2021 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS) | 978-1-6654-0115-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/SAFEPROCESS52771.2021.9693708

Lithium-ion battery state-of-charge estimation using a real-time moving horizon estimation algorithm

Jiayu Yan
Huazhong University of Science and Technology, Key Laboratory of Image Processing and Intelligent Control, Ministry of Education
Wuhan, China

Song Li
the 29th Research Institute of China Electronics Technology Group Corporation
Chengdu, China
lisongCETC@126.com

Yiming Wan*
Huazhong University of Science and Technology, Key Laboratory of Image Processing and Intelligent Control, Ministry of Education
Wuhan, China
ywan@hust.edu.cn

***Abstract*—For battery management systems, it is significant to reliably estimate state-of-charge (SOC) from limited measurements in real time. Based on a nonlinear SOC-dependent equivalent circuit model, we propose a real-time moving horizon estimation (MHE) algorithm for SOC estimation. For efficient computation, the sequential quadratic programming strategy is applied, and each quadratic programming (QP) iteration is solved via the structure-exploiting forward and backward recursions. To account for the limited computation time allowed within each sampling interval, the proposed algorithm does not fully solve the receding horizon optimization problem over each sliding window, but performs fixed number of QP iterations. The simulation results show that compared to the existing fully solved MHE applied to SOC estimation, the proposed real-time MHE algorithm significantly saves computation time at the cost of a slight loss of estimation performance.**

***Index Terms*—Lithium-ion batteries, equivalent circuit model, state-of-charge, moving horizon estimation, real-time computation.**

I. Introduction

In recent years, lithium-ion batteries (LIBs) have been widely used in various energy storage systems for transportation vehicles, smart grids, and consumer electronics [1]. To guarantee safety and reliability of LIBs, a battery management system (BMS) is needed to monitor the battery states. As a key parameter in BMS, state of charge (SOC) should be monitored to avoid overcharge and overdischarge [2].

However, SOC cannot be directly measured. Instead, we need to estimate SOC from available measurements [3]. At present, there are three main categories of SOC estimation methods, i.e., the coulomb counting method, the open-circuit voltage method, and the model-based state estimation method [4]. The coulomb counting method is vulnerable to sensor noise and highly dependent on the initial value [5]. The open-circuit voltage method needs a long relaxation time to get an accurate open circuit voltage, thus is hardly applied to online operation [6]. The model-based approach exploits a battery model to estimate SOC from the measured terminal voltage and load current.

This work is supported by the National Natural Science Foundation of China under Grant No. 61803163. Corresponding Author: Yiming Wan.

The LIB models used for SOC estimation can be classified into three categories: the electrochemical model, the data-driven model, and the equivalent circuit model (ECM) [7]. Although the electrochemical model can accurately describe the battery electrochemical reactions, the heavy computation cost limits its real-time application [8]. The data-driven model approach relies on precise and accurate training data, whose generalization performance cannot be guaranteed [9].

Compared to electrochemical and data-driven models, the ECM achieves a sufficiently accurate representation of the battery dynamics via simple circuit elements, thus is more widely adopted in practical BMSs [10]. Based on the ECM, variants of nonlinear Kalman filter such as extended Kalman filter (EKF) have been applied for the SOC estimation [11], [12].

However, it is well-known that EKF has some inherent limitations. On the one hand, the EKF with a poor initial guess often gives inaccurate state estimates which might eventually diverge from the true state [13]. On the other hand, the EKF algorithm cannot handle practical constraints of the system states [14].

To address the above limitations of EKF, moving horizon estimation (MHE) has been proposed in [15]–[17] for nonlinear constrained systems. MHE solves a constrained least-squares estimation problem at each time instant that involves the dynamic system model and the past measurement sequence over a receding horizon [15]. Compared to EKF, the MHE algorithm converges faster and has higher reliability in estimating the states when coping with inaccurate initial state guesses [16], [17].

It should be noted that the MHE requires solving an optimization problem online over each receding horizon, thus its improved estimation performance is achieved at a high computational cost [18], [19]. Moreover, the associated computational cost may grow fast as the horizon length increases [20]. The real-time computation issue has to be addressed when applying MHE into practice.

To address the above challenge for the SOC estimation, we propose a real-time MHE algorithm, which is inspired by [21]. Specifically, the sequential quadratic programming (SQP) strategy is adopted, and the Gauss-Newton method

Authorized licensed use limited to: TU Delft Library. Downloaded on May 09,2022 at 07:30:18 UTC from IEEE Xplore. Restrictions apply.

is applied in each iteration to obtain an equality-constrained quadratic programming (QP) subproblem [22]. In order to solve the QP subproblem, we use structure-exploiting forward and backward recursions [23]. To account for the limited time allowed within each sampling interval, the proposed algorithm fixes the number of iterations over each receding horizon, without fully solving each MHE problem.

This paper is divided into five sections as follows: The first part is the Introduction. Section II describes the SOC-dependent ECM. The proposed real-time MHE algorithm is elaborated in Section III. The proposed algorithm is compared with EKF and the fully solved MHE via a simulated SOC-dependent ECM in Section IV. Section V summarizes the results of this paper.

II. SOC-DEPENDENT ECM AND PROBLEM STATEMENT

In this section, we introduce the SOC-dependent ECM of the LIB, and derive its discrete-time state-space representation.

A. SOC-dependent ECM

In this paper, we use the SOC-dependent ECM for the SOC estimation as shown in Fig. 1. In the Fig. 1, the open-circuit voltage V_{OC} represents battery electromotive force. R_0 is the ohmic internal resistance, which represents the battery ohmic characteristics. R_1 and C_1 are the polarization resistance and the polarization capacitor, respectively, which represent the hysteresis characteristics during the polarization effect of LIBs. I is the load current, whose value is positive or negative for discharging or charging, respectively. Z is the SOC of the LIB, ranging from 0 to 1. V_b is the terminal voltage when connected to the load. V_1 is RC circuit voltage.

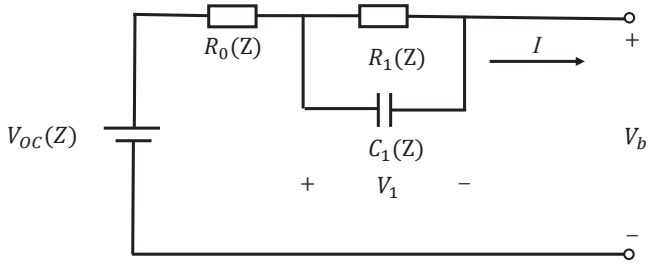

Fig. 1. SOC-dependent ECM for LIBs.

According to the circuit system theory [17], the discrete-time approximation of the ECM is expressed as:

$$Z_k = Z_{k-1} - \frac{I_{k-1}\Delta t}{C_n}, \tag{1}$$

$$V_{1,k} = V_{1,k-1}\exp\left(-\frac{\Delta t}{\tau_{1,k-1}}\right) + I_{k-1}R_{1,k-1}\left[1-\exp\left(-\frac{\Delta t}{\tau_{1,k-1}}\right)\right], \tag{2}$$

$$V_{b,k} = V_{OC}(Z_k) - V_{1,k} - I_k R_{0,k}, \tag{3}$$

where C_n is the nominal capacity of the battery, Z_k is the SOC of the LIB at the current discrete time t_k, I_k denotes the load current at the current discrete time t_k, Δt denotes the sampling interval, and $\tau_{1,k-1} = R_{1,k-1}C_{1,k-1}$ indicates the time constant of the RC circuit in the time interval $[t_{k-1},t_k]$.

In this paper, we assume constant temperature, thus only consider the ECM parameters V_{OC}, R_0, R_1, C_1 dependences on SOC, so we use polynomial functions to describe their nonlinear relationship with SOC:

$$V_{OC,k} = V_{OC}(Z_k) = \sum_{j=0}^{n}\alpha_j Z_k^j, \tag{4a}$$

$$R_{0,k} = R_0(Z_k) = \sum_{j=0}^{m}\beta_{1,j} Z_k^j, \tag{4b}$$

$$R_{1,k} = R_1(Z_k) = \sum_{j=0}^{m}\beta_{2,j} Z_k^j, \tag{4c}$$

$$C_{1,k} = C_1(Z_k) = \sum_{j=0}^{m}\beta_{3,j} Z_k^j. \tag{4d}$$

With the above description, the dynamic behavior of the LIB can be further expressed in the following standard discrete-time state-space model:

$$x_{k+1} = F(x_k, u_k) + w_k, \tag{5}$$

$$y_k = h(x_k, u_k) + v_k, \tag{6}$$

where $x_k = [Z_k, V_{1,k}]^\top$ is the state vector of the circuit at the moment of discrete time t_k; $u_k = I_k$ is the system input at time t_k; $y_k = V_{b,k}$ is the system output at time t_k; the process noise w_k and the measurement noise v_k are white and follow Gaussian distributions $N(0, Q_k)$ and $N(0, R_k)$, respectively.

$F(x_k, u_k)$ and $h(x_k, u_k)$ can be represented as:

$$F(x_k, u_k) = \begin{bmatrix} Z_k - \frac{I_k\Delta t}{C_n} \\ V_{1,k}\exp\left(-\frac{\Delta t}{\tau_{1,k}}\right) + I_k R_{1,k}\left[1-\exp\left(-\frac{\Delta t}{\tau_{1,k}}\right)\right] \end{bmatrix}, \tag{7}$$

$$h(x_k, u_k) = V_{OC}(Z_k) - V_{1,k} - I_k R_{0,k}. \tag{8}$$

So far, we have obtained the nonlinear state-space expression for the ECM of the LIB.

From the above analysis, the range of application of the ECM we have used is as follows: (1) it can meet the basic requirements of simulating the dynamic and static characteristics of the LIB; (2) the model structure is simple; (3) it is relatively easy to identify the parameters and verify the simulation; (4) the effect of temperature on the battery can be disregarded.

B. Problem statement

With the SOC-dependent ECM (5)-(8), we will exploit the MHE technique to estimate SOC from the measured terminal voltage and load current. Moreover, the fully solved MHE over each receding horizon as in [17] has a high computational time, which may not be allowed for a BMS that performs various computation tasks within a short sampling interval. As such, we need to develop a computationally efficient MHE algorithm to significantly reduce the computational cost without compromising its estimation performance compared to the fully solved MHE.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 09,2022 at 07:30:18 UTC from IEEE Xplore. Restrictions apply.

III. Real-time moving horizon estimation

In this section, we linearize the nonlinear programming problemby applying the Gauss-Newton method in the SQP strategy to obtain each QP subproblem, and solve each QP subproblem using an efficient structure. For real-time implementation, we only execute one Riccati iteration in each time interval.

A. Optimization problem and generalized Gauss-Newton SQP scheme

The optimization problem of the nonlinear system as shown in (5) and (6) can be described as the following quadratic optimization problem ($l = k - N + 1$):

$$\min_{\{x_i\}_{i=l}^{k},\{w_i\}_{i=l}^{k-1}} \ \frac{1}{2}\left\|x_l-\widehat{x}_l\right\|^2_{P_l^{-1}}+\frac{1}{2}\sum_{i=l}^{k-1}\left\|w_i\right\|^2_{Q_i^{-1}} \tag{9a}$$

$$+\frac{1}{2}\sum_{i=l}^{k}\left\|y_i-h(x_i,u_i)\right\|^2_{R_i^{-1}}$$

$$\text{s.t.}\ \ x_{i+1}=F(x_i,u_i)+w_i,\ i=l,...,k-1, \tag{9b}$$

where N denotes the moving horizon, and by introducing N, the dimensionality of the optimization problem is limited, which can reduce part of the computational effort, $\|s\|^2_{M^{-1}} = s^\top M^{-1} s$ denotes the weighted vector norm.

At each moment k, given the prior estimate $\hat{x}_l$, the input vector $\{u_l, ..., u_k\}$ and the output vector $\{y_l, ..., y_k\}$, the solution of the optimization problem can be denoted as $x_{l|k}, ..., x_{k|k}, w_{l|k}, ..., w_{k-1|k}$.

The first term of the objective function (9a) is usually called the arrival cost, which summarizes the information before the current estimation horizon. Similar to [24], the arrival cost term is updated as follows: we will use the optimal estimated value $x_{l|k-1}$ obtained by solving (9) on the previous horizon $[l-1, k-1]$ as the prior state estimate $\widehat{x}_l$ of the current horizon. It is worth noting that P, Q and R are respectively positive definite weighting matrices used to quantify the designer's confidence in the initial estimates, the dynamic system model, and the output model.

In order to obtain the equality constrained QP subproblem, the nonlinear unconstrained MHE problem (9) is first linearized at $\left\{\widehat{x}_{l|k-1}, ..., \widehat{x}_{k|k-1}\right\}$ and $\left\{\widehat{w}_{l|k-1}, ..., \widehat{w}_{k-1|k-1}\right\}$ in the Gaussian-Newton framework at the current horizon $[l, k]$. Specifically, the nonlinear functions F and h are approximated by a first-order Taylor expansion.

$$\min_{\{\Delta x_i\}_{i=l}^{k},\{\Delta w_i\}_{i=l}^{k-1}} \ \frac{1}{2}\left\|\Delta x_l\right\|^2_{P_l^{-1}}+\frac{1}{2}\sum_{i=l}^{k-1}\left\|r_{w,i}-\Delta w_i\right\|^2_{Q_i^{-1}} \tag{10a}$$

$$+\frac{1}{2}\sum_{i=l}^{k}\left\|r_{y,i}-C_i\Delta x_i\right\|^2_{R_i^{-1}}$$

$$\text{s.t.}\ \ \Delta x_{i+1}=f_i+A_i\Delta x_i+\Delta w_i, i=l,...,k-1, \tag{10b}$$

where

$$\Delta w_i = w_i - \widehat{w}_{i|k-1},\ r_{w,i} = -\widehat{w}_{i|k-1}, \tag{11a}$$

$$f_i = \widehat{w}_{i|k-1} - \widehat{x}_{i+1|k-1} + F(\widehat{x}_{i|k-1}, u_i), \tag{11b}$$

$$l \le i \le k-1,$$

$$\Delta x_i = x_i - \widehat{x}_{i|k-1},\ r_{y,i} = y_i - h(\widehat{x}_{i|k-1}, u_i), \tag{11c}$$

$$A_i = \nabla F(\widehat{x}_{i|k-1}, u_i),\ C_i = \nabla h(\widehat{x}_{i|k-1}, u_i), \tag{11d}$$

$$l \le i \le k.$$

The solution of the equality constrained QP is recorded as $\Delta x_l, ..., \Delta x_k, \Delta w_l, ..., \Delta w_{k-1}$. The solution of the unconstrained MHE problem (9) is updated as $x_{i|k} = \widehat{x}_{i|k-1} + \Delta x_i$ $(l \le i \le k)$ and $w_{i|k} = \widehat{w}_{i|k-1} + \Delta w_i$ $(l \le i \le k-1)$.

B. Solving the QP Subproblem

We know that for any optimization problem with differentiable objective function and constraint function, the optimal solution needs to satisfy the Karush-Kuhn-Tucker (KKT) condition. For the sake of simplicity, we will name the system dimensions from now on as $n = n_x$, $m = n_w$, and $p = n_y$. The Lagrange function of the equality constrained QP subproblem is described as follows:

$$\begin{aligned} L(\Delta x, \Delta w, \lambda) =& \frac{1}{2}\left\|\Delta x_l\right\|^2_{P_l^{-1}}+\frac{1}{2}\sum_{i=l}^{k-1}\left\|r_{w,i}-\Delta w_i\right\|^2_{Q_i^{-1}} \\ &+\frac{1}{2}\sum_{i=l}^{k}\left\|r_{y,i}-C_i\Delta x_i\right\|^2_{R_i^{-1}} \\ &+\sum_{i=l}^{k-1}\lambda_i(f_i+A_i\Delta x_i+\Delta w_i-\Delta x_{i+1}). \end{aligned} \tag{12}$$

We can write the KKT system as $M\xi = r$:

$$\begin{bmatrix} \Phi_l & \Gamma_l^\top & 0 & & & \\ \Gamma_l & 0 & -\gamma^\top & & & \\ & -\gamma & \Phi_{l+1} & \Gamma_{l+1}^\top & & \\ & & \ddots & \ddots & \ddots & \\ & & & \Gamma_{k-1} & 0 & -I \\ & & & & -I & \Phi_k \end{bmatrix} \begin{bmatrix} X_l \\ \lambda_l \\ X_{l+1} \\ \vdots \\ \lambda_{k-1} \\ X_N \end{bmatrix} = \begin{bmatrix} r_{d,l} \\ r_{p,l} \\ r_{d,l+1} \\ \vdots \\ r_{p,k-1} \\ r_{d,k} \end{bmatrix}, \tag{13}$$

where we define

$$X_i = \begin{bmatrix} \Delta x_i \\ \Delta w_i \end{bmatrix},\ X_k = \Delta x_k, \tag{14a}$$

$$\Phi_l = \begin{bmatrix} P_l^{-1} + C_l^\top R_l^{-1} C_l & \\ & Q_l^{-1} \end{bmatrix}, \tag{14b}$$

$$\Phi_i = \begin{bmatrix} C_i^\top R_i^{-1} C_i & \\ & Q_i^{-1} \end{bmatrix},\ \Phi_k = C_k^\top R_k^{-1} C_k, \tag{14c}$$

$$\Gamma_i = \begin{bmatrix} A_i & I \end{bmatrix},\ \gamma = \begin{bmatrix} I \\ 0 \end{bmatrix}, \tag{14d}$$

$$r_{d,i} = \begin{bmatrix} C_i^\top R_i^{-1} r_{y,i} \\ Q_i^{-1} r_{w,i} \end{bmatrix}, r_{d,k} = C_k^\top R_k^{-1} r_{y,k}, \tag{14e}$$

$$r_{p,i} = -f_i,\ l \le i \le k-1. \tag{14f}$$

Authorized licensed use limited to: TU Delft Library. Downloaded on May 09,2022 at 07:30:18 UTC from IEEE Xplore. Restrictions apply.

The KKT matrix M can be factorized by an LU decomposition [23] $M = LU$ with

$$L = \begin{bmatrix} \Sigma_{l+}^{-1} & & & & & & \\ \Gamma_l & -P_{l+1} & & & & & \\ & -\gamma & \Sigma_{l+1+}^{-1} & & & & \\ & & \ddots & \ddots & & & \\ & & & \Gamma_{k-1} & -P_k & & \\ & & & & -I & \Sigma_{k+}^{-1} \end{bmatrix}, \quad (15a)$$

$$U = \begin{bmatrix} I & \Sigma_{l+}\Gamma_l^\top & & & & & \\ & I & P_{l+1}^{-1}\gamma^\top & & & & \\ & & I & \Sigma_{l+1+}\Gamma_{l+1}^\top & & & \\ & & & \ddots & \ddots & & \\ & & & & I & P_k^{-1} & \\ & & & & & I & \end{bmatrix}. \quad (15b)$$

Where

$$\Sigma_i = \begin{bmatrix} P_i & \\ & Q_i \end{bmatrix}, \; D_i = \begin{bmatrix} C_i & 0 \end{bmatrix}, \quad (16a)$$

$$\begin{aligned} \Sigma_{i+} &= (\Sigma_i^{-1} + D_i^\top R_i^{-1} D_i^\top)^{-1} \\ &= \Sigma_i - \Sigma_i D_i^\top (R_i + D_i \Sigma_i D_i^\top)^{-1} D_i \Sigma_i, \end{aligned} \quad (16b)$$

$$P_{i+1} = \Gamma_i \Sigma_{i+} \Gamma_i^\top, \quad (16c)$$

$$\Sigma_k = P_k, D_k = C_k, \quad (16d)$$

$$\begin{aligned} \Sigma_{k+} &= (\Sigma_k^{-1} + D_k^\top R_k^{-1} D_k^\top)^{-1} \\ &= \Sigma_k - \Sigma_k D_k^\top (R_k + D_k \Sigma_k D_k^\top)^{-1} D_k \Sigma_k, \\ & l \le i \le k-1. \end{aligned} \quad (16e)$$

(16b) is the result of using Matrix Inversion Lemma [25].

The solution of the KKT system $M\xi = r$ can be divided into two parts after decomposing the KKT matrix M by (15), which are the direct forward vector recursion and the backward forward vector solution. So, we can solve $L\xi' = r$ with the direct forward vector recursion and solve $U\xi = \xi'$ with the backward forward vector solution.

We can easily see that the process of LU decomposition involves multiple iterations to find the inverse of the matrices P, which requires us to first determine whether the matrices are invertible in the computation process. In addition, during the forward and backward vector solving process, we also need to notice the change in the dimension of the matrix. This can be avoided by matrix calculation as formalized in next subsection.

C. KKT solution using modified LU factor-solve method

We further analyze the analytical solution of the optimization problem obtained by LU decomposition to avoid the inverse problem of matrix P in the iterative process and the computational dimensional changes during the forward and backward vector solving process. First, consider the forward vector solve, At time step $i = l$ we have that

$$\begin{aligned} X_l' &= \Sigma_{l+} r_{d,l} \\ &= (\Sigma_l^{-1} + D_l^\top R_l^{-1} D_l^\top)^{-1} (\Sigma_l^{-1} d_l + D_l^\top R_l^{-1} r_{y,l}) \\ &= d_l + \Sigma_l D_l^\top (R_l + D_l \Sigma_l D_l^\top)^{-1} (r_{y,l} - D_l d_l), \end{aligned} \quad (17)$$

where $d_l = \begin{bmatrix} 0 \\ r_{w,l} \end{bmatrix}$.

For $i = l \to k-1$, We can easily obtain:

$$\begin{aligned} \lambda_i' &= P_{i+1}^{-1}(-r_{p,i} + \Gamma_i X_i') \\ &= P_{i+1}^{-1}(f_i + A_i \Delta x_i' + \Delta w_i') \\ &= P_{i+1}^{-1} \Delta \widehat{x}_{i+1}' \\ &= P_{i+1}^{-1} \gamma^\top d_{i+1} \\ &= \gamma^\top \Sigma_{i+1}^{-1} d_{i+1}, \end{aligned} \quad (18)$$

where $d_{i+1} = \begin{bmatrix} \Delta \widehat{x}_{i+1}' \\ r_{w,i+1} \end{bmatrix}$,

$$\begin{aligned} X_{i+1}' &= \Sigma_{i+1+}(\gamma \lambda_i' + r_{d,i+1}) \\ &= \Sigma_{i+1+}(\Sigma_{i+1}^{-1} d_{i+1} + D_{i+1}^\top R_{i+1}^{-1} r_{y,i+1}) \\ &= d_{i+1} + \Sigma_{i+1} D_{i+1}^\top (R_{i+1} \\ &\quad + D_{i+1} \Sigma_{i+1} D_{i+1}^\top)^{-1} (r_{y,i+1} - D_{i+1} d_{i+1}). \end{aligned} \quad (19)$$

Second, consider the backward vector solve. Since $X_k = X_k'$, we only need to consider λ and perform further calculations and simplifications on it:

$$\begin{aligned} \lambda_{k-1} &= \lambda_{k-1}' - P_k^{-1} X_k \\ &= D_k^\top (R_k + D_k \Sigma_k D_k^\top)^{-1} (D_k d_k - r_{y,k}). \end{aligned} \quad (20)$$

Algorithm 1 real-time MHE algorithm

At time k, input: $\{u_l, ..., u_k\}$, $\{y_l, ..., y_k\}$.
Initialization:
$\widehat{x}_l = x_{l|k-1}$
for $i = l \to k-1$ **do**
  $\widehat{x}_{i|k-1} = x_{i|k-1}$, $\widehat{w}_{i|k-1} = w_{i|k-1}$
end for
$\widehat{x}_{k|k-1} = F(\widehat{x}_{k-1|k-1}, u_{k-1})$
Compute the QP subproblem (10) by equation (11).
Solve the QP subproblem with Algorithm 2.
Solution for the (10): $\{\Delta x_i\}_{i=l}^{k}, \{\Delta w_i\}_{i=l}^{k-1}$.
Update:
for $i = l \to k$ **do**
  $x_{i|k} = \widehat{x}_{i|k-1} + \Delta x_i$
  if $i < k$ **then**
    $w_{i|k} = \widehat{w}_{i|k-1} + \Delta w_i$
  end if
end for
$k + 1 \leftarrow k$
Output: $x_{k|k}$

Authorized licensed use limited to: TU Delft Library. Downloaded on May 09,2022 at 07:30:18 UTC from IEEE Xplore. Restrictions apply.

For $i = k-1 \to l$, we substitute (18) and $X_i = X_i' - \Sigma_{i+}\Gamma_i^\top \lambda_i$ into

$$\begin{aligned}\lambda_{i-1} &= \lambda_{i-1}' - P_i^{-1}\gamma^\top X_i \\ &= \gamma^\top D_i^\top (R_i + D_i\Sigma_i D_i^\top)^{-1}(D_i d_i - r_{y,i}) + \gamma^\top \Sigma_i^{-1}\Sigma_{i+}\Gamma_i^\top \lambda_i \\ &= A_i^\top \lambda_i + D_i^\top (R_i + D_i\Sigma_i D_i^\top)^{-1}[D_i(d_i - \Sigma_i\Gamma_i^\top\lambda_i) - r_{y,i}].\end{aligned} \tag{21}$$

Through the above analysis and calculation, we can not only avoid some matrix inversion problems, but also get some convenience in matrix calculation dimension. Inspired by [23], a Riccati-based algorithm is applied to solve the KKT system of the equality constrained QP subproblem (10), which is described in Algorithm 2.

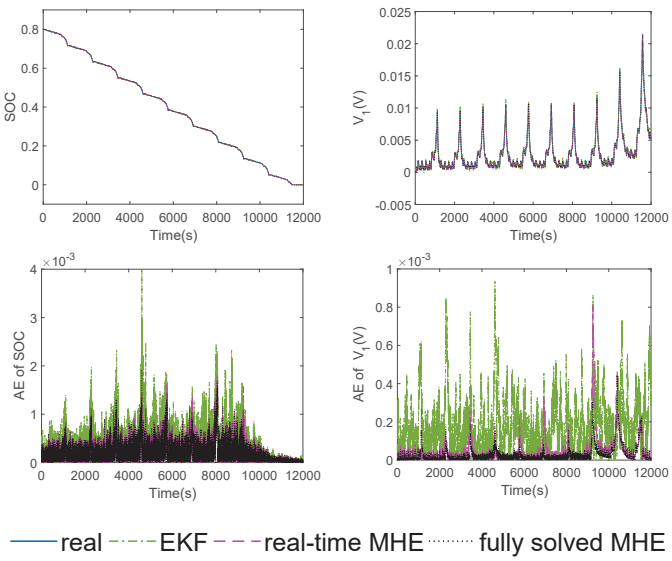

Fig. 2. Performance of different algorithms with an exact initial state guess.

IV. Simulation Results

In this section, we compare the conventional EKF, the fully solved MHE, and the proposed real-time MHE in the SOC estimation for a simulated SOC-dependent ECM. The simulation model data used in this paper are taken from [17]. The following root mean square error (RMSE) and absolute error (AE) are used to evaluate the estimation performance.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{k} |\widehat{\star}_i - \star_i|^2}{k}}, \tag{22}$$

$$\text{AE}_k = |\widehat{\star}_k - \star_k|, \tag{23}$$

where $\star$ represents Z or V_1.

We verify the effectiveness of the real-time MHE by analyzing the difference in effectiveness between the real-time MHE algorithm and the fully solved MHE algorithm and the traditional EKF algorithm for the presence or absence of bias in the initial estimation.

When there is no initial estimation bias, we use the covariance matrices are: $P = \text{diag}(1\times10^{-6}, 1\times10^{-8})$, $Q = \text{diag}(5\times10^{-8}, 1\times10^{-9})$, and $R = 1\times10^{-7}$. The matrix $\text{diag}(a,b)$ is a diagonal matrix with diagonal elements a and b. Table 1 shows the computing time and RMSE of different algorithms, and Fig. 2 shows the estimation results of different algorithms.

Algorithm 2 Solve the QP subproblem

Initialization
for $i = l \to k$ **do**
  $\Pi_i = C_i P_i$, $\Omega_i = C_i^\top (R_i + \Pi_i C_i^\top)^{-1}$
  $K_i = \Pi_i \Omega_i$, $P_i' = P_i - K_i \Pi_i$
  if $i < k$ **then**
    $P_{i+1} = A_i P_i' A_i^\top + Q_i$
  end if
end for
Direct forward vector recursion
for $i = l \to k$ **do**
  $\Delta x_i' = \Delta \widehat{x}_i' + K_i(r_{y,i} - C_i \Delta \widehat{X}_i')$
  if $i < k$ **then**
    $\Delta w_i' = r_{w,i}$
    $\Delta \widehat{x}_i' = f_i + A_i \Delta x_i' + \Delta w_i'$
  end if
end for
Backward forward vector solution
$\Delta x_k = \Delta x_k'$, $\lambda_{k-1} = -\Omega_i(r_{y,i} - C_i \Delta \widehat{X}_i')$
for $i = k-1 \to l$ **do**
  $\Delta w_i = \Delta w_i' - Q_i \lambda_i$
  $\Delta x_i = \Delta x_i' - P_i' A_i^\top \lambda_i$
  if $i > l$ **then**
    $\lambda_{i-1} = A_i^\top \lambda_i - \Omega_i(r_{y,i} - C_i \Delta \widehat{x}_i' + \Pi_i A_i^\top \lambda_i)$
  end if
end for
Output: $\{\Delta x_i\}_{i=l}^{k}, \{\Delta w_i\}_{i=l}^{k-1}, \{\lambda_i\}_{i=l}^{k-1}$.

When the initial estimation bias exists, we use the covariance matrices are: $P = \text{diag}(1\times10^{-8}, 1\times10^{-9})$, $Q = \text{diag}(1\times10^{-7}, 1\times10^{-9})$, and $R = 1\times10^{-7}$. Fig. 3 shows the estimation effects of different algorithms.

In order to ensure the relative fairness of the comparison, the tuning parameters used in different algorithms are the same. In addition, we only perform one Riccati-based iteration for the data in the current horizon to achieve real-time calculation when solving each QP subproblem.

TABLE I
Averaged computation time and RMSE of different algorithms

Algorithms	Averaged computation time	RMSE of SOC	RMSE of V_1
fully solved MHE	0.049 s	2.962×10^{-4}	5.232×10^{-5} V
real-time MHE	4.221×10^{-4} s	2.999×10^{-4}	8.532×10^{-5} V
EKF	2.533×10^{-5} s	5.586×10^{-4}	2.314×10^{-4} V

The results show that real-time MHE improves the computational accuracy compared with EKF, and the computational speed is greatly improved compared with MHE, so we can

Authorized licensed use limited to: TU Delft Library. Downloaded on May 09,2022 at 07:30:18 UTC from IEEE Xplore. Restrictions apply.

conclude that real-time MHE gets a good compromise between fully solved MHE and EKF, which ensures the estimation effect and saves the estimation time.

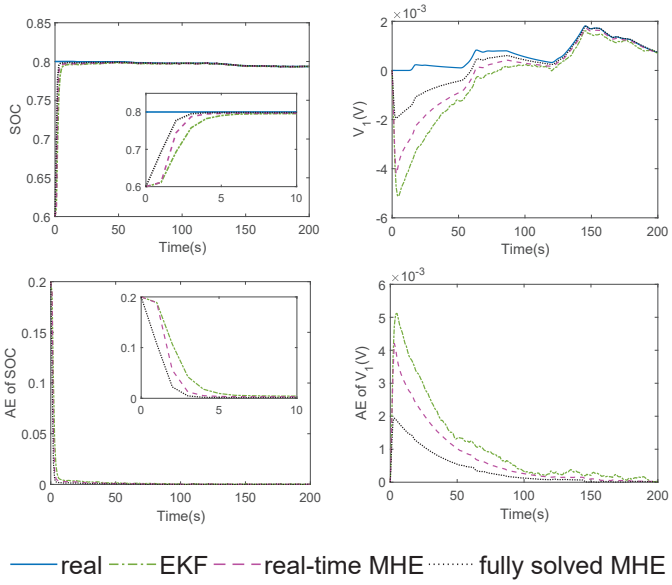

Fig. 3. Performance of different algorithms with an inexact initial state guess.

V. CONCLUSION

This paper adopt a real-time moving horizon estimation approach based on a nonlinear ECM to estimate SOC. The ECM parameters have polynomial dependence on SOC. Instead of fully solving the MHE over each receding horizon, a real-time algorithm is proposed by performing only one Gauss-Newton iteration, and the associated KKT system is solved by a structure-exploiting LU factorization. Compared with the traditional EKF and the fully solved MHE, the proposed real-time MHE offers a good compromise between estimation accuracy and computational cost. In future work, we will focus on adaptive tuning of the weight matrices for further performance improvement, and extend the proposed real-time MHE algorithm to cope with state constraints.

REFERENCES

[1] Z. Lao, B. Xia, W. Wang, W. Sun, Y. Lai, and M. Wang, "A novel method for lithium-ion battery online parameter identification based on variable forgetting factor recursive least squares," *Energies*, vol. 11, no. 6, p. 1358, 2018.

[2] X. Xu and N. Chen, "A state-space-based prognostics model for lithium-ion battery degradation," *Reliability Engineering & System Safety*, vol. 159, pp. 47–57, 2017.

[3] C. Fleischer, W. Waag, H. M. Heyn, and D. U. Sauer, "On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models part 2. parameter and state estimation," *Journal of Power Sources*, vol. 262, no. sep.15, pp. 457–482, 2014.

[4] A. Seaman, T.-S. Dao, and J. McPhee, "A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation," *Journal of Power Sources*, vol. 256, pp. 410–423, 2014.

[5] J. Xie, J. Ma, and K. Bai, "Enhanced coulomb counting method for state-of-charge estimation of lithium-ion batteries based on peukert's law and coulombic efficiency," *Journal of Power Electronics*, vol. 18, no. 3, pp. 910–922, 2018.

[6] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, 2014.

[7] M. T. Lawder, B. Suthar, P. W. Northrop, S. De, C. M. Hoff, O. Leitermann, M. L. Crow, S. Santhanagopalan, and V. R. Subramanian, "Battery energy storage system and battery management system for grid-scale applications," *Proceedings of the IEEE*, vol. 102, no. 6, pp. 1014–1030, 2014.

[8] X. Lin, Y. Kim, S. Mohan, J. B. Siegel, and A. G. Stefanopoulou, "Modeling and estimation for advanced battery management," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 393–426, 2019.

[9] D. N. How, M. Hannan, M. H. Lipu, and P. J. Ker, "State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review," *IEEE Access*, vol. 7, pp. 136 116–136 136, 2019.

[10] Y. Wang, H. Fang, L. Zhou, and T. Wada, "Revisiting the state-of-charge estimation for lithium-ion batteries: A methodical investigation of the extended Kalman filter approach," *IEEE control systems*, vol. 37, no. 4, pp. 73–96, 2017.

[11] M. Charkhgard and M. Farrokhi, "State-of-charge estimation for lithium-ion batteries using neural networks and EKF," *IEEE transactions on industrial electronics*, vol. 57, no. 12, pp. 4178–4187, 2010.

[12] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. modeling and identification," *Journal of power sources*, vol. 134, no. 2, pp. 262–276, 2004.

[13] F. Yang, Y. Xing, D. Wang, and K. L. Tsui, "A comparative study of three model-based algorithms for estimating state-of-charge of lithium-ion batteries under a new combined dynamic loading profile," *Applied energy*, vol. 164, pp. 387–399, 2016.

[14] H. He, H. Qin, X. Sun, and Y. Shui, "Comparison study on the battery SoC estimation with EKF and UKF algorithms," *Energies*, vol. 6, no. 10, pp. 5088–5100, 2013.

[15] D. G. Robertson, J. H. Lee, and J. B. Rawlings, "A moving horizon-based approach for least-squares estimation," *AIChE Journal*, vol. 42, no. 8, pp. 2209–2224, 1996.

[16] E. L. Haseltine and J. B. Rawlings, "Critical evaluation of extended Kalman filtering and moving-horizon estimation," *Industrial & engineering chemistry research*, vol. 44, no. 8, pp. 2451–2460, 2005.

[17] J. Shen, Y. He, Z. Ma, H. Luo, and Z. Zhang, "Online state of charge estimation of lithium-ion batteries: A moving horizon estimation approach," *Chemical Engineering Science*, vol. 154, pp. 42–53, 2016.

[18] M. J. Tenny and J. B. Rawlings, "Efficient moving horizon estimation and nonlinear model predictive control," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 6. IEEE, 2002, pp. 4475–4480.

[19] V. M. Zavala, C. D. Laird, and L. T. Biegler, "A fast moving horizon estimation algorithm based on nonlinear programming sensitivity," *Journal of Process Control*, vol. 18, no. 9, pp. 876–884, 2008.

[20] A. Alessandri and M. Gaggero, "Fast moving horizon state estimation for discrete-time systems using single and multi iteration descent methods," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4499–4511, 2017.

[21] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," in *Nonlinear model predictive control*. Springer, 2009, pp. 391–417.

[22] P. Kühl, M. Diehl, T. Kraus, J. P. Schlöder, and H. G. Bock, "A real-time algorithm for moving horizon state and parameter estimation," *Computers & chemical engineering*, vol. 35, no. 1, pp. 71–83, 2011.

[23] N. Haverbeke, "Efficient numerical methods for moving horizon estimation," *Diss., Katholieke Universiteit Leuven, Heverlee, Belgium*, 2011.

[24] A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753–1765, 2008.

[25] W. H. Press, H. William, S. A. Teukolsky, W. T. Vetterling, A. Saul, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 09,2022 at 07:30:18 UTC from IEEE Xplore. Restrictions apply.