred to as L1, L2 and L3, respectively. Once again, the best combination of reward function, discount factor and learning rate identified in Section 6.3.3. has been considered for the present analysis (i.e. R1, γ=0.9 and α=0.9).

Different from the results shown in the previous sections, a clear trend is highlighted by the results reported in Table 40 considering the entire set of driving missions. In fact, L1 appears to be the best performing exploration strategy as it allows the agent to solve each driving environment (i.e. complete the missions)

while minimizing the FC. On the other hand, L2 and L3 are capable of producing more accurate CS trajectories as the final battery SOCs are comparable with those obtained with DDP. Nevertheless, the rFC values are worsened with respect to those achieved with L1. Moreover, the velocity trajectory of CLUST3 cannot be fulfilled as both L2 and L3 are not capable of completing the driving missions without falling outside of the battery SOC window (0.55-0.65).

Table 40. Effects of the exploration strategy on the performances of the Q-Learning agent.

DM	SoC_f -				FC *g*			
	DP	***L1***	***L2***	***L3***	***DP***	***L1***	***L2***	***L3***
CLUST1	0.6000	0.5838	0.6020	0.6017	246.6	229.91	309.01	269.35
CLUST2	0.6000	0.5826	0.6020	0.6022	197.2	227.19	299.63	255.55
CLUST3	0.6000	0.5817	-	-	152.6	173.02	-	-
CLUST4	0.6000	0.5819	0.6010	0.6010	53.9	52.07	88.09	90.33
	rFC *g*				ΔFC *%*			
CLUST1	246.6	253.21	309.01	269.35	-	2.68	25.31	9.23
CLUST2	197.2	252.28	299.63	255.55	-	27.93	51.94	29.59
CLUST3	152.6	199.33	-	-	-	30.62	-	-
CLUST4	53.9	78.21	88.09	90.33	-	45.10	63.43	67.59

Since L1 is characterized by a very short exploration boost with respect to L2 (and even more with respect to L3), an additional analysis has been performed by assessing for the learning process of the agent during the very first episodes of the experiment. For this purpose, L1 and L3 have been applied, considering very short experiments with "only" 250 episodes on CLUST1, namely L1s and L3s.

Comparisons of the discounted returns evaluated at the first time step, 33% and 66% of the CLUST1 duration are reported in the charts from Figure 68 to Figure

70. Looking at trends of the discounted return at the beginning of the driving mission, no real difference arises when the exploration strategy is changed since an almost constant value is achieved throughout the entire episode. On the contrary, a significant difference occurs both at 33% and at 66% of the driving mission duration when L3s replaces L1s. The enlargement of the share of episodes with ε>0.05 creates a lag in the number of episodes needed by the discounted return to reach the convergence. Indeed, the stabilization is achieved faster when a shorter exploration boost is defined and a longer exploitation period is allowed to the agent.

As a confirmation of the trends highlighted by the discounted returns along the driving mission, the effectiveness of a specific RL agent for HEVs has to be confirmed through the results of SoC_f and rFC. The latter are reported in Table 41for both L1s and L3s. The results highlight that the agent is capable of reducing the rFC under L1s (with respect to L3s) even when a much shorter exploration boost is considered. The speed of convergence associated to the discounted return allows the agent for more robust performances. Nevertheless, considering the results of boost with larger experiments (Table 40), enlarging the constant ε region leads the agent toward a progressive adaptation of the final battery SOC in favor of a better rFC.

Table 41. Results of L1s and L3s on CLUST1.

DM	SoC_f -		**FC** g		**rFC** g	
	L1s	***L3s***	***L1s***	***L3s***	***L1s***	***L3s***
CLUST1	0.6014	0.6010	267.74	286.08	267.74	286.08

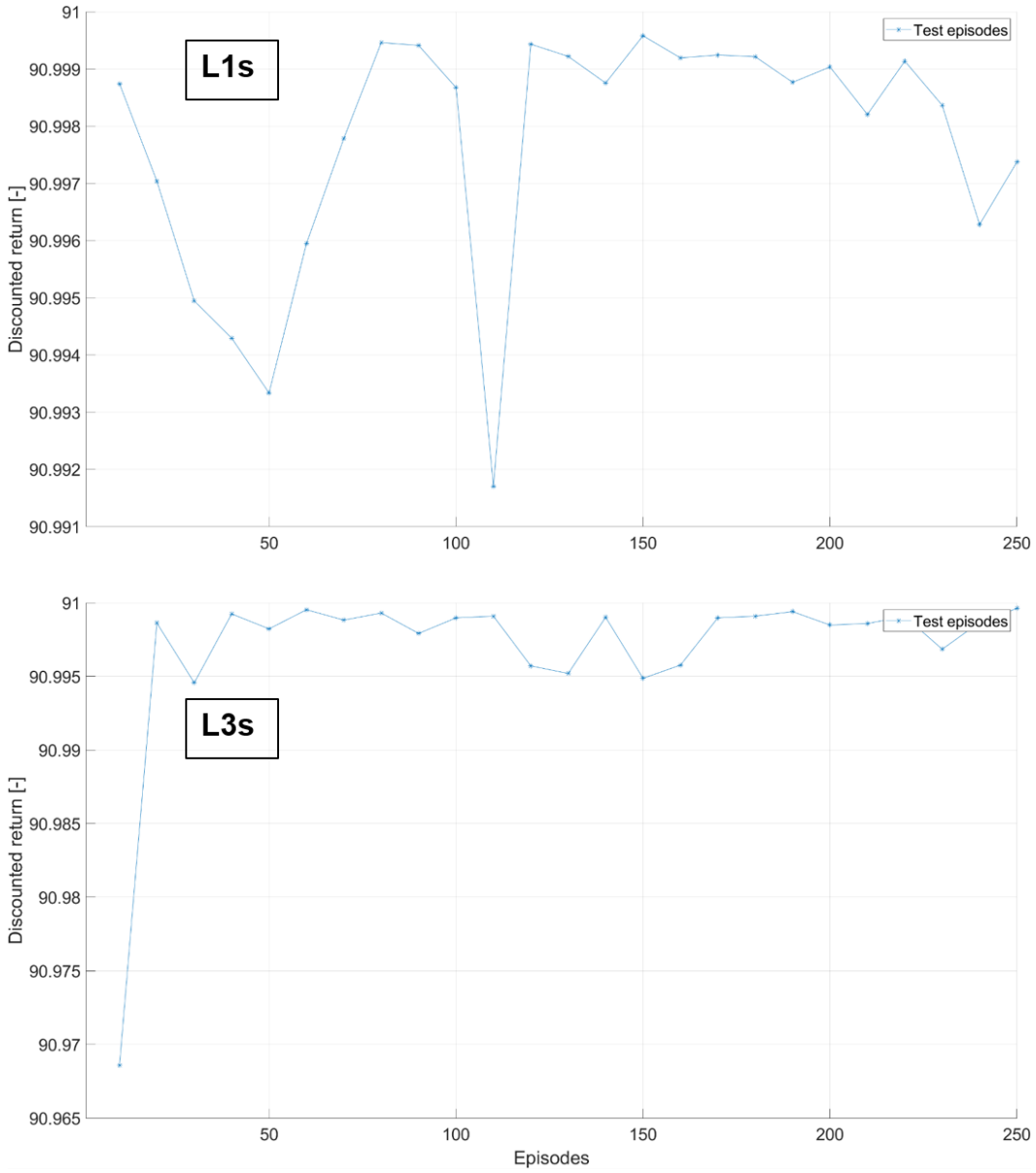

Figure 68. Comparison of the discounted return evaluated at the initial time step of the testing episodes for L1s and L3s.

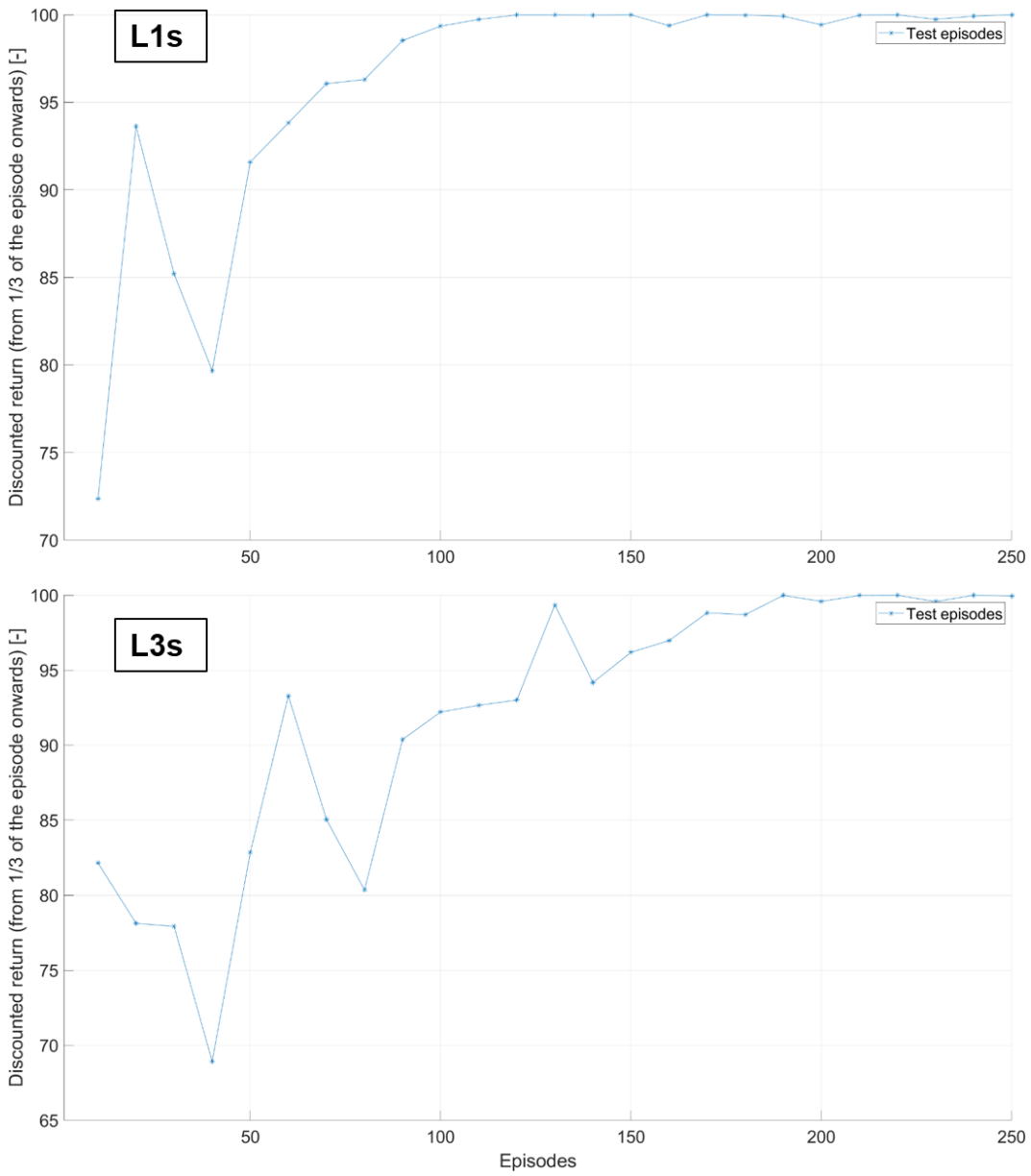

Figure 69. Comparison of the discounted return evaluated at 1/3 of the duration of the testing episodes for L1s and L3s.

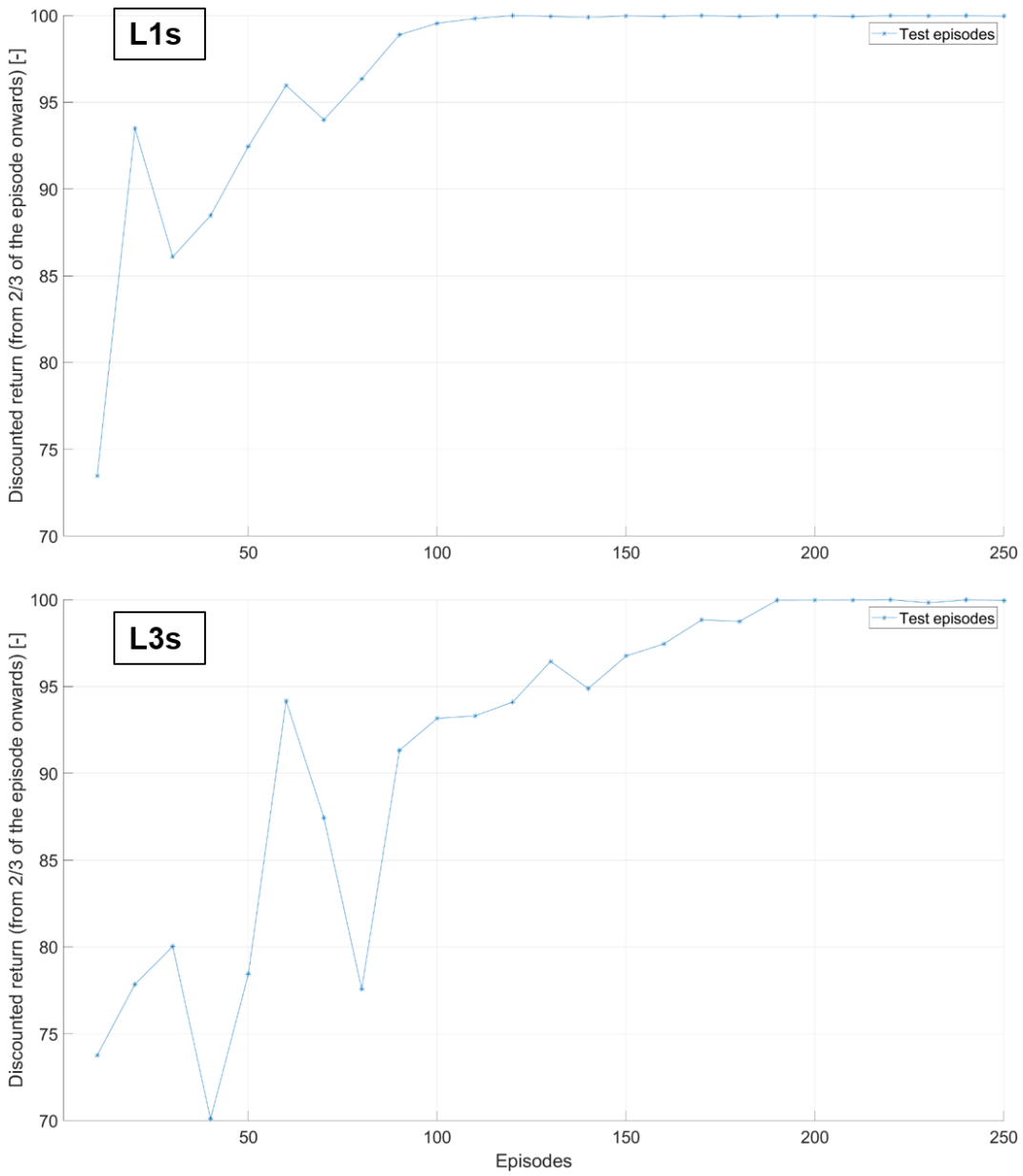

Figure 70. Comparison of the discounted return evaluated at 2/3 of the duration of the testing episodes for L1s and L3s.

6.4 Conclusions

In this Chapter, a comprehensive view of a thorough experimental test for a Q-learning algorithm to be implemented as HEV real-time capable EMS has been presented. The experiments conducted in the research activity are hence aimed at providing the reader with a complete procedure for a reliable assessment of the performance obtained by a RL algorithm for HEVs.

Two elements have been introduced to evaluate the agent performances. First, the distinction between training and testing episodes. Training episodes are featured by law about the agent exploration rate. On the contrary, the agent exploration strategy is turned off and the greedy action is always selected in testing episodes. Second, the identification of the discounted return during training and testing episodes as the most meaningful signal about the learning process of the RL agent. Contrarily, the sum of the rewards obtained by the agent in the episodes is typically used in the literature. Thanks to the present research, the trend of the discounted return computed in different time steps of training or testing episodes has proved to be a consistent index for the assessment of the RL agent performances.

In addition, the evaluation of the response of a Q-learning agent has been discussed considering a massive testing of the algorithm on different real-world driving conditions. Specifically, four different analyses have been conducted and thoroughly presented which are commonly neglected in the literature. First, the effect of the discount factor on the decisional process of the agent has been analyzed considering very different discount factors. Starting from the case of a null discount factor, an improved performance of the Q-learning has been appreciated in most of the cases when the discount factor has been set to values toward the unity. Such an outcome has been proved by means of the discounted return trajectory evaluated at three different stages of the episode. Second, a reward shaping analysis has been carried out considering an optimization of the cumulative fuel consumed at the end of the driving mission in presence of a boundary condition about the battery SOC charge sustaining. Such an analysis has been conducted considering different discount factors so as to highlight the non-trivial response of the agent when the training parameters are mixed and tested over different environments. As an outcome, the FC-oriented reward function has proved to be more efficient when higher discount factors are considered with respect to a reward focusing more on the battery SOC CS. This can be addressed to the nature of the HEV control problem. A FC-oriented control strategy can, in fact, fall into unfeasible policies with a high frequency due to the difficulty of maintaining the battery SOC under

the admitted SOC window when any information (or information with small relevance) is provided to the agent about the influence of the battery SOC variations on the system. Third, the influence of the learning rate on the performance of the Q-learning agent. Recalling the learning rate to be responsible of the Q-table rate of update, the algorithm has proved to be very sensitive to the variation of its internal learning parameter. In fact, a worsened response of the agent can be obtained when a reduction in the learning rate value is applied. On the contrary, higher values can push the Q-learning algorithm to better performances as already proved in the literature about RL for HEVs. Finally, the role of the exploration strategy has been discussed as the last analysis. Three different exploration strategies have been considered, featured by three different exploration rate decays. The fastest decay followed by a large number of episodes with a very high exploitation rate have proved to outperform other strategies featured by longer exploration rate decays. Such an outcome has been discussed both for extended experiments (with thousands of episodes) and short experiments (with few hundreds of episodes). Despite the change in the experiment duration, the Q-learning agent has achieved a faster convergence of the discounted return evaluated at different stages of the driving mission when a small exploration boost has been considered.

The results presented in the Chapter can hence be considered as a demonstration of the broadness and the depth of the tests needed to achieve a real assessment of the performances of a Q-learning agent for the energy management of hybrid powertrains. Next steps will surely involve modifications into the ISMF configuration. Specifically, an analysis of advanced RL agents, such as DQN and DDQN, will be carried out introducing a change in the Agent module. Likewise, different vehicle models will be considered in the Simulator module that will allow the assessment of the RL agents on different electrified powertrains, such as PHEVs or BEVs.

Appendix A

The power equations used to model a post-transmission and a through-the-road parallel hybrid powertrains are hereafter reported.

Post-transmission parallel hybrid electric powertrain:

$$P_{MG}(t) = \alpha_{om} \cdot P_{fd}(t) \quad \text{(A1)}$$

$$P_{MG,el}(t) = P_{MG,mech}(t) \cdot \eta_{MG}^{k}(T_{MG,mech}, \omega_{MG}) \quad \text{(A2)}$$

$$P_{REESS,el}(t) = P_{MG,el}(t) \cdot \eta_{AC/DC}^{k} + P_{aux,REESS} \quad \text{(A3)}$$

$$P_{ICE}(t) = (1 - \alpha_{om}) \cdot (P_{req}(t) + P_{aux,ICE}) \quad \text{(A4)}$$

Through-the-road parallel hybrid electric powertrain:

$$P_{MG}(t) = P_{MG,mech}(t) = \phi_{om} \cdot P_{fds}(t) \quad \text{(A5)}$$

$$P_{MG,el}(t) = P_{MG,mech}(t) \cdot \eta_{MG}^{k}(T_{MG,mech}, \omega_{MG}) \quad \text{(A6)}$$

$$P_{REESS,el}(t) = P_{MG,el}(t) \cdot \eta_{AC/DC}^{k} + P_{aux,REESS} \quad \text{(A7)}$$

$$P_{ICE}(t) = (1 - \phi_{om}) \cdot (P_{req}(t) + P_{aux,ICE}) \quad \text{(A8)}$$

For through-the-road powertrains, ϕ_{om} represents a power-split ratio between the primary (ICE) and the secondary (MG) axles.

Appendix B

The design variables considered for a post-transmission hybrid powertrain are the same considered for a pre-transmission hybrid powertrain. The design variables considered for a through-the-road parallel hybrid are hereafter reported.

Powertrain component	Design Variable
Internal Combustion Engine	Displacement
Motor-Generator	Peak power
Battery	Power-to-energy ratio
Battery	Maximum C-rate in discharge
Battery	Maximum C-rate in charge
Final drive of the primary axle	Speed ratio
Final drive of the secondary axle	Speed ratio

References

[1] Arrhenius, S. (1896). On the Influence of Carbonic Acid in the Air upon the Temperature of the Ground. *Philosophical Magazine and Journal of Science*, 41(5), pp. 237–276.

[2] Arrhenius, S. (1908). *Worlds in the Making*. New York: Harper & Brothers.

[3] Callendar, G. S. (1938). The artificial production of carbon dioxide and its influence on temperature. *Quarterly Journal of the Royal Meteorological Society*.

[4] Keeling, C. D. *et al.* (1976). Atmospheric carbon dioxide variations at Mauna Loa Observatory, Hawaii. *Tellus*, 28(6), pp. 538–551. Doi: 10.1111/j.2153-3490.1976.tb00701.x.

[5] Hansen, J. *et al.* (1988). Global climate changes as forecast by Goddard Institute for Space Studies three-dimensional model. *Journal of Geophysical Research*, 93(D8), pp. 9341–9364. Doi: 10.1029/JD093iD08p09341.

[6] Intergovernmental Panel on Climate Change [online: https://www.ipcc.ch/] (accessed 14th October 2021).

[7] United Nations for Climate Change. Kyoto Protocol, [online: https://unfccc.int/process-and-meetings/conferences/past-conferences/kyoto-climate-change-conference-december-1997/decisions-kyoto-climate-change-conference-december-1997], (accessed 14th October 2021).

[8] United Nations for Climate Change. Paris Agreement, [online: https://unfccc.int/process/conferences/pastconferences/paris-climate-change-conference-november-2015/paris-agreement] (accessed 14th October 2021).

[9] NASA. Global Climate Change. Global Temperature, [online: https://climate.nasa.gov/vital-signs/global-temperature/] (accessed 12th October 2021).

[10] Lüthi, D. *et al.* (2008). CO2 record from the EPICA Dome C 1999 (EDC99) ice core (Antarctica) covering 650 to 800 kyr BP measured at the University of Bern, Switzerland. *PANGAEA*, 453, pp. 379–383. Doi: 10.1594/PANGAEA.710901.

[11] Petit, J. *et al.* (1999). Climate and atmospheric history of the past 420,000 years from the Vostok ice core, Antarctica. *Nature*, 399, pp. 429–436. Doi: 10.1038/20859.

[12] Global Monitoring Laboratory. Trends in Atmospheric Carbon Dioxide, [online: https://gml.noaa.gov/ccgg/trends/] (accessed 15th October 2021).

[13] Blunden, J. *et al.* (2020). State of the climate in 2019. *Bulletin of the American Meteorological Society*, 101(8), pp. S1–S8. Doi: 10.1175/2020BAMSStateoftheClimate.1.

[14] Parmesan, C. and Yohe, G. (2003). A globally coherent fingerprint of climate change impacts across natural systems. *Nature*, 421(6918), pp. 37–42. Doi: 10.1038/nature01286.

[15] IPCC (2018). IPCC report Global warming of 1.5°C. *Ipcc - Sr15*, 2(October), pp. 17–20.

[16] IEA. Transport, Improving the sustainability of passenger and freight transport, [online: https://www.iea.org/topics/transport] (accessed 15th October 2021).

[17] EPA. Sources of Greenhouse Gas Emissions, [online: https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions] (accessed 15th October 2021).

[18] ICCT. Vision 2050: A strategy to decarbonize the global transport sector by mid-century, [online: https://theicct.org/publications/vision2050] (accessed 15th October 2021).

[19] EPA. Fast Facts on Transportation Greenhouse Gas Emissions, [online: https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions] (accessed 15th October 2021).

[20] ACEA. Fuel types of new cars: electric 10.5%, hybrid 11.9%, petrol 47.5% market share full-year 2020, [online: https://www.acea.auto/fuel-pc/fuel-types-of-new-cars-electric-10-5-hybrid-11-9-petrol-47-5-market-share-full-year-2020/] (accessed 15th October 2021).

[21] ACEA. New vans in the EU by fuel type, [online https://www.acea.auto/figure/vans-eu-fuel-type/] (accessed 15th October 2021).

[22] ACEA. Fuel types of new trucks: diesel 96.5%, electric 0.4%, alternative fuels 2.9% market share in 2020, [online: https://www.acea.auto/fuel-cv/fuel-types-of-new-trucks-diesel-96-5-electric-0-4-alternative-fuels-2-9-market-share-in-2020/] (accessed 15th October 2021).

[23] McKinsey & Company (2019). Reboost: A comprehensive view on the changing powertrain component market and how suppliers can succeed.

[24] UNECE. UN Regulation No. 154 - Worldwide harmonized Light vehicles Test Procedure (WLTP), [online: https://unece.org/transport/documents/2021/02/standards/un-regulation-no-154-worldwide-harmonized-light-vehicles-test] (accessed 15th October 2021).

[25] EPA. Emission Standards Reference Guide, EPA Federal Test Procedure (FTP), [online: https://www.epa.gov/emission-standards-reference-guide/epa-federal-test-procedure-ftp] (accessed 19th July 2022).

[26] BusinessEurope. The “Fit for 55 package” – a BusinessEurope position paper [online: https://www.businesseurope.eu/publications/fit-55-package-businesseurope-position-paper] (accessed 19th July 2022).

[27] Wappelhorst, S. *et al.* (2021). Europe’s CO_2 emission performance standards for new passenger cars: Lessons from 2020 and future prospects. Available at: https://theicct.org/wp-content/uploads/2021/12/eu-ev-pv-co2-emission-performance-sept21.pdf.

[28] ACEA. ACEA proposals for Euro 7 and Euro VII emission standards, [online: https://www.acea.auto/publication/acea-proposals-for-euro-7-and-euro-vii-emission-standards/] (accessed 19th July 2022).

[29] EPA. Regulations for Emissions from Vehicles and Engines, Proposed Rule to Revise Existing National GHG Emissions Standards for Passenger Cars and Light Trucks Through Model Year 2026, Revised 2023 and Later Model Year Light Duty Vehicle Greenhouse Gas Emissions Standards [online: https://www.epa.gov/regulations-emissions-vehicles-and-engines/proposed-rule-revise-existing-national-ghg-emissions] (accessed 15th October 2021).

[30] Cardoso, D. S., Fael, P. O. and Espírito-Santo, A. (2020). A review of micro and mild hybrid systems. *Energy Reports*. Elsevier Ltd, 6, pp. 385–390. Doi: 10.1016/j.egyr.2019.08.077.

[31] Enang, W. and Bannister, C. (2017). Modelling and control of hybrid electric vehicles (A comprehensive review). *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, 74(August 2016), pp. 1210–1239. Doi: 10.1016/j.rser.2017.01.075.

[32] Plötz, P., Funke, S. Á. and Jochem, P. (2018). The impact of daily and annual driving on fuel economy and CO2 emissions of plug-in hybrid electric vehicles. *Transportation Research Part A: Policy and Practice*. Elsevier, 118(January 2016), pp. 331–340. Doi: 10.1016/j.tra.2018.09.018.

[33] Onori, S., Serrao, L., Rizzoni, G. (2016). Hybrid Electric Vehicles. Springer, London, UK.

[34] Ehsani, M., Gao, Y., Longo, S., Ebrahimi, K.M. (2018). Modern Electric Hybrid Electric and Fuel Cell Vehicles. CRC Press, Boca Raton, FL, USA.

[35] Anselma, P. G. *et al.* (2019). Comparing parallel hybrid electric vehicle powertrains for real-world driving. *2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2019*. Doi: 10.23919/EETA.2019.8804609.

[36] Varga, B. O., Sagoian, A. and Mariasiu, F. (2019). Prediction of electric vehicle range: A comprehensive review of current issues and challenges. *Energies*, 12(5). Doi: 10.3390/en12050946.

[37] Cannavacciuolo, G. *et al.* (2021). A Model for the Estimation of the Residual Driving Range of Battery Electric Vehicles Including Battery Ageing , Thermal Effects and Auxiliaries. *Applied Sciences*, 11(19). Doi: 10.3390/app11199316.

[38] Verma, S. *et al.* (2021). A comprehensive review on energy storage in hybrid electric vehicle. *Journal of Traffic and Transportation Engineering (English Edition)*. Elsevier Ltd, (xxx), pp. 1–17. Doi: 10.1016/j.jtte.2021.09.001.

[39] Turing, A. M. (1950). I. – COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236), pp. 433–460, Doi: 10.1093/mind/LIX.236.433.

[40] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), pp. 386–408. Doi: 10.1037/h0042519.

[41] Mccarthy, J. *et al.* (2006). Dartmouth Summer. *AI Magazine*, 27(4), pp. 12–14.

[42] High-Level Independent Group on Artificial Intelligence (AI HLEG). A Definition of AI: Main Capabilities and Disciplines [online]. Available at: https://digital-strategy.ec.europa.eu/en/library/definition-artificial-intelligence-main-capabilities-and-scientific-disciplines.

[43] Kotsiantis, S. B., Zaharakis, I. D. and Pintelas, P. E. (2006). Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, 26(3), pp. 159–190. Doi: 10.1007/s10462-007-9052-3.

[44] Goodfellow, I. J., Bengio, Y. and Courville, A. (2016). Deep Learning. MIT Press.

[45] Dike, H. U. *et al.* (2019). Unsupervised Learning Based On Artificial Neural Network: A Review'. *2018 IEEE International Conference on Cyborg and Bionic Systems, CBS 2018*. IEEE, pp. 322–327. Doi: 10.1109/CBS.2018.8612259.

[46] Kaelbling, L. P., Littman, M. L. and Moore, A. W. (1996). Reinforcement Learning: A Survey. Journal of *Artificial Intelligence Research*, 4, pp. 237-285. Doi: 10.1613/jair.301.

[47] Sutton, R. S. and Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.

[48] Singh, K. V., Bansal, H. O. and Singh, D. (2019). A comprehensive review on hybrid electric vehicles: architectures and components. *Journal of Modern Transportation*. Springer Singapore, 27(2), pp. 77–107. Doi: 10.1007/s40534-019-0184-3.

[49] Silvaş, E., Hofman, T. and Steinbuch, M. (2012). Review of optimal design strategies for hybrid electric vehicles. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 45(30), pp. 57–64. Doi: 10.3182/20121023-3-FR-4025.00054.

[50] Anselma, P. G. and Belingardi, G. (2019). Next Generation HEV Powertrain Design Tools: Roadmap and Challenges'. *SAE Technical Papers*, (October), pp. 1–15. Doi: 10.4271/2019-01-2602.

[51] Finesso, R., Spessa, E. and Venditti, M. (2016). Robust equivalent consumption-based controllers for a dual-mode diesel parallel HEV. *Energy Conversion and Management*. Elsevier Ltd, 127, pp. 124–139. Doi: 10.1016/j.enconman.2016.08.021.

[52] Zhang, P., Yan, F. and Du, C. (2015). A comprehensive analysis of energy management strategies for hybrid electric vehicles based on bibliometrics. *Renewable and Sustainable Energy Reviews*. Elsevier, 48(205), pp. 88–104. Doi: 10.1016/j.rser.2015.03.093.

[53] Bertsekas, D. P. (1995). Dynamic Programming and Optimal Control. Athena Scientific, Belmont.

[54] Bellman, R. (1957). Dynamic Programming. Princeton University Press.

[55] Sundström, O. and Guzzella, L. (2009). A generic dynamic programming Matlab function. *Proceedings of the IEEE International Conference on Control Applications*, (7), pp. 1625–1630. Doi: 10.1109/CCA.2009.5281131.

[56] Finesso, R., Spessa, E. and Venditti, M. (2014). Layout design and energetic analysis of a complex diesel parallel hybrid electric vehicle. *Applied Energy*. Elsevier Ltd, 134, pp. 573–588. Doi: 10.1016/j.apenergy.2014.08.007.

[57] Finesso, R., Spessa, E. and Venditti, M. (2014). Optimization of the layout and control strategy for parallel through-the-road hybrid electric vehicles. *SAE Technical Papers*, 1. Doi: 10.4271/2014-01-1798.

[58] Finesso, R., Spessa, E. and Venditti, M. (2016). Cost-optimized design of a dual-mode diesel parallel hybrid electric vehicle for several driving missions and market scenarios. *Applied Energy*. Elsevier Ltd, 177, pp. 366–383. Doi: 10.1016/j.apenergy.2016.05.080.

[59] Nüesch, T. *et al.* (2014). Convex optimization for the energy management of hybrid electric vehicles considering engine start and gearshift costs. *Energies*, 7(2), pp. 834–856. Doi: 10.3390/en7020834.

[60] Gissing, J. *et al.* (2016). Optimal Control of Series Plug-In Hybrid Electric Vehicles Considering the Cabin Heat Demand. *IEEE Transactions on Control Systems Technology*, 24(3), pp. 1126–1133. Doi: 10.1109/TCST.2015.2468055.

[61] Delprat, S. *et al.* (2004). Control of a parallel hybrid powertrain: Optimal control'. *IEEE Transactions on Vehicular Technology*, 53(3), pp. 872–881. Doi: 10.1109/TVT.2004.827161.

[62] Desai, C. and Williamson, S. S. (2009). Optimal design of a parallel hybrid electric vehicle using multi-objective genetic algorithms. *5th IEEE Vehicle Power and Propulsion Conference, VPPC '09*. IEEE, pp. 871–876. Doi: 10.1109/VPPC.2009.5289754.

[63] Spano, M. *et al.* (2021). Exploitation of a particle swarm optimization algorithm for designing a lightweight parallel hybrid electric vehicle. *Applied Sciences (Switzerland)*, 11(15). Doi: 10.3390/app11156833.

[64] Oh, K. *et al.* (2007). Optimization of control strategy for a single-shaft parallel hybrid electric vehicle. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 221(5), pp. 555–565. Doi: 10.1243/09544070JAUTO93.

[65] Paganelli, G. *et al.* (2000). Simulation and assessment of power control strategies for a parallel hybrid car. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 214(7), pp. 705–717. Doi: 10.1243/0954407001527583.

[66] Sciarretta, A., Back, M. and Guzzella, L. (2004). Optimal Control of Parallel Hybrid Electric Vehicles. *IEEE Transactions on Control Systems Technology*, 12(3), pp. 352–363. Doi: 10.1109/TCST.2004.824312.

[67] Musardo, C. *et al.* (2005). A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management. *European Journal of Control*. Elsevier, 11(4–5), pp. 509–524. Doi: 10.3166/ejc.11.509-524.

[68] Pesce, T. and Lienkamp, M. (2012). Definition and optimization of the drive train topology for electric vehicles. *World Electric Vehicle Journal*, 5(1), pp. 24–35. Doi: 10.3390/wevj5010024.

[69] Nicoletti, L. *et al.* (2021). Topology analysis of electric vehicles, with a focus on the traction battery. *Forschung im Ingenieurwesen/Engineering Research*, 85(2), pp. 457–467. Doi: 10.1007/s10010-020-00422-1.

[70] Cipollone, R. *et al.* (2014). Model based design and optimization of a fuel cell electric vehicle. *Energy Procedia*. Elsevier B.V., 45, pp. 71–80. Doi: 10.1016/j.egypro.2014.01.009.

[71] Di Battista, D., Villante, C. and Cipollone, R. (2015). Optimal Components Design of a Fuel Cell Electric Vehicle. *SAE Technical Papers*, 2015-September(September). Doi: 10.4271/2015-24-2546.

[72] Zhang, X. *et al.* (2015). Efficient exhaustive search of power-split hybrid powertrains with multiple planetary gears and clutches. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 137(12). Doi: 10.1115/1.4031533.

[73] Li, D. *et al.* (2008). Mitigation of Curse of Dimensionality in Dynamic Programming'. *IFAC Proceedings Volumes*, 41(2), pp. 7778–7783. Doi: 10.3182/20080706-5-kr-1001.01315.

[74] Biswas, A. and Emadi, A. (2019). Energy management systems for electrified powertrains: State-of-the-art review and future trends. *IEEE Transactions on Vehicular Technology*. IEEE, 68(7), pp. 6453–6467. Doi: 10.1109/TVT.2019.2914457.

[75] Torreglosa, J. P. *et al.* (2020). Analyzing the improvements of energy management systems for hybrid electric vehicles using a systematic literature review: How far are these controls from rule-based controls used in commercial vehicles?. *Applied Sciences (Switzerland)*, 10(23), pp. 1–25. Doi: 10.3390/app10238744.

[76] Ali, A. M. and Söffker, D. (2018). Towards optimal power management of hybrid electric vehicles in real-time: A review on methods, challenges, and state-of-the-art solutions. *Energies*, 11(3), pp. 1–24. Doi: 10.3390/en11030476.

[77] Martinez, C. M. *et al.* (2017). Energy Management in Plug-in Hybrid Electric Vehicles: Recent Progress and a Connected Vehicles Perspective. *IEEE Transactions on Vehicular Technology*, 66(6), pp. 4534–4549. Doi: 10.1109/TVT.2016.2582721.

[78] Lin, C. C. *et al.* (2003). Power management strategy for a parallel hybrid electric truck. IEEE Trans Control Syst Technol'. *IEEE Transactions on Control Systems Technology*, 11(6), pp. 839–849. Doi: 10.1109/TCST.2003.815606.

[79] Piccolo, A. *et al.* (2001). Optimisation of energy flow management in hybrid electric vehicles via genetic algorithms. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 1(July), pp. 434–439. Doi: 10.1109/aim.2001.936493.

[80] Hofman, T. *et al.* (2008). Rule-Based Equivalent Fuel Consumption Minimization Strategies for Hybrid Vehicles. *IFAC Proceedings Volumes*. IFAC. Doi: 10.3182/20080706-5-kr-1001.00953.

[81] Pei, D. and Leamy, M. J. (2013). Dynamic programming-informed equivalent cost minimization control strategies for hybrid-electric vehicles. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 135(5). Doi: 10.1115/1.4024788.

[82] Finesso, R., Spessa, E. and Venditti, M. (2016). Robust equivalent consumption-based controllers for a dual-mode diesel parallel HEV. *Energy Conversion and Management*. Elsevier Ltd, 127, pp. 124–139. Doi: 10.1016/j.enconman.2016.08.021.

[83] Park, J. and Park, J. H. (2012). Development of equivalent fuel consumption minimization strategy for hybrid electric vehicles. International Journal of Automotive Technology, 13(5), pp. 835-843. Doi: 10.1007/s12239−012−0084−6.

[84] Borhan, H. A. *et al.* (2009). Predictive energy management of a power-split hybrid electric vehicle. *Proceedings of the American Control Conference*. IEEE, pp. 3970–3976. Doi: 10.1109/ACC.2009.5160451.

[85] Borhan, H. *et al.* (2012). MPC-based energy management of a power-split hybrid electric vehicle. *IEEE Transactions on Control Systems Technology*, 20(3), pp. 593–603. Doi: 10.1109/TCST.2011.2134852.

[86] Song, C. *et al.* (2021). A review of optimal energy management strategies using machine learning techniques for hybrid electric vehicles. *International Journal of Automotive Technology*, 22, pp. 1437–1452. Doi: 10.1007/s12239-021-0125-0.

[87] Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*. Elsevier Ltd, 61, pp. 85–117. Doi: 10.1016/j.neunet.2014.09.003.

[88] Park, J. *et al.* (2009). Intelligent vehicle power control based on machine learning of optimal control parameters and prediction of road type and traffic congestion'. *IEEE Transactions on Vehicular Technology*, 58(9), pp. 4741–4756. Doi: 10.1109/TVT.2009.2027710.

[89] Murphey, Y. L. *et al.* (2012). Intelligent hybrid vehicle power control part I: Machine learning of optimal vehicle power. *IEEE Transactions on Vehicular Technology*. IEEE, 61(8), pp. 3519–3530. Doi: 10.1109/TVT.2012.2206064.

[90] Murphey, Y. L. *et al.* (2013). Intelligent hybrid vehicle power control - Part II: Online intelligent energy management. *IEEE Transactions on Vehicular Technology*. IEEE, 62(1), pp. 69–79. Doi: 10.1109/TVT.2012.2217362.

[91] Chen, Z. *et al.* (2014). Energy management for a power-split plug-in hybrid electric vehicle based on dynamic programming and neural networks. *IEEE Transactions on Vehicular Technology*. IEEE, 63(4), pp. 1567–1580. Doi: 10.1109/TVT.2013.2287102.

[92] Wu, Y. *et al.* (2020). A predictive energy management strategy for multi-mode plug-in hybrid electric vehicles based on multi neural networks. *Energy*. Elsevier Ltd, 208, p. 118366. Doi: 10.1016/j.energy.2020.118366.

[93] Mozaffari, S. *et al.* (2020). Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15. Doi: 10.1109/tits.2020.3012034.

[94] Han, L., Jiao, X. and Zhang, Z. (2020). Recurrent neural network-based adaptive energy management control strategy of plug-in hybrid electric vehicles considering battery aging. *Energies*, 13(1), pp. 1–22. Doi: 10.3390/en13010202.

[95] Feldkamp, L., Nasr, M. A. and Kolmanovsky, I. V. (2009). Recurrent neural network training for energy management of a mild hybrid electric vehicle with an ultra-capacitor. *2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, CIVVS 2009 - Proceedings*. IEEE, pp. 29–36. Doi: 10.1109/CIVVS.2009.4938720.

[96] Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*. Elsevier B.V., 404, p. 132306. Doi: 10.1016/j.physd.2019.132306.

[97] Ruan, S. and Ma, Y. (2020). Real-Time Energy Management Strategy Based on Driver-Action-Impact MPC for Series Hybrid Electric Vehicles. *Complexity*, 2020. doi: 10.1155/2020/8843168.

[98] Wei, Z. and Zhang, Y. (2021). A Real-time Energy Management Strategy of HEVs Based on Velocity Prediction. *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. The Society of Instrument and Control Engineers - SICE, pp. 490–494.

[99] Zhang, Z. *et al.* (2020). Velocity prediction and profile optimization based real-time energy management strategy for Plug-in hybrid electric buses. *Applied Energy*. Elsevier Ltd, 280(February), p. 116001. Doi: 10.1016/j.apenergy.2020.116001.

[100] Chen, Z., Yang, C. and Fang, S. (2020). A Convolutional Neural Network-Based Driving Cycle Prediction Method for Plug-in Hybrid Electric Vehicles with Bus Route. *IEEE Access*. IEEE, 8, pp. 3255–3264. Doi: 10.1109/ACCESS.2019.2960771.

[101] Sun, C. *et al.* (2015). Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles. *IEEE Transactions on Control Systems Technology*. IEEE, 23(3), pp. 1197–1204. Doi: 10.1109/TCST.2014.2359176.

[102] Sun, C. *et al.* (2015). Dynamic Traffic Feedback Data Enabled Energy Management in Plug-in Hybrid Electric Vehicles. *IEEE Transactions on Control Systems Technology*. IEEE, 23(3), pp. 1075–1086. Doi: 10.1109/TCST.2014.2361294.

[103] Laroui, M. *et al.* (2019). Energy management for electric vehicles in smart cities: A deep learning approach. *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, pp. 2080–2085. Doi: 10.1109/IWCMC.2019.8766580.

[104] Petkevicius, L. *et al.* (2021). Probabilistic Deep Learning for Electric-Vehicle Energy-Use Prediction. *ACM International Conference Proceeding Series*, pp. 85–95. Doi: 10.1145/3469830.3470915.

[105] Gong, Q., Li, Y. and Peng, Z. (2009). Power management of plug-in hybrid electric vehicles using neural network based trip modeling. *Proceedings of the American Control Conference*. IEEE, pp. 4601–4606. Doi: 10.1109/ACC.2009.5160623.

[106] Lee, H. *et al.* (2016). A study on power management strategy of HEV using dynamic programming. *EVS 2016 - 29th International Electric Vehicle Symposium*, 8, pp. 274–280.

[107] Majed, C., Karaki, S. H. and Jabr, R. (2016). Neural network technique for hybrid electric vehicle optimization. *Proceedings of the 18th Mediterranean Electrotechnical Conference: Intelligent and Efficient Technologies and Services for the Citizen, MELECON 2016*. IEEE, (April), pp. 1–6. Doi: 10.1109/MELCON.2016.7495327.

[108] Xi, L. *et al.* (2017). Intelligent energy management control for extended range electric vehicles based on dynamic programming and neural network. *Energies*, 10(11). Doi: 10.3390/en10111871.

[109] Abdullah, H. M., Gastli, A. and Ben-Brahim, L. (2021). Reinforcement Learning Based EV Charging Management Systems-A Review. *IEEE Access*, 9, pp. 41506–41531. Doi: 10.1109/ACCESS.2021.3064354.

[110] Liu, C. and Murphey, Y. L. (2014). Power management for plug-in hybrid electric vehicles using reinforcement learning with trip information. *2014 IEEE Transportation Electrification Conference and Expo: Components, Systems, and Power Electronics - From Technology to Business and Public Policy, ITEC 2014*. IEEE, pp. 1–6. Doi: 10.1109/itec.2014.6861862.

[111] Liu, T. *et al.* (2015). Reinforcement Learning of Adaptive Energy Management With Transition Probability for a Hybrid Electric Tracked Vehicle. *IEEE Transactions on Industrial Electronics*. IEEE, 62(12), pp. 7837–7846. Doi: 10.1109/TIE.2015.2475419.

[112] Liu, T. *et al.* (2017). Reinforcement Learning Optimized Look-Ahead Energy Management of a Parallel Hybrid Electric Vehicle. *IEEE/ASME Transactions on Mechatronics*, 22(4), pp. 1497–1507. Doi: 10.1109/TMECH.2017.2707338.

[113] Mittal, N. *et al.* (2020). Optimization of Energy Management Strategy for Range-Extended Electric Vehicle Using Reinforcement Learning and Neural Network. *SAE Technical Papers*, 2020-April(April), pp. 1–12. Doi: 10.4271/2020-01-1190.

[114] Biswas, A., Anselma, P. G. and Emadi, A. (2019). Real-Time Optimal Energy Management of Electrified Powertrains with Reinforcement Learning. *ITEC 2019 - 2019 IEEE Transportation Electrification Conference and Expo*, (1), pp. 8–13. Doi: 10.1109/ITEC.2019.8790482.

[115] Xu, B. *et al.* (2019). Real-Time reinforcement learning optimized energy management for a 48V mild hybrid electric vehicle. *SAE Technical Papers*, 2019-April(April), pp. 1–9. Doi: 10.4271/2019-01-1208.

[116] Xu, B. *et al.* (2020). Parametric study on reinforcement learning optimized energy management strategy for a hybrid electric vehicle. *Applied Energy*. Elsevier, 259(November 2019), p. 114200. Doi: 10.1016/j.apenergy.2019.114200.

[117] Reddy, N. P. *et al.* (2019). An Intelligent Power and Energy Management System for Fuel Cell/Battery Hybrid Electric Vehicle Using Reinforcement Learning. *ITEC 2019 - 2019 IEEE Transportation Electrification Conference and Expo*. Doi: 10.1109/ITEC.2019.8790451.

[118] Gao, D. W., Mi, C. and Emadi, A. (2007). Modeling and simulation of electric and hybrid vehicles. *Proceedings of the IEEE*, 95(4), pp. 729–745. Doi: 10.1109/JPROC.2006.890127.

[119] Redish, E. F. (2017). Analysing the competency of mathematical modelling in physics. *Springer Proceedings in Physics*. Springer Science and Business Media, LLC, pp. 25–40. Doi: 10.1007/978-3-319-44887-9_3.

[120] Mohan, G., Assadian, F. and Longo, S. (2013). Comparative analysis of forward-facing models vs backward-facing models in powertrain component sizing. *IET Conference Publications*, 2013(621 CP), pp. 1–6. doi: 10.1049/cp.2013.1920.

[121] Chan, C. C., Bouscayrol, A. and Chen, K. (2010). Electric, hybrid, and fuel-cell vehicles: Architectures and modeling. *IEEE Transactions on Vehicular Technology*, 59(2), pp. 589–598. Doi: 10.1109/TVT.2009.2033605.

[122] Yang, Y. *et al.* (2016). State-of-the-art electrified powertrains - hybrid, plug-in, and electric vehicles. *International Journal of Powertrains*, 5(1), p. 1. Doi: 10.1504/ijpt.2016.075181.

[123] Genta, G. (1997) *Motor Vehicle Dynamics: Modeling and Simulation*. World Scientific, Singapore.

[124] Guzzella, L. and Sciarretta, A. (2007). *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Springer, Berlin, Heidelberg. Doi: 10.1007/3-540-28853-8.

[125] Bauskar, M. P. *et al.* (2019). Study of aerodynamic drag of sports utility vehicle by experimental and numerical method. *Materials Today: Proceedings*. Elsevier Ltd., 16, pp. 750–757. Doi: 10.1016/j.matpr.2019.05.155.

[126] Andersen, L. G. *et al.* (2015). Rolling resistance measurement and model development. *Journal of Transportation Engineering*, 141(2), pp. 1–10. Doi: 10.1061/(ASCE)TE.1943-5436.0000673.

[127] Smith, G. D. (1986). *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford University Press, Oxford.

[128] Pacejka, H. B. (2012). *Tyre and Vehicle Dynamics*. Butterworth-Heinemann, Oxford.

[129] Heywood, J. B. (1988). *Internal Combustion Engine Fundamentals*. McGraw-Hill, New York.

[130] Reşitoğlu, İ. A., Altinişik, K. and Keskin, A. (2015). The pollutant emissions from diesel-engine vehicles and exhaust aftertreatment systems. *Clean Techn Environ Policy 17*, pp. 15–27. Doi: 10.1007/s10098-014-0793-9.

[131] Joshi, A. (2021) 'Review of Vehicle Engine Efficiency and Emissions', *SAE Technical Papers*, 2(2021), pp. 2479–2507. Doi: 10.4271/2021-01-0575.

[132] Un-Noor, F. *et al.* (2017). A comprehensive study of key electric vehicle (EV) components, technologies, challenges, impacts, and future direction of development. *Energies*, 10(8), pp. 1–82. Doi: 10.3390/en10081217.

[133] Wang, W., Chen, X. and Wang, J. (2019). Motor/Generator Applications in Electrified Vehicle Chassis-A Survey. *IEEE Transactions on Transportation Electrification*. IEEE, 5(3), pp. 584–601. Doi: 10.1109/TTE.2019.2934340.

[134] Lipu, M. S. H. *et al.* (2021). Review of electric vehicle converter configurations, control schemes and optimizations: Challenges and suggestions. *Electronics (Switzerland)*, 10(4), pp. 1–37. Doi: 10.3390/electronics10040477.

[135] He, H., Xiong, R. and Fan, J. (2011). Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies*, 4(4), pp. 582–598. Doi: 10.3390/en4040582.

[136] Johnson, V. H. (2002). Battery performance models in ADVISOR. *Journal of Power Sources*, 110(2), pp. 321–329. Doi: 10.1016/S0378-7753(02)00194-5.

[137] Battery University. BU-402: What Is C-rate? [online: https://batteryuniversity.com/article/bu-402-what-is-c-rate] (accessed 25th November 2021).

[138] Plett, G. L. (2004). Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs - Part 3. State and parameter estimation. *Journal of Power Sources*, 134(2), pp. 277–292. doi: 10.1016/j.jpowsour.2004.02.033.

[139] Gyenes, B. *et al.* (2015). Understanding Anomalous Behavior in Coulombic Efficiency Measurements on Li-Ion Batteries. *Journal of The Electrochemical Society*, 162(3), pp. A278–A283. Doi: 10.1149/2.0191503jes.

[140] Wang, R. and Lukic, S. M. (2012). Dynamic programming technique in hybrid electric vehicle optimization. *2012 IEEE International Electric Vehicle Conference, IEVC 2012*. IEEE. Doi: 10.1109/IEVC.2012.6183284.

[141] Carroll, Y. and Johnson, C. D. (2009). Optimal sequential-decisions in the presence of exogenous-inputs with structured variations. *Proceedings of the Annual Southeastern Symposium on System Theory*. IEEE, pp. 300–305.

[142] Yang, Y. *et al.* (2016). Comparison of power-split and parallel hybrid powertrain architectures with a single electric machine: Dynamic programming approach. *Applied Energy*. Elsevier Ltd, 168, pp. 683–690. Doi: 10.1016/j.apenergy.2016.02.023.

[143] Xiao, R. *et al.* (2018). Comparisons of energy management methods for a parallel plug-in hybrid electric vehicle between the convex optimization and dynamic programming. *Applied Sciences (Switzerland)*, 8(2), pp. 16–19. Doi: 10.3390/app8020218.

[144] Miretti, F., Misul, D. and Spessa, E. (2021). DynaProg: Deterministic Dynamic Programming solver for finite horizon multi-stage decision problems. *SoftwareX*. Elsevier B.V., 14, p. 100690. Doi: 10.1016/j.softx.2021.100690.

[145] Finesso, R. *et al.* (2018). Optimal design of power-split HEVs based on total cost of ownership and CO2 emission minimization. *Energies*, 11(7). Doi: 10.3390/en11071705.

[146] Silvas, E. (2015). *Integrated optimal design for hybrid electric vehicles*. Technische Universiteit Eindhoven.

[147] Gao, W. and Porandla, S. K. (2005). Design optimization of a parallel hybrid electric powertrain. *2005 IEEE Vehicle Power and Propulsion Conference, VPPC*, 2005, pp. 530–535. Doi: 10.1109/VPPC.2005.1554609.

[148] Sundström, O., Guzzella, L. and Soltic, P. (2010). Torque-assist hybrid electric powertrain sizing: From optimal control towards a sizing law. *IEEE Transactions on Control Systems Technology*, 18(4), pp. 837–849. Doi: 10.1109/TCST.2009.2030173.

[149] Sinoquet, D., Rousseau, G. and Milhau, Y. (2011). Design optimization and optimal control for hybrid vehicles. *Optimization and Engineering*, 12(1–2), pp. 199–213. Doi: 10.1007/s11081-009-9100-8.

[150] Shams-Zahraei, M. *et al.* (2012). Integrated thermal and energy management of plug-in hybrid electric vehicles. *Journal of Power Sources*, 216, pp. 237–248. Doi: 10.1016/j.jpowsour.2012.05.055.

[151] Law, A. M. (2014). *Simulation Modeling and Analysis.* 5th Edition. McGraw-Hill Education.

[152] Burhenne, S., Jacob, D. and Henze, G. P. (2011). SAMPLING BASED ON SOBOL SEQUENCES FOR MONTE CARLO TECHNIQUES APPLIED TO BUILDING SIMULATIONS. *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association.*, pp. 14–16.

[153] Chon, D. M. and Heywood, J. B. (2000). Performance scaling of spark-ignition engines: Correlation and historical analysis of production engine data. *SAE Technical Papers*, (724). Doi: 10.4271/2000-01-0565.

[154] Maino, C. *et al.* (2021). Optimal mesh discretization of the dynamic programming for hybrid electric vehicles. *Applied Energy*. Elsevier Ltd, 292(March), p. 116920. Doi: 10.1016/j.apenergy.2021.116920.

[155] Villafuerte, P. M., Bonnel, P. and Riccobono, F. (2015). *EU-PEMS PM Pilot Program : Testing , data analysis and results.* Doi: 10.2790/46867.

[156] Fontaras, G., Zacharof, N. G. and Ciuffo, B. (2017). Fuel consumption and CO2 emissions from passenger cars in Europe – Laboratory versus real-world emissions. *Progress in Energy and Combustion Science*. Elsevier Ltd, 60, pp. 97–131. Doi: 10.1016/j.pecs.2016.12.004.

[157] Harselaar, W. Van *et al.* (2019). Improved Implementation of Dynamic Programming on the Example of Hybrid Electric Vehicle Control. *IFAC-PapersOnLine*. Elsevier Ltd, 52(5), pp. 147–152. Doi: 10.1016/j.ifacol.2019.09.024.

[158] Van Harselaar, W., Hofman, T. and Brouwer, M. (2018). Automated Dynamic Modeling of Arbitrary Hybrid and Electric Drivetrain Topologies. *IEEE Transactions on Vehicular Technology*. IEEE, 67(8), pp. 6921–6934. Doi: 10.1109/TVT.2018.2834537.

[159] Haag A. *et al.* (2016). Iterative refinement of the discretization of the Dynamic Programming State Grid. In: Bargende M., Reuss HC., Wiedemann J. (eds) 16. Internationales Stuttgarter Symposium. Proceedings. Springer, Wiesbaden. Doi: 10.1007/978-3-658-13255-2_12.

[160] Danfoss. Electric machines. Drivetrain power for vehicles, machines and marine applications, [online: https://www.danfoss.com/en/products/dps/electric-converters-and-machines/electric-converters-and-machines/electric-machines/] (accessed 10th December 2021).

[161] Busoniu, L. *et al.* (2009). *Reinforcement learning and dynamic programming using function approximators*.

[162] Cox, D. (2006). *Principles of Statistical Inference*. Cambridge: Cambridge University Press. Doi:10.1017/CBO9780511813559

[163] Arnold, B. C., Groeneveld, R. A. (1995). Measuring Skewness with Respect to the Mode. *The American Statistician*, 49:1, pp. 34-38. Doi: 10.1080/00031305.1995.10476109.

[164] Tabachnick B.G., Fidell L.S. and Ullman J. (2007). *Using multivariate statistics*. Boston, MA. Pearson.

[165] Bradley, J. (1963). Part IV: A sampling study of the central limit theorem and the robustness of one-sample parametric tests. In *Studies in research methodology*.

[166] DieselNet. Emission Standards. Summary of worldwide engine and vehicle emission standards, [online: https://dieselnet.com/standards/] (accessed 15th December 2021).

[167] Garbade, M. J. (2021). Regression Versus Classification Machine Learning: What's the Difference?, [online: https://medium.com/quick-code/regression-versus-classification-machine-learning-whats-the-difference-345c56dd15f7] (accessed 17th December 2021).

[168] Bertsekas, D. P. (1982). Distributed Dynamic Programming. *IEEE Transactions on Automatic Control*, 27(3), pp. 610–616. Doi: 10.1109/TAC.1982.1102980.

[169] Joshi, S. *et al.* (2010). Classification of Alzheimer's disease and Parkinson's disease by using machine learning and neural network methods. *ICMLC 2010 - The 2nd International Conference on Machine Learning and Computing*. IEEE, pp. 218–222. Doi: 10.1109/ICMLC.2010.45.

[170] Abe, S. (1997). Overview of Neural Networks. In *Neural Networks and Fuzzy Systems*. Springer, Boston, MA. Doi: 10.1007/978-1-4615-6253-5_1.

[171] Hailesilassie, T. (2016). Rule Extraction Algorithm for Deep Neural Networks: A Review. 14(7), pp. 376–381. Available at: http://arxiv.org/abs/1610.05267.

[172] Koutsoukas, A. *et al.* (2017). Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of Cheminformatics*. Springer International Publishing, 9(1), pp. 1–13. Doi: 10.1186/s13321-017-0226-y.

[173] Korotcov, A. *et al.* (2017). Comparison of Deep Learning with Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets. *Molecular Pharmaceutics*, 14(12), pp. 4462–4475. Doi: 10.1021/acs.molpharmaceut.7b00578.

[174] Maino, C. *et al.* (2021). A deep neural network based model for the prediction of hybrid electric vehicles carbon dioxide emissions. *Energy and AI*. Elsevier Ltd, 5, p. 100073. Doi: 10.1016/j.egyai.2021.100073.

[175] Emmert-Streib, F. *et al.* (2020). An Introductory Review of Deep Learning for Prediction Models With Big Data. *Frontiers in Artificial Intelligence*, 3(February), pp. 1–23. Doi: 10.3389/frai.2020.00004.

[176] Van Rossum, G and Drake, F. L. (2009). *Python 3 Reference Manual.* CreateSpace, Scotts Valley, CA.

[177] Singh, D. and Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*. Elsevier B.V., 97, p. 105524. Doi: 10.1016/j.asoc.2019.105524.

[178] Nayak, S. C., Misra, B. B. and Behera, H. S. (2014). Impact of Data Normalization on Stock Index Forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, 6(2014), pp. 257–269.

[179] Shen, X. *et al.* (2016). Normalization and integration of large-scale metabolomics data using support vector regression. *Metabolomics*. Springer US, 12(5), pp. 1–12. Doi: 10.1007/s11306-016-1026-5.

[180] Bhanja, S. and Das, A. (2018). Impact of Data Normalization on Deep Neural Network for Time Series Forecasting. pp. 5–10. Available at: http://arxiv.org/abs/1812.05519.

[181] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, pp. 281–305.

[182] Darapureddy, N., Karatapu, N. and Battula, T. K. (2019). Research of machine learning algorithms using k-fold cross validation. *International Journal of Engineering and Advanced Technology*, 8(6 Special issue), pp. 215–218. Doi: 10.35940/ijeat.F1043.0886S19.

[183] Chicco, D. and Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*. BMC Genomics, 21(1), pp. 1–13. Doi: 10.1186/s12864-019-6413-7.

[184] Ozer, D. J. (1985). Correlation and the coefficient of determination. *Psychological Bulletin, 97*(2), 307–315. Doi: 10.1037/0033-2909.97.2.307.

[185] Cameron, A. C. and Windmeijer, F. A. G. (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77(2), pp. 329–342. Doi: 10.1016/s0304-4076(96)01818-0.

[186] Kumar, S. (2020). Common Loss functions in machine learning for a Regression model, [online: https://medium.com/analytics-vidhya/common-

loss-functions-in-machine-learning-for-a-regression-model-27d2bbda9c93] (accessed 3rd January 2022)

[187] Koehrsen, W. (2018). Overfitting vs. Underfitting: A Complete Example, [online: https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765] (accessed 4th January 2022)

[188] Statology. RMSE vs. R-Squared: Which Metric Should You Use?, [online: https://www.statology.org/rmse-vs-r-squared/] (accessed 12th January 2022).

[189] Rumelhart, D. E., Hinton, G. and Williams, R. J. (1986a). Learning representations by back-propagating errors. *Nature*, 323, pp. 533–536.

[190] Graves, A (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, Berlin, Heidelberg. Doi: 10.1007/978-3-642-24797-2.

[191] Pascanu, R., Mikolov, T. and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *30th International Conference on Machine Learning, ICML 2013*, (PART 3), pp. 2347–2355.

[192] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput* 1997, 9 (8), pp. 1735–1780. Doi: 10.1162/neco.1997.9.8.1735.

[193] Cho, K. *et al.* (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734. Doi: 10.3115/v1/d14-1179.

[194] Marina Martinez, C. *et al.* (2018). Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3), pp. 666–676. Doi: 10.1109/TITS.2017.2706978.

[195] Chan, W. *et al.* (2015). Listen, Attend and Spell. Pp. 1–16. Available at: http://arxiv.org/abs/1508.01211.

[196] Nwankpa, C. *et al.* (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. Pp. 1–20. Available at: http://arxiv.org/abs/1811.03378.

[197] Grandini, M., Bagli, E. and Visani, G. (2020). Metrics for Multi-Class Classification: an Overview. Pp. 1–17. Available at: http://arxiv.org/abs/2008.05756.

[198] Vaswani, A. *et al.* (2017). Attention is all you need, *Advances in Neural Information Processing Systems*, 2017-December(Nips), pp. 5999–6009. Available at: https://arxiv.org/abs/1706.03762v5.

[199] Montazeri-Gh, M., Fotouhi, A. and Naderpour, A. (2011). Driving patterns clustering based on driving feature analysis. *Proceedings of the Institution of*

Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 225(6), pp. 1301–1317. Doi: 10.1177/2041298310392599.

[200] Finesso, R., Spessa, E. and Venditti, M. (2016). An unsupervised machine-learning technique for the definition of a rule-based control strategy in a complex HEV. *SAE International Journal of Alternative Powertrains*, 5(2). Doi: 10.4271/2016-01-1243.

[201] Ganesh, A. H. and Xu, B. (2022). A review of reinforcement learning based energy management systems for electrified powertrains: Progress, challenge, and potential solution. *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, 154(October 2021), p. 111833. Doi: 10.1016/j.rser.2021.111833.

[202] Spaan, M. T. J. (2012). Partially Observable Markov Decision Processes. In *Reinforcement Learning. Adaptation, Learning and Optimization*. Springer, Berlin/Heidelberg, Germany. Doi: 10.1007/978-3-642-27645-3_12.

[203] Watkins, C. J. C. H. (1989). Learning from Delayed Rewards. PhD Thesis, King's College, London, UK.

[204] Maino, C. *et al.* (2022). Project and Development of a Reinforcement Learning Based Control Algorithm for Hybrid Electric Vehicles. *Applied Sciences*, 12, p. 812. Doi: 10.3390/app12020812.

[205] Van Rossum, G. (2020). *The Python Library Reference, Release 3.8.5*. Python Software Foundation, Wilmington, DE, USA.

[206] Fusco, G. *et al.* (2018). Experimental analyses and clustering of travel choice behaviours by floating car big data in a large urban area. *IET Intelligent Transport Systems*, 12(4), pp. 270–278. Doi: 10.1049/iet-its.2018.0015.