



Health-aware optimal charging of lithium-ion batteries using deep-neural networks-based explicit constrained model predictive control

Ahmed Shokry ^{a,*} , Mehdi Abou El Qassime ^{a,b}, Antonio Espuña ^c , Eric Moulines ^{a,b}

^a École polytechnique, Paris, France

^b Mines School of Industrial Management, BenGuerir, Morocco

^c Universitat Politècnica de Catalunya, Barcelona, Spain



ARTICLE INFO

Keywords:

Lithium-ion batteries
Health-aware optimal charging
Model predictive control
Explicit control
Machine Learning
Deep Neural Networks

ABSTRACT

The use of Model Predictive Control (MPC) for optimal charging of batteries is attracting attention due to its superiority over empirical charging protocols. But, the intricate nature of physics-based battery models poses a challenge to MPC implementation, necessitating substantial computational resources. Hence, this paper presents a method for explicit MPC based on machine learning (ML) models, applied for optimal battery charging while accounting for linear health constraints. The method uses Deep Neural Networks (DNNs) to construct offline control law that precisely describe the optimal charging current as a function of the battery's state. This DNN-based control law is developed using data generated by solving the MPC problem several times while varying the battery's initial state. Then, the control law is applied online to regulate the charging by cheaply predicting the closed-loop current. The method is numerically validated by its application to two case studies, showing: i) high accuracy in predicting closed-loop charging current (a normalized root mean square error of less than 1.0 %), ii) robustness in handling random initial states of the battery, iii) capability to learn bound and linear constraints directly from the data without any knowledge of their mathematical formulations, achieving a maximum constraint violation of an order of magnitude equal to 10^{-2} , iv) applicability to distinct types of battery models, and v) a reduction in the required computational time compared to traditional MPC, which reaches up to 94.7%, in the lowest-performing testing scenario.

1. Introduction

Rechargeable batteries have become the primary energy storage and supply solution across various sectors, including smart grids, aerospace, and electric transportation (Vidal et al., 2020). In the last two decades, different battery types have been developed, e.g., nickel-cadmium, lead-acid, and lithium-ion batteries (LIBs), where the latter type has proven to strike the balance of performance in terms of power density, light weight, and low self-discharge (Torchio et al., 2016). In general, rechargeable batteries are very complex components involving complicated electrochemical reactions, thermoelectric effects, degradation phenomena, and multiple functional phases, such as discharge, charge, and self-discharge (Gao et al., 2020). Moreover, these complex components usually function under uncertain and dynamic operating (e.g., loads) and environmental (e.g., ambient temperature) conditions. These challenges make batteries critical to the operability and safety of the systems in which they are used. Therefore, the development of battery

management systems (BMSs) has become of great importance to ensure the safety and optimal operation of the battery (Liu et al., 2017; Rosewater et al., 2019). With regard to safety, BMSs use prognostics and health management (PHM) tools to estimate real-time performance indices, enabling battery health monitoring and failure prediction (Vidal et al., 2020; Wang et al., 2013). These indices include the state of charge (SOC), state of health (SOH), and remaining useful life (RUL) of the battery. Regarding the optimal operation (the focus of this article), BMSs aim to optimally manage battery charging to achieve the best charging performance that minimizes the charging time while preserving battery health during long-term operation by adhering to crucial safety and health constraints (Zou et al., 2017; Vidal et al., 2020). Examples of such safety and health constraints include excessive temperature rise and extreme overpotential (Hu et al., 2015; Tian et al., 2021). Excessive heat generation accelerates battery aging and can lead to thermal runaway and explosion, while excessive overpotential triggers side reactions and the formation of solid electrolyte film, which reduces battery capacity and causes power loss (Goldar et al., 2020).

* Corresponding author.

E-mail address: ahmed.shokry@polytechnique.edu (A. Shokry).

Nomenclature	
<i>Problem statement</i>	
$x \in R^m$	State variables
$u \in R^n$	Control variables
F	Model of the system/process
J	Cost function of the control system
$g_l, l = 1, \dots, L$	Set of L constraints imposed on the control systems
$\check{r} \in R^m$	Setpoints
N_p	Prediction horizon of the control system
N_u	Control horizon of the control system
N_c	Constraint horizon of the control system
t	Present time instance/interval
k	Counter for the discrete time intervals
Δu_{t+k}	Change increments of the control variables at time $t + k$
P	Coefficient matrix as a penalty on the terminal state
Q	Coefficient matrix that weights the cost of deviations of the state variables values their setpoints \check{r}
\mathcal{R}	Coefficient matrix as a penalty on the change increments of the control variables
N^{fl}	Maximum number of time intervals within which the system must reach its steady state
$\mathcal{K}=1, \dots, N^{fl}$	Counter for N^{fl}
u_{t+k}^*	Optimal values of the control variables at time $t + k$ calculated by the MPC
$[u_1^*, \dots, u_{\mathcal{K}}^*, \dots, u_{N^{fl}}^*]$	Optimal closed-loop trajectory of the control inputs
$[x_0, \dots, x_{\mathcal{K}}, \dots, x_{N^{fl}-1}]$	Associated closed-loop trajectory of the state variables
<i>Methodology</i>	
n^{tr}	Number of ICs sampled over the state variability range $x_{min} \leq x_t \leq x_{max}$
$n^{tr} \leq n^{tr'}$	Number of feasible ICs with respect to the constraint(s) $g_l(x_t, u_t) \leq 0$. Also, it is the number of control scenarios used for training the DNN
N^{trn}	Length of a training control scenario
n^{ts}	Number of control scenarios used for testing the DNN
\mathcal{D}	Unfolded training dataset used for fitting the DNN-based control law
\mathcal{D}^{ts}	Unfolded testing dataset used to assess the open-loop performance of the DNN-based control law
$\mathcal{D}^{ts'}$	Folded testing dataset used to assess the closed-loop performance of the DNN-based control law
\mathcal{F}_{DNN}^j	The j^{th} DNN-based control law assigned to predict the optimal value of the j^{th} control variable
\mathcal{L}	Number of hidden layers in the DNN
\mathbb{L}_ℓ	The ℓ^{th} hidden layer in the architecture of DNN, with $\ell=1, \dots, \mathcal{L}$
b_ℓ^ℓ	Bias of the ℓ^{th} neuron, localized in the ℓ^{th} layer
a_ℓ^ℓ	Output of the ℓ^{th} neuron, localized in the ℓ^{th} layer
$a_{\ell-1}^{\ell-1}$	Output of the ℓ^{th} neuron in the previous layer $\mathbb{L}_{\ell-1}$
$w_{\ell, \ell-1}, w^\ell$	Weight value assigned to the connection between the ℓ^{th} and the ℓ^{th} neurons located in the layers $\mathbb{L}_{\ell-1}$ and \mathbb{L}_ℓ , respectively
$\mathcal{N}^{\ell, \ell}$	Number of neurons in layer \mathbb{L}_ℓ
\hat{f}	Activation function
θ^j	Parameters of the j^{th} DNN-based control law that include weights and biases
<i>Cast study 1</i>	
R_s	Resistance of the electrode surface
C_s	Capacitance of the electrode surface
R_b	Resistance of the electrode inner bulk
C_b	Capacitance of the electrode inner bulk
V_b	Voltage across C_b
V_s	Voltage across C_s
U	Open-circuit voltage
R_0	Internal resistance
SOC	State of charge
I	Current
V_{tr}	Terminal voltage
f_1	Fifth order polynomial function
$\alpha_0 : \alpha_5$	Constant parameters of the polynomial function f_1
f_2	Exponential function
β_0, β_1 , and β_2	Constant parameters of the exponential function f_2
γ_1 and γ_2	Constant parameters of the health constraint
\mathcal{F}_{DNN}^{NDL}	DNN-based control law
<i>Case Study 2</i>	
q_1	Lithium-ion concentrations in the electrode surface
q_2	Lithium-ion concentrations in the electrode core
\mathcal{B}	Constant ratio between the bulk and the total electrode volumes
I	Current
A	Electrode active area
\mathcal{G}	Inverse of the diffusion time constant
δ	Constant parameter
SOC^+	Cathode state of charge
CSC^+	Cathode critical surface concentration
SOC	Anode state of charge
CSC	Anode critical surface concentration
ρ and σ	Chemistry-dependent constant parameters
V_{tr}	Terminal voltage
ΔU	Open-circuit voltage
U^+ and U^-	Equilibrium potential of the positive and negatives electrodes
$\mathcal{F}_1, \mathcal{F}_2$ and \mathcal{F}_3	Experimentally parametrizable nonlinear functions
η_s^+ and η_s^-	Surface overpotential of the positive and negative electrodes, respectively
R_f	Film resistance
θ^+ and θ^-	Constant parameters combining the kinetic contributions to the overpotentials
T	Battery temperature
T_{amp}	Ambient temperature
m_{cell}	Cell mass
Cp_{cell}	Cell specific heat
\mathbb{R}	Universal gas constant
h	Overall heat transfer coefficient
A	Heat transfer area
Q_{gen}	Heat generated by the cell
C_f	1-C current flux
B_c	Battery capacity
T_{ref}	Reference temperature
Frd	Faraday constant
DS	Entropy change
$\omega_1, \xi_1, \omega_2, \xi_2$	Parameters associated to the linear approximations of the non-linear side reaction constraints
\mathcal{F}_{DNN}^{EHM}	DNN-based control law
CC_1	Computational complexity of the data generation
CC_2	Computational complexity of training the DNN-based control laws
CC_3	Computational complexity of the open-loop evaluation
CC_4	Computational complexity of the online (closed-loop) deployment

Given the fundamental importance of battery charging as the central focus of this work, the remainder of the introduction is structured as follows. Section 1.1 outlines the research motivation, discussing the standard charging protocols commonly employed in BMSs and their inherent limitations. Section 1.2 provides a comprehensive review of recent advancements in battery charging optimization, with a particular emphasis on MPC and ML-assisted control strategies. Section 1.3 synthesizes the limitations and gaps of these existing approaches, categorizing them into four key challenges. Furthermore, it presents an overview of the proposed methodology and its novel contributions.

1.1. Research motivation

Most industrial BMSs rely on standard charging protocols that guarantee a reasonable trade-off between charging time and battery health (Gao et al., 2020). Constant Current-Constant Voltage (CC-CV) is the most common charging protocol (Fig. 1-(a)), in which the battery is charged with a predefined constant current I_{CC} until its terminal voltage rises to a predefined threshold V_{CV} . Then, the charging mode switches to constant voltage, entailing exponential decrease in the charging current (Liu et al., 2017). The charging stops when the current falls below a predefined threshold I_{Cut} . Another common protocol is the Multistage Constant Current-Constant Voltage (MCC-CV) (Fig. 1-(b)), in which the battery experiences an initial high constant current I_{boost} , followed by several charging stages with decreasing constant currents I_{C2}, \dots, I_{CN} . Finally, a constant voltage stage is applied at the end of the charging process (Liu et al., 2018; Zou et al., 2017).

The aforementioned standard charging protocols, such as CC-CV and MCC-CV, are empirically established based on the nominal specifications of the battery and/or qualification tests conducted by the manufacturer (Goldar et al., 2020; Liu, Li, et al., 2018). As a result, these protocols suffer from some limitations: i) they lack dynamic adaptation to real-time changes in battery conditions, ii) they fail to address critical safety and health considerations during charging, and iii) they do not exploit knowledge of the electrical, chemical, and/or thermal dynamics of the battery (Kolluri et al., 2020; Zhang et al., 2017). To overcome these limitations, advanced battery management systems (ABMSs) currently treat the charging process as a MPC problem, where a controller guides the system towards a setpoint determined by the required SOC (Hu et al., 2015). This framework not only ensures fast charging but, more importantly, enables i) the use of available knowledge about battery dynamics represented by precise physics-based models of the battery that can describe its behavior from various perspectives (e.g., electrochemical, electrical, thermal models) (Liu et al., 2017), ii) the consideration of real-time battery conditions through closed-loop implementation, ensuring continuous feedback of the model with measured parameters, and iii) the imposition of necessary operational, safety, and health requirements (i.e., constraints) on the charging process.

1.2. Literature review

To tackle the limitations of standard charging protocols, e.g. CC-CV and MCC-CV, recent research has increasingly focused on optimizing real-time battery charging through MPC. The key differentiating factors among these studies are the choice of the underlying physics-based model used in the MPC framework and the specific constraints incorporated into the optimization process (Rosewater et al., 2019). While most of these works have utilized equivalent circuit models (ECM) (Perez et al., 2017) due to their simplicity, which helps to reduce computational effort, more intricate electrochemical models, such as the equivalent hydraulic model (EHM) (Schorsch et al., 2016), the pseudo 2-dimensional (P2D) (Pathak et al., 2017), and the single particle model (SPM) (Torchio et al., 2016), have also been used.

For instance, Xabier and Trimboli (2015) addressed the charging of LIBs based on MPC formulation relying on a linearized first-order ECM of a battery, while considering a bound constraint on the current, SOC, and on the terminal voltage. Liu et al. (2017) used generalized predictive control (GPC) for optimal charging of LIBs relying on a first-order ECM with a thermal module for temperature prediction, which allowed constraining the maximum increase in battery temperature during charging. In their work, they applied an autoregressive integrated moving average technique to reduce the ECM's complexity, creating a computationally inexpensive data-driven dynamic model of the battery, which is then used in the MPC problem instead of the ECM. Zou et al. (2017) employed a linearized second-order ECM involving a thermal module to control battery charging, imposing bound constraints on the current, SOC, and terminal voltage and temperature. Their method exhibited a minimal computational burden during online implementation. Yan et al. (2011) used a resistor-capacitor ECM coupled with an empirical thermal module based on artificial neural networks (ANNs) for MPC of LIBs, while addressing bound constraints on the SOC and temperature. Goldar et al. (2020) employed an EHM for health-aware MPC of battery charging, incorporating bound constraints on the current and SOC, along with health linear constraints to limit electrode overpotential and thus minimizing side reactions. Romero et al. (2019a) used an EHM in combination with a thermal module, enabling the imposition of bound constraints on the current, SOC, and temperature, besides linear health constraints to mitigate the risk of electrode overpotential. Romero et al. (2019b) proposed a two-layer MPC scheme, where the lower MPC layer finds the optimal charging current satisfying constraints associated with side reactions in the electrodes exploiting either an ECM or EHM, while the upper layer considers a simple linear thermal model to control the battery temperature resulting from the optimal current determined by the lower MPC layer. Pathak et al. (2017) investigated the use of different models in a generic model control (GMC) scheme for LIBs charging, such as thin-film nickel hydroxide electrode (TFNHE) model, SPM, and a reformulated P2D model. Their work considered simple bound constraint on the charging current. Torchio et al. (2017) explored the use of an electrochemical P2D model simplified by Linear Time-Varying (LTV) and Piecewise Affine auto Regressive-eXogenous (PARX) models, which were then integrated into

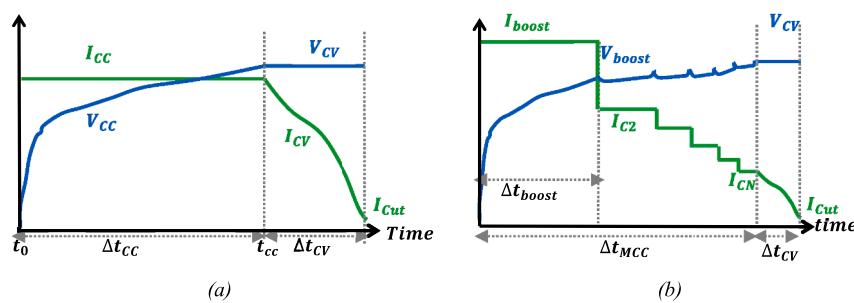


Fig. 1. Schematic representation of common empirical charging protocols (a) CC-CV and (b) MCC-CV.

the MPC scheme for battery charging. They addressed bound constraints on the current, temperature, voltage, and side reaction overpotential. [Kolluri et al. \(2020\)](#) also used a P2D model for MPC of battery charging, considering bound constraints on maximum cell voltage and electrode overpotential. A comprehensive review of battery charging MPC, focusing on the different physics-based models, can be found in ([Rosewater et al., 2019](#)) (Table 1).

Despite the previously reviewed advancements in nonlinear MPC techniques for LIB charging, the high computational demand during online implementation remains a notable challenge. This challenge stems from the need for repeated evaluations of a high-fidelity, highly nonlinear physics-based battery model to solve the optimal control problem (OCP) at each sampling interval ([Tian et al., 2021](#)). For example, [Torchio et al. \(2017\)](#) demonstrated that solving an OCP relying on a P2D battery model take up to 3600 seconds, while the sampling period may be as short as 80 seconds. Some studies ([Torchio et al., 2017; Zou et al., 2017](#)) have proposed simplifying the physics-based battery model to derive approximate linear models, which are then used in

linear MPC schemes. However, this approach requires substantial effort for approximation or linearization. Moreover, linear models are overly simplistic, capturing only basic electrical behaviours such as voltage, current, and SOC, while failing to account for more complex and critical phenomena, including thermal behaviour, health, and degradation of the battery. Consequently, advanced aspects like thermal management, health monitoring, and safety cannot be effectively integrated into the control of LIB charging when relying on linear models.

To tackle this computational challenge of MPC, the application of explicit MPC (E-MPC) methods has proven to be efficient in the field of chemical process engineering ([Charitopoulos et al., 2019; Domínguez & Pistikopoulos, 2011; Falsone et al., 2023; Kis et al., 2021; Pappas et al., 2021; Pistikopoulos et al., 2012; Sun et al., 2019](#)). E-MPC shifts the computational burden from the online to the offline domain. Specifically, control laws are precomputed offline to approximate the optimal control inputs for future sampling periods as explicit functions of the state variables in the current sampling period ([Kouramas et al., 2011](#)). These computationally inexpensive control laws are then used online to

Table 1

Summary of the reviewed works and their gaps with respect to our objectives, highlighted by light grey.

Work	Method	Application	Battery model type	Physical phenomena described by the model			Addressing Online computation time	Constraints type
				Electrical	Thermal	Health		
(Xavier & Trimboli, 2015)	MPC	LIBs charging	linearized 1 st order ECM	Yes	No	No	No	Bound constraints on current, voltage, and SOC
(Liu, Li, & Zhang, 2017)	GPC	LIBs charging	Data-driven model derived from a 1 st order ECM	Yes	Yes	No	No	Bound constraints on the current, SOC, voltage, and temperature
(Zou et al., 2017)	MPC	LIBs charging	Linearized 2 nd ECM with thermal module	Yes	Yes	No	Yes. (low)	Bound constraints on the current, temperature, and voltage
(Yan et al., 2011)	MPC	LIBs charging	Hybrid model (ECM+ ANN)	Yes	Yes	No	No	Bound constraints on SOC and temperature
(Goldar et al., 2020)	MPC	LIBs charging	EHM	Yes	No	Yes	Yes. (affordable)	Bound constraints on current and SOC, besides linear health constraints
(Romero et al., 2019 a)	MPC	LIBs charging	EHM without thermal module	Yes	Yes	Yes	No	Bound constraints on current, SOC, CSC, beside linear health constraints
(Romero et al., 2019b)	MPC	LIBs charging	Either EHM or ECM with thermal module	Yes	Yes	Yes	No	Bound constraints on current, SOC, CSC, beside linear health constraints
(Pathak et al., 2017)	GMC	LIBs charging	Several models, such as SPM, P2D, and TFNHE	Yes	No	No	No	Bound constraint on current
(Torchio et al., 2017)	MPC	LIBs charging	LTV and PARX models derived from P2D model	Yes	Yes	Yes	Yes. (affordable)	bound constraints on the current, temperature, voltage, and side reaction overpotential
(Kolluri et al., 2020)	MPC	LIBs charging	P2D model	Yes	No	Yes	No	Bound constraints on the current, voltage, and lithium plating overpotential
(Charitopoulos et al., 2019; Domínguez & Pistikopoulos, 2011; Falsone et al., 2023; Kis et al., 2021; Pappas et al., 2021; Pistikopoulos et al., 2012; Sun et al., 2019)	MP-MPC	Chemical processes	linear models	N.A.*	N.A.	N.A.	Yes. (very low)	Bound constraints on the control and/or state variables
(Fabiani & Goulart, 2023; Floriano et al., 2022; Karg & Lucia, 2020; Kis et al., 2021; Kumar et al., 2021; Li et al., 2022; Ruiz-Moreno et al., 2021; Shokry et al., 2016a, 2021)	ML-E-MPC	Chemical processes, robots, and mechanical systems	linear and nonlinear models	N.A.	N.A.	N.A.	Yes. (very low)	Bound constraints on the control and/or state variables

cheaply compute the optimal values of the control variables, thereby mitigating the significant computational costs associated with online dynamic optimization in traditional MPC (Domínguez & Pistikopoulos, 2011). Three main approaches have been explored for the development of control laws or optimal control policies, namely multiparametric programming based MPC (MP-MPC), ML-based explicit MPC (ML-E-MPC), and Reinforcement Learning (RL).

The first approach, MP-MPC, provides simple mathematical control laws in the form of piecewise affine functions, which delineate the relationship between the optimal control variables and the system's state/output variables (Chang et al., 2014). While the efficacy of MP-MPC has been evidenced in numerous control studies across various domains (Domínguez & Pistikopoulos, 2011; Pistikopoulos et al., 2012), to the author's knowledge, its application in battery charging control has been limited to only one recent study (Tian et al., 2021; Tian et al., 2019b). In this work, they used a nonlinear double-capacitor (NDC) battery model describing the behaviour of the bulk and surface of the electrode, which allowed the consideration of a health constraint to limit excessive migration of charge between them. However, the application of MP-MPC is limited to small/medium scale systems for which linear discrete-time models are available. This limitation hinders its direct applicability in scenarios necessitating the consideration of complex, large-scale, or highly nonlinear system models like those in battery charging (Rivotti et al., 2012; Shokry et al., 2016b).

The second approach, ML-E-MPC, also called neural control or approximated MPC, is comparatively more recent and relies on training ML regression models to capture the mapping between the control and state variables, representing data-driven control laws (Fabiani & Goulart, 2023; Floriano et al., 2022; Karg & Lucia, 2020; Kis et al., 2021; Kumar et al., 2021; Li et al., 2022; Ruiz-Moreno et al., 2021; Shokry et al., 2016a, 2021). Nevertheless, the ML-E-MPC approach also has some limitations: i) it has mainly been applied to linear dynamic models, and ii) its ability to handle constraints in MPC problems has not been extensively studied, being limited to cases involving simple bound constraints on the control and/or state variables. The ML-E-MPC approach has not yet been fully utilized or adapted for battery charging, despite its demonstrated potential across various application domains (Kis et al., 2021; Maddalena et al., 2020; Shokry et al., 2016b). Very recently, few studies have explored the use of ML-E-MPC for real time optimization of battery charging. Pozzi et al. (2023a) and Pozzi et al. (2023b) compared different types of ML models as control laws for guiding the charging of LIBs. They applied their method to a case study involving a SPM of a battery, incorporating bound constraints on the current, terminal voltage and temperature. Their study highlighted the superior performance of deep feedforward neural networks. It's noteworthy that this type of ML model has been endorsed by the majority of studies in this field (Fabiani & Goulart, 2023; Floriano et al., 2022; Karg & Lucia, 2020; Kumar et al., 2021; Pozzi, Moura, et al., 2023; Ruiz-Moreno et al., 2021; Shokry et al., 2016b). Shokry and Moulines (2022) employed a deep feedforward neural network as a control law to optimally guide the charging of LIBs modelled with a NDC model. They considered bound constraints on the current and terminal voltage, as well as linear health constraint imposing specific relationship among the electrode's surface voltage, bulk voltage, and SOC. To the best of the authors' knowledge, Shokry and Moulines (2022) is the sole work that addressed the performance of ML-based control for LIB in terms of its capacity to handle functional constraints. Addressing this type of constraints is important not only from a general methodological standpoint but particularly for LIBs. This is because battery health and safety are inherently complex criteria and their consideration during charge control may require the inclusion of complicated relationships among multiple system variables, rather than simply applying upper or lower bounds on individual variables (Couto et al., 2021; Goldar et al., 2020; Romero, Goldar, Couto, et al., 2019). Furthermore, these constraints pose a notable challenge for ML techniques, which must not only learn optimal control strategies but also dynamically adapt to these evolving

constrained relationships throughout the charging process.

The third approach, RL, is another machine learning paradigm that has recently been applied to process control. In RL, an "intelligent agent" learns a "policy" that determines actions (e.g., selecting a charging current) to apply within a dynamic "environment" (e.g., the battery or its model). The environment then returns a "reward" that reflects the effectiveness of those actions (e.g., a charging performance index), alongside an updated "state". Typically, the "policy" is represented by a function approximator, such as ANN, that maps the environment's state to specific actions. During training, the agent iteratively adjusts the policy's parameters to maximize cumulative rewards. Once the optimal policy is learned, it functions as a control law, enabling efficient, computationally inexpensive control of the charging process. Wei et al. (2022) developed a policy gradient RL method for battery charging, which has been experimentally validated on a single cell. Initially, a state observer was designed to estimate unmeasurable states using Extended Kalman Filtering combined with a second-order ECM that integrates thermal and aging modules. The RL agent was trained to provide the optimal charging current, guided by a reward function that balances two primary objectives: minimizing the deviation of the SOC from its target and reducing capacity loss. The reward function also incorporates soft penalty terms to address constraint violations, such as limits on voltage and core temperature. Abbasi et al. (2024) developed a proximal policy optimization RL approach for the dual objectives of optimal charging and thermal management of a battery pack. Each cell in the pack is represented by a second-order ECM incorporating thermal and aging characteristics, with aging modelled in terms of capacity loss and internal resistance increase. The pack model aggregates individual cell behaviours while accounting for heat exchange between them, creating a comprehensive environment for the RL agent. The agent was trained to optimize both the charging current and the coolant mass flow rate, guided by a reward function that balances SOC deviation from its target, capacity degradation, and internal resistance, along with soft penalties for exceeding core temperature limit. Yang et al. (2024) applied a deep Q-network RL method for battery pack charging. Each cell in the pack is modelled using a first-order ECM, then the pack model (the environment) is obtained based on a cell-to-pack equalization topology approach. The DRL agent is trained to output the optimal charging current, considering a reward function that includes: (i) charging time, (ii) the balancing effect, representing the inconsistency of SOCs among the cells, and (iii) penalties for over-voltages and exceeded SOC limits. Hao et al. (2024) proposed an actor-critic RL approach for charging optimization. To reduce computational cost, a Gaussian Process (GP) model is used to approximate a Porous Electrode Theory (PET)-based battery model, which is then used as the environment. The reward incorporated terms for SOC deviation from the setpoint and penalties for violating voltage and temperature bounds. Park et al. (2020) applied a policy gradient RL method to an environment represented by the Doyle–Fuller–Newman battery model. The reward function is similarly structured, considering the deviation of the SOC from the desired setpoint and applying penalties for exceeding voltage and temperature maximum limits. Kim et al. (2020) used the soft actor-critic RL method to determine the optimal charging current for a given charging time that can be predefined by the user. The environment included a SPM with electrolyte and thermal dynamics, combined with an aging model for solid electrolyte interface (SEI) layer growth and lithium plating. The reward function is designed to minimize battery aging, described by SEI layer formation, while accounting for bound constraints on voltage, temperature, and lithium plating.

1.3. Gaps, motivations, and novel contributions

The gaps previously detailed in the literature review (in Section 1.2) are further emphasized and consolidated in this section. Although the literature demonstrates a burgeoning interest in control technologies for optimizing LIB charging, it highlights an associated major challenge: the

substantial computational effort required online. Below, the various methods proposed in the literature to address this challenge are condensed and categorized into four distinct approaches, with a focus on highlighting the limitations of each approach, as follows:

- i). Approach 1: Model Simplification – This approach involves simplifying the high-fidelity battery model, for instance, through linearization, to derive a computationally tractable representation suitable for MPC. However, this method requires substantial effort and a deep understanding of the structural intricacies of the high-fidelity model. Furthermore, as the complexity and nonlinearity of the original high-fidelity model increase, the accuracy of the simplified representation may deteriorate. Notably, simplification may omit critical features of battery behavior, such as thermal dynamics, aging effects, and degradation processes, which are inherently complex and nonlinear.
- ii). Approach 2: utilization of MP-MPC techniques (as observed in one study (Tian et al., 2021)) to formulate simple mathematical charging control laws that can be used online to guide charging with minimal computational cost. Nonetheless, this approach encounters limitations similar to the first one (i.e., approach 1), as it relies on the availability of linear discrete-time models with moderate dimensionality.
- iii). Approach 3: The adoption of ML-E-MPC techniques to develop cost-effective ML-based control laws for optimizing the charging process. While this approach holds promise for LIBs, it remains insufficiently explored. Recently, only a few studies have investigated the use of ML-E-MPC for LIB charging (Pozzi, Barbierato, et al., 2023; Pozzi, Moura, et al., 2023; Shokry & Moulines, 2022). Some of these studies (Pozzi, Barbierato, et al., 2023; Pozzi, Moura, et al., 2023) solely addressed simple bound constraints, often without quantitatively assessing how well ML-based control laws adhere to constraint requirements, such as measuring levels of constraint violation. Other works (Shokry & Moulines, 2022) addressed both bound and linear constraints, demonstrating good performance with respect to constraints satisfaction. However, this latter work lacks a comprehensive methodological framework and a robustness assessment across varying constraint configurations, such as constraint tightness, as well as a broader evaluation across different case studies.
- iv). Approach 4: The application of RL techniques to develop optimal policies (i.e., ML models) has shown promise in approximating the relationship between a battery's state and the optimal charging current, which can then serve as a control law for managing the charging process. RL approaches have demonstrated strong performance, particularly advantageous in cases where no physics-based model of the battery is available. This approach has primarily been applied in scenarios with only bound constraints, also without evaluating performance regarding constraint satisfaction.

The motivation for this work stems from the limitations previously identified in the four existing approaches, as well as the untapped potential of the third and fourth approaches for controlling LIBs charging. Given the widespread availability and proven reliability of physics-based battery models, which are now considered the industry standard, this research aims to further advance ML-E-MPC to improve the control and optimization of the charging process. In particular, our work addresses critical gaps and fundamental needs within the current ML-E-MPC framework for battery charging, including: i) demonstrating applicability across different battery models based on different physical principles; ii) effectively managing functional constraints that represent time-varying relationships between system states and control variables; and iii) providing a quantitative evaluation of the performance of ML-based control laws in satisfying constraints at different levels of constraint tightness.

The proposed method involves the offline development of control laws based on DNNs to approximate the relationship between future values of the control variables and the current values of the state/output variables. These DNNs are trained using optimal control and state trajectories obtained by repeatedly solving the constrained MPC problem with different initial state values. The initial state values are uniformly sampled over an overestimated feasible range using Design of Computer Experiments (DOCE) techniques. The resulting DNN-based control laws are then applied in a closed-loop for real-time control of the charging process. To validate the effectiveness of the method, it is applied to two case studies from the existing literature, focusing on MPC of battery charging considering health constraints.

The novel contributions of this work are:

- i). Development of a deep learning explicit MPC (DL-E-MPC) method for optimal real-time charging of LIBs with robust constraint handling capability. Beyond simple bound constraints, this method is designed to handle functional constraints that represent time-varying relationships among battery variables — a capability that, to the authors' knowledge, has not yet been addressed in the existing machine learning-assisted control literature. Such constraints are generally relevant to various control problems but are particularly important for health- and safety-conscious control of LIB charging. This is because ensuring the battery health and safety during charging requires modelling complex relationships between battery variables and not just setting upper or lower limits for individual variables (Couto et al., 2021; Goldar et al., 2020; Romero, Goldar, Couto et al., 2019). The importance of this contribution also lies in the challenge these constraints pose to ML techniques, as ML models must not only learn optimal control actions but also dynamically adapt to evolving constraints during the charging process.
- ii). Demonstrating the robustness of the proposed methodology in managing a range of constraint adjustments and varying levels of constraint tightness. This contrasts with most of the existing machine learning-assisted MPC literature, which generally considers only a single, fixed constraint setting.
- iii). Applying the proposed DL-E-MPC methodology to two distinct case studies on health-aware MPC for battery charging, utilizing two different battery models—an ECM and an EHM—to demonstrate its broad applicability

The subsequent sections of the paper are structured as follows. Section 2 describes the problem statement, providing an introduction to MPC, with a focus on its challenges in the context of battery charging applications. Section 3 outlines the detailed steps of the proposed methodology. In Section 4, we present the application details of the methodology to two distinct case studies. Finally, Section 5 concludes the work by summarizing the key findings of the developed framework and providing a brief overview of the main results.

2. Problem statement

MPC stands out as one of the most popular and efficient control technologies, widely utilized across various fields (Pappas et al., 2021; Rosewater et al., 2019). Its notable superiority lies in its ability to handle nonlinear multivariate systems with intricate multiple-input multiple-output relationships, while accommodating hard constraints on the state and/or control variables (Pistikopoulos et al., 2011; Pistikopoulos et al., 2002). These capabilities make MPC a promising solution for optimal operation of batteries, which are multivariate systems involving complex and highly nonlinear chemical, physical, electrical, and thermal phenomena, while their charging must satisfy numerous safety and health requirements, i.e., hard constraints (Pathak et al., 2017).

The MPC (Fig. 2-(a)) relies on an accurate dynamic model (Eq. (3)) of the real system describing the relationships between the state $x \in R^m$

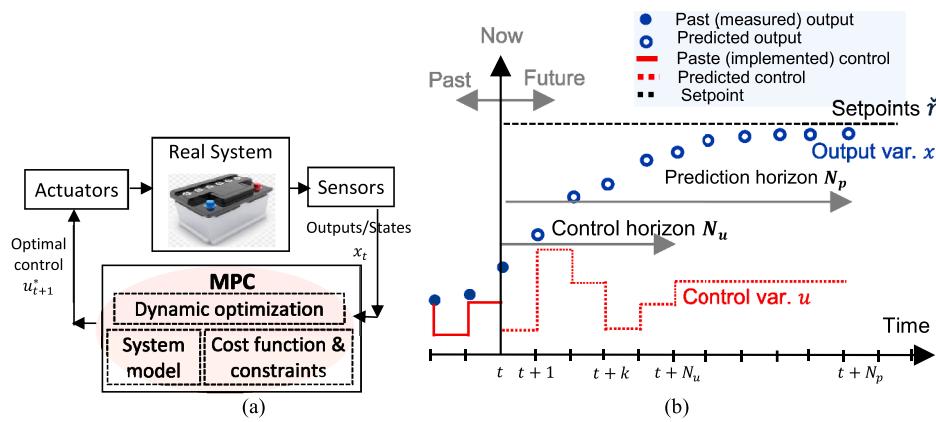


Fig. 2. (a) Illustration of the main components of the MPC technology and (b) representation of addressed open-loop optimal control problem at each sampling period k .

and the control variables $u \in R^v$. It also integrates a cost function J (Eq. (1)) that must be minimized while considering a set of g_l constraints (Eq. (4)) (Domínguez et al., 2010). The maximum time period required for the system to reach its setpoint, $\hat{r} \in R^m$, is discretized into an equal number of sampling periods N^{fl} . For each sampling period $k = 1, \dots, N^{fl}$, an open-loop OCP (dynamic optimization) is solved (Fig. 2-(b)) based on the system model F (Shokry et al., 2016b). First, the dynamic model is updated using real measurements of the state/output variables acquired by sensors, which represent the Initial Conditions (ICs) of the model at the current sampling period (Eq. (2)) (Pistikopoulos et al., 2011). Secondly, the dynamic optimization problem is solved to find the optimal profile of the control variables u_{t+k}^* , where $k = 1, \dots, N_p$ and N_p is the prediction horizon (Rivotti et al., 2012). Then, only the values of the computed optimal control profile corresponding to the first sampling period, u_{t+1}^* , are implemented in the real system. At the end of the sampling period, the state/output variables are measured and their updated values are used as IC to set up the next open-loop OCP, and so on (Pistikopoulos et al., 2011).

The final solution of this MPC problem is defined by the optimal closed-loop trajectory of control inputs $[u_1^*, \dots, u_k^*, \dots, u_{N^{fl}}^*]$ and the associated trajectory of the output/state variables $[x_0, \dots, x_k, \dots, x_{N^{fl}-1}]$.

specific use-case, such as the prediction horizon N_p , the control horizon N_u , and the constraint horizon N_c (with $N_u \leq N_p$ and $N_c \leq N_{p-1}$) (Pistikopoulos et al., 2011)). The parameters also include the coefficients of the cost function: i) P , which is a positive definite $m \times m$ matrix acting as a penalty on the terminal state, ii) Q , which is a positive definite $m \times m$ matrix that weights the cost of deviations of the state variables x_{t+k} from their setpoints \hat{r} , and iii) \mathcal{R} , which is a positive definite $v \times v$ matrix representing penalty for the change increments of the control variables Δu_{t+k} (Tian et al., 2021).

This work addresses the challenge associated with the complexity, high nonlinearity, and/or lack of transparency (i.e., black box) of most available battery models, which can easily lead to two drawbacks from an MPC perspective:

- i). These models compromise the effective application of MPC technology for battery charging due to the enormous computational effort required, which is not economically affordable on the hardware of most commercial BMSs,
- ii). They complicate the application of mathematical E-MPC methods.

Therefore, to tackle these drawbacks, we propose a DL-E-MPC

$$\min_{u_{t+1}, \dots, u_{t+N_p}} J = x_{t+N_p}' P x_{t+N_p} + \sum_{k=1}^{N_p-1} [(x_{t+k} - \hat{r})' Q (x_{t+k} - \hat{r})] + \sum_{k=1}^{N_u} \Delta u_{t+k}' \mathcal{R} \Delta u_{t+k} \quad (1)$$

S.T.:

$$IC = x_{t+k}, k = 0 \quad (2)$$

$$x_{t+k} = F(x_{t+k-1}, u_{t+k-1}), x \in R^m, u \in R^v \quad (3)$$

$$g_l(x_{t+k}, u_{t+k}) \leq 0, l = 1, 2, \dots, L, k = 1, \dots, N_c \quad (4)$$

$$x_{\min} \leq x_{t+k} \leq x_{\max} \quad (5)$$

$$u_{\min} \leq u_{t+k} \leq u_{\max} \quad (6)$$

$$\Delta u_{t+k} = u_{t+k} - u_{t+k-1}, k = 1, 2, \dots, N_u \quad (7)$$

The general formulation of a MPC problem encompasses various parameters, whose values strongly depend on the characteristics of the

methodology for health-constrained battery charging.

3. Methodology

The main four steps of the proposed methodology are summarized in Fig. 3 and elaborated upon in greater detail in Sections 3.1, 3.2, 3.3, and 3.4.

3.1. Step 1: Sampling of the initial state values

The first step in the proposed methodology is the offline generation of input-output training and testing datasets. This step aims to ensure that the training dataset contains: i) information reflecting the behavior of the controller across the different subspaces of the state ICs' domain (Eq. (2)), and ii) information about the controller's behavior in situations when the considered constraints (Eqs. (4)-(6)) are active. The latter point increases the likelihood that the resulting ML-based control laws

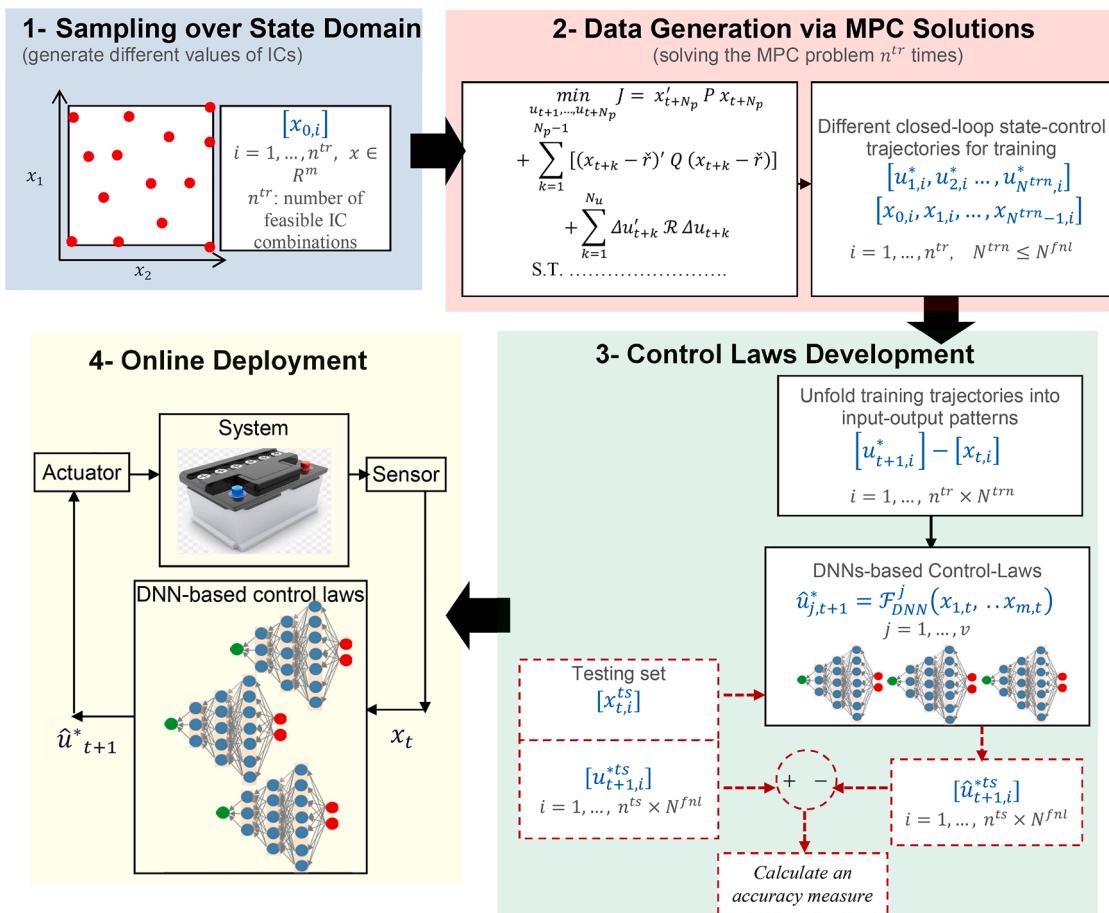


Fig. 3. General schematization of the four steps of the proposed DL-E-MPC methodology. The four steps are explained in detail in Sections 3.1, 3.2, 3.3, and 3.4.

will exhibit robust and feasible performance. To this end, DOCE techniques are used to sample a number of n^{tr} ICs over the permissible range of variability in the system state variables represented by the constraint $x_{min} \leq x_t \leq x_{max}$ (Eq. (5)). Subsequently, the feasibility is further ensured by checking each point in this initial sampling plan against the constraint(s) $g_l(x_t, u_t) \leq 0$ (Eq. (4)), thereby excluding infeasible points and reaching to an appropriate set of $n^r \leq n^{tr}$ combinations of feasible ICs $[x_{0,i}]$, $i = 1, \dots, n^r$, $x \in R^m$. A wide range of DOCE techniques can be found in the literature, such as Latin hypercube sampling, low-discrepancy sequences like the Hammersley technique, space-filling designs, and adaptive sampling (Shokry et al., 2020; Shokry et al., 2021). Selecting a specific DOCE entails balancing the quality of the provided sampling plan (e.g., in terms of its uniformity) and the computational effort required. For further insights into the fundamentals of various DOCE techniques, along with their advantages and limitations, refer to (Garud et al., 2017). In this work, we adopt a hybrid DOCE technique that combines the Hammersley sequence and full-factorial design, because it is able to generate sampling plans characterized by high uniformity at very low computational cost. The Hammersley sequence, a quasi-Monte Carlo sampling technique, employs a quasi-random low discrepancy sequence for sample generation and can be viewed as a natural generalization of the one-dimensional Van der Corput points. For detailed information on the Hammersley sequence and its mathematical formulation consult (Garud et al., 2017). The idea behind this hybrid technique is to leverages the full-factorial design to compensate for the limited ability of the Hammersley sequence to select sample points at the boundaries and vertices of the input space, while taking advantage of the high uniformity of sample points selected by the Hammersley sequence over the bulk of the input space (Shokry et al., 2016).

3.2. Step 2: Data generation via MPC solution

Having obtained the sampling plan of the initial state variables, $[x_{0,i}]$, $i = 1, \dots, n^{tr}$, $x \in R^m$, the MPC problem is solved n^{tr} times exactly as formulated in Eqs. (1)-(7), but only over a finite time horizon $N^{trn} \leq N^{fml}$, where, in each solution, one combination of this plan, $x_{0,i}$, is considered as the IC for the MPC problem (see Eq. (2)). This process enables the acquisition of optimal trajectories for the closed-loop control $[u_{1,i}^*, u_{2,i}^*, \dots, u_{N^{trn},i}^*]$ and the corresponding trajectories of the state variables $[x_{0,i}, x_{1,i}, \dots, x_{N^{trn}-1,i}]$, $i = 1, \dots, n^{tr}$. It is essential to emphasize that the data, comprising the aforementioned trajectories of process states and control inputs, are generated through the application of the mathematical MPC framework, while relying on the complex or high-fidelity process model. Then, the n^{tr} pairs of state-control trajectories are unfolded to constitute the final training set \mathcal{D} that includes input-output patterns in the form of $[x_{t,i}, u_{t+1,i}^*]$, $i = 1, \dots, n^{tr} \times N^{trn}$.

The testing data are generated in the same way as described in Sections 3.1 and 3.2. But in this case, the MPC problem is solved over the entire time horizon N^{fml} to obtain the test dataset \mathcal{D}^{ts} , which includes n^{ts} pairs of optimal state-control trajectories corresponding to different n^{ts} initial state values, i.e., $[x_{0,i}^{ts}, x_{1,i}^{ts}, \dots, x_{N^{fml}-1,i}^{ts}] - [u_{1,i}^{ts}, u_{2,i}^{ts}, \dots, u_{N^{fml},i}^{ts}]$, $i = 1, \dots, n^{ts}$. These test state-control trajectories, \mathcal{D}^{ts} , are also transformed to an unfolded dataset \mathcal{D}^{ts} that involves input-output patterns in the form of $[x_{t,i}^{ts}, u_{t+1,i}^{ts}]$, $i = 1, \dots, n^{ts} \times N^{fml}$. Note that the open-loop accuracy of the control law is assessed using the unfolded test dataset \mathcal{D}^{ts} , while its closed-loop accuracy is evaluated using the folded version, \mathcal{D}^{ts} . Note

that the different n^t combinations of initial state values in the test dataset are randomly selected to avoid bias in the testing procedure and to evaluate the robustness of the control laws in terms of their ability to return the system to its setpoint starting from arbitrary ICs.

The required number of training scenarios, n^r , depends proportionally on i) the number of state variables under consideration within the MPC problem, denoted as m , ii) the degree of nonlinearity of the dynamic model F , and iii) the complexity of the mathematical formulation of the MPC problem itself, particularly regarding the number and intricacy of constraints. For example, these constraints might encompass simple bounds on the control/state variables, as described in Eqs. (4) and (5), or more intricate relationships among these variables, as outlined in Eq. (6). Ultimately, the choice of n^r should strike a delicate balance between the precision of the control laws and the computational efforts necessary for solving the MPC problem and training the control laws using machine learning techniques.

On the other hand, various methods exist for solving the dynamic optimization problem embedded in the MPC scheme, such as Control Vector Parametrization (CVP) (Shokry & Espuña, 2014), direct multiple shooting (Diehl et al., 2006), and collocation methods (Biegler, 2007). In this work, the CVP method is considered due to its straightforward implementation, where only the control variables are discretized as piecewise low-order polynomials and then a nonlinear optimization problem is solved in the space of discretized control variables. The optimization task is executed exploiting the MATLAB solver "fmincon" based on a sequential quadratic programming algorithm. The parameters of the optimization algorithm (e.g., maximum number of function evaluations, tolerance for constraint violation, termination tolerance for the decision variables and objective) are adjusted for each specific use case.

The parameters of the MPC (N_p , N_u , N_c , P , Q , \mathcal{R}) are selected according to the specific case-use characteristics (Pistikopoulos et al., 2011). Additionally, it is important to note that the generic formulation of the objective function J (Eq. (1)) is not necessarily considered in every MPC problem. For instance, certain problems may not incorporate penalties for the terminal state (i.e., the term $x_{t+N_p} P x_{t+N_p}$ in Eq. (1)) and/or for changes in the control variables (i.e., the term $\sum_{k=1}^{N_c} \Delta u_{t+k}^\top \mathcal{R} \Delta u_{t+k}$ in Eq. (1)). Algorithm 1, in the Appendix, shows the pseudocode for the training data generation procedure explained in Steps 1 and 2. Note that the same pseudocode is used to generate the testing dataset, but with different values for the ICs, while the MPC time horizon is set to N^{fnl} .

3.3. Step 3: Development of control laws based on DNNs

The input-output training dataset $[x_{t,i}] - [u_{t+1,i}^*]$, $i = 1, \dots, n^r \times N^{tn}$, $x \in R^m$, $u \in R^v$, is used to develop a set of v ML-based control laws, see Eq. (8), where $\mathcal{F}_{DNN}^j = 1, 2, \dots, v$ are feedforward DNNs for regression (Katz et al., 2020), and θ^j are the parameters of the DNNs. Each one of these control laws approximates the optimal value of one control variable at the next sampling period as a function of the state/output variables values at the current time instance.

$$\hat{u}_{t+1}^* = \mathcal{F}_{DNN}^j(x_t; \theta^j), \quad j = 1, \dots, v \quad (8)$$

Notice that, as a general methodological framework, the number of DNNs (i.e., the number of ML-based control laws) to be developed equals the number of control variables considered in the MPC problem, denoted as v (see Eq. (8)). However, in the specific application of LIBs (see Section 4), the charging current is the sole control variable involved in the MPC problem, i.e., $v=1$. Consequently, only one DNN is developed.

We adopted deep learning among the many available types of ML models, such as GP or Support Vector Regression (Baraldi et al., 2020), because of its powerful and universal capabilities for approximating the behaviour of complex systems (Maddalena et al., 2020). These

capabilities derive from their deep architecture, which consists of multiple nonlinear data processing stages that enable better learning of the hidden hierarchical representation of latent information in the data (Sivaram et al., 2020). Additional information and details regarding the feedforward DNN can be found in the Appendix.

After training the DNN-based control laws, their open-loop response performance is evaluated using the unfolded test dataset $[x_{t,i}^{ts}, u_{t+1,i}^{ts}]$, $i = 1, \dots, n^t \times N^{fnl}$ generated in Section 3.2. Hence an accuracy measure such as the Normalized Root Mean Square Error (NRMSE) can be calculated for each of the v control laws (Eq. (9)).

$$NRMSE = 100 \times \frac{\sqrt{\frac{1}{n^t \times N^{fnl}} \sum_{i=1}^{n^t \times N^{fnl}} (u_{t+1,i}^{ts} - \hat{u}_{t+1,i}^{ts})^2}}{(u_{max}^{ts} - u_{min}^{ts})} \quad (9)$$

where u_{max}^{ts} and u_{min}^{ts} are the maximum and minimum values of the control variables in the testing dataset, respectively.

In the Appendix, Algorithm 2 presents the pseudocode for the development of the DNN-based control law, while Algorithm 3 shows the evaluation of their open-loop accuracy.

3.4. Step 4: Online (closed-loop) deployment

After ensuring that the DNN-based control laws exhibit good open-loop prediction accuracy, they can be integrated into a closed-loop control system to predict the entire control trajectory starting from arbitrary initial conditions of the state variables (Fig. 3).

In real applications, at each sampling period, the DNN-based control law receives, as input, sensor-acquired measurements of the state variables and predicts the optimal future values of the control inputs. These predicted control values are then transmitted to the actuators for implementation in the real system. In this study, due to the unavailability of real battery systems, this scenario is emulated by treating the high-fidelity physics-based model as the real system. Specifically, the physics-based model receives the control values predicted by the DNN-based control law, simulates the system's dynamic behavior, and provides the updated state variable values. Accordingly, the representation of Step 4 in the general schematic diagram (Fig. 3) has been adapted as shown in Fig. 4. Furthermore, it is important to note that the selected case studies are based on physics-based models that have been rigorously calibrated with real experimental data, as documented in the literature. (Tian et al., 2021; Tian et al., 2019; Goldar et al., 2020; Romero et al., 2019). Put differently, once the control laws compute the optimal control values, they are transmitted to the physics-based model to simulate the system's response, and the resulting output/state variables values are then fed-back to the control laws, and so on. Hence, the closed-loop accuracy of the control laws are evaluated using the n^t pairs of closed-loop state-control trajectories generated in Section 3.2, $[x_{0,i}^{ts}, x_{1,i}^{ts}, \dots, x_{N^{fnl}-1,i}^{ts}] - [u_{1,i}^{ts}, u_{2,i}^{ts}, \dots, u_{N^{fnl},i}^{ts}]$, $i = 1, \dots, n^t$, as follows:

- by comparing the control trajectories predicted by the DNNs, $[\hat{u}_{1,i}^{ts}, \hat{u}_{2,i}^{ts}, \dots, \hat{u}_{N^{fnl},i}^{ts}]$, with their exact optimal values, $[u_{1,i}^{ts}, u_{2,i}^{ts}, \dots, u_{N^{fnl},i}^{ts}]$, a closed-loop NRMSE can be individually calculated for each control trajectory. These individual NRMSE values are then averaged over the n^t trajectories, such as:

$$NRMSE = \frac{1}{n^t} \sum_{i=1}^{n^t} \left(100 \times \sqrt{\frac{\sum_{\ell=1}^{N^{fnl}} (u_{\ell,i}^{ts} - \hat{u}_{\ell,i}^{ts})^2}{N^{fnl}}} \right) \quad (10)$$

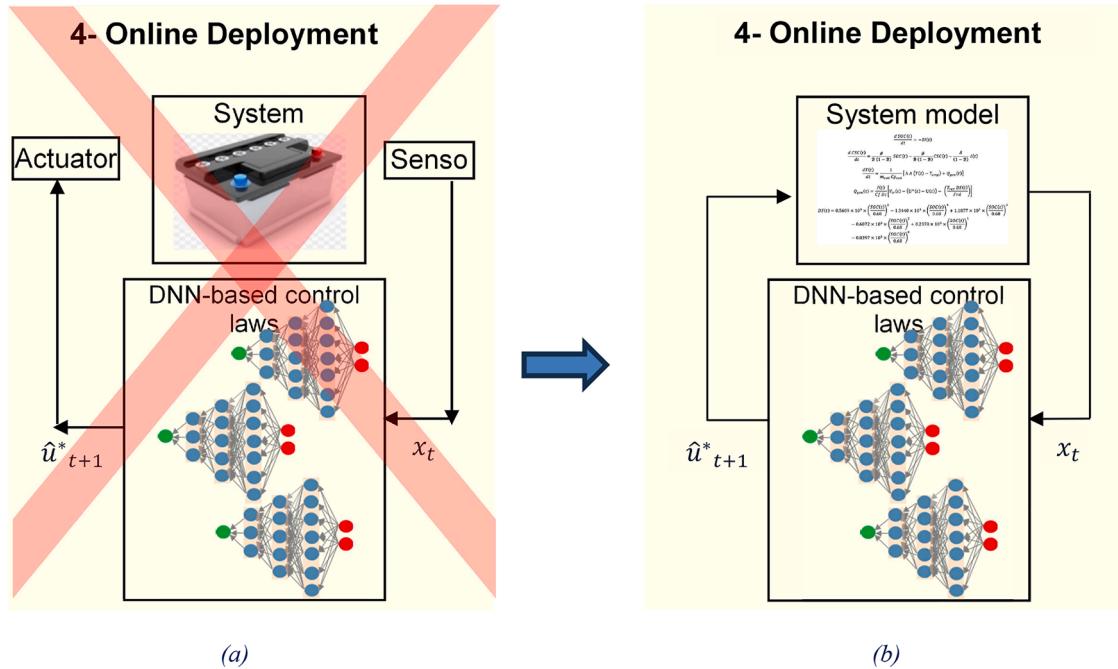


Fig. 4. Schematic representation of Step 4 of the proposed DL-E-MPC methodology for online closed-loop deployment, as described in Section 3.4. (a) General representation for real system application, and (b) the case in the present work, where the high-fidelity model is used to mimic the real system behavior.

- ii). by comparing the system state resulting from the application of the predicted control trajectories, $[x_{0,i}^{ts}, x_{1,i}^{ts}, \dots, x_{N^{ts}-1,i}^{ts}]$, with their exact optimal values, $[x_{0,i}^{*ts}, x_{1,i}^{*ts}, \dots, x_{N^{ts}-1,i}^{*ts}]$, a closed-loop NRMSE can be computed for each state trajectory and then averaged over the n^{ts} trajectories, such as:

$$NRMSE = \frac{1}{n^{ts}} \sum_{i=1}^{n^{ts}} \left(100 \times \sqrt{\frac{\sum_{h=1}^{N^{ts}} (x_{h,i}^{*ts} - \hat{x}_{h,i}^{*ts})^2}{N^{ts}}} \right) \quad (11)$$

where x_{max}^{*ts} and x_{min}^{*ts} are the maximum and minimum values of the state variables in the testing dataset, respectively.

- i). by comparing the predicted n^{ts} control trajectories, $[\hat{u}_{1,i}^{*ts}, \hat{u}_{2,i}^{*ts}, \dots, \hat{u}_{N^{ts},i}^{*ts}]$, and their corresponding states, $[x_{0,i}^{ts}, x_{1,i}^{ts}, \dots, x_{N^{ts}-1,i}^{ts}]$, to the limits of the considered constraints (i.e., Eqs. (4), (5), and (6)), constraint violation metrics can be calculated. These metrics quantify the ability of the DNN-based control laws to learn the constraints imposed on the MPC problem directly from the data, without any knowledge about their mathematical formulations.

Algorithm 4, in the Appendix, presents the pseudocode for the evaluation of the closed-loop accuracy the developed DNNs-based control laws

4. Applications

In this section, two case studies were selected from the literature on health-constrained charging of LIBs using MPC, including a NDC model

and an EHM. These specific cases were selected because: i) they represent two different types/families of battery models and thus can evaluate the applicability and flexibility of the proposed method, ii) they consider relatively complex health constraints to the charging process in the form of linear functions of state variables, and iii) all parameters of the models are publicly available.

4.1. Nonlinear double-capacitors model

The NDC model (Tian et al., 2021) can be viewed as two electrical circuits: the first one (Fig. 5-(a)) represents an electrode that stores electric charge, which is modelled as two capacitors-resistors ($C_b - R_b$, $C_s - R_s$) connected in parallel. The $R_s - C_s$ part represents the electrode surface exposed to the electrode-electrolyte interface and mimics the high-frequency part of the charging/discharging response (Tian et al., 2019a). The $R_b - C_b$ part acts as the electrode's inner bulk and emulates the low-frequency part of the charging/discharging response. The parallel connection allows to simulate the distribution and migration of charge within the electrode. It should be highlighted that C_b stores the majority of the charge relative to C_s , i.e., $C_b \gg C_s$ and $R_b \gg R_s$. Note that this case study, including its mathematical formulation and parameters' value, is adapted from the work of Tian et al. (2021).

The second circuit, Fig. 5-(b), consists of i) a voltage source $U = f_1(V_s)$ which is analogous to an Open-Circuit Voltage, with V_s is the voltage across C_s and ii) a resistor R_0 that represents the electrolyte resistance (i.e., battery internal resistance) and is modelled as a function

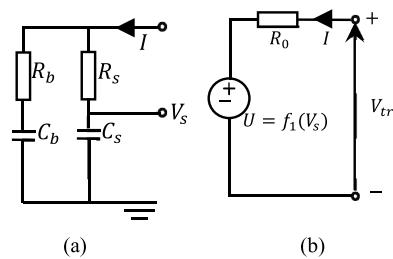


Fig. 5. Schematic representation of the NDC model.

f_2 of the SOC, i.e. $R_0 = f_2(\text{SOC})$. Therefore, the NDC dynamics can be described as follows:

$$\begin{bmatrix} \frac{dV_b(t)}{dt} \\ \frac{dV_s(t)}{dt} \end{bmatrix} = A \begin{bmatrix} V_b(t) \\ V_s(t) \end{bmatrix} + B I(t) \quad (12)$$

$$SOC(t) = \frac{C_b V_b(t) + C_s V_s(t)}{C_b + C_s} \quad (13)$$

$$V_{tr}(t) = U(t) + R_0(t) I(t) \quad (14)$$

$$A = \begin{bmatrix} -1 & 1 \\ \frac{1}{C_b(R_b + R_s)} & \frac{1}{C_b(R_b + R_s)} \\ \frac{1}{C_s(R_b + R_s)} & \frac{-1}{C_s(R_b + R_s)} \end{bmatrix}, B = \begin{bmatrix} \frac{R_s}{C_b(R_b + R_s)} \\ \frac{R_b}{C_s(R_b + R_s)} \end{bmatrix} \quad (15)$$

where $V_b(t)$ is the voltage across C_b , $I(t)$ is the applied current (+ for charging, - for discharging) and $V_{tr}(t)$ is the battery terminal voltage. Also, notice that, in Eq. (13), $C_b + C_s$ represents the maximum SOC of the battery, while $C_b V_b(t) + C_s V_s(t)$ represents the available or stored SOC (Tian et al., 2021).

With respect to the relation $U(t) = f_1(V_s)$, it can be parametrized as a fifth order polynomial, while the relation $R_0(t) = f_2(\text{SOC})$ is described in exponential terms, as follows:

$$U(t) = \alpha_0 + \alpha_1 V_s(t) + \alpha_2 V_s(t)^2 + \alpha_3 V_s(t)^3 + \alpha_4 V_s(t)^4 + \alpha_5 V_s(t)^5 \quad (16)$$

$$R_0(t) = \beta_0 + \beta_1 \exp^{-\beta_2(1-SOC(t))} \quad (17)$$

Table 1 shows the values of the NDC model parameters, which have been experimentally identified in (Tian et al., 2021; Tian et al., 2019).

From Eqs. (12–17), it can be noticed that $V_b(t)$ and $V_s(t)$ determine the entire state of the systems. The variability range of $V_s(t)$ is set between the bounds $[V_{s,\min}, V_{s,\max}] = [0, 1]$, being that at the maximin capacity, $SOC(t) = 100\%$, $V_b(t) = V_s(t) = 1$, while at the minimum capacity, $SOC(t) = 0\%$, $V_b(t) = V_s(t) = 0$.

In this case study (Tian et al., 2021), a MPC problem (Eqs. (18)–(24)) is addressed to optimally charge the battery starting from an initial SOC = 20% and reaching to a setpoint $\hat{r} = 90\%$ of charge (Eq. (19)). The system is required to reach the setpoint within a maximum of $N^{fl} = 150$ sampling periods (60 seconds each). The safety requirements imply bounds on the applied current I_{t+k} (Eq. (20)), an upper bound on the electrode surface voltage $V_{s,t+k}$ (Eq. (21)), and a maximum bound on the battery terminal voltage $V_{tr,t+k}$ (Eq. (22)).

Finally, we consider a health constraint related to the migration of charge from the surface to the bulk of the electrode (Eq. (23)), where the empirical variable $V_{s,t+k} - V_{b,t+k}$ expresses this phenomenon. More precisely, this variable can be considered as equivalent to Lithium-ion (Li-ion) concentration gradients within the SPM dynamics, which can lead to internal stresses, battery heating, and formation of a SEI film, which eventually accelerates battery degradation. The restriction applies that $V_{s,t+k} - V_{b,t+k} \leq \gamma_1 SOC_{t+k} + \gamma_2$, limiting sharp gradients during charging, and the strength of this limitation becomes stronger as the SOC increases where the battery is more likely to be exposed to large concentration gradients (Tian et al., 2021). Note that $\gamma_1 \leq 0$ while $\gamma_2 \geq 0$.

$$\min_{I_{t+1}, \dots, I_{t+N_p}} J = \sum_{k=0}^{N_p-1} [(SOC_{t+k} - \hat{r})' Q (SOC_{t+k} - \hat{r}) + \Delta I_{t+k}' \mathcal{R} \Delta I_{t+k}] \quad (18)$$

S.T.:

$$SOC_0 = 20\%, \hat{r} = 90\% \quad (19)$$

$$0 \leq I_{t+k} \leq 3 \text{ A}, k = 1, \dots, N_p \quad (20)$$

$$V_{s,t+k} \leq 0.95 \text{ V}, k = 1, \dots, N_c \quad (21)$$

$$V_{tr,t+k} \leq 4.2 \text{ V}, k = 1, \dots, N_c \quad (22)$$

$$V_{s,t+k} - V_{b,t+k} \leq -0.04 SOC_{t+k} + 0.08, k = 1, \dots, N_c \quad (23)$$

$$\dot{V}_{b,t}, \dot{V}_{s,t}, SOC_t, V_{tr,t} = F(\dots), \text{ where } F \text{ is the NDC model} \\ N_p = 10, N_u = 2, N_c = 1, Q = 1, R = 0.1 \quad (24)$$

The proposed methodology is applied according to the steps described in Section 3. The first step is to sample over the feasible range of variability of the state variables $[V_{s,\min}=0.0 : V_{s,\max}=0.9, V_{b,\min}=0.0 : V_{b,\max}=0.9]$ using the hybrid DOCE technique to generate the initial state matrix $[V_{s,0,i}, V_{b,0,i}]_{i=1,n^r}, n^r = 400$. In more detail, this sampling plan is composed by 324 samples generated by the Hammersley sequence, and other 76 samples generated by the full-factorial design. Then the MPC problem (Eqs. (18):(24)) is solved 400 times over a horizon of $N^{tm} = 5$ sampling periods to find the optimal trajectories of the closed-loop control $[I_{1,i}^*, I_{2,i}^*, \dots, I_{5,i}^*]_{i=1:400}$ and the corresponding states $[V_{s,0,i}, V_{s,1,i}, \dots, V_{s,4,i}]_{i=1:400}, [V_{b,0,i}, V_{b,1,i}, \dots, V_{b,4,i}]_{i=1:400}$, which are then unfolded into the input-output training matrix $[V_{st,i}, V_{bt,i}]_{i=1:2000} - [I_{t+1,i}^*]_{i=1:2000}$. Fig. 6-(a) shows the input training data while Fig. 6-(b) shows the distributions of the corresponding output training data. Note that the input training data (red and blue circles in Fig. 6-(a)) only cover the lower-right triangle of Fig. 6-(a) because upper-left triangle is not feasible due to the health constraints (Eq. (23)). Moreover, although the initial values of the state variables cover their entire feasible variability domain (blue circles in Fig. 6-(a)), the generated state trajectories (red circles in Fig. 6-(a)) spans only a narrow subregion where the optimality of the dynamic optimization problem embedded in the MPC is achieved.

Using this input-output training dataset $[V_{st,i}, V_{bt,i}]_{i=1:2000} - [I_{t+1,i}^*]_{i=1:2000}$, a DNN-based control law, $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$, is developed to approximate the optimal current to be applied at the next sampling period as a function of the state values at the present sampling period, i.e., $\hat{I}_{t+1}^* = \mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}(V_{s,t}, V_{b,t}; \theta_{\gamma_1=-0.04})$, where $\theta_{\gamma_1=-0.04}$ are the optimal weights and biases of the DNN in this application scenario. The entire training set is randomly split into two subsets: 90% for training and 10% for testing. Then, a cut-and-try procedure is employed to determine the optimal network configurations that maximize the prediction accuracy of the testing subset. This process entails experimental exploration of:

- a) various structures constrained by either two or three hidden layers, with the number of neurons in each layer ranging from 2 to 20,
- b) different types of activation functions for the neurons in the hidden layers, including sigmoid, tanh-sigmoid, or RELU, and
- c) two types of training algorithms: a classical Levenberg-Marquardt Backpropagation (LMB) and a Bayesian Regularization LMB (BR-LMB) that minimizes a loss function combining both the sum of network prediction errors and the sum of network weights.

Ultimately, the optimal configurations (see to Fig. 7) are identified as a structure comprising three hidden layers with 7, 5, and 3 neurons respectively, a sigmoid transfer function, and a BR-LMB training algorithm. Note that Fig. 7 does not show a validation loss curve because the chosen training algorithm does not rely on a validation dataset to regularize the training, but instead minimizes the DNN weights as a form of regularization.

To evaluate the developed control law, a test dataset is generated as described in Section 3.3, which includes a number of $n^s = 30$ optimal closed-loop control profiles with the associated state trajectories, i.e.,

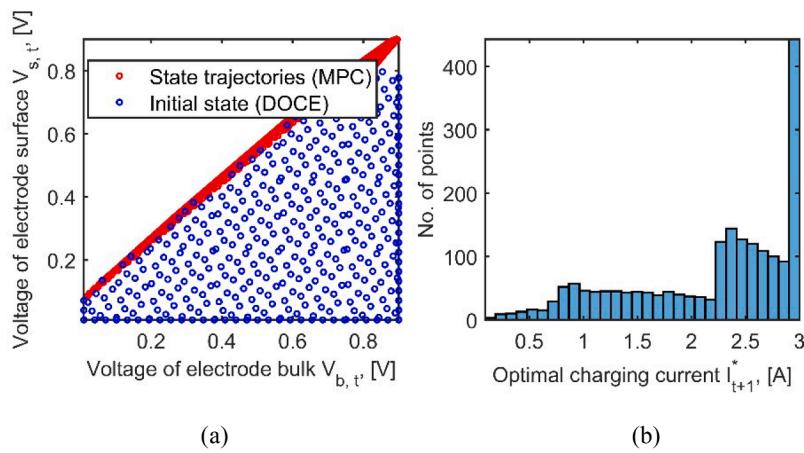


Fig. 6. Unfolded training dataset of the NDC case study: (a) state variables, and (b) distribution of the optimal control variable.

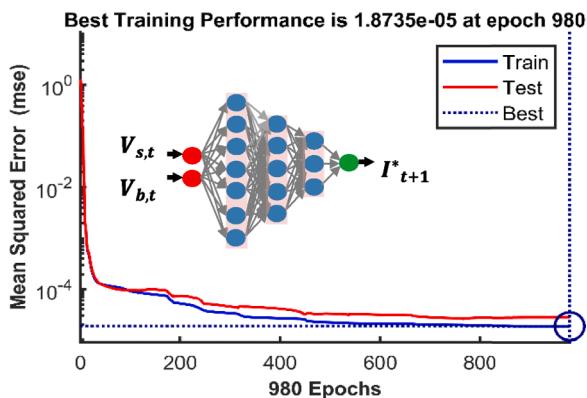


Fig. 7. Best identified structure and loss curves (on training and testing subsets) of the DNN-based control law $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$.

$[I_{1,i}^{ts}, I_{2,i}^{ts}, \dots, I_{150,i}^{ts}]_{i=30}, [V_{s,0,i}^{ts}, V_{s,1,i}^{ts}, \dots, V_{s,149,i}^{ts}]_{i=30}$ and $[V_{b,0,i}^{ts}, V_{b,1,i}^{ts}, \dots, V_{b,149,i}^{ts}]_{i=1:30}$. The unfolded form of the test trajectories, i.e., $[V_{bt,i}^{ts}, V_{bt,i}^{ts}]_{i=1:4500} - [I_{t+1,i}^{ts}]_{i=1:4500}$, is used to evaluate the open-loop accuracy of the control law, while the whole trajectories are used to evaluate its closed-loop performance.

Table 3 shows the computational time required for the offline development of the DNN-based control law $\mathcal{F}_{DNN, \gamma_1=0.04}^{NDC}$, and also presents its promising performance in terms of: i) high open-loop prediction accuracy, ii) high closed-loop prediction accuracy which is well below 1% of NRMSE, iii) high accuracy of the system state (V_b , V_s , V_{tr} and SOC) resulting from the application of the predicted closed-loop control

trajectories compared to that of resulting from the application of the exact closed-loop trajectories, and vi) the very low online computational overhead required to predict the control actions (96.8 s) compared to that of the exact MPC (5226.8 s), resulting in 98.1 % saving of computational time. Also, Fig. 8 evidences the high accuracy of the developed DNN-based control law when compared to the target MPC law.

Table 4 shows that the predicted control trajectories achieve very low average constraints violation (over thirty predicted control trajectories) which demonstrates the ability of the DNN-based control law,

$\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$, to learn the constraints imposed to the MPC problem from the data, which in this case include bound constraints on the charging current (Eq. (20)), upper bound on the terminal voltage (Eq. (22)), and a health constraint (Eq. (23)). It is worth mentioning that the constraint $V_{s,t+k} \leq 0.95$ (Eq. (21)) is not listed in Table 4 since it is never active (neither in the training nor in the testing scenarios), because it is dominated by the required setpoint of the problem. In other words, to reach to a setpoint of $SOC = 90\%$, the maximum required V_s value is well below 0.95 V (i.e., the constraint limit).

Fig. 9-(a and b) shows the thirty predicted closed-loop charging current current trajectories versus the resulting behavior of SOC and the terminal voltage (colored crosses and colored dashed lines) versus their exact values (grey hollow circles and grey solid lines), while Fig. 9-(c-f) shows magnified details. Note that each predicted trajectory is shown in a different color, starts from different initial conditions (represented by a filled circle), and ends at the setpoint or steady state of the system (represented by a yellow star). In Fig. 9, the horizontal solid black lines represent the constraints on the current (Eq. (20)), while the vertical solid black lines represent the setpoint and the constraints on the terminal voltage (Eq. (22)). The figure illustrates the high accuracy of the DNN-based control law $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$ (see Table 2) and its ability to learn the constraints (see Table 3).

Fig. 10-(a) visualizes the behavior of the thirty predicted trajectories

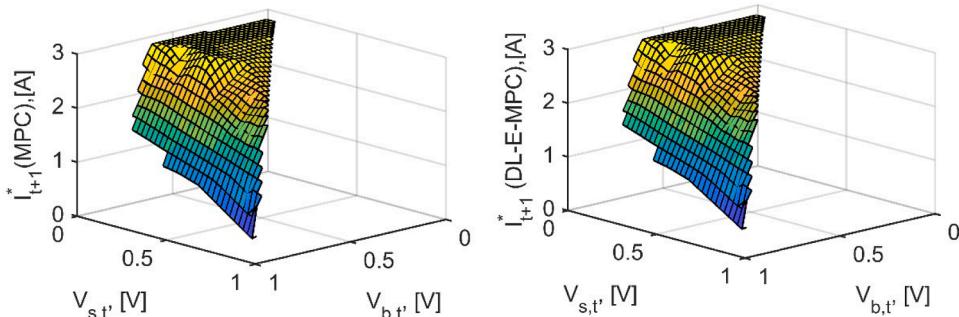


Fig. 8. Exact control law (left) compared to the approximated DNN-based control law $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$ (right).

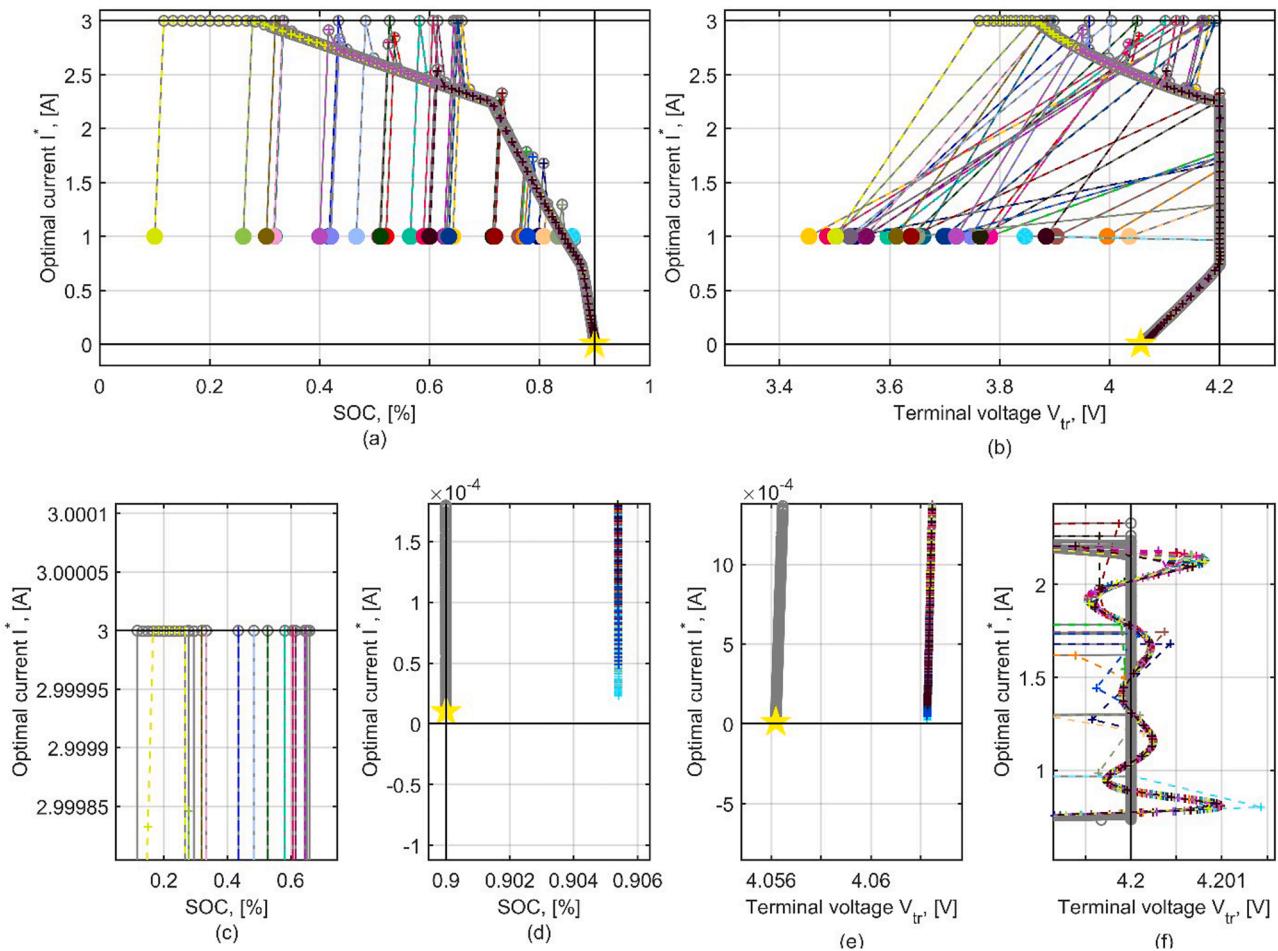


Fig. 9. Predicted vs. exact closed-loop trajectories of the charging current with the resulting SOC (a) and terminal voltage (b), along with different zoomed-in details (c, d, e, f). Colored filled circle are the initial conditions, grey hollow circles and grey solid lines are the exact behaviors, colored crosses and colored dashed lines are the predicted behaviors, yellow star is the setpoint or the steady state, horizontal solid black lines are the constraints on the current, and vertical solid black lines are the setpoint and the constraint on the terminal voltage.

Table 2

Parameters of the NDC model.

C_b [F]	C_s [F]	R_b [Ω]	R_s [Ω]	β_0	β_1	β_2	α_0	α_1	α_2	α_3	α_4	α_5
9913	887	0.025	0	0.09	0.35	-10	3.2	3.041	11.475	24.457	23.536	8.513

Table 3

Computational cost and accuracy of the developed DNN-based control law $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$.

Data generation via MPC	Offline CPU time (s)*		Accuracy (NRMSE (%))				Online CPU time (s)*	Saved time %
	Training	Testing	DNN fitting	Open-loop		Closed-loop		
			Control	Control	Resulting states			
			I^*	I^*	V_b V_s V_{tr} SOC			
$\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$	935.4	5226	12.15	0.90	0.38	0.49 0.48 0.82 0.49	95.8	98.1

*Intel core (TM) i7-8565U CPU@ 18 GHz, 12 GB RAM.

of the charging current with respect to the health constraint $V_{s,t+k} - V_{b,t+k} \leq -0.04 SOC_{t+k} + 0.08$ represented by a solid black line, while Fig. 10-(b) shows an enlarged view. The figure highlights the very good ability of the DNN-based control law, $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$, to satisfy the health constraint starting from different initial conditions represented by the colored filled circles. Overall, Figs. 9, 10, and Table 4 confirm the

very good ability of the DNN-based control law to learn the constraints imposed on the MPC problem from the data only, without prior knowledge of its mathematical formulation.

Fig. 11 provides further insight into the efficient performance of the developed DNN-based control law, $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$. It shows three (out of thirty) examples of the time evolution of the predicted closed-loop

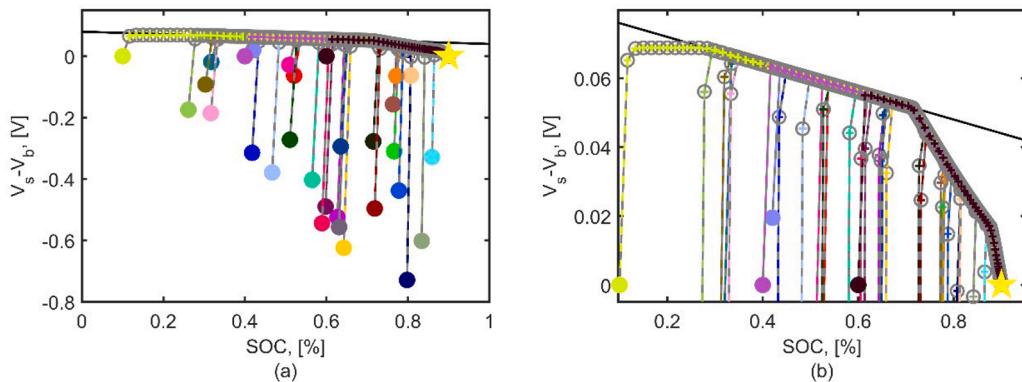


Fig. 10. Behavior of the predicted and the exact closed-loop trajectories of the charging current with respect to the health constraint (a), and zoomed-in view (b). Colored filled circle are the initial conditions, grey hollow circles and grey solid lines are the exact behavior, colored crosses and colored dashed lines are the predicted behavior, yellows star is the setpoint or the steady state, and solid black line is the health constraint $V_{s,t+k} - V_{b,t+k} \leq -0.04 SOC_{t+k} + 0.08$.

Table 4
Average constraint violations of the developed DNN-based control law
 $\mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}$.

Constraint 1	Constraint 2	Constraint 3
$I_t = 3$	$0 - I_t$	$V_{tr,t} - 4.2$
0	0	$V_{s,t} - V_{b,t} - \gamma_1 SOC_t - 0.08$ 3.1×10^{-4}
		1.5×10^{-2}

control trajectories (dashed lines in Fig. 11-(e)) and their resulting states (dashed lines in Fig. 11-(a-d)) compared to their exact values (solid lines). The three scenarios are characterized by the following initial conditions: $[V_{s0} = 0.2, V_{b0} = 0.2]$, $[V_{s0} = 0.4, V_{b0} = 0.4]$, and $[V_{s0} =$

$0.6, V_{b0} = 0.6]$, which correspond to initial state of charges of 20%, 40%, and 60%, respectively. The figure again highlights the high accuracy of the predicted control curves and their ability to accurately satisfy the terminal voltage and current constraints represented by the dashed black lines in Fig. 11-(c) and in Fig. 11-(e), respectively. Fig. 12 shows how the predicted control curves are able to accurately meet the health constraints in the three scenarios.

To further investigate the robustness of the proposed method with respect to the handling of constraints, we applied it several times to the same MPC problem formulation with the same parameters (as in Eqs. (18)-(24)), with the exception that different adjustments to the health constraint $V_{s,t+k} - V_{b,t+k} \leq \gamma_1 SOC_{t+k} + \gamma_2$ are addressed. In more details, three adjustments correspond to three different values of the parameter $\gamma_1 \in \{0.00, -0.07, -0.08\}$ are considered, where lower

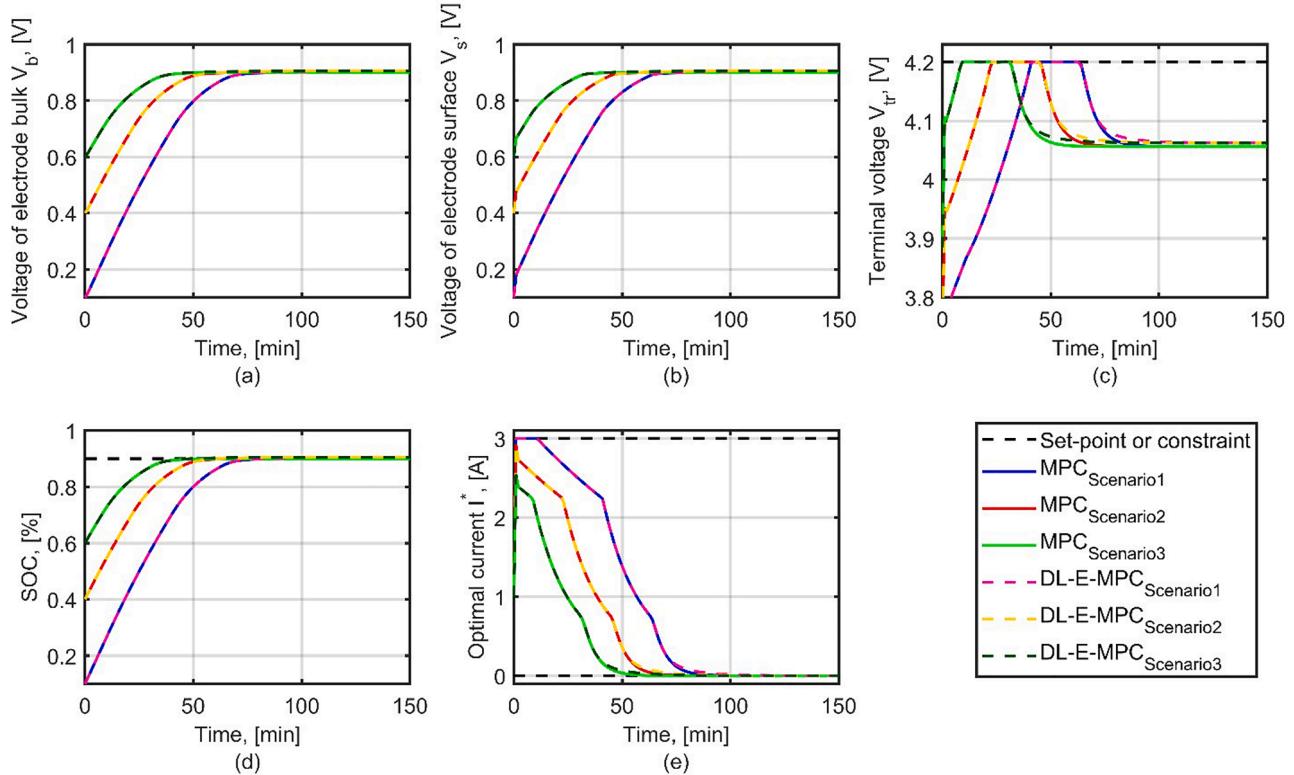


Fig. 11. Closed-loop profiles of charging currents predicted by the DNN-based control law $\hat{I}_{t+1}^* = \mathcal{F}_{DNN, \gamma_1=-0.04}^{NDC}(V_{s,t}, V_{b,t}; \theta_{\gamma_1=-0.04})$ and the resulting states (magenta, yellow and dark-green dashed lines) compared to their exact values obtained by solving the MPC problem (blue, red and green solid lines), considering three different initial conditions.

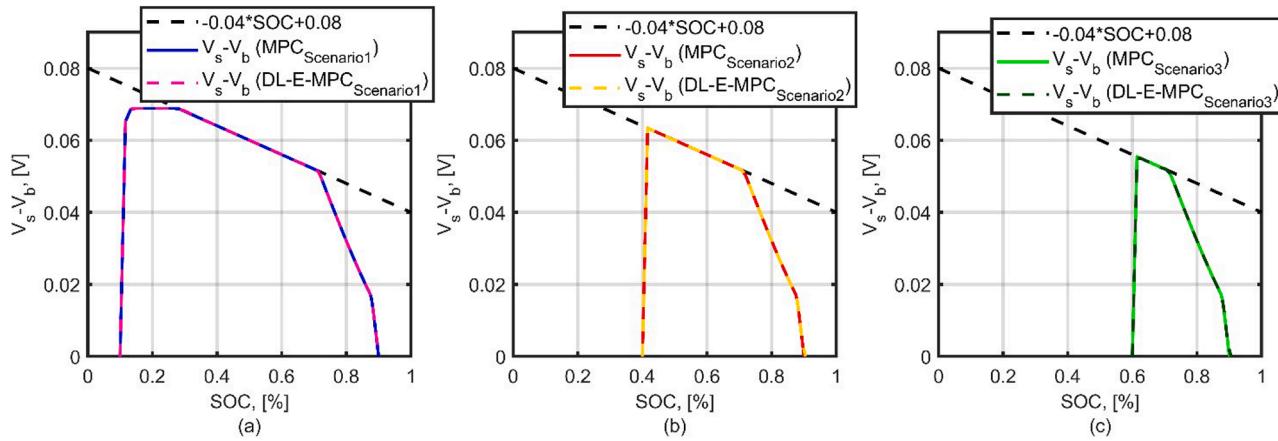


Fig. 12. Performances, with respect to the health constraint $V_{s,t+k} - V_{b,t+k} \leq -0.04 \text{SOC}_{t+k} + 0.08$, of the closed-loop profiles of charging currents predicted the DNN-based control law $\hat{I}_{t+1}^* = \mathcal{F}_{\text{DNN}, \gamma_1}^{NDC}(V_{s,t}, V_{b,t}; \theta_{\gamma_1=-0.04})$ (magenta, yellow and dark-green dashed lines) compared to their exact values (blue, red and green solid lines), considering three different initial conditions.

values of γ_1 yields tighter restrictions on $V_{s,t+k} - V_{b,t+k}$. The method is applied each time following exactly the same steps mentioned above: i) sampling over the feasible range of state variables to generate a set of 400 different values of initial conditions, ii) solving the MPC problem 400 times over the horizon of $N^{trn} = 5$ sampling periods to generate the training dataset, iii) developing DNN-based control law given the same previously mentioned structure and configurations of the network, and iv) evaluating the performance of the DNN-based control laws using a new test dataset containing 30 different closed-loop control profiles.

The computational time required for the offline development of the three control laws, $\mathcal{F}_{\text{DNN}, \gamma_1=0.0}^{NDC}$, $\mathcal{F}_{\text{DNN}, \gamma_1=-0.07}^{NDC}$, $\mathcal{F}_{\text{DNN}, \gamma_1=-0.08}^{NDC}$ and the numerical metrics expressing their very good performance are shown in Table 5, while the average constraint violation, in the three application settings is shown in Table 6. Fig. 13 shows a comparison between the exact control laws (top) and the devolved DNN-based control laws (bottom), corresponding to the three different values of γ_1 . The figure illustrates that decreasing the value of γ_1 leads to more health-preserving control laws, implying the application of relatively lower charging currents. The figure and the two tables illustrate the high accuracy of the developed DNN-based control laws and the high robustness and adaptability of the proposed methodology in terms of capturing the different control behaviors resulting from the change in the tightness of the health constraint.

Fig. 14 visualizes the performance of one DNN-based control law in the application setting where $\gamma_1 = -0.07$ (i.e., $\mathcal{F}_{\text{DNN}, \gamma_1=-0.07}^{NDC}$), by comparing the thirty predicted closed-loop profiles of charging current and the associated SOC and terminal voltage with their exact values, while Fig. 15 visualizes their performance with respect to the health

Table 6

Average constraint violations in the three application settings characterized by different values of γ_1 (0.00, -0.07 and -0.08).

Average constraint violations			
$V_{tr,t} - 4.2$	$V_{s,t} - V_{b,t} - \gamma_1 \text{SOC}_t - 0.08$	$I_t - 3$	$0 \leq I_t$
$\mathcal{F}_{\text{DNN}, \gamma_1=0.0}^{NDC}$	1.76×10^{-4}	00.00	0 0
$\mathcal{F}_{\text{DNN}, \gamma_1=-0.07}^{NDC}$	1.0×10^{-3}	1.5×10^{-5}	0 0
$\mathcal{F}_{\text{DNN}, \gamma_1=-0.08}^{NDC}$	0.0	1.3×10^{-5}	0 0

constraint $V_{s,t+k} - V_{b,t+k} \leq -0.07 \text{SOC}_{t+k} + 0.08$.

Fig. 16 shows, just as example, one testing scenario that corresponds to an initial condition of $[V_{s0} = 0.1, V_{b0} = 0.1]$. The Figure compares the closed-loop profiles of charging current predicted by the three control laws $\mathcal{F}_{\text{DNN}, \gamma_1=0.0}^{NDC}$, $\mathcal{F}_{\text{DNN}, \gamma_1=-0.07}^{NDC}$, and $\mathcal{F}_{\text{DNN}, \gamma_1=-0.08}^{NDC}$ (dashed lines in Fig. 16-(e)), corresponding to the three values of $\gamma_1 \in \{0.00, -0.07, -0.08\}$, and their respective states (dashed lines Fig. 16-(a-d)) compared to the exact values obtained by solving the MPC problems (solid lines). In detail, Fig. 16-(c and e) and Fig. 17 confirm the capabilities and robustness of the DNN-based control laws to learn from the data without knowledge of the mathematical formulations/characteristics the different tightness of the constraints imposed on the MPC problem (dashed black lines).

Note that the DNN-based control laws provide closed-loop current profiles that can adjust as the health constraint become tighter. This adjustment can be seen in Fig. 16 -(e): as γ_1 decreases, the "front" of the charging current profile tends to get lower, while the "tail" gains height and length, resulting in a longer charging time (see Fig. 16-(d)) (Tian

Table 5

Computational cost and accuracy of the developed DNN-based control laws in the three application settings characterized by different values of γ_1 (0.00, -0.07 and -0.08).

	Offline CPU time (s)*		Accuracy (NRMSE (%))						Online CPU time (s)	Saved time %			
	Data generation via MPC		DNN fitting		Open-loop		Closed-loop						
	Training	Testing	Control	Control	Resulting states								
					I^*	I^*	V_b	V_s	V_{tr}	SOC			
$\mathcal{F}_{\text{DNN}, \gamma_1=0.0}^{NDC}$	960.5	2323.7	7.55	0.40	0.16	0.10	0.10	0.20	0.10	50.0	97.8		
$\mathcal{F}_{\text{DNN}, \gamma_1=-0.07}^{NDC}$	3554.0	1684.2	15.83	0.4	0.20	0.22	0.21	0.38	0.22	89.2	94.7		
$\mathcal{F}_{\text{DNN}, \gamma_1=-0.08}^{NDC}$	1022.1	2352.3	7.15	0.57	0.26	0.21	0.21	0.41	0.21	66.0	97.2		

*Intel core (TM) i7-8565U CPU@ 18 GHz, 12 GB RAM.

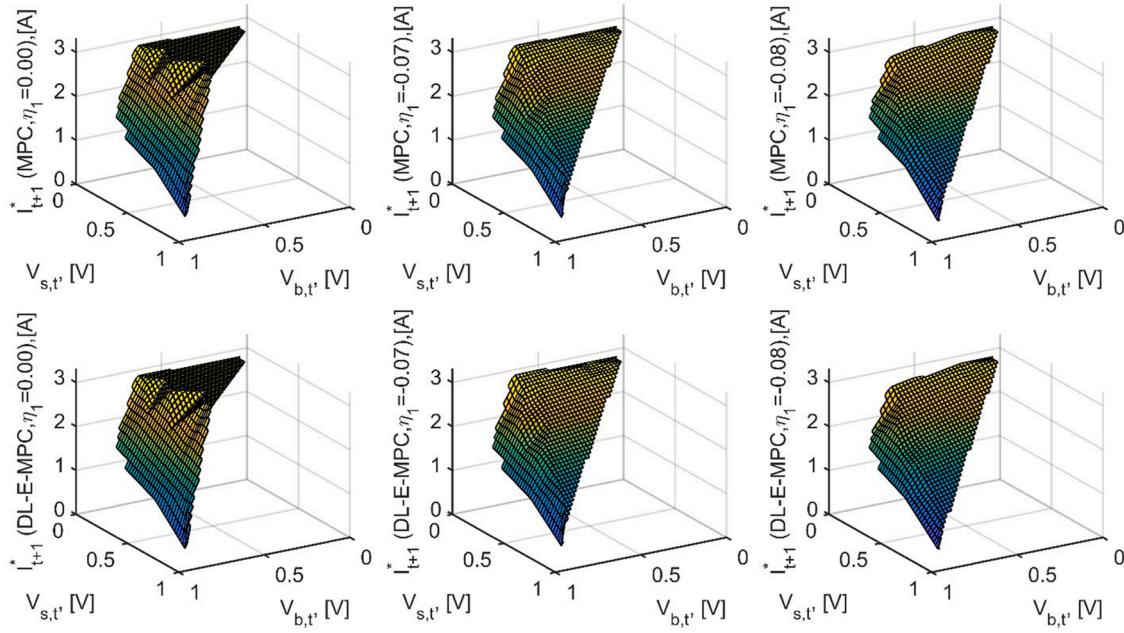


Fig. 13. Exact control laws (top) compared to the DNN-based control laws (bottom) for different values of γ_1 (from left to right: 0.00, -0.07 and -0.08).

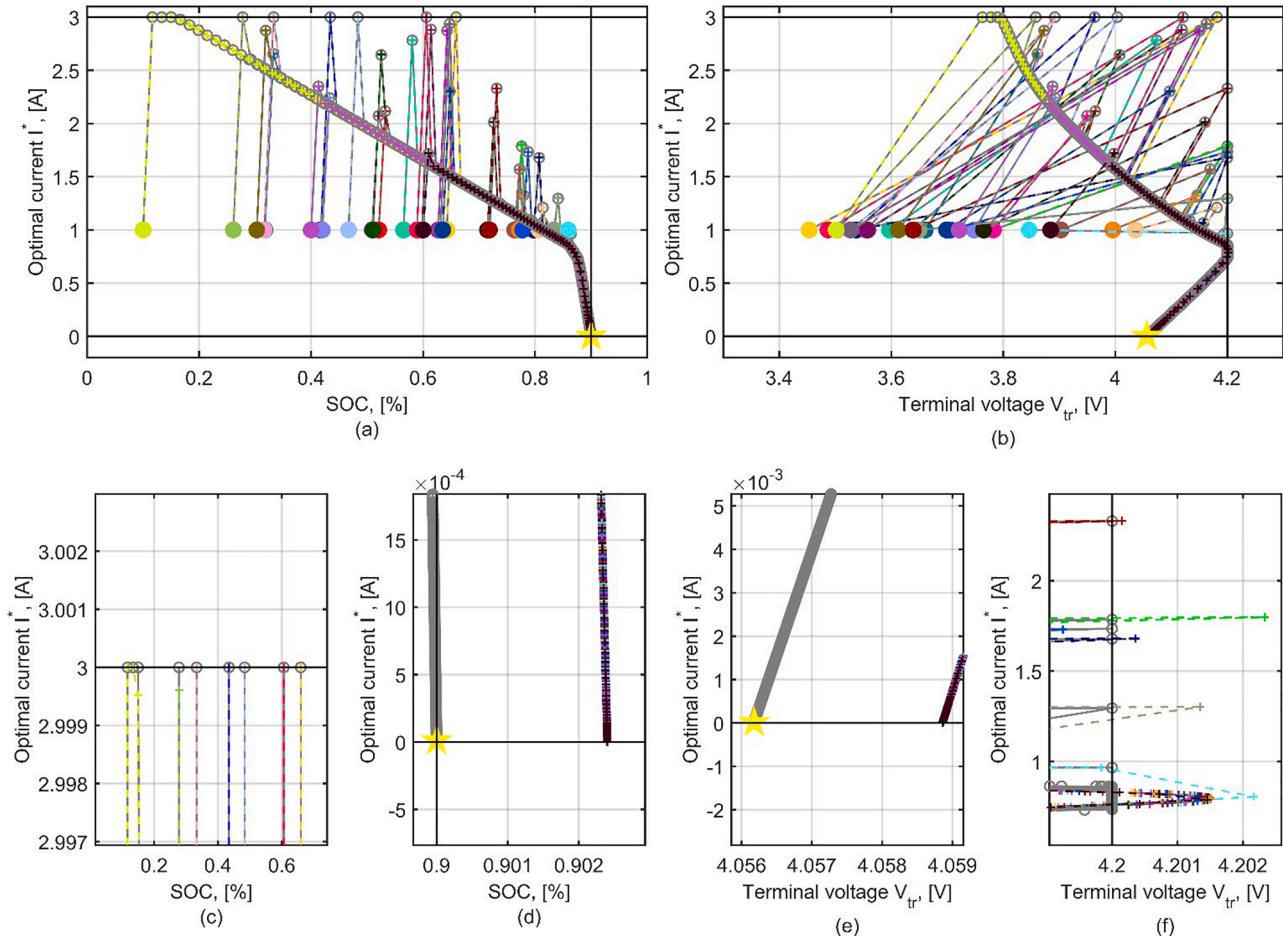


Fig. 14. Predicted vs. exact closed-loop trajectories of the charging current with the resulting SOC (a) and terminal voltage (b), along with different enlarged details (c, d, e, f) in the application setting where $\gamma_1 = -0.07$. colored filled circle are the initial conditions, grey hollow circles and grey solid lines are the exact behaviors, colored crosses and colored dashed lines are the predicted behaviors, yellows star is the setpoint or the steady state, horizontal solid black lines are the constraints on the current, and vertical solid black lines are the setpoint and the constraint on the terminal voltage.

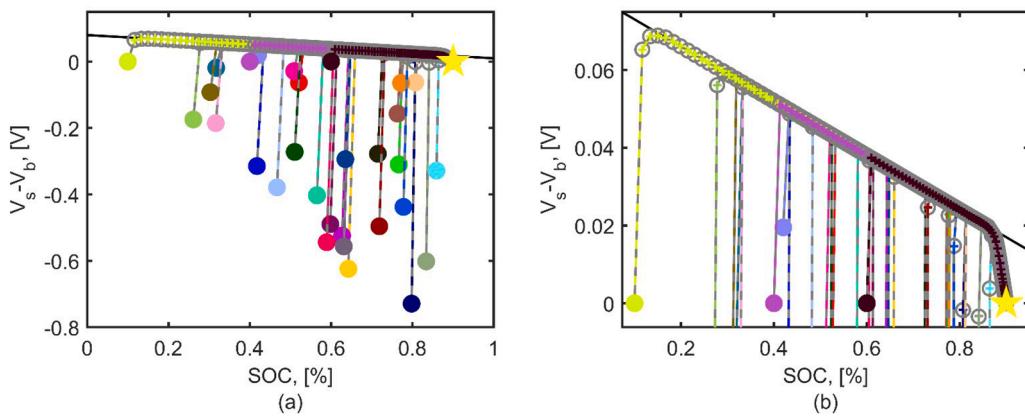


Fig. 15. Behavior of the predicted and the exact closed-loop trajectories of the charging current with respect to the health constraint (a), and zoomed-in view (b). Colored filled circle are the initial conditions, grey hollow circles and grey solid lines are the exact behavior, colored crosses and colored dashed lines are the predicted behavior, yellows star is the setpoint or the steady state, and the solid black line is the health constraint $V_{s,t+k} - V_{b,t+k} \leq -0.07 SOC_{t+k} + 0.08$.

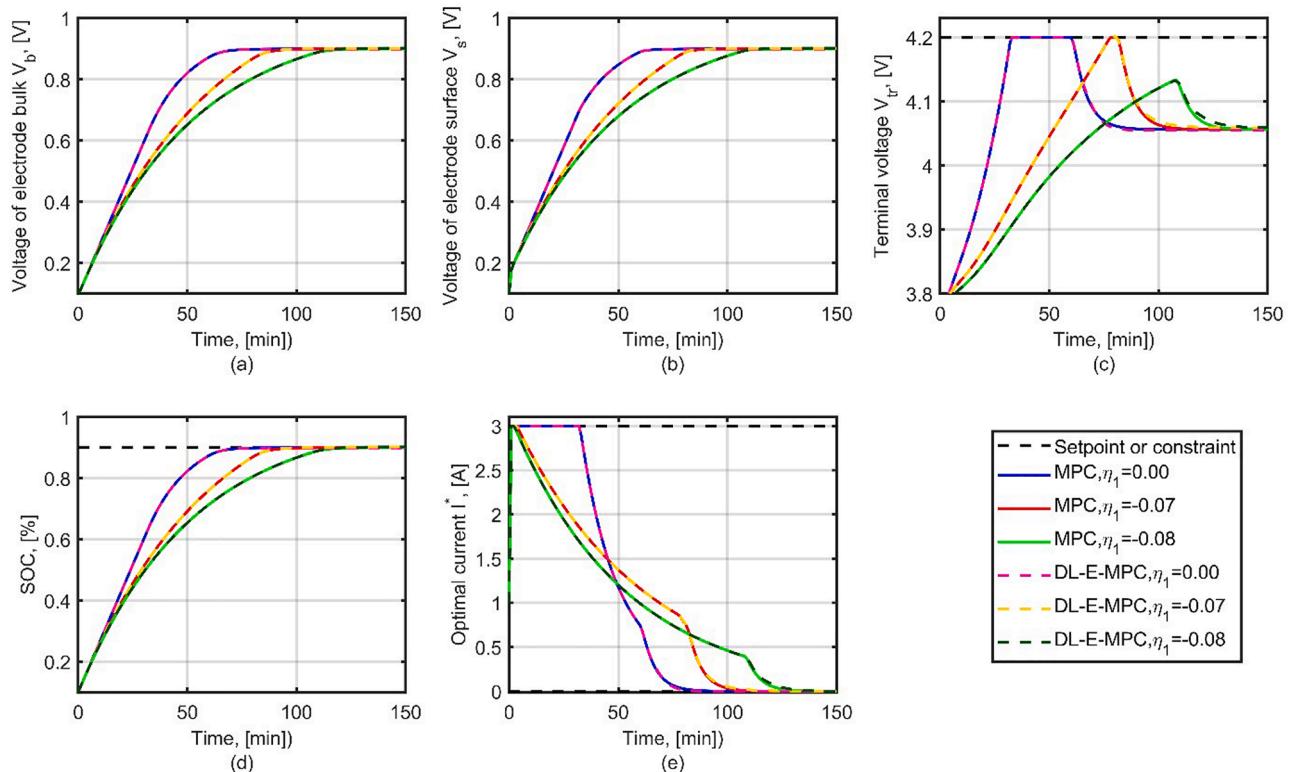


Fig. 16. Closed-loop profiles of charging current predicted by the DNN-based control laws and the resulting states (magenta, yellow and dark-green dashed lines) compared to their exact values (blue, red and green solid lines) in the three application settings characterized by different values of $\gamma_1 \in \{0.00, -0.07, -0.08\}$ and considering the same initial conditions.

et al., 2021). This is a consequence of imposing tighter restrictions on the voltage difference between the electrode bulk and surface $V_{s,t+k} - V_{b,t+k}$ (via decreasing the value of γ_1), which forces the controller to select lower charging currents, especially as the SOC approaches its setpoint, since the constraint becomes more stringent as the size of SOC increases (see the right hand side of the constraint $\gamma_1 SOC_{t+k} + \gamma_2$). The effect of the predicted conservative profiles of current on the charging process can be seen in Fig. 16-(c and d), where the terminal voltage and the SOC rise tardily as the value of γ_1 decreases.

4.2. Equivalent hydraulic model

The second case study, including its mathematical formulation and

parameter values, is adapted from the works of (Goldar et al., 2020; Romero et al., 2019). This case considers an EHM which is a reduced-order electrochemical battery model, and can be seen as an equivalent to a simplified SPM, where, in the latter, each electrode is represented by a spherical particle. The EHM describes each spherical particle (i.e., an electrode, see Fig. 18-(b)) by two state variables: i) the SOC which represents the total Li-ion concentration in the whole electrode and ii) the Critical Surface Concentration (CSC) that describes the Li-ion concentration at the surface layer of the electrode (Romero et al., 2019). The dynamics between the two states (i.e., the charge transfer between the bulk and the surface of the electrode) is mimicked by the hydraulics in two interconnected tanks, see Fig. 18-(a). The levels $q_1(t)$ and $q_2(t)$ in the tank represent the Li-ion concentrations in the electrode

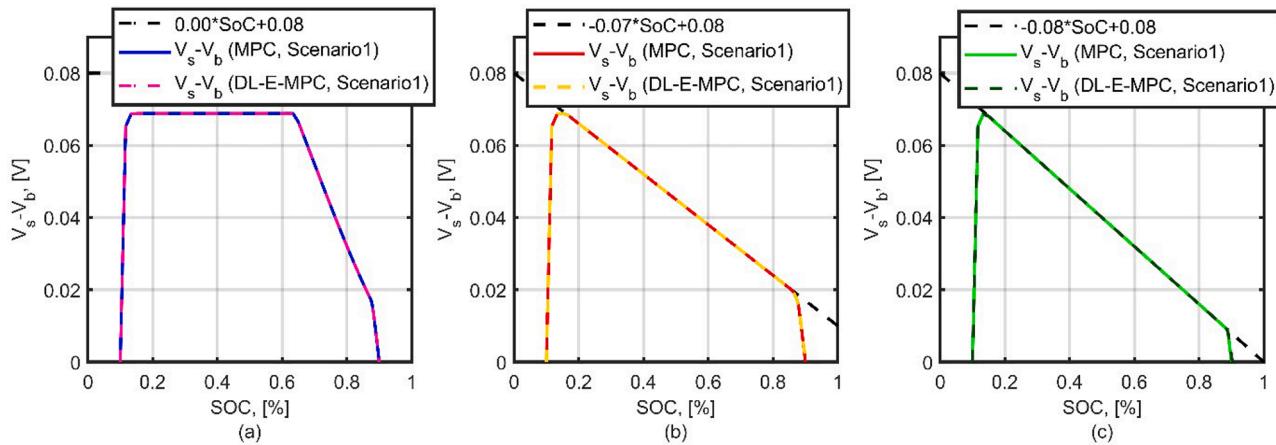


Fig. 17. Performance, with respect to the health constraint, of three closed-loop profiles of charging currents predicted by DNN-based control laws (magenta, yellow and dark-green dashed lines) compared to the exact ones (blue, red and green solid lines) in the three application settings characterized by different values of $\gamma_1 \in \{0.00, -0.07, -0.08\}$ and considering the same initial conditions.

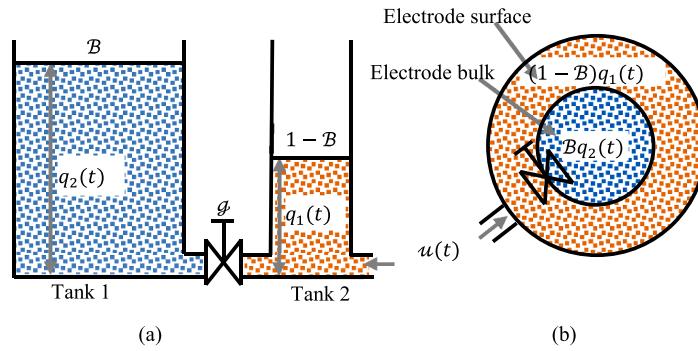


Fig. 18. Representation of the analogy between the EHM (a) and the SPM (b).

surface and core, respectively, while the tanks cross-sectional areas \mathcal{B} and $1 - \mathcal{B}, \mathcal{B} \in [0, 1]$, represent the constant ratio between the bulk and the total electrode volumes. The two tanks exchange Li-ions with a constant rate g that equals to the inverse of the diffusion time constant. The inlet or outlet flow to/from the system $u(t)$ characterizes the charging or discharging current $I(t)$ [$A \cdot m^{-2}$] normalized to the electrode active area by a constant parameter δ , i.e., $u(t) = \delta I(t)$ (Schorsh et al., 2016). Thus, the anode state can be expressed as follows:

$$\frac{d SOC(t)}{dt} = -\delta I(t) \quad (25)$$

$$\frac{d CSC(t)}{dt} = \frac{g}{\mathcal{B}(1-\mathcal{B})} (SOC(t) - CSC(t)) - \frac{\delta}{1-\mathcal{B}} I(t) \quad (26)$$

The cathode state can be described by the variables $SOC^+(t)$ and $CSC^+(t)$. Assuming that the dynamics of the cathode is much faster than that of the anode, it can be assumed to be instantaneous and linearly dependent on the anode, as described by equations (25) and (26), where the chemistry-dependent constant parameters ρ and σ can be obtained from the material balance in the electrodes (Romero et al., 2019).

$$SOC^+(t) = \rho SOC(t) + \sigma \quad (27)$$

$$CSC^+(t) = SOC^+(t) \quad (28)$$

The battery voltage is expressed as:

$$V_{tr}(t) = \Delta U(t) - \eta_s^+(t) - \eta_s(t) - R_f \delta I(t) \quad (29)$$

where:

i) $\Delta U(t)$ is the open-circuit voltage, which equals to the difference between the equilibrium potential of the positive electrode $U^+(t) = \mathcal{F}_1(CSC^+(t))$ and the equilibrium potential of the negative electrode $U(t) = \mathcal{F}_2(CSC(t))$, where \mathcal{F}_1 and \mathcal{F}_2 are experimentally parametrizable nonlinear functions.

ii) η_s^+ and η_s are the surface overpotentials of the positive and negative electrodes, respectively. Equation (30) shows the formula for calculating η_s^+ .

$$\eta_s^+(t) = C \sinh^{-1} \left[\frac{\theta^+}{\sqrt{CSC^+(t)(1-CSC^+(t))}} I(t) \right] \quad (30)$$

where C and θ^+ are constant parameters combining the kinetic contributions to the overpotentials (Goldar et al., 2020). Note that η_s is calculated using the same equation, but considering the associated parameters and states of the negative electrode, i.e., θ and CSC .

i) R_f is the film resistance.

In order to include the effect of the cell temperature resulting from the heat generated during charging/discharging, a thermal module is augmented, where the dynamics of the cell temperature is described as (Romero, Goldar, & Garone, 2019):

$$\frac{dT(t)}{dt} = \frac{1}{m_{cell} C_{p,cell}} [h A (T(t) - T_{amp}) + Q_{gen}(t)] \quad (31)$$

where $T(t)$ [$^{\circ}\text{C}$] is the battery temperature, T_{amp} is the ambient temperature, m_{cell} is the cell mass [kg], $C_{\text{p,cell}}$ is the cell specific heat [$\text{J} \cdot \text{kg}^{-1} \cdot \text{K}$], A is the heat transfer area [m^2], h is the overall heat transfer coefficient [$\text{W} \cdot \text{m}^{-2} \cdot \text{K}$], and $Q_{\text{gen}}(t)$ is the heat generated by the cell which is modelled as follows:

$$Q_{\text{gen}}(t) = \frac{I(t)}{C_f B_c} \left[V_{tr}(t) - (\mathbb{U}^+(t) - \mathbb{U}(t)) - \left(\frac{T_{\text{ref}} DS(t)}{Frd} \right) \right] \quad (32)$$

where C_f is the 1-C current flux [$\text{A} \cdot \text{m}^{-2}$], B_c is the battery capacity

\dot{SOC}_t , CSC_t , \dot{T}_t , DS_t , CSC_t^+ , \mathbb{U}_t^+ , \mathbb{U}_t , $V_{tr,t} = \mathbf{F}$ (.....), where \mathbf{F} is the EHM

$$N_p = 5, N_c = 2, Q = 100, R = 0.0001 \quad (40)$$

[Ah], T_{ref} is the reference temperature [K], Frd is the Faraday constant [c.mol^{-1}], $DS(t) = \mathcal{F}_3(SOC(t))$ is the entropy change [$\text{J} \cdot \text{mol}^{-1} \cdot \text{K}$], and f_3 is an experimentally parametrized function. The full mathematical description of the EHM and the values of its parameters, which are adapted from (Goldar et al., 2020; Romero, Goldar, & Garone, 2019), can be found in the Appendix.

In this case study, a MPC problem is formulated (Eqs. (33):(40)) in order to drive the SOC of the battery to its setpoint $SOC_{\text{ref}} = 0.975 SOC_{\text{max}}$, $SOC_{\text{max}} = 0.62$ (Eq. (34)) (Goldar et al., 2020; Romero, Goldar, & Garone, 2019). The system must achieve its setpoint within a maximum of $N^{fl} = 70$ sampling periods each with 1 minute duration. Operational and safety requirements imply maximum bounds on the applied charging current I_{t+k} (Eq. (35)) and on the cell voltage $V_{tr,t+k}$ (Eq. (36)). The problem is associated with several health constraints, including i) a constraint (Eq. (37)) ensuring that the charge entering to the battery must not exceed its maximum value SOC_{max} , to avoid too many Li-ions entering the electrode, which accelerate the degradation of the battery, ii) a constraint (Eq. (38)) that imposes a maximum limit on the battery temperature during charging to minimize the aging risk, iii) two linear constraints (Eqs. (39)) limiting the overpotentials so as to minimize the electrochemical side-reactions that compromise the battery life (Romero, Goldar, & Garone, 2019).

$$\min_{I_{t+1}, \dots, I_{t+N_p}} J = \sum_{k=0}^{N_p-1} (SOC_{t+k} - SOC_{\text{ref}})' Q (SOC_{t+k} - SOC_{\text{ref}}) + \Delta I_{t+k}' \mathcal{R} \Delta I_{t+k} \quad (33)$$

$$SOC_0 = 0.01, SOC_{\text{ref}} = 0.6045 \text{ (90\%)} \quad (34)$$

$$0 \leq I_{t+k} \leq 170 \text{ A} \cdot \text{m}^{-2}, k = 1, \dots, N_p \quad (35)$$

$$V_{tr,t+k} \leq 4.2 \text{ V}, k = 1, \dots, N_C \quad (36)$$

$$CSC_{t+k} \leq SOC_{\text{max}}, SOC_{\text{max}} = 0.62, k = 1, \dots, N_C \quad (37)$$

$$T_{t+k} \leq 40 \text{ } ^{\circ}\text{C}, k = 1, \dots, N_C \quad (38)$$

$$\begin{aligned} I_{t+k} &\geq \omega_1 CSC_{t+k} + \xi_1 \\ I_{t+k} &\geq \omega_2 CSC_{t+k} + \xi_2 \end{aligned} \Big\}, k = 1, \dots, N_C \quad (39)$$

The proposed methodology is applied following the same steps described in Sections 3. Over the estimated feasible domain of the state variables, $[SOC_{\text{min}} = 0 : SOC_{\text{max}} = 0.6045, CSC_{\text{min}} = 0 : CSC_{\text{max}} = 0.6045, T_{\text{min}} = 13 : T_{\text{max}} = 40]$, a set including different values of the initial state $[SOC_{0,i}, CSC_{0,i}, T_{0,i}]_{i=1:500}$ is generated by the hybrid DOCE technique. In more detail, this sampling plan is composed by 350 samples generated by the Hammersley sequence, and other 150 samples generated by the full-factorial design. Although the EHM of the battery involves many variables, only SOC , CSC and T are considered because they completely determine the full state of the model. Then, the

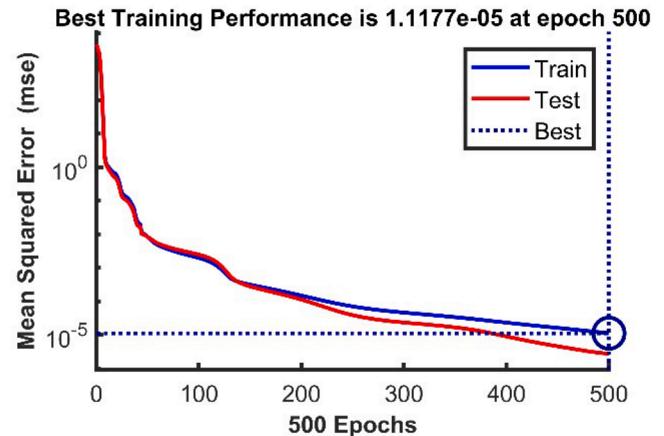
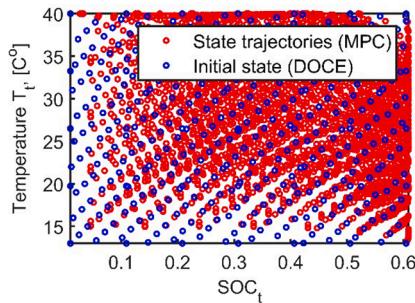
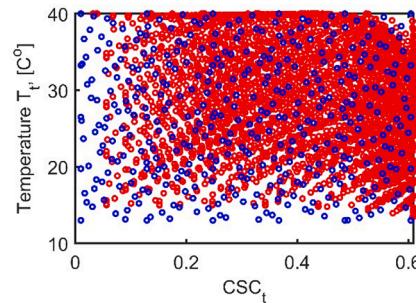


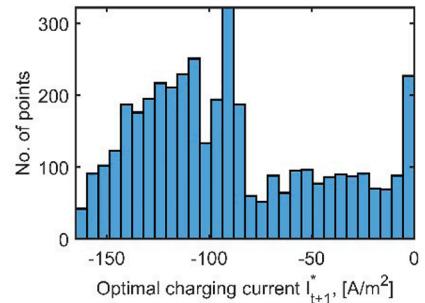
Fig. 20. Loss curves (on training and testing subsets) of the DNN-based charging control law, $P_{\text{DNN}}^{\text{EHM}}, T_{\text{max}}=40$.



(a)



(b)



(c)

Fig. 19. Unfolded training dataset of the EHM case study: (a,b) state variables, and (c) distributions of the optimal charging current.

MPC problem (Eqs. (33):(40)) is solved 500 times with a horizon of $N^{th} = 8$ sampling periods to obtain the optimal closed-loop trajectories of current $[I_{1,i}^*, I_{2,i}^*, \dots, I_{8,i}^*]_{i=1:500}$ and the resulting state trajectories $[SOC_{0,i}, SOC_{1,i}, \dots, SOC_{7,i}]_{i=1:500}$, $[CSC_{0,i}, CSC_{1,i}, \dots, CSC_{7,i}]_{500}$, and $[T_{0,i}, T_{1,i}, \dots, T_{7,i}]_{500}$. Fig. 19 shows the generated input-output training dataset.

The unfolded training dataset $[SOC_{t,i}, CSC_{t,i}, T_{t,i}]_{4000} - [I_{t+1,i}^*]_{4000}$ is used to develop a DNN-based charging control law, $\hat{I}_{t+1}^* = F_{DNN, T_{max}=40}^{EHM}(SOC_t, CSC_t, T_t; \theta_{T_{max}=40})$. The same cut-and-try procedure employed in Section 4.1 is utilized here as well, resulting in the determination of an optimal structure comprising three hidden layers involving 14, 7, and 3 neurons, all utilizing a sigmoid transfer function. Additionally, a BR-LMB is utilized to train the DNN in this case. Fig. 20 shows the loss functions over the training and testing subsets.

To evaluate the performance of the developed control law $F_{DNN, T_{max}=40}^{EHM}$ a test set is generated, which contains a number of $n^{ts} = 30$ optimal closed-loop control profiles $[I_{1,i}^{*ts}, I_{2,i}^{*ts}, \dots, I_{70,i}^{*ts}]_{i=1:30}$ and the associated state trajectories $[SOC_{0,i}^{ts}, SOC_{1,i}^{ts}, \dots, SOC_{69,i}^{ts}]_{i=1:30}$, $[CSC_{0,i}^{ts}, CSC_{1,i}^{ts}, \dots, CSC_{69,i}^{ts}]_{i=1:30}$ and $[T_{0,i}^{ts}, T_{1,i}^{ts}, \dots, T_{69,i}^{ts}]_{i=1:30}$. The evaluation metrics, see Tables 7 and 8, show promising performance of the control law in terms of very high prediction accuracy (NRMSEs of the open-loop, closed-loop, and resulting states are well below 1.0%), very low online computational cost for predicting the control actions (15.9 s) compared to those of mathematical MPC (406.0 s), and the low violations of the constraints (Eqs. (35), (36), (38) and (39)). Once again, Table 8 does not consider the constraint $CSC_{t+k} \leq SOC_{max}$ (Eq. (37)) because it is always idle (neither in the training nor in the testing scenarios), since it is dominated by the required setpoint.

Fig. 21-(a, b, c) compares the 30 predicted closed-loop control trajectories of charging current and the resulting SOC, temperature, and terminal voltage (colored crosses and colored dashed lines) with their exact values (grey hollow circles and grey solid lines). Note that each of the predicted trajectories is identified by a different color, starts from a different initial condition (represented by a colored filled circle), and terminates at the setpoint or steady state of the system (represented by a yellow star). In addition, the setpoint, the constraints on the charging current (Eq. (35)), temperature (Eq. (38)) and the terminal voltage (Eq. (36)) are represented by horizontal or vertical black lines. Again, the figure illustrates and emphasizes the high accuracy of the developed control law, $F_{DNN, T_{max}=40}^{EHM}$, and its potential ability to learn the constraints from the data only, without any prior knowledge of the mathematical formulations.

Fig. 22 illustrates the very good performance of the 30 closed-loop charging current trajectories (dashed colored lines) predicted by the DNN-based control law $F_{DNN, T_{max}=40}^{EHM}$ in satisfying the two health constraints (Eq. (39)) represented by the slanted black lines delineating the feasible space in the upper left region of the figure. Each predicted current profile is denoted by a different color, starts from a different IC

which is represented by a colored filled square, and ends at steady state, which is represented by a yellow star. Note that the health constraints are active in all 30 scenarios. Therefore, the predicted current profiles are difficult to see because they almost coincide with the slanted black lines (i.e., the health constraints), see the magnified details in the same figure. If we look at the magnified view on the left, we can find that the solid black line representing a health constraint is actually 30 solid black lines that are almost identical with extremely small errors. These errors are due to the fact that for each charging scenario, a health constraint (e.g., $\omega_1 CSC_{t+k} + \xi_1$) is constructed by the CSC resulting from the application of the predicted trajectory of current which itself contains prediction errors of the DNN-based control law.

It is promising that the DNN-based control law, $F_{DNN, T_{max}=40}^{EHM}$, is able to propose charging profiles that satisfy the health constraints by different behaviors, depending on the IC (mainly the initial SOC and temperature) of each scenario. These behaviors can be classified into 3 groups:

- The first group includes scenarios with relatively low initial SOC and temperature that are, mostly, the ones whose ICs (filled colored squares in Fig. 22) lies on the solid black line representing the health constraint $\omega_1 CSC_{t+k} + \xi_1$. In these scenarios, the first health constraint $\omega_1 CSC_{t+k} + \xi_1$ is active during the earliest charging period, and therefore, it leads the charging current trajectory until the SOC becomes relatively high, then the second health constraint $\omega_2 CSC_{t+k} + \xi_2$ becomes active and proceed to direct the charging current, and finally, just before the steady state (yellow star), the two constraints become inactive until the end of the charging.
- The second group contains scenarios with relatively high initial SOC and low temperature, which are the ones whose ICs lies on the solid black line representing the health constraint $\omega_2 CSC_{t+k} + \xi_2$. Here, only the second health constraint $\omega_2 CSC_{t+k} + \xi_2$ is active from the beginning of the charging until approaching the steady state (yellow star).
- The third group involves only two scenarios characterized by relatively low initial SOC and high temperature. The two scenarios can be distinguished in Fig. 22 as the ones with ICs represented by red and mauve filled squares located on the first health constraint line $\omega_1 CSC_{t+k} + \xi_1$. At the start of charging, the first health constraint is active for a while, then the temperature rapidly approaches its upper bounds ($40^\circ C$) therefore the DNN controller reduces the charging current to satisfy the temperature constraint (see Fig. 22, where the red and mauve dashed lines deviate from the first health constraint line). After that, the temperature decreases, and the first health constraint return to be active again and the rest of the behavior continues as in the first group.

Fig. 31, in the Appendix, visualizes the performance of the predicted profiles of charging current with respect to the health constraints in Eq.

Table 7

Computational cost and accuracy of the developed DNN-based control law $F_{DNN, T_{max}=40}^{EHM}$.

Offline CPU time (s)*										Accuracy (NRMSE (%))								Online CPU time (s)*	Saved time %	
Data generation via MPC		DNN fitting		Open-loop		Closed-loop														
Train.	Test.			Control	Control	Resulting states														
		I^*		I^*		SOC	CSC	T	DS	CSC^+	U^+	U^-	V							
$F_{DNN, T_{max}=40}^{EHM}$	898	406	208	0.095	0.045	0.012	0.012	0.003	0.011	0.012	0.034	0.002	0.026	15.9					96.0	

*Intel core (TM) i7-8565U CPU@ 18 GHz, 12 GB RAM.

Table 8

Average constraint violations of the developed DL-based control law $F_{DNN, T_{max}=40}^{EHM}$.

Constraint 1	Constraint 2	Constraint 3	Constraint 4	Constraint 5
$I_t = 0$	$-170 - I_t$	$T_t - 40$	$V_{tr, t} - 4.2$	$I_t \geq \omega_1 CSC_t + \xi_1$
2.5×10^{-2}	0	0	0	$I_t \geq \omega_2 CSC_t + \xi_2$

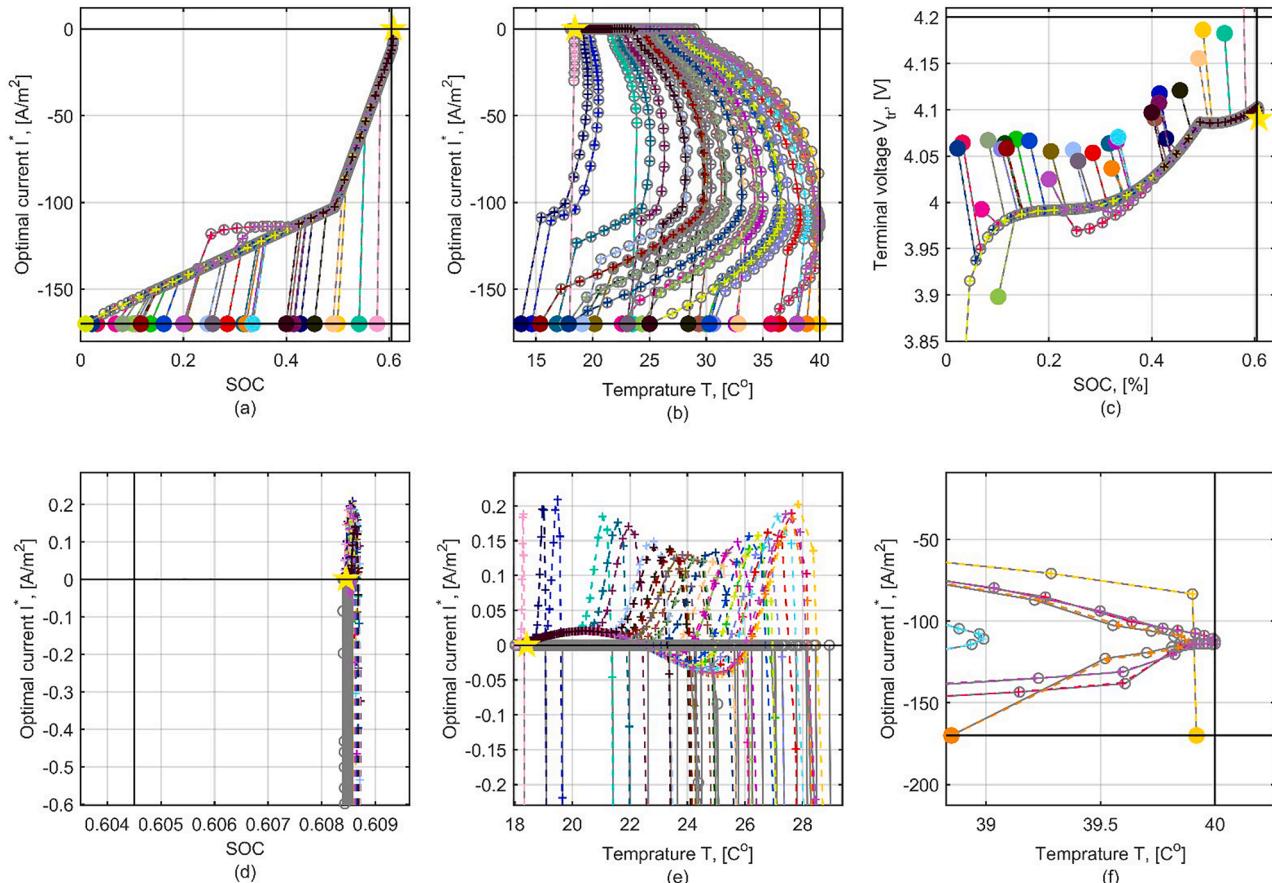


Fig. 21. Predicted vs. exact closed-loop charging current trajectories and the resulting SOC (a), temperature (b), and terminal voltage (c), along with enlarged details (d, e, f). colored filled circle are the different initial conditions, colored crosses and colored dashed lines are the predicted behaviors, grey hollow circles and grey solid lines are the exact behaviors, yellow stars are the setpoint or the steady state, and horizontal and vertical solid black lines are the setpoint and constrains on the current, temperature and the terminal voltage.

(39), but in the time dimension: each subplot is devoted to three (out of thirty) scenarios, while each scenario is distinguished by a different color where the hollow squares are the predicted charging current trajectories, dashed lines are the first health constraint $\omega_1 CSC_{t+k} + \xi_1$, and dash-dotted lines are the second constraint $\omega_2 CSC_{t+k} + \xi_2$. Here, it is easier to recognize the previously mentioned three groups of behavior of the DNN-based controller, just for example: any scenario in Appendix (Fig. 31-(a, g, i)) belongs to the first group, the yellow scenario in Appendix (Fig. 31-(b)) and the turquoise scenario in Appendix (Fig. 31-(c)) are from the second group, and finally, the red scenario in Appendix (Fig. 31-(f)) and the mauve scenario in Appendix (Fig. 31-(j)) are the third group.

Fig. 23-(i) shows the time evolution of three (out of thirty) closed-loop charging current profiles predicted by the control law $F_{DNN, T_{max}=40}^{EHM}$ (magenta, yellow and dark-green dashed lines) starting from different initial conditions compared to their exact values (red, blue and green solid lines). These initial conditions are $[SOC_{01} = 0.01, CSC_{01} = 0.01, T_{01} = 25 C^\circ]$, $[SOC_{02} = 0.20, CSC_{02} = 0.20, T_{02} = 38 C^\circ]$, and $[SOC_{03} = 0.40, CSC_{03} = 0.40, T_{03} = 25 C^\circ]$. Fig. 23-(a:h)

compares the state of the battery resulted from the application the predicted closed-loop charging current (dashed lines) with the optimal states resulted from the application of the exact closed-loop charging profiles (solid lines). Also, Fig. 24-(a-c) shows the performance of these three predicted charging current profiles with respect to the health constraints (Eq. (39)). The two figures confirm, again, that the DNN-based control law $F_{DNN, T_{max}=40}^{EHM}$ is able to accurately predict closed-loop charging current profiles and to satisfy the constraints, which include upper and/or lower bound(s) on the temperature (Eq. (38)), terminal voltage (Eq. (36)) and on the applied current (Eq. (35)) represented by black dashed lines in Fig. 23-(c,h,i), respectively, additional to the health constraints represented by dashed and dash-dotted black lines in Fig. 24.

Also, in this case study, we will test the robustness of the methodology with respect to the stringency of the constraints. In this case, we choose the temperature constraint (Eq. (38)) because, it has been active in very few testing scenarios, see Fig. 21-(b, f). Therefore, the method is applied three times with exactly the same steps and details, but considering different upper bound values of the temperature constraint,

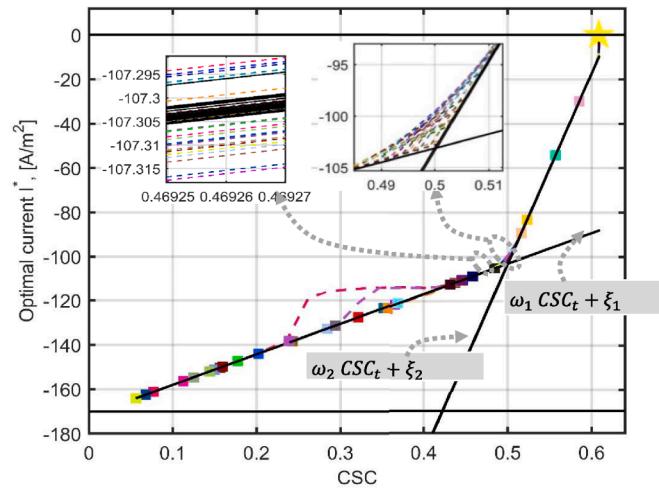


Fig. 22. The 30 closed-loop charging current trajectories predicted by the DNN-based control law $F_{DNN, T_{max}=40}^{EHM}$ (colored dashed lines) versus the resulting CSC, in the case of $T_{max} = 40 \text{ }^{\circ}\text{C}$. Colored filled squares are the initial conditions, yellow star is the steady state, horizontal solid black lines are the bound constraints on the current, and inclined solid black lines represent the two health constraints.

i.e., $T_{t+k} \leq 32 \text{ }^{\circ}\text{C}$, $T_{t+k} \leq 30 \text{ }^{\circ}\text{C}$ and $T_{t+k} \leq 28 \text{ }^{\circ}\text{C}$. Table 9 reports the results of the three applications in terms of i) the offline computational cost required for generating the training and testing datasets and for fitting the control laws $F_{DNN, T_{max}=32}^{EHM}$, $F_{DNN, T_{max}=30}^{EHM}$, and $F_{DNN, T_{max}=28}^{EHM}$ ii) the accuracy of the predicted charging current and the resulting states, and iii) the required online computational cost. Also, Table 10 shows the average constraint violations achieved by the predicted charging current trajectories in each of the three application campaigns. The two tables emphasize that the methodology is able to maintain approximately the same promising performance while handling tighter constraints on the battery temperature rise.

Fig. 25 shows the good performance of the proposed method in one of the three application settings (i.e., $T_{max} = 30 \text{ }^{\circ}\text{C}$) by comparing all the thirty predicted closed-loop charging profiles and the corresponding states (temperature, SOC, and terminal voltage) with their exact values. Note that the temperature constraint is now active in a larger number of the testing scenarios (Fig. 25-(b, f)) compared to the original application (Fig. 21-(b, f), where $T_{max} = 40 \text{ }^{\circ}\text{C}$). Fig. 26 shows the ability of the predicted current trajectories to satisfy the two health constraints in the application setting where $T_{max} = 30 \text{ }^{\circ}\text{C}$. Note that, the predicted trajectories show the same three groups of behavior highlighted in the original application (see Fig. 22, where $T_{max} = 40 \text{ }^{\circ}\text{C}$), but with more scenarios belonging to the third group as a result of considering tighter upper bound on the temperature rise. Similarly, Fig. 32 in the Appendix, visualizes the performance, in the time dimension, of the predicted closed-loop current trajectories with respect to the two health constraints.

Fig. 27-(i) shows the time evolution of three predicted closed-loop trajectories of charging current (magenta, yellow and dark-green dashed lines), each is predicted by one of the three DNN-based charging control laws developed in the three application settings (i.e., $F_{DNN, T_{max}=32}^{EHM}$, $F_{DNN, T_{max}=30}^{EHM}$ and $F_{DNN, T_{max}=28}^{EHM}$) considering the same initial conditions [$SOC_{01} = 0.01$, $CSC_{01} = 0.01$, $T_{01} = 25 \text{ }^{\circ}\text{C}$]. Note that, even starting from the same initial conditions, the control laws are able to suggest different closed-loop charging scenarios to meet the different upper bounds of the temperature constraints (Fig. 27-(c)). Also, Fig. 28 shows the performance of these three predicted closed-loop current profiles with respect to the two health constraints. This confirms the robustness of the proposed DL-E-MPC, in terms of its ability to adapt the ML-based control laws to the different settings of the problem constraints.

Finally, it is worth discussing the computational complexity of the

proposed method in light of the four algorithms (or pseudo-codes) and the four steps detailed in Section 3. First, the generation of the training and test datasets is the most computationally intensive step, see Sections 3.1 and 3.2, along with Algorithm 1. This can be seen in Tables 3 and 5 for the first case study, and in Tables 7 and 9 for the second case study, specifically on the field of “Data generation via MPC”. The reason for this is that the required data consists of several state-control scenarios, which are generated by repeatedly solving an expensive MPC problem. This computational complexity, denoted as CC_1 , can be approximated by Eq. (41). Note that CC_1 increases linearly with the chosen prediction horizon N_p , the number of initial conditions n^r , the time horizon of the MPC problem N^{trn} , and the number of iterations required by the MPC solver (i.e., optimization algorithm) to converge K . Moreover, the complexity CC_1 scales cubically with the complexity of executing one iteration of the optimization algorithm, which is expressed by the term $O((m + v + n_{cons})^3)$. Here $m + v$ represents the total number of state and control variables, and n_{cons} denotes the number of constraints imposed on the MPC problem (i.e., Eqs. (4):(6)).

$$CC_1 = O(N_p \times n^r \times N^{trn} \times K \times ((m + v + n_{cons})^3 + m)) \quad (41)$$

Second, the computational complexity of training the DNN-based control laws, as described in Section 3.3 and Algorithm 2, is influenced by several factors, including the number of DNN-based control laws to be developed v , the DNN architecture, the number of training epochs, and the size of the training dataset $n^r \times N^{trn}$. For a generic DNN architecture consisting of L hidden layers, where each layer L_l includes N^{L_l} neurons (with $l = 1, \dots, L$), this computational complexity is expressed by the following equation:

$$CC_2 = O(v \times n_{ep} \times n^r \times N^{trn} \times L \times (N^{L_l})^2) \quad (42)$$

Fortunately, both the data generation and the fitting of the DNN-based control laws are offline tasks. Once the DNN-based control law is fitted and ready, it can be used for prediction with minimal computational effort and cost.

Third, the computational complexity of the open-loop evaluation, described in Section 3.3 and in Algorithm 3, involves successive forward passes through the DNN models, which can be estimated as follows:

$$CC_3 = O(v \times n^{ls} \times N^{fl} \times L \times (N^{L_l})^2) \quad (43)$$

Fourth, for the online or closed-loop deployment (refer to step 4 and Algorithm 4), successive forward passes through the DNN models are

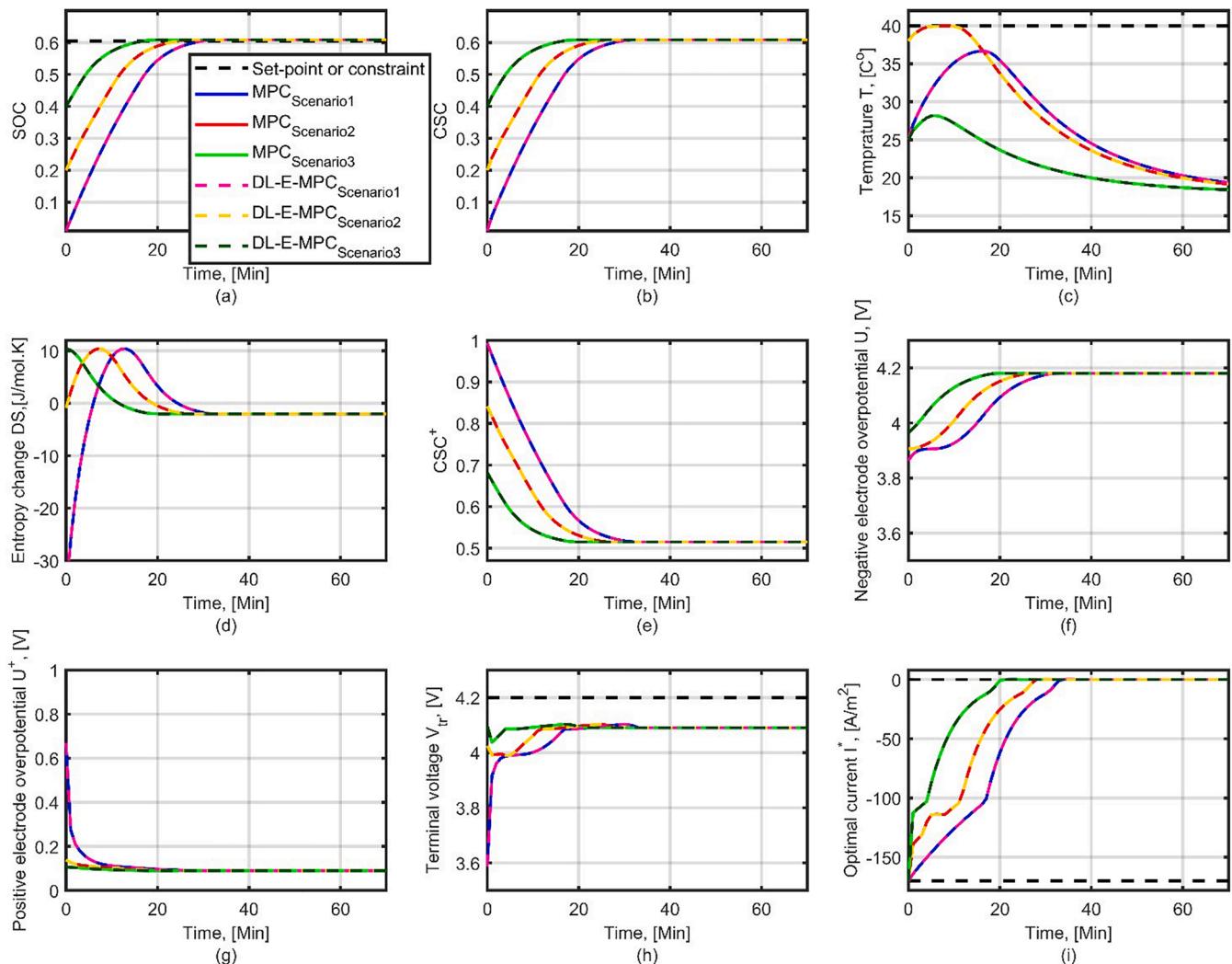


Fig. 23. Three closed-loop charging current profiles predicted by the DNN-based control law $P_{DNN, T_{max}=40}^{EHM}$ and the resulting state (magenta, yellow and dark-green dashed lines) compared to their exact values obtained by solving the MPC problem (blue, red and green solid lines), considering three different initial conditions.

also performed similarly to those in CC_3 . However, an additional step is necessary to update the state variables using the dynamic model F , which mimics the real plant in our numerical study. The computational complexity of updating the state variables using the dynamic model F is approximated as $O(n^{ts} \times N^{fl} \times m)$, where it depends on the size of the

test dataset $n^{ts} \times N^{fl}$ and the dimensionality of state variables m described by the dynamic model F . Therefore, the computational complexity of Algorithm 4, CC_4 , can be expressed as the following equation:

$$CC_4 = CC_3 + O(n^{ts} \times N^{fl} \times m) \quad (44)$$

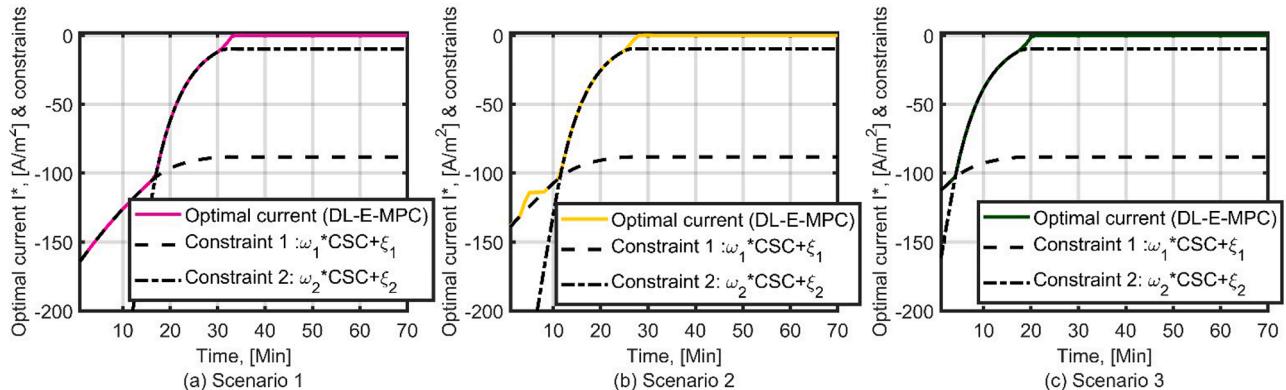


Fig. 24. Performances of three closed-loop charging current profiles predicted the DNN-based control law $P_{DNN, T_{max}=40}^{EHM}$ with respect to the two health constraints (Eq. (39)), considering three different initial conditions.

Table 9

Computational cost and accuracy of the developed DNN-based control laws in three application settings characterized by different values of T_{\max} : 32, 30 and 28 $^{\circ}\text{C}$.

Data generation via MPC	Offline CPU time (s)*		Accuracy (NRMSE(%))									Online CPU time (s)*	Saved time %		
	DNNs fitting	Train.	Test.	Control	Resulting states										
					I^*	I^*	SOC	CSC	T	DS	CSC ⁺	U^+	U^-	V_{tr}	
$F_{DNN,T_{\max}=32}^{EHM}$	2494	481	159	0.41	0.01	0.035	0.035	0.02	0.032	0.035	0.099	0.004	0.76	22.8	95.2
$F_{DNN,T_{\max}=30}^{EHM}$	1790	483	64.5	0.47	0.14	0.036	0.037	0.02	0.039	0.036	0.091	0.03	0.10	22.4	95.4
$F_{DNN,T_{\max}=28}^{EHM}$	1171	495	142	0.74	0.26	0.041	0.043	0.04	0.04	0.042	0.10	0.01	0.13	17.3	96.5

*Intel core (TM) i7-8565U CPU@ 18 GHz, 12 GB RAM.

Table 10

Average constraint violations in the three application settings characterized by different values of T_{\max} : 32, 30 and 28 $^{\circ}\text{C}$.

	Constraint 1		Constraint 2		Constraint 3		Constraint 4		Constraint 5	
	$I_t = 0$	$-170 - I_t$	$T_t = 40$	$V_{tr,t} = 4.2$	$I_t \geq \omega_1 CSC_t + \xi_1$	$I_t \geq \omega_2 CSC_t + \xi_2$				
$F_{DNN,T_{\max}=32}^{EHM}$	3.2×10^{-3}	0	0	0	3.9×10^{-2}	5.3×10^{-2}				
$F_{DNN,T_{\max}=30}^{EHM}$	4.9×10^{-2}	0	0	0	3.8×10^{-2}	7.2×10^{-2}				
$F_{DNN,T_{\max}=28}^{EHM}$	5.5×10^{-2}	0	0	0	3.4×10^{-2}	6.2×10^{-2}				

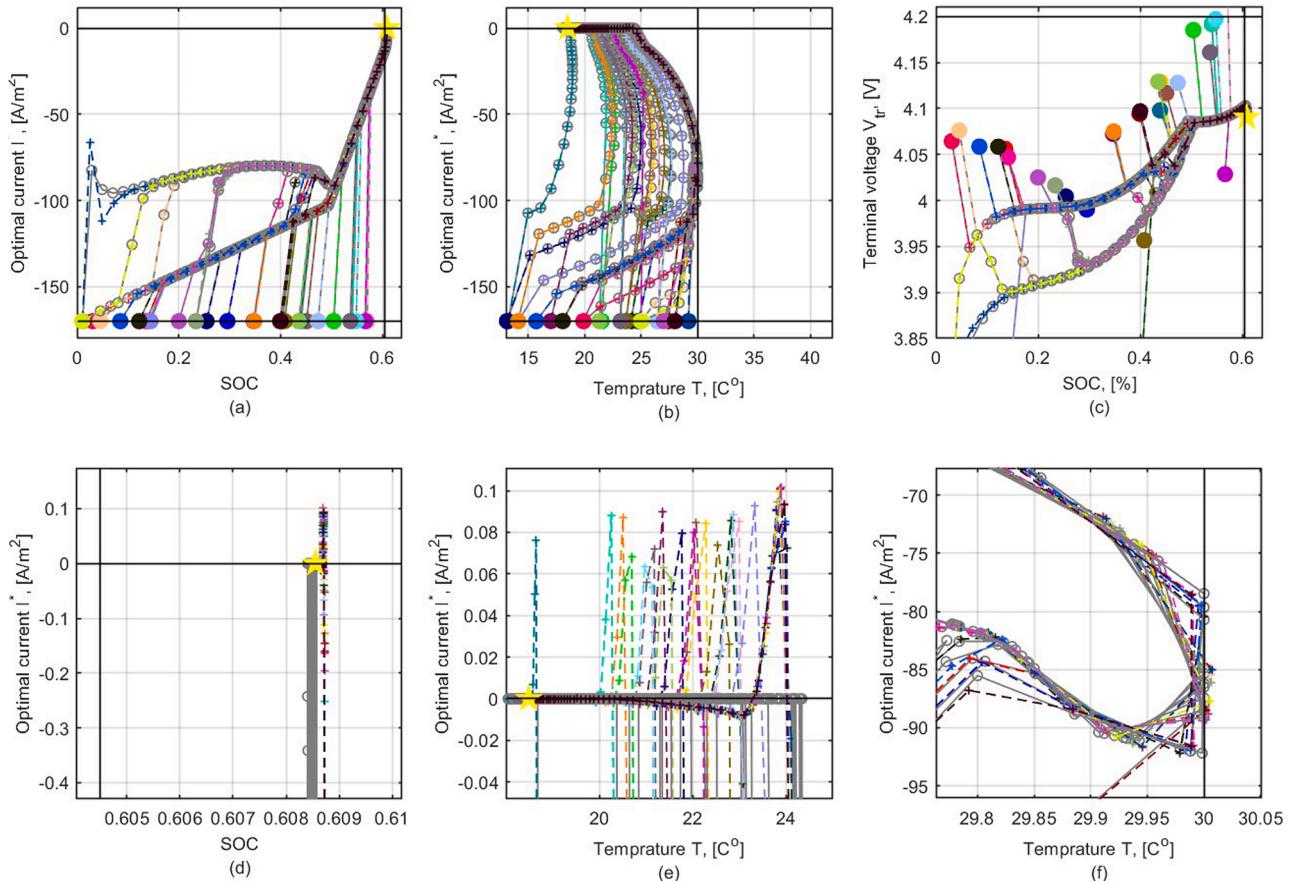


Fig. 25. Predicted vs. exact closed-loop charging current trajectories and resulting SOC (a), temperature (b), and terminal voltage (c), along with magnified details (d, e, f) in the application case where $T_{\max} = 30$ $^{\circ}\text{C}$. Colored filled circle are the different initial conditions, colored crosses and colored dashed lines are the predicted behaviors, grey hollow circles and grey solid lines are the exact behaviors, yellow stars are the setpoint or the steady state, and the horizontal and vertical solid black lines are the setpoint and constrains on the current, temperature and on the terminal voltage.

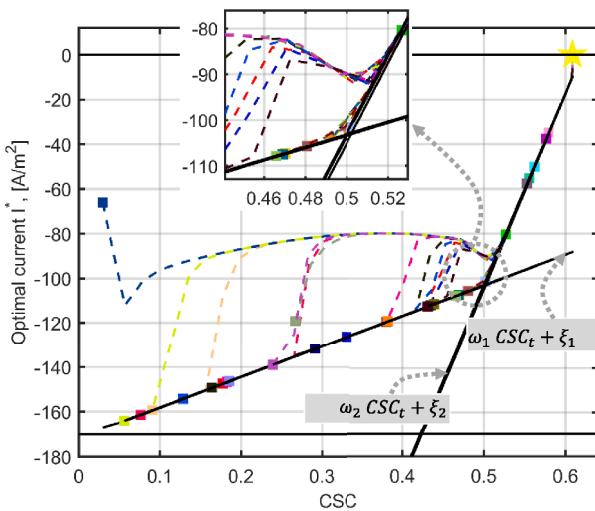


Fig. 26. The 30 closed-loop trajectories of the charging current predicted by the DNN-based control law $F_{DNN}^{EHM}, T_{max}=30$ (colored dashed lines) versus the resulting CSC, in the case of $T_{max} = 30$ C°. Colored filled squares are the initial conditions, yellow star is the steady state, horizontal solid black lines are the bound constraints on the current, while the inclined solid black lines represent the two health constraints.

Finally, the overall complexity CC_4 can be expressed as:

$$CC_4 = O\left(n^{ts} \times N^{fl} \times \left(v \times L \times (N^{L_i})^2 + m\right)\right) \quad (45)$$

It is important to note that our study is purely numerical, where the dynamic model F serving as the real plant during the closed-loop implementation. However, in real-world application, the control actions predicted by the DNN models are directly implemented in the actual plant. In this scenario, evaluating the process model F is unnecessary, which leads to reduced online computational demands.

Apart from the size of the testing set, the complexity of the online deployment is primarily influenced by the design and parameterization of the DNN models. To achieve low computational complexity while maintaining high accuracy, it is crucial to optimize the DNN architectures and training parameters (such as L and N^{L_i}) to strike a balance between accuracy and computational demands. For both case studies, the significant reduction in online computational effort can be observed by referring to the "Online CPU Time" field in the previously indicated tables.

4.3. Controller performance against noise

This section investigates the impact of process noise on the performance of the DNN-based controller, with a particular focus on Gaussian noise, which frequently affects process outputs and state measurements due to sensor inaccuracies and external disturbances. To assess this effect, we employ the first case study (Section 4.1) as a representative

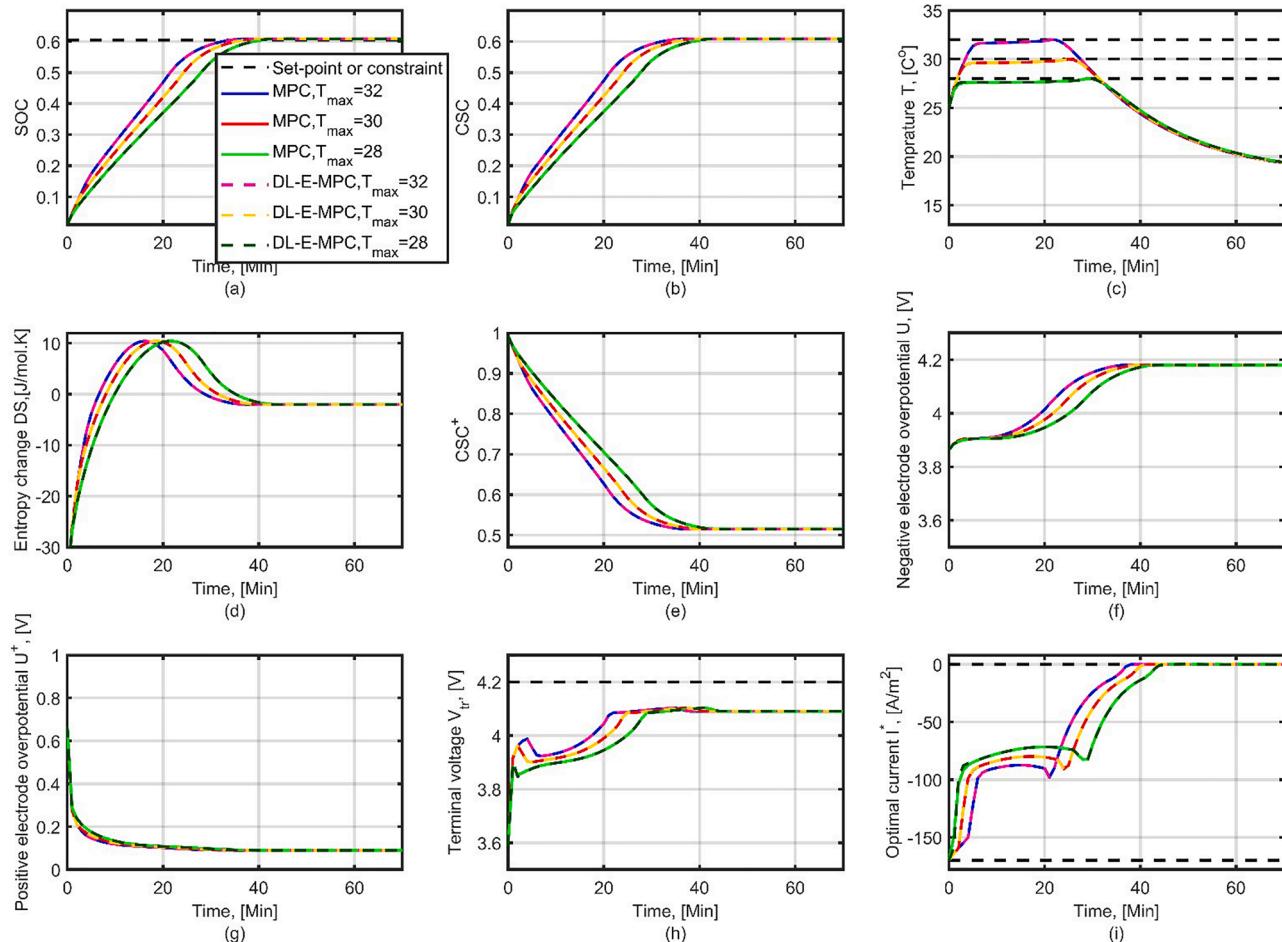


Fig. 27. Three closed-loop charging currents profiles predicted by the three DNN-based control laws (magenta, yellow, and dark-green dashed lines) compared to their exact values obtained by solving the MPC problems (blue, red, and green solid lines) considering the same initial condition, but different values for the temperature constraint bound $T_{max} \in \{32, 30, 28$ C° $\}$.

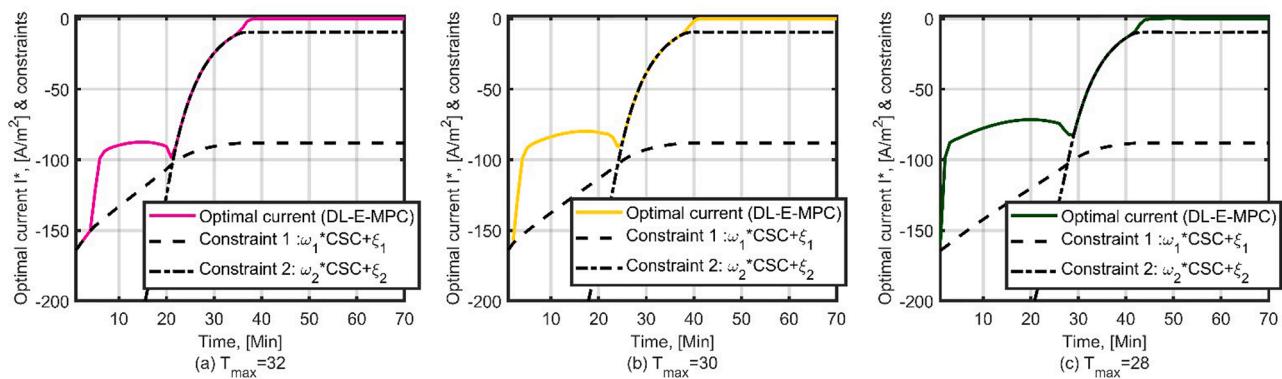


Fig. 28. . Performance of three closed-loop charging currents profiles predicted the three DNN-based control laws with respect to the two health constraints (Eq. (39)) considering the same initial condition, but different values for the temperature constraint bound $T_{\max} \in \{32, 30, 28 \text{ C}^\circ\}$.

Table 11

Accuracy of the DNN-based control law $F_{DNN, \gamma_1=-0.04}^{NDC}$ tested under four noise levels, represented by the standard deviations $\sigma_{noise} \in \{0.003, 0.005, 0.007 \text{ and } 0.010\}$.

Noise level	Accuracy (NRMSE (%))				
	Closed-loop				
	Control	Resulting states			
I^*	V_b	V_s	V_{tr}	SOC	
$\sigma_{noise} = 0.003$	1.04	0.94	0.92	1.4	0.94
$\sigma_{noise} = 0.005$	1.56	1.19	1.17	1.85	1.2
$\sigma_{noise} = 0.007$	2.08	1.52	1.5	2.44	1.52
$\sigma_{noise} = 0.010$	2.93	2.1	2.08	3.54	2.1

Table 12

Average constraint violations of the DNN-based control law $F_{DNN, \gamma_1=-0.04}^{NDC}$ tested under four noise levels, represented by the standard deviations $\sigma_{noise} \in \{0.003, 0.005, 0.007 \text{ and } 0.010\}$.

Noise level	Average constraint violations			
	$V_{tr,t} - 4.2$	$V_{s,t} - V_{b,t} - \gamma_1 SOC_t - 0.08$	$I_t - 3$	$0 \leq I_t$
$\sigma_{noise} = 0.003$	3.7×10^{-3}	1.44×10^{-4}	0	0
$\sigma_{noise} = 0.005$	6.5×10^{-3}	2.29×10^{-4}	0	0
$\sigma_{noise} = 0.007$	9.5×10^{-3}	2.84×10^{-4}	0	0
$\sigma_{noise} = 0.010$	1.3×10^{-2}	4.48×10^{-4}	0	0

example, specifically considering the health-related constraint $V_{s,t+k} - V_{b,t+k} \leq \gamma_1 SOC_{t+k} + \gamma_2$ where $\gamma_1 = -0.04$.

The same previously trained DNN-based controller, $F_{DNN, \gamma_1=-0.04}^{NDC}$, is employed. The key difference in this analysis lies in the testing procedure: artificial Gaussian noise is incorporated into the process states trajectories (V_s and V_b) of the original testing scenarios. In more details, four different levels of noise represented by four distinct values for the noise standard deviation $\sigma_{noise} \in \{0.003, 0.005, 0.007, 0.010\}$ are investigated. These noise levels are selected in accordance with those considered in similar studies in the literature (Poza et al., 2023 b).

The performance metrics, including the accuracy of the DNN-based controller and the average constraint violation across the four noise levels, are presented in Tables 11 and 12, respectively. The results indicate that the controller demonstrates robust performance: while both prediction accuracy and constraint satisfaction decrease as the noise level increases, the DNN-based controller continues to perform well even at the highest noise level. Notably, the results in Tables 11 and 12, which depict the controller's performance under different noise levels, remain comparable to the noise-free performance previously reported in Section 4.1 (Tables 3 and 4).

Fig. 29 illustrates the performance of the DNN-based controller in a single testing scenario under various noise levels, compared to the noise-free performance of the classical MPC (solid blue line). This again indicates the robustness of DNN-based controller against noisy measurements.

5. Conclusions

Most of the available physics-based models of batteries can easily hinder the exploitation of the promising capabilities of MPC technologies for real-time optimal charging. This is because these complex, highly nonlinear, and -in some cases- black box models, when integrated into an MPC scheme, can result in a large computational overhead required to repeatedly solve an open-loop OCP online at each sampling instance. To overcome this challenge, this paper proposes a method for explicit MPC based on ML models for real-time optimal health-aware charging.

The proposed method combines various techniques such as DOCE, state-of-the-art dynamic optimization methods and ML models and consists of four steps: (i) sampling over the expected feasible space of state variables using DOCE techniques to create a sampling plan containing different expectations of initial state variable values, (ii) multiple solution of the MPC problem corresponding to each of the previously created combinations of the initial state variable values to obtain different closed-loop control-state trajectories, (iii) off-line development of an accurate control law based on DNNs using the generated data (in the previous two steps) that approximate the optimal charging current to be applied to the battery in the next sampling period as a function of the battery state variables value at the present sampling period, (iv) integration of the developed DNNs-based control law into a control-loop scheme to predict on-line control profiles required for optimal charging of the battery, considering different safety and health constraints.

The method has been applied to two case studies, adopted from the literature, dealing with real-time health-constrained charging of batteries using MPC technology. While the first case study relies on an electrical model (a NDM), the second considers an electrochemical model (an EHM) combined with a thermal module. Results illustrate the promising capabilities of the proposed method, which include:

- i). The method provides efficient DNNs based control laws capable of predicting the closed-loop charging current profiles with a very high prediction accuracy evaluated with respect to the optimal control profiles computed by the classical/mathematical solution of the MPC problems (less than 1.0 % NRMSE in both case studies and in all different application settings). Moreover, the predicted current profiles accurately satisfy the constraints imposed on the charging process, including operational, safety, and health constraints (a maximum constraint violation of an order of

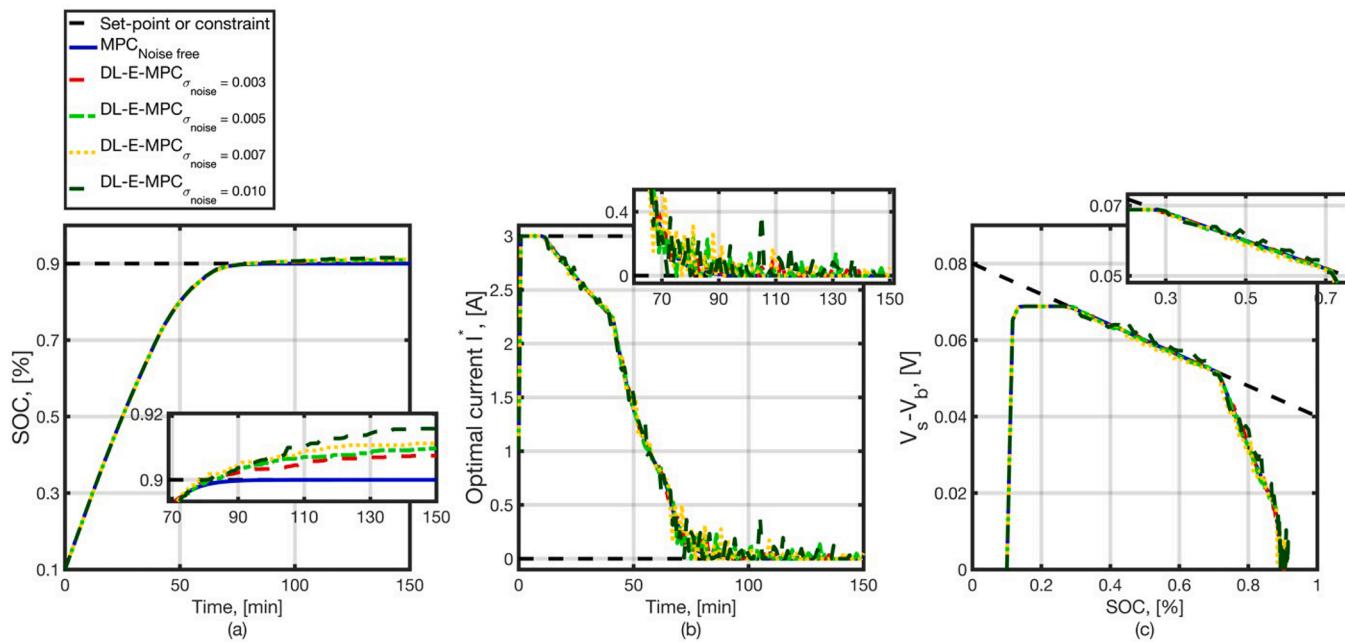


Fig. 29. Closed-loop profiles of charging current predicted by the DNN-based control laws (b) and the resulting SOC (a) along with the behavior concerning the linear health constraint (c), under the four noise levels. This is compared to the exact solution provided by the noise free MPC.

magnitude equal to 10^{-2}). This highlights the potential capabilities of these ML-based control laws for learning the problem constraints directly from the data, without knowledge of the mathematical formulations of the problem. Here, the novel data generation procedure described in Steps 1 and 2 of the methodology plays a crucial role by providing datasets (state-control trajectories) that include scenarios where the various constraints imposed on the MPC problem are active. This enables the DNNs to learn the satisfaction of the constraints while predicting the optimal control actions. Therefore, it is important to emphasize that the ability of the DNN to learn how to satisfy the constraints is contingent upon the availability of sufficient training data/trajecories in which these constraints are active.

- ii). The method shows high applicability in terms of handling health-constrained charging processes via MPC relying on battery models of very distinct characteristics, in terms of a) the fundamental or the phenomenal basis of the model (electrical, electrochemical and thermal), b) scale of the model, and c) the nonlinearity of the model: the NDC model involves moderate nonlinear behavior, such as variables multiplication (in Eq. (14)), power (in Eq. (16)) and exponential (in Eq. (17)) terms, while the EHM includes highly nonlinear behavior, such as power (in Eqs. (50), (52) and (53)), variables multiplication (in Eqs. (49) and (54)) division (in Eq. (54)), exponential (in Eqs. (52) and (53)) and trigonometric (in Eq. (54)) terms.
- iii). The developed DNN-based control laws demonstrate robustness in terms of their ability to accurately predict the closed-loop charging current profiles in different situations characterized by different initial conditions of the battery and tightness of the charging constraints.
- iv). The proposed approach achieves a substantial reduction in computational time compared to the conventional mathematical solution of the MPC problem, yielding a 94.7% decrease even in the lowest-performing testing scenario. This efficiency gain is particularly paramount for the practical deployment and commercialization of real-time optimal charging technologies based on MPC. Given the limited computational resources of most BMSs, solving an MPC problem in real-time remains a significant challenge, even when employing simplified battery models.

While the proposed method shows promise, there are some limitations that should be addressed for future enhancement. One limitation is its lack of applicability to real-world case studies that involve actual operating conditions, noisy measurements, and disturbances such as weather variations. These factors are essential for fully evaluating the robustness of the developed DNN-based control laws. Another minor limitation is its current application scope, which does not extend to more complex case studies involving physics-based models with higher dimensionality and/or nonlinearity, such as the SPM or TFNHE models. However, the methodology detailed in Section 3 can be directly applied to such systems. The primary challenge lies in the increased computational burden during the data generation step, which involves the repetitive solution of the MPC problem based on the physics-based model. Fortunately, this data generation process is performed offline, prior to the online application. Importantly, the online computational performance of the resulting DNN-based controller remains unaffected.

In future research lines, we will investigate the enhancement of the methodology capabilities in two main specific aspect:

- i). to explicitly provide the ML models with information about the mathematical constraints of the MPC problem. With respect to this goal, so-called "constrained DNNs" or "Physics-Informed DNN" represent potential solutions where the network training loss function penalizes both the prediction errors and the constraint violations. This will be convenient in situations where it's difficult to generate enough training data/scenarios with active constraints.
- ii). to consider real-world applications and the unknown disturbances affecting the process, in addition to noise and data inaccuracies, with the objective of developing robust ML-based charging control laws.

CRediT authorship contribution statement

Ahmed Shokry: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mehdi Abou El Qassime:** Writing – review & editing, Visualization, Investigation. **Antonio Espuña:** Writing – review & editing, Supervision. **Eric**

Moulines: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors acknowledge the support received from the AI and predictive maintenance chair of the Centre for Applied Mathematics-Ecole Polytechnique and Europorte.

Appendix

7.1. Equivalent Hydraulic Model

$$\frac{d \text{SOC}(t)}{dt} = -\delta I(t) \quad (46)$$

$$\frac{d \text{CSC}(t)}{dt} = \frac{g}{B(1-B)} \text{SOC}(t) - \frac{g}{B(1-B)} \text{CSC}(t) - \frac{\delta}{(1-B)} I(t) \quad (47)$$

$$\frac{dT(t)}{dt} = \frac{1}{m_{cell} C_p_{cell}} [h A (T(t) - T_{amp}) + Q_{gen}(t)] \quad (48)$$

$$Q_{gen}(t) = \frac{I(t)}{C_f B_c} \left[V_r(t) - (U^+(t) - U(t)) - \left(\frac{T_{ref} D S(t)}{Frd} \right) \right] \quad (49)$$

$$\begin{aligned} D S(t) = 0.5609 \times 10^3 \times \left(\frac{\text{SOC}(t)}{0.68} \right)^5 - 1.3440 \times 10^3 \times \left(\frac{\text{SOC}(t)}{0.68} \right)^4 + 1.1877 \times 10^3 \times \left(\frac{\text{SOC}(t)}{0.68} \right)^3 - 0.6072 \times 10^3 \times \left(\frac{\text{SOC}(t)}{0.68} \right)^2 \\ + 0.2378 \times 10^3 \times \left(\frac{\text{SOC}(t)}{0.68} \right)^1 - 0.0397 \times 10^3 \times \left(\frac{\text{SOC}(t)}{0.68} \right)^0 \end{aligned} \quad (50)$$

$$CSC^+(t) = -\rho \times SOC(t) + \sigma \quad (51)$$

$$\begin{aligned} U^+(t) = [0.654807602368402 \times (1 - CSC^+(t))^{3.196972561445755}] + 3.85516954 + 1.247319422 \\ \times (1 - CSC^+(t, 1)) - 11.15240126 \times (1 - CSC^+(t))^2 + 42.8184855 \times (1 - CSC^+(t))^3 \\ - 67.71099749 \times (1 - CSC^+(t))^4 + 42.50815332 \times (1 - CSC^+(t))^5 \\ - 6.13244713 \times 10^{-4} \times \exp(-7.657419995 \times (CSC^+(t)^{115})) \end{aligned} \quad (52)$$

$$\begin{aligned} U(t) = 8.002296379 + 5.064722977 \times CSC(t) - 12.57808059 \times CSC(t)^{\frac{1}{2}} \\ - 8.632208755 \times 10^{-4} \times CSC(t)^{-1} + 2.176468281 \times 10^{-5} \times CSC(t)^{\frac{3}{2}} \\ - 0.4601573522 \times \exp(15.0 \times (0.06 - CSC(t))) \\ - 0.5536351675 \times \exp(-2.432630003 \times (CSC(t) - 0.92)) \end{aligned} \quad (53)$$

$$\begin{aligned} V_r(t) = -U(t) + U^+(t) - (R_f \delta I(t)) - \left(\frac{1}{C} \right) \times \left(\operatorname{asinh} \left(\frac{1 \times \theta \times I(t)}{\sqrt{CSC(t) \times (1 - CSC(t))}} \right) \right) \\ + \left(\frac{1}{C} \right) \times \left(\operatorname{asinh} \left(\frac{-1 \times \theta^+ \times I(t)}{\sqrt{CSC^+(t) \times (1 - CSC^+(t))}} \right) \right) \end{aligned} \quad (54)$$

Electrochemical parameters of the EHM

$$\delta = 0.0000054581$$

$$g = 0.042653404659871$$

$$\beta = 0.7$$

$$Bc = 0.16 \text{ [Ah]}$$

$$C_f = 34 \text{ [}\frac{\text{A}}{\text{m}^2}\text{]}$$

$$Frd = 96487 \text{ [}\frac{\text{C}}{\text{mol}}\text{]}$$

$$T_{ref} = 298.15$$

$$\Psi = 0.5$$

$$R = 8.314472 \text{ [J K}^{-1} \text{ mol}^{-1}\text{]}$$

$$C = (\Psi Frd) / (R \times T_{ref})$$

$$\theta^+ = 0.003753150372049$$

$$\theta = 0.002390386732068$$

(continued on next page)

(continued)

$R_f = 0.000846744769459 [\Omega \text{ m}^2]$
$\rho = 0.798857289559742$
$\sigma = 1.001138873133620$
Heat transfer parameters of the EHM
$h = 10 [\frac{\text{W}}{\text{m}^2 \text{K}}]$
$A = 4.7e-4 [\text{m}^2]$
$m_{\text{cell}} = 0.0045 [\text{kg}]$
$Cp_{\text{cell}} = 800 [\frac{\text{J}}{\text{kg K}}]$

7.2. Pseudocodes for different methodology steps

Algorithm 1

Data generation.

Input : Dynamic model of the system F , MPC objective function J , its parameters (N_p, N_u, N_c, P, Q, R), constraints g_i , and allowable bounds on the state and control variables $x_{\min/\max}$ and $u_{\min/\max}$.
Output : Training dataset \mathcal{D} that contains state-control patterns $[x_{t,i}, u_{t+1,i}^*]$, with $i = 1, \dots, n^{tr} \times N^{trn}$.
Parameters: Number of combinations of initial conditions $n^{tr'}$ and MPC time horizon N^{trn} .

```

1 Sample a number of  $n^{tr'}$  combinations of ICs for the state variables  $x_{0,i}$ , such that  $x_{\min} \leq x_{0,i} \leq x_{\max}$ .
2 for  $i = 1, \dots, n^{tr'}$  do
3   if  $g_i(x_{0,i}, u_0) \leq 0$  then
4     | include  $x_{0,i}$  the to initial conditions set: ICs  $\leftarrow x_{0,i}$ .
5   end
6 end
//get the number of the feasible ICs  $n^{tr}$ .
7 for  $i = 1, \dots, n^{tr}$  do
8   initialize the MPC problem with one combination of the feasible ICs  $x_{0,i}$ .
9   for  $t = 1, \dots, N^{trn}$  do
10    | solve MPC problem (Eqs.(1):(7)).
11    | collect the optimal control  $u_{t,i}^*$ , and the corresponding state variable  $x_{t-1,i}$ .
12   end
13  collect the  $i$ -th state-control trajectory  $[u_{1,i}^*, u_{2,i}^*, \dots, u_{N^{trn},i}^*] = [x_{0,i}, x_{1,i}, \dots, x_{N^{trn}-1,i}]$ .
14 End
15 //unfold the  $n^{tr}$  pairs of state-control trajectories.
16  $\mathcal{D} \leftarrow [x_{t,i}, u_{t+1,i}^*], i = 1, \dots, n^{tr} \times N^{trn}$ .
17 return dataset  $\mathcal{D}$ .
```

Algorithm 2

Development of control laws based on DNNs.

Input : Training dataset $\mathcal{D} = [x_{t,i}, u_{t+1,i}^*]$, with $i = 1, \dots, n^{tr} \times N^{trn}$.
Output : ML-based control laws \mathcal{F}_{DNN}^j , $j = 1, \dots, v$.
// note: in the addressed application of battery charging control $v = 1$.
Hyperparameters: DNNs architecture (number of layers and neurons), transfer function f , and number of training epochs n_{ep} .

```

1 Initialize the parameters  $\theta^j = \{\varpi^j, b^j\}$  of the DNN model.
2 for  $j = 1, \dots, v$  do
3   for  $t = 1, \dots, n_{ep}$  do
4     | update the weights and biases  $(\varpi_t^j, b_t^j)$  with respect to the training loss function.
5   end
6 end
7 return  $\mathcal{F}_{DNN}^j$ .
```

Algorithm 3

Evaluation of the open-loop accuracy of the control laws.

Input : The fitted ML-based control laws $\mathcal{F}_{DNN}^j, j = 1, \dots, v$, and the unfolded test dataset $\mathcal{D}^{ts} = [x_{t,i}^{ts}, u_{t+1,i}^{ts}]$, with $i = 1, \dots, n^{ts} \times N^{fnl}$.
Output : Open-loop accuracy.

```

1   for  $j = 1, \dots, v$  do
2     for  $i = 1, \dots, n^{ts} \times N^{fnl}$  do
3        $\hat{u}_{t+1,i}^{j*} = \mathcal{F}_{DNN}^j(x_{t,i}^{ts}; \theta^j)$ .
4     end
5   end
6   calculate NRMSE (Eq.(9)) using test dataset  $\mathcal{D}^{ts}$ .
```

Algorithm 4

Evaluation of the closed-loop accuracy of the control laws.

Input : The fitted ML-based control laws \mathcal{F}_{DNN}^j , the folded test dataset $\mathcal{D}^{ts} = [x_{0,i}^{ts}, x_{1,i}^{ts}, \dots, x_{N^{fnl}-1,i}^{ts}] - [u_{1,i}^{ts}, u_{2,i}^{ts}, \dots, u_{N^{fnl},i}^{ts}]$ with $i = 1, \dots, n^{ts}$, and the dynamic model of the system \mathbf{F} .
Output : Closed-loop accuracy of the control laws.

```

1   for  $i = 1, \dots, n^{ts}$  do
2     for  $t = 1, \dots, N^{fnl}$  do
3       if  $t = 1$  then
4          $\tilde{x}_{t,i}^{ts} = x_{0,i}^{ts}$ 
5       end
6       for  $j = 1, \dots, v$  do
7          $\hat{u}_{t+1,i}^{j*} = \mathcal{F}_{DNN}^j(\tilde{x}_{t,i}^{ts}; \theta^j)$ .
8       end
9        $\tilde{x}_{t+1,i}^{ts} = \mathbf{F}(\tilde{x}_{t,i}^{ts}, \hat{u}_{t+1,i}^{*})$ .
10      end
11      //collect the i-th predicted state-control trajectory
12       $[x_{0,i}^{ts}, \tilde{x}_{1,i}^{ts}, \dots, \tilde{x}_{N^{fnl}-1,i}^{ts}] - [u_{1,i}^{ts}, u_{2,i}^{ts}, \dots, u_{N^{fnl},i}^{ts}]$ .
13    end
14    calculate the NRMSE of the control variables (Eq.(9)).
15    calculate The NRMSE of the state variables (Eq.(9)).
16    calculate the constraints violation.
```

7.3. Feedforward DNN structure

With this very general definition, many deep learning structures are available in the literature, such as feedforward DNNs, autoencoders, convolutional neural networks, and deep long short-term memory ANNs (Fan et al., 2020). A feedforward DNN (Fig. 4-(a)) consists of multiple data processing units called neurons (blue and green circles) arranged in a structure with a certain number of layers (pink rectangles) (Kumar et al., 2021). The neurons are interconnected (grey arrows) to process the information, and the importance of each connection is calibrated by a weight value. Additionally, an independent input, called bias (brown arrows), is introduced for each neuron.

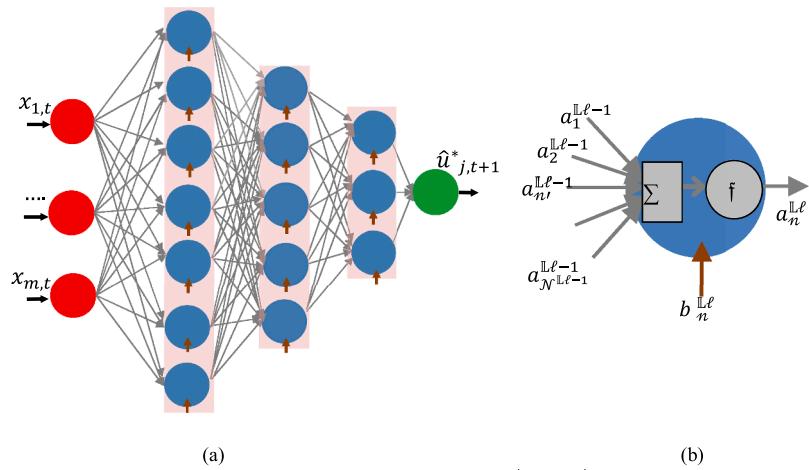


Fig. 30. (a) General schematic illustration of the DNN, $\hat{u}_{j,t+1}^* = F_{DNN}^j(x_t; \theta^j)$, consisting in three hidden layers, and (b) illustration of the mechanism of a neuron.

Fig. 30-(a) depicts a schematic representation of a feedforward DNN, $\hat{u}_{j,t+1}^* = F_{DNN}^j(x_t; \theta^j)$, comprising three hidden layers. **Fig. 30-(b)** illustrates the mechanism of the generic n^{th} neuron in the generic l^{th} hidden layer, L_l with $l=1,..,L$, where the inputs to this neuron are the outputs of each neuron in the preceding layer L_{l-1} , in addition to an independent bias $b_n^{L_l}$. The output of a generic neuron can be expressed as follows:

$$a_n^{L_l} = f\left(b_n^{L_l} + \sum a_n^{L_{l-1}} w_{n^{L_{l-1}}, n^{L_l}}\right), \quad n = 1, \dots, N^{L_l}, \quad n = 1, \dots, N^{L_{l-1}}$$

where $a_n^{L_{l-1}}$ represents the output of the n^{th} neuron in the preceding layer L_{l-1} , $w_{n^{L_{l-1}}, n^{L_l}}$ denotes the weight value assigned to the connection between these two neurons, N^{L_l} is the number of neurons in layer L_l , while $N^{L_{l-1}}$ is the number of neurons in layer L_{l-1} . The function f serves as an activation function, which can be either linear, sigmoid, tanh-sigmoid, or RELU (Katz et al., 2020).

Training a neural network entails solving a nonlinear optimization problem in which a cost function is minimized to determine the optimal values of weights and biases. The cost function is typically associated with the errors between the network's predicted outputs and their target values. To select the number of hidden layers, their respective sizes, the type of activation functions, and the training algorithm, a cut-and-try procedure is adopted which seeks the best compromise between prediction accuracy and structural simplicity (Maddalena et al., 2020).

7.4 Supplementary figures

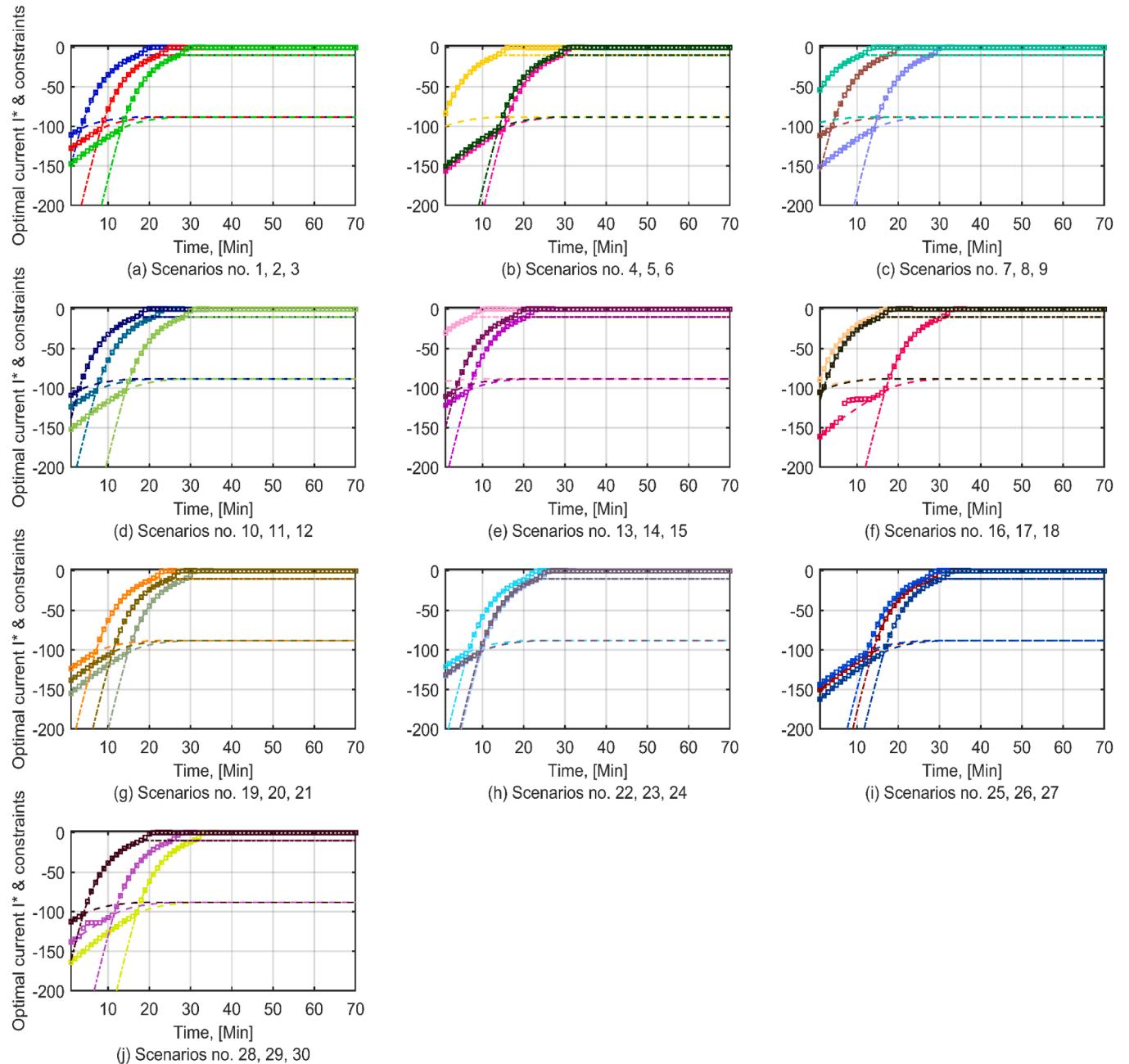


Fig. 31. Performance of the 30 predicted closed-loop charging current trajectories with respect to the health constraints (in case of $T_{max} = 40 C^0$), where each color distinguishes one testing scenario. Squares are the predicted closed-loop charging current, dashed lines are the health constraints $I_{t+k} \geq \omega_1 CSC_{t+k} + \xi_1$ and dash-dotted lines are the health constraints $I_{t+k} \geq \omega_2 CSC_{t+k} + \xi_2$.

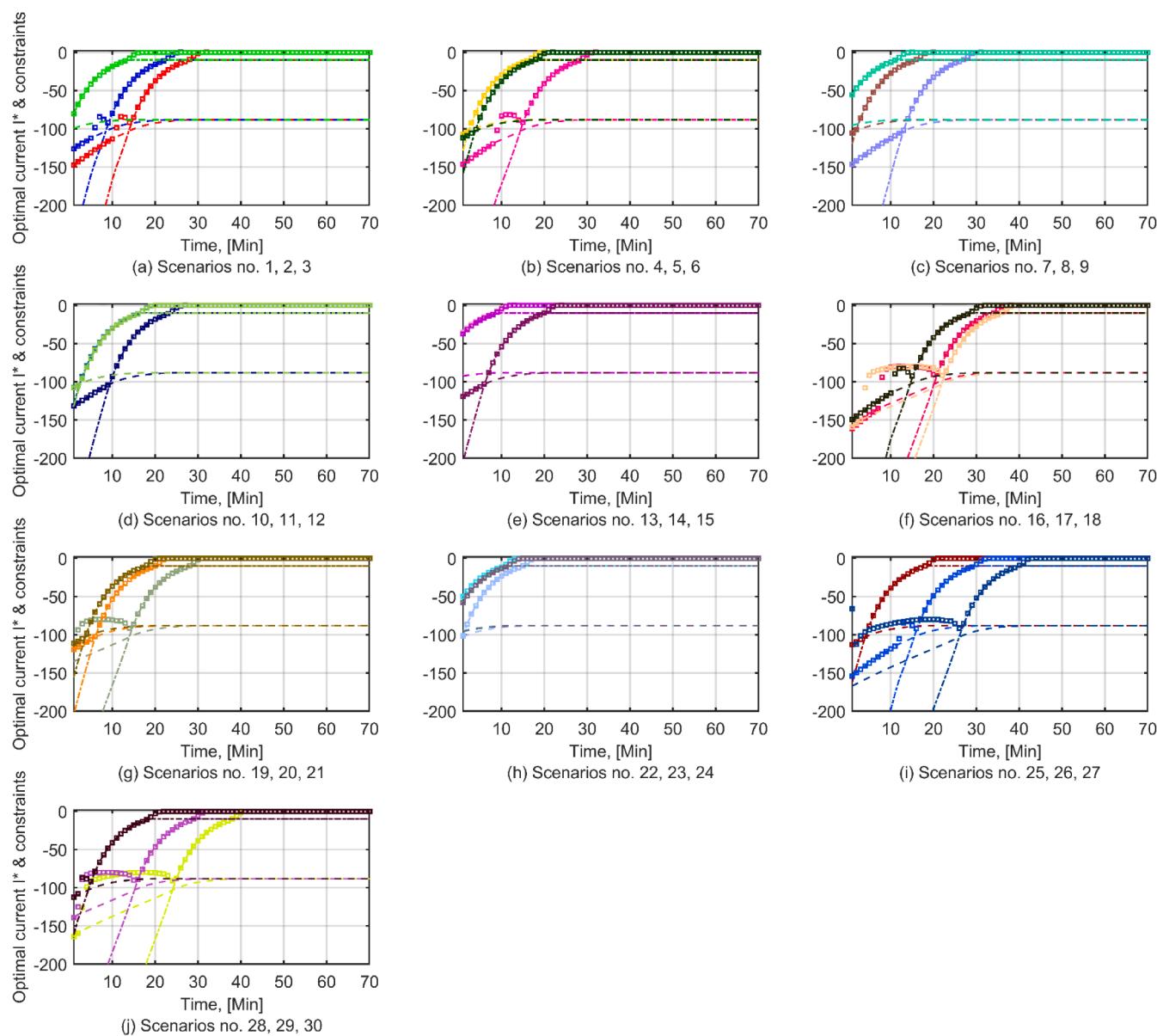


Fig. 32. Performance of the 30 predicted closed-loop charging current trajectories with respect to the health constraints (in case of $T_{max} = 30 C^0$), where each color distinguishes one testing scenario. Squares are the predicted closed-loop charging current, dashed lines are the health constraints $I_{t+k} \geq \omega_1 CSC_{t+k} + \xi_1$ and dash-dotted lines are the health constraints $I_{t+k} \geq \omega_2 CSC_{t+k} + \xi_2$.

Data availability

The data that has been used is confidential.

References

- Abbasi, M.H., Arjmandzadeh, Z., Zhang, J., Xu, B., Krovi, V., 2024. Deep reinforcement learning based fast charging and thermal management optimization of an electric vehicle battery pack. *J. Energy Storage* 95 (June), 112466. <https://doi.org/10.1016/j.est.2024.112466>.
- Baraldi, P., Castellano, A., Shokry, A., Gentile, U., Serio, L., Zio, E., 2020. A feature selection-based approach for the identification of critical components in complex technical infrastructures: application to the CERN Large Hadron Collider. *Reliab. Eng. Syst. Saf.* 201, 106974. <https://doi.org/10.1016/j.ress.2020.106974>.
- Biegler, L.T., 2007. An overview of simultaneous strategies for dynamic optimization. *Chem. Eng. Process.: Process Intensif.* 46 (11), 1043–1053.
- Chang, H., Krieger, A., Astolfi, A., Pistikopoulos, E.N., 2014. Robust multi-parametric model predictive control for LPV systems with application to anaesthesia. *J. Process Control* 24 (10), 1538–1547. <https://doi.org/10.1016/j.jprocont.2014.07.005>.
- Charitopoulos, V.M., Papageorgiou, L.G., Dua, V., 2019. Closed-loop integration of planning, scheduling and multi-parametric nonlinear control. *Comput. Chem. Eng.* 122, 172–192. <https://doi.org/10.1016/j.compchemeng.2018.06.021>.
- Couto, L.D., Romagnoli, R., Park, S., Zhang, D., Moura, S.J., Kinnaert, M., Garone, E., 2021. Faster and healthier charging of lithium-ion batteries via constrained feedback control. *IEEE Trans. Control Syst. Technol.* 30 (5), 1990–2001. <https://doi.org/10.1109/TCST.2021.3135149>.
- Domínguez, L.F., Narciso, D.A., Pistikopoulos, E.N., 2010. Recent advances in multiparametric nonlinear programming. *Comput. Chem. Eng.* 34 (5), 707–716. <https://doi.org/10.1016/j.compchemeng.2009.10.012>.
- Domínguez, L.F., Pistikopoulos, E.N., 2011. Recent advances in explicit multiparametric nonlinear model predictive control. *Ind. Eng. Chem. Res.* 50 (2), 609–619. <https://doi.org/10.1021/ie100245z>.
- Fabiani, F., Goulart, P.J., 2023. Reliably-stabilizing piecewise-affine neural network controllers. *IEE Trans. Automat. Contr.* 68 (9), 5201–5215. <https://doi.org/10.1109/TAC.2022.3216978>.
- Falsone, A., Bianchi, F., Prandini, M., 2023. Dealing with infeasibility in multiparametric programming for application to explicit model predictive control. *Automatica* 157, 111279. <https://doi.org/10.1016/j.automatica.2023.111279>.
- Fan, Y., Xiao, F., Li, C., Yang, G., Tang, X., 2020. A novel deep learning framework for state of health estimation of lithium-ion battery. *J. Energy Storage* 32 (June), 101741. <https://doi.org/10.1016/j.est.2020.101741>.

- Floriano, B.R.O., Vargas, A.N., Ishihara, J.Y., Ferreira, H.C., 2022. Neural-network-based model predictive control for consensus of nonlinear systems. *Eng. Appl. Artif. Intell.* 116 (July), 105327. <https://doi.org/10.1016/j.engappai.2022.105327>.
- Gao, Y., Member, S., Zhang, X., Member, S., Guo, B., Zhu, C., Wiedemann, J., Wang, L., & Cao, J. (2020). Health-aware multiobjective optimal model for lithium-ion battery. *16*(5), 3417–3429.
- Garud, S.S., Karimi, I.A., Kraft, M., 2017. Design of computer experiments: a review. *Comput. Chem. Eng.* 106, 71–95. <https://doi.org/10.1016/j.compchemeng.2017.05.010>.
- Goldar, A., Romagnoli, R., Couto, L.D., Romero, A., Kinnaert, M., Garone, E., 2020. MPC strategies based on the equivalent hydraulic model for the fast charge of commercial Li-ion batteries. *Comput. Chem. Eng.* 141, 107010. <https://doi.org/10.1016/j.compchemeng.2020.107010>.
- Hao, Y., Lu, Q., Wang, X., Jiang, B., 2024. Adaptive model-based reinforcement learning for fast-charging optimization of lithium-ion batteries. *IEE Trans. Industr. Inform.* 20 (1), 127–137. <https://doi.org/10.1109/TII.2023.3257299>.
- Hu, X., Perez, H.E., Moura, S.J., 2015. Battery charge control with an electro-thermal-aging coupling. In: ASME 2015 Dynamic Systems and Control Conference, DSCC 2015, 1. <https://doi.org/10.1115/DSCC2015-9705>.
- Karg, B., Lucia, S., 2020. Efficient representation and approximation of model predictive control laws via Deep learning. *IEE Trans. Cybern.* 50 (9), 3866–3878. <https://doi.org/10.1109/TCYB.2020.2999556>.
- Kim, M., Baek, J., & Han, S. (2020). Optimal Charging Method for Effective Li-ion Battery Life Extension Based on Reinforcement Learning. 8–11.
- Katz, J., Pappas, I., Avraamidou, S., Pistikopoulos, E.N., 2020. Integrating deep learning models and multiparametric programming. *Comput. Chem. Eng.* 136, 106801. <https://doi.org/10.1016/j.compchemeng.2020.106801>.
- Kis, K., Klaucio, M., Kvasnicka, M., 2021. Explicit MPC in the form of sparse neural networks. In: Proceedings of the 2021 23rd International Conference on Process Control, PC 2021, pp. 163–168. <https://doi.org/10.1109/PC52310.2021.9447528>.
- Kolluri, S., Aduru, S.V., Pathak, M., Braatz, R.D., Subramanian, V.R., 2020. Real-time nonlinear model predictive control (NMPC) strategies using physics-based models for advanced lithium-ion battery management system (BMS). *J. Electrochem. Soc.* 167 (6), 063505. <https://doi.org/10.1149/1945-7111/ab7bd7>.
- Kouramas, K.I., Faisca, N.P., Panos, C., Pistikopoulos, E.N., 2011. Explicit/multiparametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming. *Automatica* 47 (8), 1638–1645. <https://doi.org/10.1016/j.automatica.2011.05.001>.
- Kumar, P., Rawlings, J.B., Wright, S.J., 2021. Industrial, large-scale model predictive control with structured neural networks. *Comput. Chem. Eng.* 150, 107291. <https://doi.org/10.1016/j.compchemeng.2021.107291>.
- Li, Y., Hu, K., Cao, Y., 2022. Using stochastic programming to train neural network approximation of nonlinear MPC laws. *Automatica* 146, 110665. <https://doi.org/10.1016/j.automatica.2022.110665>.
- Li, K., Li, K., Ma, H., Zhang, J., Peng, Q., 2018. Multi-objective optimization of charging patterns for lithium-ion battery management. *Energy Convers. Manage.* 159 (January), 151–162. <https://doi.org/10.1016/j.enconman.2017.12.092>.
- Li, K., Li, K., Yang, Z., Zhang, C., Deng, J., 2017. An advanced lithium-ion battery optimal charging strategy based on a coupled thermoelectric model. *Electrochim. Acta* 225, 330–344. <https://doi.org/10.1016/j.electacta.2016.12.129>.
- Li, K., Li, K., Zhang, C., 2017. Constrained generalized predictive control of battery charging process based on a coupled thermoelectric model. *J. Power. Sources.* 347, 145–158. <https://doi.org/10.1016/j.jpowsour.2017.02.039>.
- Liu, K., Zou, C., Li, K., Wik, T., 2018. Charging pattern optimization for lithium-ion batteries with an electrothermal-aging model. *IEE Trans. Industr. Inform.* 14 (12), 5463–5474. <https://doi.org/10.1109/TII.2018.2866493>.
- Maddalena, E.T., da Moraes, C.G.S., Waltrich, G., Jones, C.N., 2020. A neural network architecture to learn explicit MPC controllers from data. *IFAC-PapersOnLine* 53 (2), 11362–11367. <https://doi.org/10.1016/j.ifacol.2020.12.546>.
- Pappas, I., Kenefake, D., Burnak, B., Avraamidou, S., Ganesh, H.S., Katz, J., Diangelakis, N.A., Pistikopoulos, E.N., 2021. Multiparametric Programming in Process Systems engineering: recent developments and path forward. *Front. Chem. Eng.* 2 (January), 1–15. <https://doi.org/10.3389/fceng.2020.620168>.
- Park, S., Pozzi, A., Whitmeyer, M., Perez, H., Joe, W.T., Raimondo, D.M., Moura, S., 2020. Reinforcement learning-based fast charging control strategy for li-ion batteries. In: CCTA 2020 - 4th IEEE Conference on Control Technology and Applications, 8, pp. 100–107. <https://doi.org/10.1109/CCTA41146.2020.9206314>.
- Pathak, M., Kolluri, S., Subramanian, V.R., 2017. Generic model control for lithium-ion batteries. *J. Electrochem. Soc.* 164 (6), A973–A986. <https://doi.org/10.1149/2.1521704jes>.
- Perez, H.E., Hu, X., Dey, S., Moura, S.J., 2017. Optimal charging of Li-ion batteries with coupled electro-thermal-aging dynamics. *IEE Trans. Veh. Technol.* 66 (9), 7761–7770. <https://doi.org/10.1109/TVT.2017.2676044>.
- Pistikopoulos, E.N., Dominguez, L., Panos, C., Kouramas, K., Chinchuluun, A., 2012. Theoretical and algorithmic advances in multi-parametric programming and control. *Comput. Manag. Sci.* 9 (2), 183–203. <https://doi.org/10.1007/s10287-012-0144-4>.
- Pistikopoulos, E.N., Dua, V., Bozinis, N.A., Bemporad, A., Morari, M., 2002. On-line optimization via off-line parametric optimization tools. *Comput. Chem. Eng.* 26 (2), 175–185. [https://doi.org/10.1016/S0098-1354\(01\)00739-6](https://doi.org/10.1016/S0098-1354(01)00739-6).
- Pistikopoulos, E.N., Georgiadis, M.C., Dua, V., 2011. Multi-parametric model-based control. In: Theory and Applications, 2. Wiley-VCH Verlag GmbH & Co. KGaA.
- Pozzi, A., Barbierato, E., Toti, D., 2023. Optimizing battery charging using neural networks in the presence of unknown states and parameters. *Sensors* 23 (9). <https://doi.org/10.3390/s23094404>.
- Pozzi, A., Moura, S., Toti, D., 2023. A deep learning-based predictive controller for the optimal charging of a lithium-ion cell with non-measurable states. *Comput. Chem. Eng.* 173 (March), 108222. <https://doi.org/10.1016/j.compchemeng.2023.108222>.
- Pozzi, A., Torchio, M., Raimondo, D.M., 2018. Assessing the performance of model-based energy saving charging strategies in Li-ion cells. In: 2018 IEEE Conference on Control Technology and Applications, CCTA 2018, pp. 806–811. <https://doi.org/10.1109/CCTA.2018.8511463>.
- Rivotti, P., Lambert, R.S.C., Pistikopoulos, E.N., 2012. Combined model approximation techniques and multiparametric programming for explicit nonlinear model predictive control. *Comput. Chem. Eng.* 42, 277–287. <https://doi.org/10.1016/j.compchemeng.2012.01.009>.
- Romero, A., Goldar, A., Couto, L.D., Garone, E., Maestre, J.M., 2019. Fast charge of li-ion batteries using a two-layer distributed mpc with electro-chemical and thermal constraints. In: 2019 18th European Control Conference, ECC 2019, pp. 1796–1803. <https://doi.org/10.23919/ECC.2019.8795679>.
- Romero, A., Goldar, A., Garone, E., 2019. A model predictive control application for a constrained fast charge of lithium-ion batteries. In: Proceedings of the 13th International Modelica Conference. Regensburg, Germany, 157, pp. 229–238. <https://doi.org/10.3384/ecp19157229>. March 4–6, 2019.
- Rosewater, D.M., Copp, D.A., Nguyen, T.A., Byrne, R.H., Santoso, S., 2019. Battery energy storage models for optimal control. *IEE Access.* 7, 178357–178391. <https://doi.org/10.1109/ACCESS.2019.2957698>.
- Ruiz-Moreno, S., Frejo, J.R.D., Camacho, E.F., 2021. Model predictive control based on deep learning for solar parabolic-trough plants. *Renew. Energy* 180, 193–202. <https://doi.org/10.1016/j.renene.2021.08.058>.
- Schorsh, J., Couto, L.D., Kinnaert, M., 2016. SOC and SOH estimation for Li-ion battery based on an equivalent hydraulic model. Part II: SOH power fade estimation. In: Proceedings of the American Control Conference, 2016-July, pp. 4029–4034. <https://doi.org/10.1109/ACC.2016.7525554>.
- Shokry, A., Baraldi, P., Zio, E., Espuna, A., 2020. Dynamical surrogate modelling for multistep-ahead prediction of multivariate nonlinear chemical processes. *Ind. Eng. Chem. Res.* <https://doi.org/10.1021/acs.iecr.0c00729> acs.iecr.0c00729.
- Shokry, A., Dombayci, C., Espuna, A., 2016a. Multiparametric metamodels for model predictive control of chemical processes. *Comput. Aided Chem. Eng.* 38. <https://doi.org/10.1016/B978-0-444-63428-3.50161-2>.
- Shokry, A., Dombayci, C., Espuna, A., 2016b. Multiparametric metamodels for model predictive control of chemical processes. *Comput. Aided Chem. Eng.* 38 (2), 937–942. <https://doi.org/10.1016/B978-0-444-63428-3.50161-2>.
- Shokry, A., Medina-González, S., Baraldi, P., Zio, E., Moulines, E., Espuna, A., 2021. A machine learning-based methodology for multi-parametric solution of chemical processes operation optimization under uncertainty. *Chem. Eng. J.* 425. <https://doi.org/10.1016/j.cej.2021.131632>.
- Shokry, A., Moulines, E., 2022. Health-constrained explicit model predictive control based on deep-neural networks applied to real-time charging of batteries. *IFAC-PapersOnLine* 55 (16), 142–147. <https://doi.org/10.1016/j.ifacol.2022.09.014>.
- Sivaram, A., Das, L., Venkatasubramanian, V., 2020. Hidden representations in deep neural networks: part 1. Classification problems. *Comput. Chem. Eng.* 134, 106669. <https://doi.org/10.1016/j.compchemeng.2019.106669>.
- Sun, M., Villanueva, M.E., Pistikopoulos, E.N., Chachuat, B., 2019. Methodology for robust multi-parametric control in linear continuous-time systems. *J. Process. Control* 73, 58–74. <https://doi.org/10.1016/j.jprocont.2018.09.005>.
- Tian, N., Fang, H., Chen, J., Wang, Y., 2019. Nonlinear double-capacitor model for rechargeable batteries: modeling, identification and validation. *ArXiv*, 29 (1), 370–384.
- Tian, N., Fang, H., Wang, Y., 2019a. Parameter identification of the nonlinear double-capacitor model for lithium-ion batteries: from the wiener perspective. In: Proceedings of the American Control Conference, pp. 897–902. <https://doi.org/10.23919/acc.2019.8814957>, 2019-July.
- Tian, N., Fang, H., Wang, Y., 2019b. Real-time optimal charging for lithium-ion batteries via explicit model predictive control. In: IEEE International Symposium on Industrial Electronics, pp. 2001–2006. <https://doi.org/10.1109/ISIE.2019.8781259>, 2019-June(December).
- Tian, N., Fang, H., Wang, Y., 2021. Real-time optimal lithium-ion battery charging based on explicit model predictive control. *IEEE Trans. Industr. Inform.* 17 (2), 1318–1330. <https://doi.org/10.1109/TII.2020.2983176>.
- Torchio, M., Magni, L., Braatz, R.D., Raimondo, D.M., 2017. Design of piecewise affine and linear time-varying model predictive control strategies for advanced battery management systems. *J. Electrochem. Soc.* 164 (4), A949–A959. <https://doi.org/10.1149/2.0201706jes>.
- Torchio, M., Magni, L., Gopaluni, R.B., Braatz, R.D., Raimondo, D.M., 2016. LIONSIMBA: A matlab framework based on a finite volume model suitable for Li-ion battery design, simulation, and control. *J. Electrochem. Soc.* 163 (7), A1192–A1205. <https://doi.org/10.1149/2.0291607jes>.
- Vidal, C., Malysz, P., Kollmeyer, P., Emadi, A., 2020. Machine learning applied to electrified vehicle battery state of charge and state of health estimation: state-of-the-art. *IEE Access.* 8, 52796–52814. <https://doi.org/10.1109/ACCESS.2020.2980961>.
- Wang, D., Miao, Q., Pecht, M., 2013. Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model. *J. Power. Sources.* 239, 253–264. <https://doi.org/10.1016/j.jpowsour.2013.03.129>.
- Wei, Z., Quan, Z., Wu, J., Li, Y., Pou, J., Zhong, H., 2022. Deep deterministic policy gradient-DRL enabled multiphysics-constrained fast charging of lithium-ion battery. *IEEE Trans. Ind. Electronics* 69 (3), 2588–2598. <https://doi.org/10.1109/TIE.2021.3070514>.
- Xavier, M.A., Trimboli, M.S., 2015. Lithium-ion battery cell-level control using constrained model predictive control and equivalent circuit models. *J. Power. Sources.* 285, 374–384. <https://doi.org/10.1016/j.jpowsour.2015.03.074>.

- Yan, J., Xu, G., Qian, H., Xu, Y., Song, Z., 2011. Model predictive control-based fast charging for vehicular batteries. *Energies (Basel)* 4 (8), 1178–1196. <https://doi.org/10.3390/en4081178>.
- Yang, Y., He, J., Chen, C., Wei, J., 2024. Balancing awareness fast charging control for lithium-ion battery pack using deep reinforcement learning. *IEEE Trans. Ind. Electronics* 71 (4), 3718–3727. <https://doi.org/10.1109/TIE.2023.3274853>.
- Zhang, C., Jiang, J., Gao, Y., Zhang, W., Liu, Q., Hu, X., 2017. Charging optimization in lithium-ion batteries based on temperature rise and charge time. *Appl. Energy* 194, 569–577. <https://doi.org/10.1016/j.apenergy.2016.10.059>.
- Zou, C., Hu, X., Wei, Z., Tang, X., 2017. Electrothermal dynamics-conscious lithium-ion battery cell-level charging management via state-monitored predictive control. *Energy* 141, 250–259. <https://doi.org/10.1016/j.energy.2017.09.048>.