

Article

Optimizing Battery Charging Using Neural Networks in the Presence of Unknown States and Parameters

Andrea Pozzi , Enrico Barbierato  and Daniele Toti 

Department of Mathematics and Physics, Catholic University of the Sacred Heart, 25133 Brescia, Italy; enrico.barbierato@unicatt.it (E.B.); daniele.toti@unicatt.it (D.T.)

* Correspondence: andrea.pozzi@unicatt.it

Abstract: This work investigates the effectiveness of deep neural networks within the realm of battery charging. This is done by introducing an innovative control methodology that not only ensures safety and optimizes the charging current, but also substantially reduces the computational complexity with respect to traditional model-based approaches. In addition to their high computational costs, model-based approaches are also hindered by their need to accurately know the model parameters and the internal states of the battery, which are typically unmeasurable in a realistic scenario. In this regard, the deep learning-based methodology described in this work was been applied for the first time to the best of the authors' knowledge, to scenarios where the battery's internal states cannot be measured and an estimate of the battery's parameters is unavailable. The reported results from the statistical validation of such a methodology underline the efficacy of this approach in approximating the optimal charging policy.

Keywords: machine learning; deep learning; neural networks; computational complexity; predictive control; battery management systems



Citation: Pozzi, A.; Barbierato, E.; Toti, D. Optimizing Battery Charging Using Neural Networks in the Presence of Unknown States and Parameters. *Sensors* **2023**, *23*, 4404. <https://doi.org/10.3390/s23094404>

Academic Editor: Chun Sing Lai

Received: 27 March 2023

Revised: 25 April 2023

Accepted: 28 April 2023

Published: 30 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Managing Lithium-Ion batteries is a multifaceted undertaking that necessitates the careful negotiation of various factors and approaches, which may frequently be at odds with one another. This involves defining appropriate current profiles for both charging and discharging the batteries, implementing mechanisms to mitigate performance degradation over time, and ensuring their safety throughout their lifespan. Given the breadth of these considerations, it is evident that managing Lithium-Ion batteries is a challenging task that requires meticulous attention [1]. Battery management systems used in the industry usually rely on rule-based algorithms. One widely used class of algorithms is based on the constant-current/constant-voltage principle, which seeks to optimize the charging time of the batteries while staying within the voltage limits that have been established [2]. Despite their widespread use, these mechanisms tend to employ a conservative approach that often fails to fully leverage the potential of Lithium-Ion batteries, whether it be to expedite the charging time or safeguard their safety. Consequently, the outcomes of these algorithms are often suboptimal. Moreover, due to the fixed nature of voltage constraints, such basic algorithms overlook the gradual deterioration of batteries and the progressive alteration of their internal characteristics as they undergo numerous cycles of charging and discharging [3].

In reality, there are methods available that can offer a more effective control strategy by exploiting a mathematical model of the battery. Within this context, different control approaches have been considered, such as fuzzy logic [4,5], empirical rules [6,7], and optimization-based strategies [8,9]. Among the latter, one such noteworthy technique is model predictive control (MPC) [10], which has achieved considerable success and widespread adoption in the context of battery management (see for instance [11–18]).

This is owing to its ability to manage complex, nonlinear processes involving multiple variables while satisfying constraints related to inputs and states. Despite its effectiveness in simulations, MPC is not without limitations. Its practical use in real-world scenarios is often impeded by computational complexities, which can render it less efficient and effective than its theoretical potential suggests, especially when accurate nonlinear models of the system are employed. This is due to the nature of the algorithm, which must solve a constrained optimal control problem at each time step in real time. When it is not able to computationally be on par with a fast sampling time of the control law, MPC becomes useless.

To address this challenge, researchers have proposed a potential solution in the form of an explicit MPC algorithm, which has been applied in the context of battery charging by the authors in [19]. This type of MPC, as introduced in [20], aims to reduce the computational burden of real-time operations by simplifying them to the evaluation of a straightforward function. In theory, the explicit MPC should require fewer computations since it precomputes the optimal control action in the form of a piece-wise function of the state and reference vectors, and only needs to detect the region in which the states are located during real-time operations. However, in practice, the detection of such a region can be computationally demanding, especially when dealing with numerous constraints and long prediction horizons, leading to performance degradation that may not be acceptable [21].

The computational cost of the explicit MPC has been a challenge, leading to several works in the literature attempting to address the issue (for instance, [22–24]). Most of these works propose an approximation of the control law, with machine learning models gaining attention in recent years, giving birth to the concept of learning-based model-predictive control. Among the various learning models, deep neural networks have been successful, leading to the development of the so-called deep MPC. Previous works in deep MPC include [25], where the authors describe a model that is robust to input errors, and [21], where a deep predictive controller is shown to be able to exactly represent—with a sufficiently high number of neurons and layers—an explicit MPC control law. As far as the battery control is concerned, a deep MPC formulation has been proven successful in mitigating the issues of the computational complexity of a standard predictive controller in [26,27], where a learning-based charging approach is applied to a battery modeled as an equivalent circuit model and as an electrochemical one, respectively.

Beyond the problem of computational complexity, which can be reduced with the use of approximations based on neural networks, the use of a predictive control paradigm within the context of battery management is also limited by the fact that it relies on the following assumptions: (i) the availability of an accurate model of the battery dynamics (with accurately estimated parameters), (ii) measurability of all the relevant states of the system. Such assumptions hold only in an ideal scenario in which the controller has a perfect knowledge of the system, while in practice the parameters and the internal states of the system need to be inferred from the available measurements (voltage, current, and surface temperature) and only the model structure can be assumed to be known a priori. The problem of parameter and state estimations for Lithium-Ion batteries has been largely discussed in the literature and it is usually considered a key issue in the development of reliable controllers. The proposed solutions usually involve the execution of ad hoc experiments to collect the data required for the estimation of the parameters (see [28]) and the design of suitable observers to reconstruct the state trajectory online (see [29]). However, it is worth highlighting that the accuracy of the state observer is highly related to that of the estimated electrochemical parameters, which can vary greatly, even among cells of the same type, and may change as the battery ages. In conclusion, determining these parameters with the required level of accuracy often needs time-intensive and intrusive experiments.

It is clear that a deep MPC approach that simply approximates an MPC feedback law, such as the one proposed in [26,27], still suffers from the aforementioned issues related to state/parameter estimation. In this paper, an extension is proposed for the work presented in [27], with the aim of considering a more realistic scenario in which only the current,

voltage, and temperature are assumed to be measurable, and the battery parameters are unknown. Specifically, the implementation of an output-based deep MPC is considered here, which is able to charge the battery with the knowledge of the system restricted to the model structure, which is considered here as that of the well-known single particle model (SPM) [30]. The results show that the proposed approach can accurately reproduce the benchmark performance in a practical scenario.

For the first time, to the knowledge of the authors, a computationally-efficient neural network-based predictive controller was successfully employed for the optimal charging of a Lithium-Ion battery in the presence of non-measurable states and unknown parameters.

The structure of this work is as follows: in Section 2, the battery model is presented in detail. Section 3 presents the formulation of the learning-based predictive controller. Section 4 shows the training approach. The experimental results are shown in Section 5, while Section 6 discusses potential improvements to the proposed approach. Finally, Section 7 provides the concluding remarks.

2. Model

The primary categories of models used in sophisticated battery management systems (BMSs) include equivalent circuit models (ECMs) [31,32] and electrochemical models (EMs) [33,34]. ECMs are relatively simple and intuitive, whereas EMs offer a comprehensive explanation of the electrochemical processes occurring within a cell. Electrochemical models are more appropriate for simulation purposes rather than for real-time control applications. Furthermore, the implementation of electrochemical models in a control framework is constrained by issues of identifiability and observability [35]. As a result, researchers have been focusing on the creation of simplified electrochemical models that are quicker to simulate, identifiable, observable, and still provide adequate representation of the internal cell phenomena [36,37]. The single-particle model (SPM) [30], which is obtained from the pseudo-two-dimensional model [38] by considering the two electrodes as spherical particles, is one notable example among these models. SPM is used in this paper to mathematically describe the battery dynamics. Such a simplified electrochemical model has been largely adopted for battery control and estimation of the states, due to its ability in achieving a reasonable trade-off between the computational cost and accuracy (see e.g., [27,39–41]). The accuracy of such a model was demonstrated in [42], among others. Note that the battery model is enriched with the two-state temperature dynamics proposed by the authors in [43] to account for thermal phenomena.

Only the equations pertaining to the primary variables of the model are mentioned below; for a more comprehensive explanation, the reader is directed to reference [27]. Specifically, the variable $\text{soc}(t) \in [0, 1]$ represents the state of charge of the battery, whose temporal evolution is given by:

$$\frac{d \text{soc}(t)}{dt} = \frac{I(t)}{3600C} \quad (1)$$

where the applied current is denoted as $I(t)$, with the convention that a positive current charges the cell and C represents the cell capacity in [Ah]. It is important to note that when the battery is fully charged, the state of charge is at $\text{soc}(t) = 1$, and when completely discharged, the state of charge is at $\text{soc}(t) = 0$. Moreover, the battery voltage is given by the following equation:

$$V(t) = U_p(t) - U_n(t) + \eta_p(t) - \eta_n(t) + R_{sei}I(t) \quad (2)$$

where the terms $U_i(t)$ and $\eta_i(t)$, for $i \in \{n, p\}$, represent the open circuit potential and overpotential, respectively, as defined in Section 2 of [27], while the term $R_{sei}I(t)$ describes the voltage drop in the solid electrolyte interphase (SEI) resistance. Note that the open circuit potentials and the overpotential are nonlinear functions of the applied current, state of charge, and battery average temperature. As far as the latter is concerned, the two-state

model proposed in [43] is adopted here for the thermal dynamics, in which the core and the surface temperatures are represented by $T_c(t)$ and $T_s(t)$, respectively. In particular, it holds that:

$$C_c \frac{dT_c(t)}{dt} = Q(t) - \frac{T_c(t) - T_s(t)}{R_{c,s}} \quad (3a)$$

$$C_s \frac{dT_s(t)}{dt} = \frac{T_c(t) - T_s(t)}{R_{c,s}} - \frac{T_s(t) - T_{env}}{R_{s,e}} \quad (3b)$$

where $R_{c,s}$ and $R_{s,e}$ denote the thermal resistances between the core and surface and between the surface and the external environment, respectively, whereas C_c and C_s represent, respectively, the heat capacity of the cell's core and surface. Finally, $Q(t)$ represents the amount of heat generated, which is defined as follows:

$$Q(t) = |I(t)(V(t) - U_p(t) + U_n(t))|. \quad (4)$$

It is important to highlight that the electrochemical parameters in nominal form have been extracted from the experimental characterization of a commercial cell, specifically the Kokam SLPB 75106100, as presented in [44,45], while the thermal ones are based on those employed by [43].

3. Methodology

This section outlines the methodology behind the learning-based predictive controller. First of all, the nonlinear equations that describe the plant dynamics are formulated in a state space form and discretized as follows:

$$\mathbf{x}(t_{k+1}) = f_d(\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{p}) \quad (5a)$$

$$\mathbf{y}(t_{k+1}) = g(\mathbf{x}(t_{k+1}), \mathbf{u}(t_k), \mathbf{p}) \quad (5b)$$

in order to consider a digital controller that applies a piece-wise constant input at the discrete times t_k , $k \in \mathbb{N}$ with sample time t_s . Specifically, $\mathbf{x}(t_k) \in \mathbb{R}^{n_x}$, $\mathbf{u}(t_k) \in \mathbb{R}^{n_u}$, and $\mathbf{y}(t_k) \in \mathbb{R}^{n_y}$ represent the state, input, and output vectors, respectively. Moreover, the model parameters are represented as $\mathbf{p} \in \mathbb{R}^{n_p}$ (the dependence of a variable on the parameter vector will be made explicit below only where necessary). Furthermore, the functions $f_d : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$ map the current state and the current input into the next state ($x(t_{k+1})$) and the output ($y(t_k)$), respectively. Finally, note that the generic input sequence applied in the time interval $[t_k, t_{k+H}]$, with $H \in \mathbb{N}$, is represented as:

$$\mathbf{u}_{[t_k, t_{k+H}]} = \left[\mathbf{u}(t_k)^\top \mathbf{u}(t_{k+1})^\top \cdots \mathbf{u}(t_{k+H-1})^\top \right]^\top. \quad (6)$$

The rest of the section is organized as follows. The main features of a standard predictive controller are discussed in Section 3.1, while its approximation through a learning-based algorithm under the assumption of state measurability and parameter knowledge is presented in Section 3.2. Finally, in Section 3.3, a novel algorithm is proposed, which still relies on neural networks and consists of the adaptation of the deep MPC for the case in which the states and parameters are unknown.

3.1. Model Predictive Control

Predictive control methodologies that rely on a receding horizon framework [46] have been shown to be particularly effective in dealing with nonlinear processes subject to the input and state constraints [10]. In this context, the model's predictive control scheme computes the optimal control sequence $\mathbf{u}_{[t_k, t_{k+H}]}^*$ over the prediction horizon H at each time step t_k , by solving a constrained optimization problem with a cost function that depends on predictions made by a mathematical model of the plant. Then, according to the receding

horizon paradigm, only the first element $\mathbf{u}^*(t_k)$ of the resulting optimal input sequence is applied, while the remaining future optimal moves are discarded.

Specifically, the following optimization problem is solved at each time step in order to compute the control action:

$$\mathbf{u}_{[t_k, t_{k+H}]}^* = \underset{\mathbf{u}_{[t_k, t_{k+H}]}}{\operatorname{argmin}} J(\mathbf{x}(t_k)) \quad (7)$$

subject to:

$$\text{system dynamic in (5)} \quad (8a)$$

$$\mathbf{u}^{lb} \leq \mathbf{u}(t_i) \leq \mathbf{u}^{ub}, \quad i = k, k+1, \dots, k+H-1 \quad (8b)$$

$$\mathbf{x}^{lb} \leq \mathbf{x}(t_i) \leq \mathbf{x}^{ub}, \quad i = k+1, k+1, \dots, k+H \quad (8c)$$

$$\mathbf{y}^{lb} \leq \mathbf{y}(t_i) \leq \mathbf{y}^{ub}, \quad i = k+1, k+1, \dots, k+H \quad (8d)$$

with $\mathbf{u}^{lb}, \mathbf{u}^{ub} \in \mathbb{R}^{n_u}$ being the limits for the input vector, $\mathbf{x}^{lb}, \mathbf{x}^{ub} \in \mathbb{R}^{n_x}$ the ones for the state vector, and $\mathbf{y}^{lb}, \mathbf{y}^{ub} \in \mathbb{R}^{n_y}$ the ones for the output vector. The cost function $J(\mathbf{x}(t_k))$ to be minimized is formulated as follows:

$$J(\mathbf{x}(t_k)) = \|\mathbf{x}(t_{k+H}) - \mathbf{x}_{\text{ref}}\|_{Q_H}^2 + \sum_{i=k+1}^{k+H} \left[\|\mathbf{x}(t_i) - \mathbf{x}_{\text{ref}}\|_{Q_x}^2 + \|\mathbf{y}(t_i) - \mathbf{y}_{\text{ref}}\|_{Q_y}^2 \right] + \sum_{i=k}^{k+H-1} \|\mathbf{u}(t_i) - \mathbf{u}_{\text{ref}}\|_R^2 \quad (9)$$

where the vectors $\mathbf{x}_{\text{ref}} \in \mathbb{R}^{n_x}$, $\mathbf{y}_{\text{ref}} \in \mathbb{R}^{n_y}$ and $\mathbf{u}_{\text{ref}} \in \mathbb{R}^{n_u}$ correspond to the reference point that the MPC aims to track and the matrices $Q_x \in \mathbb{R}^{n_x \times n_x}$, $Q_y \in \mathbb{R}^{n_y \times n_y}$, $Q_H \in \mathbb{R}^{n_x \times n_x}$, and $R \in \mathbb{R}^{n_u \times n_u}$ are design parameters, with $Q_x, Q_y, Q_H \geq 0$ and $R > 0$. Note that the term $\|\mathbf{x}(t_{k+H}) - \mathbf{x}_{\text{ref}}\|_{Q_H}^2$ represents a suitably tuned terminal penalty used to improve the controller stability.

The MPC control law is defined as follows:

$$\mathbf{u}_{\text{mpc}}(\mathbf{x}(t_k)) = \mathbf{u}(t_k)^* \quad (10)$$

which is computed by solving the problem in (7) for the state vector $\mathbf{x}(t_k)$.

3.2. State-Based Deep MPC

Learning-based predictive control leverages the representation capabilities of machine learning models to achieve a precise approximation of the feedback law used in standard predictive controllers. This approach offers an alternative to the concept of “explicit” MPC, with the goal of further reducing the online computational burden of the control scheme. Explicit MPC relies on a piece-wise function to represent the control law, which requires finding the polytopic region where the states are located online. This process can be computationally expensive, especially if the number of regions that define the corresponding piece-wise function is high. In contrast, learning-based methodologies solve this issue by directly mapping the current states to the optimal action using a machine learning model.

The notion of learning-based MPC was initially introduced by [47]. However, its practical application in designing fast MPC techniques has only become reliable recently, primarily due to new advancements in the theoretical description of neural networks. In the following part of this section, the deep MPC formulation used in this study is presented.

A deep learning model $\mathcal{N} = \mathcal{N}(\mathbf{x}, \mathbf{r}, \theta)$ with n_x inputs and n_u outputs can be described in this fashion, where θ stands for the parameters of the data-driven model, whereas \mathbf{r} represents the vector of the MPC references, i.e., $\mathbf{r} = [\mathbf{x}_{\text{ref}}^T, \mathbf{y}_{\text{ref}}^T, \mathbf{u}_{\text{ref}}^T]^T$. Synthetic data for training are produced by solving Equation (7) for n_{tr} different reference samples and

different states, i.e., \mathbf{r}_i and $\mathbf{x}_{tr,i}$, respectively, for $i = 1, 2, \dots, n_{tr}$, and by saving the tuples $(\mathbf{x}_{tr,i}, \mathbf{r}_i, \mathbf{u}_{mpc}(\mathbf{x}_{tr,i}))$ in the dataset \mathcal{B}_{tr} , where $\mathbf{u}_{mpc}(\cdot)$ represents the MPC control action.

The model denoted by \mathcal{N} is subjected to offline training using the dataset \mathcal{B}_{tr} via the backpropagation method to resolve the following optimization problem:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{n_{tr}} \sum_{i=0}^{n_{tr}} \|\mathbf{u}_{mpc}(\mathbf{x}_{tr,i}) - \mathcal{N}(\mathbf{x}_{tr,i}, \mathbf{r}_i, \theta)\|_2^2 \quad (11)$$

where the loss function is the mean squared error between the model prediction and the target. Finally, the actual control action is obtained as follows:

$$\mathbf{u}_{s-dmpc}(\mathbf{x}(t_k)) = \mathcal{N}(\mathbf{x}(t_k), \mathbf{r}, \theta^*) \quad (12)$$

where the subscript “s-dmpc” indicates the state-based deep MPC algorithm.

3.3. Output-Based Deep MPC

As previously stated, the reliability in a practical scenario of the algorithm proposed in Section 3.2 is limited by the assumption of full-state measurability. Within this context, it is worth noticing that the design and tuning of a suitable state observer is usually considered a time-consuming process that still relies on the strong assumption of knowledge of the model parameters. In a realistic situation, the latter can only be estimated with a certain degree of accuracy, which may significantly affect the performance of the state observer. Moreover, inaccuracies in the parameters may lead to wrong predictions of the system’s state evolution, thus making pointless the predictive nature of the controller. As a possible solution, a novel methodology is proposed here, which relies only on the available measurements to approximate the optimal control action. In particular, inspired by the techniques used in the context of the partially observable Markov decision process, a fixed window of historical measurements is used as input for the neural network during the training phase, thus allowing the model to learn a map from the available measurements to the optimal action.

Therefore, the training phase is firstly reformulated by relying on a dataset \mathcal{B}_{tr}^{out} , with $|\mathcal{B}_{tr}^{out}| = n_{tr} - n_w$, in which the i -th sample consists of a feature vector \mathbf{f}_i and the target $\mathbf{u}_{mpc}(\mathbf{x}_{tr,i})$, with the former defined as follows:

$$\mathbf{f}_i = [\mathbf{y}_{tr,i-n_w}, \dots, \mathbf{y}_{tr,i-1}, \mathbf{u}_{tr,i-n_w}, \dots, \mathbf{u}_{tr,i-1}, \mathbf{r}_i] \quad (13)$$

where $\mathbf{y}_{tr,i}$ and $\mathbf{u}_{tr,i}$ are the samples of the system’s outputs and inputs, respectively, while n_w is the length of the window of the considered historical measurements. The neural network \mathcal{N} is then trained offline on the dataset \mathcal{B}_{tr}^{out} by solving the following optimization through the backpropagation method:

$$\theta_{out}^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{n_{tr} - n_w} \sum_{i=n_w}^{n_{tr}} \|\mathbf{u}_{mpc}(\mathbf{x}_{tr,i}) - \mathcal{N}(\mathbf{f}_i, \theta)\|_2^2 \quad (14)$$

where the loss function is the mean squared error between the network prediction and the target. The control action is obtained from the following equation:

$$\mathbf{u}_{o-dmpc}(\mathbf{x}(t_k)) = \mathcal{N}(\mathbf{f}_i, \theta_{out}^*) \quad (15)$$

where the subscript “o-dmpc” indicates the output-based deep MPC algorithm.

The proposed output-based deep MPC relies on the idea that a sufficiently large window of historical measurements contains the information necessary to reconstruct the state of the system, and that the neural network is able to approximate the optimal action to take in a particular state by recognizing its pattern in the available measurements. However, the possibility of using the presented technique in a realistic scenario strictly depends on how the training set \mathcal{B}_{tr}^{out} is generated. For instance, if the model’s parameters are kept constant through the whole generation procedure, the algorithm is expected to achieve high

performance only if the parameters of the system coincide with the ones used to generate the training set. However, as mentioned above, the assumption of accurate knowledge of the model's parameters has limited practical support. For this reason, an adaptation of the output-based deep MPC is presented here with the aim of making the control algorithm robust to variations in the parameters of the system. Specifically, such a result is achieved by generating each sample i of the training dataset \mathcal{B}_{tr}^{out} by integrating the system dynamics in (5) and solving the optimization problem (7) with a different value of the parameter vector \mathbf{p}_i . As a consequence, the training of the neural network becomes

$$\theta_{out}^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{n_{tr} - n_w} \sum_{i=n_w}^{n_{tr}} \|\mathbf{u}_{mpc}(\mathbf{x}_{tr,i}, \mathbf{p}_i) - \mathcal{N}(\mathbf{f}_i, \theta)\|_2^2 \quad (16)$$

which is based on the idea that a deep learning model can learn the optimal action to take in a particular state and parameter configuration by searching for a specific pattern in the available historical measurements.

4. Training of the Controller

This section is devoted to the definition of the optimal charging problem for a Lithium-Ion cell in Section 4.1, as well as the dataset generation and model training phases in Section 4.2 and Section 4.3, respectively.

4.1. Optimal Battery Charging

Within this section, a problem of optimal control is presented concerning the fast charging of a battery. The main objectives are to track the state of charge while reducing the current flow, with the additional constraint of maintaining safe voltage and temperature levels. The battery dynamics are modeled as a nonlinear and discrete system where the current applied to the battery acts as the input variable. Finally, the optimization problem is formulated according to the following procedure:

$$\min_{I_{[t_k, t_{k+H}]}} q_{soc} \sum_{i=k+1}^{k+H} (\operatorname{soc}(t_k) - \operatorname{soc}_{ref})^2 + r \sum_{i=k}^{k+H-1} I(t_k)^2 + q_H (\operatorname{soc}(t_{k+H}) - \operatorname{soc}_{ref})^2 \quad (17)$$

that for $i = k, k + 1, \dots, k + H - 1$ are subject to:

$$\text{battery dynamics in (1)–(4)} \quad (18a)$$

$$0 \leq I(t_i) \leq 10 \text{ A} \quad (18b)$$

$$0 \leq \operatorname{soc}(t_i) \leq 1 \quad (18c)$$

$$T_c(t_i) \leq 313.15 \text{ K} \quad (18d)$$

$$T_s(t_i) \leq 313.15 \text{ K} \quad (18e)$$

$$V(t_i) \leq 4.2 \text{ V} \quad (18f)$$

with $q_{soc} = 1$, $r = 10^{-6}$ and $q_H = 1$. Moreover, the length of the prediction horizon is chosen as $H = 4$ and the sample time is taken as $t_s = 10 \text{ s}$. Interestingly enough, the reference state of charge (soc_{ref}) is left unspecified due to its dependence on charging preferences. This is in contrast to the current, temperature, and voltage limits, which are determined solely by the battery chemistry. During the dataset generation phase for the learning-based algorithm, the reference state of charge is modeled as a random variable with a uniform probability distribution within a designated range. This enables the learning-based MPC to adapt the charging profile to the various potential values of the reference. Finally, the input reference, which is the current reference, is always set to $I_{ref} = 0$ since the system is marginally stable.

4.2. Dataset Generation

In order to train the learning-based algorithm effectively, it is necessary to have a large dataset that includes information on the system's states and parameters, as well as the optimal action taken by the predictive controller. The quality of the data is critical, as it can significantly affect the performance of the proposed data-driven algorithm. Having inaccurate, inconsistent, or flawed data can have a negative impact on the performance of a data-driven algorithm, such as the one presented in this manuscript, potentially leading to incorrect or biased results, poor performance, and decreased trust in the algorithm and its outcomes. Furthermore, the quantity of data used is also crucial in avoiding overfitting in a machine learning model. Overfitting takes place whenever a model ends up being too complex and includes too many parameters with respect to the amount of training data; this phenomenon makes it so that the model memorizes the training data, with the consequence of displaying a poor performance when applied to new, previously unseen data. Overfitting can be alleviated by increasing the amount of training data, so that the model is provided with more examples to learn from and, thus, may be able to generalize to new data. Even so, a mere increase in the quantity of training data may not be enough to prevent the overfitting phenomenon since the characteristics of the training data used need to be considered. In fact, it is important for those data to be as varied and representative as possible with respect to their real-world distribution, in order for the model not to memorize specific patterns or features in the data. In this regard, robust models for machine learning, with a high potential for generalization when dealing with unseen data, need to be fed both high-quality and large quantities of data.

As a result, firstly, an ideal MPC controller that executes the control task specified in Section 4.1 is implemented. To produce the dataset, a synthetic method is utilized, involving the offline execution of 2000 simulations, each containing 200 time steps. For each sample time i , a dataset sample is created, including the available measurements, the reference for the state of charge (determined for the j -th simulation), and the optimal current value for the subsequent time step:

$$(V_{i-1}, T_{s,i-1}, I_{i-1}, \text{soc}_{\text{ref},j}, I_i^*). \quad (19)$$

In order to include a window of historical measurements of size n_w , the database is reshaped with the rows as follows:

$$(V_{i-n_w}, T_{s,i-n_w}, I_{i-n_w}, \dots, V_{i-1}, T_{s,i-1}, I_{i-1}, \text{soc}_{\text{ref},j}, I_i^*) \quad (20)$$

i.e., the previous measurements are used to predict the optimal current. Note that for each simulation j , the battery's initial conditions are extracted from a random uniform distribution: $\text{soc}(t_0) \sim \mathcal{U}(0, 1)$ and $T_s(t_0), T_c(t_0) \sim \mathcal{U}(298.15 \text{ K}, 313.15 \text{ K})$. Similarly, the reference state of charge for the j -th simulation is sampled as $\text{soc}_{\text{ref},j} \sim \mathcal{U}(0.7, 1)$. Finally, in order to make the algorithm robust to changes in the parameters during the battery aging, the battery capacity and the SEI resistance are sampled from a uniform distribution, thus obtaining a different parameter vector \mathbf{p}_j in every simulation: $C_j \sim \mathcal{U}(5.5 \text{ Ah}, 8 \text{ Ah})$ and $R_{\text{sei},j} \sim \mathcal{U}(0.014 \Omega, 0.019 \Omega)$. Note that the pair $C = 8 \text{ Ah}$ and $R_{\text{sei}} = 0.014 \Omega$ corresponds to that of a brand new battery just released from the factory, while $C = 5.5 \text{ Ah}$ and $R_{\text{sei}} = 0.019 \Omega$ represents a battery at the end of its life.

To reiterate, it is crucial to remember that for each step i , the optimal current I_i^* is determined by solving the optimization problem described in (17). This process involves executing an ideal MPC, assuming full-state measurability and complete knowledge of the relevant parameters. To increase exploration, Gaussian noise is added to the MPC control action during the battery dynamics evolution for every simulation, using a standard deviation of $2, A$. Lastly, the dataset is split into three separate sets, as is customary in machine learning: training, validation, and testing.

4.3. Training Phase and Model Selection

In this section, the training phase of several machine learning configurations is conducted, with a particular focus on feed-forward deep neural networks (DNNs) and recurrent neural networks (RNNs). As can be seen from Table 1, which reports the results of the training phase for the different architectures and measurement window n_w , the use of RNNs appears to be particularly suitable for solving the considered task, thanks to their ability in dealing with sequential information, such as the one contained in the dataset samples (see (20)). Note that all of the considered models have a *ReLU* activation function in the hidden layers and a *tanh* activation function in the output layer. The *tanh* activation function on the last layer is used to constrain the output of the network within a specific range, which means, in the context of battery charging, constraining the optimally applied current within its range of operation. In order to enhance the learning capabilities of all of the considered models, a preprocessing pipeline is considered that involves both the scaling and standardization of the dataset's features. Moreover, for the training of the deep learning models, the stochastic gradient descent algorithm is employed, and, in particular, the *Adam* optimizer is adopted, with the mean squared error as the loss function and learning rate equal to 5×10^{-4} . Finally, it is worth highlighting that Gaussian noise was applied to the features in the training set with the dual motivation of preventing any overfitting and making the model robust to a realistic scenario, in which the values of surface temperature and voltage were affected by a measurement disturbance due to malfunctioning or inaccurate sensors. Specifically, Gaussian noises with standard deviations of 20 mV for the voltage and 1 K for the temperature were considered.

Table 1. Mean square error (MSE) over the validation set of the machine learning models considered as possible candidates for the MPC law approximation under the assumption of non-measurable states and unknown model parameters.

Model	Measurements Window (n_w)	Mean Squared Error
DNN (Dense: 3×100 , 3×50 , 3×10)	10	0.256
DNN (Dense: 3×100 , 3×50 , 3×10)	15	0.212
DNN (Dense: 3×100 , 3×50 , 3×10)	20	0.199
RNN (LSTM: 64, 32, 16, Dense: 3×100 , 3×50 , 3×10)	10	0.202
RNN (LSTM: 64, 32, 16, Dense: 3×100 , 3×50 , 3×10)	15	0.182
RNN (LSTM: 64, 32, 16, Dense: 3×100 , 3×50 , 3×10)	20	0.124
RNN (LSTM: 128, 64, 32, 16, Dense: 2×100 , 50, 10)	10	0.193
RNN (LSTM: 128, 64, 32, 16, Dense: 2×100 , 50, 10)	15	0.133
RNN (LSTM: 128, 64, 32, 16, Dense: 2×100, 50, 10)	20	0.109

The model that achieved the highest performance consisted of a recurrent neural network with 4 long short-term memory (LSTM) hidden layers (with 128, 64, 32, and 16 neurons each) and 4 fully connected hidden layers (2 of them with 100 neurons each, one with 50 neurons, and the last one with 10 neurons), with a window of historical measurements $n_w = 20$. Therefore, such a model was selected to properly approximate the predictive control law in the simulations considered in Section 5.

It is important to note that the loss of validation for the selected model converged after 23 epochs, as depicted in Figure 1, and that no overfitting was present due to the fact that the prediction error during the testing phase ($e_{\text{test}} = 0.109$) was coherent with that achieved on the training set.

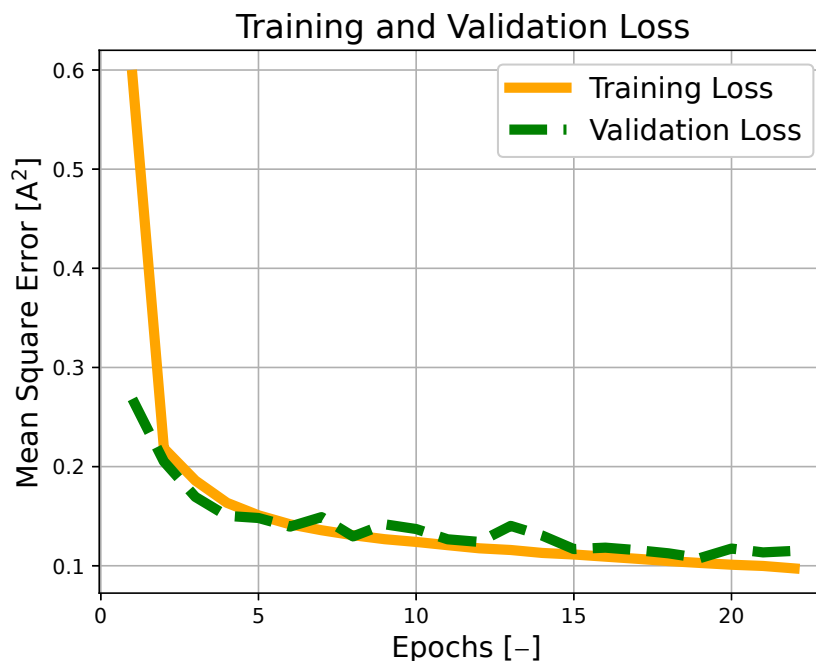


Figure 1. Training and validation loss profiles against epochs for the RNN model with 4 long short-term memory (LSTM) hidden layers (with 128, 64, 32, and 16 neurons each) and 4 fully connected hidden layers (2 of them with 100 neurons each, one with 50 neurons, and the last one with 10 neurons), with a window of historical measurements $n_w = 20$.

5. Results

This section illustrates the results of the comparison, regarding the case of battery charging, between an ideal predictive controller, which operates under the assumptions of state measurability and parameter knowledge, and an output-based deep MPC, for which only the model structure is assumed to be known a priori. More specifically, the effectiveness of the proposed algorithm in tracking the optimal charging profile for a battery with uncertain parameters by exploiting only noisy measurements is demonstrated in Section 5.1, for the first time, to the knowledge of the authors. Finally, in Section 5.2, details on the software implementation are provided.

5.1. Approximation of the Optimal Charging Profile in the Presence of Unknown States and Parameters

In the following subsection, the effectiveness of the neural network-based approach in approximating the charging profile achieved by the ideal predictive controller will be evaluated, with particular attention to different combinations of initial conditions, references, and battery parameters. The results are illustrated in Figures 2–6, with crossed lines representing the output-based deep MPC profiles and solid lines representing the ideal predictive controller. Specifically, the figures on the left refer to an almost new battery with parameters $C = 7.5$ Ah and $R_{sei} = 0.015$ Ω , while the ones on the right represent an aged battery with $C = 6$ Ah and $R_{sei} = 0.018$ Ω . Moreover, two scenarios of initial conditions and state-of-charge references are considered: blue lines are used for the profiles, which refer to the simulation with $\text{soc}_{\text{ref}} = 0.8$ and initial states given by $\text{soc}(t_0) = 0.2$, $q_n(t_0) = 0$, $q_p(t_0) = 0$ and $T_s(t_0) = T_c(t_0) = 305.15$ K (also referred as *sim 1*), while the red lines are used for the case of $\text{soc}_{\text{ref}} = 1$ and initial states given by $\text{soc}(t_0) = 0.05$, $q_n(t_0) = 0$, $q_p(t_0) = 0$, and $T_s(t_0) = T_c(t_0) = 300.15$ K (*sim 2*). For all of the simulations, a prediction horizon $H = 4$ is considered, equal to the one used for the generation of the training dataset.

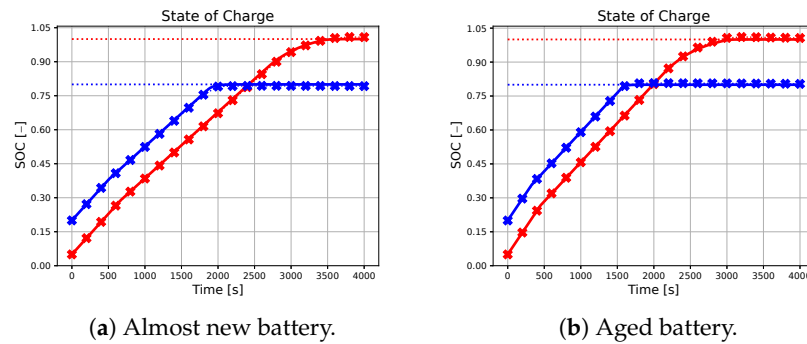


Figure 2. State-of-charge profiles of the two considered methodologies (solid line for the ideal predictive controller and crossed line for the output-based deep MPC), with *sim 1* in blue and *sim 2* in red. The blue and red dotted lines represent the SOC references for the first and second simulations, respectively. The case of a new battery is considered on the left, while an aged battery is considered on the right.

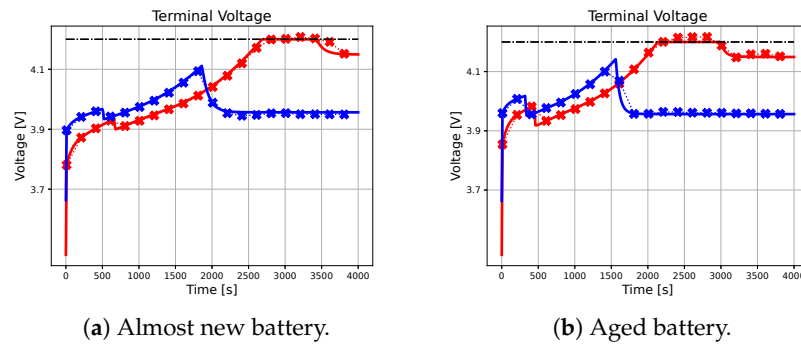


Figure 3. The figure illustrates the profiles of the terminal voltage achieved by the benchmark and the proposed methodology for different initial conditions and battery parameters. The dash-dotted black line represents the upper-bound voltage.

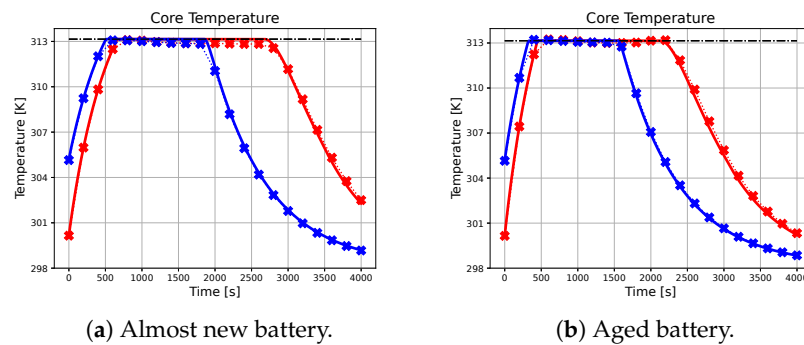


Figure 4. The figure illustrates the profiles of the core temperature achieved by the benchmark and the proposed methodology for different initial conditions and battery parameters. The dash-dotted black line represents the upper-bound for the core temperature.

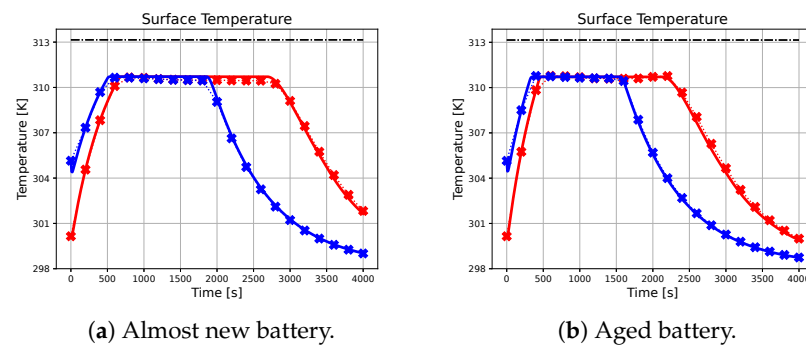


Figure 5. The figure illustrates the profiles of the surface temperature achieved by the benchmark and the proposed methodology for different initial conditions and battery parameters. The dash-dotted black line represents the upper-bound for the surface temperature.

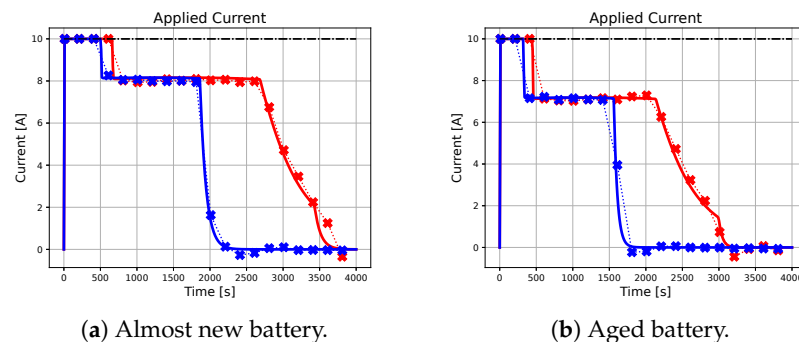


Figure 6. The figure illustrates the profiles of the current applied by the benchmark and the proposed methodology for different initial conditions and battery parameters. The dash-dotted black line represents the upper-bound for the applied current.

It is important to recall that in every simulation the output-based deep MPC does not have access to the battery states and is not aware of the value of the battery parameters, but it relies only on the available measurements, which are assumed to be affected by a Gaussian noise with a standard deviation of 20 mV for the voltage and 1 K for the surface temperature.

As seen from the results, the trajectory of the system controlled by the proposed algorithm exhibits characteristics that are very similar to those achieved by using the ideal MPC. It is important to note that the ideal MPC is based on strong assumptions, such as full-state measurability and known parameters, which may not hold in a realistic scenario. Therefore, it is considered as a benchmark in this study. In particular, Figure 2 demonstrates that both controllers are capable of tracking the state of charge references in all considered situations. Moreover, Figures 3–5 show that the output-based deep MPC is almost always able to satisfy the constraints on voltage and temperature. Furthermore, in Figure 6, the applied current is shown, where the constraints are satisfied by the design of the output-based deep MPC, while in the case of the ideal predictive controller, they are imposed as bounds on the input. The control strategy obtained with both the controllers for *sim 2* is similar to the constant-current/constant-temperature/constant-voltage protocol, which has been proven to be effective in decreasing the charging time while adhering to the current, voltage, and temperature restrictions (see [48,49]).

It should be noted that in realistic scenarios, such as the one considered for the proposed algorithm, slight violations of voltage and temperature constraints may occur due to unknown internal states, unknown parameters, and measurement noises affecting the available output. In practical situations, it is not always feasible to ensure strict satisfaction of hard constraints on states and outputs. Instead, these constraints should be considered

soft constraints, meaning they should be avoided if possible. Appropriate measures can be taken to minimize the likelihood of a constraint violation. For instance, using a more accurate machine learning model in output-based deep MPC may help reduce constraint violations. As for the satisfaction of input constraints, the inclusion of a *tanh* activation function on the output layer of the neural network model ensures that the applied current does not exceed the safety limits.

It should be noted that a similar approximation capability for the output-based deep MPC can be achieved for initial conditions, references, and parameters that fall in the intervals of variation considered during the training phase. These intervals can be set wide enough to account for the entire range of realistic scenarios. To support this claim, the paper provides a statistical analysis comparing the output-based deep MPC profiles with a benchmark over 100 simulations with randomly chosen starting states, references, and parameters. In particular, it emerges that the approximation error for the applied current is, on average, -3.9 mA, with a standard deviation of 487.9 mA, as depicted in a histogram that resembles a normal distribution in Figure 7.

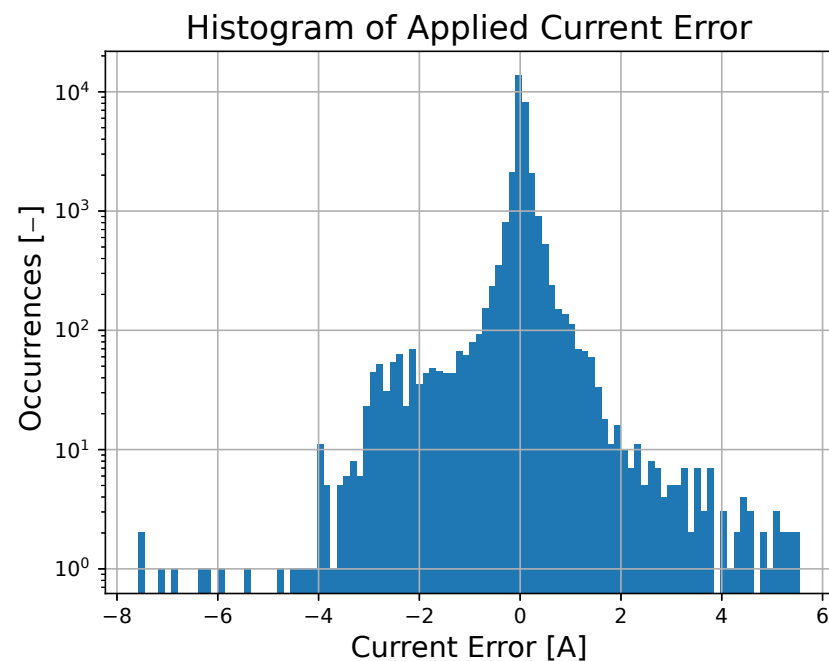


Figure 7. Histogram of the errors observed in over 100 simulations by the output-based deep MPC approach in approximating the ideal predictive control law.

Table 2 summarizes the approximation errors and standard deviations for the state of charge, voltage, and core temperature profiles. The average approximation error for the state of charge profile is -0.2×10^{-3} , and the standard deviation is 10.1×10^{-3} . The voltage profile has an average error of -0.2 mV, with a standard deviation of 14.4 mV, while the core temperature has an average error of -24.4 mK, with a standard deviation of 401.7 mK. The algorithm has a higher probability of making errors in the approximation of the MPC control law when the system state is near the reference or constraints. Although these errors can cause small oscillations around the reference, they do not affect the stability of the control law. However, including additional samples corresponding to critical points in the dataset could improve the accuracy of the algorithm.

Finally, we note that, although the output-based dMPC does not imply an initial guess for the battery's states or the battery parameters, a guess for the first n_w control action is required, due to the fact that the proposed approach can predict the optimal current only when a window of n_w historical measurements is available. Such an applied current guess is computed offline as the average current applied at the beginning of the charging phase according to the available data in the training set.

Table 2. Statistical description of over 100 simulations of the difference between the trajectory of the model variables obtained with the 2 considered control methodologies.

Statistics	soc [10^{-3}]	V [mV]	T_c [mK]
Mean	−0.2	−0.1	−24.4
Standard deviation	10.1	14.4	401.7

5.2. Implementation Details

The simulations presented in this paper were conducted on a personal computer running Windows 10 with an i7-8750H processor and 16 GB of RAM, and the implementation was carried out using Python 3.7. The deep learning model was created and trained using TensorFlow 2.0, while the equations of the model were integrated via CasADi, a symbolic framework that allows for automatic differentiation. CasADi was also employed to solve the optimal control problem described in Equation (7).

6. Discussion

As stated, the methodology described in this paper is a novel contribution. However, corresponding issues need to be taken into account. Specifically, the aspects mentioned in the following paragraphs may be considered in future research.

- The proposed methodology is designed to be robust with respect to uncertainty in the battery parameters. However, it should be highlighted that a high performance can be achieved in a practical scenario only under the assumption that all of the relevant physical phenomena have been considered in the battery model used to generate the synthetic dataset. Although this assumption is more realistic than assuming perfect knowledge of states and parameters, it is important to investigate the consequences of its violation and develop countermeasures to mitigate the effects of potential inaccuracies in the modeling phase.
- The output-based deep MPC predicts the optimal input for charging a battery based on previous measurements. However, at the start of the charging process, there are only a limited number of previous measurements available, making it challenging to effectively use the proposed algorithm. To address this challenge, a rough guess for the initial control actions is utilized. This guess is computed offline as the average current applied at the beginning of the charging phase, based on the available data in our training set. Unfortunately, if the guess input deviates from the optimal one, the controller's performance during the first stage of charging may be affected, potentially leading to safety issues. To mitigate this issue, multiple output-based deep MPC algorithms can be employed during the first stage of charging. Each algorithm utilizes a different number of previous measurements, starting from no previous measurements, then gradually incorporating one, two, and so on, until a certain number of measurements are available.

Addressing the aforementioned issues in the future may pave the way to further improve the proposed approach.

7. Conclusions

In this paper, the optimal charging of a battery is addressed by exploiting a deep learning-based methodology. This methodology has been designed to be effective even in a realistic scenario in which the battery parameters are unknown and only noisy measurements are available. To the best of the authors' knowledge, for the first time, the proposed technique relies on recurrent neural networks to approximate the optimal control law of a predictive controller, by using only voltage and temperature data. Firstly, battery charging was formalized as an optimal control problem. Then, a detailed description of the generation of a synthetic dataset for the training phase was provided, highlighting the importance of exploring different battery conditions and parameters. In addition, an analysis was conducted to select the most suitable machine learning model, in order to find a reasonable

trade-off between accuracy and computational complexity. Finally, the validation of the proposed algorithm was performed against an ideal predictive controller (with full-state measurability and parameter knowledge) used as a benchmark. The results show that the output-based deep MPC is able to achieve a performance similar to the benchmark without relying on the unrealistic assumption of knowing the battery states and parameters, but only on noisy measurements. In this regard, a statistical analysis was provided to highlight the effectiveness of the proposed approach over 100 different simulations.

Author Contributions: Conceptualization, A.P.; data curation, A.P. and D.T.; investigation, A.P. and D.T.; methodology, A.P.; software, A.P.; validation, A.P. and D.T.; writing—original draft, A.P., E.B. and D.T.; writing—review and editing, A.P., E.B. and D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, L.; Han, X.; Li, J.; Hua, J.; Ouyang, M. A review on the key issues for Lithium-Ion battery management in electric vehicles. *J. Power Sour.* **2013**, *226*, 272–288. [[CrossRef](#)]
2. Shen, W.; Vo, T.T.; Kapoor, A. Charging algorithms of Lithium-Ion batteries: An overview. In Proceedings of the 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), Singapore, 18–20 July 2012; pp. 1567–1572.
3. Chaturvedi, N.A.; Klein, R.; Christensen, J.; Ahmed, J.; Kojic, A. Algorithms for advanced battery-management systems. *IEEE Control Syst.* **2010**, *30*, 49–68.
4. Hsieh, G.C.; Chen, L.R.; Huang, K.S. Fuzzy-controlled Li-ion battery charge system with active state-of-charge controller. *IEEE Trans. Ind. Electron.* **2001**, *48*, 585–593. [[CrossRef](#)]
5. Wang, S.C.; Liu, Y.H. A PSO-based fuzzy-controlled searching for the optimal charge pattern of Li-ion batteries. *IEEE Trans. Ind. Electron.* **2014**, *62*, 2983–2993. [[CrossRef](#)]
6. Purushothaman, B.; Landau, U. Rapid charging of Lithium-Ion batteries using pulsed currents a theoretical analysis. *J. Electrochem. Soc.* **2006**, *153*, A533–A542. [[CrossRef](#)]
7. Chen, L.R.; Hsu, R.C.; Liu, C.S. A design of a grey-predicted Li-ion battery charge system. *IEEE Trans. Ind. Electron.* **2008**, *55*, 3692–3701. [[CrossRef](#)]
8. Perez, H.E.; Hu, X.; Dey, S.; Moura, S.J. Optimal charging of Li-ion batteries with coupled electro-thermal-aging dynamics. *IEEE Trans. Veh. Technol.* **2017**, *66*, 7761–7770. [[CrossRef](#)]
9. Romagnoli, R.; Couto, L.D.; Kinnaert, M.; Garone, E. Control of the state-of-charge of a Li-ion battery cell via reference governor. *IFAC-PapersOnLine* **2017**, *50*, 13747–13753. [[CrossRef](#)]
10. Camacho, E.F.; Alba, C.B. *Model Predictive Control*; Springer Science & Business Media: Berlin, Germany, 2013.
11. Klein, R.; Chaturvedi, N.A.; Christensen, J.; Ahmed, J.; Findeisen, R.; Kojic, A. Optimal charging strategies in Lithium-Ion battery. In Proceedings of the 2015 American Control Conference (ACC), San Francisco, CA, USA, 29 June–1 July 2011; pp. 382–387.
12. Zou, C.; Manzie, C.; Nešić, D. Model predictive control for Lithium-Ion battery optimal charging. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 947–957. [[CrossRef](#)]
13. Torchio, M.; Wolff, N.A.; Raimondo, D.M.; Magni, L.; Krewer, U.; Gopaluni, R.B.; Paulson, J.A.; Braatz, R.D. Real-time model predictive control for the optimal charging of a Lithium-Ion battery. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 4536–4541.
14. Torchio, M.; Magni, L.; Braatz, R.D.; Raimondo, D. Design of piecewise affine and linear time-varying model predictive control strategies for advanced battery management systems. *J. Electrochem. Soc.* **2017**, *164*, A949. [[CrossRef](#)]
15. Lucia, S.; Torchio, M.; Raimondo, D.M.; Klein, R.; Braatz, R.D.; Findeisen, R. Towards adaptive health-aware charging of Li-ion batteries: A real-time predictive control approach using first-principles models. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 4717–4722.
16. Pozzi, A.; Torchio, M.; Raimondo, D.M. Assessing the performance of model-based energy saving charging strategies in Li-ion cells. In Proceedings of the 2018 IEEE Conference on Control Technology and Applications (CCTA), Copenhagen, Denmark, 21–24 August 2018; pp. 806–811.
17. Mrugalska, B.; Stetter, R. Health-aware model-predictive control of a cooperative AGV-based production system. *Sensors* **2019**, *19*, 532. [[CrossRef](#)]

18. Kakouche, K.; Rekioua, T.; Mezani, S.; Oubelaid, A.; Rekioua, D.; Blazek, V.; Prokop, L.; Misak, S.; Bajaj, M.; Ghoneim, S.S. Model predictive direct torque control and fuzzy logic energy management for multi power source electric vehicles. *Sensors* **2022**, *22*, 5669. [[CrossRef](#)]
19. Tian, N.; Fang, H.; Wang, Y. Real-Time Optimal Charging for Lithium-Ion Batteries via Explicit Model Predictive Control. In Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), Vancouver, BC, Canada, 12–14 June 2019; pp. 2001–2006.
20. Alessio, A.; Bemporad, A. A survey on explicit model predictive control. In *Nonlinear Model Predictive Control*; Springer: New York, NY, USA, 2009; pp. 345–369.
21. Karg, B.; Lucia, S. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Trans. Cybern.* **2020**, *50*, 3866–3878. [[CrossRef](#)]
22. Parisini, T.; Zoppoli, R. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica* **1995**, *31*, 1443–1451. [[CrossRef](#)]
23. Bemporad, A.; Oliveri, A.; Poggi, T.; Storace, M. Ultra-fast stabilizing model predictive control via canonical piecewise affine approximations. *IEEE Trans. Autom. Control* **2011**, *56*, 2883–2897. [[CrossRef](#)]
24. Csekő, L.H.; Kvasnica, M.; Lantos, B. Explicit MPC-based RBF neural network controller design with discrete-time actual Kalman filter for semiactive suspension. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 1736–1753. [[CrossRef](#)]
25. Hertneck, M.; Köhler, J.; Trimpe, S.; Allgöwer, F. Learning an approximate model predictive controller with guarantees. *IEEE Control Syst. Lett.* **2018**, *2*, 543–548. [[CrossRef](#)]
26. Pozzi, A.; Moura, S.; Toti, D. A Neural Network-Based Approximation of Model Predictive Control for a Lithium-Ion Battery with Electro-Thermal Dynamics. In Proceedings of the 2022 IEEE 17th International Conference on Control & Automation (ICCA), Naples, Italy, 27–30 June 2022; pp. 160–165.
27. Pozzi, A.; Moura, S.; Toti, D. Deep Learning-Based Predictive Control for the Optimal Charging of a Lithium-Ion Battery with Electrochemical Dynamics. In Proceedings of the 2022 IEEE Conference on Control Technology and Applications (CCTA), Bridgetown, Barbados, 16–18 August 2022; pp. 785–790.
28. Park, S.; Kato, D.; Gima, Z.; Klein, R.; Moura, S. Optimal experimental design for parameterization of an electrochemical Lithium-Ion battery model. *J. Electrochem. Soc.* **2018**, *165*, A1309. [[CrossRef](#)]
29. Waag, W.; Fleischer, C.; Sauer, D.U. Critical review of the methods for monitoring of Lithium-Ion batteries in electric and hybrid vehicles. *J. Power Sources* **2014**, *258*, 321–339. [[CrossRef](#)]
30. Santhanagopalan, S.; Guo, Q.; Ramadass, P.; White, R.E. Review of models for predicting the cycling performance of lithium ion batteries. *J. Power Sources* **2006**, *156*, 620–628. [[CrossRef](#)]
31. He, H.; Xiong, R.; Fan, J. Evaluation of Lithium-Ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies* **2011**, *4*, 582–598. [[CrossRef](#)]
32. Nejad, S.; Gladwin, D.; Stone, D. A systematic review of lumped-parameter equivalent circuit models for real-time estimation of Lithium-Ion battery states. *J. Power Sources* **2016**, *316*, 183–196. [[CrossRef](#)]
33. Gomadam, P.M.; Weidner, J.W.; Dougal, R.A.; White, R.E. Mathematical modeling of Lithium-Ion and nickel battery systems. *J. Power Sources* **2002**, *110*, 267–284. [[CrossRef](#)]
34. Ramadesigan, V.; Northrop, P.W.; De, S.; Santhanagopalan, S.; Braatz, R.D.; Subramanian, V.R. Modeling and simulation of Lithium-Ion batteries from a systems engineering perspective. *J. Electrochem. Soc.* **2012**, *159*, R31. [[CrossRef](#)]
35. Moura, S.J. Estimation and control of battery electrochemistry models: A tutorial. In Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 3906–3912.
36. Zou, C.; Manzie, C.; Anwar, S. Control-oriented modeling of a Lithium-Ion battery for fast charging. *IFAC Proc. Vol.* **2014**, *47*, 3912–3917. [[CrossRef](#)]
37. Zou, C.; Manzie, C.; Nešić, D. A framework for simplification of PDE-based Lithium-Ion battery models. *IEEE Trans. Control. Syst. Technol.* **2015**, *24*, 1594–1609. [[CrossRef](#)]
38. Doyle, M.; Fuller, T.F.; Newman, J. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *J. Electrochem. Soc.* **1993**, *140*, 1526–1533. [[CrossRef](#)]
39. Pozzi, A.; Xie, X.; Raimondo, D.M.; Schenkendorf, R. Global Sensitivity Methods for Design of Experiments in Lithium-ion Battery Context. *IFAC-PapersOnLine* **2020**, *53*, 7248–7255. [[CrossRef](#)]
40. Pozzi, A.; Toti, D. Lexicographic model predictive control strategy in ageing-aware optimal charging procedure for Lithium-Ion batteries. *Comput. Chem. Eng.* **2022**, *163*, 107847. [[CrossRef](#)]
41. Perez, H.E.; Hu, X.; Moura, S.J. Optimal charging of batteries via a single particle model with electrolyte and thermal dynamics. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 4000–4005.
42. Moura, S.J.; Argomedeo, F.B.; Klein, R.; Mirtabatabaei, A.; Krstic, M. Battery state estimation for a single particle model with electrolyte dynamics. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 453–468. [[CrossRef](#)]
43. Perez, H.; Dey, S.; Hu, X.; Moura, S. Optimal charging of li-ion batteries via a single particle model with electrolyte and thermal dynamics. *J. Electrochem. Soc.* **2017**, *164*, A1679. [[CrossRef](#)]
44. Ecker, M.; Tran, T.K.D.; Dechent, P.; Käbitz, S.; Warnecke, A.; Sauer, D.U. Parameterization of a physico-chemical model of a Lithium-Ion battery I. Determination of parameters. *J. Electrochem. Soc.* **2015**, *162*, A1836–A1848. [[CrossRef](#)]

45. Ecker, M.; Käbitz, S.; Laresgoiti, I.; Sauer, D.U. Parameterization of a physico-chemical model of a Lithium-Ion battery II. Model validation. *J. Electrochem. Soc.* **2015**, *162*, A1849–A1857. [[CrossRef](#)]
46. Bemporad, A.; Morari, M. Robust model predictive control: A survey. In *Robustness in Identification and Control*; Springer: New York, NY, USA, 1999; pp. 207–226.
47. Åkesson, B.M.; Toivonen, H.T. A neural network model predictive controller. *J. Process Control* **2006**, *16*, 937–946. [[CrossRef](#)]
48. Patnaik, L.; Praneeth, A.; Williamson, S.S. A closed-loop constant-temperature constant-voltage charging technique to reduce charge time of Lithium-Ion batteries. *IEEE Trans. Ind. Electron.* **2018**, *66*, 1059–1067. [[CrossRef](#)]
49. Park, S.; Pozzi, A.; Whitmeyer, M.; Perez, H.; Kandel, A.; Kim, G.; Choi, Y.; Joe, W.T.; Raimondo, D.M.; Moura, S. A Deep Reinforcement Learning Framework for Fast Charging of Li-ion Batteries. *IEEE Trans. Transp. Electrification* **2022**, *8*, 2770–2784. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.