

# A Model Predictive Control Application for a Constrained Fast Charge of Lithium-ion Batteries

Alberto Romero\* Alejandro Goldar Emanuele Garone

Service d'Automatique et d'Analyse des Systèmes, Université libre de Bruxelles, Belgium,  
{aromerof, agoldard, egarone}@ulb.ac.be

## Abstract

The spread of electrical storage devices continues to be underpinned by the limited charging currents that can be applied. The limitation arises from the lack of sufficient high power charging stations, either at home or along roads and highways, and from the maximum admissible current that can be applied to the battery before undesirable degradation mechanisms are triggered. Accordingly, most traditional charging protocols limit the charging current as a function of the standing state of charge of the battery. These protocols are designed empirically and restrict the potential benefit of more flexible charging options. However, the alternative to traditional protocols must rely on a more precise knowledge of the operating constraints and on advanced control techniques to compute online the best operating plan. This work presents a model predictive control (MPC) application to minimize the charging time of a lithium-ion battery subject to electrochemical and thermal constraints. The satisfaction of these constraints ensures that the battery degradation is minimized, or at least mitigated. The programming language Modelica and the optimization and simulation framework JModelica.org is used in combination with Python language to assess the computing time and potential use of MPC and the developed cell models in commercial batteries.

*Keywords:* Fast-charge, nonlinear MPC, optimization, battery aging

## 1 Introduction

Lithium-based battery cells dominate the spectrum of electrical storage when it comes to portable electronic/electric devices. These cells can withstand thousands of charge-discharge cycles before degradation makes them unusable. However, when the batteries are charged at high rates, the expected life is reduced. In general, the degradation rate depends on the charging current and the cell temperature. Empirical results show that operation above 40 °C can dramatically reduce the life of the cell (Wang et al., 2011; Ecker et al., 2012). The same results from operating at low temperatures (below 10°C).

High charging current increases the voltage of the battery due to the effect on the cell of the electrochemically induced overpotentials. This observable effect is a con-

sequence of the ohmic, charge transfer, and diffusion resistances (Gerl et al., 2014). When these exceed certain thresholds then the side-reactions that cause cell degradation are triggered.

The overpotentials depend on the current applied  $I$ , the temperature  $T$ , and the state of charge (SOC). It is expected, therefore, that if certain operating constraints defined as a function of these variables are not violated, then the degradation mechanisms will be ceased or at least slowed down. How these constraints are identified and modeled has been covered recently by different authors (e.g. (Moura et al., 2013; Romagnoli et al., 2017)). It has been more common, however, to use empirical results of capacity and power fade (degradation) as a function of SOC,  $I$  and  $T$ , to subsequently obtain empirical degradation expressions by some fitting procedure. The resulting expressions can be used in combination with control schemes to maximize the benefit of utilizing the cells in the best possible way, often optimal under certain criterion.

In the recent years, Modelica has been chosen by many research institutions and companies to study the interaction of batteries with other systems such as power-trains, cooling devices, or power electronics. The component-oriented programming facilitated by Modelica can be used to compare multiple configurations in the design stage, as well as to optimize the size and operation of battery systems. Several libraries have been developed in the last years ((Dao and Schmitke, 2015; Uddin and Picarelli, 2014; Gerl et al., 2014; Bouvy et al., 2012; Brembeck and Wielgos, 2011; Einhorn et al., 2011; Janczyk et al., 2016)), which have been validated with experimental data. Applications have focused mostly on hybrids, plug-in hybrids, and full electric vehicles, with emphasis on fuel economy (Batteh and Tiller, 2009; Spike et al., 2015), thermal management (Bouvy et al., 2012), and battery aging (Gerl et al., 2014). Despite the significant effort, there is still some room for improvement in the area of constrained control techniques using Modelica to explicitly account for electrochemical and thermal operating boundaries.

This paper presents an optimal control strategy to charge a lithium-ion battery cell subject to electrochemical constraints. The model used to describe the cell dynamics and to draw the operational limits is the so-called Equivalent Hydraulic Model (EHM), which is linear on

\*Corresponding author

the two electrochemical states (state of charge and critical surface concentration), and nonlinear on the output voltage. In addition to the electrochemical states, the cell temperature dynamics is also modeled and constrained. The optimal charging profile is obtained and applied over a receding horizon following a classical MPC approach, and is implemented in Python and JModelica.org. The optimal control strategy is compared with a commercial constant-current/constant-voltage (CCCV) charging protocol that is the standard in most applications, illustrating the benefits of the optimal constrained charging strategy.

The remainder of this work can be summarized as follows. Section 2 introduces the EHM model and its extension to include the temperature state. Section 3 describes the conventional and the optimal charging strategies, as well as the implementation details. Results and their discussion are covered in Section 4, followed finally by the conclusions and future work.

## 2 Battery modeling and control

This section presents a reduced order model of the battery cell taken from the literature, as well the description of the coupled thermal model and two control strategies that are later compared for battery charging.

### 2.1 Equivalent Hydraulic Model

The equivalent hydraulic model (EHM) was first proposed by (Manwell and McGowan, 1993) for lead acid batteries, but it has been recently applied to lithium-ion cells as a reduced-order electrochemical battery model. Figure 1 depicts the physical meaning of the model states and how they interact with the flow of lithium ions. The model captures the dynamics of an idealized two-layer single particle within which lithium accumulates. The dynamics can be represented with more traditional two-tank hydraulic model (?). In continuous-time, this model takes the form

$$\frac{dSOC}{dt} = -\gamma I \quad (1)$$

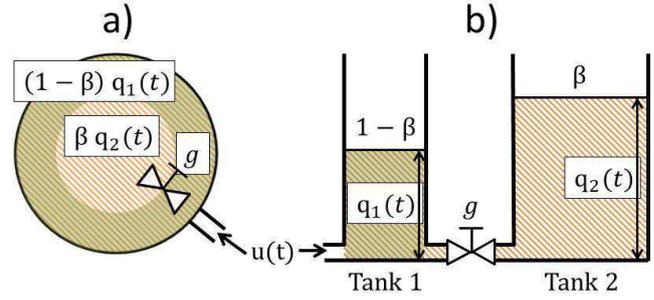
$$\frac{dCSC}{dt} = \frac{g(SOC - CSC)}{\beta(1-\beta)} - \frac{\gamma}{1-\beta} I \quad (2)$$

where SOC and CSC are the state of charge and the critical surface concentration of lithium ions respectively,  $I$  is the applied current in  $[A \cdot m^{-2}]$ , and  $g$ ,  $\beta$  and  $\gamma$  are constant parameters as defined in (Couto and Kinnaert, 2018). The convention of negative currents for battery charge is respected here.

The positive electrode dynamics are usually faster than the negative electrode ones, which motivates the assumption of fast dynamics for the former electrode, i.e.

$$\begin{aligned} CSC^+ &= SOC^+ \\ SOC^+ &= \rho SOC + \sigma \end{aligned} \quad (3)$$

where  $\rho$  and  $\sigma$  are also constant parameters. The output voltage of the battery is a nonlinear function consisting of



**Figure 1.** Equivalent hydraulic model and the spherical-solid particle representation (Couto et al., 2016).  $u(k)$ ,  $q_1$  and  $q_2$  are, respectively, the lithium flow, and the bulk and surface lithium concentration

the open-circuit voltage  $\Delta U$ , the surface overpotential  $\eta_s^\pm$  and the film resistance  $R_f$  given by

$$V = \Delta U + \eta_s^+ - \eta_s^- - R_f \gamma I \quad (4)$$

where  $\Delta U = f(SOC, CSC)$  and

$$\eta_s^\pm = C \sinh^{-1} \left( \frac{\theta^\pm}{\sqrt{z(1-z)}} I \right),$$

where  $z = CSC^+ = \rho SOC + \sigma$  for the positive component and  $z = CSC$  for the negative one, and  $C$  and  $\theta^\pm$  are constant parameters. All parameters and functions of the EHM are chemistry dependent.

In order to avoid the main side reactions that compromise battery life and its safe operation, the battery overpotentials should be restricted through constraints. These constraints are in general nonlinear, which may result in solution spaces that are nonconvex (Romagnoli et al., 2017). However, it is possible to convexify the solution space with more conservative linear constraints of the form:

$$I \geq \alpha_i CSC + \beta_i, \quad i = 1, \dots, n_c \quad (5)$$

where  $\alpha_i$  and  $\beta_i$  are the parameters associated to the linear approximations of the nonlinear side reaction constraints, and  $n_c$  is the considered number of linear constraints. In this work, only two linear approximations are used to describe the nonconvex solution space.

Besides electrochemical side reactions, there are operational limits of the electrode materials that need to be respected. Introducing or extracting more Li-ions that the allowed limit  $SOC_{max}$  results in accelerated battery degradation (Tang et al., 2009; Hausbrand et al., 2015). Finally, a maximum input current  $I_{max}$  could also be imposed, which represents a given maximum C-rate, which is a multiple of the current that charges the battery in 1 hour. If the capacity of the battery is  $34 [Ah \cdot m^{-2}]$ , then a C-rate 1C corresponds to  $-34 [A \cdot m^{-2}]$ , 3C to  $-102 [A \cdot m^{-2}]$ , and so on.

These constraints take the following form for the charging process

$$\text{SOC} \leq \text{SOC}_{\max}, \text{CSC} \leq \text{SOC}_{\max} \quad (6)$$

$$\text{SOC}^+ \geq \text{SOC}_{\max}^+, \text{CSC}^+ \geq \text{SOC}_{\max}^+ \quad (7)$$

$$I \leq I_{\max} \quad (8)$$

## 2.2 Thermal Model

The previous reduced-order electrochemical model can be augmented to include the effect on the cell temperature of the heat generated when charging or discharging the battery. The augmented model is nonlinear because the heat generated depends on the product of voltage and current, both variables of the system. The first order differential equation governing the thermal dynamics can be written as

$$m_{\text{cell}} C_{p,\text{cell}} \frac{dT_{\text{cell}}}{dt} = hA(T_{\text{cell}} - T_{\text{amb}}) + \dot{Q}_{\text{gen}} \quad (9)$$

where  $T_{\text{cell}}$  is the cell temperature,  $m_{\text{cell}}$  is the cell mass,  $C_{p,\text{cell}}$  the specific heat,  $h$  is the overall heat transfer coefficient (accounting for convection and conduction),  $A$  is the heat exchange area,  $T_{\text{amb}}$  is the ambient temperature, and  $\dot{Q}_{\text{gen}}$  is the heat generated by the cell.

The thermal model for simulation is based on (Onda et al., 2003), where the total heat generated is defined as

$$\dot{Q}_{\text{gen}} = I(\Delta U - V - T_{\text{ref}} \frac{d(\Delta U)}{dT}) \quad (10)$$

where  $T_{\text{ref}}$  is a reference temperature, and  $\frac{d(\Delta U)}{dT}$  can be calculated as the entropy change  $\Delta S$  divided by the Faraday constant ( $F$ ). The value of  $\Delta S$  is based on a LiNiCoO<sub>2</sub> pouch cell (Uddin et al., 2014).

The cell ohmic resistance  $R_{\text{int}}$  can also be used to calculate the heat generated as  $\dot{Q}_{\text{gen}} = I^2 R_{\text{int}}$ . In this work, and for the benefit of the predictive controller proposed in the following section, this expression of the heat is used in combination with the one-dimensional thermal model of the battery to approximate the heat exchange process.

Temperature constraints are motivated by the larger aging rates of batteries at higher temperatures as their lifespan roughly halves for each 13°C increase in average battery temperature (Keyser et al., 2017). The resulting upper constraint can be defined as follows:

$$T_{\text{cell}} \leq T_{\text{cell,max}} \quad (11)$$

A lower bound constraint is not considered in this work, but it would be necessary should ambient temperatures drop below 10°C.

## 2.3 Standard Charging Protocol

Standard charging protocols such as the constant current-constant voltage (CCCV) and its variations (Keil and Jossen, 2016) rely only on voltage measurements to reach

a desired SOC. A typical CCCV protocol consists of a charge period under constant current (CC), followed by a constant voltage (CV) stage. The CV stage begins when a predefined voltage threshold is reached, and terminates when either a fixed maximum duration time or a minimum current threshold is achieved. The CV stage can be driven by a proportional feedback controller which in practice leads to a progressive reduction of the current as the reference voltage is reached; this is the approach followed in the present study.

## 2.4 Constrained Control

A widely used strategy to cope with constrained control problems is MPC (Camacho and Bordons, 2004). Model predictive controllers calculate the future control actions on the process,  $u(t)$ , by solving an on-line optimization problem subject to constraints that can be written as

$$\begin{aligned} \min_{u(t)} \quad & \int_{t_0}^{t_f} (\text{SOC}(t) - \text{SOC}_{\text{ref}})^2 dt \\ \text{s.t.} \quad & \text{model dynamics} \\ & \text{electrochemical constraints} \\ & \text{thermal constraints} \end{aligned} \quad (12)$$

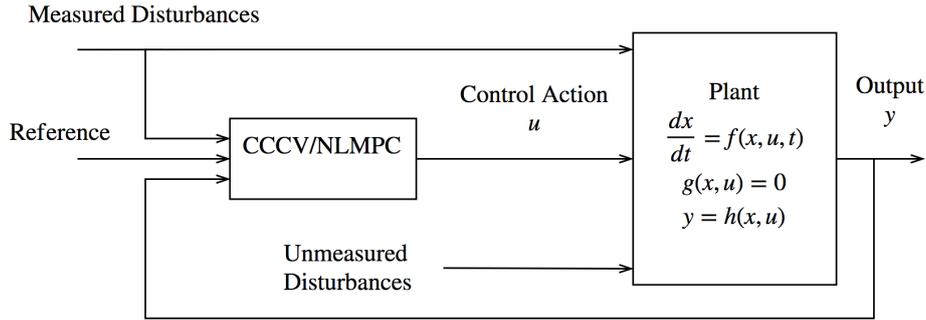
The optimization problem (12) minimizes a cost function that depends on the predicted tracking error (given a desired reference  $\text{SOC}_{\text{ref}}$ ) and the control effort. The former is the difference between the SOC and its reference. The latter depends on  $u(t)$ , which is the sequence of control actions on the system, i.e., the applied current. The time horizon ( $t_f - t_0$ ) is discretized into a finite number of time steps in which the variables of the problem are defined. Constraints include the predictions performed with a simplified battery model, as well as upper and lower bounds on states, outputs, input, or a combination of these.

## 3 Case Studies and Implementation

Figure 2 shows the control scheme adopted. The controller, which is either based on the CCCV protocol or on MPC, can receive information regarding the reference, outputs and measured disturbances. It is assumed in this work that the effect of the disturbances is negligible.

**Table 1.** CCCV controller

Variable	Value	Units
$I_{\text{end}}$	-0.15	[A · m <sup>-2</sup> ]
$V_{\text{thres}}$	4.19	[V]
$V_{\text{ref}}$	4.2	[V]
$K_p$	-5250	[A · V <sup>-1</sup> ]
$K_i$	0	[A · (V · s) <sup>-1</sup> ]
$\Delta t_{\text{sample}}$	1	[s]
$\Delta t_{\text{sim}}$	0.1	[s]



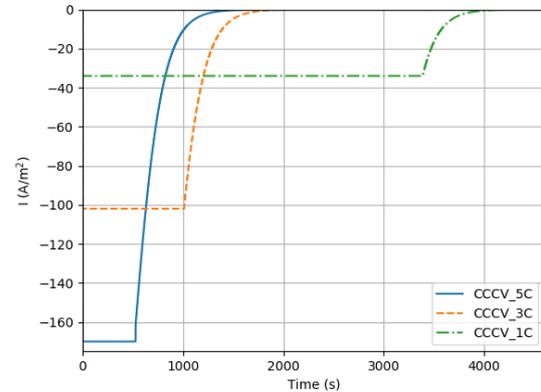
**Figure 2.** Feedback control loop including different controllers and the plant model.

Regarding the CCCV protocol, the stopping conditions and setup parameters for the proportional controller used are shown in Table 1.  $I_{\text{end}}$  and  $V_{\text{thres}}$  are the termination current and the voltage threshold respectively.  $\Delta t$  coincides with the simulation time step, and is the rate at which the controller is implemented.  $K_P$  and  $K_I$  are the proportional and integral coefficients for the PI controller, and  $V_{\text{ref}}$  define the voltage reference to calculate the error. In order to ensure a smooth transition between the CC and CV stages, the applied current in the later was filtered using the arithmetic mean of the calculated and a number of past applied currents (this number is 9 in this work).

**Table 2.** NMPC controller

Variable	Value	Units
$t_f$	200	[s]
$\text{SOC}_{\text{ref}}$	0.665	[-]
$n_e$	20	[-]
$n_{\text{cp}}$	1	[-]
$H$	200	[s]
$\Delta t_{\text{MPC}}$	10	[s]
$\Delta t_{\text{sim}}$	1	[s]
solver	IPOPT	

The implementation in Modelica of both the plant model (Appendix I), used for simulation, as well as the prediction model (Appendix II, including constraints) are included at the end of the document. Both are based on experimental data obtained for a commercial LCO batteries (Turnigy Nano-tech, 160mAh (Turnigy, 2018)). The optimal control problem was solved with the programming language Python and JModelica.org. The latter is an open source platform for optimization, simulation and analysis of complex dynamic systems (Link et al., 2015). It interfaces the numerical solver IPOPT and CasADi, which is an open source symbolic framework for automatic differentiation and optimal control (Andersson et al., 2012). The Python program to compute the MPC problem is also included in Appendix III. Table 2 shows the settings for the NMPC algorithm, which is solved using the interior point method, where  $n_e$  is the number of finite elements,



**Figure 3.** Current profiles for CCCV protocols (1C, 3C and 5C).

$n_{\text{cp}}$  is the number of collocation points in each element,  $H$  is the control horizon,  $\Delta t_{\text{MPC}}$  is the sampling time and the length of each time interval for the MPC,  $\Delta t_{\text{sim}}$  is the simulation timestep.

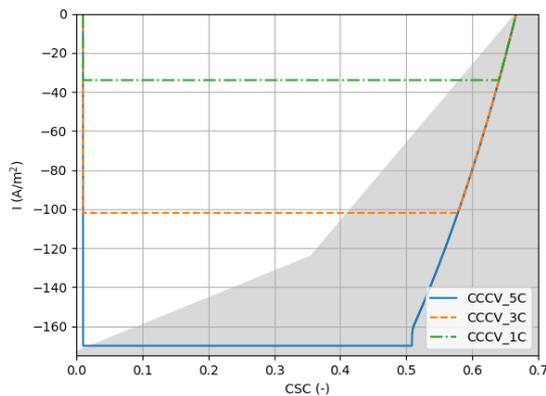
## 4 Results

In this section, we present results regarding current, temperature, voltage and critical surface concentration. The values for the current applied are shown in the units of the model ( $\text{I} \cdot \text{m}^{-2}$ ), but are also referred to in terms of the equivalent C-rate.

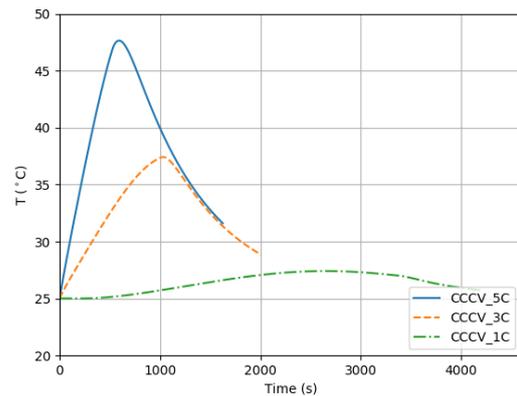
### 4.1 CCCV Protocols

The results of the application of three CCCV charge profiles with increasing constant current are discussed first. Figure 3 presents the current profiles, calculated by the proportional controller and limited by the prescribed CC current.

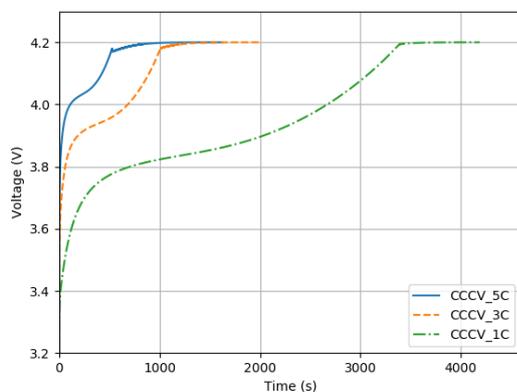
As expected, the increase of the CC rate reduces the charging time: around 4165 s for 1C, 1975 s for 3C, and 1590 s for 5C. These values depend on the battery chemistry, the controller setup, and the termination conditions. Figure 4 shows the consequence of using charging protocols that overlook the existence of an electrochemically safe operating region. The grey area denotes the combination of  $I$  and CSC where side reactions are triggered.



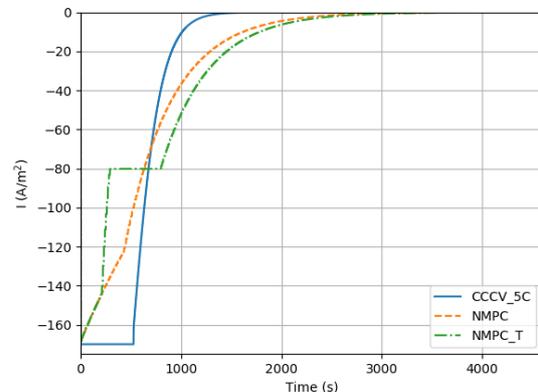
**Figure 4.** Charging trajectories in the I-CSC plane for CCCV protocols.



**Figure 6.** Temperature profiles for CCCV protocols (1C, 3C and 5C).



**Figure 5.** Voltage profiles for CCCV protocols (1C, 3C and 5C).



**Figure 7.** Current profiles for the nonlinear controllers and the 5C-CCCV protocol.

Figures 5 and 6 show the respective results for voltage and temperature. The first figure illustrates the effect of the proportional controller, which takes the same parameters for the three CCCVs. Regarding the temperature, as expected, the maximum temperature achieved varies significantly with the current applied. However, an equally important fact is that the higher the C-rate of the CC stage, the longer the cell's temperature remain above ambient temperatures. This means that calendar aging, i.e., the aging that takes place with zero current conditions and which depends on the cell temperature, will be higher for the higher C-rate CCCVs. For 1C CCCV the temperature increase remains below  $5^{\circ}\text{C}$ . Given the relationship between aging, current and temperature, it is not surprising that most manufacturers recommend the use of low C-rates for charging (below 1C).

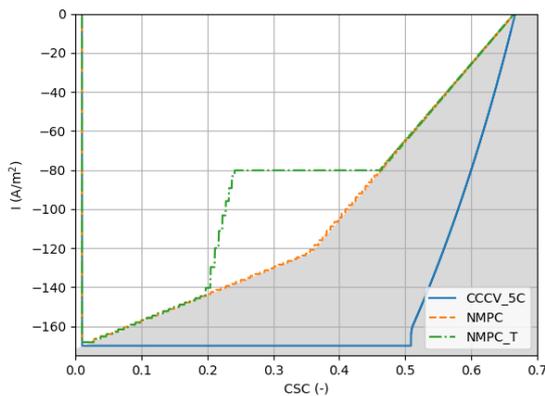
## 4.2 Nonlinear MPC

In this section, we compare the previous results for 5C CCCV with the thermally unconstrained (NMPC) and constrained (NMPC\_T) optimal controllers. For the latter, the maximum temperature allowed ( $T_{\max}$ ) is  $35^{\circ}\text{C}$ . This choice is arbitrary, but well below the maximum temper-

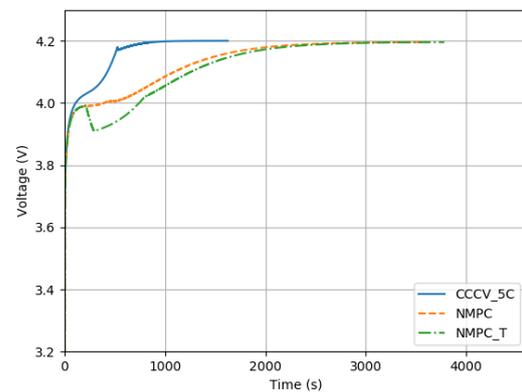
ature achieved by the 5C CCCV, and so suitable for illustrating the methodology. In practice, to properly select the boundary one should have an estimation of its impact in the battery's long-term performance, so to measure the economic gains from setting such a boundary. This issue is not addressed in the present paper.

Figure 7 shows the current profiles. Regarding the optimized profiles, they remain qualitatively close for most of the charge, except for an interval in which the current is decreased to enforce the temperature constraints for the NMPC\_T, which slightly increases the charging time.

Figure 8 shows the admissible charging region, now with the NMPC and the NMPC\_T profiles. Both remain electrochemically feasible throughout the charging process but, as a consequence of the temperature constraints, the NMPC\_T departs from the electrochemical convex hull. In this figure, as well as in Fig. 7, it can be observed that the NMPC\_T resembles a CCCV with boost charge. This charging protocol can reduce significantly the life of the cell, as shown by (Keil and Jossen, 2016). In their work, the authors present experimental results considering boost current of 5C that fills 40% of the capacity, including



**Figure 8.** Charging trajectories in the I-CSC plane for nonlinear controllers and 5C-CCCV protocols.



**Figure 9.** Voltage profiles for the nonlinear controllers and the 5C-CCCV protocol.

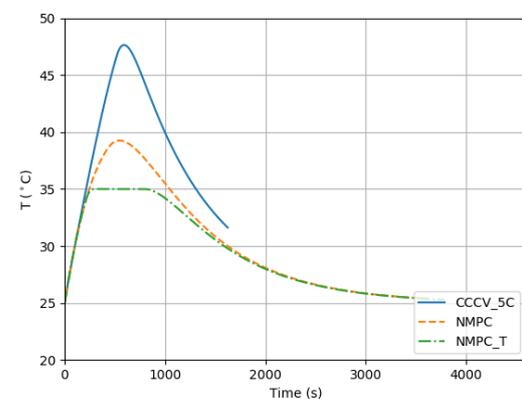
the following intervals: 0%-40%, 10%-50%, 20%-60%. The first interval results in the highest degradation, which is explained as a higher cell resistance at low SOC. Nevertheless, these three boost-charge profiles bring the 5C current into the region of side reactions, similar to what results with the 5C CCCV in the present paper (Fig. 5).

Finally, results for voltage and temperature are presented. As anticipated in the previous subsection, the electrochemical constraints limit the rate at which the voltage approaches the SOC reference. Regarding the temperature profiles, a  $\Delta T = 4^\circ\text{C}$  reduction is obtained just by imposing electrochemical constraints with a further decrease when activating the thermal constraints. Taking into account that aging increases exponentially with temperature, these differences could improve the battery life at the expense of increasing the charging times. These are 3520 s for the thermally unconstrained, and 3720 s for the thermally constrained. It is worth noting that NMPC is charged up to 80% (SOC = 0.55) in 960 s, while the NMPC\_T is charged up to this SOC in 1500 s. For the model used, CSC deviates only slightly from the SOC, and thus the charging times can be obtained comparing I-CSC and I-t plots.

Solution times when solving the NMPC problem at each time interval remain close to 0.15 s, while for the thermally constrained NMPC rise slightly up to 0.20 s. In any case the total time surpassed 0.3 s. Given the nonlinear controller update rate (10 s), this ensures that in absence of great disturbances, both controllers can be applied online, if the tools needed for estimation remain in the same order of magnitude.

## 5 Conclusions and Future Work

The fast and safe charge of lithium-ion batteries remains an open problem. Most charging protocols rely on empirically obtained parameters, and generally result too conservative, limiting the flexibility of operation. This work presents the modeling, simulation and control efforts to better understand the open challenges. In particular a



**Figure 10.** Temperature profiles for the nonlinear controllers and the 5C-CCCV protocol.

NMPC scheme was implemented in Python and JModelica.org, which provides an excellent platform to compare the results of optimization-based control methods with more traditional charging protocols that rely on relay, proportional, or PI controllers.

It can be concluded that the NMPC solver can be applied on-line, and that charging time can be reduced compared with CCCV protocols while complying with electrochemical constraints, which will presumably extend the battery life. However, adding thermal constraints can significantly limit the advantage of the optimal controller regarding the charging time. Arising from the simultaneous application of electrochemical and thermal constraints are boost-charge-like charging strategies, which make possible to charge up to 80% the battery in less than 30 min. Ongoing work focuses on time constraints to enforce a certain level of SOC within a given time, and on an optimization scheme to pre-compute electrochemically and thermal safe CCCV with boost charge.

## References

- Joel Andersson, Johan Åkesson, and Moritz Diehl. Casadi: A symbolic package for automatic differentiation and optimal control. In *Recent advances in algorithmic differentiation*, pages 297–307. Springer, 2012.
- John Batteh and Michael Tiller. Implementation of an extended vehicle model architecture in modelica for hybrid vehicle modeling: development and applications. In *Proceedings of the 7th International Modelica Conference; Como; Italy; 20-22 September 2009*, pages 823–832. Linköping University Electronic Press, 2009. 043.
- Claude Bouvy, Sidney Baltzer, Peter Jeck, Jörg Gißing, Thomas Lichius, and Lutz Eckstein. Holistic vehicle simulation using modelica-an application on thermal management and operation strategy for electrified vehicles. In *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany*, pages 264–270. Linköping University Electronic Press, 2012. 076.
- Jonathan Brembeck and Sebastian Wielgos. A real time capable battery model for electric mobility applications using optimal estimation methods. In *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical University; Dresden; Germany*, pages 398–405. Linköping University Electronic Press, 2011. 063.
- Eduardo F Camacho and Carlos Bordons. *Model predictive control. Advanced textbooks in control and signal processing*. Springer-Verlag, London, 2004.
- L. D. Couto, J. Schorsch, M. M. Nicotra, and M. Kinnaert. Soc and soh estimation for li-ion batteries based on an equivalent hydraulic model. part i: Soc and surface concentration estimation. In *2016 American Control Conference (ACC)*, pages 4022–4028, July 2016. doi:10.1109/ACC.2016.7525553.
- Luis D Couto and Michel Kinnaert. Partition-based unscented kalman filter for reconfigurable battery pack state estimation using an electrochemical model. In *2018 Annual American Control Conference (ACC)*, pages 3122–3128. IEEE, 2018.
- Thanh-Son Dao and Chad Schmitke. Developing mathematical models of batteries in modelica for energy storage applications. In *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, pages 469–477. Linköping University Electronic Press, 2015. 118.
- Madeleine Ecker, Jochen B. Gerschler, Jan Vogel, Stefan Käbitz, Friedrich Hust, Philipp Dechent, and Dirk Uwe Sauer. Development of a lifetime prediction model for lithium-ion batteries based on extended accelerated aging test data. *Journal of Power Sources*, 215:248–257, 2012. ISSN 03787753. doi:10.1016/j.jpowsour.2012.05.012.
- M Einhorn, FV Conte, C Kral, C Niklas, H Popp, and J Fleig. A modelica library for simulation of electric energy storages. In *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical University; Dresden; Germany*, pages 436–445. Linköping University Electronic Press, 2011. 63.
- Johannes Gerl, Leonard Janczyk, Imke Krüger, and Nils Modrow. A modelica based lithium ion battery model. In *Proceedings of the 10th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, pages 335–341. Linköping University Electronic Press, 2014. 096.
- R. Hausbrand, G. Cherkashinin, H. Ehrenberg, M. Gröting, K. Albe, C. Hess, and W. Jaegermann. Fundamental degradation mechanisms of layered oxide Li-ion battery cathode materials: Methodology, insights and novel approaches. *Materials Science and Engineering: B*, 192:3–25, 2015.
- Leonard Janczyk, Klemens Esterle, Stephan Diehl, Michael Seibt, Arthur Gauthier, and Viry Guillaume. Validation of a battery management system based on autosar via fmi on a hil platform. In *The First Japanese Modelica Conferences, May 23-24, Tokyo, Japan*, pages 87–94. Linköping University Electronic Press, 2016. 124.
- Peter Keil and Andreas Jossen. Charging protocols for lithium-ion batteries and their impact on cycle life-An experimental study with different 18650 high-power cells. *Journal of Energy Storage*, 6:125–141, 2016.
- Matthew Keyser et al. Enabling fast charging Battery thermal considerations. *Journal of Power Sources*, 367:228–236, 2017.
- Kilian Link, Leo Gall, Monika Mühlbauer, and Stephanie Gallardo-Yances. Experience with industrial in-house application of fmi. In *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, pages 17–22. Linköping University Electronic Press, 2015. 118.
- James F. Manwell and Jon G. McGowan. Lead acid battery storage model for hybrid energy systems. *Solar Energy*, 50(5):399 – 405, 1993. ISSN 0038-092X. doi:https://doi.org/10.1016/0038-092X(93)90060-2.
- AB Modelon. Jmodelica. org user guide, verison 2.2, 2018.
- Scott J Moura, Nalin A Chaturvedi, and M Krstić. Constraint management in li-ion batteries: A modified reference governor approach. In *American Control Conference (ACC), 2013*, pages 5332–5337. IEEE, 2013.
- Kazuo Onda, Hisashi Kameyama, Takeshi Hanamoto, and Kohei Ito. Experimental study on heat generation behavior of small lithium-ion secondary batteries. *Journal of the Electrochemical Society*, 150(3):A285–A291, 2003.
- R. Romagnoli, L. D. Couto, M. M. Nicotra, M. Kinnaert, and E. Garone. Computationally-efficient constrained control of the state-of-charge of a li-ion battery cell. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1433–1439, Dec 2017.
- Jonathan Spike, Johannes Friebe, Chad Schmitke, Christian Donn, Michael Folie, Valerie Bensch, and Christine Schwarz. Holistic virtual testing and analysis of a concept hybrid electric vehicle model. In *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, pages 537–545. Linköping University Electronic Press, 2015. 118.

Maureen Tang, Paul Albertus, and John Newman. Two-Dimensional Modeling of Lithium Deposition during Cell Charging. *Journal of The Electrochemical Society*, 156(5): A390–A399, 2009.

Turnigy, 2018. URL <http://www.turnigy.com/batteries/nano-tech/>.

Kotub Uddin and Alessandro Picarelli. Phenomenological lithium battery modelling in dymola. In *Proceedings of the 10th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, pages 327–334. Linköping University Electronic Press, 2014. 96.

Kotub Uddin, Alessandro Picarelli, Christopher Lyness, Nigel Taylor, and James Marco. An acausal Li-ion battery pack model for automotive applications. *Energies*, 7(9):5675–5700, 2014. doi:10.3390/en7095675.

John Wang, Ping Liu, Jocelyn Hicks-Garner, Elena Sherman, Souren Soukiazian, Mark Verbrugge, Harshad Tataria, James Musser, and Peter Finamore. Cycle-life model for graphite-LiFePO4 cells. *Journal of Power Sources*, 196(8):3942–3948, 2011. ISSN 03787753. doi:10.1016/j.jpowsour.2010.11.134. URL <http://dx.doi.org/10.1016/j.jpowsour.2010.11.134>.

## Appendix I. Battery plant model

```
model EHMcell
```

```
// Heat transfer problem parameters
parameter Modelica.SIunits.Resistance R =
  0.330 "Internal resistance";
parameter
  Modelica.SIunits.CoefficientOfHeatTransf
  h = 10 "Heat transfer [W/m2.K]";
parameter Modelica.SIunits.Area A = 4.7
  e-4 "Heat transfer area [m2]";
parameter Modelica.SIunits.Mass M =
  0.0045 "Cell mass [kg]";
parameter
  Modelica.SIunits.SpecificHeatCapacity
  Cp = 800 "Cell specific heat [J/kg.K
  ]";

// Electrochemical parameters
parameter Real gamma = 0.0000054581;
parameter Real g = 0.042653404659871;
parameter Real beta = 0.7;
parameter Real Ccell = 0.16 "Battery
  capacity [Ah]";
parameter Real p_oneC = 34 "1C current
  flux [A/m2]";
parameter Real p_alpha = 0.5;
parameter Real p_Faraday(unit = "C/mol")
  = 96487 "Faraday constant";
parameter Real p_R = 8.314472;
parameter Real p_T_ref = 298.15;
parameter Real p_aFRT = (p_alpha*
  p_Faraday)/(p_R*p_T_ref);
parameter Real p_thetap =
  0.003753150372049;
parameter Real p_thetan
  =0.002390386732068;
```

```
parameter Real p_Rf = 0.000846744769459;
parameter Real p_ro = 0.798857289559742;
parameter Real p_sigma =
  1.001138873133620;

// Constraint parameters
parameter Real alpha1 = 137.436438945007;
parameter Real alpha2 = 858.977743406295;
parameter Real beta1 = -171.795548681259;
parameter Real beta2 = -532.566200911903;

// The states and variables
Real SOC(start = 0.01) "State of charge";
Real CSC(start = 0.01) "Critical surface
  concentration";
Real CSCp(start = 0.01) "CSC
  Overpotential positive electrode";
Modelica.SIunits.Temp_C T(start = 25.) "
  Battery temperature";
Modelica.SIunits.Voltage Up(start = 0.1)
  "Surface overpotential positive
  electrode";
Modelica.SIunits.Voltage Un(start = 3.5)
  "Surface overpotential negative
  electrode";
Modelica.SIunits.Voltage V(start = 3.5) "
  Battery voltage";
Modelica.SIunits.MolarEntropy DS "Entropy
  change [J/mol.K]";
```

```
// The control signal
```

```
input Modelica.SIunits.ElectricCurrent I
  "Input current flux [A/m2]";
```

```
equation
```

```
der(SOC) = - gamma * I;
der(CSC) = g/beta/(1-beta) * SOC - g/beta
  /(1-beta) * CSC - gamma * I/(1-beta)
  ;
der(T) = ((-I/(p_oneC/Ccell)*(V - (-Un +
  Up) + p_T_ref*DS/p_Faraday) ) - h*A*(
  T - 25))/M/Cp;
DS = 0.5609*1e+3*(SOC/0.68)^5 -1.3440*1e
  +3*(SOC/0.68)^4 + 1.1877*1e+3*(SOC
  /0.68)^3 -0.6072*1e+3*(SOC/0.68)^2 +
  0.2378*1e+3*(SOC/0.68)^1 -0.0397*1e
  +3*(SOC/0.68)^0 ;
CSCp = -p_ro * SOC + p_sigma;
Up = (0.654807602368402*((1-CSCp)
  .^3.196972561445755))+ 3.85516954 +
  1.247319422*(1-CSCp) -
  11.15240126*(1-CSCp).^2 +
  42.8184855*(1-CSCp).^3 -
  67.71099749*(1-CSCp).^4 +
  42.50815332*(1-CSCp).^5 - 6.13244713
  e-4*Modelica.Math.exp(-7.657419995*(
  CSCp.^115.0));
Un = 8.002296379 + 5.064722977*CSC -
  12.57808059*CSC.^(1/2) - 8.632208755
  e-4*CSC.^(-1) + 2.176468281e-5*CSC.
  ^(3/2) - 0.4601573522*
  Modelica.Math.exp(15.0*(0.06 - CSC))
  - 0.5536351675*Modelica.Math.exp
  (-2.432630003*(CSC - 0.92));
V = -Un + Up -(p_Rf*I) - (1/p_aFRT)*(
  Modelica.Math.asinh((1*p_thetan*I)/
```

```

sqrt(CSC*(1-CSC))) + (1/p_aFRT)* (
Modelica.Math.asinh((-1*p_thetap*I)/
sqrt(CSCp*(1-CSCp))));

```

```
end EHMcell;
```

## Appendix II. Battery optimization model

```

optimization EHMTV_Opt (objectiveIntegrand
= (SOC - 0.665)^2,
    startTime = 0,
    finalTime = 200)

// Heat transfer problem parameters
// ...

// Electrochemical parameters
// ...

// Constraint parameters
// ...

// The states and variables
Real SOC(start = 0.01) "State of charge";
Real CSC(start = 0.01) "Critical surface
concentration";
Real CSCp(start = 0.01) "CSC
Overpotential positive electrode";
Modelica.SIunits.Temp_C T(start = 25.) "
Battery temperature";
Modelica.SIunits.Voltage Up(start = 0.1)
"Surface overpotential positive
electrode";
Modelica.SIunits.Voltage Un(start = 3.5)
"Surface overpotential negative
electrode";
Modelica.SIunits.Voltage V(start = 3.5) "
Battery voltage";
Modelica.SIunits.MolarEntropy DS "Entropy
change [J/mol.K]";

// The control signal
input Modelica.SIunits.ElectricCurrent I
"Input current flux [A/m2]";

equation
// ...
der(T) = (R*(I/(p_oneC/Ccell))^2 - h*A*(T
- 25))/M/Cp;
// ...

constraint
SOC <= 0.665 + 0*0.627319647304968;
CSC <= 0.665 + 0*0.627319647304968;
SOC >= 0.001;
CSC >= 0.001;
I >= alpha1 * CSC + beta1;
I >= alpha2 * CSC + beta2;
I >= -34*5;
I <= 0.;
T <= 300;
V <= 4.5;

end EHMTV_Opt;

```

## Appendix III. Python implementation of the MPC, based on JModelica.org User Guide (Modelon, 2018)

```

# // Program that demonstrate the native
# // FMI interface for simulation
# // Integration takes place within python

import numpy as N
import pylab as plt
from pymodelica import compile_fmu
from pyfmi.fmi import load_fmu

# // Import the function for compilation of
# // models and the load_fmu method
from pyjmi import
transfer_optimization_problem

# // Compile model and load FMU
fmu_name = compile_fmu("EHMcell", "C:\...\
EHMcell_old.mo")
model = load_fmu(fmu_name)

# // Transfer the optimization problem to
# // casadi
# // This function transfers the
# // optimization problem into Python
# // and expresses its variables,
# // equations, etc., using the
# // automatic differentiation tool
# // CasADi.
op = transfer_optimization_problem("
EHMTV_Opt", "C:\...\EHMTV_Opt.mop")

# // Optimization options
opts = op.optimize_options()
opts['n_e'] = 20
opts['n_cp'] = 1
opts['solver'] = 'IPOPT'
opts['expand_to_sx'] = 'NLP'

# // Simulation times and model
# // initialization
Tstart = 0
Tend = 3600+1000
model.time = Tstart
I0 = -1.
I02 = I0
model.set('I', I0)
h0 = 10.
model.set('h', h0)
model.initialize()

# // Data to be stored in the integration
# // loop
# // Get continous States
x = model.continuous_states
# // Get the Nominal Values
x_nominal = model.nominal_continuous_states
# // Get the Event Indicators
event_ind = model.get_event_indicators()

# // Values for the solution
# // Retrieve the valuerferences for the
# // values 'SOC', 'CSC' and 'T'
states0 = [model.get_variable_valueref('SOC
')] + [model.get_variable_valueref('CSC
')]

```

```

    ')] + [model.get_variable_valueref('T')
    ]
vol0 = model.get_variable_valueref('V')
voltage = [model.get_real(vol0)]
vol = model.get_real(vol0)
t_sol = [Tstart]
sol = [model.get_real(states0)]
I_sol = [I0]

# // Initialize integration time and define
# the step-size
time = Tstart
Tnext = Tend # Used for time events
dt = 1 # Step-size
Tend = -0.15

# // Main integration loop using explicit
# Euler method.
# // This is the integration loop for
# advancing the solution one step at a
# time.
# // The loop continues until the final
# time has been reached or
# // if the FMU reported that the
# simulation is to be terminated.

count = 0
while time < Tend and not
model.get_event_info().
terminateSimulation:

# // Compute the derivative of
# the previous step  $f(x(n), t(n))$ 
dx = model.get_derivatives()

# // Advance
h = min(dt, Tnext - time)
time = time + h

# // Set the time
model.time = time

# // Set the states at  $t = time$ 
# // Perform the step using  $x(n+1) = x(
n) + h*f(x(n), t(n))$ 
x = x + h*dx

# // To make sure that the plant
# does not feed back infeasible
# states
if x[2] >30 and x[2]<31:
x[2]=30
if x[2] > 31:
break

model.continuous_states = x

# // Retrieve solutions at  $t = time$  for
# outputs
# // model.get_real, get_integer,
# get_boolean, get_string(valueref)

t_sol += [time]
sol += [model.get_real(states0)]
vol = model.get_real(vol0)
voltage += [vol]

if vol > 4.5:
I0 = 0.;
model.set('I', I0)

op.set('SOC0', float(N.array(sol)[-1,0])
)
op.set('CSC0', float(N.array(sol)[-1,1])
)
op.set('CSCp0', float(N.array(sol)
[-1,1]))
op.set('T0', float(N.array(sol)[-1,2]))

clause = count \% 10
if clause <= 1e-5:
resopt = op.optimize(options = opts
)
Iopt = resopt['I']
print Iopt[0]
if count > 100:
I0 = Iopt[0]
else:
I0 = Iopt[0]
model.set('I', I0)

I_sol += [I0]
count=count+1
if I0 >= Tend:
break

# // Plots...

```