POLITECNICO DI TORINO
Repository ISTITUZIONALE

Development of a neural network-based energy management system for a plug-in hybrid electric vehicle

Original
Development of a neural network-based energy management system for a plug-in hybrid electric vehicle / Millo, F., Rolando, L., Tresca, L., Pulvirenti, L.. - In: TRANSPORTATION ENGINEERING. - ISSN 2666-691X. - ELETTRONICO. - 11:(2023), p. 100156. [10.1016/j.treng.2022.100156]

Availability:
This version is available at: 11583/2973887 since: 2022-12-15T15:36:31Z

Publisher:
Elsevier

Published
DOI:10.1016/j.treng.2022.100156

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

26 November 2026

Transportation Engineering 11 (2023) 100156

Contents lists available at ScienceDirect

Transportation Engineering

journal homepage: www.sciencedirect.com/journal/transportation-engineering

Full Length Article

Development of a neural network-based energy management system for a plug-in hybrid electric vehicle

Federico Millo *, Luciano Rolando, Luigi Tresca, Luca Pulvirenti

Politecnico di Torino, C.so Duca degli Abruzzi, 24, Turin 10129, TO, Italy

ARTICLE INFO

Keywords:
Hybrid electric vehicle
Energy management system
Artificial intelligence
LSTM deep learning

ABSTRACT

The high potential of Artificial Intelligence (AI) techniques for effectively solving complex parameterization tasks also makes them extremely attractive for the design of the Energy Management Systems (EMS) of Hybrid Electric Vehicles (HEVs). In this framework, this paper aims to design an EMS through the exploitation of deep learning techniques, which allow high non-linear relationships among the data characterizing the problem to be described. In particular, the deep learning model was designed employing two different Recurrent Neural Networks (RNNs). First, a previously developed digital twin of a state-of-the-art plug-in HEV was used to generate a wide portfolio of Real Driving Emissions (RDE) compliant vehicle missions and traffic scenarios. Then, the AI models were trained off-line to achieve CO_2 emissions minimization providing the optimal solutions given by a global optimization control algorithm, namely Dynamic Programming (DP). The proposed methodology has been tested on a virtual test rig and it has been proven capable of achieving significant improvements in terms of fuel economy for both charge-sustaining and charge-depleting strategies, with reductions of about 4% and 5% respectively if compared to the baseline Rule-Based (RB) strategy.

Definitions/Abbreviations

AI	artificial intelligence
AT	automatic transmission
AWD	all-wheel drive
BEV	battery electric vehicle
BMEP	brake mean effective pressure
BMS	battery management system
BSFC	brake specific fuel consumption
CD	charge depleting
CO_2	carbon dioxide emissions
CS	charge sustaining
DC	direct current
DL	deep learning
DNN	deep neural network
DP	dynamic programming
EC	european commission
ECMS	equivalent consumption minimization strategy
ECU	electronic control unit
EM	electric machine
EMS	energy management system
EU	european union
GHG	greenhouse gas
GPS	global positioning system
HCU	hybrid control unit
HEV	hybrid electric vehicle
HV	high voltage
ICE	internal combustion engine
LB	learning-based
LiNMC	Li-ion nickel-manganese-cobalt-oxide
LSTM	long short-term memory
LV	low voltage
NEDC	new european driving cycle
NMC	nickel–manganese-cobalt
NN	neural network
OB	optimization-based
PEMS	portable emissions measurement system
PHEV	plug-in hybrid electric vehicle
PID	proportional–integral–derivative
RB	rule-based
RDE	real driving emission
RL	reinforcement learning

* Corresponding author.
E-mail address: federico.millo@polito.it (F. Millo).

https://doi.org/10.1016/j.treng.2022.100156
Received 25 August 2022; Received in revised form 27 November 2022; Accepted 9 December 2022
Available online 10 December 2022
2666-691X/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

RMSE	root mean square error
RNN	recurrent neural network
SoC	state of charge
TA	type approval
TTW	tank-to-wheel
V2X	vehicle-to-everything
WLTC	worldwide harmonized light-duty cycle

1. Introduction

The mitigation of global warming and the reduction of carbon footprints are among the most crucial tasks to be tackled in the near future in order to limit the increase of the Earth's temperature to about 1.5 °C above the pre-industrial levels [1]. Focusing on road vehicles, which account for nearly three-quarters of CO_2 emissions coming from the transport sector [2], in June 2022, the European Parliament (EP) approved the *Fit for 55* package, which will tighten the CO_2 emissions targets set for 2030 from -37.5% to -55% and from -31% to -50%, for passenger cars and light-duty vehicles respectively, in comparison with the 1990 levels. Furthermore, the package aims to achieve carbon neutrality by 2050, banning Internal Combustion Engine (ICE) powered vehicles by 2035 [3].

In this framework, powertrain electrification seems to be one of the most promising technologies for achieving these challenging targets, since it can ensure lower local CO_2 emissions than a traditional ICE powertrain, cutting them to zero for a Battery Electric Vehicle (BEV) with a Tank-To-Wheel (TTW) approach. Furthermore, the growth of the electrified powertrains market is being strongly encouraged by the European Commission (EC) through incentives and benefits for the purchase of new electrified cars [4]. These policies led to a remarkable growth in the sales of electrified vehicles in 2020, leading to a 21% decrease of the average type approval CO_2 emissions of new vehicles in Europe [5].

Nevertheless, looking at the full spectrum of electrification, BEVs still have strong limitations related to their short range, the long recharging time of the battery, the lack of charging infrastructures, and their high costs [6]. Furthermore, to achieve a realistic net-zero carbon emissions propulsion system, powertrain electrification must be linked to grid decarbonization, adopting renewable sources for electricity production [7]. Therefore, in a short-term scenario, Hybrid Electric Vehicles (HEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) may represent a stepping stone toward higher electrification levels of the powertrain, since they can combine the benefits of both electric and conventional powertrains [8].

As regards HEVs, the introduction of an additional power source increases the degrees of freedom of the powertrain control system. The overall performance of the vehicle in terms of fuel consumption highly depends on how the actuators are exploited. Therefore, an additional layer of control, called the Energy Management System (EMS), has to be added to the vehicle control hierarchy in order to optimize the power split between the power actuators installed on board [9]. Several optimization procedures have been proposed in the literature and, as described in depth in [10], they can be broadly classified into three main categories: Rule-Based (RB), Optimization-Based (OB), and Learning-based (LB). The last category is based on Artificial Intelligence (AI) techniques, which are extremely promising thanks to their ability to find hidden and complex relationships between the data that characterize the modeled problem. Therefore, AI models are particularly suitable for applications that involve high non-linear data, whose rule-based description could be quite complex and not very accurate. These features have led to the exploitation of AI techniques in technological sectors quite different from data science. The automotive field has started to adopt AI models quite recently, thanks to the increased computational power installed on-board and to the combined use of vehicle connectivity and cloud computing, which can provide much more information for the optimization of the energy flows. Among LB techniques, Deep Learning (DL) and Reinforcement Learning (RL) [11] are the most promising ones, since they can obtain quasi-optimal results [10]. The former is a complex class of neural networks (a subset of supervised learning techniques) that must be trained off-line on a target solution [12,13].

Reinforcement learning techniques, on the other hand, can provide significant benefits for planning and optimization tasks, thanks to their ability to self-learn from errors by directly interacting with the external environment (i.e., the vehicle model) through a trial-and-error procedure. In addition, the RL framework can combine the advantages of its learning method with the potentiality of deep learning techniques to design the controller of the model. However, it still presents numerous difficulties that many researchers are trying to address, such as a tricky and time-consuming training process due to the lack of a target solution [14,15]. In the literature, hybrid electric vehicle control process is usually modeled as a Markov Decision Process (MDP) [15,17], so that the actual state is sufficient to completely describe the environment. In [16] several works are described where RL techniques are applied to electrified vehicle power management. Among the different RL algorithms, Q-Learning and Deep Q-Learning (DQN) are the most common ones. In [15] a Q-Learning method is used to optimize the electric machine torque in a mild-HEV, without using the battery State of Charge (SoC) as an environment state. However, this methodology relies only on the WLTC driving cycle. In [17] an improved TD3 agent controls the power split of a parallel HEV. The performance of the training phase is enhanced thanks to an off-line computed optimal experience buffer, that is employed in addition to the online learned experience. Concerning a Machine Learning (ML) framework, in [13] a combined DP-ML-based approach is developed to control the power flow and the gear ratio of an HEV. This methodology is also used for training different control logics, to be more effective in various driving scenarios and drivers' behaviors. Finally, in [18] an innovative Adaptation algorithm for an Equivalent Consumption Minimization Strategy exploiting V2X connectivity (A-V2X-ECMS) is developed where driving pattern identification is employed to adapt the equivalence factor of the ECMS according to the future driving conditions.

In this framework, this paper proposes an innovative deep learning-based EMS able to efficiently handle the energy management of a PHEV, achieving sub-optimal results both in Charge-Sustaining (CS) and Charge-Depleting (CD) operating conditions. The supervised learning model has been trained off-line by providing the optimal solutions given by the Dynamic Programming (DP), a global optimization strategy capable of ensuring optimal performance [19], over a wide range of driving and traffic scenarios. Moreover, by considering sequential information about the history of the vehicle, it is possible to identify some crucial information to decide the optimal power split management. For this reason, the Authors modeled the hybrid electric vehicle control process as a Partially Observable Markov decision Process (POMDP) by designing the deep learning model through a subset of deep neural networks, called Recurrent Neural Networks (RNNs) since they can deal with the temporal information of input data. RNN architecture can update the current state based on the feedback of both the current input data and the past states (the so-called "short-term memory) [20], allowing relevant past information (i.e., driving pattern, past SoC trend), to be provided to the network. In particular, the so-called Long Short-Term Memory (LSTM) layer has been adopted because, for traditional RNN dealing with a large time gap between the relevant input data, the error signal "flowing backwards in time" tend to either blow up or vanish ("vanishing gradient problem" [21]). To overcome these back-flow error problems, and correctly handle the so-called "long-term dependency", in 1997 Hochreiter and Schmidhuber proposed the above-mentioned LSTM layer [22–24]. The potential of the proposed methodology in real-world conditions was assessed through numerical simulation on a virtual test rig of a PHEV, built in previous work [25]: the vehicle was modeled in GT-SUITE® and coupled to an EMS developed in Simulink®.

This paper is organized as follows: after a brief introduction of the case study (Section 2), the virtual test rig developed and the methodology proposed are described in Sections 3 and 4 respectively. Afterward, the performance of the developed EMS is assessed on two different driving scenarios. The first assumes the vehicle to be operated with a fully discharged battery in CS mode, while the second considers the vehicle working in CD mode starting from the maximum SoC (Section 5). Finally, the paper summarizes the main findings of the research activity and its potential future developments.

2. Case study

2.1. Vehicle specifications

The vehicle under investigation is a state-of-the-art diesel PHEV available in the European market. It features a P2 architecture, and the powertrain layout is schematically shown in Fig. 2.1. A Euro 6d-temp 1950 cc diesel engine, fitted in the front of the vehicle in a longitudinal position, is integrated and connected, through an auxiliary clutch (K0), to an Electric Machine (EM) of Permanent Magnet (PM) synchronous type. Both the ICE and the EM are connected, through a Torque Converter (TC) and a 9-speed Automatic Transmission (AT), to the rear axle. The EM is powered by a 13.5 kWh Li-Ion Nickel-Manganese-Cobalt-oxide (Li-NMC) HV battery. A DC/DC converter allows the HV battery to feed the 12 V battery and all the Low Voltage (LV) loads (i.e., the 12V starter and the electrical oil pump for gearbox lubrication).

Table 2.1 summarizes the main vehicle and powertrain characteristics.

3. Virtual test rig

In order to assess the potential of advanced energy management strategies based on artificial intelligence algorithms, a virtual test rig of the investigated vehicle developed in [25] was exploited. It was built in the GT-SUITE® [27] software environment and validated against experimental results. In this context, with the aim of estimating vehicle fuel consumption, a quasi-static approach was adopted [28]: a virtual vehicle driver - i.e., a Proportional–Integral–Derivative (PID) controller - compares the actual vehicle speed to a target one and generates a power demand profile to follow the target speed. The code computes the actual vehicle speed by solving the longitudinal vehicle dynamics, while fuel consumption is computed based on steady-state performance maps. The vehicle and powertrain parameters are shown in

Table 2.1. The parameters along with the performance maps were derived from [26], where a set of dedicated tests was performed to estimate the relevant powertrain data required for fuel consumption-oriented modeling of the vehicle.

As far as the EMS is concerned, the Rule-Based (RB) control strategy was derived from [25], where a reverse engineering investigation was

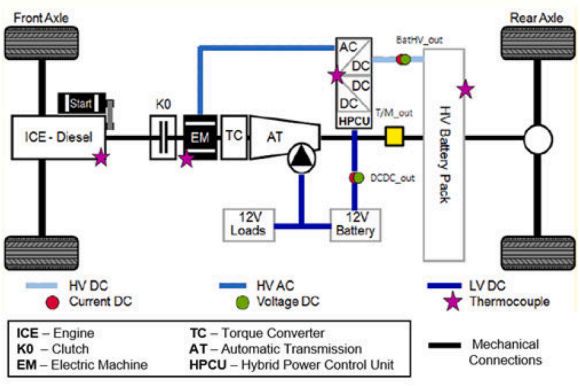

Fig. 2.1. Powertrain layout with instrumentation details [26].

Table 2.1
Vehicle and powertrain main specifications.

Vehicle			
Curb Weight	2060 kg		
Power	14.9 kW @ 100 km/h		
Configuration	Rear Wheel Drive		
Transmission			
Type	9-AT w/ Torque Converter		
Speed Ratios	I 5.36	IV 1.64	VII 0.87
	II 3.25	V 1.22	VIII 0.72
	III 2.26	VI 1.00	IX 0.61
Reverse	-4.93	Final Drive	2.65
Engine			
Engine Type	In-line 4 cylinders Turbo Diesel		
Displacement	1950 cm^3		
Max Power/Max Torque	143 kW @ 3800 rpm / 400 Nm @ 1600-2800 rpm		
Compression Ratio	15.5:1		
Electric Machine			
Type	PM Synchronous Motor		
Max Power/ Max Torque	90 kW @ 2000 rpm / 440 Nm @ 1750 rpm		
Max Speed	6000 rpm		
High Voltage Battery			
Type	Li-NMC		
Rated Voltage	365 V		
Capacity	13.5 kWh / 37 Ah		
Cooling System	Water Cooled		

carried out to extract the strategy implemented in the EMS of the actual vehicle. The control logic was implemented in the Simulink environment and coupled with the GT-SUITE vehicle model. For the sake of example, Figs. 3.1 and 3.2 show the model validation performed on an RDE cycle. The simulation results (red line) are compared to the experimental measurements (black dashed line). The accurate prediction of the battery SoC trajectory - Fig. 3.1 (b) - and the good agreement between simulated and measured engine torque - Fig. 3.2 - prove both the robustness of the simulation platform and the accuracy of the implemented control strategy.

Hereinafter, the extracted control logic will be indicated as Rule-Based (RB) and will be used as a reference for the assessment of the deep learning model performance. The same modeling approach (i.e., implementation in Simulink, and coupling with the GT-SUITE model) was adopted for the deep learning-based control strategy. The dataset to train and validate the neural networks must be filled with the results coming from DP optimization on a wide range of driving cycles. Since the computational effort of this algorithm strongly depends on the

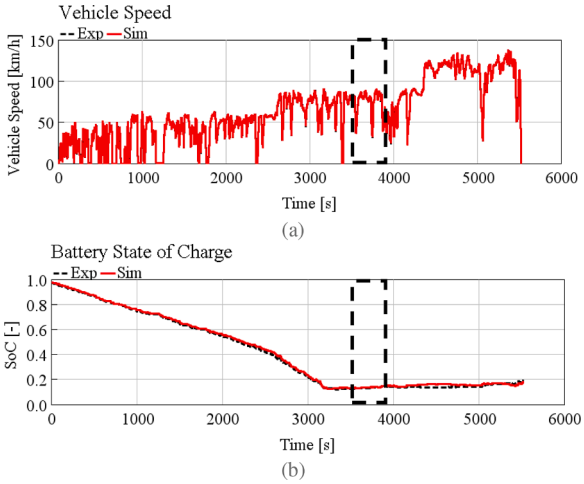

Fig. 3.1. Comparison between experimental data and numerical results.

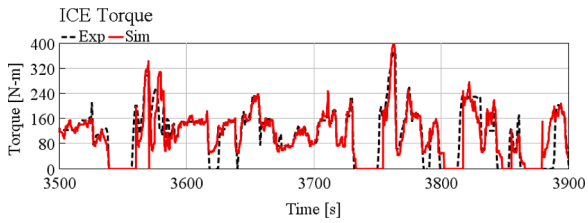

Fig. 3.2. Comparison between experimental and simulation ICE torque for a section of the cycle.

complexity of the interaction environment, a backward kinematic approach [28] was exploited to build a simplified vehicle model to carry out DP optimization.

3.1. Driving cycles database

The performance of deep learning algorithms depends on the amount of available data, and its increase does not reach an upper boundary, differently from traditional machine learning techniques (see Fig. 3.3). For this reason, all the driving cycles acquired during the experimental campaign, both Type-Approval (i.e., NEDC and WLTC) and Real Driving Emissions (RDE) cycles, were used to create the initial database. Then, thanks to the reliability of the validated test rig, additional simulations were carried out to make the database cover a wider spectrum of driving patterns.

In this process, new RDE-compliant [30] driving cycles were created through a simplified methodology, inspired by [31]. Each driving cycle performed during the experimental campaign was split into sub-cycles starting and finishing with a vehicle stop. Then, each sub-cycle was assigned to the Urban, Rural, or Highway categories, depending on its speed profile pattern. Finally, the extracted sub-cycles were randomly combined to generate new driving cycles satisfying the RDE-compliant conditions, summarized in Table 3.1. Fig. 3.4 depicts one driving cycle generated with the methodology described above. It should be noted that some amount of variability was introduced by adding random noise to the original speed profile.

The database created through the abovementioned methodology was then split into train, validation, and test datasets. The first one was used to train the algorithm in finding out the optimal parameters for the deep learning model, i.e., weights and biases of each node of the network (see Section 4.3 for more details). The validation dataset was used during the training phase for a preliminary model performance evaluation. Finally, the test dataset was used to evaluate the performance of the trained model. It should be noted that, in order to avoid the model overfitting on training data [32], it is of the utmost importance that the test dataset differs from the train and validation ones.

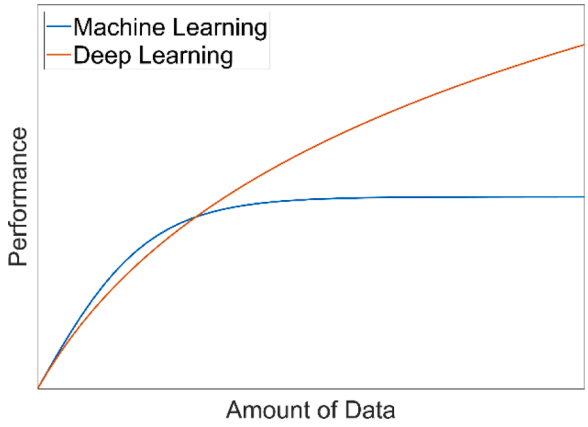

Fig. 3.3. General dependency of machine learning and deep learning models' performance on the amount of data [29].

4. Deep learning-based EMS

The RB control strategy, mentioned in Section 3, does not guarantee the optimal or a sub-optimal solution to the energy management problem. On the contrary, a global optimization strategy, such as DP, can provide the optimal solution to the control problem with the drawback of high computational efforts. However, it needs the a-priori knowledge of the whole driving cycle which cannot be available on an on-board application. In this work, the DP was used to benchmark the benefits that an optimization-based EMS can bring to the energy management of the case study. Moreover, it was provided as the target solution to be pursued by the artificial intelligence model during the training phase [33]. The AI cannot guarantee the optimal solution, but, if effectively trained, can provide a sub-optimal solution very close to the optimal one, with much lower computational effort compared with DP, achieving a good compromise in terms of on-board feasibility. In this paper, the authors decided to exploit deep learning techniques to extract control relationships among the provided data, thanks to their potential to achieve sub-optimal results while being theoretically feasible in a vehicle Electronic Control Unit (ECU).

4.1. Energy management problem

Since the design of the Energy Management System can be considered as an optimal control problem, its mathematical formulation is presented. Consider a generic dynamic system with a state equation expressed in Eq. (4.1) [19]:

$$\dot{x} = f(x, u, t) \tag{4.1}$$

Where $x \in R^n$ is the vector of the state variable, $u \in R^m$ is the vector of the control inputs, and t is the time. Eq. (4.2) defines the cost function (or performance index) that has to be minimized by choosing the law $u(t)$: $[t_0, t_f] \rightarrow R^m$. The right choice of $u(t)$ leads to the definition of the optimal control problem in the time interval $t \in [t_0, t_f]$

$$J = \Phi\left(x(t_f), t_f\right) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt \tag{4.2}$$

The boundaries conditions related to the optimal control problem definition are related to the terminal conditions:

$$\psi\left(x(t_f), t_f\right) = 0 \tag{4.3}$$

And the local constraints:

$$\begin{aligned} &G(x(t), t) \leq 0 \\ &x(t) \in X(t) \qquad \forall t \in \left[t_0,\ t_f\right] \\ &u(t) \in U(t) \end{aligned} \tag{4.4}$$

$L(\boldsymbol{x}(t), u(t), t) \in$ R is the instantaneous cost function, $\Phi(\boldsymbol{x}(t_f), t_f)$ is the terminal cost, $G(\boldsymbol{x}(t), t)$ is the set of constraints, $U(t)$ and $X(t)$ indicate, respectively, the set of admissible control and states values at time t. The general formulation of the optimal control problem expressed in Eq. (4.2) can be applied to an HEV control problem through Eq. (4.5) [34]. The only state variable is the battery state of charge, while several control variables can be selected.

$$J = \int_{t_0}^{t_f} \dot{m}_f(SoC(t), u(t), t)dt \tag{4.5}$$

Where $\dot{m}_f$ [g/s] is the instantaneous mass flow rate and SoC is the state of charge of the battery. Since the objective function J must be minimized under a set of both local and global constraints on the state and control variables, the optimal energy management problem of an

Table 3.1
RDE-compliant driving cycle characteristics.

Segment	Distance percentage	Minimum distance	Instantaneous speed	Average speed
Urban	29-44%	16 km	$v \leq 60\ km/h$	$15km/h < v \leq 40km/h$
Rural	23-43%	16 km	$60\ km/h < v \leq 90\ km/h$	$60km/h < v \leq 90km/h$
Motorway	23-43%	16 km	$v > 90\ km/h$	$v > 90km/h$

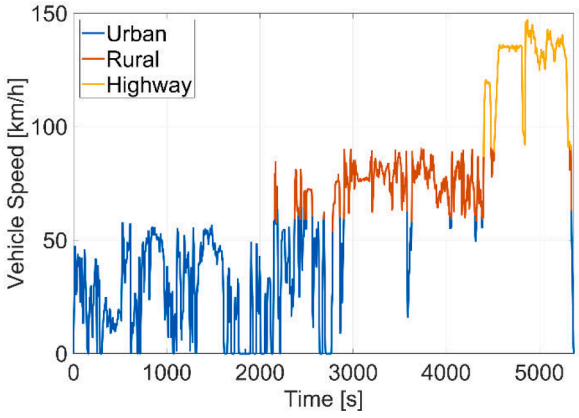

Fig. 3.4. Example of RDE-compliant generated driving cycle.

HEV is a constrained, finite-time optimal control problem: it means that the minimization of J is subject to constraints related to physical limitations of the actuators and the energy stored in the Rechargeable Energy Storage System (RESS). Moreover, the battery SoC must be always contained within prescribed limits. The optimal control problem can be addressed with several methods, such as through Dynamic Programming (DP) control algorithm [34].

4.2. Dynamic programming

DP [19] belongs to global optimization strategies. It is a numerical method for solving multistage decision-making problems. It can provide the optimal solution, but only in a simulation environment since it is non-casual. DP algorithm is based on Bellman's principle of optimality [9]. A DP optimization on a kinematic model of the PHEV (see Section 3) was performed by exploiting an open-source MATLAB code developed at ETH-Zurich [35,36] which solves discrete-time optimal control problems using Bellman's Dynamic Programming algorithm. The battery SoC is the only state variable, while the engine status (ON/OFF) u_{ICE} and the electric machine power P_{EM} are the control variables. The goal of the optimization is to minimize the fuel consumption of the vehicle. Both the state and control variables were discretized as described in Table 4.1.

The mathematical description of Bellman's principle of optimality applied to the HEV control problem can be formulated in Eq. (4.6):

$$SoC_{k+1} = f_k\left(SoC_k, u_{ICE,k}, P_{EM,k}\right) \tag{4.6}$$

$k = 0, 1, \ldots, N-1$, is the discretization time step, while the control and state variables are subject to the local constraints expressed in Eq. (4.4). The optimal control policy π^*, expressed in Eq. (4.7), must be defined in order to minimize the cost function, expressed in Eq. (4.8), from the initial state SoC_0 to the final state SoC_N.

Table 4.1
State and control variables grid description.

	SoC (State)	Engine Status (Control)	Electric Machine Power (Control)
Bottom Value	8%	0	-90000 W
Upper Value	100%	1	90000 W
Discretization Step	0.5%	1	200 W
Number of Elements	185	2	901

$$\pi^* = \left\{ \begin{matrix} u_{ICE,0^*} \\ P^*_{EM,0} \end{matrix}, \begin{matrix} u_{ICE,1^*} \\ P^*_{EM,1} \end{matrix}, \ldots, \begin{matrix} u_{ICE,N-1^*} \\ P^*_{EM,N-1} \end{matrix} \right\} \tag{4.7}$$

$$J_\pi(SoC_0) = m_{f,N}\left(SoC_N, u_{ICE,k}, P_{EM,k}\right) + \sum_{k=0}^{N-1} m_{f,k}\left(SoC_k, u_{ICE,k}, P_{EM,k}\right) \tag{4.8}$$

Consider now the cost function from the time step k_i to the time step k_N, optimizing the tail subproblem:

$$J_{\pi_i^*}(SoC_i) = m_{f,N}\left(SoC_N, u_{ICE,k}, P_{EM,k}\right) + \sum_{k=i}^{N-1} m_{f,k}\left(SoC_k, u_{ICE,k}, P_{EM,k}\right) \tag{4.9}$$

Exploiting Bellman's principle of optimality, it can be stated that *the tail policy π_i^* is optimal for the tail subproblem.* It is possible to determine the optimal sequence of control actions $\pi^* = \{\pi^*_{N-1}, \ldots, \pi^*_i, \ldots, \pi^*_1, \pi^*_0\}$ proceeding backward and choosing at each time step the path that minimizes the cost-to-go $J_\pi(SoC_k)$. The presence of a backward phase in this control algorithm makes it unfeasible for an online application because, for the optimization of the energy management of HEVs, the complete a-priori knowledge of the driving cycle is needed.

4.3. Long Short-Term Memory (LSTM) neural network

LSTM networks [22] are a particular class of Recurrent Neural Networks (RNNs) [20], which can deal with temporal sequences, using feedback connection to store information coming from recent input events. LSTM can overcome the vanishing or exploding gradient problem [21] that affects RNNs when the error signal of information with a large time lag must be backpropagated in time. This is possible thanks to the presence of four gates, that act as traditional neurons, each of them with a specific task in the network behavior, enabling the network to handle both the short-term dependency, available also with RNNs, and the long-term dependency. Focusing on the LSTM layer mathematical description, Fig. 4.1 shows a schematic representation of the standard LSTM cell applied to the EMS framework. At time step t, the block uses the previous state of the network (c_{t-1}), the previous output (h_{t-1}), and the current inputs (x_t) to compute the output (h_t) and update the cell state (c_t), so that it contains information from previous time steps. The layer adds or removes information from the cell state through the gates: gate f is in charge of deciding what information must be kept or discarded, while gates g and i update the state with the information coming from the actual input x_t. Gate o collects the actual input and previous time step output information which must be later combined with the state information to compute the output. Based on the connections shown in Fig. 4.1, the LSTM cell can be mathematically expressed as in Eq. (4.10) [37]:

$$\begin{aligned} i_t &= \sigma_g\left(W_i x_t + R_i h_{t-1} + b_i\right) \\ f_t &= \sigma_g\left(W_f x_t + R_f h_{t-1} + b_f\right) \\ g_t &= \sigma_c\left(W_g x_t + R_g h_{t-1} + b_g\right) \\ o_t &= \sigma_g\left(W_o x_t + R_o h_{t-1} + b_o\right) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\ h_t &= o_t \cdot \sigma_c(c_t) \end{aligned} \tag{4.10}$$

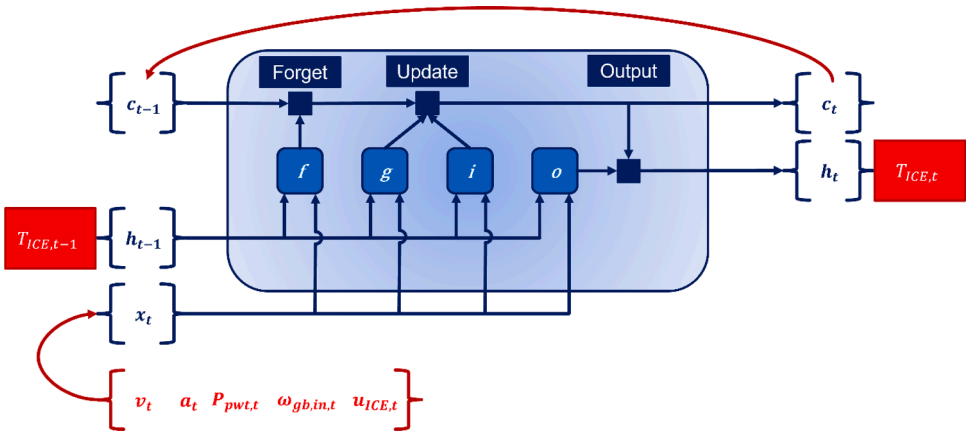

Fig. 4.1. LSTM layer layout for CS operations: the network inputs are vehicle speed, vehicle acceleration, powertrain power demand, engine rotational speed, and engine state; the network output is engine torque.

W and R are the input weights and recurrent weights vectors respectively for the relative gate, while b are the bias values. σ_g and σ_c are the activation functions whose main purpose is to introduce some non-linearity into the model. Their standard definition is expressed in Eq. (4.11) [37]:

$$\sigma_g(x) = \frac{1}{1+e^{-x}} \qquad \sigma_c(x) = \tanh x \tag{4.11}$$

Concerning the HEV energy management design, Fig. 4.1 displays the LSTM layer application to predict engine torque. The network inputs are vehicle speed, vehicle acceleration, powertrain power demand, engine rotational speed, and engine state.

4.4. LSTM-based EMS

The proposed methodology exploits a deep neural network model, based on LSTM layers architecture [22], whose flowchart is represented in Fig. 4.2. Firstly, the deep learning control algorithm was trained with the target solution provided by the DP [19], which was applied to all the available driving cycles. During the training phase, a backward kinematic model of the vehicle was used for running the DP optimization. Finally, the pre-trained deep learning model was tested online, by integrating the controller into the GT-SUITE vehicle model. The RDE-compliant driving cycles were split into training (57 cycles), validation (28 cycles), and testing (6 cycles) sets. It should be noted that the driving cycles used for the testing phase were not employed during training and validation.

Concerning the EMS architecture, two different networks were trained for CS and CD operations. This distinction is necessary since the management of the battery significantly changes between the two conditions, and the same controller cannot provide sub-optimal results on both of them. Both models have the same architecture, as shown in Fig. 4.3. A first network (classification model) decides the engine state and a second network (regression model) controls the engine torque. Higher performance can be obtained with the double neural network architecture if compared to the single one since the engine state represents a crucial input for the second neural network: thus, the latter can output engine torque values closer to the target ones when the ICE is turned on.

The training effectiveness strongly depends on the quality of the training data. Concerning an HEV application, the driving cycle must be described by features strongly related to the output prediction (e.g., vehicle speed profile, vehicle power demand profile). The extracted features are summarized in Tables 4.2 and 4.3, for CS and CD conditions respectively. The networks take as inputs some features obtained from the driving cycle as evident from Fig. 4.3.

As evident from Table 4.3, both information about the distance to the final destination and the time needed to reach it are used as inputs to the first neural network in the CD network. A good estimation of both variables can be easily obtained from the navigation system, standard equipment in many vehicles on the market. Moreover, the increasing connectivity level of last-generation vehicles can further enhance the

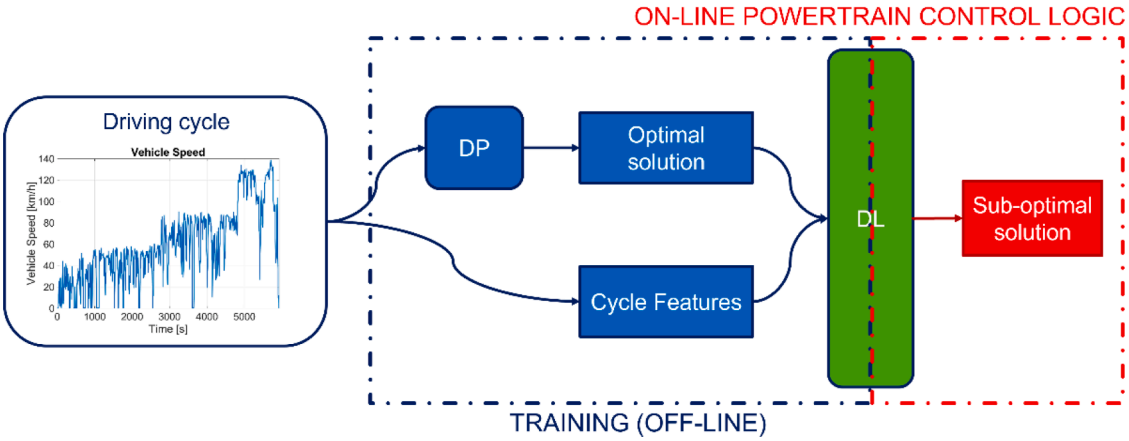

Fig. 4.2. Flowchart of the proposed methodology: a supervised learning algorithm is trained off-line where the target is given by the DP; the pre-trained deep learning model is then tested online.

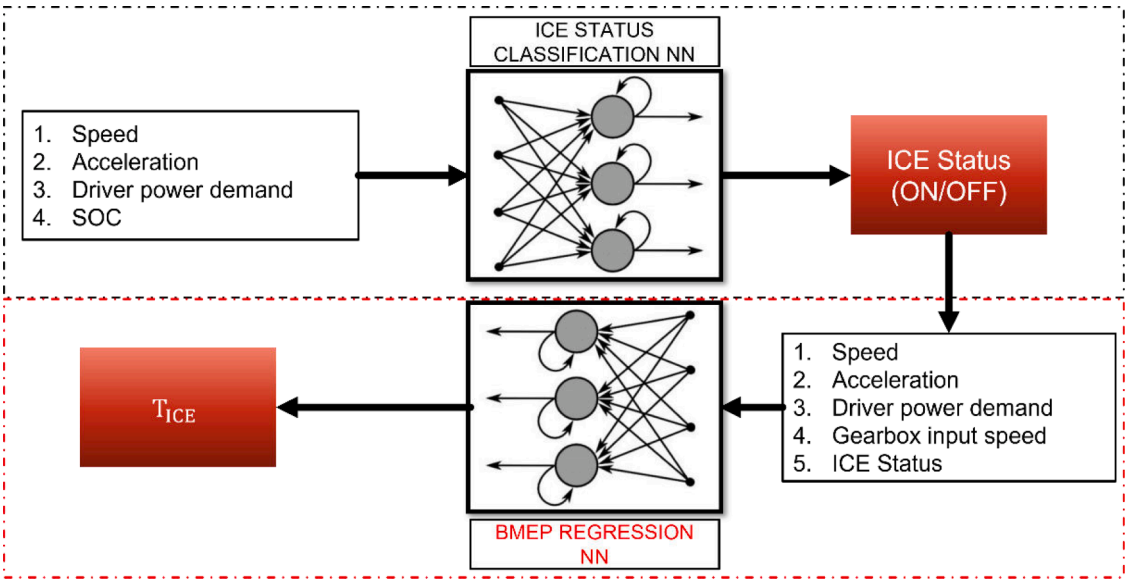

Fig. 4.3. EMS based on two LSTM NN architecture representation.

Table 4.2
Input features for CS networks.

CS ICE status NN		CS ICE Torque NN	
Vehicle speed	v_i	Vehicle speed	v_i
Vehicle acceleration	a_i	Vehicle acceleration	a_i
Driver power demand	$P_{pwt,i}$	Driver power demand	$P_{pwt,i}$
State of charge	SoC_i	Input gearbox speed	$\omega_{gb,i}$
		Engine state	$u_{ICE,i}$

Table 4.3
Input features for CD networks.

CD ICE status NN		CD ICE Torque NN	
Time percentage	$\frac{t_i}{t_f}$	Vehicle speed	v_i
Distance	$\int_{t_0}^{t_i} v(t)dt$	Vehicle acceleration	a_i
Distance percentage	$\frac{\int_{t_0}^{t_i} v(t)dt}{\int_{t_0}^{t_f} v(t)dt}$	Driver power demand	$P_{pwt,i}$
Vehicle speed	v_i	Input gearbox speed	$\omega_{gb,i}$
Vehicle acceleration	a_i	State of charge	SoC_i
Driver power demand	$P_{pwt,i}$	Engine state	$u_{ICE,i}$
State of charge	SOC_i		

accuracy of time prediction. In this work, a perfect knowledge of the future speed profile is assumed (e.g., the uncertainties related to the estimation of traffic conditions are neglected), leading to a perfect estimation of the total travel time. The Experiment Manager tool integrated into the MATLAB environment [38] was employed to choose, through a Bayesian optimization procedure [39], the network hyperparameters: i.e., all the parameters that are not optimized through the training algorithm, but should be manually set by the user, e.g., learning rate, network deep, regularization factor, etc. Table 4.4 displays the hyperparameters of the LSTM networks used in the following section.

5. Results

In Table 5.1 the results of the training and validation phases in terms of Performance Indices (PIs) are reported. The engine state and engine torque networks were evaluated through the accuracy [40] and Root Mean Square Error (RMSE) indices [41], respectively. The choice of the PI index depends on the type of the network, i.e., accuracy and RMSE for classification and regression networks, respectively. It should be noted that training and validation values are pretty comparable: this proves that the networks do not overfit the training data.

Table 4.4
Values of the hyperparameters for all the LSTM networks. They were optimized by employing the Experiment Manager tool.

Hyperparameters	CS ICE State	CS ICE Torque	CD ICE State	CD ICE Torque
Number of LSTM - Hidden Layers	1	3	2	5
nNodes1	48	54	42	43
nNodes2	/	50	30	75
nNodes3	/	24	/	20
nNodes4	/	/	/	33
nNodes5	/	/	/	19
Dropout	0.015	0.2	0.6	0.5
Learn Rate	0.025	0.003	0.03	0.009
Regularization Factor	0.01	$1.03 * 10^{-7}$	0.17	$1 * 10^{-4}$

Table 5.1
Performance Indices (PI) for training and validation phases.

Hyperparameters	Index	Train PI	Validation PI
CS Engine State	Accuracy	98%	96%
CS ICE Torque	RMSE	0.014	0.021
CD Engine State	Accuracy	98%	95%
CD ICE Torque	RMSE	0.012	0.017

In this section, for the sake of brevity, the results are shown only on one test cycle for both CS and CD strategies. The results of the proposed EMS were compared with two different control strategies: the Rule-Based (RB) strategy, extracted from the actual vehicle as described in Section 3, and an Equivalent Consumption Minimization Strategy (ECMS). The ECMS is a local optimization algorithm that can provide a sub-optimal solution while being online implementable in a vehicle ECU [34]. It should be pointed out that the DP results can be considered as an optimum benchmark only in terms of optimal battery management, but not in terms of fuel consumption. The fuel consumption coming from the two models cannot be directly compared, due to the discrepancies in terms of modeling approaches. A simplified model, i.e., backward

kinematic, is used to perform the DP optimization (see Section 3), entailing lower fuel consumption. The SoC trend provided by the DP, instead, is used as a benchmark for the battery management strategy and can be directly compared with the one obtained by the proposed EMS. The driving cycle used to assess the EMS performance is an RDE cycle, conducted on the public roads in the surroundings of the Italian city of Turin, during the experimental campaign described in [26]. Table 5.2 displays the most relevant features of the test cycle, while Fig. 5.1 plots the vehicle speed as a function of time.

5.1. Charge sustaining

Fig. 5.2 shows the SoC trend of the proposed EMS (LSTM) compared to RB, DP, and ECMS logic. It should be noted that both LSTM and ECMS better handle the battery energy management if compared to the RB strategy since the final SoC value is closer to the target one. This leads to lower fuel consumption, as shown in Fig. 5.3, where the trade-off between CO_2 emission and final SoC value is shown for all three considered cases. The proposed control strategy leads to a reduction of CO_2 emissions of about 3.5% and 2.2% if compared to RB and ECMS, respectively. Focusing on Fig. 5.2, it is eye-catching that the SoC trend of the LSTM logic is the one that better mimics the DP one. For this reason, the control law defined by the LSTM is able to reach better fuel economy if compared to the ECMS one (see Fig. 5.3), although the latter presents a final SoC level closer to the target one. Thus, despite the worse exploitation of the energy stored in the battery (higher final SoC value), the LSTM can reduce fuel consumption if compared to the ECMS control strategy.

Fig. 5.4 shows the differences between LSTM, ECMS, RB, and DP in terms of engine operation on its Brake Specific Fuel Consumption (BSFC) map. The engine operating points are represented by circle markers whose size is proportional to the time spent by the engine in that efficiency region. It should be noted that the LSTM - Fig. 5.4 (a) - has a more uniform distribution on the BSFC map than the ECMS - Fig. 5.4 (b). It means that the engine can work at higher load conditions, thus higher efficiency, entailing a better fuel economy. Furthermore, the distribution of the operating points for the LSMT is pretty similar to DP one - Fig. 5.4 (d) - which represents the best solution. It is important to underline that the engine operating points are not affected by the different modeling approaches between LSTM and DP (see Section 3) because it only involves ICE torque management.

5.2. Charge depleting

According to the literature [42,43], the fuel consumption of a fully charged PHEV can be minimized by following an almost linear SoC discharge trend on a distance-based plot, while PHEVs available in the market usually follow a CD+CS logic. However, in order to obtain a linear discharge trend, it is necessary to have reliable information about the distance to the final destination and the time needed to reach it. For this reason, these variables were introduced as inputs to the neural networks (see Table 4.3). Moreover, all the electrical energy stored onboard should be depleted at the end of the cycle. However, since very low values of SoC could accelerate the aging of the battery, a minimum threshold value of 0.2 is imposed for this variable. In this section, the results of the proposed EMS based on LSTM are compared only with the baseline RB control strategy, since the ECMS controller was only developed for a CS application. From Fig. 5.5 it is eye-catching that the SoC trend obtained by the LSTM is completely different from the RB one. While the RB tends to follow a strategy typically employed on PHEV, i.e., a CD mode followed by a CS operation, the LSTM can better mimic the DP control law. Thus, the LSTM tends to have an almost linear SoC discharge if considered in the distance-based plot, reliable information about the distance to the final destination and the time needed to reach it is available. This behavior, as confirmed by Fig. 5.6, allows obtaining a

Table 5.2
Characteristic values of the RDE cycle used for testing the proposed EMS performance.

Time	Distance	Avg. Speed	Max Speed	Avg. Acc.	Max Acc.	Required Energy
[s]	[km]	[km/h]	[km/h]	[m/s^2]	[m/s^2]	[Wh/km]
5926	97	59	139	0.38	3.42	225

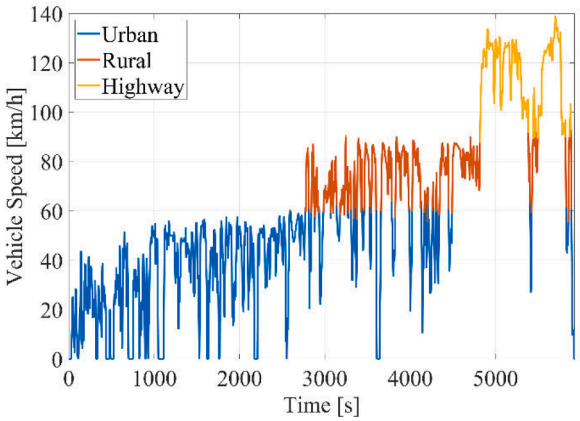

Fig. 5.1. Vehicle speed as a function of time of the RDE cycle used for testing the proposed EMS performance.

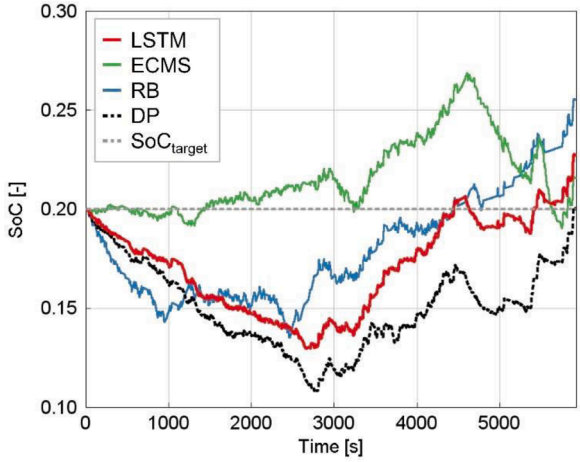

Fig. 5.2. SoC trends over the RDE cycle for the considered control strategies.

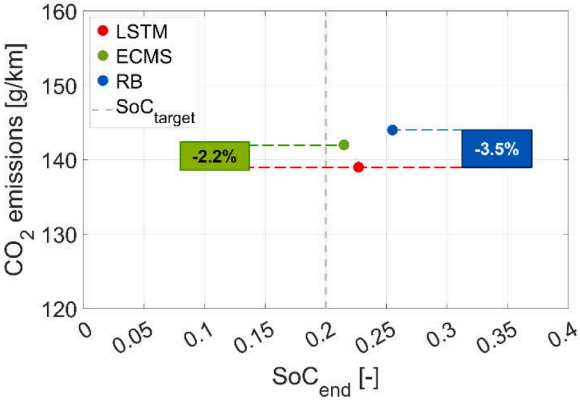

Fig. 5.3. Trade-off between CO_2 emissions and final SoC values of the compared control strategies in CS mode.

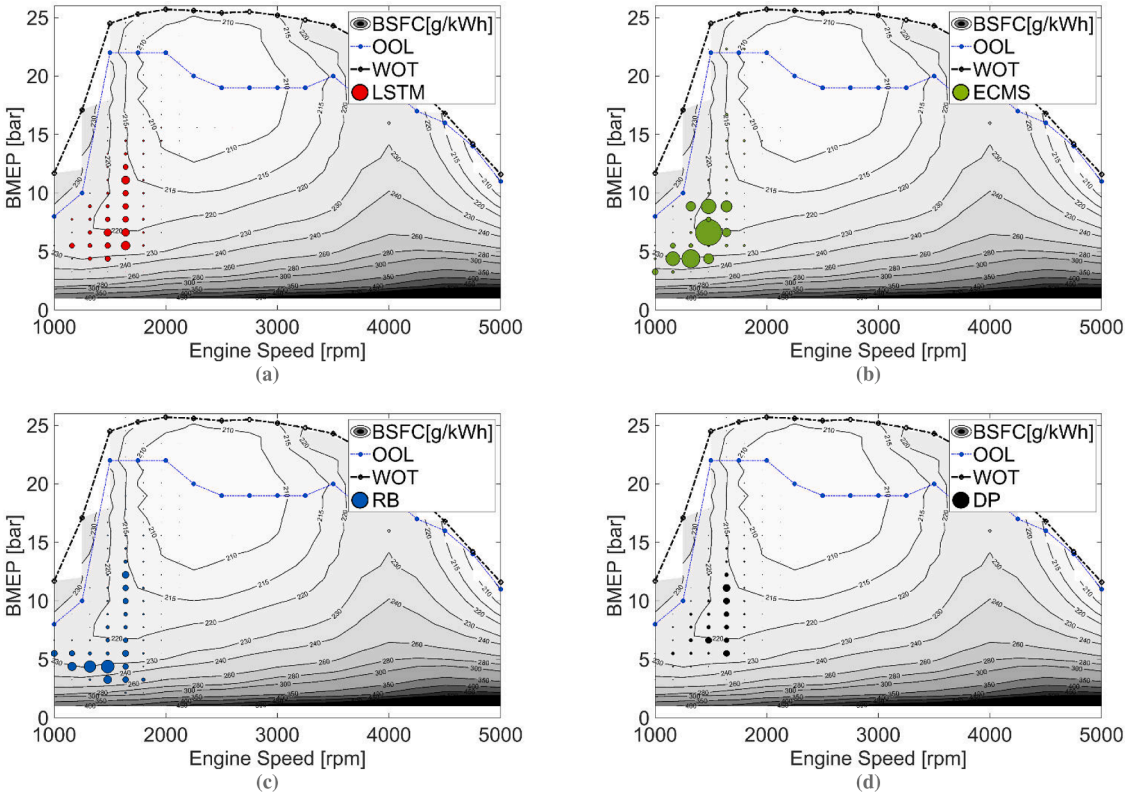

Fig. 5.4. Time distribution of the engine operating points of LSMT (a), ECMS (b), RB (c), and DP (d).

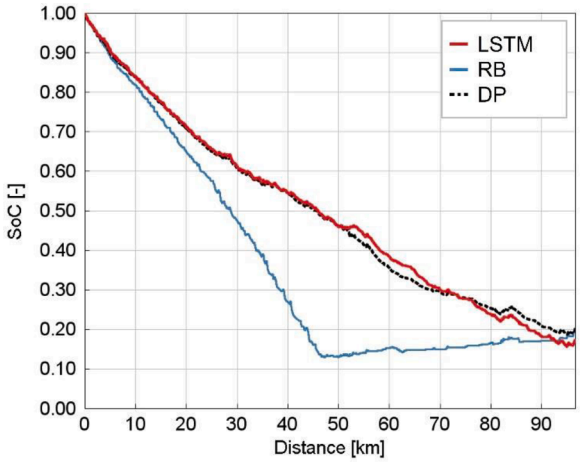

Fig. 5.5. SoC trend function of distance run by the vehicle of the two compared control strategies.

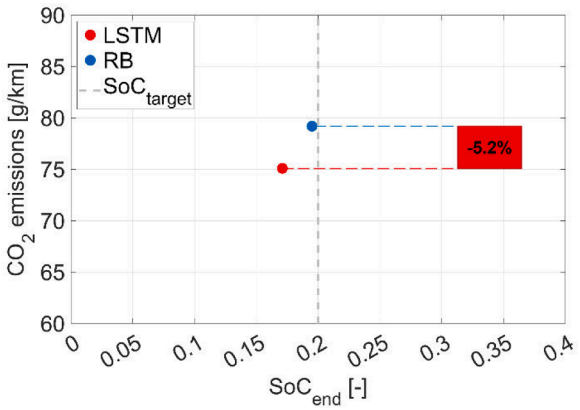

Fig. 5.6. Trade-off between CO_2 emission and final SoC value of the compared control strategies in CD mode.

CO_2 emissions reduction of about 5.2% if compared to the RB strategy.

The CO_2 reduction for the LSTM strategy can be related to a slightly lower final SoC and to more efficient management of both the engine operating points and the battery energy. The improvement of the engine efficiency can be seen in Fig. 5.7, where the engine operating points are plotted, by means of time distribution markers, on its Brake Specific Fuel Consumption (BSFC) map. It is eye-catching that the LSTM distribution - Fig. 5.7 (a) - features operating points at a higher load than the RB one - Fig. 5.7 (b) - since it mimics the time distribution of the DP logic - Fig. 5.7 (c).

5.3. *Sensitivity analysis*

Since, in real-life applications, a PHEV cannot always start a trip with a fully charged battery, its energy management system should be able to handle also different levels of initial SoC. In order to train the LSTM also on these conditions, additional DP optimizations were performed by varying the energy content of the battery at the beginning of the cycles. Fig. 5.8 shows the results of a sensitivity analysis performed on the trained deep learning model to assess the robustness of its performance in the case of initial SoC variability. 6 different scenarios were investigated with the initial SoC spanning from 0.5 to 1. The results of the LSTM - Fig. 5.8 (a) – are compared with the ones obtained by the RB logic - Fig. 5.8 (b) – and the DP - Fig. 5.8 (c). As expected, the DP results show that whichever is the initial value, the optimal strategy always follows a quasi-linear trend of SoC in a distance-based plot. On the other

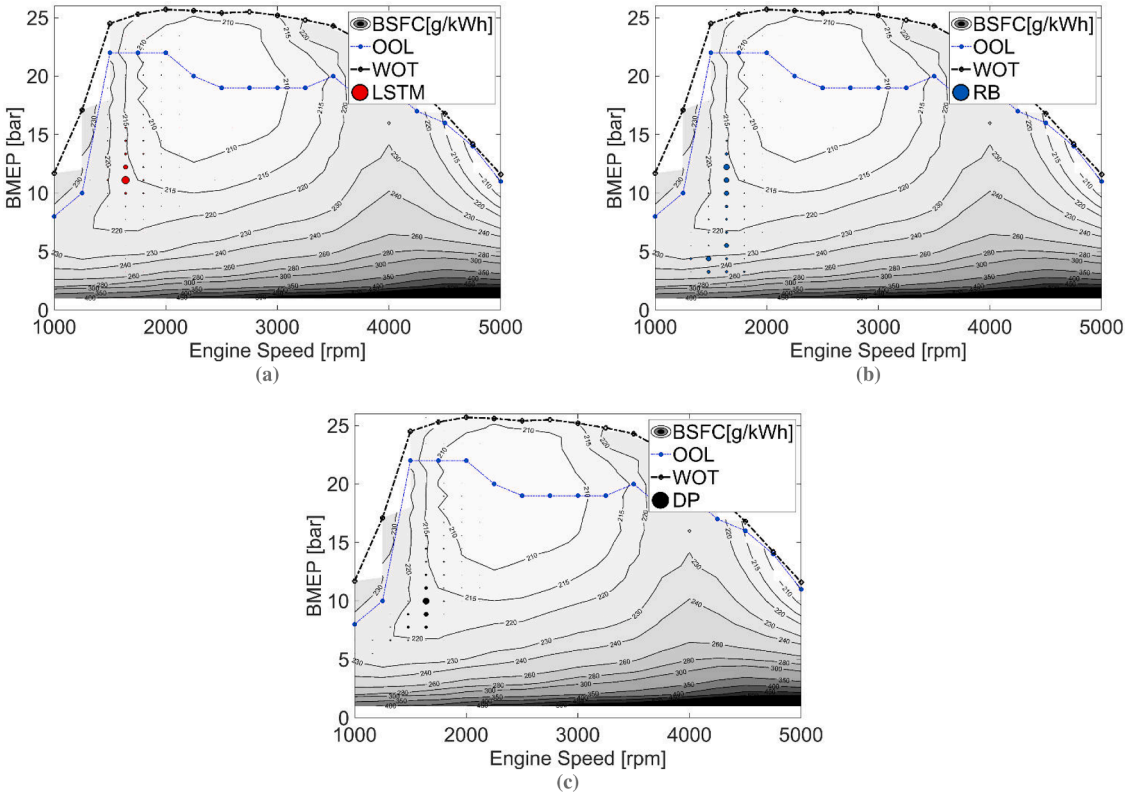

Fig. 5.7. Time distribution of the engine operating points of LSMT-EMS (a), ECMS-EMS (b), and DP (c).

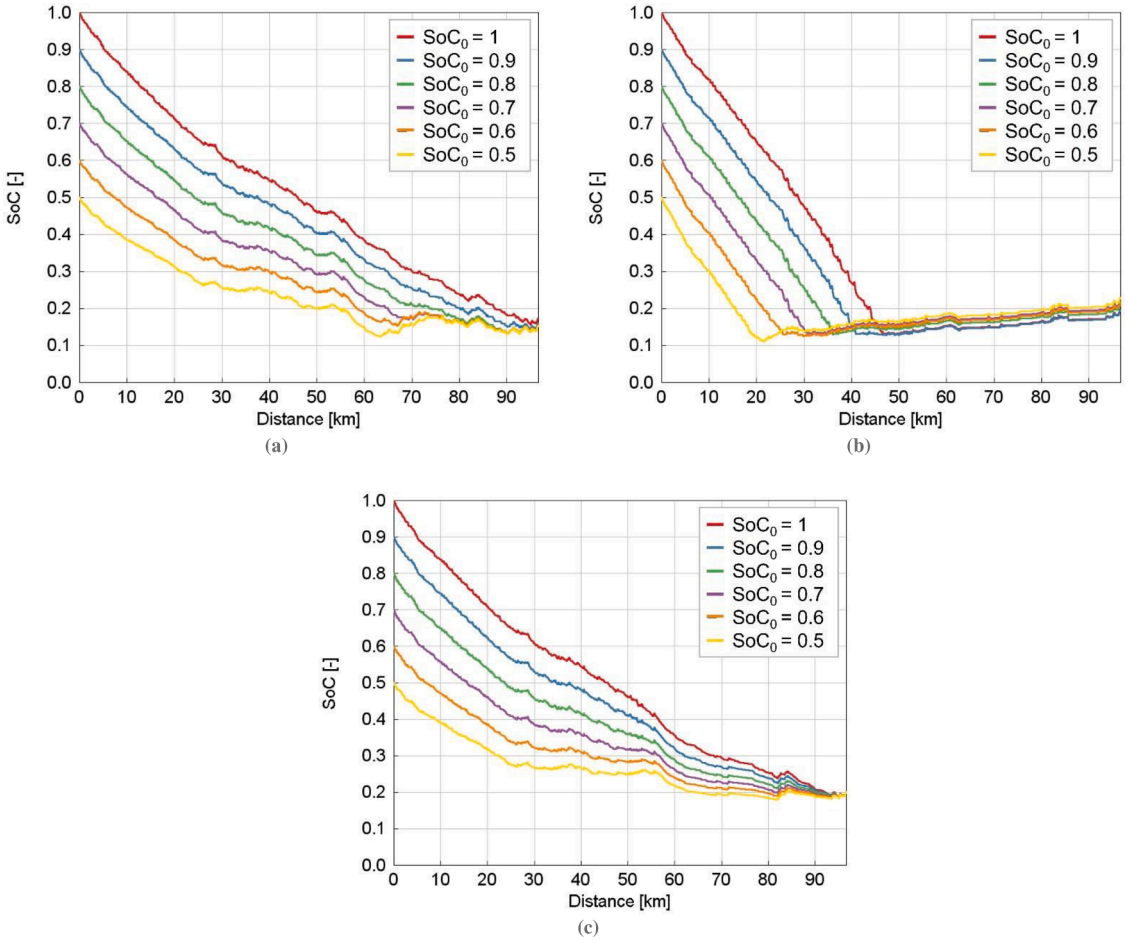

Fig. 5.8. SoC trend in CD mode starting from different initial SoC values spanning from 1 to 0.5 for LSTM (a) RB (b) and DP optimization (c).

Table 5.3
CO2 emissions of the LSTM-based EMS, along with CO2 emissions reduction from the RB logic, for 6 cases with the initial SoC spanning from 0.5 to 1.

SoC initial value	LSTM - CO_2 emissions [g/km]	CO_2 emissions reduction from RB strategy
100%	75	-5.2%
90%	82	-5.1%
80%	90	-5.6%
70%	98	-5.1%
60%	106	-3.7%
50%	113	-4.5%

hand, the RB strategy always follows the same behavior, i.e., a CD mode followed by a CS operation. It should be noted that starting from a lower initial SoC level only leads to a higher share of the CS mode, but the discharge rate of the battery is unchanged as well as the control law in the second part of the driving cycle. It is eye-catching that the LSTM strategy is able to efficiently handle the battery energy when starting with a not fully charged battery since it well mimics the DP behavior. All the different cases present an almost linear battery depletion, with a residual SoC, at the end of the cycle, that spans from 0.15 to 0.18. It should be noticed that only with an initial SoC of 0.5 - the yellow line in Fig. 5.8 (a) – there is a deviation of the SoC trend from the linear one: at about 63 km the minimum value is reached and the vehicle starts operating in CS mode.

Table 5.3 reports the CO_2 emissions of the LSTM-based EMS for all the analyzed cases. For each case, Table 5.3 also reports the CO_2 emissions reduction using the RB strategy as a benchmark. The LSTM-based EMS can improve the engine efficiency if compared to the RB strategy, enhancing the vehicle fuel economy and reducing its CO_2 emissions for all the analyzed cases. It should be noted that the CO_2 emissions reduction is lower for an initial SoC value of 60% or 50%. Since the phase in CS is longer for these two cases, the overall behavior of the LSTM is more similar to RB one, leading to smaller improvements in terms of fuel economy.

6. Conclusions

This paper developed a methodology to design a deep learning-based Energy Management System for a Plug-in Hybrid Electric Vehicle (PHEV), through the exploitation of deep Recurrent Neural Networks. The proposed architecture is based on two different neural networks that decide the vehicle operating mode (i.e., full electric or hybrid propulsion) and the power split in the case of hybrid propulsion. The networks were trained off-line using the optimal results provided by Dynamic Programming (DP). In this way, DP optimization significantly simplifies the training process in comparison with other techniques proposed in the literature, such as Reinforcement Learning based controllers, which are characterized by a long and tricky training process based on a trial-and-error procedure. The results obtained were then compared with two different powertrain control strategies, the first based on a Rule-Based (RB) strategy, extracted through an experimental campaign carried out on the real vehicle, and the second exploiting an Equivalent Consumption Minimization Strategy (ECMS) optimization. A significant improvement in terms of CO_2 emissions, and so in terms of fuel economy, was achieved for both charge-sustaining and charge-depleting strategies. For charge-sustaining, a CO_2 emissions reduction of about 4% if compared to the RB strategy was achieved, while for charge-depleting a CO_2 emissions reduction of about 5% was obtained. Finally, a sensitivity analysis on the initial energy content of the battery proved the robustness of the proposed EMS, which can obtain sub-optimal performance for all the values of initial State of Charge (SoC) considered. Future work will assess the potential benefits that the introduction of information deriving from Vehicle-to-Everything (V2X) connectivity could provide to the Neural Network-based EMS. A prediction on the future vehicle speed, and real-time information about traffic conditions and road infrastructure could further enhance the fuel economy of the proposed strategy.

CRediT authorship contribution statement

Federico Millo: Conceptualization, Supervision, Writing – review & editing, Project administration. **Luciano Rolando:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Luigi Tresca:** Conceptualization, Methodology, Software, Writing – original draft, Visualization. **Luca Pulvirenti:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The authors do not have permission to share data.

Acknowledgments

This research was financially supported by Regione Piemonte (Italy) under the Program L.R. 34/04–Programma d'intervento per le at t ività produt t ive 2011/2017–Asse 3 (Internazionalizzazione), Misura 3.1 "Contratto d'insediamento". Progetto: "Sviluppo di una nuova generazione di sistemi di propulsione di veicoli ibridi ed elettrici"/"Development of a new generation of hybrid and electric propulsion systems", -Soc. FEV Italia Srl e Politecnico di Torino.

References

[1] European Commission, "Paris agreement." https://ec.europa.eu/clima/eu-action/international-action-climate-change/climate-negotiations/paris-agreement_en (accessed Dec. 09, 2021).
[2] IEA, "Largest end-uses of energy by sector in selected IEA countries, 2018 – Charts – Data & Statistics - IEA." https://www.iea.org/data-and-statistics/charts/largest-end-uses-of-energy-by-sector-in-selected-iea-countries-2018 (accessed Feb. 07, 2022).
[3] ICCT, "Fit for 55: a review and evaluation of the European Commission proposal for amending the CO2 targets for new cars and vans." https://theicct.org/publications/fit-for-55-review-eu-sept21 (accessed Dec. 09, 2021).
[4] ACEA, "Electrifies vehicles: tax, benefits and purchase incentives", (2022).
[5] ICCT, "Europe's CO2 emission performance standards for new passenger cars: Lessons from 2020 and future prospects." https://theicct.org/publications/eu-ev-pv-co2-emission-performance-sept21 (accessed Dec. 09, 2021).
[6] J.A. Sanguesa, V. Torres-Sanz, P. Garrido, F.J. Martinez, J.M. Marquez-Barja, A review on electric vehicles: technologies and challenges, Smart Cities 4 (1) (2021) 372–404, https://doi.org/10.3390/smartcities4010022.
[7] F. Millo, L. Rolando, R. Fuso, F. Mallamo, Real CO2 emissions benefits and end user's operating costs of a plug-in Hybrid Electric Vehicle, Appl. Energy 114 (2014) 563–571, https://doi.org/10.1016/J.APENERGY.2013.09.014. Feb.
[8] L.H. Björnsson, S. Karlsson, Electrification of the two-car household: PHEV or BEV? Transp. Res. Part C Emerg. Technol. 85 (2017) 363–376, https://doi.org/10.1016/J.TRC.2017.09.021. Dec.
[9] A. Sciarretta, L. Guzzella, Control of hybrid electric vehicles, IEEE Control Syst. 27 (2) (2007) 60–70, https://doi.org/10.1109/MCS.2007.338280. Apr.
[10] D.D. Tran, M. Vafaeipour, M. El Baghdadi, R. Barrero, J. Van Mierlo, O. Hegazy, Thorough state-of-the-art analysis of electric and hybrid vehicle powertrains: Topologies and integrated energy management strategies, in: Renewable and Sustainable Energy Reviews, 119, Elsevier Ltd, 2020, https://doi.org/10.1016/j.rser.2019.109596. Mar. 01.
[11] L. Pack Kaelbling, M.L. Littman, A.W. Moore, S. Hall, Reinforcement learning: a survey, J. Artif. Intell. Res. 4 (1996) 237–285.
[12] Z. Chen, C.C. Mi, J. Xu, X. Gong, C. You, Energy management for a power-split plug-in hybrid electric vehicle based on dynamic programming and neural networks, IEEE Trans. Veh. Technol. 63 (4) (2014) 1567–1580, https://doi.org/10.1109/TVT.2013.2287102.
[13] C.K.D. Harold, S. Prakash, T. Hofman, Powertrain Control for Hybrid-Electric Vehicles Using Supervised Machine Learning, Vehicles 2 (2) (2020) 267–286, https://doi.org/10.3390/vehicles2020015. May.

[14] R. Liessner, C. Schroer, A. Dietermann, B. Bäker, Deep reinforcement learning for advanced energy management of hybrid electric vehicles, in: Proceedings of the 10th International Conference on Agents and Artificial Intelligence ICAART 2, 2018, pp. 61–72, https://doi.org/10.5220/0006573000610072.
[15] B. Xu, F. Malmir, D. Rathod, Z. Filipi, Real-time reinforcement learning optimized energy management for a 48V mild hybrid electric vehicle, SAE Tech. Pap. 2019-April (April) (2019), https://doi.org/10.4271/2019-01-1208. Apr.
[16] J.A. Sanguesa, V. Torres-Sanz, P. Garrido, F.J. Martinez, J.M. Marquez-Barja, A review on electric vehicles: technologies and challenges, Smart Cities 4 (1) (2021) 372–404, https://doi.org/10.3390/smartcities4010022. Mar. 01.
[17] J. Zhou, S. Xue, Y. Xue, Y. Liao, J. Liu, W. Zhao, A novel energy management strategy of hybrid electric vehicle via an improved TD3 deep reinforcement learning, Energy 224 (2021), https://doi.org/10.1016/j.energy.2021.120118.
[18] L. Pulvirenti, L. Rolando, and F. Millo, "Energy management system optimization based on V2X connectivity," pp. 13–17, 2021, doi: 10.46720/F2020-ADM-087.
[19] D.P. Bertsekas, Dynamic programming and optimal control, (2022).
[20] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: Lstm cells and network architectures, Neural Comput. 31 (7) (2019) 1235–1270, https://doi.org/10.1162/NECO_A_01199. Jul.
[21] A. Animesh, "The problem of vanishing gradients." https://towardsdatascience.com/the-problem-of-vanishing-gradients-68cea05e2625 (accessed Oct. 07, 2021).
[22] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, https://doi.org/10.1162/NECO.1997.9.8.1735. Nov.
[23] J.A. Pérez-Ortiz, F.A. Gers, D. Eck, J.U. Schmidhuber, Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets, Neural Netw. 16 (2) (2003) 241–250, https://doi.org/10.1016/S0893-6080(02)00219-8. Mar.
[24] Y. Chen, et al., 2-D regional short-term wind speed forecast based on CNN-LSTM deep learning model, Energy Convers. Manag. 244 (2021), 114451, https://doi.org/10.1016/J.ENCONMAN.2021.114451. Sep.
[25] F. Millo, L. Rolando, L. Pulvirenti, G. Di Pierro, A methodology for the reverse engineering of the energy management strategy of a plug-in hybrid electric vehicle for virtual test rig development, SAE Int. J. Electrified Veh. 11 (1) (2021), https://doi.org/10.4271/14-11-01-0009. Sep.
[26] G. Dipierro, E. Galvagno, G. Mari, F. Millo, M. Velardocchia, A. Perazzo, A reverse-engineering method for powertrain parameters characterization applied to a P2 plug-in hybrid electric vehicle with automatic transmission, SAE Tech. Pap. 2020 (June) (2020), https://doi.org/10.4271/2020-37-0021. JuneJun.
[27] GT-SUITE, "Vehicle driveline and HEV application manual." Gamma Technologies LLC.
[28] F. Millo, L. Rolando, M. Andreata, Numerical simulation for vehicle powertrain development, Numer. Anal. Theory Appl. (2011), https://doi.org/10.5772/24111. Sep.
[29] I.H. Sarker, Machine learning: algorithms, real-world applications and research directions, SN Comput. Sci. 2 (3) (2021) 1–21, https://doi.org/10.1007/S42979-021-00592-X, 2021 2:3Mar.
[30] "EUR-Lex - 32017R1151 - EN - EUR-Lex." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32017R1151 (accessed Nov. 17, 2022).
[31] J. Claßen et al., "RDE cycle generation – a statistical approach to cut down testing effort and provide a secure base to approve RDE legislation compliance," pp. 37–56, 2019, doi: 10.1007/978-3-658-26528-1_3.
[32] S. Salman and X. Liu, "Overfitting mechanism and avoidance in deep neural networks," Jan. 2019.
[33] D. Bianchi et al., "A rule-based strategy for a series/parallel hybrid electric vehicle: an approach based on dynamic programming," 2010.
[34] S. Onori, L. Serrao, G. Rizzoni, Hybrid Electric Vehicles: Energy Management Strategies, Springer Publishing Company, 2016, https://doi.org/10.1007/978-1-4471-6781-5 no. 9781447167792.
[35] O. Sundström, L. Guzzella, A generic dynamic programming MATLAB function, in: Proceedings of the IEEE International Conference on Control Applications, 2009, pp. 1625–1630, https://doi.org/10.1109/CCA.2009.5281131.
[36] O. Sundström, Optimal Control and Design of Hybrid-Electric Vehicles, ETH, Zurich, 2009.
[37] MathWorks, "Long short-term memory networks - MATLAB." https://it.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html (accessed Sep. 14, 2021).
[38] MathWorks, "Experiment manager - MATLAB." https://it.mathworks.com/help/deeplearning/ref/experimentmanager-app.html (accessed Sep. 28, 2021).
[39] P.I. Frazier, "A tutorial on Bayesian optimization," Jul. 2018.
[40] K. Nighania, "Various ways to evaluate a machine learning model's performance." https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15 (accessed Feb. 08, 2022).
[41] MathWorks, "Compute R-square, RMSE, correlation, and sample mean error of predicted and observed LGDs - MATLAB modelAccuracy - MathWorks Italia." https://it.mathworks.com/help/risk/regression.modelaccuracy_lgd.html#mw_cc5e06a6-9705-4bf9-a6a3-85c8e953fbd3 (accessed Feb. 08, 2022).
[42] V. Larsson, L. Johannesson, B. Egardt, A. Lassson, Benefit of route recognition in energy management of plug-in hybrid electric vehicles, in: Proceedings of the American Control Conference, 2012, pp. 1314–1320, https://doi.org/10.1109/ACC.2012.6314910.
[43] V. Larsson, L. Johannesson, B. Egardt, Impact of Trip Length Uncertainty on Optimal Discharging Strategies for PHEVs, in: Proceedings of the 6th IFAC Symposium Advances in Automotive Control, Munich; 12 July 2010 through 14 July 2010 43, 2010, pp. 55–60, https://doi.org/10.3182/20100712-3-DE-2013.00131.