# A Split-Future MPC Algorithm for Lithium-Ion Battery Cell-Level Fast-Charge Control

**Marcelo A. Xavier** * **Aloisio Kawakita de Souza** **
**M. Scott Trimboli** **

* *Research & Advanced Engineering, Ford Motor Company, Dearborn, MI 48124 USA (e-mail: maraujox@ford.com).*
** *Department of Electrical and Computer Engineering, University of Colorado Colorado Springs, Colorado Springs, CO 80918, USA)*
*(e-mail: {akawakit, mtrimbol}@uccs.edu).*

**Abstract:** Constrained predictive control has emerged as a viable candidate for generating optimal real-time charging strategies for lithium ion batteries. Able to conform to hard constraints on problem variables, model predictive control (MPC) formulates an optimal 2-norm solution at each time step and can thus assure safe and reliable fast-charge operation. Standard MPC implementations make certain simplifying assumptions regarding future control actions beyond a specified control horizon. This paper demonstrates the potential gains that can be realized for the battery charge problem by relaxing these assumptions.

*Keywords:* Control of renewable energy resources, model predictive control, lithium-ion battery

## 1. INTRODUCTION

A growing desire for ever-shorter charging times presents an important challenge and remains a key obstacle to EV market penetration, Chan and Wong (2004). The ability to bring a battery to a specified state-of-charge in the shortest possible time is ultimately limited by internal electrochemical processes. Indeed, exceeding certain current rates and cell voltages can cause irreversible damage and capacity loss; hence, ensuring operation within a carefully bounded operating window is critically important for battery health and safety.

The well-known constant-current constant-voltage charge profile (CCCV), although easy to apply, cannot take full advantage of the true operating range of the battery, Klein et al. (2011). Alternative charging strategies can be found in the literature with the most promising motivated by optimal control theory, e.g., Remmlinger et al. (2014), Suthar et al. (2015), and Abdollahi et al. (2016). In general, these approaches are limited by either infeasibility for real-time applications, sub-optimality of the computed control policy, or the inability to handle hard constraints.

Model predictive control (MPC) has emerged as an attractive solution to the constrained optimal control problem designed to achieve critical cell-level performance objectives, where respecting certain design limits can be shown to influence both instantaneous and long-term cell performance, as shown by Klein et al. (2011), Xavier and Trimboli (2015), and Zou et al. (2018), for example. Employing a 'look-ahead' approach, MPC can foresee dynamic changes before they happen and efficiently compute stepwise-optimal input control to achieve a quadratic performance objective. More importantly, MPC is able to conform to hard (as well as soft) constraints imposed on designated problem variables, allowing the system to operate in a safe manner close to its constraints boundaries while driving the controlled variables to desired set-points along optimal trajectories.

Standard MPC makes certain simplifying assumptions regarding future control actions beyond a specified control horizon. More specifically, it assumes that all future control inputs within the interval bounded by the control and prediction horizons are held constant. In the context of the receding horizon principle governing MPC algorithms, this is an important factor, as the solution of the entire optimal sequence of future control inputs is heavily influenced by this assumption. Different approaches that relax this assumption have been proposed. For example, in Kouvaritakis et al. (1998) and Rossiter et al. (1998) the authors proposed a dual-mode MPC in which the first 'mode', called near-future, comprises a standard set of predicted control actions defined as control perturbations, whereas the second mode, referred to as far-future, relies on an infinite-horizon linear quadratic control solution. This dual-mode algorithm brings a stability guarantee and will always give a well-posed and meaningful optimization, irrespective of horizon choices, therefore overcoming a known limitation of finite-horizon MPC. Wang (2004) proposed the design of discrete-time MPC using a set of Laguerre functions where the future control trajectory is expressed using an orthonormal expansion. In this framework, the problem of finding the future control trajectory is converted into one of finding a small set of optimal coefficients for the expansion. As with standard MPC, accuracy and stability are improved as the number of terms increase.

In this paper we propose a different approach for dealing with the far-future control predictions. In order to investigate the hypothesis of achieving a more aggressive input current profile, and therefore reduce the time required to charge the battery from an initial SOC to a desired final value, we propose a modification to the standard MPC algorithm by zeroing the remaining prediction values of the control input. Simulation results are then compared against standard linear MPC which shows that the new framework indeed delivers a shorter time-to-charge. The remainder of this paper is organized as follows: in Section 2 we present the battery model used in the simulations; in Section 3 we review the standard MPC algorithm and derive the modified MPC algorithm; the fast-charge problem is defined in Section 4; simulation results and accompanied discussion are presented in Section 5 followed by conclusions in Section 6.

## 2. BATTERY CELL MODELING

State-of-the-art battery management systems (BMS) rely on model-based methods of state of charge (SOC), state of health (SOH), state of power (SOP) and capacity estimation. The mathematical models employed in battery management algorithms must be simple enough for real-time embedded applications, yet sufficiently accurate to describe lithium ion cell behavior in order to extend battery life, increase performance and guarantee safe operation. Lithium ion battery models may be divided into two broad categories: (i) physics-based, and (ii) empirical. The physics-based approach employs a set of partial differential equations (PDEs) coupled with an algebraic expression that together describe the underlying electrochemical reactions that occur within the cell structure. Such models are not suitable for embedded applications, however, as their solution requires significant computational power.

On the other hand, empirical approaches using equivalent-circuit models (ECM) rely on electrical circuit elements to mimic cell behavior. The ECM method is the most widely applied in xEV applications due to its fundamental simplicity and computational efficiency. ECM model parameters are normally obtained by data fitting of input/output relationships using a physical cell in laboratory experiments. As a proof of concept, we shall use in this paper a first-order RC model (also known as a Thevenin model), Liaw et al. (2004). Figure 1 depicts the Thevenin model and its components.

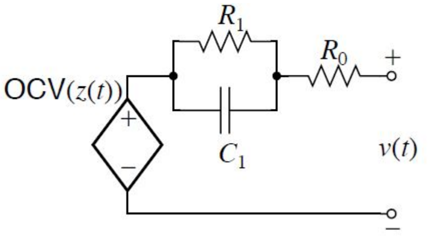

Fig. 1. The Thevenin model.

The open-circuit voltage (OCV) is SOC and temperature dependent and models the difference of equilibrium electrical potential between the positive and negative electrode. The OCV is computed from a lookup table which is generated using lab data. Since BMSs are implemented in digital computers, we present the model equations in discrete time with an intersample period $\triangle t$ in seconds. If we assume $i_k > 0$ for discharge, the SOC is computed by integrating the input current as

$$z_{k+1} = z_k - \frac{\eta \Delta t}{3600 Q} i_k, \qquad (1)$$

where $Q$ represents the cell capacity in ampere-hours, and $\eta$ is an efficiency factor which is assumed to be $\eta = 1$ for discharge and $\eta < 1$ for charge. The resistor-capacitor pair models the diffusion voltage. This phenomenon is observed when a cell is allowed to rest after polarization and the voltage decays gradually instead of returning immediately to OCV. This voltage decay occurs due to slow diffusion processes inside the cell. The diffusion voltage phenomenon can be represented by one or more resistor-capacitor pairs and its equation is given by

$$i_{R,k+1} = \exp\left(-\beta\right) i_{R,k} + \big(1 - \exp\left(-\beta\right)\big)\, i_k, \qquad (2)$$

where $\beta = \frac{\Delta t}{R_1 C_1}$. The output voltage equation the OCV, the diffusion voltage, and the voltage drop (when the cell is under load) as follows

$$v_k = \mathrm{OCV}\left(z_k\right) - R_1 i_{R,k} - R_0 i_k, \qquad (3)$$

where $R_0$ is the cell ohmic resistance. Thevenin model parameters for the present study were obtained for a 25Ah lithium nickel manganese cobalt oxide ($\mathrm{LiNiMnCoO_2}$ or NMC) cell using the identification methodology presented by Plett (2015). The identified parameters are provided in Table 1 for 25 °C. Figure 2 depicts the voltage estimate obtained using experimental data in which the cell was discharged from 95 % to 5 % SOC using the UDDS (Urban Dynamometer Drive Schedule) cycles. Excellent model accuracy is observed.

Table 1. Model identification results.

| Cell Parameters | |
|---|---|
| $Q$ | 24.88 Ah |
| $R_0$ | 0.0011 Ω |
| $R_1$ | 0.282 mΩ |
| $C_1$ | 12.93 kF |

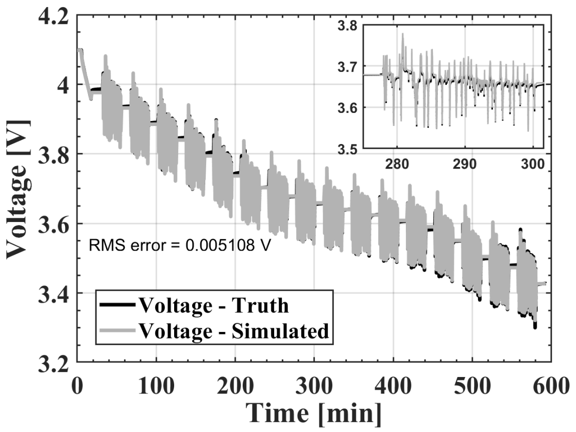

Fig. 2. Output voltage estimation using the identified Thevenin model parameters.

## 3. MODEL PREDICTIVE CONTROL

Predictive control refers to a general "approach" to control design rather than a specific algorithm; MPC is one particular predictive method of this type. More precisely, MPC comprises a class of real-time computer control algorithms that make use of an explicit process model to predict the future responses of a plant.

Optimal control inputs are computed as those that minimize a performance index normally described in terms of a 2-norm quadratic measure. An MPC algorithm continuously updates the predictions and resulting decisions to take account of the most recent measurement data. Loosely speaking, at each control interval MPC attempts to optimize future plant behavior by computing a sequence of future manipulated variable adjustments. The first input in the optimal sequence is used according to the receding horizon principle; the entire calculation is repeated at subsequent sample-time intervals.

### *3.1 Standard MPC With Feedthrough Term*

Standard forms of MPC generically assume a strictly proper system, which implies $\boldsymbol{D} = 0$ in the state-space description. However, mathematical models of battery dynamics must include a non-zero feedthrough term in order to correctly account for ohmic resistance, required to model cell voltage as seen in (3) and subsequently used for constraint handling. For this reason, the standard MPC formulation must be modified to account for a non-zero state-space $D$-term.

We presented in Xavier and Trimboli (2015) a state-space predictive control formulation that accommodates a lithium-ion battery cell equivalent circuit model by incorporating a non-zero direct feedthrough term in the algorithm development. The resulting 'modified' augmented model retains an embedded integrator and like its standard counterpart, admits a cost function in terms of the rate of change of the control input. This 'modified' standard MPC algorithm with feedthrough term is summarized next.

Practical MPC relies on digital implementations and discrete-time representations of dynamic systems, thus we consider a linearized, discrete-time, state-space model of a SISO system:

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}u_k \\ y_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}u_k,\end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$, and $(u, y) \in \mathbb{R}$, are, respectively, the system state, control input, and system output, $n_x$ is the number of system states, and $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are the matrices defining the state-space model. The sub-index $k$ is the time sampling instant.

Re-defining the state vector as $\boldsymbol{\chi}_k = \left[\, \boldsymbol{x}_k^{\mathsf{T}} | u_k^{\mathsf{T}} \,\right]^{\mathsf{T}}$, allows us to define the following augmented state-space

$$\boldsymbol{\chi}_{k+1} = \tilde{\boldsymbol{A}}\boldsymbol{\chi}_k + \tilde{\boldsymbol{B}}\Delta u_{k+1} \tag{4}$$

$$y_k = \tilde{\boldsymbol{C}}\boldsymbol{\chi}_k, \tag{5}$$

where $\Delta u_{k+1} = u_{k+1} - u_k$, and

$$\tilde{\boldsymbol{A}} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{0}_{1\times n_x} & 1 \end{array}\right],\; \tilde{\boldsymbol{B}} = \left[\begin{array}{c} \boldsymbol{0}_{n_x\times 1} \\ \hline 1 \end{array}\right],\; \tilde{\boldsymbol{C}} = \left[\, \boldsymbol{C} | \boldsymbol{D} \,\right].$$

The state and output equations in (4) and (5) can be propagated resulting in the following compact notation for the predicted outputs at a given sampling instant $k_i$:

$$\underrightarrow{\boldsymbol{y}}_{k+1} = \Phi\tilde{A}\boldsymbol{\chi}_{k_i} + G\underrightarrow{\Delta\boldsymbol{u}}_{k+1}. \tag{6}$$

The matrices $\boldsymbol{\Phi} \in \mathbb{R}^{n_p\times(n_x+1)}$ , and $\boldsymbol{G} \in \mathbb{R}^{n_p\times n_c}$, here denoted *MPC gains*, are defined as

$$\boldsymbol{\Phi} = \left[ \left(\tilde{\boldsymbol{C}}\right)^{\mathsf{T}} \left(\tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}\right)^{\mathsf{T}} \left(\tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^2\right)^{\mathsf{T}} \cdots \left(\tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^{n_p-1}\right)^{\mathsf{T}} \right]^{\mathsf{T}},$$

$$\boldsymbol{G} = \begin{bmatrix} \tilde{\boldsymbol{C}}\tilde{\boldsymbol{B}} & 0 & \cdots & 0 \\ \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} & \tilde{\boldsymbol{C}}\tilde{\boldsymbol{B}} & \cdots & 0 \\ & \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^{n_p-1}\tilde{\boldsymbol{B}} & \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^{n_p-2}\tilde{\boldsymbol{B}} & \cdots & \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^{n_p-n_c}\tilde{\boldsymbol{B}} \end{bmatrix},$$

and the vectors containing the predicted outputs and control input rates of change as

$$\underrightarrow{\boldsymbol{y}}_{k+1} = \left[\, y_{k_i+1|k_i} \; y_{k_i+2|k_i} \; \cdots \; y_{k_i+n_p|k_i} \,\right]^{\mathsf{T}}, \tag{7}$$

$$\underrightarrow{\Delta\boldsymbol{u}}_{k+1} = \left[\, \Delta u_{k_i+1|k_i} \; \Delta u_{k_i+2|k_i} \; \cdots \; \Delta u_{k_i+n_c|k_i} \,\right]^{\mathsf{T}}, \tag{8}$$

respectively. The subscripts $n_c$ and $n_p$ denote the control and prediction horizons, respectively. Note that because the system is non-strictly proper this formulation assumes that at the current time sample $k_i$ the current control input $u_{k_i}$ was computed previously (at time sample $k_i - 1$) and uses this information and the current states $\boldsymbol{x}_{k_i}$ to compute the future rate of change of the control $\Delta u_{k_i+1}$ according to the receding horizon control principle.

The control law is determined from the optimization of a quadratic measure of predicted performance,

$$J_{k_i} = \left\| \underrightarrow{\boldsymbol{r}}_{k+1} - \underrightarrow{\boldsymbol{y}}_{k+1} \right\|_Q^2 + \left\| \underrightarrow{\Delta\boldsymbol{u}}_{k+1} \right\|_{\bar{R}}^2, \tag{9}$$

where the first term aims to minimize errors between the predicted output and the reference while the second term penalizes the size of $\underrightarrow{\Delta\boldsymbol{u}}_k$. Note that prediction and control horizons are implicit via equations (7) and (8). $\boldsymbol{Q}$ is a positive semi-definite matrix assumed here to be an appropriately dimensioned identity matrix. The control step penalty $\bar{R} \in \mathbb{R}$ may be used as a tuning parameter. The solution of the optimization problem

$$\min_{\underrightarrow{\Delta\boldsymbol{u}}_{k+1}} J_{k_i}, \text{ s.t } \boldsymbol{M}\underrightarrow{\Delta\boldsymbol{u}}_{k+1} \leq \boldsymbol{\gamma} \tag{10}$$

can be found by solving the corresponding dual quadratic programming problem. The present work implements an iterative Gauss-Seidel variant to generate the quadratic program solution (see Hildreth (1957)). The method utilizes a primal-dual approach (see Fiacco and McCormick (1964)) to solve a linear system of equations using an element-by-element reduction which avoids matrix inversion. The algorithm has been shown to be robust in practice.

### 3.2 MPC Incorporating Split-Future Control Dynamic

The standard MPC construction of $\underrightarrow{\Delta\boldsymbol{u}}_{k_i+1}$, for the case $n_c < n_p$, assumes that all future control input increments beyond the control horizon up to the prediction horizon are equal to zero. This implies that all future control inputs within the interval $(k_i + n_c + 1) \leq k \leq n_p$ are held constant, i.e., $u_j = u_{k_i+n_c}$ for $j = k_i + n_c + 1, k_i + n_c + 2, \ldots, k_i + n_p$. This is an important observation, as the solution of the entire optimal sequence of future control inputs is heavily influenced by this assumption. Loosely speaking, the 'far-future' interval of constant control inputs is taken into account by the optimization, which produces a resulting sequence $\underrightarrow{\Delta\boldsymbol{u}}_{k_i+1}$ that may not achieve the optimal battery fast-charge objective we seek. For the case $n_c = n_p$, all future input increments are computed explicitly (at higher computational cost), and the proposed technique is not applicable.

We propose here an alternative formulation that permits far-future control actions to assume a simple and computationally manageable form. By permitting additional flexibility in the future control profile, it is conjectured the algorithm can better address highly dynamic system response without the additional computational burden of increasing $n_c$. The formulation is based on an exponential time series, $u_{k_i+m} = u_{k_i} e^{-\alpha m \Delta t}$, where $m = \{0, 1, 2, \ldots, n_p - n_c\}$ and $\alpha$ is chosen to deliver a desired decay rate. This functional form is extremely simple, yet provides a wide range of suitable future control profiles (see Fig. 3). In order to investigate the hypothesis of achieving a more aggressive input current profile, we shall examine here only the bounding case where $\alpha \to \infty$, which brings the far-future control inputs immediately to zero.

The proposed modification to the standard MPC algorithm can be achieved by augmenting the control increment vector $\underrightarrow{\Delta\boldsymbol{u}}_{k_i+1}$ one time step beyond the control horizon, in order to introduce a *zero-dynamic* to the remaining prediction values of the control input. A comparison of this modified MPC approach, which we denote here by "split-future" MPC, and the standard MPC formulation is depicted in Fig. 3

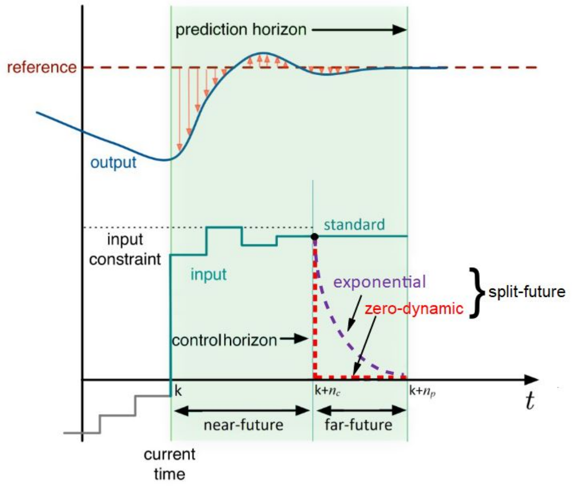

Fig. 3. Comparison of MPC methods.

We begin by defining the augmented control increment vector $\underrightarrow{\Delta\boldsymbol{U}}_{k+1} = \left[\underrightarrow{\Delta\boldsymbol{u}}^{\mathsf{T}}_{k+1,\mathrm{n}} \middle| \underrightarrow{\Delta\boldsymbol{u}}^{\mathsf{T}}_{k+1,\mathrm{f}}\right]^{\mathsf{T}}$ and the augmented MPC gain $\tilde{\boldsymbol{G}} = \left[\boldsymbol{G}_{\mathrm{n}} \middle| \boldsymbol{G}_{\mathrm{f}}\right]$, where the subscripts 'n' and 'f' denote "near-future" and "far-future", respectively, $\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}} = \underrightarrow{\Delta\boldsymbol{u}}_{k+1}$, $\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{f}} \in \mathbb{R}$, $\boldsymbol{G}_{\mathrm{n}} = \boldsymbol{G}$, and $\boldsymbol{G}_{\mathrm{f}} \in \mathbb{R}^{n_p}$. The near-future interval is defined as $(k_i + 1) \leq k \leq n_c$, whereas the far-future interval is bounded as $(k_i + n_c + 1) \leq k \leq n_p$.

Using these partitioned matrices, we can define an expression for the predicted outputs similar to (6). Multiplying out the terms yields the following result:

$$\underrightarrow{\boldsymbol{y}}_{k+1} = \Phi\tilde{A}\boldsymbol{\chi}_{k_i} + \boldsymbol{G}_{\mathrm{n}}\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}} + \boldsymbol{G}_{\mathrm{f}}\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{f}}. \tag{11}$$

Given the decision variable for the optimization problem described in (10) is $\underrightarrow{\Delta\boldsymbol{u}}_{k+1} = \underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}}$, the tasks are now: (i) to formulate $\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{f}}$ in terms of $\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}}$, and (ii) to define $\boldsymbol{G}_{\mathrm{f}}$. We begin with the former.

Since we wish to bring $u_{j|k_i}$ to zero for $j = k_i + n_c + 1, k_i + n_c + 2, \ldots, k_i + n_p$, we proceed by first writing

$$\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{f}} = \Delta u_{k_i+n_c+1|k_i} = -u_{k_i+n_c|k_i}. \tag{12}$$

Next, we must define an expression for $u_{k_i+n_c|k_i}$ in terms of the elements in $\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}}$. Recall that $\Delta u_{k+1} = u_{k+1} - u_k$, where $u_k$ is the last implemented control signal according to the receding horizon principle. If we propagate this expression up to the control horizon $n_c$ we obtain

$$u_{k+n_c} = \Delta u_{k+n_c} + \Delta u_{k+n_c-1} + \ldots + \Delta u_{k+1} + u_k,$$

which can be organized in the following compact form:

$$u_{k+n_c} = \left(\sum_{j=1}^{n_c} \Delta u_{k+j}\right) + u_k. \tag{13}$$

Since the elements in the summation are the future predictions of the input increments within the near-future prediction window, and hence, are not available yet, we need to write (13) in vector form such that

$$u_{k+n_c} = \mathbf{1}\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}} + u_k, \tag{14}$$

where $\mathbf{1}$ is a row vector of ones of appropriate length.

We can build $\boldsymbol{G}_{\mathrm{f}}$ by extending the original $\boldsymbol{G}$ matrix by one more column, and then make $\boldsymbol{G}_{\mathrm{f}}$ equal to this extra column. If we use the result in (14), we can re-write (11) as

$$\underrightarrow{\boldsymbol{y}}_{k+1} = \Phi\tilde{A}\boldsymbol{\chi}_{k_i} + (\boldsymbol{G}_{\mathrm{n}} - \boldsymbol{G}_{\mathrm{f}}\mathbf{1})\,\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}} - \boldsymbol{G}_{\mathrm{f}}u_k. \tag{15}$$

The expression above is used in (9) to solve for $\underrightarrow{\Delta\boldsymbol{u}}_{k+1,\mathrm{n}}$ in the optimization problem defined in (10).

## 4. FAST-CHARGE PROBLEM FORMULATION

The fast-charge problem involves finding the control policy over the input current required to bring a battery cell from an initial SOC to a final SOC in the shortest possible time, i.e., $\min_{I_{app}}\{\text{time} - \text{to} - \text{charge}\}_{SOC_0 \to SOC_f}$.

This is a minimum-time optimal control problem which is notoriously difficult to solve; so instead we fashion a

Table 2. Test sets to verify the influence of the tuning parameters

| Test set | $n_c$ | $n_p$ | $\lambda$ |
|---|---|---|---|
| 1 | 2 | $[10, 30]$ | $10^{-7}$ |
| 2 | $[2, 6]$ | 10 | $10^{-7}$ |
| 3 | 2 | 20 | $[10^{-4}, 10^{-7}]$ |

’pseudo’ min-time problem as in (10) designed to give essentially the same result (see Xavier and Trimboli (2015)). In this framework, the cost is defined as in (9), where we make $\underrightarrow{\boldsymbol{z}}_{k+1} = \underrightarrow{\boldsymbol{y}}_{k+1}$, $\underrightarrow{\boldsymbol{r}}_{k+1} = \mathbf{1}^{\mathsf{T}} z_{\text{final}}$, and $\underrightarrow{\Delta \boldsymbol{i}}_{k+1} = \underrightarrow{\Delta \boldsymbol{u}}_{k+1}$. In order to achieve a near-optimal solution that admits large step-wise input increments, the input current is penalized by de-tuning the control weight $\bar{R}$. The linear constraints are grouped as follows

$$\underbrace{\begin{bmatrix} M_i \\ M_v \\ M_z \end{bmatrix}}_{M} \underrightarrow{\Delta \boldsymbol{i}}_{k+1} \leq \underbrace{\begin{bmatrix} \boldsymbol{\gamma}_i \\ \boldsymbol{\gamma}_v \\ \boldsymbol{\gamma}_z \end{bmatrix}}_{\boldsymbol{\gamma}},$$

where we define

$$M_i = \left[\frac{C_u}{-C_u}\right], \; M_v = \left(G_{v,\text{n}} - G_{v,\text{f}}\mathbf{1}\right),$$

$$M_z = \left(G_{z,\text{n}} - G_{z,\text{f}}\mathbf{1}\right), \; \boldsymbol{\gamma}_i = \left[\frac{\mathbf{1}\left(i_{\max} - i_{k_i}\right)}{-\mathbf{1}\left(i_{\min} - i_{k_i}\right)}\right],$$

$$\boldsymbol{\gamma}_v = \mathbf{1}^{\mathsf{T}} v_{\max} - \Phi_v \tilde{A} \boldsymbol{\chi}_{k_i} - \mathbf{1}^{\mathsf{T}} \text{OCV}\left(z_{k_i}\right) + \boldsymbol{G}_{v,\text{f}} i_{k_i},$$

$$\boldsymbol{\gamma}_z = \mathbf{1}^{\mathsf{T}} z_{\max} - \Phi_z \tilde{A} \boldsymbol{\chi}_{k_i} + \boldsymbol{G}_{z,\text{f}} i_{k_i},$$

and $C_u \in \mathbb{R}^{n_c \times n_c}$ is a lower triangular matrix with all non-zero entries equal to one.

## 5. SIMULATION RESULTS

In order to show the viability of achieving a more aggressive control policy – and therefore a shorter time-to-charge – using the split-future MPC approach, we compare simulation results against standard MPC. For the simulation study, we aim to drive the battery SOC from an initial value of 10% to a final value of 90%. For simplicity, we assume full state information; hence no estimators are employed. The operating constraints are defined as follows: $-150\,\text{A} \leq i_{k_i} \leq 0\,\text{A}$, $v_{k_i} \leq 4.2\,\text{V}$, and $z_{k_i} \leq 90\,\%$.

In addition, we investigate the influence of the MPC design parameters on the resulting charge profile. For this we ran three different test sets; these are summarized in Table 2. The effect of varying the prediction horizon is verified by test set 1 and is shown in Figs. 4 and 5. For this test set, both control horizon and control weight are kept fixed. It can be seen that split-future MPC yields a more aggressive input profile as it displays sharper transitions towards the end of the charging event. Also, it is noted that split-future MPC is independent from the prediction horizon, while standard MPC slows down as $n_p$ increases.

The effect of varying $n_c$ is verified by test set 2 and can be seen in Figs. 4 and 6. For this set, the prediction horizon and control penalty are kept fixed. Here we note, as $n_c$ increases, the difference in charging time between the approaches decreases. This is due to the fact that a longer control horizon reduces the slack time between the control and prediction horizons, hence reducing the effect of zeroing the predicted control inputs within this prediction window. Also, a larger $n_c$ reduces overall charge time regardless of MPC approach.

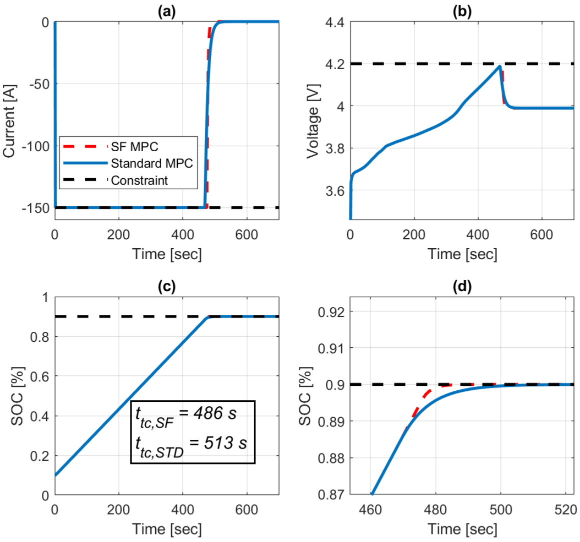

Fig. 4. $n_c = 2$, $n_p = 10$ and $\bar{R} = 10^{-7}$. (a) Charging current, (b) Voltage, (c) SOC, (d) Zoomed SOC.

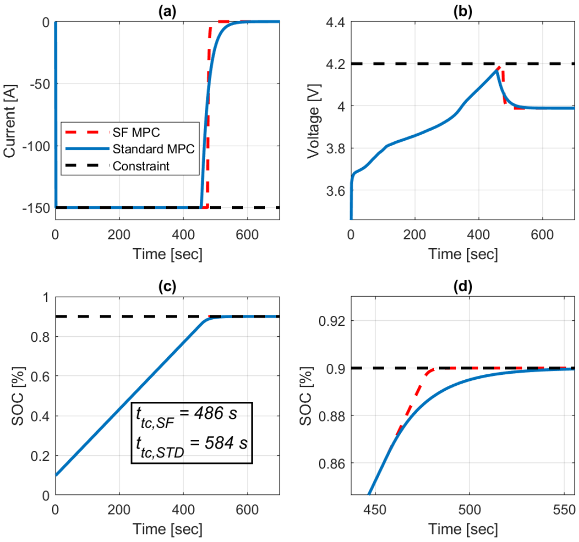

Fig. 5. $n_c = 2$, $n_p = 30$ and $\bar{R} = 10^{-7}$. (a) Charging current, (b) Voltage, (c) SOC, (d) Zoomed SOC.

Finally, the effect of varying the control penalty while varying the control and prediction horizons can be observed in Figs. 7 and 8. The trend here is that a larger penalty will decrease the total charging time for both approaches. Note that in Fig. 7, although a larger penalty would delay the split-future MPC to compute maximum input current, it is still able to deliver a shorter charging time.

## 6. CONCLUSIONS

We propose a modification to the standard MPC algorithm that introduces flexible, yet simple, far-future control predictions in order to achieve a more aggressive input current profile, and thus reduce time-to-charge a battery from an initial SOC to a desired final value. Split-future MPC

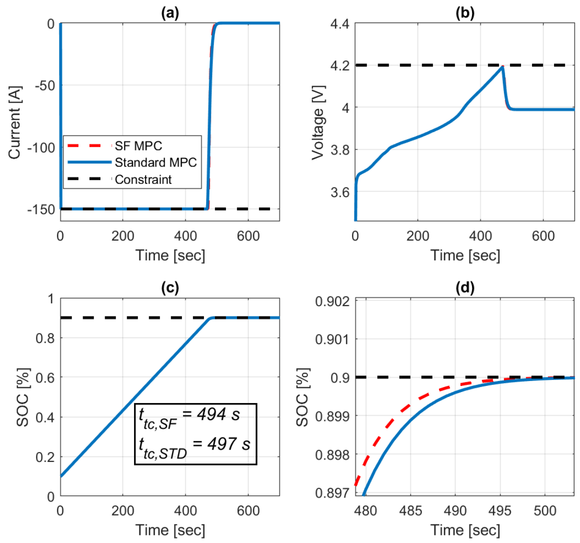

Fig. 6. $n_c = 6$, $n_p = 10$ and $\bar{R} = 10^{-7}$. (a) Charging current, (b) Voltage, (c) SOC, (d) Zoomed SOC.

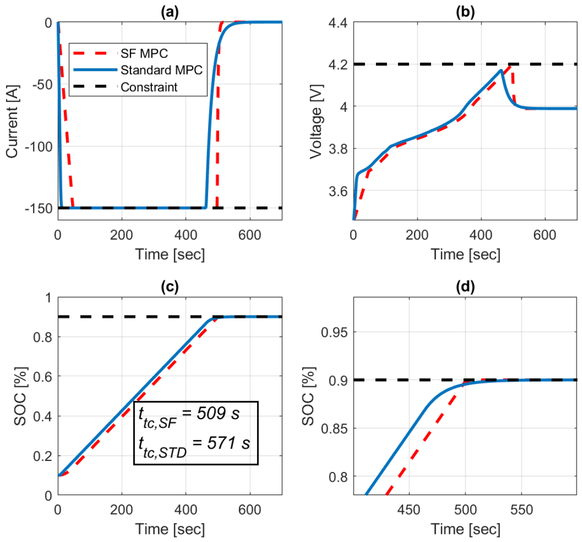

Fig. 7. $n_c = 2$, $n_p = 20$ and $\bar{R} = 10^{-4}$. (a) Charging current, (b) Voltage, (c) SOC, (d) Zoomed SOC.

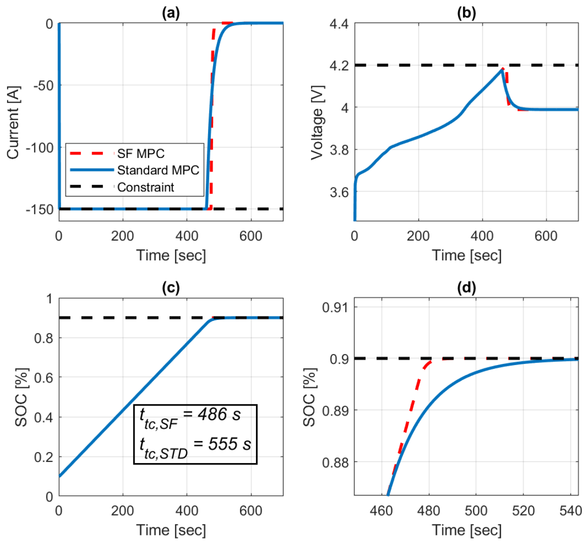

Fig. 8. $n_c = 2$, $n_p = 20$ and $\bar{R} = 10^{-7}$. (a) Charging current, (b) Voltage, (c) SOC, (d) Zoomed SOC.

assumes prediction values of the control input between the control and prediction horizons are set to an exponential decay, as opposed to standard MPC which keeps the future control inputs constant. Simulations showed that split-future MPC out-performs standard MPC by delivering faster charging times via more aggressive input current profiles, despite choice of design parameters.

## REFERENCES

Abdollahi, A., Han, X., Avvari, G., Raghunathan, N., Balasingam, B., Pattipati, K., and Bar-Shalom, Y. (2016). Optimal battery charging, part i: Minimizing time-to-charge, energy loss, and temperature rise for ocv-resistance battery model. *Journal of Power Sources*, 303, 388–398.

Chan, C. and Wong, Y. (2004). Electric vehicles charge forward. *IEEE P&E Magazine*, 2(6), 24–33.

Fiacco, A.V. and McCormick, G.P. (1964). The sequential unconstrained minimization technique for nonlinear programing, a primal-dual method. *Management Science*, 10(2), 360–366.

Hildreth, C. (1957). A quadratic programming procedure. *Naval research logistics quarterly*, 4(1), 79–85.

Klein, R., Chaturvedi, N.A., Christensen, J., Ahmed, J., Findeisen, R., and Kojic, A. (2011). Optimal charging strategies in lithium-ion battery. In *Proceedings of the 2011 American Control Conference*, 382–387. IEEE.

Kouvaritakis, B., Rossiter, J.A., and Cannon, M. (1998). Linear quadratic feasible predictive control. *Automatica*, 34(12), 1583–1592.

Liaw, B.Y., Nagasubramanian, G., Jungst, R.G., and Doughty, D.H. (2004). Modeling of lithium ion cells—a simple equivalent-circuit model approach. *Solid State Ionics*, 175(1), 835–839. Fourteenth International Conference on Solid State Ionics.

Plett, G.L. (2015). *Battery Management Systems, Volume I: Battery Modeling*. Artech House.

Remmlinger, J., Tippmann, S., Buchholz, M., and Dietmayer, K. (2014). Low-temperature charging of lithium-ion cells part ii: Model reduction and application. *Journal of Power Sources*, 254, 268–276.

Rossiter, J.A., Kouvaritakis, B., and Rice, M. (1998). A numerically robust state-space approach to stable-predictive control strategies. *Automatica*, 34(1), 65–73.

Suthar, B., Sonawane, D., Braatz, R.D., and Subramanian, V.R. (2015). Optimal low temperature charging of lithium-ion batteries. *IFAC-PapersOnLine*, 48(8), 1216–1221.

Wang, L. (2004). Discrete model predictive controller design using laguerre functions. *Journal of process control*, 14(2), 131–142.

Xavier, M.A. and Trimboli, M.S. (2015). Lithium-ion battery cell-level control using constrained model predictive control and equivalent circuit models. *Journal of Power Sources*, 285, 374–384.

Zou, C., Hu, X., Wei, Z., Wik, T., and Egardt, B. (2018). Electrochemical estimation and control for lithium-ion battery health-aware fast charging. *IEEE Transactions on Industrial Electronics*, 65(8), 6635–6645.