Battery Aging-Aware Online Optimal Control: An Energy Management System for Hybrid Electric Vehicles Supported by a Bio-Inspired Velocity Prediction

GIAMMARCO VALENTI, EDOARDO PAGOT, LUCA DE PASCALI, AND FRANCESCO BIRAL
Department of Industrial Engineering, University of Trento, 38123 Trento, Italy
Corresponding author: Giammarco Valenti (giammarco.valenti@unitn.it)

ABSTRACT In this manuscript, we address the problem of online optimal control for torque splitting in hybrid electric vehicles that minimises fuel consumption and preserves battery life. We divide the problem into the prediction of the future velocity profile (i.e. driver intention estimation) and the online optimal control of the hybrid powertrain following a Model Predictive Control (MPC) scheme. The velocity prediction is based on a bio-inspired driver model, which is compared on various datasets with two alternative prediction algorithms adopted in the literature. The online optimal control problem addresses both the fuel consumption and the preservation of the battery life using an equivalent cost given the estimated speed profile (i.e. guaranteeing the desired performance). The battery degradation is evaluated by means of a state-of-the-art electrochemical model. Both the predictor and the Energy Management System (EMS) are evaluated in simulation using real driving data divided into 30 driving cycles from 10 drivers characterised by different driving styles. A comparison of the EMS performances is carried out on two different benchmarks based on an offline optimization, in one case on the entire dataset length and in the second on an ideal prediction using two different receding horizon lengths. The proposed online system, composed of the velocity prediction algorithm and the optimal control MPC scheme, shows comparable performances with the previous ideal benchmarks in terms of fuel consumption and battery life preservation. The simulations show that the online approach is able to significantly reduce the capacity loss of the battery, while preserving the fuel saving performances.

INDEX TERMS Hybrid electric vehicles (HEVs), online optimal control, velocity prediction algorithm, energy management system (EMS), battery electrochemical model.

I. INTRODUCTION
The impact of mobility on the emissions of greenhouse gases represents a critical reason to make transportation cleaner and more sustainable. Hybrid Electric Vehicles (HEVs) are a step forward to reduce road traffic pollution: one or more electric motors can act as a generator during the braking phase of driving, recovering kinetic energy that would be otherwise wasted as heat. Moreover, another point of strength is the capability of HEVs to reduce emissions via the redundancy of the power output. The electric motor and the internal combustion engine (ICE) present different efficiency characteristics that depend on torque and speed. Therefore, the torque request can be partitioned among these two sources to obtain the best out of the coupling in terms of efficiency. The set of control strategies to achieve this goal goes under the name of Energy Management Systems (EMS). The EMS optimises the energy efficiency and at the same time guarantees the torque request made by the driver via throttle pedal. EMS typically uses the actual power request, but the knowledge of future torque needed and/or of driver’s intended maneuver might greatly improve the system performance both in terms of energy efficiency [1] and also of driver satisfaction. For these reasons, control strategies for EMS and driver’s intention estimation have gained interest in research.
In this work we address both the problem of estimating the future torque required through the speed profile, and the online optimization of the torque distribution for hybrid electric vehicles which minimizes fuel consumption and preserves battery life. The literature on the development of control strategies for HEV is extended and varied; exhaustive surveys on the most recent methods can be found in [2]–[5]. A recent review that focuses on the differences between strategies is [6]. It provides a clear classification of HEV energy management strategies between rule-based, optimisation based and learning based and discusses an equivalent factor to compare fuel consumption and electric energy cost in the optimisation objective function. The authors of [6] also discuss other key technologies necessary to have good performances in EMS, like velocity prediction. The recent work in [7] provides a comprehensive review of EMS for hybrid vehicles trained on driving cycles data. It discusses the features of the driving cycles and their relationship with the adopted learning methods. The most appropriate driving cycles are also listed. A first distinction can be made between the Rule-Based (RB) strategies, which rely on a predefined logic, and the optimization-based strategies, which exploit the possibility to minimize some target quantity (e.g. fuel consumption) by means of different techniques (e.g. optimal control). The rule-based strategies still represent the easiest solutions to equip a hybrid vehicle with a real-time implementable control [5]. Compared to any type of optimization, a set of predefined rules require less computational power and, in the vehicle design and testing phases, it can benefit from the experience of the designers [8]. However, despite these valid arguments for RB, a significant portion of their design process consists of tuning the parameters and, when we consider the performances, optimization strategies outperform the RB ones [8], [9]. Advanced rule-based strategies are often combined with some optimization strategies; the authors of [10] combined fixed rules with an ECMS approach and successfully compared the algorithm with a benchmark obtained from the resolution of an offline dynamic-programming optimal control problem.

All optimization-based approaches have in common the minimisation of some cost function that usually features the fuel consumption and other variables like battery degradation. Despite this common goal, different techniques are employed in the current literature: Equivalent Consumption Minimization Strategies (ECMS), model predictive control (MPC) and offline optimizations. The widespread ECMS technique is based on the minimization of the fuel consumption by satisfying the necessary optimality conditions, which are obtained using the Pontryagin Minimum Principle (PMP). The ECMS cost function contains the fuel usage and the power output (with sign) of the battery. The latter is weighted by a factor named equivalence factor, which converts the battery power into an equivalent fuel consumption rate [11]. More recent instances of this method can be found in [12] and [8]. In [8] the authors extended the ECMS method by taking into account the aging of the battery, adding a model of this phenomenon that depends on the Ah-throughput. In [13] the battery aging is taken into account by means of a “severity factor map”, which associates instantaneous operating conditions to different rates of battery degradation. Another recent example is [14], where the authors improve a standard ECMS by learning a map of the equivalence factor on a specific driving cycle. ECMS can be used as an instantaneous optimization algorithm, therefore it does not necessarily require knowledge of the future power demand. In [15] an adaptive energy management strategy based on the ECMS framework is developed to optimise both flexible torque request and the gearshift commands. Some works presented offline optimal control techniques [16], [17], which cannot be implemented in real-time, but they can provide references for tuning or design rule-based approaches. On the other hand, MPC (or receding horizon) approaches require some knowledge on the future power demand which, on the other hand, may be exploited to implement cooperative power split optimization for a fleet of intelligent electric vehicles [18]. Moreover, the optimal control techniques applied in an receding horizon scheme can pose a limitation on real-time feasibility because of their relevant computational cost. Recently, the authors in [19] combined dynamic programming with reinforcement learning. The authors in [20] employed a speed prediction algorithm (in order to predict the power request on the future horizon), a simple equivalent-circuit battery model, a semi-empirical battery aging model and online dynamic programming optimization to minimize an equivalent cost, which takes into account battery aging, fuel consumption and electric energy consumption. Similarly, in [21] an online dynamic programming approach is used in pair with a hierarchical velocity prediction strategy in order to minimize fuel consumption. In [22] the speed profile and the reference state of charge (SOC) are tracked with a multi-objective optimisation problem that minimises both electric and fuel costs and battery degradation and it is solved in real-time with a direct multiple shooting method and sequential quadratic programming algorithm. Most of the energy management systems aim at the reduction of fuel consumption. However, it is well known that a severe usage of the battery leads to a deterioration of its performances (battery aging): this can compromise the performance of the overall system in terms of cost and environmental impact, since the battery may need a replacement after few years (or kilometers of usage). In all of the aforementioned approaches the battery is modeled with equivalent circuits, which are not able to model the battery dynamics through all the range of the battery working conditions. In [23] a state-of-the-art electrochemical model is used to update the control constraints taking into account the aging, but in the optimal control problem the authors approximated the battery dynamics by using the equivalent-circuit battery model. The internal state of the electrochemical model is crucial to accurately represent the aging dynamics [24], [25]. In our architecture, we consider the entire electrochemical model and battery dynamics in the Optimal Control Problem (OCP), an approach that was first.
introduced in [26] in an offline OCP on some standard driving cycles.

In this paper we extend the results of [26] from the offline simulation to the online. The optimization problem and the battery model validated in [26] are cast in a Non-linear Model Predictive Control (N-MPC) framework in order to evaluate the overall energy and battery-life saving performances. To the best of the authors’ knowledge, an online EMS that considers such detailed electrochemical battery model and dynamics is not present in the literature. Moreover, the proposed EMS is paired with a novel bio-inspired velocity profile prediction scheme validated on naturalistic driving data.

Our work presents a complete architecture for an energy management system of a hybrid electric vehicle. It is comprehensive of a bio-inspired prediction algorithm that uses vehicle’s velocity and acceleration only, and an online optimization strategy that takes into account battery aging modelling, while being executable in real-time. The model used for the optimal control problem (OCP) is a state-of-the-art electrochemical battery model that includes the aging dynamics and the consequent capacity loss. The OCP minimises the total cost: the price of the battery and the price of the fuel are used as weights in order to convert the cost function into an equivalent economic cost. The velocity prediction algorithm is compared to other three effective approaches that are well known in literature. The approach is validated on naturalistic driving telemetries (a dataset of 30 driving cycles performed by 10 drivers). The offline approach proposed by some of the authors in [26] is reproduced on the naturalistic dataset and it is used as a benchmark for the online approach. The proposed EMS is firstly tested in pair with an exact velocity prediction (employing two different prediction time-horizon lengths) in order to validate the choice of the prediction horizon length. Then, the EMS is tested in pair with the novel prediction scheme to assess the effectiveness of the prediction in feeding the future intention of the driver to the EMS.

1) ARTICLE MAIN CONTRIBUTIONS AND OUTLINE

The manuscript main goal is to propose a new bio-inspired approach to predict the driver’s intended manoeuvre and exploit the estimated speed profile in an MPC based EMS to optimise fuel consumption and battery life. More in details the contributions of the work are:

1) the bio-inspired framework to predict the driver’s intended maneuver and consequently the future speed profile (in a time window of 5s). The approach is validated on standard driving cycles (e.g. UDDS and Manhattan) and on driving cycles extracted from a dataset of naturalistic driving that includes drivers with diverse driving styles,

2) the prediction algorithm does not need to be trained, depends on few parameters (although learning the parameters may improve the prediction) and requires as input standard signals already available in CAN bus line of all vehicles,

3) the bio-inspired framework can be easily scaled to include other maneuvers such as car-following and overtakes as soon as data of other vehicles are available from the on-board sensors, to improve the prediction,

4) to propose an EMS based on a model predictive control approach for online optimization of an hybrid vehicle with a detailed electrochemical battery model comprehensive of the aging effect,

5) to show that the combination of the proposed prediction framework and the MPC with the battery electrochemical model allows to reduce fuel consumption and extend battery life with performance close to the off-line optimal solution.

The paper is structured as follows: Section II illustrates the prediction scheme and its theoretical background. In addition, Section II introduces the experimental driving dataset employed to validate the proposed prediction algorithm and compares the latter with other methods in the literature. In Section III the powertrain and battery models used in the EMS are discussed. The EMS is described in Section IV in terms of the optimal control problems formulation: both off-line over the entire driving cycle and the online receding horizon approaches are presented. In Section V the validation methodology and the simulation results are discussed. In the last section we draw the conclusions.

II. VELOCITY PREDICTION

The prediction of the future velocity profile plays an important role in MPC-based EMS for hybrid electric vehicles. These strategies often require some knowledge on the future power (or torque) request in order to get the best out of the limited capacity of the battery and to guarantee the driver’s needs. Several methods are available in the literature for future velocity prediction. A first classification of these methods regards the horizon length: the prediction can be defined as short term or long term. The first category involves horizon lengths up to a minute, while the second indicates predictions that accounts for all the driving route. Long term approaches usually rely on the a-priori route knowledge coming from the vehicle navigation system and they make higher approximations on the velocity profile. In this work we are interested in short-term predictions, since the navigation route is considered unknown although it could be taken into account if available. A further and effective way to distinguish between prediction methods is between parametric and data-driven. In the parametric category we collocate all methods that propose gray-box models. The data-driven category comprehends all the methods that use historical driving data to calibrate black-box prediction models.

In the first category, we can find models that take into account the vehicle dynamics and models of the driver behavior. In [27] the authors use the information of the speed limits over the route to formulate a parametric velocity prediction model aimed to feed an automatic gear shift. An auto-regressive model (ARMA) is used in [28]: the model is tuned...
using historical data and it provides effective results in short horizons on standard driving cycles. Other parametric methods take into account the interactions between the vehicles to guess the future velocity profile. In [29] the velocity is predicted in the case of vehicles in a convoy: the authors use traffic flow models combined with the Intelligent Driver Model (IDM) [30], assuming that the leader vehicle shares the information about its velocity by means of Vehicle to Vehicle (V2V) technology. In another work, [31], the same authors propose to use 5 different auto-regressive models representing 5 macro cases of behaviours: from hard deceleration to hard acceleration. The transitions among these models are represented by a Markov process where the states are fuzzy encoded in the acceleration variable. This encoding is different depending on some velocity regions. The approach is then validated on real driving data. More recently, in [32], the authors use again the IDM combined with a set of look-ahead data. The prediction occurs at two levels: long and short term. The long term is the one used to predict with remarkable accuracy the SOC evolution along the trip. The system is validated in simulation and the necessary parameters of the IDM are assumed as known. In [33] a pattern-sequence method, obtained through a clustering technique, is proposed to predict the future velocity profile.

Data-driven methods gained popularity in the last decade. Several methods are used and compared in [34]: kernel regression, neural networks, and nonlinear auto-regressive networks with exogenous inputs (NARX). In [35] the NARX based approach is compared with four stochastic prediction models: two of them relying on linear and Gaussian assumptions and the other two defined as non-parametric. These four models were validated on naturalistic driving data. In [36] an RBF neural network with a single hidden layer is trained on a wide set of standard driving cycles and validated on UDDS for a horizon of 10s with an RMSE of 1.37 m/s. Other approaches that use non-parametric models are characterized by employing historical driving data not only for training the prediction schemes but also as inputs. In [37] and [38] a sliding window of the past velocity is used to predict the future velocity profile.

The advantages and disadvantages of the methods are discussed in detail in the cited works, therefore we summarise the main pros and cons of the different groups. The parametric models are robust and require few parameters, but in order to improve the accuracy in the short term they may need some parameters that are specific for every single driver and difficult to estimate. Non-parametric models in most of the cases consist of neural networks. Those methods usually show higher accuracy and they do not require to know specific parameters of the vehicle or of the driver. However, they certainly need a significant amount of data to be trained and it is difficult to prove their performance in advance on different vehicles and drivers.

In this work, we take advantage of an established bio-inspired framework to design a velocity predictor. The bio-inspired framework is briefly explained in subsection II-A. The velocity predictor is designed and validated having in mind some precise objectives:

- It must be suitable for the proposed EMS
- It can be tuned easily with few parameters
- It must use only the state of the vehicle as input
- It must be valid for a broad set of different drivers

For the first requirement, the validation is made by comparing the performance of the EMS using the predictor and using the perfect knowledge of the future profile (exact prediction). For the other three requirements, the validation is made comparing the prediction results in the available naturalistic data set of 10 drivers. The requirement of using only the actual state of the vehicle is, on one side a limitation of the prediction performance achievable, on the other does not require to equip the car with expensive additional sensors (e.g., radar, video cameras, etc). Despite this, it is important to point out that the proposed method is already suitable to scale up to predict a larger set of maneuvers making use of data describing the surrounding driving scenario. Although an option not considered in this work.

A. THEORETICAL BACKGROUND

The prediction scheme proposed in this paper is derived from the concept of the Co-Driver that is presented in [39]: an agent able both to drive like a human and to understand driving intentions. The Co-Driver consists of a modular framework aimed at modeling the behavior of human drivers and designed based on several biological ideas that replicate the adaptive behavior and motion control of animals and humans. The first notion to consider is the evidence that optimality criteria govern human motion and manipulation of objects, as revealed in the studies [40]–[51]. Two optimality criteria that emerge are the minimum variance principle and the minimum jerk. The first, presented in [47], states that humans tend to minimize the variance of movements when close to the final target regardless of the previous states. Instead, the minimum jerk criterion states that the derivative of the acceleration is minimized [52].

The second notion to consider is the concept of affordance [53], which acquires importance in the affordance competition hypothesis [54]. The concept of affordances in driving is already explored in some studies: [55], [56], and [57]. An affordance is a latent behaviour or action that can be executed to interact with an object or a situation. In the affordance competition hypothesis, acting is conceived as a continuous selection between the possible affordances that the environment “offers”. The work in [56] provides a good example, at intersection. In this case, the driver continuously considers at least two different affordances: crossing the intersection or stopping before it. The same affordances are artificially generated in [58] to understand which one the driver intends to pursue, and returning a warning if the chosen affordance is dangerous. The concept of affordance is hereby used to generate atomic motor units called longitudinal motion primitives, which link the current state of the...
vehicle to a goal, represented by a final state. The concept of atomic units of motion planning has been also explored in neuroscience [59], [60]. The longitudinal motion primitives are the fundamental building blocks of our prediction scheme and are explained in Section II-B. The Co-Driver paradigm is used in different contexts. Similarly to [58], in [61] a Co-Driver instance uses the vehicle state to predict dangerous behaviours and issue warnings, by means of V2V communication and sensors on the road. In [57] a Co-driver instance mimics the stop maneuvers of human drivers by concatenating different motion primitives to achieve more sophisticated maneuvers. In [62], a Co-Driver becomes a highly adaptive artificial agent to drive an autonomous vehicle and in [63] a Co-Driver is employed by some of the authors for velocity profile prediction, using a scheme that is a precursor of the one proposed in this paper.

B. MOTION PRIMITIVES

The longitudinal motion primitives are the basic component of the Co-Driver. They represent atomic maneuvers that a driver can plan and pursue. The objective of these maneuvers is to provide a human-like link from the current vehicle state to a specific goal of driving (e.g. reaching a desired velocity), thus resembling the concept of affordance. The longitudinal motion primitives directly derive from the minimum jerk principle applied to driving. Here, the longitudinal jerk is the control variable. It is also assumed that human drivers plan the longitudinal motion of the vehicle following a "simplified" dynamics [64]. In fact we assume that a driver is able to control the dynamics of the vehicle, driving a desired jerk profile. The simplified dynamics of the driver-vehicle loop is expressed by:

\[ \dot{s}(t) = v(t) \quad \dot{v}(t) = a(t) \quad \dot{a}(t) = j(t), \]  

(1)

where \( s(t) \) is the longitudinal coordinate in the direction of the velocity, \( v(t) \) is the longitudinal velocity, \( a(t) \) is the longitudinal acceleration and \( j(t) \) is the longitudinal jerk, which acts as the control variable.

The primitives are the atomic units of the longitudinal driving. The driver is assumed to plan the control input for the entire duration of one motion primitive and he/she can plan for several primitives at the same time (affordance competition). The maneuver is formulated as an optimal control problem that makes use of the model in (1) and of a cost function that reflects the minimum jerk principle (and the minimum variance principle for this specific model). The initial conditions of the primitives are given by the state of the vehicle at the present time and the final conditions depend on the goal of the primitive:

\[ s(0) = 0 \quad v(0) = v(t_i) \quad a(0) = a(t_i) \]

\[ v(T_m) = v_f \quad a(T_m) = a_f. \]  

(2)

The initial curvilinear position \( s \) is always assumed to be equal to zero since the primitive represents a relative longitudinal motion from the current position. The final position is free. The optimal control problem that generates the primitive is:

\[ \min_{j(\cdot)} \int_0^{T_m} j^2(t) \]  

subject to (1), (2).

(3)

In our case we define only one type of primitive, changing the final conditions to express the three types of actions considered in this work. The parameters of the maneuvers are then three: \( T \) (maneuvering time), \( v_f \) and \( a_f \) final speed and acceleration, respectively. In the following sections, we define specific primitives that represent different types of action. To make the notation compact, the generation of a primitive is indicated by the following expression:

\[ M_{\text{instance}} = M(T_m, a_f, v_f). \]  

(4)

Note that the initial conditions are given by (2) and they always correspond to the current state of the vehicle, while the final ones (and the time \( T \)) are provided to the \( "M" \) function in the order indicated in (4). The solution of the OCP problem in Equation (3) can be found analytically and it is reported in (5) and (6). The solution (referring to \( s(t) \)) is a fourth order polynomial in \( t \), and the states and the controls can be easily computed through its derivatives using Equation (1).

\[ p(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4. \]  

(5)

The coefficients of Equation (5) depend on the duration of the maneuver \( T_m \) and on the initial conditions, and are expressed by:

\[ c_0 = 0 \]

\[ c_1 = v_0 \]

\[ c_2 = a_0 \]

\[ c_3 = \frac{v_f - v_0}{T_m^2} - \frac{2a_0}{3T_m} \]

\[ c_4 = \frac{v_0 - v_f}{2T_m^3} + \frac{a_0}{4T_m^2}. \]  

(6)

An important quantity in the primitives is the initial jerk \( j_0 = j(0) \), since it represents the instantaneous effort required to start a maneuver. This quantity can be derived directly from the fourth coefficient of the polynomial in Equation (6):

\[ j_0 = 6c_3. \]

C. PREDICTION SCHEME

In this work, we propose a simple architecture for the prediction that is based on a longitudinal motion primitive that mimics the minimum jerk profiles for longitudinal driving generated by human drivers. Different instances of these maneuvers are generated and one of them is selected through another bio-inspired action-selection process: the Multi Hypothesis Sequential Ratio Test (MSPRT). The types of maneuvers are chosen as a trade-off between accuracy and generality in order to design a predictor that is suitable for different types of drivers without the necessity of tuning parameters. In Figure 1, a high-level scheme of the
The predictor is shown. The predictor is based on a Co-Driver scheme factoring two blocks: one action-priming block that produces the competing affordances in the form of motion primitives, and one action-selection block that picks out one among them. All the operations are executed at every time step, from the action-priming to the action-selection. The computational load of this operation is negligible compared to the one required to solve an MPC step of the EMS proposed in this work. To improve the predictor reactiveness, the time step of the predictor is 0.1 seconds, 5 times shorter than the one of the EMS. This allows the MSPRT to collect more samples and to exploit the fast refresh rate of the vehicle sensors.

D. ACTION-PRIMING

The action-priming block generates a new set of actions at every time step. It gets the current state \([v(t), a(t)]^\top\) as input at each time step \(t_i\) and yields three types of possible future maneuvers:

- Maintain
- Accelerate
- Decelerate.

Every action type corresponds to one maneuver, each of them explained in the following subsections. It is essential to point out that the duration of the maneuver \(T_m\) does not necessarily coincide with the horizon time \(T\) used in the prediction phase, which is typically shorter and of the order of \(5s\).

1) MAINTAIN

This action has the objective of maintaining the current speed almost constant, keeping a “cruise” state. One may be tempted to use a constant speed profile, but this solution would not account for small (but not negligible) speed adjustments. Therefore, we use a longitudinal motion primitive that drives smoothly the profile to an estimated velocity adjustment. The final target velocity \(v_T\) is expected to be very similar to the current one \(v_0\). It is estimated by selecting the longitudinal motion primitive with the least instantaneous effort, which corresponds to \(j_0 = 0\). To find \(v_T\), at horizon time \(T\), we impose the aforementioned condition on the third coefficient of the maneuver polynomial (5) to get the estimated final velocity:

\[
v_T = \frac{1}{6} (4a_0T_m + 6v_0). \tag{7}
\]

Note that only the initial jerk is zero, thus it is not a constant acceleration maneuver. Once this velocity is known, the maneuver is defined and \(T_m = T\).

\[
M_{mant} = M (T, 0, v_T). \tag{8}
\]

2) ACCELERATE

This type of action assumes that the driver wants to increase his/her velocity by a significant amount, for instance, to enter a highway or to change to a higher velocity lane. The final target velocity \(v_T\) is estimated using the current acceleration value assuming it remains constant in the horizon \(T\). Then the estimated final velocity \(v_T\) is used in (5) and (6) to get a more realistic motion primitive to accelerate and reach the target speed \(v_T\). In the final speed \(v_T\) estimation, the current acceleration \(a_0\) is preferred to the current jerk \(j_0\) because the latter is difficult to estimate, as stated in [65]. The maneuver used for acceleration is therefore the following:

\[
M_{acc} = M (T_m, 0, v_0 + a_0T_m). \tag{9}
\]

Note that \(T_m\) is not the prediction horizon time \(T\), because the duration of the maneuver does not necessarily correspond to the latter. For “accelerate” and “decelerate” actions a higher duration (10 seconds) is selected.

3) DECELERATE

This last case is the dual of the accelerate case. This action models the driver’s intention to stop or to reduce his/her velocity consistently. For the stop maneuvers, following the minimum jerk concept, there exist more sophisticated models based on motion primitives as in [57], but they require more parameters that may change among drivers. We aim at modeling the main phase of the stop maneuvers (see [57]), which is the most important in terms of duration. The employed maneuver is the same as in the acceleration case, plus an additional constraint for the final velocity to be nonnegative.

\[
M_{dec} = M (T_m, 0, \max (v_0 + a_0T_m, 0)). \tag{10}
\]

The deceleration and the acceleration actions are mutually exclusive, since the final velocity depends on the acceleration sign. This means that actually only one of them is generated, but in the action-selection process both are taken into account since the selection process depends also on past values. To illustrate how the primitives are used, in Figure (2) three examples of predictions are shown. They make use of deceleration, acceleration and maintain respectively, from top to bottom. The dotted lines show the entire primitive \((T_m = 10\text{ seconds})\), the red lines illustrate the portion of the primitives used for the predictions \((T = 5\text{ seconds})\) and the blue lines are the actual velocity from the dataset. For each of the three predictions, the maneuver that is not selected is shown in grey, note that only one excluded maneuver is shown since acceleration and deceleration actions are mutually exclusive.

4) ACTION-SELECTION: MSPRT

The action-selection block chooses an action among the pool of available ones by means of the MSPRT algorithm [66].
One example of application in the automotive field can be found in [67]. This algorithm models the action-selection process occurring in the basal ganglia of the human brain [68]. The MSPRT process, in the context of this prediction, is based on collecting evidence on which of the three possible actions the driver is actually pursuing (the action is also called channel). The key idea is to accumulate evidence at every time step for each channel (the accumulated evidence is called salience). One channel is then selected after the accumulated evidence is considered enough. Here the evidence metric $ea_k$ is the reciprocal of the mean square error of the acceleration in a time window of the previous 0.5 seconds. The error, in this case, is the difference between the actual acceleration and the values given by the candidate maneuvers of a passed prediction step (the step occurred 0.5 seconds before current time $t_i$). Given the index of the different channels $k = 1, 2, 3$, the evidence is expressed by:

$$ea_{k,i} = \frac{1}{\sqrt{\sum_{t=0}^{4} ((a(t_i - 0.5 + p t_s) - \hat{a}_k(t_i - 0.5, p t_s))^2 / 5}}$$

where $t_i$ is the current time, $t_s$ is the sampling time of the prediction (0.1 seconds). The values of the true acceleration are $a(t_i - 0.5 + p t_s)$, which are known since are in the past up to the current value. The acceleration values $\hat{a}_k(t_i - 0.5, p t_s)$ are instead drawn from the three action candidates of 0.5 seconds before $t_i$. Past candidates are used instead of current candidates, since the motion primitives cannot be evaluated in $t < 0$, because such values are out of the maneuvers time domain $t \in [0, T_m]$. For the computation of $ea_{k,i}$ we therefore require a set of past maneuvers that can be obtained in two ways indifferently. Past maneuvers can be stored in memory, or they can be computed again by means of stored past vehicle states. The evidence $ea_{k,i}$ is accumulated into the salience, appending and summing past values of the evidence. The number of past evidence values used for the Salience is defined as $w$. This buffer length $w$ is limited to 10 in order to avoid using outdated data. The salience is then defined as the mean of the evidence along a past window containing a number of samples equal to $w$:

$$Y_{k,i} = \frac{\sum_{j=0}^{w-1} ea_{k,(i-j)}}{w}.$$  \hfill (12)

The length of the window $w$ along which $Y_k$ is initialized to 1 and it is incremented by one at every step, up to 10. If one action is selected, $w$ is reset to one. Finally, the quantity used to select a channel is the likelihood $L_k$ of each channel. It takes into account the salience of each channel in relation to the salience of the others. The likelihood assumes values from 0 to 1, $L_k \in [0, 1]$ and the sum of the likelihoods of the channels is always equal to 1, $\sum_{k=1}^{3} L_k = 1$. It reads as:

$$L_k = \exp \left( Y_{k,i} - \log \left( \sum_{k=1}^{3} \exp (Y_{k,i}) \right) \right).$$  \hfill (13)

One channel is selected if its likelihood reaches a predefined threshold $L_{th}$. If more than one channel reaches the threshold, the one with the higher likelihood is selected. Thus, the channel is selected according to:

$$k_i^* = \arg \max_k \left\{ L_{k,i} \mid L_{k,i} > L_{th} \right\}$$  \hfill (14)

where $k_i^*$ is the selected channel at time $t_i$. The complete procedure for the MSPRT is illustrated in Algorithm 2.

Finally, the threshold $L_{th}$ must be tuned: if the value is too low, the channel changes continuously and more than one channel may reach the threshold simultaneously. Conversely, if the value is too high, the MSPRT algorithm may stick without choosing a channel since the likelihoods of the channels never reach the threshold. Note that if the chosen threshold $L_{th}$ is more than 0.5, then only one channel at a time can exceed the threshold, and the arg max operator in Equation (14) is not required.

### E. Naturalistic Driving Dataset

We hereby introduce the experimental data for the validation of the prediction scheme (they will also be used to validate the EMS). These data were collected during the user tests of the EU FP7 ‘interactIVe’ project [69]. In the project, data from several sensors were logged at 100 ms. The signals relevant for this work are the longitudinal velocity and the longitudinal acceleration. The test route consisted of a heterogeneous path around Turin, Italy. The route included extra-urban roads, motorways, and different types of intersections, including roundabouts. Figure 3 shows the route on a map. The circuit was driven by 10 different drivers. Each circuit telemetry is then cut in three segments of 6 minutes, leading to 30 driving data sets. A long phase of almost constant velocity was not used in the analysis since it could not exploit the benefit of the hybrid transmission and it could affect the statistics by making the prediction overrated due to the triviality of predicting constant velocity. In the data sets, the slope of the road is not available: this will affect the performance of the
the knowledge of the road ahead in terms of intersections, they would require knowledge of the environment, such as

section horizons could be estimated, but, in order to be accurate, exact future velocity. Longer (more than 10 seconds) predic-

tion will not improve when a 10 seconds horizon of the predicted future is provided, the latter being an approximation of the predicted energy management system does not sensibly change in Section VI, where we show that the performance of the pro-

posed algorithm.

section VI, where we show that the performance of the pro-

posed energy management system does not sensibly change in Section VI, where we show that the performance of the pro-

posed algorithm.

system in term of potential fuel saved, but in terms of velocity profiles we assume that the driver always pursues his/her desired maneuver, compensating for the slope.

F. VALIDATION

The scope of the prediction scheme is to predict the velocity profile to feed the energy management system and estimate the future requested total torque. We validated the proposed prediction scheme by setting the prediction horizon length to 5 seconds. This choice is motivated by the results presented in Section VI, where we show that the performance of the pro-

posed energy management system does not sensibly change when a 10 seconds window is used as horizon to inject the future exact velocity. Since the EMS performance does not improve with the exact velocity profile, as a consequence it will not improve when a 10 seconds horizon of the predicted velocity is provided, the latter being an approximation of the exact future velocity. Longer (more than 10 seconds) prediction horizons could be estimated, but, in order to be accurate, they would require knowledge of the environment, such as the knowledge of the road ahead in terms of intersections, roundabouts, corners, position and velocity of the preceding vehicle, speed limits etc. One goal of the prediction scheme (and EMS) proposed in this work is to only make use of the basic CAN signals of the vehicle.

Another goal is to provide a suitable prediction regardless of the driver. For this purpose, the prediction parameter for the acceleration/deceleration action \( T_m \) and the threshold of the MSPRT \( L_{th} \) were manually tuned on the data set presented in Section II-E. The chosen values are 0.5 for \( L_{th} \) and 10 seconds for \( T_m \). To tune these parameters, we employed 2 out of 3 datasets for each driver, we stopped the tuning when the RMSE errors (see Equation (15)) lied under 3 m/s (excluding the outliers) for all drivers. When we meet this requirement on the first two datasets, we validate the choice of the values on the third dataset (for each driver), applying the same requirement. The errors of all datasets are shown in Figure 4. To evaluate the performance of the prediction, the error on the velocity in each prediction time horizon starting at time \( t_i \) is computed as the root mean square along the prediction:

\[
\text{RMSE}_v(t_i) = \sqrt{\frac{\sum_{k=1}^{L} (v(t_i + kT) - \bar{v}(t_i, kT))^2}{L}}, \tag{15}
\]

where \( t_i \) is the current time at which the error is calculated, \( v(\cdot) \) is the actual velocity and \( \bar{v}(t_i, \tau) \) is the instance at \( t_i \) of the prediction evaluated at time \( \tau \) along the maneuver. Note that \( \bar{v}(t_i, 0) = v(t_i) \). \( L \) is the number of time frames in the predicted maneuver time horizon, which contains 50 samples in our case. These results are shown in Figure 4, where the distribution of the RMS errors for each driver are shown.

In Figure 5 (left), the overall error distribution is shown as well as the error at the end of the prediction expressed by \( e_{end} = |\bar{v}(t, T_h) - v(t + T_h)| \) where \( T_h = t_i L \) is the prediction horizon length (5 seconds). An example of predictions on a dataset is shown in the same figure on the right.

By looking at Figure 4 and Figure 5 we can draw some considerations. While most of the predictions present small errors, there is a strong presence of outliers. These predictions with a higher error can be explained by the cases in which the driver’s intention changes and the MSPRT needs some more time samples in order to collect enough evidence of the new current action. This is especially true when there is a switch between acceleration and maintain velocity cases, which can be easily noticed in some of the peaks where some few
“diverging” predictions appears (For instance, in Figure 5 a zoom of the plot is provided at $t = 60s$). The availability of additional information, e.g. the presence of obstacles with their speed or the distance to an incoming stop sign, would allow us to introduce other affordances (i.e. potential maneuvers and related channel in the MSPRT algorithm) and greatly improve the prediction. However, in this work, the scope of the velocity predictor is to supply a reliable velocity prediction using only the basic vehicle state, i.e. the information of the vehicle velocity and acceleration.

G. COMPARISON WITH OTHER METHODS

We compare the proposed Bio-Inspired velocity prediction method with two other approaches that are widely used in the literature, which are:

- Exponential decreasing torque demand
- Markov chains

All the methods are compared on a prediction horizon length equal to 5 seconds. The choice of this horizon length is motivated in Section II-F In the following subsections, the implementations of the two methods are explained and the results of the comparison are discussed.

1) EXPONENTIAL DECREASING TORQUE DEMAND

This method assumes that the driver torque request decreases exponentially in time, as suggested by the authors of [70], [71]. The predicted torque profile on the receding horizon is thus given by

$$T_w(t_k + k\delta) = T_w(t_1) \exp \left( \frac{-k\delta}{\alpha_d} \right), \quad k = 0, 1, \ldots, L - 1,$$

(16)

where $t_1$ is the starting time of the prediction horizon, $L$ is the number of time samples in the prediction horizon, $T_w(t_1)$ is the current torque request and $\alpha_d$ is the decaying rate of the torque request. We compute the parameter $\alpha_d$ by solving an unconstrained nonlinear optimization problem. The resulting $\alpha_d$ is the one that minimises the root mean square error between the predicted velocity and the real one on the dataset. Once the future torque request profile is estimated, the velocity profile can be computed by the integration of the longitudinal acceleration obtained from the longitudinal dynamics equation (21).

2) MARKOV-CHAIN BASED VELOCITY PREDICTION

The use of discrete-time stochastic models in the form of Markov chains to predict the future power, torque or velocity profiles requested by the driver is widely found in the literature ([33], [72]–[77]). We use a multivariate Markov chain in order to predict the future velocity (and acceleration) profile ([72], [74]). The velocity and acceleration (which are continuous quantities) are divided in $M$, $N$ discrete states, respectively:

$$v \in \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_M\}$$

$$a \in \{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_N\}$$

(17)

A Markov chain is defined by a transition probability matrix: in the case of the 1-Stage Markov chain the elements of the multi-dimensional transition probability matrix $T_{1S} \in \mathbb{R}^{M \times N \times M \times N}$ are defined as:

$$i_{(m,n),(p,q)}^{1S} = P \left[ v(t_1 + t_s) = \hat{v}_p, a(t_1 + t_s) = \hat{a}_q \mid v(t_1) = \hat{v}_m, a(t_1) = \hat{a}_n \right],$$

$$m, p = 1, \ldots, M, \quad n, q = 1, \ldots, N,$$

(18)

where $P[x]$ is the probability of event $x$ happening, $t_1$ is the current time instant and $(t_1 + t_s)$ is the next time instant. In the case of the 3-Stage Markov chain, two past velocity values are also considered as additional variables ([33], [72]). We obtain the transition probability matrix $T_{3S} \in \mathbb{R}^{M^3 \times N \times M \times N}$, which elements are:

$$i_{(m_1,m_2,m_3,n),(p,q)}^{3S} = P \left[ v(t_1 + t_s) = \hat{v}_p, a(t_1 + t_s) = \hat{a}_q \mid v(t_1 - 2t_s) = \hat{v}_{m_1}, v(t_1 - t_s) = \hat{v}_{m_2}, v(t_1) = \hat{v}_{m_3}, a(t_1) = \hat{a}_n \right],$$

$$m_1, m_2, m_3, p = 1, \ldots, M, \quad n, q = 1, \ldots, N,$$

(19)

Both the transition probability matrix are estimated from the naturalistic driving dataset described in Section II-E and from standard driving cycles, as described in the following subsection.

3) RESULTS OF COMPARISON ON STANDARD DRIVING CYCLES AND NATURALISTIC DRIVING DATASET

The proposed and the two benchmark methods are compared on standard driving cycles, such as UDDS and MANHATTAN, and on our naturalistic driving dataset. Figure 6 shows the velocity prediction methods outcomes for one sample driving segment picked from the naturalistic test set, with the prediction horizon of 5 seconds. An objective metric to evaluate the quality of the prediction is the root mean square of the velocity error (RMSE) for a horizon of 5s. The metric (Equation (15)) is computed for each time step on the three available validation sets (i.e. UDDS, MANHATTAN, naturalistic driving dataset). The average
RMSE is then calculated for each validation set. However, differently than the Bio-inspired and the Exponential Decreasing torque methods, the Markov chains need a training dataset. Therefore, we evaluate the Markov chains with different combinations of the training set and validation set. The results are reported in Table 2 divided into group A and group B. In group A the validation sets are also the training set, or are contained in the training set. In this case the Markov chain approaches work quite well but it is obviously an over-fitting of the training dataset. On the contrary, in group B the validation sets are different from training sets and the Markov chains perform poorly, especially when the validation set has features, such as speed range, not considered in the training set. Table 1 reports the results (average RMSE) for the proposed method and the Exponential Decreasing torque method on the three validation sets. It is worth noting that these methods do not need a training dataset. The proposed Bio-inspired method performs better than the Exponential method and it is in line with the Markov chains results on naturalistic driving set in the case the validation and the training sets are the same (i.e. Group A). As a general outcome of the comparison, we can conclude that the proposed Bio-Inspired velocity predictor is more accurate, in terms of RMSE, compared to the other methods. The Exponential Decreasing torque method presents good performances when the accelerations (positive and negative) are small, but the proposed Bio-inspired approach can provide a better prediction in presence of high and continuous values of acceleration. The Markov chain based methods can provide better prediction in presence of high and continuous values of acceleration. The Markov chain approaches work quite well but it is obviously an over-fitting of the training dataset. On the contrary, in group B the validation sets are different from training sets and the Markov chains perform poorly, especially when the validation set has features, such as speed range, not considered in the training set. Table 1 reports the results (average RMSE) for the proposed method and the Exponential Decreasing torque method on the three validation sets. It is worth noting that these methods do not need a training dataset. The proposed Bio-inspired method performs better than the Exponential method and it is in line with the Markov chains results on naturalistic driving set in the case the validation and the training sets are the same (i.e. Group A). As a general outcome of the comparison, we can conclude that the proposed Bio-Inspired velocity predictor is more accurate, in terms of RMSE, compared to the other methods. The Exponential Decreasing torque method presents good performances when the accelerations (positive and negative) are small, but the proposed Bio-inspired approach can provide a better prediction in presence of high and continuous values of acceleration. The Markov chain based methods can provide better performances regardless of the acceleration values, even in the presence of intention changes (for example switching from braking to accelerating) but only on the dataset they are trained in. Additionally, the Markov chains are memory and data-intensive approaches. The Bio-inspired method provides maneuvers that are based on a model of the human driver and therefore they resemble feasible maneuvers. The Bio-inspired method, therefore, represents an improvement with respect to the proposed comparison methods from the literature, and it is computationally light. Furthermore, it does not require large datasets to be trained since the two parameters involved ($L_{th}$ and $T_{m}$) can be even manually tuned.

### III. POWERTRAIN AND BATTERY MODEL

The powertrain model studied in this work is the same presented in [26], i.e. a parallel mild-hybrid electric vehicle. A schematic of the power-train is shown in Figure 7. An electric motor is always connected to the internal combustion engine (ICE) shaft through a geared coupling. The ICE shaft is then coupled to an 8-speed automatic transmission through a clutch. The torques $T_{\text{eng}}(t)$ and $T_{\text{mot}}(t)$ generated by the ICE and by the electric motor sum up at the ICE shaft and are transmitted to the driving wheels after being multiplied by the transmission and differential gear ratios. The total torque $T_w(t)$ at the driving wheels is given by the following expression:

$$
T_w(t) = \left(T_{\text{eng}}(t) - J_{\text{eng}} \omega_{\text{eng}}(t) + T_{\text{mot}}(t) \gamma_{\text{mot}} \right)
\times \eta_{\text{trn}}^{\text{sign}(\alpha_{x}(t))} \gamma_{\text{trn}}(t) \gamma_{\text{axle}} + T_{\text{brk}}(t),
$$

where $J_{\text{eng}}$ is the ICE rotational inertia, $\omega_{\text{eng}}(t)$ is the ICE shaft angular acceleration, $\alpha_{x}(t)$ is the vehicle longitudinal acceleration and $T_{\text{brk}}(t)$ is the mechanical braking torque generated by the brake disks. The parameter $\gamma_{\text{mot}} > 1$ represents the reduction ratio between the electric motor and the ICE. Consequently, the kinematics relationship between the ICE and the electric motor angular velocities is $\omega_{\text{mot}}(t) = \omega_{\text{eng}}(t) \gamma_{\text{mot}}$. The differential gear ratio (final drive) is given by the constant parameter $\gamma_{\text{axle}}$ while the

| Validation set | Bio-Inspired [m/s] | E. Var. Torque [m/s] |
|----------------|---------------------|---------------------|
| Manh.          | 0.70                | 0.80                |
| UDDS           | 0.64                | 0.83                |
| Nat. Drv.      | 0.74                | 0.89                |

**TABLE 1.** Average RMS error for Bio-inspired and Decreasing torque prediction methods for three validation datasets.

| Group | Training set | Validation set | M. 1-S [m/s] | M. 3-S [m/s] |
|-------|--------------|----------------|--------------|--------------|
| A     | Manh.        | Manh.          | 0.49         | 0.47         |
|       | UDDS         | UDDS           | 0.53         | 0.52         |
|       | Nat. Drv.    | Nat. Drv.      | 0.61         | 0.77         |
|       | Manh. + UDDS | Manh.          | 0.57         | 0.54         |
|       | Manh. + UDDS | UDDS           | 0.58         | 0.55         |
| B     | Manh. + UDDS | Nat. Drv.      | 5.56         | 4.89         |
|       | UDDS         | Nat. Drv.      | 2.69         | 5.60         |
|       | Nat. Drv.    | Manh.          | 1.56         | 3.95         |
|       | Nat. Drv.    | UDDS           | 1.19         | 2.40         |

**TABLE 2.** Average RMS error for Markov chain prediction methods for two groups of combinations of validation and training set. In group B validation and training sets are different, whereas in Group A they are equal or the validation set is contained in the training set.

![Figure 6](image-url)
At each time instant $t$ the total torque at the driving wheels can be computed from the longitudinal dynamics equation

$$T_w(t) = \left(c_f v_x(t)^2 + m_v a_x(t)\right)r_w + [g m_v \left(\sin(\sigma(t)) + c_{tr0} \cos(\sigma(t))\right)]r_w,$$  \hspace{1cm} (21)

where $a_x(\cdot)$ and $v_x(\cdot)$ are the longitudinal acceleration and speed of the vehicle, and $\sigma_{road}(\cdot)$ is the road slope (the road slope is set to zero since it was not available in the driving dataset). The parameters of (21) can be found in Table 3.

### B. ENGINE MODEL

The ICE instantaneous fuel consumption is a function of the engine speed and torque at the shaft. Willan’s lines approach [78] approximates the fuel consumption rate $\dot{m}_f$ as an affine function of the engine mechanical power at the shaft, thus allowing us to write the following relationship:

$$\dot{m}_f(\omega_{eng}(t), T_{eng}(t)) = \alpha(\omega_{eng}(t)) P_{eng} + \beta(\omega_{eng}(t)),$$

$$\alpha(\omega_{eng}(\cdot)) = \sum_{i=0}^{3} \alpha_i \omega_{eng}^{i-1}(\cdot),$$

$$\beta(\omega_{eng}(\cdot)) = \sum_{i=0}^{3} \beta_i \omega_{eng}^{i-1}(\cdot).$$  \hspace{1cm} (22)

The Willan’s parameter values used in this study are the same of [26] and they are listed in Table 4. A representation of the consumption map is depicted in Section V, Figure 15, where the EMS engine usage is shown.

### C. ELECTRIC MOTOR MODEL

The electric motor is connected to a 48V battery and can act as a motor or as a generator depending on whether the vehicle is braking or accelerating. This electrical machine is also called internal starter generator - or ISG unit. The power flowing through the ISG in generator mode or motor mode is a function of the ISG shaft speed $\omega_{mot}(\cdot)$ and torque $T_{mot}(\cdot)$, and of the ISG efficiency $\eta_{mot}(\omega_{mot}(\cdot), T_{mot}(\cdot))$. The efficiency is usually given in motor data sheets as a contour plot of iso-efficiency lines on the motor Torque-Velocity graph. For each pair $(\omega_{mot}(\cdot), T_{mot}(\cdot))$, the motor actual power can be computed as

$$P_{mot}(\omega_{mot}(t), T_{mot}(t)) = \begin{cases} 
\omega_{mot}(t) T_{mot}(t) \eta_{mot}(t), & T_{mot}(t) < 0, \text{ gener. mode} \\
\frac{\omega_{mot}(t) T_{mot}(t)}{\eta_{mot}(t)}, & T_{mot}(t) \geq 0, \text{ motor mode}
\end{cases}$$  \hspace{1cm} (23)
while taking into account the motor efficiency. The result of Equation (23) is the smooth surface of Figure 8, which can be easily approximated using the third-order polynomial

\[
P_{\text{mot}}(\alpha_{\text{mot}}(t), T_{\text{mot}}(t)) = \sum_{j=0}^{3} \sum_{i=0}^{3} p_{i,j} \alpha_{\text{mot}}(t)^j T_{\text{mot}}(t)^i. \tag{24}
\]

The fitting of the power surface of Figure 8 is preferred, since the fitting of the less-smooth efficiency function \(\eta_{\text{mot}}(\alpha_{\text{mot}}(\cdot), T_{\text{mot}}(\cdot))\) would require an higher order polynomial.

### D. BATTERY SYSTEM

The battery, or the Energy Storage System (ESS), consists of 180 gr/NMC lithium-ion 18650 cylindrical cells. A number of \(n_p = 16\) parallel modules are present with \(n_s = 13\) cells for a total of 180 units. Each cell has a nominal voltage of 3.6 V for a nominal capacity of 2.0 Ah. The battery pack voltage is then 48 V and it stores approximately 1.5 kWh of energy in nominal conditions. The maximum discharge and charge powers are 17 kW and 11 kW respectively, coming from the corresponding maximum cells currents: 30 A and 20 A respectively. The overall battery power at time \(t\) is given by:

\[
P_{\text{bat}}(t) = V(t) n_s I(t) n_p,
\]

where \(V(\cdot)\) is the terminal voltage of one cell and \(I(\cdot)\) is the current applied to it. We consider the temperature of the cells constant, since we assume that the cooling system is able to maintain this condition. Generally this does not necessarily hold for standard batteries, in particular high current values may lead to local temperature gradients.

#### 1) CHARGE/DISCHARGE DYNAMICS

In this subsection, we briefly recall the equations of the electrochemical battery model presented in [26], which is the model our EMS is based on. The equation of the model is:

\[
V(t) = (U^p(t) - U^o(t)) + (p^o(t) - \eta^o(t))
+ (\phi_c^L(L, t) - \phi_c^e(0, t)) - R_O I(t), \tag{26}
\]

where the input of the model is the current \(I(t)\) applied to the cell (galvanostatic mode). The output is the terminal voltage \(V(t)\) measured between the positive and negative current collectors and it is the sum of the potential and overpotential terms. The values and description of the cell model parameters are the same of the work in [26], and they can be found there in the Appendix.

The equilibrium potentials \(U^e(t)\) depend on the lithium ion concentration on the surface of the electrode particle (namely the solid-electrolyte interface), indicated by \(c_{s,e}(t)\). Given the stoichiometry ratio as \(\frac{\theta^i(t)}{\theta^o(t)} = c_{s,e}(t)/c_{s,max} \in [0; 1]\), the functional form of the equilibrium potential at the cathode side (see [79]) is

\[
U^e(\theta^o(t)) = -10.72[\theta^o(t)]^4 + 23.88[\theta^o(t)]^3
- 16.77[\theta^o(t)]^2 + 2.595 \theta^o(t) + 4.563 \tag{27}
\]

and at the anode side (see [80]) is

\[
U^e(\theta^o(t)) = 0.1493 + 0.8493 \exp(-61.79 \theta^o(t))
+ 0.3824 \exp(-665.8 \theta^o(t))
- \exp(39.42 \theta^o(t) - 41.92)
- 0.0313 \tan^{-1}(25.59 \theta^o(t) - 4.099)
- 0.009434 \tan^{-1}(32.49 \theta^o(t) - 15.74). \tag{28}
\]

The Butler-Volmer equation relates the kinetic overpotential terms \(\eta(\cdot)\) to the current density \(j(\cdot)\) through the following relation:

\[
\eta(\cdot) = \frac{R_s T}{\alpha F} \sinh^{-1}\left[\frac{j(\cdot)}{2\alpha I(\cdot)F \eta(\cdot)}\right], \quad i = p, n. \tag{29}
\]

The exchange current density \(j_0^i(\cdot)\) is related as well to the concentration at the electrode surface \(c_{s,e}^i(\cdot)\) and to the concentration in the electrolyte \(c_{e}^i(\cdot)\); this relation reads as:

\[
j_0^i(\cdot) = k^i \sqrt{c_{e}^i(\cdot)(c_{s,max} - c_{s,e}^i(\cdot))c_{s,e}^i(\cdot)}, \quad i = p, n. \tag{30}
\]

Under the assumption of constant current density throughout the electrodes, we can compute the electrolyte overpotential \(\Delta \phi_c(t) = \phi_c^L(L, t) - \phi_c^e(0, t)\) by integrating the equation of the conservation of charge in the liquid phase. The latter reads as:

\[
\Delta \phi_c(t) = \kappa_d \log[c_{e}^p(t)] - \log[c_{e}^o(t)] - I(t) R_e, \tag{31}
\]

where \(\kappa_d = \frac{2R_s T(1-\alpha)}{F} (1 + \beta)\). The electrolyte resistance is expressed by \(R_e = \frac{1}{\kappa c_{e}^L \left( \frac{D_p}{\rho} + \frac{L_e}{\rho} \right)}\), where \(\kappa\) is the electrolyte conductivity, which depends on the temperature. In [81] equations for this dependency are detailed, presenting a thorough experimental analysis of the electrochemical properties of a LiPF\(_6\)-based electrolyte.

The transfer function between the SOC and the surface concentration at the electrodes was presented in [26],
This model also takes temperature variations into account; its state-space realization is given by:

\[
\dot{x}(t) = Ax(t) + Bl(t), \quad \text{with } x \in \mathbb{R}^7,
\]

\[
y(t) = Cx(t), \quad \text{with } y(t) = \begin{bmatrix} \tilde{c}^{\text{SEI}}_{e}(t) \\ \tilde{c}^{\text{SEI}}_{s}(t) \\ \tilde{c}^{\text{AM}}_{e}(t) \\ \tilde{c}^{\text{AM}}_{s}(t) \\ \tilde{SOC}(t) \end{bmatrix} \in \mathbb{R}^5.
\]

where the tilde indicates a variation from the equilibrium condition. The formulation of the state-space matrices is reported in Appendix. The vector \( x(t) = [x_1, \ldots, x_7]^T \) contains the state variables, while \( y(t) \) is the output vector (the interested reader can refer to [26] for further details on the system states and outputs).

The kinetic constants \( k^i \), the solid phase diffusion coefficients \( D^j \) and the activity coefficient \( \beta \) increase with temperature according to the Arrhenius-like equation

\[
\Gamma(T) = \frac{\Gamma_{\text{ref}} \exp \left[ -\frac{E_{\text{act}}}{R} \left( \frac{1}{T} - \frac{1}{T_{\text{ref}}^\Gamma} \right) \right]}{R_g}.
\]

Conversely, \( R_g \) decreases with the temperature; this reverse trend can again be described by (33), but with a sign change inside the exponential term.

2) BATTERY AGING

To model the aging of the cylindrical 2.0 Ah cell considered in this work, we adopt the aging model proposed in [82].

The Butler-Volmer equation for the solvent reduction kinetics is simplified as described in [26], leading to

\[
j_e(t) = -\frac{k_{\text{SEI}}(T)}{2A_0(1 + \lambda \theta(t))\sqrt{t}},
\]

where \( \theta(t) = \exp \left[ \frac{1}{R} \left( \eta^0(t) + U^0(t) - U_{\text{sei}} \right) \right] \). The behaviour of the kinetic coefficient for the side reaction \( k_{\text{SEI}}(T) \) is described by the Arrhenius dependency in (33). The effect of the anode potential on the SEI growth is accounted by the fitting parameter \( \lambda \). The integral of the side-reaction rate over time, i.e.

\[
Q_{\text{SEI}}(t) = \int_0^t j_e \Delta u \, dt,
\]

is equal to the capacity loss associated to the SEI formation. According to [82], the increased capacity loss observed after discharging and charging cycles is due to the structural damages that constantly isolate the active material. Assuming a uniform utilization, the phenomenon can be described by the variation of the active material, expressed by:

\[
\frac{d\varepsilon_{\text{AM}}(t)}{dt} = -\kappa_e(T) j_a(t),
\]

where \( \kappa_e(T) \) is related to the temperature according to (33).

The SOC-dependent rate of capacity loss induced by the reduction of the volume fraction reads as

\[
\frac{dQ_{\text{AM}}(t)}{dt} = \frac{\varepsilon_{\text{AM}}(t)}{\varepsilon_{\text{AM}}^{\text{ref}}} \cdot SOC(t) \cdot V^0 \cdot c_{\text{AM}}^{\text{act}}.
\]

The two capacity loss mechanisms are now combined, assuming the superposition of effects, modeling the total capacity loss at time \( t \). The final expression of the capacity loss is then:

\[
Q_{\text{loss}}(t) = Q_{\text{SEI}}(t) + Q_{\text{AM}}(t)
\]

\[
= -\int_0^t \frac{k_{\text{SEI}}(T)}{2(1 + \lambda \theta(t))\sqrt{t}} \, dt
\]

\[
-\int_0^t \kappa_{\text{AM}}(T) \cdot SOC(t) \cdot I(t) \, dt.
\]

The constants in (37) are condensed in the equivalent parameter \( \kappa_{\text{AM}}(T) \). Notice that the last term in (38) implies that high charge/discharge cycles at high SOC values will accelerate the battery aging.

IV. ENERGY MANAGEMENT SYSTEM OPTIMIZATION

The goal of the energy management optimization is to find the optimal torque split between the electric motor and the ICE in order to minimize the fuel consumption and the life cost of the battery. The battery life-aware torque split strategy is achieved by solving a non-linear optimal control problem on a fixed horizon that recedes in front of the vehicle (online optimal control). To assess how far the online solution is from an ideal benchmark, we also formulate the same optimal control problem over the entire driving cycle with the exact speed profile. The latter is called offline solution.

In the following subsections, we first introduce the general optimal control problem (section IV-A) and subsequently we describe the formulation details of the offline and online problems (in subsections IV-B and IV-C, respectively).

A. OPTIMAL CONTROL PROBLEM FORMULATION

From (21) we have that the total torque required by the vehicle driving wheels at each time instant is a function of the velocity and acceleration profiles. Thus, following from (20), the engine and motor torque \( T_{\text{eng}} \) and \( T_{\text{mot}} \), the mechanical braking torque \( T_{\text{brk}} \) and the cell current \( I \) must be optimally chosen to obtain the required total torque \( T_w \). These quantities are chosen as the controls for the problem. We thus define the control vector \( u(t) \) as

\[
u(t) = [T_{\text{eng}}(t), T_{\text{mot}}(t), T_{\text{brk}}(t), I(t)]^T.
\]

The general formulation of the optimal control problem for both the offline and online cases can be described as follows:

\[
\begin{align*}
\min \, J(x(t), u(t)) \\
\text{subject to :} \\
\dot{x}(t) &= f(x(t), u(t), t), \quad \text{battery dynamics} (32) \\
x(0) &= x(0), \quad x(T) = x(T) \\
\end{align*}
\]

with \( t \in [t_0, t_f] \) and \( t_0, t_f \) the initial and final time of the optimization time horizon respectively. The states are \( x \in \mathbb{R}^7 \) and they must respect the battery dynamics described by the system of differential equations (32). For both offline
and online optimal control problems, we also set some path constraints [26]. The battery power (25) must be equal to the electric power required by the motor (23). The battery state-of-charge (SOC) must remain between $SOC_{min} = 15\%$ and $SOC_{max} = 95\%$. The cell voltage must remain between 2.4 V and 4.2 V, while the electrode surface concentrations are constrained to be $c_{s, min}^i \leq c_s^i \leq c_{s, max}^i$, $i = p, n$. Finally, the sum of all the powertrain torques (20) is always equal to the required torque (21). For the offline problem the initial condition on the battery SOC is always $SOC(0) = 50\%$, while for the online problem this hold only for the first instance, after that the initial SOC depends obviously on the previous solution.

The cost function $J(x(t), u(t))$ to minimize comprehends two main terms, namely, the final total fuel consumption and the final battery capacity loss. In order to transform this problem into a single-objective problem, we would like to uniform their units: we chose to minimize the final monetary expense due to fuel consumption and battery degradation, expressed in euros [26]. In order to do so, the final fuel consumption (expressed in kg) and battery capacity loss (expressed in percentage of battery loss capacity) are multiplied by the two scaling terms $\Gamma_{fuel}$ and $\Gamma_{age}$, respectively.

$$J = \int_0^{t_f} \Gamma_{fuel} \dot{m}_f(x(t), u(t)) \, dt + \gamma \, \Gamma_{age} Q_{loss} (x(t_f), u(t_f)),$$

(40)

An additional binary variable $\gamma \in \{1, 0\}$ multiplies the battery aging term. It is introduced to compare the results between the same approach considering both the terms ($\gamma = 1$) and the fuel consumption only ($\gamma = 0$). Once the two terms are converted to the same unit, one could introduce a Pareto coefficient. In [26] a Pareto analysis on the offline problem is conducted. In this work we consider only the two discrete values of $\gamma$.

**B. OFFLINE OPTIMAL CONTROL**

The offline solution is obtained by solving the optimal control problem presented in IV-A on the time horizon $[0, t_f]$, with $t_f$ representing the final time of the driving datasets described in II-E. The speed profile is the actual velocity of the entire driving cycle. The cost function to be minimized is (40). The mild-hybrid electric vehicle studied in this work can recharge its battery only through regenerative braking. In order to guarantee constant fuel-saving performances during multiple road trips, the battery must never discharge completely. Typically, non-plugin hybrid vehicles feature a charge sustaining strategy in order to maintain the battery state of charge around 50%. To take into account the charge sustaining condition, we constrain the battery final SOC within the range $50\% - tol \leq SOC(t_f) \leq 50\% + tol$, with $tol = 2\%$.

**C. ONLINE OPTIMAL CONTROL**

In the receding horizon MPC, the optimal control problem is solved over a finite future horizon $[t_0, t_f]$ with $t_0 \ll t_f$ and $t_0 = 0$, $t_h = 5s$. The speed profile used in the horizon is the one predicted by the Bio-inspired Velocity prediction algorithm (described in II). The finite future horizon $[t_0, t_h]$ is discretized over a grid of $N_h = \frac{h-\delta_0}{\delta}$ intervals, where $t_0 \leq t_j = t_0 + j \delta_0 \leq t_h = t_0 + N_h \delta_0$ for $j = 0, \ldots, N_h$. The grid spacing $\delta_0$ is equal to 0.5s and the MPC solution is updated with a frequency $f_{MPC} = \frac{1}{\delta_0}$ once the numerical solution is obtained, only the first element of the control vector (i.e. $u(t_0)$) is applied to the vehicle driveline. Then the vehicle state is updated ($x(t_0)$), a new speed profile is predicted from the new state, and the optimization is re-computed for the horizon receding in front of the vehicle. A repeated online optimization allows having a continuously updated knowledge of the system’s state thanks to the measurement feedback.

1) **CHARGE SUSTAINING STRATEGY**

An a-priori knowledge of the complete driving cycle is not possible for the online case, thus we must enforce the charge-sustaining mode by adding a SOC-dependent term to the cost function in (40), i.e.

$$J = \int_0^{t_h} \Gamma_{fuel} \dot{m}_f(x(t), u(t)) \, dt + \gamma \, \Gamma_{age} Q_{loss} (x(t_h), u(t_h)) + \rho \left(SOC(t_h) - SOC_{trg}^f\right)^2,$$

(41)

where $\rho \leq 0$ is a weighting term that is selected based on the desired performance of the energy management strategy. The additional term is a quadratic penalty pushing the final state of charge $SOC(t_h)$ to be close to the target final state of charge $SOC_{trg}^f$. The parameter $SOC_{trg}^f$ and the weight $\rho$ must be tuned. In Section V-1 we tuned them to obtain a SOC value around 50% at the very end of each driving cycle. We recall that $\gamma \in \{0, 1\}$ is a binary variable that, when set to 0, excludes the aging term.

**D. ENERGY MANAGEMENT SYSTEM STRUCTURE**

The structure of the EMS proposed in this work falls within the canonical MPC scheme. Figure 9 shows a high-level representation with two main blocks, namely the velocity prediction and the online optimal control blocks. At each time step $t_i$, the Bio-Inspired velocity prediction algorithm (see Section II-C) estimates the velocity profile that the driver wants to pursue in the future time horizon $[t_i, t_i + t_h]$. The velocity profile is then used in the online optimal control problem (see Section IV-C) to find the optimal torque split solution between the engine and the motor over the time horizon $[t_i, t_i + t_h]$. The first values $u(t_0)$ of these controls are implemented by the motor and the engine. Then the system states are updated and the process is repeated. The actual velocity of the vehicle does not depend on the outcome of the optimal controls obtained, since the sum of the torques does not change. Therefore, the EMS optimizes the fuel consumption and the battery life without affecting the total torque request coming from the driver.

It is worth pointing out that the term ‘optimal’ for the online solution does not mean ‘global’ optimum. It only refers
to the fact that the solution is coming from the resolution of an optimal control problem. In fact, a direct approach is used to numerically solve it (see subsection IV-E) and the ‘global’ property of the solution cannot be guaranteed. Additionally, the online solution is ‘local’ because it cannot correspond to the reference off-line one, being the horizon much shorter than the length of the entire driving cycle. Moreover, the EMS uses the estimated speed profile, which differs from the actual one of the driving cycle. Therefore, the online solution is indeed a local optimum. Nevertheless, the MPC-based strategy here presented can approximate fairly well the baseline solution computed off-line as demonstrated in section VI and yields to satisfactory energy and battery life savings.

E. SOLUTION METHOD

Ideally, the optimal control problem defined in Section IV-B should be solved using the Differential Programming (DP) approach to guarantee the global optimum solution. However, the drive-line model adopted in this paper has 4 inputs and 7 states that together with the length of the planning horizon (several minutes) makes the problem numerically intractable with DP (the issue is known as the curse of dimensionality [83]). On the contrary, the direct approaches are numerically efficient, but they do not necessarily converge to the global optimum. To search for the best possible local optimum, which hopefully approximates the global solution, one could use the Monotonic Basin Hopping approach ([84]).

However, due to the smoothness of the differential system (32) the differences in the local solutions are quite small and the off-line solution, found with the direct method, can be regarded as a good (local) optimum, and thus used as a baseline for the online strategy.

For all of the above reasons, in this work both the online and the offline optimal control problems are solved using the direct approach and, specifically, the direct collocation method, which discretizes both the states and the controls.

The constant value \( t_s = \frac{N_s}{N} = 0.5 \) s is chosen as the spacing of the discretized time grid, the latter defined as:

\[
\mathcal{G}_N = \{t_0 < t_1 < \ldots < t_j < \ldots < t_N = t_f\}
\]

\[t_j = t_0 + j t_s, \quad j = 1, \ldots, N, \tag{42}\]

with \( N \) the number of grid points. Since \( t_s \) is fixed, the length of the discretized time length is a function of the horizon initial and final time instants, \( t_0 \) and \( t_f \) respectively. The Tustin method is used to discretize the differential equation system (32), leading to:

\[
x(t_{j+1}) = \left(II_{n_s} - \frac{t_s}{2}A\right)^{-1}\left(II_{n_s} + \frac{t_s}{2}A\right)x(t_j) + \left[\left(II_{n_s} - \frac{t_s}{2}A\right)^{-1}B\right]u\left(t_{j+\frac{1}{2}}\right),
\]

\[j = 0, \ldots, N - 1\] (43)

where \( II_{n_s} \) is the \( n_s \times n_s \) identity matrix. The continuous state-space matrices \( A \) and \( B \) are the ones shown in (32). The term \( u\left(t_{j+\frac{1}{2}}\right) = \frac{n_s + n_{s+1}}{2} \) represents the controls evaluated in the time-grid midpoints. We discretize the controls by means of piecewise constant functions so that the value of each control is constant between two grid points on the time-grid \( \mathcal{G}_N \). Finally, the constraints and the objective function (41) or (40) are discretized and evaluated on each point of \( \mathcal{G}_N \).

The discretization of the problem formulated in Section IV-A leads to a nonlinear programming problem (NLP), which is numerically solved with Ipopt, a state-of-the-art solver based on the interior-point algorithm [85]. Since the interior-point algorithm is a gradient-based method, the Jacobian of the constraints and of the gradient of the objective function must be supplied to Ipopt. The exact Hessian of the Lagrangian can be optionally supplied, otherwise Ipopt will numerically approximate it. The required analytical derivatives are computed using ADiGator [86], an algorithmic differentiator, which can run on Matlab. We solve both the offline and the online optimal control problems on Matlab 2019b running on a MacBook Pro equipped with a 2.7 GHz Intel i7 processor with 16 GB of memory. In the case of online optimal control, the average execution time of all the MPC steps is 0.157 seconds with a standard deviation of 0.11 seconds. As stated in Section IV-C, the optimization is executed every 0.5 seconds. Thus, the obtained computation time makes the whole algorithm suitable for a real-time implementation also for less powerful CPUs. In fact, it is also important to point out that the code was executed in script mode and a further computational time decrease by 5 to 10 times can be achieved when the code is translated into C++ and compiled. The gain in computational speed should be enough to compensate the (potential) use on less powerful architectures such as the ARM, which might be about 5 times slower than the i7 processor used to get the results in this work.

V. VALIDATION METHODOLOGY

In this section, our novel EMS scheme is validated by simulating the vehicle longitudinal dynamics and the powertrain and battery models with real driving data. As mentioned in the previous sections, our online EMS framework is composed of the prediction algorithm and by the receding horizon MPC. We hereby define three frameworks: one is our novel EMS framework, while the additional two will be employed as benchmarks.
**Bio-inspired Prediction (BP-MPC)** This approach is the proposed one. It uses the prediction algorithm proposed in II-C to estimate the future velocity profile and it solves the online optimal control problem described in Section IV-C. The length of the receding horizon is 5 seconds. Firstly, this framework is simulated employing the full cost function in (41). Secondly, the same framework is simulated without the aging term, i.e. by setting \( \gamma = 0 \) in (41).

**Exact Prediction (EP-MPC)** In this framework, we employ the same MPC procedure of BP-MPC. The difference lies in the velocity profile fed into the MPC. In this case, the future velocity profile is taken directly from the driving datasets, thus emulating a perfect velocity prediction. This framework has two purposes:

- to validate the choice of the prediction horizon length for the BP-MPC. We tested the EP-MPC with two horizon lengths of 5 seconds (EP-MPC-5S) and 10 seconds (EP-MPC-10S).
- to use the EP-MPC as a benchmark to evaluate the performance of the EMS when changing the length of the future prediction.

As in BP-MPC, all simulations of EP-MPC are performed with and without the aging term in (41).

**Offline (OF)** In this framework we solve the offline OCP problem formulated in Section IV-B. The optimal control problem is solved on the entire horizon (6 minutes) of each of the 30 naturalistic driving datasets. Again, the framework is simulated with and without the aging term, i.e., setting \( \gamma = 1 \) and \( \gamma = 0 \) in (40), respectively.

The naturalistic driving datasets used in the simulations were presented in Section II-E.

### 1) CHARGE SUSTAIN TUNING

In the frameworks featuring the MPC, we must tune the parameters \( \rho \) and \( \text{SOC}^\text{trg} \) present in the cost function (41). We want to maintain the final State Of Charge (SOC) around the initial value of 50% for two reasons. Firstly, we want to ensure charge sustainability, i.e. the battery must not tend to constantly or completely discharge during driving. Secondly, we want to keep the comparison between simulation results consistent. For instance, if one simulation ends with 50% of SOC and another one with 20%, the second simulation would have saved much more fuel, since it used some of the battery stored energy. Therefore, a comparison of the fuel saving results of the two simulations would not be fair.

The two parameters are manually tuned in order to obtain a final SOC value of about 50% on all the 30 driving datasets. In all the frameworks featuring MPC, the parameter \( \text{SOC}^\text{trg} \) is set to 60%. The weight \( \rho \) in the cost function (41) is set to 0.3 for the cases where the battery aging term is present (\( \gamma = 1 \)). It is set to 0.8 when the aging term is removed (\( \gamma = 0 \)).

### A. SIMULATION METHODOLOGY

We hereby detail how the driving datasets and the vehicle model are used to simulate the previously introduced frameworks. Let us start from the OF framework, since it requires a separate discussion. Indeed, this framework is closely related to the work in [26], which was validated on standard driving cycles. By making use of naturalistic driving data, the OF framework further validates the work in [26]. Most importantly, the results of the OF framework serve as a benchmark for the frameworks featuring the receding horizon MPC.

Finally, conversely to the MPC-based frameworks, the OF frameworks takes the entire velocity and acceleration profile of each dataset as inputs, and returns the solution of the offline OCP (see Section IV-B) on the entire dataset length.

The MPC-based frameworks instead, require the current velocity and acceleration information from the driving dataset at each simulation time step. To represent the overall data flow, Figure 10 is provided.

Firstly, for each simulation step \( i \) at the time instant \( t_i \), the current speed and acceleration \( v(t_i) \) and \( a(t_i) \) are used by the predictor to compute the future velocity profile \( \tilde{v}(t_i, \tau) \). The longitudinal dynamics equation (21) allows us to compute the total torque request profile \( \tilde{T}_w(t_i, \tau) \) from the predicted velocity and acceleration profiles. The first value of the predicted acceleration (and velocity) profile coincides with the current acceleration (and velocity) value of the dataset, i.e.

\[
\tilde{a}(t_i, 0) = a(t_i)
\]
\[
\tilde{v}(t_i, 0) = v(t_i).
\]  

(44)

For each time step, the values in (44) are used to compute the exact amount of instantaneous required total driving torque \( T_w(t_i) \) by means of Equation (21). By doing so, the longitudinal dynamics of the simulated powertrain is constrained to track the driving cycle from the dataset.

The input of the MPC is the future torque profile \( \tilde{T}_w(t_i, \tau) \). The MPC will optimally split the required total torque between the ICE, the electric motor, and the mechanical brakes on the receding horizon \( \tau \in [0, t_h] \) at each MPC step. The MPC recomputes the solution every 0.5 seconds.

### B. METRICS

In order to discuss the simulation results, we introduce some performance metrics. The first metric is the fuel saving percentage, expressed by

\[
\text{FSP} = 100 \cdot \left(1 - \frac{m_{f,\text{HYB}}}{m_{f,\text{ICE}}} \right),
\]  

(45)

where \( m_{f,\text{HYB}} \) is the fuel mass in Kg consumed by the hybrid vehicle. The quantity \( m_{f,\text{ICE}} \) is the fuel consumption that the same vehicle would need in the case only the internal combustion engine is used. The latter quantity is computed.
by simulating the longitudinal dynamics of the powertrain on the driving dataset while imposing $T_{mot}(t) = 0$ in Equation (20).

The second metric we consider is the relative capacity loss percentage of the battery, which is defined as

$$RCLP = 100 \cdot \left(1 - \frac{Q_{\text{loss}}^1}{Q_{\text{loss}}^0}\right),$$

(46)

where $Q_{\text{loss}}^{0}$ is the final capacity loss in the framework cases where capacity loss is not minimized, and $Q_{\text{loss}}^1$ where capacity loss is minimized. The RCLP metric can be seen as the percentage of the capacity of the battery that is saved on a driving set by introducing the aging term in the cost function, i.e. minimizing the battery capacity loss. The metrics are collected for all the frameworks: BP-MPC, EP-MPC-5s, EP-MPC-10s and OF. Every framework is simulated with and without the capacity loss minimization ($\gamma = 1$ and $\gamma = 0$ respectively) for a total of 8 frameworks.

**VI. RESULTS**

In this section we present the results obtained applying the simulation methodology and the metrics explained in subsections V-A and V-B. In Figure 11 the fuel efficiency (absolute and relative) is shown for each driver segment and for each framework, considering the presence of the aging term (setting $\gamma = 1$). It is expressed in km/l to be more readable. The fuel consumption is computed again for each framework removing the aging term, setting $\gamma = 0$, and the fuel consumption results for each framework are shown in Figure 12. By looking at these two figures we can make some qualitative considerations. Firstly, we can notice in both plots that the knowledge of a longer future velocity profile improves the fuel efficiency. Secondly, the dependency of the fuel efficiency on the knowledge of the future velocity is emphasized by the addition of the aging term, i.e. when also the battery aging is minimized in the optimization. Indeed, in Figure 11 the MPC frameworks fuel efficiencies (BP-MPC, EP-MPC5SS and EP-MPC10S) tend to be farther of the corresponding OF fuel efficiencies when compared with the results without aging minimization of Figure 12.

**TABLE 5. Average fuel savings with respect to the ICE-only case.**

| Aging term | BP-MPC | EP-MPC-5S | EP-MPC-10S | OF   |
|------------|--------|-----------|------------|------|
| $\gamma = 1$ | 8.16%  | 8.56%     | 8.51%      | 10.03% |
| $\gamma = 0$ | 8.43%  | 8.92%     | 8.98%      | 10.19% |

Next we analyze the average results in terms of fuel saving in order to draw quantitative conclusions about the performance of the system. The average fuel saving percentage (FSP) among all the 30 driving datasets is computed for each of the presented frameworks and the results are shown in Table 5. We recall that the FSP is computed by means of Equation (45). Let us start from the results that take into account the battery aging effect (first row of Table 5). As expected, the fuel consumption increases as we get farther away from the ideal case provided by the OF framework. The highest gap lies between the OF and the EP-MPC-10S, where about 15% of the potential fuel saving is lost. Most of the fuel saving loss occurs when passing from the knowledge of the velocity profile on the entire horizon (as in OF) to the receding horizon MPC frameworks. Conversely, the fuel saving performances of EP-MPC-10S and EP-MPC-5S are identical. We use the latter result to validate the choice of setting the horizon length of the velocity prediction in the BP-MPC to 5 seconds. Indeed, increasing the prediction horizon up to 10 seconds in the BP-MPC framework would not improve the fuel savings, since no improvement happens even with the exact predictions of EP-MPC-10S and EP-MPC-5S. The average fuel saving of the BP-MPC framework is slightly lower than the EP-MPC-5S framework (which uses the same horizon length, but perfect prediction), showing that the bio-inspired velocity prediction is suitable to feed the online EMS. Finally, in average the proposed BP-MPC framework is able to achieve the 81% of the potential fuel saving of the benchmark OF framework. Moving to the second line of Table 5, we can see the average fuel savings obtained when in the frameworks the battery aging is not minimized in the cost function. As expected, in all four frameworks, the fuel saving is higher but the difference is small with respect to their respective battery-aging aware cases. We can conclude that minimizing the battery aging does not affect significantly the fuel savings performances. This conclusion was also drawn in [26] for the offline case only (OF), on standard driving cycles: here, we extend such conclusion on naturalistic driving data and also on online MPC frameworks. In other words, we prove that the results of [26] are equally valid in the real-time application.

The second metric regards the relative battery capacity loss percentage, which is defined by Equation (46). In contrast to the RCLP, we hereby define the absolute capacity loss as the percentage of the battery capacity loss w.r.t. the total capacity of the battery. In Figure 13 the absolute and relative capacity loss is shown for each driver segment and for each framework when the battery-aging minimisation is active (setting $\gamma = 1$), while in Figure 14 the capacity loss is shown for all the frameworks without aging minimisation (setting $\gamma = 0$). From the relative capacity loss plots of Figure 13 and from...
the average results of the RCLP metric (see subsection V-B) we can draw some general considerations. Table 6 shows the average RCLP, i.e. the average percentage of the capacity of the battery that is saved introducing the aging term in the cost function. For each framework, the average percentage of the relative capacity loss is taken among all the 30 driving datasets. The battery aging-minimization performances worsen significantly when the knowledge of the future horizon shortens. Indeed, the best results are obtained when all the velocity horizon is known, i.e. the case of the OF framework. From the benchmark framework (OF) to the BP-MPC, only about 30% of the potential capacity saving is achieved. Comparing BP-MPC with its perfect velocity prediction version (EP-MPC-5S), the performance decreases by 50%. This performance loss is due only to the errors in the velocity prediction, since the horizon length is the same.

These are remarkable results: in the real-time application the prediction horizon length has a more important effect on battery aging than on fuel consumption. Additionally, an accurate speed profile prediction is critical for battery aging minimization.

In Figure 15 we show the fuel consumption map of the engine, while the motor efficiency map is drawn in Figure 16. On these figures, plots of the engine and motor usage are shown with a color map for the BP-MPC and OF frameworks. The plots of each framework represent the overall usage on
all 30 naturalistic driving datasets. Notice that the difference between online (MPC) and offline cases is barely visible. The engine modeled in this work is quite powerful, thus it is under-utilized during normal driving. On the other hand, the much less powerful electric motor is fully exploited, especially in high efficiency regions.

The evolution of the SOC of the battery in time allows us to make some considerations regarding the charge sustaining and the performance in general. In Figure 17, for all the datasets, the maximum, the minimum, and the median of the SOC at every time instant are plotted for BP-MPC (in green) and OF (in violet) frameworks. From the BP-MPC SOC range in time, we can conclude that the charge sustaining strategy is effective, since the minimum SOC is bounded above 47% even with different driving styles. The OF framework, thanks to the knowledge of the entire dataset, can exploit a wider range of the SOC without affecting the final SOC. The violet area in Figure 17 ranges down to 40% while the green one is maintained over the value of 47% by the necessary charge sustaining strategy. The impossibility for the BP-MPC framework to exploit lower values of the SOC has a double effect on the battery. Firstly, higher SOC values correspond to higher battery aging (see eqnarray (38)), resulting in a loss of capacity-saving performance between offline and online frameworks. Secondly, the BP-MPC framework frequently cannot fully recover energy from deceleration, resulting in a loss of performance in terms of fuel saving.

Finally, in order to validate the effectiveness of the charge sustaining strategy, Figure 18 shows the distribution of the
final SOC of the BP-MPC framework (considering all the 30 datasets). All the values of the final SOC of the BP-MPC framework are included in the window 47%-52%. Similar results are obtained for all the simulations of the MPC-based frameworks, which are also reported in Figure 18.

VII. CONCLUSION
In this paper, we presented a full energy management system for hybrid electric vehicles. Firstly, we designed a prediction scheme able to estimate 5 seconds of the future velocity profile the driver intends to pursue. Then, we presented an online energy management system able to optimally split the requested driving torque between the electric motor and the internal combustion engine. The EMS is based on a state-of-the-art model of the battery; it takes into account the battery aging in the form of capacity loss and it is implemented by means of optimal control solved through the direct approach. We validated the prediction scheme and then the entire online EMS framework (BP-MPC) on 30 naturalistic driving segments of 6 minutes each from 10 different drivers. The prediction scheme was also validated on standard driving cycles and compared against two benchmark prediction methods which are widely employed in the literature. The proposed EMS system was then compared with three ideal benchmark (one “global” solution (OF) and two MPC-scheme fed markups (one “global” solution (OF) and two MPC-scheme fed predictions). These information, typically available from ADAS applications, would allow to extend the set of affordances of the bio-inspired prediction scheme and increase the ability of the algorithm to predict the actual driver’s intended maneuver.

Despite the latter result, the real-time system was able to extend the battery life, making it suitable for an implementation. Increasing the battery life up to the life of the vehicle itself may significantly reduce the cost for the final user and additionally reduce the environmental impact of the battery.

The approach can also be modified by employing different costs in the objective function, for instance an equivalent environmental impact of both fuel and battery capacity loss.

APPENDIX A
BATTERY DYNAMICS
In (47), (48), (49) we present the matrices of the state-space realization (32). Only the nonzero elements of the state-space matrices are shown.

\[
A(1, 1) = -\frac{189D_p^0}{[R_c^0]^2}
\]
\[
A(1, 2) = -\frac{3465[D_e^0]^2}{[R_c^0]^4}
\]
\[
A(1, 5) = \frac{189D_s^0(\theta_p^{0%} - \theta_p^{0%})c_{s,max}}{[R_c^0]^2}
\]
\[
A(2, 1) = 1
\]
\[
A(2, 5) = -(\theta_p^{0%} - \theta_p^{0%})c_{s,max}
\]
\[
A(3, 3) = -\frac{189D_n^0}{[R_c^0]^2}
\]
\[
A(3, 4) = \frac{3465[D_e^0]^2}{[R_c^0]^4}
\]
\[
A(3, 5) = -\frac{189D_s^n(\theta_p^{0%} - \theta_p^{0%})c_{s,max}^n}{[R_c^n]^2}
\]
\[
A(4, 3) = 1
\]
\[
A(4, 5) = (\theta_p^{0%} - \theta_p^{0%})c_{s,max}^n
\]
\[
A(6, 6) = -\frac{9.8710D_s^L}{L^2}
\]
\[
A(7, 7) = -\frac{9.5842D_s^L}{L^2}
\]
\[
B(1, 1) = \frac{\varepsilon_p F_{AP} L^5}{15 D^p_{s} \varepsilon_p F_{AP} L^5}
\]
\[
B(2, 1) = -\frac{\varepsilon_n F_{AP} L^5}{7}
\]
\[
B(3, 1) = \frac{\varepsilon_n F_{AP} L^5}{7}
\]
\[
B(4, 1) = -\frac{\varepsilon_n F_{AP} L^5}{15 D^n_{s} \varepsilon_n F_{AP} L^5}
\]
\[
B(5, 1) = -\frac{1}{Q_N}
\]
\[
B(6, 1) = \frac{3.1463(n_0^+ - 1)}{\varepsilon_c LFA_p}
\]
\[ B \left( 7, 1 \right) = 2.9351 \left( \frac{V_0}{e} \right)^{1 - 1} e^{\frac{LFA}{n}} \]  

\[ C \left( 1, 1 \right) = 1 \]  

\[ C \left( 2, 3 \right) = -1 \]  

\[ C \left( 3, 5 \right) = 1 \]  

\[ C \left( 4, 6 \right) \]  

\[ C \left( 5, 7 \right) = 1 \]  

REFERENCES

[1] J. Fleming, X. Yan, C. Allison, N. Stanton, and R. Lot, “Real-time predictive eco-driving assistance considering road geometry and long-range radar measurements,” IET Intell. Transp. Syst., vol. 15, no. 4, pp. 573–583, Feb. 2021.

[2] N. Xu, Y. Kong, L. Chu, H. Ju, Z. Yang, Z. Xu, and Z. Xu, “Towards a smarter energy management system for hybrid vehicles: A comprehensive review of control strategies,” Appl. Sci., vol. 9, no. 10, p. 2026, May 2019.

[3] A. Biswas and A. Emadi, “Energy management systems for electrified powertrains: State-of-the-art review and future trends,” IEEE Trans. Veh. Technol., vol. 68, no. 7, pp. 6453–6467, Jul. 2019.

[4] D.-D. Tran, M. Vafeaipour, M. E. Baghdadi, R. Barrero, J. Van Mierlo, and O. Hegazy, “Thorough state-of-the-art analysis of electric and hybrid vehicle powertrains: Topologies and integrated energy management strategies,” Renew. Sustain. Energy Rev., vol. 119, Mar. 2020, Art. no. 109596.

[5] J. P. Torreglosa, P. Garcia-Triviño, D. Vera, and D. A. López-García, “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?” Appl. Sci., vol. 10, no. 23, p. 8744, Dec. 2020.

[6] Z. Chen, Y. Liu, M. Ye, Y. Zhang, Z. Chen, and G. Li, “A survey on key techniques and development perspectives of equivalent consumption minimization strategy for hybrid electric vehicles,” Renew. Sustain. Energy Rev., vol. 151, Nov. 2021, Art. no. 116107. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032121008832

[7] T. Liu, W. Tan, X. Tang, J. Zhang, Y. Xing, and D. Cao, “Driving conditions-driven energy management strategies for hybrid electric vehicles: A review,” Renew. Sustain. Energy Rev., vol. 151, Nov. 2021, Art. no. 115121. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032121007991

[8] B. Zhou, J. B. Burl, and A. Rezaei, “Equivalent consumption minimization strategy with consideration of battery aging for parallel hybrid electric vehicles,” IEEE Access, vol. 8, pp. 204770–204781, 2020.

[9] N. Kim and A. P. Rousseau, “Comparison between rule-based and instantaneous optimization for a single-mode, power-split HEV,” SAE, Warren, PA, USA, Tech. Rep., 2011.

[10] T. Hofman, R. M. van Druten, A. F. A. Serraens, and M. Steinbuch, “Rule-based energy management strategies for hybrid vehicles,” Int. J. Electr. Hybrid Veh., vol. 1, no. 1, pp. 71–94, 2007.

[11] C. Musardo, G. Rizzoni, Y. Guezennec, and B. Staccia, “A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management,” Eur. J. Control, vol. 1, no. 1, pp. 71–94, 2007.

[12] S. Onori, L. Serrao, and G. Rizzoni, Hybrid Electric Vehicles, Energy Rev., vol. 10, no. 23, p. 8744, Dec. 2020.

[13] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 62, no. 2, pp. 1805–1824, 2000, doi: 10.1103/PhysRevE.62.1805.

[14] J. Jing, D. Filev, A. Kurt, E. Özatay, J. Michelini, and Ü. Özgüner, “Vehicle speed prediction for driver assistance systems,” SAE, Warrendale, PA, USA, Tech. Rep., 2004.

[15] A. Rezaei and J. B. Burl, “Prediction of vehicle velocity for model predictive control,” IFAC-PapersOnLine, vol. 48, no. 15, pp. 257–262, 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896315019138

[16] J. Ziegmann, F. Denk, U. Vogeley, and C. Endisch, “Stochastic driver behavior modeling and stochastic control,” in Proc. 18th Int. Conf. Intell. Transp. Syst., Sep. 2015, pp. 2861–2868. [Online]. Available: http://ieeexplore.ieee.org/document/7313552/

[17] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 62, no. 2, pp. 1805–1824, 2000, doi: 10.1103/PhysRevE.62.1805.

[18] C. Zhai, F. Luo, and Y. Liu, “Vehicle speed prediction in a convex using V2V communication,” in Proc. IEEE 18th Int. Conf. Intell. Transp. Syst., Sep. 2015, pp. 2861–2868. [Online]. Available: http://ieeexplore.ieee.org/document/7995827/

[19] J. Ziegmann, J. Shi, T. Schnörer, and C. Endisch, “Analysis of vehicle speed prediction of electrified powertrain for look ahead control,” Appl. Energy, vol. 279, Dec. 2020, Art. no. 115903. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306261920313684

[20] C. Zhai, F. Luo, and Y. Liu, “A novel predictive energy management strategy for electric vehicles based on velocity prediction,” IEEE Trans. Veh. Technol., vol. 69, no. 11, pp. 12559–12569, Sep. 2020.

[21] J. Ziegmann, J. Shi, T. Schnörer, and C. Endisch, “Analysis of individual driver velocity prediction using data-driven driver model with environmental factors,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2017, pp. 517–522. [Online]. Available: https://ieeexplore.ieee.org/document/7995827/

[22] J. Ziegmann, F. Denk, U. Vogeley, and C. Endisch, “Stochastic driver velocity prediction with environmental features on naturalistic driving data,” in Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2018, pp. 1805–1814. [Online]. Available: https://ieeexplore.ieee.org/document/8560767/
G. Valenti et al.: Battery Aging-Aware Online Optimal Control: Energy Management System
[80] T. R. Tanim, C. D. Rahn, and C.-Y. Wang, “A temperature dependent, single particle, lithium ion cell model including electrolyte diffusion,” J. Dyn. Syst., Meas., Control, vol. 137, no. 1, Jan. 2015, Art. no. 011005.

[81] L. O. Valoøen and J. N. Reimers, “Transport properties of LiPF$_6$-based Li-ion battery electrolytes,” J. Electrochem. Soc., vol. 152, no. 5, p. A882, 2005.

[82] X. Jin, A. Vora, V. Hoshing, T. Saha, G. Shaver, R. E. García, O. Wasynczuk, and S. Varigonda, “Physically-based reduced-order capacity loss model for graphite anodes in Li-ion battery cells,” J. Power Sources, vol. 342, pp. 750–761, Feb. 2017.

[83] F. Biral, E. Bertolazzi, and P. Bosetti, “Notes on numerical methods for solving optimal control problems,” IEEJ J. Ind. Appl., vol. 5, no. 2, pp. 154–166, 2016.

[84] R. H. Leary, “Global optimization on funneling landscapes,” J. Global Optim., vol. 18, no. 4, pp. 367–383, 2000.

[85] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” Math. Program., vol. 106, no. 1, pp. 25–57, May 2006.

[86] M. J. Weinstein, M. A. Patterson, and A. V. Rao, “Utilizing the algorithmic differentiation package ADiGator for solving optimal control problems using direct collocation,” in Proc. AIAA Guid., Navigat., Control Conf., Jan. 2015, p. 1085.

GIAMMARCO VALENTI received the M.S. degree in mechatronics engineering from the University of Trento, Italy, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Industrial Engineering. He has been a Visiting Student at Universität der Bundeswehr München, and he worked in the EU research project SAFESTRIP, developing bio-inspired ADAS in a novel intelligent road infrastructure. His research interests include the modeling and prediction of human drivers, ADAS, and connected autonomous vehicles.

EDOARDO PAGOT received the M.S. degree in mechatronics engineering from the University of Trento, Italy, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Industrial Engineering. His research interests include multi-body dynamics modeling and optimal control (online and offline) in the field of vehicle dynamics, for both intelligent transportation and motorsport sectors.

LUCA DE PASCALI received the Ph.D. degree in mechatronics engineering from the University of Trento, Italy, in 2019, with a thesis on modeling, optimization, and control of hybrid powertrains. During his studies, he has been a Visiting Student at Technische Universität München, Germany, and a Visiting Scholar at the Clemson University International Center for Automotive Research, USA. He is currently working at Dana Incorporated on the development of electric and electric-hybrid powertrain systems for off-highway vehicles.

FRANCESCO BIRAL received the master’s degree in mechanical engineering from the University of Padova, Italy, and the Ph.D. degree in mechanics and machine theory from the University of Brescia, Italy, in 2000, for his work on minimum lap time of racing vehicles with the use of optimal control. He is currently an Associate Professor with the Department of Industrial Engineering, University of Trento. His research interests include symbolic and numerical multi-body dynamics and optimization, and constrained optimal control, mainly in the field of vehicle dynamics with special focus on intelligent vehicles and optimal maneuver for racing vehicles. He has 15 years of experience in the development and validation of ADAS and AD functions, both for cars and PTWs, gained in several European and industrial funded research projects.