Neural adaptive MPC with online metaheuristic tuning for power management in fuel cell hybrid electric vehicles

Neural Adaptive MPC with Online Metaheuristic Tuning for Power Management in Fuel Cell Hybrid Electric Vehicles

Lorenzo Calogero, *Graduate Student Member, IEEE*, Michele Pagone, *Member, IEEE*, Francesco Cianflone, Edoardo Gandino, Carlo Karam, and Alessandro Rizzo, *Senior Member, IEEE*

***Abstract*—In this paper, we present an advanced control framework for power management applications, named Neural Adaptive Model Predictive Control (NA-MPC), designed to provide an optimal power allocation among multiple energy sources, perform a multi-objective online adaptation of the optimal control policy, and ensure a fast real-time execution with low computational demand. NA-MPC augments general MPC problems with three key features: 1) an online metaheuristic tuning strategy adapts the MPC cost function weights, to attain multiple concurrent control objectives at once; 2) through neural emulation, the MPC control policy is replaced by an equivalent neural MPC controller, exhibiting universal approximation guarantees and ensuring real-time feasibility; 3) a neural black-box MPC prediction model is employed, identified only via noise-corrupted input-output measurements from the plant, which is assumed to be unknown. The general formulation and versatility of NA-MPC make it potentially applicable to several power management scenarios; in this work, we apply NA-MPC to the case study of power management in fuel cell hybrid electric vehicles (FCHEVs), a topic of growing interest within the frame of sustainable transportation, for which novel and efficient strategies are still lacking. The effectiveness of NA-MPC is thoroughly assessed via numerical simulations, demonstrating its capability to optimally attain multiple control objectives concurrently in real time; moreover, NA-MPC consistently outperforms the most prominent state-of-the-art HEV power management strategies.**

***Note to Practitioners*—The aim of this paper is to introduce an advanced online-adaptive optimal control strategy, named NA-MPC, and employ it as a novel power management strategy for FCHEVs, with the purpose of addressing several technical shortcomings of the existing state-of-the-art strategies. Specifically, the latter typically fail in performing effective trade-offs between accurate power tracking and supply consumption, proving a merely suboptimal control action. Such strategies have also very limited adaptation capabilities, being either offline-tuned or employing simple non-optimal adaptation policies. Moreover, only few basic optimal control strategies are proposed in the literature, with little focus on their real-time feasibility. By contrary, our NA-MPC strategy provides an optimal power allocation, effectively attains multiple concurrent control objectives, and, thanks to its neural embedding, is real-time feasible and easily implementable on hardware with limited computational resources. Furthermore, the general formulation and versatility of NA-MPC enable its potential application across a wide variety of different power management scenarios.**

***Index Terms*—Model predictive control, neural networks, metaheuristic optimization, hybrid vehicles, energy management.**

The work of L. Calogero was supported by the NGEU-PNRR Project (MUR - D.M. 352/2022). The work of M. Pagone and A. Rizzo was supported by the MOST (Sustainable Mobility National Research Center) and funded by the European Union NextGenerationEU Project (Piano Nazionale di Ripresa e Resilienza - Mission 4, Component 2, Investment 1.4 - D.D. 1033 17/06/2022) under Grant CN00000023. *(Corresponding author: A. Rizzo.)*

L. Calogero, M. Pagone, and A. Rizzo are with the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy (e-mail: {lorenzo.calogero, michele.pagone, alessandro.rizzo}@polito.it).

F. Cianflone, E. Gandino, and C. Karam are with Dumarey Softronix S.r.l., Turin, Italy (e-mail: {francesco.cianflone, edoardo.gandino, carlo.karam}@dumarey.com).

I. Introduction

HYBRID electric vehicles (HEVs) have established as a new paradigm in transportation, driven by the rapid shift of automotive industry towards sustainable alternatives to internal combustion engines. In recent years, fuel cell hybrid electric vehicles (FCHEVs), which integrate hydrogen fuel cells (FCs), electric batteries, and other supplementary energy sources (e.g., supercapacitors [1]), have gained prominence, thanks to their enhanced autonomy and capability to withstand high power demands [1]–[3]. As a response to this growing interest in hybrid transportation, novel and efficient strategies for energy management are becoming essential to enable such a new transportation paradigm.

To fulfill this requirement, this research effort borrows from the established framework of optimal control to address the challenges associated with power management. In this context, we introduce a novel control strategy, termed Neural Adaptive Model Predictive Control (NA-MPC).

The NA-MPC strategy is formulated starting from a general MPC optimal control problem and augments it with three key additional features (Figure 1):

1) To achieve multiple, potentially concurrent, control objectives at once, the weights of the MPC cost function are adapted in real time by an online metaheuristic tuning strategy, which minimizes a set of performance-metric functions, each one encoding a different control objective.
2) To ensure real-time feasibility and low computational demand, the MPC control policy is replaced by a neural emulation of it, yielding a neural MPC controller.
3) For the sake of applicability in real-world scenarios, a neural black-box MPC prediction model is employed, identified only via noise-corrupted input-output measurements from the plant to control, which is assumed to be unknown.

A depiction of the NA-MPC control system scheme is reported in Figure 1. In this work, we tailor the NA-MPC framework for the task of power management, consisting of finding

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3534402

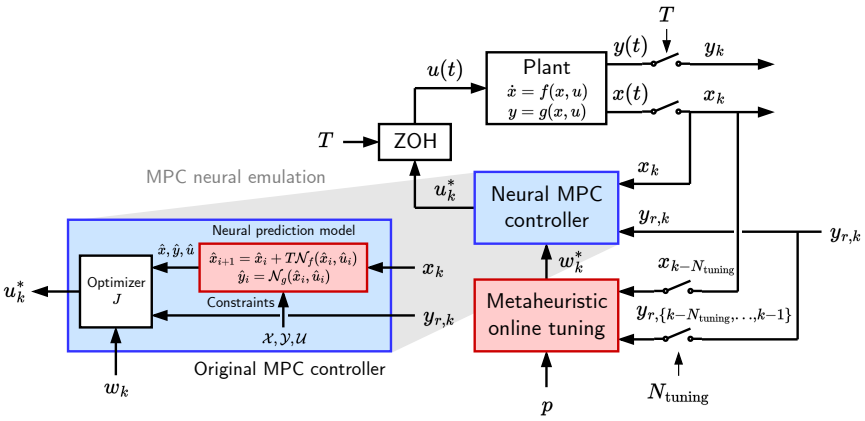

Fig. 1. Neural Adaptive MPC (NA-MPC) control system scheme, comprising: the neural black-box MPC prediction model (Sec. III-A and III-B); the neural MPC controller (Sec. III-C); the online metaheuristic tuning policy (Sec. IV).

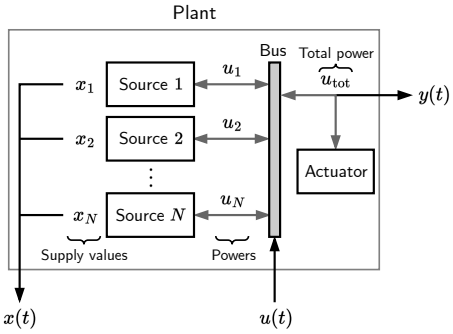

Fig. 2. General plant structure for power management applications.

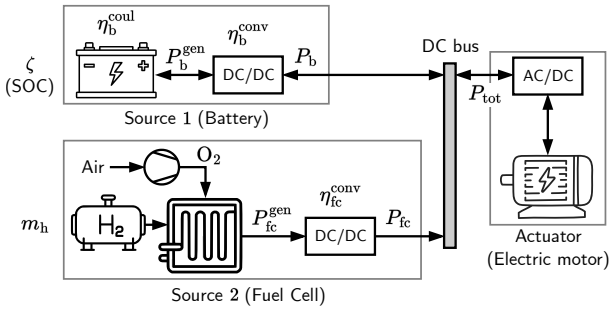

Fig. 3. FCHEV power supplies and distribution system (Sec. V).

the optimal power allocation among multiple energy sources driving a shared actuator (Figure 2). The immediate focus in this study are FCHEVs (Figure 3); still, several other potential applications can be found within the domain of power management (namely, different hybrid powertrain configurations, microgrids with energy storage systems, HVAC systems, etc.).

In the context of FCHEV power management, the MPC optimal control problem is tasked to find the optimal allocation between battery power and FC power, while predicting the consumption of supplies over time, i.e., battery state of charge (SOC) and hydrogen mass. As MPC prediction model, a black-box neural networks-based model of the FCHEV power sources and distribution system is employed; such a model is identified only via noise-corrupted input-output measurements collected from a high-fidelity white-box model.

As mentioned above, NA-MPC is capable to optimally achieve multiple control objectives concurrently, by means of an online metaheuristic tuning strategy, adapting the MPC cost function weights in real time. For FCHEV power management, such objectives can be accurate power tracking, consumption minimization, distinct consumption priority for each supply, etc. In more detail, at fixed time intervals, the tuning strategy simulates back the closed-loop MPC control action over the previous time interval for a given set of weights; such weights are updated by a metaheuristic optimization policy, which finds the optimal weights minimizing the set of performance-metric functions associated to the current control objectives. Such optimal weights are then employed during the next time interval, at the end of which a new tuning is initiated.

Concerning instead the neural emulation of the MPC control policy, the neural MPC controller employed by NA-MPC is realized with a feedforward neural network (FNN), which takes as inputs all the parameters of the original MPC optimal control problem (including the MPC cost function weights) and provides as output the optimal control input. In this regard, we provide theoretical guarantees on the smoothness of the MPC policy so that the universal approximation theorem of FNNs [4] holds for the MPC neural emulation. Unlike solving the original MPC problem, evaluating its neural approximation requires near-zero computation time, ensuring the real-time feasibility of the NA-MPC strategy, even in case of limited computational resources.

Thanks to its capabilities, our NA-MPC strategy represents an advancement in the state of the art on power management strategies and, particularly, within the domain of HEVs. At present, several investigations have delved into HEV energy management, delineating two main approaches: rule-based control and optimization-based control [1]. For such two approaches, the most prominent strategies are Fuzzy-Logic Control (FLC) [5]–[7] and Equivalent Consumption Minimization Strategy (ECMS) [8]–[10], respectively. FLC classifies power and supply values into qualitative ranges; then, power allocation is performed by means of logic implications on such qualitative values. ECMS, instead, seeks the optimal power values by minimizing, at each time instant, the instantaneous supply rates. FLC bears the hallmarks of simplicity, interpretability, and fast execution, while ECMS has the advantage of providing a locally optimal power allocation. Nevertheless, both FLC and ECMS have significant shortcomings with respect to our NA-MPC strategy: FLC and ECMS do not perform any prediction, thus they cannot handle constraints on the supply values; due to their formulation, they are inherently unable to perform an effective trade-off between tracking accuracy and supply saving; also, so far, only few and non-optimal adaptive strategies have been developed for FLC and ECMS. Within the optimization-based approaches for power management, also MPC has been investigated by several works. Most of these studies, however, focus on conventional HEVs, thus not including FCs as a power source [11], [12]. While some works have proposed

MPC strategies that augment the power management task with additional features (e.g., vehicle speed prediction [13], vehicle mass online estimation [14], etc.), all of them mainly focus only on requested power tracking and charge sustainment, not including additional control objectives. Also, in most cases, the MPC problem formulation is merely an extension of the ECMS policy over a certain time horizon. Moreover, little focus is given to the MPC computational demand to assess real-time feasibility [15].

The effectiveness of our NA-MPC strategy is validated through an extensive simulation campaign. First, we assess the capability of NA-MPC to effectively achieve multiple control objectives concurrently (among which accurate power tracking, consumption minimization, and distinct consumption priority for each supply). Second, the NA-MPC online metaheuristic tuning strategy is compared with an "ideal" tuning policy that assumes to know the whole power request in advance. Third, NA-MPC is compared with the most prominent state-of-the-art power management strategies, i.e., FLC [6], [7] and ECMS with online adaptation [8]. Simulations demonstrate the capability of NA-MPC in successfully attaining all the given control objectives. The NA-MPC online tuning strategy proves its effectiveness by achieving very similar results with respect to the ideal tuning policy. Also, NA-MPC outperforms both FLC and ECMS, consistently achieving the given control task with better performance.

Outline: The remainder of the paper is structured as follows. In Section II, we introduce the MPC framework. In Sections III and IV, we detail the construction of the NA-MPC controller, including the neural black-box MPC prediction model identification, the MPC neural emulation with universal approximation guarantees, and the online metaheuristic tuning strategy for the MPC weights. In Section V, we tailor the NA-MPC strategy for FCHEV power management, also outlining the construction of a high-fidelity white-box plant model. Section VI validates the NA-MPC strategy through extensive simulations and comparisons with state-of-the-art techniques. Our conclusions are drawn in Section VII, along with perspectives for future research avenues.

Notation: In the following, $x = [x_i]_{i=1}^n$ denotes the vector $x \in \mathbb{R}^n$ with components x_i; $A = [c_i]_{i=1}^n$ and $A = [r_i^\top]_{i=1}^m$ denote the partition of matrix $A \in \mathbb{R}^{m\times n}$ by columns $c_i \in \mathbb{R}^m$ and rows $r_i \in \mathbb{R}^n$, respectively; for a vector $x \in \mathbb{R}^n$, $(x)_I$ denotes the vector collecting the components of x indexed by the set $I \subset \{1,\dots,n\}$; for vectors $x, y \in \mathbb{R}^n$, any equality or inequality relation $x \lessgtr y$ is considered component-wise, i.e., $(x)_i \lessgtr (y)_i$, $\forall\, i \in \{1,\dots,n\}$.

II. Model Predictive Control Formulation

Let us consider the power system depicted in Figure 2, which can be modeled as a continuous-time (CT) dynamical system, evolving on the state-output manifold $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$, as follows:

$$\dot{x} = f_c(x,u), \quad y = g(x,u), \quad x(0) = x_0, \tag{1}$$

where $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ are the state, output, and input vectors, respectively. For system (1), the following regularity assumptions are considered:

Assumption 1. *a) $f_c(x,\cdot)$ is continuous and $f_c(\cdot,u)$ is Lipschitz continuous; b) g is continuous.*

Assumption 2. *The input signal $u(t) \in \mathscr{U}$, where $\mathscr{U}$ is the space of piecewise continuous signals.*

Assumption 3. *Full information on the system state x is available for all $t \geq 0$.*

Assumption 4. *The sets $\mathcal{X}$, $\mathcal{U}$, $\mathcal{Y}$ are convex and compact polytopes, i.e., $\mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\}$, $\mathcal{U} = \{u \in \mathbb{R}^{n_u} : H_u u \leq h_u\}$, $\mathcal{Y} = \{y \in \mathbb{R}^{n_y} : H_y y \leq h_y\}$.*

Remark 1. *By Assumption 1a, the Picard-Lindelöf theorem guarantees the global existence and uniqueness of the solution $(x(t), y(t))$ of system (1) [16].*

Remark 2. *By Assumption 3, without loss of generality, we can set $(y)_{\{1,\dots,n_x\}} = x$, i.e., the output vector y contains the state vector x; then, by Assumption 4, the state constraints defined by $\mathcal{X}$ become redundant, being already included in $\mathcal{Y}$.*

Let us now consider system (1) as a CT plant to be controlled using Model Predictive Control (MPC).

By considering a control signal $u(t)$ for plant (1) that is constant over each time interval $[kT_s, (k+1)T_s]$, $k \in \mathbb{Z}_{\geq 0}$ (thus, satisfying Assumption 2), where T_s is the discrete time step, we can control the CT plant (1) with a discrete-time (DT) control policy. Specifically, we can compute a discretization of plant (1) as follows:

$$x_{k+1} = x_k + \int_{kT_s}^{(k+1)T_s} f_c(x(\tau), u_k) d\tau \approx f(x_k, u_k), \tag{2a}$$

$$y_k = g(x_k, u_k), \tag{2b}$$

where $x_k = x(kT_s)$, $u_k = u(kT_s)$, $y_k = y(kT_s)$, and f is obtained by approximating the integral in Eq. (2a) with some discretization method of choice; by employing the forward Euler method, we obtain

$$x_{k+1} = f(x_k, u_k) = x_k + T_s f_c(x_k, u_k), \tag{3a}$$

$$y_k = g(x_k, u_k). \tag{3b}$$

We formulate the MPC optimal control problem as follows, for each $k \geq 0$:

$$\min_{\hat{u},\hat{y},\hat{x}} \quad J(\hat{u},\hat{y}),$$

$$J(\hat{u},\hat{y}) = \sum_{i=0}^{N_p-1} \ell_i(\hat{u},\hat{y}) + \sum_{i=1}^{N_p-1} \ell_{\Delta,i}(\hat{u},\hat{y}), \tag{4a}$$

$$\ell_i(\hat{u},\hat{y}) = \|\hat{y}_i - y_{r,k}\|^2_{Q_k} + \|\hat{u}_i\|^2_{R_k},$$

$$\ell_{\Delta,i}(\hat{u},\hat{y}) = \|\hat{y}_i - \hat{y}_{i-1}\|^2_{Q_{\Delta,k}} + \|\hat{u}_i - \hat{u}_{i-1}\|^2_{R_{\Delta,k}}$$

$$\text{s.t.} \quad \forall\, i = 0,\dots,N_p-1,$$

$$\hat{x}_0 = x_k, \quad \hat{x}_{i+1} = f(\hat{x}_i, \hat{u}_i), \quad \hat{y}_i = g(\hat{x}_i, \hat{u}_i), \tag{4b}$$

$$H_u \hat{u}_i \leq h_u, \quad H_y \hat{y}_i \leq h_y. \tag{4c}$$

In Eq. (4), $\hat{u} = \{\hat{u}_i\}_{i=0}^{N_p-1}$, $\hat{y} = \{\hat{y}_i\}_{i=0}^{N_p-1}$, and $\hat{x} = \{\hat{x}_i\}_{i=0}^{N_p}$ are the inputs, outputs, and states predicted i steps ahead at time k, respectively, which act as decision variables; Eq. (4a) reports the MPC cost function ($\|x\|^2_M \equiv \frac{1}{2} x^\top M x$); Eq. (4b) reports the MPC prediction model constraints (with prediction

horizon N_p), given by the DT plant model (3); Eq. (4c) reports input and output constraints, satisfying Assumptions 3, 4, and Remark 2; $y_{r,k}$ is the reference output at the current time instant k; $Q_k, Q_{\Delta,k}, R_{\Delta,k} \succeq 0$, $R_k \succ 0$ are diagonal weighting matrices of suitable dimensions, possibly changing at each time instant k. The optimal solution of the MPC problem (4) is denoted as $(\hat{u}^*, \hat{y}^*, \hat{x}^*)$.

The cost function (4a) is composed by four terms:

- $\|\hat{y}_i - y_{r,k}\|^2_{Q_k}$ serves as tracking term for the predicted output $\hat{y}_i$ towards the reference output $y_{r,k}$;
- $\|\hat{u}_i\|^2_{R_k}$ acts as regularization term for the predicted inputs;
- $\|\hat{y}_i - \hat{y}_{i-1}\|^2_{Q_{\Delta,k}}$ and $\|\hat{u}_i - \hat{u}_{i-1}\|^2_{R_{\Delta,k}}$ penalize the variation in time of the predicted outputs and inputs, respectively, to obtain smoother predicted trajectories [17].

The diagonal elements of the weighting matrices are denoted as MPC weights w_k, i.e.,

$$\begin{aligned} w_k &= [w_{Q,k}^\top,\ w_{R,k}^\top,\ w_{Q_\Delta,k}^\top,\ w_{R_\Delta,k}^\top]^\top \in \mathbb{R}^{n_w}, \\ w_{Q,k} &= \mathrm{diag}(Q_k), \qquad w_{R,k} = \mathrm{diag}(R_k), \\ w_{Q_\Delta,k} &= \mathrm{diag}(Q_{\Delta,k}), \quad w_{R_\Delta,k} = \mathrm{diag}(R_{\Delta,k}). \end{aligned} \tag{5}$$

The CT plant (1) is controlled via the one-step receding horizon policy, i.e., at each time instant k, the first optimal predicted input $\hat{u}_0^*$ given by the MPC problem (4) is applied to plant (1) over the time interval $[kT_s, (k+1)T_s]$.

A. Relaxation of the MPC Optimal Control Problem

The MPC optimal control problem (4) is a nonlinear program (NLP), due to the nonlinear prediction model constraints (4b). To obtain a more tractable optimization problem, we relax the MPC problem (4) into a quadratic program (QP), obtaining a QP-MPC problem. To do so, the prediction model (4b) is relaxed into an affine parameter-varying (APV) model, parametrized by the current state x_k, i.e.,

$$\begin{aligned} \hat{x}_{i+1} &= f(\hat{x}_i, \hat{u}_i) \approx A(x_k)\hat{x}_i + B(x_k)\hat{u}_i + b(x_k), \\ \hat{y}_i &= g(\hat{x}_i, \hat{u}_i) \approx C(x_k)\hat{x}_i + D(x_k)\hat{u}_i + d(x_k), \end{aligned} \tag{6}$$

where $\star(x_k) \equiv \star_k$, $\star = A, B, b, C, D, d$.

The approach for constructing the parameter-varying terms in Eq. (6) is reported in Section III-B.

Furthermore, to avoid infeasibility issues, the hard output constraints in Eq. (4c) are softened by introducing a slack variable $\varepsilon \in \mathbb{R}^{N_{I,y}}$, where $N_{I,y}$ is the number of output inequality constraints (i.e., $H_y \in \mathbb{R}^{N_{I,y} \times n_y}$, $h_y \in \mathbb{R}^{N_{I,y}}$), with a related additional term in cost function (4a), i.e.,

$$J'(\hat{u}, \hat{y}, \varepsilon) = J(\hat{u}, \hat{y}) + \|\varepsilon\|^2_S, \tag{7}$$

$$H_y \hat{y}_i \le h_y + \varepsilon, \quad \varepsilon \ge 0, \quad i = 0, \dots, N_p - 1, \tag{8}$$

with $S \succ 0$ and diagonal.

By replacing the prediction model in Eq. (4b) with Eq. (6), the cost function in Eq. (4a) with Eq. (7), and the output constraints in Eq. (4c) with Eq. (8), we obtain the following relaxed QP-MPC problem:

$$\min_{\hat{u}, \hat{y}, \hat{x}, \varepsilon} \quad J'(\hat{u}, \hat{y}, \varepsilon) \tag{9a}$$

$$\text{s.t.} \quad \forall\, i = 0, \dots, N_p - 1,$$

$$\hat{x}_0 = x_k, \quad \hat{x}_{i+1} = A_k \hat{x}_i + B_k \hat{u}_i + b_k, \tag{9b}$$

$$\hat{y}_i = C_k \hat{x}_i + D_k \hat{u}_i + d_k, \tag{9c}$$

$$H_u \hat{u}_i \le h_u, \quad H_y \hat{y}_i \le h_y + \varepsilon, \quad \varepsilon \ge 0. \tag{9d}$$

III. Neural Adaptive MPC

In the following, starting from the MPC formulation in Section II, we present our novel Neural Adaptive MPC (NA-MPC) control strategy. As stated in the Introduction, the construction of NA-MPC requires three main steps:

1) Identify a neural networks-based prediction model for the MPC problem (9), so to enable the controller applicability when accurate white or gray-box models of the plant are difficult to obtain.
2) Recast the MPC problem (9) as a static control policy $u_k^* = \pi(\Pi_k)$, where u_k^* is the optimal control input and Π_k are the parameters of the MPC problem (9), and employ a suitable neural network to accurately emulate it.
3) Formulate an MPC tuning algorithm that, given a set of control specifications, is able to obtain optimal parameters Π_k^* in real time, adapting the MPC policy π online.

Steps 1 and 2 are assessed in this section, while step 3 is assessed in Section IV.

Several results presented in the following rely on the property of universal approximation of feedforward neural networks.

Definition 1. *A feedforward neural network (FNN) with L fully-connected layers is a static function $\mathcal{N} : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$ with the following structure:*

$$z_{\text{out}} = \mathcal{N}(z_{\text{in}}) = (\phi_L \circ \cdots \circ \phi_1)(z_{\text{in}}), \tag{10a}$$

$$z_{l+1} = \phi_l(z_l) = \sigma_l(W_l z_l + b_l), \quad l = 1, \dots, L, \tag{10b}$$

where each layer ϕ_l is characterized by the learnable parameters $W_l \in \mathbb{R}^{N_{l+1}, N_l}$ and $b_l \in \mathbb{R}^{N_{l+1}}$ (i.e., weights and biases), the number of neurons N_l ($N_1 = n_{\text{in}}$, $N_{L+1} = n_{\text{out}}$), and the activation function σ_l; $\circ$ is the function composition operator.

The approximation capability of FNNs is formally proven by the well-known universal approximation theorem:

Theorem 1 ([4], [18])**.** *Let $f : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$ be a continuous function. Then, for any $\delta > 0$, it exists a FNN $\mathcal{N} : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$ with $L \ge 2$ layers and activation functions being locally bounded, piecewise continuous, and non-polynomial, such that, for any z within a given compact set $\mathcal{Z}$, it holds*

$$\|\mathcal{N}(z) - f(z)\| < \delta, \quad \forall\, z \in \mathcal{Z}. \tag{11}$$

In practical terms, Theorem 1 states that a continuous function can be approximated with arbitrary accuracy by a FNN with a properly chosen number of layers and neurons, and trained with a sufficiently rich dataset [4].

A. Neural Identification of the MPC Prediction Model

MPC-based control strategies require a sufficiently accurate prediction model of the plant to deliver an effective control action. However, real-world scenarios often present a lack of

relevant physical information on the plant, posing a significant challenge for conventional white or gray-box modeling.

Therefore, in the following, we treat the CT plant (1) as an unknown system, onto which only input-output measurements $\{\tilde{u}_i\}_{i=1}^N$ and $\{\tilde{y}_i\}_{i=1}^N$, corrupted by measurement errors n_u and n_y, can be collected, as follows:

$$\dot{x}_i = f_c(x_i, u_i), \qquad y_i = g(x_i, u_i), \tag{12a}$$
$$\tilde{u}_i = u_i \odot (1 + n_u), \quad \tilde{y}_i = y_i \odot (1 + n_y), \tag{12b}$$
$$n_u \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_u, \sigma_u^2), \quad n_y \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_y, \sigma_y^2), \tag{12c}$$

where $\odot$ denotes the element-wise product. In Eq. (12), n_u and n_y are multiplicative percentage errors, modeled as two vectors of independent and identically distributed (i.i.d.) random variables, with normal distribution $\mathcal{N}$ (mean $\mu_\star = 0$, variance defined by the "3-sigma" value $\overline{n}_\star \equiv 3\sigma_\star$, $\star = u, y$).

By Assumption 3, we can measure the error-corrupted state trajectory $\{\tilde{x}_i\}_{i=1}^N$; from the latter, we can also estimate the error-corrupted values $\{\dot{\tilde{x}}_i\}_{i=1}^N$, by approximating the derivative with finite differences. Finally, we construct the input and output datasets $\mathcal{D}_{\text{in}} = \{(\tilde{x}_i, \tilde{u}_i)\}_{i=1}^N$, $\mathcal{D}_{\text{out}} = \{(\dot{\tilde{x}}_i, \tilde{y}_i)\}_{i=1}^N$.

Since we assumed that no accurate physical knowledge of the plant is available for white or gray-box identification, we employ the collected data $\mathcal{D}_{\text{in}}$, $\mathcal{D}_{\text{out}}$ to identify a black-box model of plant (1). Such a black-box model is composed by two FNNs, $\mathcal{N}_{f_c}$ and $\mathcal{N}_g$, for functions f_c and g, respectively[1],

$$\dot{x} = \mathcal{N}_{f_c}(x, u), \quad y = \mathcal{N}_g(x, u). \tag{13}$$

By Assumptions 1a and 1b, f_c and g are continuous, as required by Theorem 1. Moreover, if the activation functions σ_l are Lipschitz continuous, then the full network is also Lipschitz continuous [19]. Therefore, the FNN black-box model (13) satisfies Assumptions 1a, 1b, and 3.

The identified model (13) is then employed as prediction model for the MPC problem (9), prior to its relaxation.

B. Relaxation of the Neural MPC Prediction Model

The identified neural black-box MPC prediction model (13) has to be relaxed into an APV model (6) to obtain the QP-MPC problem (9), as described in Section II-A.

We want to ensure that, for any value of the parameter x_k, the related APV model, given by the terms $\star(x_k)$, $\star = A, B, b, C, D, d$, is the most accurate affine approximation of the true model dynamics (13) over the set $\hat{\Xi}_k = \hat{\mathcal{X}}_k \times \mathcal{U}$, where

$$\begin{aligned}
&\hat{\mathcal{X}}_k = \{x \in \mathbb{R}^{n_x} : x_k + T_s N_p \, \underline{\dot{x}} \leq x \leq x_k + T_s N_p \, \overline{\dot{x}}\},\\
&(\underline{\dot{x}})_j = \min_{(x,u)\in\Xi} (\mathcal{N}_{f_c}(x,u))_j,\\
&(\overline{\dot{x}})_j = \max_{(x,u)\in\Xi} (\mathcal{N}_{f_c}(x,u))_j, \quad j = 1, \dots, n_x,
\end{aligned} \tag{14}$$

and $\Xi \equiv \mathcal{X} \times \mathcal{U}$; $\hat{\mathcal{X}}_k$ is the set of all possibly attainable states within the MPC prediction horizon, starting from the current state x_k. To obtain the most accurate APV model over $\hat{\Xi}_k$, we compute the parameter-varying terms $\star(x_k) \equiv \star_k$, $\star = A, B, b, C, D, d$, through least-squares fitting, using the following fitting data ($\xi \equiv [x^\top, u^\top]^\top$):

$$\xi_i \in \hat{\Xi}_k, \quad \phi_\xi = [\xi_i]_{i=1}^N \in \mathbb{R}^{n_x + n_u \times N}, \tag{15a}$$
$$\phi_x = [\dot{x}_i]_{i=1}^N = [\mathcal{N}_{f_c}(\xi_i)]_{i=1}^N \in \mathbb{R}^{n_x \times N}, \tag{15b}$$
$$\phi_y = [y_i]_{i=1}^N = [\mathcal{N}_g(\xi_i)]_{i=1}^N \in \mathbb{R}^{n_y \times N}. \tag{15c}$$

Defining $\Phi = [\phi_\xi^\top, 1_N]$ ($1_N \equiv [1, \dots, 1]^\top \in \mathbb{R}^N$), by least-squares fitting we have that

$$[A_k, B_k, b_k] = \phi_x {\Phi^\dagger}^\top, \quad [C_k, D_k, d_k] = \phi_y {\Phi^\dagger}^\top, \tag{16}$$

where the superscript † denotes the pseudoinverse matrix.

By construction, the APV prediction model (16) is the most accurate one – in least-squares sense – over the set $\hat{\Xi}_k$.

Remark 3. *It is worth noticing that, by the least-squares fitting in Eqs.* (15) *and* (16)*, the parameters A_k, B_k, b_k, C_k, D_k, d_k, are all continuous functions of the current state x_k.*

C. Neural Emulation of the MPC Optimal Control Policy

The QP-MPC problem (9), employing the relaxed FNN black-box prediction model (13), can be rewritten as follows:

$$\min_v \frac{1}{2} v^\top H_k v + c_k^\top v \quad \text{s.t.} \quad M_k v = p_k, \quad N v \leq q, \tag{17}$$

where $v = \text{vec}\,(\hat{u}, \hat{y}, \hat{x}, \varepsilon)$ are the decision variables arranged in a vector, $H_k = H_k^\top \succ 0$, and M_k has full row rank [20], [21]. Each time-variant element in Eq. (17) is explicitly dependent on the time-variant parameters of the MPC problem (9), i.e.,

$$\begin{aligned}
&H_k = H(Q_k, Q_{\Delta,k}, R_k, R_{\Delta,k}), \quad c_k = c(y_{r,k}, Q_k),\\
&M_k = M(A_k, B_k, C_k, D_k), \qquad p_k = p(x_k, b_k, d_k).
\end{aligned} \tag{18}$$

In Eq. (18), H, c, M, and p are all continuous functions [20]. Let us then denote with Π_k the tuple of time-variant parameters of the MPC problem (9) at the current time instant k,

$$\begin{aligned}
\Pi_k = (&x_k, y_{r,k}, Q_k, R_k, Q_{\Delta,k}, R_{\Delta,k},\\
&A_k, B_k, b_k, C_k, D_k, d_k).
\end{aligned} \tag{19}$$

In Eq. (17), the cost and the inequality constraints are convex functions; thus, the QP (17) – and, hence, the MPC problem (9) – admits a unique global optimum v^* [22]. Therefore, we can think in principle to associate, to each feasible value of Π_k, the related global solution $u_k^* = \hat{u}_0^*$ of the MPC problem (9) (according to the one-step receding horizon, see Section II). In this way, we are constructing a static policy

$$u_k^* = \pi(\Pi_k), \tag{20}$$

corresponding to the control action given by the MPC problem (9). Since Π_k includes both the MPC weighting matrices Q_k, R_k, $Q_{\Delta,k}$, $R_{\Delta,k}$ and the APV prediction model A_k, B_k, b_k, C_k, D_k, d_k, the computation of the static policy (20) cannot be tackled with classic explicit MPC arguments and multi-parametric QP [23]. Therefore, in the following we rely on the universal approximation capability of FNNs to obtain an accurate estimate of the policy (20).

To properly approximate the MPC policy (20) with a FNN, we have to verify first that such policy is continuous. Several

[1] In the following, we split up FNN inputs into multiple arguments to improve readability (e.g., in Eq. (13), $\mathcal{N}_{f_c}(x,u) \equiv \mathcal{N}_{f_c}(\xi)$, where $\xi \equiv [x^\top, u^\top]^\top$).

works have investigated the continuity of QPs, both in terms of parameters perturbation [24] and upper-lower semicontinuity [25]. For our purposes, we employ the following result:

Theorem 2 ([24])**.** *Let the following QP problem be given:*

$$\min_x \frac{1}{2}x^\top Hx + c^\top x \quad \textit{s.t.} \quad Mx = p, \quad Nx \leq q, \tag{21}$$

where $H = H^\top \succ 0$ and M has full column rank. Let us denote with $\Pi = (H, c, M, p, N, q)$ the tuple of parameters of the QP (21)*. Let us also consider the function $x^* = \pi(\Pi)$, which gives the unique optimum of the QP* (21) *as function of the parameters* Π*.*

Then, there exist $\alpha, \delta > 0$ such that, for any $\varepsilon \leq \delta$, if $\Pi' = (H', c', M', p', N', q')$ satisfies

$$\begin{aligned}\max\{&\|H - H'\|, \|c - c'\|, \|M - M'\|,\\ &\|p - p'\|, \|N - N'\|, \|q - q'\|\} \leq \delta,\end{aligned} \tag{22}$$

then it holds

$$\|x^* - x^{*\prime}\| = \|\pi(\Pi) - \pi(\Pi')\| \leq \alpha\varepsilon. \tag{23}$$

Therefore, Theorem 2, leveraging Weierstrass' definition of continuity, proves that the optimum x^* of the QP (21) is a continuous function of the QP parameters Π.

Since Eq. (17) matches the structure of Eq. (21) and the functions in Eq. (18) are all continuous, then, by Theorem 2, the MPC policy π in Eq. (20) is continuous. Therefore, by Theorem 1, it can be effectively approximated by a FNN $\mathcal{N}_\pi$,

$$u_k^* = \mathcal{N}_\pi(\Pi_k). \tag{24}$$

The FNN (24) constitutes our neural MPC controller.

The number of inputs Π_k of the neural MPC controller (24) can be reduced by recalling Remark 3, for which we can keep only the parameter x_k in place of A_k, B_k, b_k, C_k, D_k, d_k, while still satisfying Theorem 1. Moreover, from Eq. (5), the parameters Q_k, R_k, $Q_{\Delta,k}$, $R_{\Delta,k}$ can be replaced with their diagonal elements, i.e., the MPC weights w_k in Eq. (5). Then, the final structure of the neural MPC controller is

$$u_k^* = \mathcal{N}_\pi(x_k, y_{r,k}, w_k). \tag{25}$$

The neural MPC controller (24) is trained by constructing input and output datasets $\mathcal{D}_{\text{in}}$, $\mathcal{D}_{\text{out}}$: $\mathcal{D}_{\text{in}}$ is obtained by selecting a set of feasible values of Π; $\mathcal{D}_{\text{out}}$ is obtained evaluating the MPC problem (9) for each value in $\mathcal{D}_{\text{in}}$.

Emulating the MPC policy (20) using a FNN (24) offers a significant advantage: unlike solving the original MPC problem (9), evaluating the policy (24) requires near-zero computation time, since FNNs are static functions. This is a key feature for enabling an online optimal tuning strategy for the neural MPC controller, as explained in Section IV.

IV. Online Metaheuristic Strategy for Adaptive MPC Tuning

A. Optimal Tuning Problem

Online tuning has the purpose of adapting in real time the MPC weights w_k in Eq. (5), allowing to maintain consistent control performance over time and to perform optimal trade-offs between concurrent control objectives.

Our tuning strategy consists of three main steps:

1) For a given value of the MPC weights w, simulate back the system in closed-loop over the past N_{tuning} time instants.
2) From the simulated closed-loop trajectories, assess how well each control objective is attained, by evaluating a set of performance-metric functions p_i: when p_i is minimized, the i-th control objective is achieved at its best.
3) Find the weights w^* providing the best values for all p_i.

This tuning approach respects causality, since only past information up to the current time instant is employed.

The tuning strategy is then formulated as a multi-objective optimization problem, which, at time instant k, seeks the optimal weights w_k^* satisfying

$$\min_{w_k} \quad J_{\text{tuning}}(\hat{y}), \quad J_{\text{tuning}}(\hat{y}) = \sum_{i=1}^{n_p} \alpha_i p_i(\hat{y}) \tag{26a}$$

$$\text{s.t.} \quad \forall\, i = k - N_{\text{tuning}}, \dots, k-1,$$

$$\hat{u}_i^* = \mathcal{N}_\pi(\hat{x}_i, y_{r,i}, w_k), \tag{26b}$$

$$\hat{x}_{i+1} = \hat{x}_i + \int_{iT_s}^{(i+1)T_s} \mathcal{N}_{f_c}(\hat{x}(\tau), \hat{u}_i^*)\, d\tau, \tag{26c}$$

$$\hat{y}_i = \mathcal{N}_g(\hat{x}_i, \hat{u}_i^*), \quad \hat{x}_{k-N_{\text{tuning}}} = x_{k-N_{\text{tuning}}}, \tag{26d}$$

where $\hat{y} = \{\hat{y}_i\}_{i=k-N_{\text{tuning}}}^{k-1}$ is the simulated closed-loop output trajectory; Eq. (26a) reports the tuning cost function, that is the linear combination of the performance-metric functions p_i, $i = 1, \dots, n_p$; α_i are scalars for setting the priority of each metric over the others; Eqs. (26b)-(26d) represent the closed-loop system dynamics, under the control of the neural MPC controller (25) and employing as plant the FNN black-box model (13). Successive tunings are performed every N_{tuning} time instants; therefore, the optimal weights w_k^* are kept constant over the time interval $[kT_s, (k + N_{\text{tuning}} - 1)T_s]$ and the tuning problem (26) is solved only at time instants $k = nN_{\text{tuning}}$, $n \in \mathbb{N}_{\geq 1}$. Therefore, the weights w_0, employed during the first time interval $[0, (N_{\text{tuning}} - 1)T_s]$, have to be initialized by the user.

As stated in Section III-C, the prior simulation in Eqs. (26b)-(26d) can be performed in real time since the control policy is evaluated through the neural MPC controller $\mathcal{N}_\pi$ (25), which, being a FNN, exhibits near-zero computation time.

The union of the neural MPC controller (25) with the online tuning strategy (26) constitutes the complete NA-MPC controller.

B. Tuning Problem Solution via Metaheuristic Optimization

The tuning problem (26) requires to perform the prior simulation (26b)-(26d) in order to evaluate the cost function (26a). As a consequence, the tuning problem (26) cannot be effectively solved with classic optimization algorithms. For this reason, we leverage metaheuristic optimization.

Among the many available metaheuristic optimization techniques [26], [27], we employ Particle Swarm Optimization (PSO), the most well-regarded metaheuristic algorithm among the family of swarm intelligence techniques. PSO is widely recognized for its conceptual simplicity, fast convergence, and effectiveness for multi-objective optimization problems [26].

Note that, a thorough validation of PSO, also comparing it with Genetic Algorithm (GA) (another widely-employed metaheuristic optimization strategy), is reported in the supplementary material.

The particle swarm $\{w_i\}_{i=1}^{N_{\rm part}}$ is randomly initialized over the hyper-rectangle $\mathcal{W} = \{w \in \mathbb{R}^{n_w} : \underline{w} \leq w \leq \overline{w}\}$, bounding the possible MPC weight values. Each particle is associated with a velocity vector $v_i \in \mathbb{R}^{n_w}$, $i = 1, \ldots, N_{\rm part}$. The PSO algorithm is iterated $N_{\rm PSO}$ times; at each iteration $j = 0, \ldots, N_{\rm PSO} - 1$, the particle positions $w_i^{(j)}$ are updated as follows:

$$v_i^{(j+1)} = m v_i^{(j)} + c_I r_I (p_i^{(j)} - w_i^{(j)}) + c_S r_S (g^{(j)} - w_i^{(j)}), \quad (27a)$$

$$w_i^{(j+1)} = w_i^{(j)} + v_i^{(j+1)}, \quad i = 1, \ldots, N_{\rm part}. \quad (27b)$$

In Eq. (27), m is the particle inertia; c_I and c_S are the individual and social attraction coefficients, respectively; $p_i^{(j)}$ is the personal best position of particle i up to iteration j; $g^{(j)}$ is the global best position of the swarm up to iteration j; $r_I, r_S \sim \mathcal{U}([0,1])$ are random numbers with uniform distribution $\mathcal{U}$.

The optimal solution w_k^* of the tuning problem (26), for $k = nN_{\rm tuning}$, $n \geq 1$, is computed as follows:

$$w_k^* = \frac{1}{2}(g^{(N_{\rm PSO})} + w_{k-N_{\rm tuning}}^*), \quad (28)$$

where $g^{(N_{\rm PSO})}$ is the global best position at the last PSO iteration. The policy (28) has the purpose of smoothing the evolution of the optimal MPC weights over time.

V. NA-MPC for Power Management in Fuel Cell Hybrid Electric Vehicles

In this work, we apply our versatile NA-MPC strategy to the case study of power management in fuel cell hybrid electric vehicles (FCHEVs). The FCHEV power sources and distribution system are depicted in Figure 3, matching the general power system structure in Figure 2.

In the following, we outline the construction of a white-box first-principles model of the FCHEV power system, to be used both as unknown plant to control and as source of input-output measurements (see Section III-A). Then, we describe the tailoring of the NA-MPC strategy to achieve the main control tasks of FCHEV power management.

A. Battery Model

The battery model takes as input the power requested to the battery ($P_{\rm b}$) and returns as output the state of charge of the battery ($SOC \equiv \zeta$) at the current time instant. The SOC is defined as the ratio between the battery charge $Q_{\rm b}$ and the nominal battery capacity $Q_{\rm nom}$, i.e.,

$$SOC \equiv \zeta = \frac{Q_{\rm b}}{Q_{\rm nom}} \in [0, 1]. \quad (29)$$

Differentiating both sides, we can relate the SOC rate $\dot{\zeta}$ with the battery current $I_{\rm b}$ as follows:

$$\dot{\zeta} = -\frac{\eta_{\rm b,1}(P_{\rm b})\, I_{\rm b}}{Q_{\rm nom}},$$

$$\eta_{\rm b,1}(P_{\rm b}) = \begin{cases} \frac{1}{\eta_{\rm b}^{\rm coul}} & \text{if } P_{\rm b} > 0 \quad \text{(discharge)}, \\ \eta_{\rm b}^{\rm coul} & \text{if } P_{\rm b} < 0 \quad \text{(charge)}, \end{cases} \quad (30)$$

where $\eta_{\rm b}^{\rm coul} \in (0, 1)$ is the Coulombic efficiency, quantifying the fraction of current that is lost during battery charge and discharge [28].

The battery is made of $N_{\rm b}$ battery cells connected in series. Each cell is modelled as an ideal voltage source $V_{\rm b}^{\rm oc,s}$ with a series output resistance $R_{\rm b}^{\rm o,s}$; the latter quantities can be combined as $V_{\rm b}^{\rm oc} = N_{\rm b} V_{\rm b}^{\rm oc,s}$ and $R_{\rm b}^{\rm o} = N_{\rm b} R_{\rm b}^{\rm o,s}$, respectively. In real batteries, $V_{\rm b}^{\rm oc}$ and $R_{\rm b}^{\rm o}$ are functions of the battery SOC. Such dependence on ζ is derived from experimental data taken from [28]; the functions $V_{\rm b}^{\rm oc}(\zeta)$ and $R_{\rm b}^{\rm o}(\zeta)$ are then constructed by means of piecewise polynomial fitting.

The power generated by the battery ($P_{\rm b}^{\rm gen}$) is then equal to

$$P_{\rm b}^{\rm gen} = V_{\rm b}^{\rm oc} I_{\rm b} - R_{\rm b}^{\rm o} I_{\rm b}^2. \quad (31)$$

To compute the actual delivered power $P_{\rm b}$, we have to account for the losses due to the power converters interfacing the battery with the DC bus. Therefore, we set

$$P_{\rm b}^{\rm gen} = \eta_{\rm b,2}(P_{\rm b}) P_{\rm b}, \quad \eta_{\rm b,2}(P_{\rm b}) = \begin{cases} \frac{1}{\eta_{\rm b}^{\rm conv}} & \text{if } P_{\rm b} > 0, \\ \eta_{\rm b}^{\rm conv} & \text{if } P_{\rm b} < 0, \end{cases} \quad (32)$$

where $\eta_{\rm b}^{\rm conv} \in (0, 1)$ is the efficiency of the battery power converters. The efficiency $\eta_{\rm b,2}$ accounts for the bidirectional flow of the battery power during charge and discharge. Solving Eq. (31) for $I_{\rm b}$ and adding Eq. (32) yields

$$I_{\rm b} = \frac{1}{2R_{\rm b}^{\rm o}} \left(V_{\rm b}^{\rm oc} - \sqrt{V_{\rm b}^{\rm oc\,2} - 4R_{\rm b}^{\rm o}\, \eta_{\rm b,2}(P_{\rm b})\, P_{\rm b}} \right). \quad (33)$$

The battery model is then the union of Eqs. (30) and (33),

$$\dot{\zeta} = -\frac{\eta_{\rm b,1}(P_{\rm b})}{Q_{\rm nom}} \frac{1}{2R_{\rm b}^{\rm o}(\zeta)} \Big(V_{\rm b}^{\rm oc}(\zeta) - \sqrt{V_{\rm b}^{\rm oc}(\zeta)^2 - 4R_{\rm b}^{\rm o}(\zeta)\, \eta_{\rm b,2}(P_{\rm b})\, P_{\rm b}} \Big) \equiv f_{c,\rm b}(\zeta, P_{\rm b}). \quad (34)$$

Model (34) forces an upper bound on $P_{\rm b}$, i.e.,

$$P_{\rm b} \leq \eta_{\rm b}^{\rm conv} \frac{V_{\rm b}^{\rm oc}(\zeta)^2}{4R_{\rm b}^{\rm o}(\zeta)} \equiv \overline{P}_{\rm b}^{\rm model}(\zeta). \quad (35)$$

Real battery specifications also provide a range of admissible values for $P_{\rm b}$ and ζ, denoted as $[\underline{P}_{\rm b}^{\rm spec}, \overline{P}_{\rm b}^{\rm spec}]$ and $[\underline{\zeta}, \overline{\zeta}]$, respectively. Typically, $\overline{P}_{\rm b}^{\rm spec} < \min_{\zeta \in [\underline{\zeta}, \overline{\zeta}]} \overline{P}_{\rm b}^{\rm model}(\zeta)$, so that the upper bound on $P_{\rm b}$ is independent of ζ and is feasible for any $\zeta \in [\underline{\zeta}, \overline{\zeta}]$. Therefore, ζ and $P_{\rm b}$ range as follows:

$$\zeta \in [\underline{\zeta}, \overline{\zeta}], \quad P_{\rm b} \in [\underline{P}_{\rm b}^{\rm spec}, \overline{P}_{\rm b}^{\rm spec}] \equiv [\underline{P}_{\rm b}, \overline{P}_{\rm b}]. \quad (36)$$

B. Fuel Cell Model

The fuel cell model takes as input the power requested to the fuel cell ($P_{\rm fc}$) and returns as output the hydrogen mass left ($m_{\rm h}$) at the current time instant. In the model, proton-exchange membrane (PEM) fuel cell behaviour is assumed.

The fuel cell is made of $N_{\rm fc}$ smaller cells connected in series as a stack. The electrical behaviour of each smaller cell is defined by its characteristic curve, relating the cell voltage $V_{\rm fc}^{\rm s}$ to the cell current density $i_{\rm fc}^{\rm s}$ [29], [30], defined as the cell

current per active area, i.e., $i_{\rm fc}^{\rm s} = I_{\rm fc}^{\rm s}/A_{\rm fc}^{\rm s}$. Within a wide range of current densities, such a curve can be conveniently fitted by a linear function [29], i.e.,

$$V_{\rm fc}^{\rm s} \approx V_{\rm fc}^{\rm oc,s} - r_{\rm fc}^{\rm o,s} i_{\rm fc}^{\rm s} = V_{\rm fc}^{\rm oc,s} - R_{\rm fc}^{\rm o,s} I_{\rm fc}^{\rm s}. \tag{37}$$

The fuel cell stack voltage and current are then equal to $V_{\rm fc} = N_{\rm fc} V_{\rm fc}^{\rm s}$ and $I_{\rm fc} = I_{\rm fc}^{\rm s}$, respectively. We also define $V_{\rm fc}^{\rm oc} = N_{\rm fc} V_{\rm fc}^{\rm oc,s}$ and $R_{\rm fc}^{\rm o} = N_{\rm fc} R_{\rm fc}^{\rm o,s}$.

The power generated by the fuel cell ($P_{\rm fc}^{\rm gen}$) is equal to the ideal power $P_{\rm fc}^{\rm id} = V_{\rm fc} I_{\rm fc}$, reduced by a quantity $P_{\rm fc}^{\rm aux}$, which is the power needed to keep the fuel cell on [29]. It can be shown that $P_{\rm fc}^{\rm aux}$ depends quadratically on $I_{\rm fc}$ [29] and its expression can be further simplified with a linear relation [31], i.e., $P_{\rm fc}^{\rm aux} \approx P_{\rm fc}^{\rm aux\prime} + V_{\rm fc}^{\rm aux} I_{\rm fc}$. Therefore, we have that

$$P_{\rm fc}^{\rm gen} = (V_{\rm fc} - V_{\rm fc}^{\rm aux}) I_{\rm fc} - P_{\rm fc}^{\rm aux\prime}. \tag{38}$$

As done for the battery, to compute the actual delivered power $P_{\rm fc}$, we have to account for the losses due to the power converters interfacing the fuel cell with the DC bus. Thus,

$$P_{\rm fc}^{\rm gen} = \tfrac{1}{\eta_{\rm fc}^{\rm conv}} P_{\rm fc}, \tag{39}$$

where $\eta_{\rm fc}^{\rm conv} \in (0,1)$ is the efficiency of the fuel cell power converters. Combining Eqs. (37)-(39) and solving for $I_{\rm fc}$ yields

$$I_{\rm fc} = \frac{1}{2R_{\rm fc}^{\rm o}}\Big((V_{\rm fc}^{\rm oc} - V_{\rm fc}^{\rm aux}) - \sqrt{(V_{\rm fc}^{\rm oc} - V_{\rm fc}^{\rm aux})^2 - 4R_{\rm fc}^{\rm o}(\tfrac{1}{\eta_{\rm fc}^{\rm conv}} P_{\rm fc} + P_{\rm fc}^{\rm aux\prime})}\Big). \tag{40}$$

Experimental data on both the fuel cell characteristic curve and auxiliary power are taken from [29].

Finally, we relate the hydrogen mass rate $\dot{m}_{\rm h}$ with the fuel cell current $I_{\rm fc}$ recalling the oxidation semi-reaction taking place at the fuel cell anode, i.e., $\mathrm{H_2} \rightarrow 2\mathrm{H^+} + 2\mathrm{e^-}$. The number of hydrogen molecules that dissociate every second equals $I_{\rm fc}/n_{\rm e}q$, where $n_{\rm e} = 2$ (number of electrons released) and q is the elementary charge. The hydrogen rate $\dot{m}_{\rm h}$ of the fuel cell stack is then equal to

$$\dot{m}_{\rm h} = -N_{\rm fc} \frac{I_{\rm fc}\, M_{\rm h}}{n_{\rm e}\, q\, N_{\rm A}}, \tag{41}$$

being $M_{\rm h}$ the $\mathrm{H_2}$ molar mass and $N_{\rm A}$ the Avogadro's number.

The fuel cell model is then the union of Eqs. (40) and (41),

$$\begin{aligned}\dot{m}_{\rm h} = & -\frac{N_{\rm fc}\, M_{\rm h}}{n_{\rm e}\, q\, N_{\rm A}} \frac{1}{2R_{\rm fc}^{\rm o}}\Big((V_{\rm fc}^{\rm oc} - V_{\rm fc}^{\rm aux}) - \\ & \sqrt{(V_{\rm fc}^{\rm oc} - V_{\rm fc}^{\rm aux})^2 - 4R_{\rm fc}^{\rm o}(\tfrac{1}{\eta_{\rm fc}^{\rm conv}} P_{\rm fc} + P_{\rm fc}^{\rm aux\prime})}\Big) \\ \equiv & f_{c,\rm fc}(P_{\rm fc}).\end{aligned} \tag{42}$$

Model (42) forces an upper bound on $P_{\rm fc}$, i.e.,

$$P_{\rm fc} \le \eta_{\rm fc}^{\rm conv}\left(\frac{(V_{\rm fc}^{\rm oc} - V_{\rm fc}^{\rm aux})^2}{4R_{\rm fc}^{\rm o}} - P_{\rm fc}^{\rm aux\prime}\right) \equiv \overline{P}_{\rm fc}^{\rm model}. \tag{43}$$

Real fuel cell specifications also provide an upper bound on $P_{\rm fc}$, denoted as $\overline{P}_{\rm fc}^{\rm spec}$, satisfying $\overline{P}_{\rm fc}^{\rm spec} < \overline{P}_{\rm fc}^{\rm model}$. The hydrogen mass is bounded by $[\underline{m}_{\rm h}, \overline{m}_{\rm h}]$, where $\overline{m}_{\rm h}$ is the hydrogen tank capacity. Thus, $m_{\rm h}$ and $P_{\rm fc}$ range as follows:

$$m_{\rm h} \in [\underline{m}_{\rm h}, \overline{m}_{\rm h}], \quad P_{\rm fc} \in [0, \overline{P}_{\rm fc}^{\rm spec}] \equiv [\underline{P}_{\rm fc}, \overline{P}_{\rm fc}]. \tag{44}$$

C. FCHEV Power System Model

The FCHEV power system model is the union of Eqs. (34) and (42). The model states, inputs, and outputs are

$$x = [\zeta, m_{\rm h}]^\top, \quad u = [P_{\rm b}, P_{\rm fc}]^\top, \quad y = [x^\top, P_{\rm tot}]^\top, \tag{45}$$

which are constrained by the bounds in Eqs. (36) and (44), defining the constraint sets $\mathcal{X} = \{x \in \mathbb{R}^{n_x} : \underline{x} \le x \le \overline{x}\}$ and $\mathcal{U} = \{u \in \mathbb{R}^{n_u} : \underline{u} \le u \le \overline{u}\}$, which satisfy Assumption 4.

The model equations are

$$\dot{x} = f_c(x,u) = \begin{bmatrix} f_{c,\rm b}(\zeta, P_{\rm b}) \\ f_{c,\rm fc}(P_{\rm fc}) \end{bmatrix}, \quad y = g(x,u) = \begin{bmatrix} x \\ P_{\rm tot} \end{bmatrix}. \tag{46}$$

where the total delivered power $P_{\rm tot}$ is given by

$$P_{\rm tot} = \sum_{i=1}^{n_u} \sigma(x_i, u_i), \tag{47a}$$

$$\sigma(x_i,u_i) = \begin{cases} 0 & \text{if } x_i < \underline{x}_i,\ u_i > 0 \\ & \text{or } x_i > \overline{x}_i,\ u_i < 0, \\ u_i & \text{otherwise.} \end{cases} \tag{47b}$$

Eq. (47b) states that a source cannot deliver power if its supply is depleted and the requested power is positive ($x_i < \underline{x}_i$, $u_i > 0$), or the supply is full and the requested power is negative ($x_i > \underline{x}_i$, $u_i < 0$). Then, the constraint set $\mathcal{Y}$ can be directly inferred from $\mathcal{X}$, $\mathcal{U}$, and Eq. (47), and satisfies Assumption 4.

Remark 4. *For each source, if $u_i = 0$ (null requested power), then $\dot{x}_i = 0$, $\forall\, x \in \mathcal{X}$ (the supply level does not change).*

Remark 5. *By Remark 4 and Eq.* (47)*, for any input $u(t) \in \mathcal{U}$, the state $x(t) \in \mathcal{X}$, $\forall\, t \ge 0$. Then, under the constraint $x \in \mathcal{X}$, Eq.* (47a) *simplifies as $P_{\rm tot} = \sum_{i=1}^{n_u} u_i$.*

By Eqs. (34), (42), (47), and Remark 5, model (46) satisfies Assumptions 1a, 1b, and 3. Finally, for model (46), the following data is employed: $N_{\rm b} = 200$, $Q_{\rm nom} = 60$ A h, $\eta_{\rm b}^{\rm coul} = 0.95$, $\eta_{\rm b}^{\rm conv} = 0.97$, $\underline{P}_{\rm b} = -290$ kW, $\overline{P}_{\rm b} = 290$ kW, $\underline{\zeta} = 0.2$, $\overline{\zeta} = 0.8$, $N_{\rm fc} = 500$, $V_{\rm fc}^{\rm oc,s} = 0.812$ V [29], $R_{\rm fc}^{\rm o,s} = 1.21$ mΩ [29], $P_{\rm fc}^{\rm aux\prime} = 100$ W [29], $V_{\rm fc}^{\rm aux} = 50$ mV [29], $\eta_{\rm fc}^{\rm conv} = 0.97$, $\overline{P}_{\rm fc} = 45$ kW, $\underline{m}_{\rm h} = 100$ g, $\overline{m}_{\rm h} = 1000$ g.

D. NA-MPC Strategy For FCHEVs Power Management

In this section, we tailor the NA-MPC strategy for FCHEV power management. Specifically, NA-MPC should be able to address the following three main control tasks:

1) accurate requested power tracking;
2) combined supplies saving;
3) selective supplies saving (i.e., either battery or FC).

1) Neural MPC Controller: With reference to Section III-C, the neural MPC controller (25) for FCHEV power management takes the following inputs: the current state $x_k = [\zeta_k, m_{{\rm h},k}]^\top$, the reference output $y_{r,k} = [\zeta_r, m_{{\rm h},r}, P_{{\rm tot},r,k}]^\top$, and the MPC weights w_k, i.e.,

$$\begin{aligned} & Q_k = \mathrm{diag}(w_{\zeta,k}, w_{m_{\rm h},k}, w_{P_{\rm tot},k}), \\ & Q_{\Delta,k} = \mathrm{diag}(w_{\Delta,\zeta,k}, w_{\Delta,m_{\rm h},k}, 0), \\ & R = \mathrm{diag}(w_{P_{\rm b}}, w_{P_{\rm fc}}), \ \ R_\Delta = \mathrm{diag}(w_{\Delta,P_{\rm b}}, w_{\Delta,P_{\rm fc}}). \end{aligned} \tag{48}$$

In Eq. (48), the weights that do not effectively contribute in achieving the control tasks are set to constant values.

Concerning the reference output $y_{r,k}$:

- The supply references are set to their respective upper bounds, i.e., $\zeta_r = \overline{\zeta}$ and $m_{\mathrm{h},r} = \overline{m}_{\mathrm{h}}$; this encodes the supplies saving task within the MPC cost function.
- The total delivered power reference is set to the current requested power, i.e., $P_{\mathrm{tot},r,k} = P_{\mathrm{req},k}$; this encodes the power tracking task in the MPC cost function.

The neural MPC controller has then $n_\Pi = 8$ inputs, i.e.,

$$u_k^* = \begin{bmatrix} P_{\mathrm{b}}^* \\ P_{\mathrm{fc}}^* \end{bmatrix} = \mathcal{N}_\pi(\Pi_k) = \mathcal{N}_\pi(\zeta_k, m_{\mathrm{h},k}, P_{\mathrm{req},k}, w_k). \quad (49)$$

Data for training the FNN black-box prediction model (13) is "virtually" collected from the white-box model of the FCHEV power system (46), as described in Section III-A. In this sense, the high-fidelity model (46) has the purpose of emulating a real-world FCHEV, onto which real measurement data would be collected. The prediction model is relaxed as described in Sections II-A and III-B.

To ensure input constraint satisfaction, the output of the neural MPC controller (49) is saturated within the bounds $\underline{u}$ and $\overline{u}$ in Eqs. (36) and (44), i.e., $u_k^* = \mathrm{sat}_{[\underline{u},\overline{u}]}(\mathcal{N}_\pi(\Pi_k))$.

2) Performance-Metric Functions for Online Tuning: With reference to Section IV-A, we define $n_p = 3$ performance-metric functions p_i, $i = 1, \dots, n_p$, to quantify the satisfaction of the three power management control tasks:

- Requested power tracking (p_1):

$$p_1(\hat{y}) = \begin{cases} m(1 - \frac{\hat{e}_k}{\overline{e}})^2 & \text{if } \hat{e}_k \le \overline{e}, \\ \frac{2M}{1+\exp(-K(\hat{e}_k - \overline{e}))} - M & \text{if } \hat{e}_k > \overline{e}, \end{cases} \quad (50a)$$

$$\hat{e}_k = \frac{\max_{i \in \{k-N_{\mathrm{tuning}}, \dots, k-1\}} |\hat{P}_{\mathrm{tot},i} - P_{\mathrm{req},i}|}{\max\{|\underline{P}_{\mathrm{req}}|, |\overline{P}_{\mathrm{req}}|\}}, \quad (50b)$$

where $\overline{P}_{\mathrm{req}} \equiv \overline{P}_{\mathrm{b}} + \overline{P}_{\mathrm{fc}}$ and $\underline{P}_{\mathrm{req}} \equiv \underline{P}_{\mathrm{b}} + \underline{P}_{\mathrm{fc}}$,

- Battery saving (p_2):

$$p_2(\hat{y}) = \begin{cases} M & \text{if } \hat{\zeta}_{k-1} < \underline{\zeta}, \\ M\left(\frac{\hat{\zeta}_{k-1} - \overline{\zeta}}{\underline{\zeta} - \overline{\zeta}}\right)^2 & \text{if } \hat{\zeta}_{k-1} \in [\underline{\zeta}, \overline{\zeta}], \\ 0 & \text{if } \hat{\zeta}_{k-1} > \overline{\zeta}, \end{cases} \quad (50c)$$

- Fuel cell saving (p_3):

$$p_3(\hat{y}) = \begin{cases} M & \text{if } \hat{m}_{\mathrm{h},k-1} < \underline{m}_{\mathrm{h}}, \\ M\left(\frac{\hat{m}_{\mathrm{h},k-1} - \overline{m}_{\mathrm{h}}}{\underline{m}_{\mathrm{h}} - \overline{m}_{\mathrm{h}}}\right)^2 & \text{if } \hat{m}_{\mathrm{h},k-1} \in [\underline{m}_{\mathrm{h}}, \overline{m}_{\mathrm{h}}], \\ 0 & \text{if } \hat{m}_{\mathrm{h},k-1} > \overline{m}_{\mathrm{h}}, \end{cases} \quad (50d)$$

where m, M, K, and $\overline{e}$ are design parameters; $\hat{y} = \{\hat{y}_i\}_{i=k-N_{\mathrm{tuning}}}^{k-1}$, with $\hat{y}_i = [\hat{\zeta}_i, \hat{m}_{\mathrm{h},i}, \hat{P}_{\mathrm{tot},i}]^\top$, is the simulated closed-loop output trajectory, computed according to the tuning optimization problem (26). Concerning the performance-metric functions, p_1 minimizes when the normalized power tracking error (50b) is close to $\overline{e}$ from below (thus, $\overline{e}$ is a "soft" upper bound on the maximum tracking error); p_2 and p_3 penalize the reduction of supplies with respect to their upper bounds. Therefore, p_1 defines the trade-off between power tracking accuracy and supplies saving; p_2 and p_3 set the consumption priority between battery and fuel cell.

3) Characterization of Closed-Loop Trajectories: When a control system is tasked with regulation (or tracking) of the state x (and output y) towards an equilibrium point (or a trajectory) (x_r, u_r) of the plant (1), enforcing asymptotic stability on the closed-loop system (as defined in, e.g., [20]) allows to achieve such a task, i.e., $\forall\, x_0 \in \mathcal{X}$, $x(t) \to x_r(t)$, $y(t) \to y_r(t)$ for $t \to +\infty$.

In our case study, we address an alternative task, distinct from state regulation/tracking, namely power management. Given a reference power $P_{\mathrm{tot},r}(t)$, our objective is to ascertain the optimal allocation of the delivered powers $\{u_i\}_{i=1}^{n_u}$, ensuring that $P_{\mathrm{tot}}(t) = \sum_{i=1}^{n_u} u_i(t) \to P_{\mathrm{tot},r}(t)$. Such a task cannot be formulated as state regulation/tracking, since our aim is not to guide the closed-loop state trajectories $x(t)$ (i.e., the supplies) towards an equilibrium state or trajectory of the system; rather, the supplies $x(t)$ are permitted to evolve organically under the influence of the optimal delivered powers $u_i(t)$, calculated through the optimal allocation. In this regard, while the supplies $x(t)$ are not requested to track a specific trajectory (or regulate towards an equilibrium), their time evolution contributes in the evaluation of the optimal power allocation, as detailed in Sections V-D1 and V-D2.

For the above reasons, classic stability arguments are not frequently employed within the scope of power management.

In terms of characterization of the closed-loop trajectories, the most relevant property that we can highlight for controlled power systems is that, as reported in Remark 5, the closed-loop supply trajectories $x(t)$ are ensured to remain always bounded within the set $\mathcal{X}$ for any control input $u(t) \in \mathcal{U}$ (for the FCHEV power system, the sets $\mathcal{X}$ and $\mathcal{U}$ are given by Eqs. (36) and (44)).

VI. Simulations and Results

In this section, we validate the NA-MPC strategy through simulations. The nonlinear white-box model of the FCHEV power system (46) is employed as CT plant. The plant is controlled by the neural MPC controller (49), which is adapted in real time by the online metaheuristic tuning strategy described in Section IV. The full control system scheme is depicted in Figures 1 and 3.

Simulations encompass the following aspects:

- capability of NA-MPC to deliver the requested control tasks;
- efficacy of the online tuning strategy;
- comparison of NA-MPC with state-of-the-art techniques for FCHEV power management;
- NA-MPC execution time and real-time feasibility.

A. Preliminaries

1) Implementation Details: The NA-MPC strategy is implemented in MATLAB® 2023b and Python™ 3. The MPC optimal control problem (9) is modelled with YALMIP [32]. The FNNs for the black-box prediction model (13) and the neural MPC controller (49) are constructed and trained using PyTorch. Simulations are performed in MATLAB on a 13th Gen Intel® Core™ i7 CPU at 1.7 GHz.

2) Simulations Data: The following data is shared by all simulations: simulation time $T = 1 \times 10^3$ s; discrete time step $T_s = 1$ s; plant initial state $x_0 = [\zeta_0, m_{\mathrm{h},0}]^\top = [0.6,\ 900\ \mathrm{g}]^\top$.

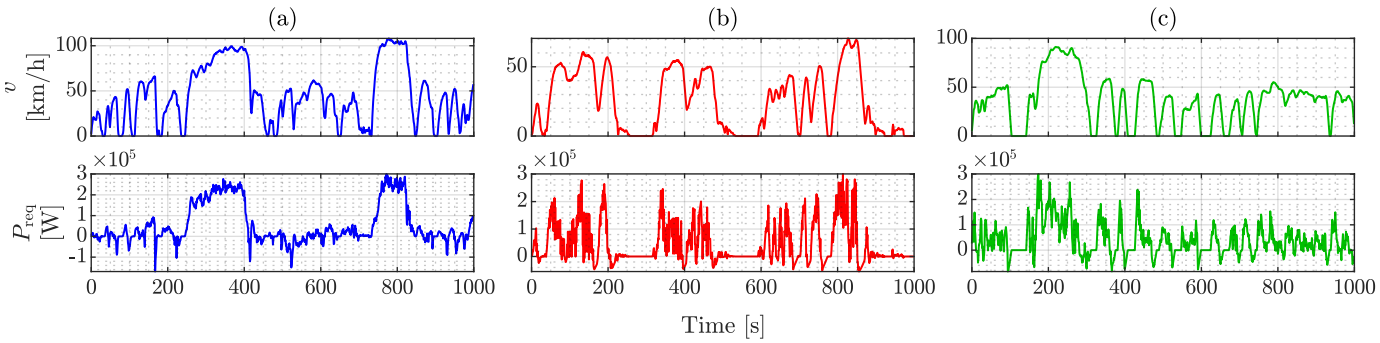

Fig. 4. Driving cycles (v) and corresponding requested power (P_{req}): (a) California Unified Cycle (UC/LA-29); (b) City Suburban Heavy Vehicle Cycle (CSHVC); (c) EPA Federal Test Procedure (FTP-72/UDDS/LA-4).

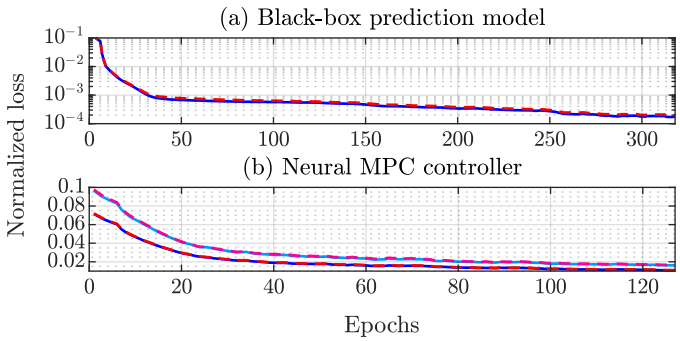

Fig. 5. NA-MPC FNNs training and validation losses over the training epochs. (a) Black-box prediction model ($\mathcal{N}_{f_c}$): training (——); validation (- - -). (b) Neural MPC controller ($\mathcal{N}_\pi$) (reporting the losses on each FNN output and on the sum of the two outputs): training (P_{b}, P_{fc} ——; P_{tot} ——); validation (P_{b}, P_{fc} - - -; P_{tot} - - -).

3) NA-MPC Data: The NA-MPC strategy employs the following data:

- Neural MPC controller (49): $N_p = 10$.
- Performance-metric functions (50): $m = 2.5, M = 10, K = 2$.
- Metaheuristic optimizer (PSO): $\underline{w} = [1, 1, 1, 1, 1]^\top$; $\overline{w} = [10, 10, 10, 100, 100]^\top$; m linearly decreasing from 0.9 to 0.2; $N_{\text{part}} = 10$; $N_{\text{PSO}} = 10$; $c_I = 2$; $c_S = 2$.

Concerning PSO, the values chosen for the parameters m, c_I, and c_S allow, in general, to achieve the best balance between exploration and exploitation [27].

4) Vehicle Requested Power: The vehicle requested power $P_{\text{req},k}$, $k = 0, \ldots, N-1$, $N = \lfloor \frac{T}{T_s} \rfloor$, is generated from a driving cycle – i.e., the sequence of vehicle velocity values over time – by computing the corresponding power through the following vehicle longitudinal model:

$$
\begin{aligned}
&P_{\text{req},k} = F_{\text{tot},k}\, v_k, \quad F_{\text{tot},k} = F_{\text{vis},k} + F_{\text{roll},k} + F_{\text{slope},k} + m a_k, \\
&F_{\text{vis},k} = \frac{1}{2}\rho A_{\text{f}} C_{\text{d}} v_k^2, \quad F_{\text{roll},k} = C_{\text{r}} m g \cos(\theta_k)\, \text{sign}(v_k), \\
&F_{\text{slope},k} = m g \sin(\theta_k), \quad k = 0, \ldots, N-1.
\end{aligned} \tag{51}
$$

where m is the vehicle mass, g the gravity acceleration, v_k the vehicle speed, a_k the vehicle acceleration, ρ the air density, A_{f} the vehicle frontal area, C_{d} the air viscous friction coefficient, C_{r} the rolling friction coefficient, and θ_k the road slope.

Figure 4 reports the three driving cycles employed in the simulations (with the corresponding requested power): the California Unified Cycle (UC/LA-29, Figure 4a), the City Suburban Heavy Vehicle Cycle (CSHVC, Figure 4b), and the EPA Federal Test Procedure (FTP-72/UDDS/LA-4, Figure 4c). These driving cycles are standard – based on real-world vehicle operation – and widely adopted for validation and testing in automotive scenarios [33].

TABLE I
NA-MPC FNNs APPROXIMATION ERRORS.

FNN	Output quantity	Normalized error Min	Average	Max
Black-box prediction model ($\mathcal{N}_{f_c}$)	$\dot{\zeta}$	0.13%	1.28%	6.04%
	$\dot{m}_{\text{h}}$	0.07%	0.84%	2.97%
Neural MPC controller ($\mathcal{N}_\pi$)	P_{b}	0.00%	0.38%	8.07%
	P_{fc}	0.00%	0.16%	8.60%
	P_{tot}	0.00%	0.30%	5.87%

For model (51), the following data is employed: $m = 5 \times 10^3$ kg, $A_{\text{f}} = 8$ m^2, $C_{\text{d}} = 5 \times 10^{-1}$, $C_{\text{r}} = 1 \times 10^{-2}$. For the road slope θ_k, a smooth road profile has been randomly generated, with $\theta_k \in [-3,\ 9]$ deg.

Remark 6. *With model* (51)*, by measuring v_k, a_k, and θ_k through the vehicle on-board sensors, we are also able to compute the value of $P_{\text{req},k}$ to be fed to the neural MPC controller* (49) *at each time instant k.*

B. NA-MPC FNNs Training and Validation

In this section, we report training and validation results for the two FNNs employed by the NA-MPC strategy: the black-box prediction model (13) ($\mathcal{N}_{f_c}$, $\mathcal{N}_g$) and the neural MPC controller (49) ($\mathcal{N}_\pi$).

For the black-box prediction model, since the output function g in Eq. (46) is known, only $\mathcal{N}_{f_c}$ is constructed. The input and output datasets $\mathcal{D}_{\text{in}}$, $\mathcal{D}_{\text{out}}$ are obtained as reported in Section III-A, ensuring that $\mathcal{D}_{\text{in}}$ uniformly spans the set $\Xi = \mathcal{X} \times \mathcal{U}$ defined by the constraints in Eqs. (36) and (44). N_{data} data points are collected, of which N_{train} are used for training and N_{val} for validation. The data points are then subdivided into batches, with $N_{\text{train}}^{\text{batch}}$ and $N_{\text{val}}^{\text{batch}}$ data points each. The FNN is trained with the Adam optimization algorithm [34], employing the following loss function:

$$
L = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \ell(z_i, \hat{z}_i) + \alpha \|p\|_2^2. \tag{52}
$$

In Eq. (52), z and $\hat{z}$ are the target and predicted outputs of the FNN, respectively; ℓ is the L_2 loss, i.e., $\ell(z, \hat{z}) = (z -$

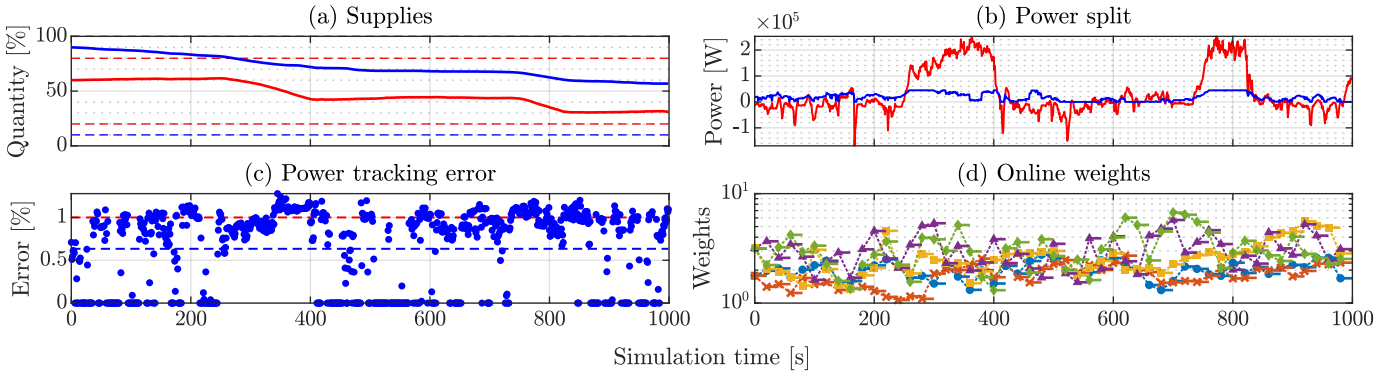

Fig. 6. Simulation of NA-MPC with online tuning (control task 1, driving cycle (a)). (a) Supplies (SOC ——; H_2 mass ——) and corresponding upper-lower bounds (dashed lines). (b) Power allocation (P_b ——; P_{fc} ——). (c) Normalized power tracking error $e_k = |P_{tot,k} - P_{req,k}| / \max_{i\in\{0,\dots,N-1\}} |P_{req,i}|$ (e •; mean(e) - - -; $\bar{e}$ - - -). (d) Optimal MPC weights obtained from the online tuning (w_ζ •; w_{m_h} ×; $w_{P_{tot}}$ ■; $w_{\Delta,\zeta}$ ▲; w_{Δ,m_h} ◆).

TABLE II
CONTROL PERFORMANCE: NA-MPC ONLINE TUNING AND IDEAL NON-CAUSAL TUNING (DRIVING CYCLE (A)).

Task	Max. tracking error	Average tracking error	Battery consumption	Fuel cell consumption	Cumulative consumption
NA-MPC (online tuning)					
1	1.28%	0.63%	72.43%	41.44%	56.93%
2	16.72%	5.00%	48.54%	43.48%	46.01%
3	13.06%	4.10%	31.71%	79.39%	55.55%
4	15.44%	3.33%	83.53%	11.61%	47.57%
NA-MPC (ideal non-causal tuning)					
1	1.18%	0.47%	71.54%	42.07%	56.81%
2	10.54%	3.08%	58.81%	39.77%	49.29%
3	10.85%	3.02%	31.74%	89.45%	60.59%
4	10.28%	2.03%	92.35%	9.57%	50.96%

TABLE III
CONTROL PERFORMANCE OF NA-MPC WITH ONLINE TUNING (DRIVING CYCLES (B) AND (C))

Task	Max. tracking error	Average tracking error	Battery consumption	Fuel cell consumption	Cumulative consumption
Driving cycle (b)					
1	1.89%	0.55%	76.57%	36.75%	56.66%
2	15.39%	3.60%	53.31%	42.14%	47.73%
3	17.18%	3.80%	31.59%	71.91%	51.75%
4	16.94%	3.70%	70.60%	18.71%	44.66%
Driving cycle (c)					
1	1.70%	0.62%	53.53%	30.96%	42.25%
2	15.30%	4.72%	25.98%	37.88%	31.93%
3	15.81%	4.72%	−0.58%	79.71%	39.56%
4	14.33%	4.37%	45.10%	12.12%	28.61%

$\hat{z})^2$; p are the learnable parameters of the FNN, on which L_2 regularization is applied.

Data of the black-box prediction model $\mathcal{N}_{f_c}$ is as follows: $n_{in} = 4$; $n_{out} = 2$; number of layers $L = 3$; neurons for each layer $[N_l]_{l=1}^{L+1} = [4, 10, 10, 2]$; activation functions are the rectified linear unit ($\mathrm{ReLU}(z)$) for layers $l < L$ and the linear function (z) for layer $l = L$; $N_{data} = 50\times10^3$; $N_{train} = 47.5\times10^3$; $N_{val} = 2.5\times10^3$; $N_{train}^{batch} = 47.5\times10^2$; $N_{val}^{batch} = 2.5\times10^2$; errors on collected data $\overline{n}_u = 5\%$, $\overline{n}_y = 5\%$ (see Section III-A).

For the neural MPC controller $\mathcal{N}_\pi$, the datasets are obtained as reported in Section III-C, ensuring that $\mathcal{D}_{in}$ uniformly spans the feasible set of parameters Π in Eq. (49). The datasets are split for training/validation and are subdivided into batches. The FNN is trained with the Adam optimization algorithm [34], employing the following loss function:

$$L = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \lambda\Big(\ell(P_{b,i}, \hat{P}_{b,i}) + \ell(P_{fc,i}, \hat{P}_{fc,i})\Big) + (1-\lambda)\,\ell(P_{b,i} + P_{fc,i}, \hat{P}_{b,i} + \hat{P}_{fc,i}) + \alpha\|p\|_2^2, \quad (53)$$

where $\lambda \in (0,1)$ and ℓ is the L_1 loss, i.e., $\ell(z,\hat{z}) = |z - \hat{z}|$. The loss function (53) penalizes both the error on each output of the neural MPC controller (i.e., P_b and P_{fc}) and the error on the sum of the two outputs (which corresponds to the total delivered power $P_{tot} = P_b + P_{fc}$).

Data of the neural MPC controller $\mathcal{N}_\pi$ is as follows: $n_{in} = 8$; $n_{out} = 2$; $L = 4$; $[N_l]_{l=1}^{L+1} = [8, 100, 100, 100, 2]$; activation functions are the rectified linear unit ($\mathrm{ReLU}(z)$) for layers $l < L$ and the linear function (z) for layer $l = L$; $N_{data} = 500\times10^3$; $N_{train} = 475\times10^3$; $N_{val} = 25\times10^3$; $N_{train}^{batch} = 47.5\times10^2$; $N_{val}^{batch} = 2.5\times10^2$.

For all FNNs, input and output data points are normalized into ranges with zero mean and variance 1. Also, early stopping regularization is employed to prevent overfitting; cross-validation is not employed.

Figures 5a and 5b report the training and validation losses over the training epochs. In both cases, we observe a good convergence behaviour and no overfitting.

To further assess the FNNs emulation capability, we evaluate, for each $\mathcal{N}_\star$, $\star = f_c, \pi$, and for each output component $i = 1, \dots, n_o$, the normalized error $(e_\star(z))_i = |(\mathcal{N}_\star(z) - \star(z))_i| / \max_z |(\star(z))_i|$ for all input data points $z \in \mathcal{D}_{in}$. Results are summarized in Table I, showing satisfactory approximation errors for our purposes.

C. NA-MPC Performance

The NA-MPC strategy with online tuning is tested on four different power management control tasks; such tasks are reported in the following, together with the related parameters for the performance-metric functions in Eqs. (26) and (50):

1) Requested power tracking ($\alpha_1 = 10$, $\alpha_2 = \alpha_3 = 1$, $\bar{e} = 1\%$);
2) Requested power tracking and combined supplies saving ($\alpha_1 = 10$, $\alpha_2 = \alpha_3 = 5$, $\bar{e} = 10\%$);
3) Requested power tracking and battery saving ($\alpha_1 = \alpha_2 = 10$, $\alpha_3 = 1$, $\bar{e} = 10\%$);

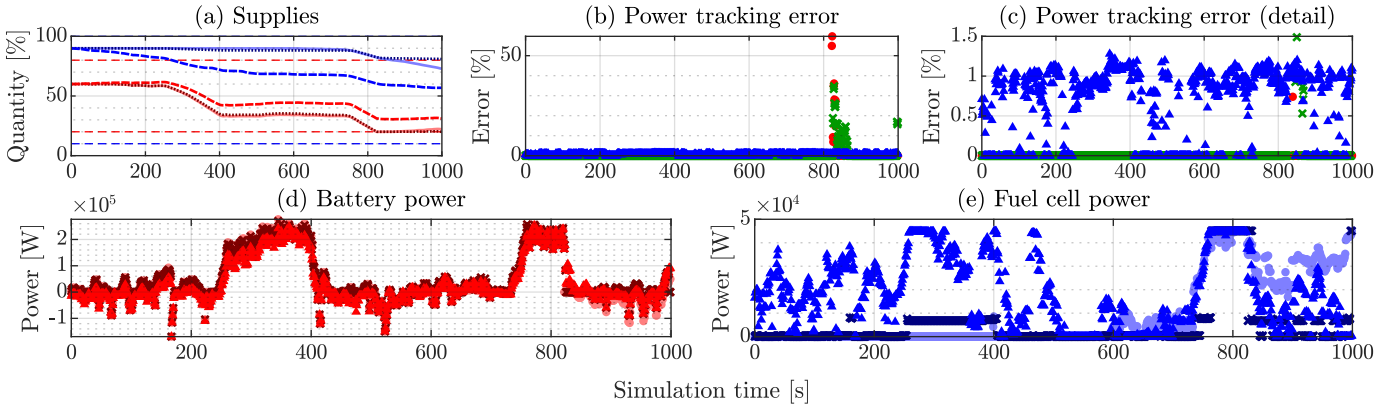

Fig. 7. Comparison of NA-MPC with A-ECMS$^{(1)}$ and FLC$^{(1)}$ (control task 1, driving cycle (a)). (a) Supplies: SOC (NA-MPC ----; A-ECMS ——; FLC ·····), H_2 mass (NA-MPC - - -; A-ECMS ——; FLC ·····), and corresponding upper-lower bounds (dashed lines). (b), (c) Normalized power tracking error $e_k = |P_{\text{tot},k} - P_{\text{req},k}| / \max_{i\in\{0,\dots,N-1\}} |P_{\text{req},i}|$, with detail of the error range $[0, 1.5]\%$ (NA-MPC ▲; A-ECMS ●; FLC ×). (d) Battery power P_{b} (NA-MPC ▲; A-ECMS ●; FLC ×). (e) Fuel cell power P_{fc} (NA-MPC ▲; A-ECMS ●; FLC ×).

TABLE IV
CONTROL PERFORMANCE: NA-MPC AND STATE-OF-THE-ART TECHNIQUES (CONTROL TASK 1, DRIVING CYCLE (A)).

	Max. tracking error	Average tracking error	Battery consumption	Fuel cell consumption	Cumulative consumption
NA-MPC	1.28%	0.63%	72.43%	41.44%	56.93%
A-ECMS$^{(1)}$	59.67%	0.33%	94.70%	21.60%	58.15%
A-ECMS$^{(2)}$	0.00%	0.00%	73.57%	51.10%	62.33%
FLC$^{(1)}$	34.69%	0.50%	99.93%	11.60%	55.76%
FLC$^{(2)}$	0.00%	0.00%	61.70%	65.16%	63.43%

Note. (1): setup 1; (2): setup 2 (see Section VI-D).

4) Requested power tracking and fuel cell saving ($\alpha_1 = \alpha_3 = 10$, $\alpha_2 = 1$, $\overline{e} = 10\%$).

To verify the effectiveness of the online tuning strategy, we compare it with an "ideal" tuning policy. Such an ideal tuning policy is formulated with the same optimization problem (26) and performance-metric functions (50), but, at each tuning instant k, the prior simulation (26b)-(26d) is performed over the whole time interval $[kT_s, (N-1)T_s]$, assuming to know in advance the future power request $P_{\text{req},\{k+1,\dots,N-1\}}$. Thus, such a tuning strategy provides ideal results, but is not realizable in practice, since it does not respect causality.

The initial weights w_0 for the neural MPC controller are set to $w_0 = \frac{w+\overline{w}}{2}$ (see Section VI-A3).

Figures 6a-6d report the simulation results of NA-MPC with online tuning for control task 1, employing driving cycle (a) (i.e., California Unified Cycle, Figure 4a). Simulation results for all control tasks and all driving cycles, including both online tuning and ideal non-causal tuning, are summarized in Tables II and III. Detailed simulation results for driving cycles (b) and (c) can be found in the supplementary material.

We notice that NA-MPC with online tuning consistently attains all the given tasks: the power tracking error stays close from below to the soft upper bound $\overline{e}$, which matches the intended behaviour enforced by the function p_1 in Eqs. (50a) and (50b); also, the trade-off between tracking accuracy and supplies saving is effectively attained, both for combined and selective supplies saving.

Ideal non-causal tuning, as expected, yields more precise and consistent results, owing to its knowledge of the future power request. Nonetheless, online tuning exhibits remarkably similar performance to ideal non-causal tuning, proving the effectiveness of online causal tuning in delivering equally accurate control performance.

D. Comparison of NA-MPC with State-of-the-Art Techniques

We now compare NA-MPC with online tuning against the two most prominent state-of-the-art techniques for HEV power management: Equivalent Consumption Minimization Strategy (ECMS) with online adaptation [8] and Fuzzy-Logic Control (FLC) [6], [7].

Adaptive ECMS (A-ECMS) seeks the optimal powers P_{b}, P_{fc} minimizing, at each time instant k, the instantaneous supply rates [8], i.e.,

$$\begin{aligned} \min_{P_{\text{b}}, P_{\text{fc}}} \quad & \dot{\zeta}(\zeta', P_{\text{b}}) + s \cdot \dot{m}_{\text{h}}(P_{\text{fc}}) \\ \text{s.t.} \quad & P_{\text{b}} + P_{\text{fc}} = P_{\text{req},k}, \\ & P_{\text{b}} \in [\underline{P}_{\text{b}}, \overline{P}_{\text{b}}], \quad P_{\text{fc}} \in [\underline{P}_{\text{fc}}, \overline{P}_{\text{fc}}], \end{aligned} \tag{54}$$

where $\zeta' \equiv \frac{\zeta+\overline{\zeta}}{2}$ makes the cost function depend only on P_{b} and P_{fc} [8]. The functions $\dot{\zeta}(\zeta, P_{\text{b}})$ and $\dot{m}_{\text{h}}(P_{\text{fc}})$ are taken from the FNN black-box model (13). Adaption is performed every N_{tuning} time steps and, at each $k = nN_{\text{tuning}}$, $n \in \mathbb{N}_{\geq 1}$, the parameter s is tuned through the following policy [8]:

$$s_n = \frac{1}{2}(s_{n-1} + s_{n-2}) + K\frac{\zeta_k - \zeta'}{\overline{\zeta} - \underline{\zeta}}. \tag{55}$$

FLC, at each k, takes as inputs the current SOC ζ_k and requested power $P_{\text{req},k}$ and returns as output the FC power $P_{\text{fc},k}$; the battery power is then computed through the equality $P_{\text{b},k} = P_{\text{req},k} - P_{\text{fc},k}$. Following Mamdani's FLC paradigm [7], inputs and outputs are mapped to linguistic variables by trapezoidal membership functions, according to their value (i.e., "negative", "zero", "very low", "low", "medium", "high") [6], [7]. The output linguistic value is obtained from the inputs by means of logic implications. Finally, the output numerical value is computed from the linguistic one via centroid defuzzification.

As also stated in the Introduction, due to their formulations [7], [8], both A-ECMS and FLC are inherently unable to perform an effective trade-off between tracking accuracy and supplies saving (due to the enforced equality $P_{\rm b}+P_{\rm fc}=P_{\rm req}$); therefore, the two techniques are compared with NA-MPC only for task 1 (i.e., requested power tracking). Also, both A-ECMS and FLC are unable to handle constraints on the supply values; thus, when one of ζ, $m_{\rm h}$ goes outside the bounds $[\underline{\zeta},\overline{\zeta}]$, $[\underline{m}_{\rm h},\overline{m}_{\rm h}]$, the corresponding power $P_{\rm b}$, $P_{\rm fc}$ goes to 0, according to Eq. (47b).

A-ECMS is initialized with two sets of values: 1) $s_0 = s_{-1} = 1$, $K = 1.5$; 2) $s_0 = s_{-1} = 2$, $K = 3$. Also FLC is set up with two sets of membership functions and logic rules, where the first set is taken from [7] (adapting the ranges of inputs and output). For both techniques, these two sets are denoted with the superscripts "(1)" and "(2)", respectively.

Simulation results for control task 1 and driving cycle (a) are reported in Figures 7a-7d and Table IV. Additionally, simulation results for driving cycles (b) and (c) can be found in the supplementary material.

We notice that, contrary to NA-MPC, A-ECMS$^{(1)}$ and FLC$^{(1)}$ do not track effectively the requested power, since they fail to meet the SOC lower bound at time $t = 825$ s. With setup (2), both A-ECMS$^{(2)}$ and FLC$^{(2)}$ attain tracking, but they are unable to perform a suitable trade-off between tracking accuracy and supplies saving; indeed, NA-MPC manages to save $\sim$8-9$\%$ more supplies, with a rather minimal increase in the power tracking error. This also shows that A-ECMS and FLC, unlike NA-MPC, are highly sensitive to the choice of initial values/membership functions and logic rules.

E. Execution Time

With reference to the simulations in Section VI-C, the execution time of a single NA-MPC control step is within $[0.0126, 0.4292]$ ms, with average value 0.0248 ms. Concerning online tuning, at each tuning instant the optimal MPC weights are computed in $[108.34, 198.80]$ ms, with average value 119.36 ms. Both execution times are within the discrete time step $T_s = 1$ s, meaning that NA-MPC is real-time feasible.

Remark 7. *Assuming that the neural MPC controller and the online tuning can run in parallel, the discrete time step T_s could be reduced up to the maximum execution time of the NA-MPC control step. This also allows for a finer sampling of the requested power $P_{{\rm req},k}$.*

VII. Conclusions

In this paper, we proposed an advanced optimal control strategy for power management applications, termed Neural Adaptive Model Predictive Control (NA-MPC). NA-MPC provides an optimal power allocation, a multi-objective online adaptation, and ensures real-time feasibility by means of three main ingredients: an online metahuristic tuning algorithm, adapting the MPC cost function weights to attain multiple control objectives concurrently; a neural MPC controller, emulating the original one with universal approximation guarantees; a neural black-box MPC prediction model, identified via only input-output plant data.

While the general formulation and versatility of NA-MPC make it applicable to several power management scenarios, in this work we focused on the case study of fuel cell hybrid electric vehicles (FCHEVs).

The NA-MPC strategy was validated through an extensive simulation campaign, demonstrating its effectiveness in attaining multiple concurrent control objectives at once and showing the excellent performance of the metaheuristic adaptation policy. Also, NA-MPC consistently outperformed the most prominent HEV energy management strategies, namely, fuzzy-logic rule-based control (FLC) and adaptive equivalent consumption minimization strategy (A-ECMS).

The promising results presented in this paper pave the way for several avenues of future research. First, additional power sources will be integrated within the FCHEV powertrain (such as flywheels and supercapacitors). Second, NA-MPC will be augmented within the frame of Hybrid Model Predictive Control, allowing to embed logic conditions and mixed-logical dynamical prediction models in the optimal control policy, with the aim of further improving the controller performance.

Acknowledgments

This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] V. K. Kasimalla and V. Velisala, "A Review on Energy Allocation of Fuel Cell/Battery/Ultracapacitor for Hybrid Electric Vehicles," *Int. J. Energy Res.*, vol. 42, no. 14, pp. 4263–4283, 2018.

[2] N. Sulaiman, M. Hannan, A. Mohamed, E. Majlan, and W. W. Daud, "A Review on Energy Management System for Fuel Cell Hybrid Electric Vehicle: Issues and Challenges," *Renew. Sustain. Energy Rev.*, vol. 52, pp. 802–814, Dec. 2015.

[3] Y. Qiu, T. Zeng, C. Zhang, G. Wang, Y. Wang, Z. Hu, M. Yan, and Z. Wei, "Progress and Challenges in Multi-Stack Fuel Cell System for High Power Applications: Architecture and Energy Management," *Green Energy Intell. Transp.*, vol. 2, no. 2, 2023.

[4] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[5] H. He, R. Xiong, K. Zhao, and Z. Liu, "Energy Management Strategy Research on a Hybrid Power System by Hardware-In-Loop Experiments," *Appl. Energy*, vol. 112, pp. 1311–1317, Dec. 2013.

[6] H. Hemi, J. Ghouili, and A. Cheriti, "A Real Time Fuzzy Logic Power Management Strategy for a Fuel Cell Vehicle," *Energy Convers. Manag.*, vol. 80, pp. 63–70, Apr. 2014.

[7] M. Essoufi, B. Hajji, and A. Rabhi, "Fuzzy Logic Based Energy Management Strategy for Fuel Cell Hybrid Electric Vehicle," in *Proc. Int. Conf. Electr. Inform. Technol.*, 2020.

[8] S. Onori, L. Serrao, and G. Rizzoni, "Adaptive Equivalent Consumption Minimization Strategy for Hybrid Electric Vehicles," in *Proc. ASME Dyn. Syst. Control Conf.*, 2010, pp. 499–505.

[9] W. Zhang, J. Li, L. Xu, and M. Ouyang, "Optimization for a Fuel Cell/Battery/Capacity Tram with Equivalent Consumption Minimization Strategy," *Energy Convers. Manag.*, vol. 134, pp. 59–69, Feb. 2017.

[10] T. Zeng, C. Zhang, Y. Zhang, C. Deng, D. Hao, Z. Zhu, H. Ran, and D. Cao, "Optimization-Oriented Adaptive Equivalent Consumption Minimization Strategy Based on Short-Term Demand Power Prediction for Fuel Cell Hybrid Vehicle," *Energy*, vol. 227, July 2021.

[11] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "MPC-Based Energy Management of a Power-Split Hybrid Electric Vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 3, pp. 593–603, 2011.

[12] J. Chen, A. Behal, Z. Li, and C. Li, "Active Battery Cell Balancing by Real-Time Model Predictive Control for Extending Electric Vehicle Driving Range," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 3, pp. 4003–4015, 2023.

[13] S. Quan, Y.-X. Wang, X. Xiao, H. He, and F. Sun, "Real-Time Energy Management for Fuel Cell Electric Vehicle Using Speed Prediction-Based Model Predictive Control Considering Performance Degradation," *Appl. Energy*, vol. 304, Dec. 2021.

[14] W. Xin, E. Xu, W. Zheng, H. Feng, and J. Qin, "Optimal Energy Management of Fuel Cell Hybrid Electric Vehicle Based on Model Predictive Control and On-Line Mass Estimation," *Energy Rep.*, vol. 8, pp. 4964–4974, Nov. 2022.

[15] D. F. Pereira, F. da Costa Lopes, and E. H. Watanabe, "Nonlinear Model Predictive Control for the Energy Management of Fuel Cell Hybrid Electric Vehicles in Real Time," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3213–3223, 2020.

[16] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[17] L. Calogero, M. Mammarella, and F. Dabbene, "Learning Model Predictive Control for Quadrotors Minimum-Time Flight in Autonomous Racing Scenarios," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1063–1068, 2023.

[18] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural Netw.*, vol. 6, no. 6, pp. 861–867, 1993.

[19] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of Neural Networks by Enforcing Lipschitz Continuity," *Mach. Learn.*, vol. 110, pp. 393–416, Feb. 2021.

[20] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[21] L. Calogero, M. Pagone, and A. Rizzo, "Enhanced Quadratic Programming via Pseudo-Transient Continuation: An Application to Model Predictive Control," *IEEE Control Syst. Lett.*, vol. 8, pp. 1661–1666, June 2024.

[22] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[23] A. Bemporad, "Explicit Model Predictive Control," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. Springer, 2015.

[24] J. W. Daniel, "Stability of the Solution of Definite Quadratic Programs," *Math. Program.*, vol. 5, no. 1, pp. 41–53, 1973.

[25] N. N. Tam, "On Continuity Properties of the Solution Map in Quadratic Programming," *Acta Math. Vietnam.*, vol. 24, no. 1, pp. 47–61, 1999.

[26] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of Metaheuristics*. Springer, 2019.

[27] S. Mirjalili, J. Song Dong, and A. Lewis, Eds., *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*. Springer, 2020.

[28] S. Onori, L. Serrao, and G. Rizzoni, *Hybrid Electric Vehicles: Energy Management Strategies*. Springer, 2016.

[29] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Springer, 2012.

[30] Y. Zhang, C. Huang, H. Huang, and J. Wu, "Multiple Learning Neural Network Algorithm for Parameter Estimation of Proton Exchange Membrane Fuel Cell Models," *Green Energy Intell. Transp.*, vol. 2, no. 1, 2023.

[31] L. Guzzella and A. Amstutz, "CAE Tools for Quasi-Static Modeling and Optimization of Hybrid Powertrains," *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 1762–1769, 1999.

[32] J. Löfberg, "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 284–289.

[33] NREL DriveCAT: Drive Cycle Analysis Tool. National Renewable Energy Laboratory. [Online]. Available: www.nrel.gov/transportation/drive-cycle-tool

[34] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Lorenzo Calogero received the B.Sc. degree (summa cum laude) in Electronics Engineering and the M.Sc. degree (summa cum laude) in Mechatronics Engineering from Politecnico di Torino, Italy, in 2020 and 2022, respectively. He is currently pursuing the Ph.D. degree in Electrical, Electronics and Communications Engineering at Politecnico di Torino. His research interests include model predictive control, optimization, nonlinear systems, and machine learning, with applications in the fields of automotive and autonomous flight.

Michele Pagone received the B.Sc. and M.Sc. degrees in Aerospace/Astronautical Engineering and the Ph.D. degree in Electrical, Electronics and Communications Engineering from Politecnico di Torino, Italy, in 2014, 2016, and 2022, respectively. Currently, he is Assistant Professor in the Department of Electronics and Telecommunications at Politecnico di Torino. He is author of about 20 scientific peer-reviewed publications in international journals and conferences. He has been involved in several national and international projects in collaboration with Italian and European companies. His research interests include nonlinear systems, nonlinear and robust model predictive control, game theory, optimization, Lyapunov stability, with applications in the fields of space flight mechanics, orbital and attitude control, automotive, and energy.

Francesco Cianflone received the Laurea degree in Physics and the Ph.D. degree in Biomedical Technologies. He is currently a Senior Engineering Group Manager at Dumarey Softronix, a company part of the Dumarey Group, based in Turin, Italy, mainly active in the automotive, transportation, and energy sectors. He is the manager of a team of control systems engineers, focusing on the development of advanced control strategies for electronic propulsion control units. His more than 25 years of experience span from R&D to Product Engineering within various companies. He is also author of several publications and patents.

Edoardo Gandino received the M.Sc. degree in Mathematical Engineering and the Ph.D. degree in Mechanics from Politecnico di Torino, Italy, in 2009 and 2013, respectively. He was assigned a research grant from 2009 to 2014, working in the Dynamics and Identification Research Group. He joined the Control Systems department of General Motors Propulsion Torino, now Dumarey Softronix, in 2014, working on diagnostics software functionalities for electronic propulsion control units. Now, he is Technical Leader, working on advanced control strategies for automotive applications. His interests include optimization, model predictive control, diagnostics, and identification of dynamical systems, authoring several papers on international journals and conference proceedings.

Carlo Karam received the B.Sc. degree in Mechanical Engineering from the American University of Beirut and the M.Sc. degree in Mechatronics Engineering from Politecnico di Torino, Italy, in 2020 and 2022, respectively. He is currently a control systems engineer at Dumarey Softronix, working on model predictive control for selective catalytic reduction in diesel engines. His interests include stochastic model predictive control, optimization, and nonlinear systems.

Alessandro Rizzo received the Laurea degree (summa cum laude) in Computer Engineering and the Ph.D. degree in Automation and Electronics Engineering from the University of Catania, Italy, in 1996 and 2000, respectively. Between 2012 and 2024, he was a Visiting Professor with the New York University Tandon School of Engineering, Brooklyn, NY, USA. In November 2015, he joined Politecnico di Torino, Italy, where he is an Associate Professor in the Department of Electronics and Telecommunications and established the Complex Systems Laboratory. His research interests include modeling and control of nonlinear systems, robotics, complex networks and systems. He is the author of one book, two international patents, and more than 200 papers on international journals and conference proceedings.