Journal of Power Sources 331 (2016) 208–223

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

State-of-charge estimation in lithium-ion batteries: A particle filter approach

Aditya Tulsyan [a, *], Yiting Tsai [b], R. Bhushan Gopaluni [c], Richard D. Braatz [a]

[a] Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[b] Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[c] Department of Chemical and Biological Engineering & Institute of Applied Mathematics, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

HIGHLIGHTS

- We derive a state-space formulation of the standard pseudo two-dimensional model.
- We propose an iterative method to simulate the stochastic Li-ion battery model.
- We propose a novel algorithm to estimate all the states in the Li-ion battery.
- We estimate the state-of-charge of the battery using the proposed algorithm.
- This is the first ever nonlinear stochastic estimator for Li-ion batteries.

ARTICLE INFO

Article history:
Received 30 May 2016
Received in revised form
20 August 2016
Accepted 26 August 2016
Available online 16 September 2016

Keywords:
Li-ion batteries
Stochastic P2D model
Simulation
State estimation
State-of-charge estimation

ABSTRACT

The dynamics of lithium-ion batteries are complex and are often approximated by models consisting of partial differential equations (PDEs) relating the internal ionic concentrations and potentials. The *Pseudo two-dimensional model* (P2D) is one model that performs sufficiently accurately under various operating conditions and battery chemistries. Despite its widespread use for prediction, this model is too complex for standard estimation and control applications. This article presents an original algorithm for state-of-charge estimation using the P2D model. Partial differential equations are discretized using implicit stable algorithms and reformulated into a nonlinear state-space model. This discrete, high-dimensional model (consisting of tens to hundreds of states) contains implicit, nonlinear algebraic equations. The uncertainty in the model is characterized by additive Gaussian noise. By exploiting the special structure of the pseudo two-dimensional model, a novel particle filter algorithm that sweeps in time and spatial coordinates independently is developed. This algorithm circumvents the degeneracy problems associated with high-dimensional state estimation and avoids the repetitive solution of implicit equations by defining a 'tether' particle. The approach is illustrated through extensive simulations.

1. Introduction

Lithium-ion (Li-ion) batteries are prevalent due to their applications in a variety of low-power consumer gadgets, high-power automobiles and spacecrafts. Their widespread use is due to numerous desirable properties, such as high energy density, high efficiency, slow material degradation, lack of memory effect, low self-discharge, and minimal maintenance requirements. Despite their advantages, Li-ion batteries are susceptible to over-heating and explosions caused by overcharging and/or high coulomb rates. Disasters caused by overheated lithium batteries include the fiery destruction of laptops and commercial aircraft. Given the variety of applications that benefit from the use of Li-ion batteries, it is essential that they are operated safely and reliably under a wide range of load and weather conditions. To ensure safe operation, a method is needed to predict the onsets of overcharging and runaway temperatures. The State-of-Charge (SOC), which is a measure of the amount of charge remaining in a battery, is a property that can be used for this purpose.

The dynamics of a Li-ion battery are dictated by a set of complex equations that govern the electrochemical reaction kinetics and transport phenomena occurring within the battery. A typical battery model accounts for variations in the concentrations and potentials

* Corresponding author.
E-mail addresses: tulsyan@mit.edu (A. Tulsyan), yiting@interchange.ubc.ca (Y. Tsai), bhushan.gopaluni@ubc.ca (R.B. Gopaluni), braatz@mit.edu (R.D. Braatz).

http://dx.doi.org/10.1016/j.jpowsour.2016.08.113
0378-7753/

across the length of the battery. In most practical cases, other than the voltage and current, the concentrations and potentials within the battery are almost impossible to measure. Interestingly, the SOC is a direct function of the concentration of lithium ions, and not of voltage and current. Therefore, voltage and current measurements are uninformative with regards to SOC estimation. An obvious, alternate solution to this problem is to use a dynamic model.

Literature has demonstrated the modeling of Li-ion batteries at various levels of abstraction, ranging from simple empirical models (such as equivalent circuit models) to fully-coupled, complex models (such as molecular level). The Equivalent Circuit Models (ECM) approximates battery dynamics using analogous combinations of electrical resistors and capacitors, but disregards the physics and chemistry within the battery [1,2,3]. Therefore, ECMs perform poorly in cases where the physical and chemical effects become prevalent, such as during high discharge rates. A comparative study of different ECMs is presented in Ref. [4]. On the other hand, the Single Particle Model (SPM) reflects the physical and chemical properties of a battery by representing each electrode as a a single large particle. By doing so, the overall Li-ion intercalation process is appropriately approximated [5] (see Fig. 1(a) for an illustration of single particle model). SPMs consist of two algebraic equations and a diffusion equation based on Fick's second law. The model equations can be solved easily and require insignificant computational resources [6]. Finally, the Pseudo Two-Dimensional (P2D) Model is the most comprehensive fundamental model that is commonly used in the battery modeling literature. This model is based on porous electrode theory and provides implicit chemical and electrochemical relations between concentrations, potentials, and current [7,8]. Other complex models such as 3*D* thermal model, *P2D* stress model, population balance model and molecular dynamic models are also available [9,10,11]. The accuracy of these models roughly improves with their complexity; however, the more complex models are difficult to simulate faster than real time

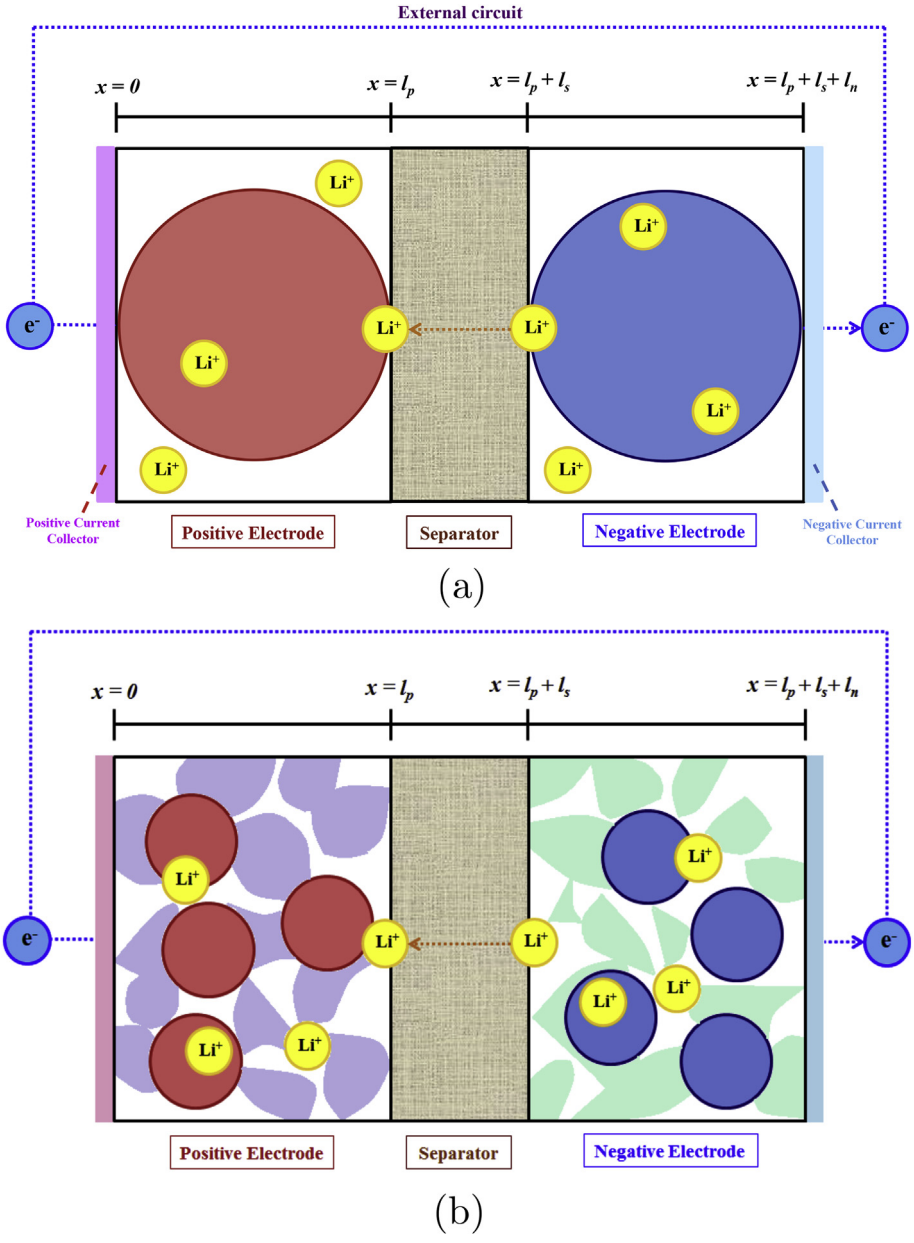

Fig. 1. An illustration of (a) standard Single Particle Model and (b) Pseudo Two-dimensional Model with a positive electrode (left side), a separator (center), and a negative electrode (right side) during a discharge cycle.

and have not been useful for estimation and control ([12]). On the other hand, simple models are not accurate and are weakly connected to the phenomena occurring in the battery. Therefore, it is important to strike a balance between the model accuracy and the real-time implementation speed so that estimation and control algorithms can be embedded in microprocessors with limited memory and processing power. It has been shown in practice that the pseudo two-dimensional (P2D) model, developed by Ref. [8], is accurate over a wide range of coulomb rates and for different battery chemistries. The P2D model is not as complex as molecular dynamic models, since size and morphology of the lithium battery material are not considered [13,14], but is still complex enough that its use in online battery estimation and control algorithms has been limited. In this paper, the P2D model is reformulated to enable the application of real-time estimation and control algorithms.

The P2D model consists of a set of coupled nonlinear partial differential equations (PDEs). A number of approximations of this model have been developed in literature to convert the model into a form that is suitable for analysis and to simulate the model at a fast rate for use in online applications. Most approximations of this model are either accurate at small coulomb rates or are too slow for online implementation ([12,6]). The only known high-speed implementation of the P2D model is developed by Ref. [15] using orthogonal collocation methods. However, this method is not suitable for direct application of standard estimation algorithms.

1.1. Current challenges

A variety of approximate models have been used for state and parameter estimation in Li-ion batteries ([16]). ECMs in conjunction with nonlinear observers are used in Refs. [17,18], and linear time-variant models are used in Ref. [19] for state estimation. The extended Kalman filter has been widely used on approximate models including linearized P2D and SP models ([20,21,22,23]). An adaptive extended Kalman filter and an adaptive nonlinear observer for an SP model are developed in Refs. [24] and [25]. Moreover [26], demonstrated the use of a nonlinear Luenberger observer. This observer was based on an approximation of the P2D model, with the important assumption of uniform concentration across the battery. Finally [27], designed an unscented Kalman filter based on an ordinary differential equation (ODE) approximation of the P2D model. These model and estimator approximations are accurate at low discharge or charge rates, but often exhibit severe inaccuracies at high discharge rates. Therefore, an estimator that can be applied on the original P2D model is preferred and expected to provide better estimates. Although intensive research efforts have been applied in the area of state estimation for P2D models, no known approach has been developed that can be applied to the P2D model without order-reducing model simplifications.

1.2. Contributions

In light of the previous arguments, this article will demonstrate a state estimation technique for P2D models that requires no model simplifications, except two minor ones. The method requires numerical approximations that are common in most discrete time-based approaches, as well as an approximation on solid concentrations. The P2D model is reformulated as a state-space model with linear, nonlinear, and algebraic states. Measurement and model uncertainties are characterized by stochastic noise. State estimation is accomplished by a modified particle filter that sweeps independently in time and spatial domains. This article is an expansion of our conference paper [28]. The proposed approach has three main advantages: (1) The estimator is developed for the highly accurate P2D model, (2) The partial differential equations (PDEs) can be solved very fast, and (3) The estimation algorithm is low-dimensional.

In summary, there are several articles written on the topic of state of charge estimation in Li-ion batteries spanning more than a decade. The differences in these articles are either due to the differences in models or due to the differences in the estimation algorithms employed to solve this problem. Naturally, the quality of the SOC estimates depends both on the quality of the model and the estimation algorithm. The literature is replete with algorithms to solve this problem with rather simplistic models that are not realistic in an operational setting. The P2D is one of the more sophisticated and accurate models that are available in the literature. However, to the best of our knowledge there are just a handful of articles that have used the P2D model and even those articles do not consider a stochastic model. It is not surprising that nobody has so far proposed using a stochastic P2D model because there are no off-the-shelf estimation algorithms that can be applied on a stochastic P2D model. We believe this is the very first article where a novel solution to the SOC estimation problem is proposed with one of the most sophisticated models for Li-ion batteries. The contributions of this paper are two-fold: 1) a realistic stochastic model is used, and 2) a novel estimation algorithm is developed.

2. Li-ion battery model

A typical intercalation Li-ion battery consists of three standard regions: a positive electrode, a separator, and a negative electrode. A thorough description of the various chemical, transport, and electrochemical phenomena that occur in a Li-ion battery can be found in Ref. [29]. The positive and negative electrodes contain electrolyte to transport the lithium ions and an active material that holds the lithium ions. The electrolyte and the active material are held together by fillers and other binding material. The separator contains the electrolyte, fillers and a binding material but no active material. The separator prevents the battery from short circuiting and allows transportation of positively charged lithium ions but blocks negatively charged electrons. During a discharge cycle, the lithium ions leave the active material in the negative electrode, travel through the separator with the help of the electrolyte, and then become deposited (or intercalated) in the active material in the positive electrode. The electrons leave the negative electrode, travel through an external circuit and react with lithium ions in the positive electrode as the ions are deposited on the active material (see Fig. 1(b) for a typical battery operation during a discharge cycle). This process reverses itself during a charging cycle.

2.1. Pseudo two-dimensional (P2D) model

A typical Li-ion battery dynamics is modeled by writing the conservation of mass and the conservation of charge equations on lithium ions. These equations can be written using the electrolyte concentration $c_{e,i}(x,t) \in \mathbb{R}^+$, the electrolyte potential $\Phi_{e,i}(x,t) \in \mathbb{R}$, the active material potential (also called the *solid potential*) $\Phi_{s,i}(x,t) \in \mathbb{R}$, and the concentration of lithiu, ions in the spherical particles of the active material $c_{s,i}(x,r(x),t) \in \mathbb{R}^+$, where $x \in \mathbb{R}$ denotes the one-dimensional spatial direction along which the lithium ions are transported, $t \in \mathbb{R}^+$ is the time, and $r(x) \in \mathbb{R}^+$ is the radial distance within an active particle at location x. Also, the index $i = \{p,s,n\}$ indicates the region of the battery − positive electrode ($i = p$), separator ($i = s$), and negative electrode ($i = n$). The standard P2D model can be derived using concentrated solution and porous electrode theories [7,30,31,8]. The original P2D model, as proposed in Ref. [30], consists of two PDEs for diffusion of lithium ions through the electrolyte and through the active particles. The original model has been slightly modified by approximating the diffusion equations corresponding to active material in Ref. [32]. This

Table 1
A modified pseudo two-dimensional (P2D) model for a standard Li-ion battery. The positive electrode extends from $x = 0 (n = 1)$ to $x = l_p (n = N_p)$, separator up to $x = l_p + l_s (n = N_p + N_s)$, and negative electrode up to $x = l_p + l_s + l_n (n = N_p + N_s + N_n)$.

Conservation equations	Boundary conditions			
	$x = 0$	$x = l_p$	$x = l_p + l_s$	$x = l_p + l_s + l_n$
Mass				
$(M1)\ \varepsilon_i \frac{\partial c_{e,i}}{\partial t} = D_i \left(\frac{\partial^2 c_{e,i}}{\partial x^2}\right) + a_i(1 - t_+) j_i$	$\frac{\partial c_{e,p}}{\partial x} = 0$	$-D_p \frac{\partial c_{e,p}}{\partial x} = -D_s \frac{\partial c_{e,s}}{\partial x}$	$-D_s \frac{\partial c_{e,s}}{\partial x} = -D_n \frac{\partial c_{e,n}}{\partial x}$	$\frac{\partial c_{e,n}}{\partial x} = 0$
$(M2)\ \frac{\partial \bar{c}_{s,i}}{\partial t} = -3 \frac{j_i}{R_i}$				
$(M3)\ c^*_{s,i} - \bar{c}_{s,i} = -\frac{R_i}{D_{s,i}} \frac{j_i}{5}$				
Charge				
$(C1)\ \kappa_i \frac{\partial \Phi_{e,i}}{\partial x} = -i_e + \frac{2\kappa_i RT}{F}(1 - t_+) \frac{\partial \ln c_{e,i}}{\partial x}$	$\frac{\partial \Phi_{e,p}}{\partial x} = 0$	$-\kappa_p \frac{\partial \Phi_{e,p}}{\partial x} = -\kappa_s \frac{\partial \Phi_{e,s}}{\partial x}$	$-\kappa_s \frac{\partial \Phi_{e,s}}{\partial x} = -\kappa_n \frac{\partial \Phi_{e,n}}{\partial x}$	$\frac{\partial \Phi_{e,n}}{\partial x} = 0$
$(C2)\ \frac{\partial i_{e,i}}{\partial x} = a_i F j_i$	$i_{e,p} = 0$	$i_{e,p} = I$	$i_{e,n} = I$	$i_{e,n} = 0$
$(C3)\ \frac{\partial \Phi_{s,i}}{\partial x} = \frac{i_{e,i} - I}{\sigma_i}$				

modified P2D model and the corresponding boundary conditions are shown in Table 1 ([32,15]). Using index $i \in \{p,s,n\}$ to denote positive electrode, separator, and negative electrodes: $\varepsilon_i \in (0,1)$ and $D_i \in \mathbb{R}^+$ are the porosity and the effective diffusion coefficient in section i, respectively; $\bar{c}_{s,i}(x,t) \in \mathbb{R}^+$ is the average lithium ion concentration in the solid particles; $c^*_{s,i}(x,t) \in \mathbb{R}^+$ is the surface concentration of the solid particles; $a_i \in \mathbb{R}^+$ is the ratio between the particle surface area to its volume; $t_+ \in (0,1)$ is the transference number; $R_i \in \mathbb{R}^+$ is the radius of the particle; $i_{e,i}(x,t) \in \mathbb{R}$ is the current in the electrolyte; $\kappa_i(x) \in \mathbb{R}^+$ is the conductivity of electrolyte; R is the universal gas constant; $T \in \mathbb{R}^+$ is the temperature; F is the Faraday constant; $I(t) \in \mathbb{R}^+$ is the current applied to the battery; $\sigma_i \in \mathbb{R}^+$ is the effective solid-phase conductivity; and $j_i(x,t) \in \mathbb{R}$ is the lithium ion flux at the surface of solid particles. Finally, in the modified P2D model, it is assumed that the flux and other concentrations only vary in the x-direction and are constant in the y and z directions. The list of all the parameters in the P2D model and their values are provided in Table 2.

The modified P2D model consists of two coupled PDEs and an algebraic equation that define the conservation of mass in each of the three sections of the battery (see $(M1)$ $(M2)$, and $(M3)$ in Table 1). The conservation equation $(M1)$ is obtained using the concentration solution theory and material balances. The boundary conditions for $(M1)$ has the electrolyte concentration flux being zero at the boundary between the electrode and the environment and equal to the flux on the side of the separator at the corresponding boundary. Equations $(M2)$ and $(M3)$ are simplifications of the Fick's diffusion equation for lithium ions in the active material in each electrode [32]. These equations relate lithium ion flux (j_i) to the average ($\bar{c}_{s,i}$) and surface ($c^*_{s,i}$) concentrations of the lithium ions in the active material. The modified P2D model also consists of three PDEs that define the conservation of charge in the electrodes and the separator (see $C1$, $C2$ and $C3$ in Table 1). The charge conservation equations are derived using porous electrode theory [7]. Equation $(C1)$ provides an expression for the electrolyte current and the boundary conditions ensure that the current is zero outside the battery and equal on both sides of the boundary between the electrodes and the separator. Equation $(C2)$ provides a relation between the electrolyte current and the lithium ion flux through the active

Table 2
List of variables and parameters used in the P2D model in Table 1. Unless otherwise noted, all constants used for the positive and negative electrodes and separator are from Ref. [15]. Note that U_i and κ_i are variables defined as a function of the electrolyte concentration [15].

Symbol	Definition	Values			Units
		Positive	Separator	Negative	
		$(i = p)$	$(i = s)$	$(i = n)$	
Constants					
a_i	Particle surface area to volume ratio	885000		723600	M^2/m^3
c_i^{max}	Maximum lithium concentration in solid particles	51554		30555	mol/m^3
D_i	Diffusivity	1.65×10^{-11}	2.06×10^{-10}	4.15×10^{-11}	M^2/s
$D_{s,i}$	Solid phase diffusivity	1.0×10^{-14}		3.9×10^{-14}	M^2/s
ε_i	Porosity	0.385	0.724	0.485	–
F	Faraday's constant		96487		C/mol
k_i	Reaction rate constant	2.334×10^{-11}		5.034×10^{-11}	$M^{2.5}/mol^{0.5}/s$
l_i	Length of region	80×10^{-6}	25×10^{-6}	88×10^{-6}	m
R	Universal gas constant		8.314		J/mol/K
R_i	Particle radius	2.0×10^{-6}		2.0×10^{-6}	m
σ_i	Effective solid-phase conductivity	100		100	S/m
T	Temperature	298.15	298.15	298.15	K
t_+	Transference number		0.364		–
Variables					
$c_{e,i}$	Electrolyte concentration				mol/m^3
$\bar{c}_{s,i}$	Average lithium concentration in solid particles				mol/m^3
$c^*_{s,i}$	Surface lithium concentration on solid particles				mol/m^3
j_i	Lithium ion flux				$mol/m^2/s$
I	Current applied to battery				A/m^2
$i_{e,i}$	Current in electrolyte				A/m^2
$\Phi_{e,i}$	Electric potential in electrolyte				V
$\Phi_{s,i}$	Electric potential in solid particles				V
U_i	Open-circuit voltage		[15]		V
κ_i	Liquid phase conductivity		[15]		S/m

material. The boundary conditions on ($C2$) ensure that the electrolyte current is zero outside the battery and equal to the applied current I in the separator. Equation ($C3$) is Ohm's law relating the solid potential to the electrolyte current and the applied current. In Table 1, $j_i(x,t)$ is the approximate flux of lithium ions across the surface of the active material at distance x and time t. It is a function of the spatial and temporal coordinates and is given by Butler-Volmer kinetics [33]:

$$j_i = 2k_i\left[c_{e,i}c^*_{s,i}\left(c_i^{\max}-c^*_{s,i}\right)\right]^{0.5}\sinh\left(\frac{0.5F}{RT}\left(\Phi_{s,i}-\Phi_{e,i}-U_i\right)\right), \quad (1)$$

where: k_i is the kinetic reaction constant; $c_i^{\max}\in\mathbb{R}_+$ is the maximum possible lithium ion concentration in the solid particles; $U_i\in\mathbb{R}$ is the open-circuit voltage; and $i=\{p,s,n\}$ is an index variable denoting the section of the battery. Here p, s, and n correspond to positive electrode, separator and negative electrode, respectively. The fluxes j_p and j_n in the electrodes are typically non-zero, while the flux j_s in the separator is always zero. This is because the separator does not include any active material.

3. Simulation of P2D model

Developing an efficient, fast and a reliable numerical solution for the complex nonlinear PDEs in Table 1 is a non-trivial exercise. In Ref. [8], the PDEs were iteratively solved using a first-order Taylor series approximation, and later in Refs. [34,15] the PDEs were solved using a coordinate transformation followed by collocation methods. Since 1995, numerous numerical solutions have been proposed to efficiently solve the PDEs in Table 1; however, these methods do not lend themselves to the state-space model (SSM) form suitable for estimation and control algorithms.

3.1. Proposed numerical solution

In this paper, a numerical approach is developed reformulates the numerical solution to the modified P2D model into a SSM representation. The proposed numerical solution is based on the following important observations about the modified P2D model in Table 1. The PDEs and their corresponding boundary conditions are nonlinear and highly coupled and require advanced numerical techniques to solve ($M1$). and ($C1$) have two Neumann boundary conditions in each section and therefore, explicit numerical approximations can become unstable ($C3$). does not have any explicit boundary conditions but are implicitly enforced through ($C2$) ($C2$). has two boundary conditions in each section despite being a first-order PDE. These boundary conditions are enforced by finding a suitable solid potential Φ_s and ionic flux j_i. The proposed numerical method divides the time and spatial coordinates into M equal time grid points and N_i equal spatial grid points, where $i=\{p,s,n\}$ as shown in Fig. 2. The concentrations, potentials, and other variables at time and space coordinates (t,x) are denoted by the corresponding discrete coordinate (m,n) where m denotes discrete time and n denotes discrete spatial location.[1] The bold-faced variables are obtained by concatenating their corresponding values at all the spatial locations. For instance, $\mathbf{c}_{e,i}(m)\equiv[c_{e,i}(m,1),\, c_{e,i}(m,2),\cdots,c_{e,i}(m,N_i)]^{\mathrm{T}}$. The Crank-Nicolson approximation of ($M1$) leads to

$$\mathbf{c}_{e,i}(m) = \mathbf{A}_{c,i}\mathbf{c}_{e,i}(m-1) + \mathbf{B}_{c,i}\mathbf{j}_i(m), \quad (2)$$

where $\mathbf{A}_{c,i}$ and $\mathbf{B}_{c,i}$ are constant matrices of appropriate dimensions.

[1] Note that the same notation is used for continuous and discrete variables, with the meaning being clear from the context.

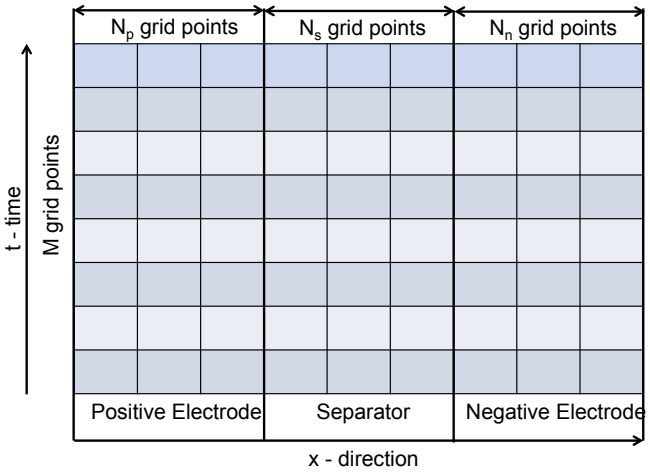

Fig. 2. The time-space discretization grid used in solving the PDEs.

Observe that the constants $\mathbf{A}_{c,i}$ and $\mathbf{B}_{c,i}$ depend on the index $i\in\{p,s,n\}$. The boundary conditions for ($M1$) are numerically approximated using second-order discretization methods and the corresponding equality constraints are included in (2). Observe that in (2), $\mathbf{c}_{e,i}$ at time m can be expressed as a function of $\mathbf{c}_{e,i}$ at $m-1$ and j_i at m. This makes (2) implicit in $\mathbf{c}_{e,i}$, since j_i itself depends on $\mathbf{c}_{e,i}$ (see (1)). For the sake of brevity, the derivation of (2) is not shown here, but can be easily derived. Similarly, using first-order implicit numerical approximation, ($C2$) and ($C3$) can be written as follows

$$\begin{bmatrix}\boldsymbol{\Phi}_{s,i}(m)\\ \mathbf{i}_{e,i}(m)\end{bmatrix} = \mathbf{A}_{\Phi,i}\mathbf{j}_i(m) + \mathbf{B}_{\Phi,i}I(m), \quad (3)$$

where $\mathbf{A}_{\Phi,i}$ and $\mathbf{B}_{\Phi,i}$ are constant matrices of appropriate dimensions. As in (2), the boundary conditions on ($C2$) are included in (3). Observe that although ($C3$) does not have any defined boundary conditions, ($C2$) and ($C3$) together are well-posed, and can be solved uniquely using (3) for any assumed boundary conditions.

The PDE in ($C1$) is similarly approximated using first-order implicit equations to obtain

$$\boldsymbol{\Phi}_{e,i}(m) = \mathcal{F}_\Phi\left(\mathbf{i}_{e,i}(m),\mathbf{c}_{e,i}(m)\right), \quad (4)$$

where $\mathcal{F}_\Phi$ is an appropriate nonlinear function of $\mathbf{i}_{e,i}$ and $\mathbf{c}_{e,i}$. The PDE ($M2$) and the algebraic equation ($M3$) can similarly be discretized and expressed as

$$\bar{\mathbf{c}}_{s,i}(m) = \bar{\mathbf{c}}_{s,i}(m-1) + \mathbf{B}_{\mathrm{avg},i}\mathbf{j}_i(m), \quad (5)$$

$$\mathbf{c}^*_{s,i}(m) = \bar{\mathbf{c}}_{s,i}(m) + \mathbf{B}_{\mathrm{surf},i}\mathbf{j}_i(m), \quad (6)$$

with constant matrices $\mathbf{B}_{\mathrm{avg},i}$ and $\mathbf{B}_{\mathrm{surf},i}$. Both (5) and (6) are implicit equations due to $\mathbf{j}_i$, which is given as

$$\mathbf{j}_i(m) = \mathcal{F}_j\left(\mathbf{c}_{e,i}(m),\mathbf{c}^*_{s,i}(m),\mathbf{i}_{e,i}(m),\boldsymbol{\Phi}_{s,i}(m),\boldsymbol{\Phi}_{e,i}(m)\right), \quad (7)$$

where $\mathcal{F}_j$ is an appropriate nonlinear function. Finally, the modified P2D model in Table 1 is represented by its discrete approximation through (2)–(6). The implementation of the proposed numerical solution is discussed next.

3.2. Implementation

The implementation of the proposed numerical solution is based on the following observations in Table 1:

(*i*) equations (*M*1), (*M*2), (*M*3), and (*C*2) are linear PDEs once j_i is known;

potentials must be accurately estimated to ensure that the corresponding flux is physically realizable and meaningful. Note that the

Algorithm 1 Deterministic Li-ion battery simulation

Inputs: $\Phi_{s,p}(1,1)$, $\Phi_{s,n}(1,1)$ and $\mathbf{c}_{e,i}(1)$, $\bar{\mathbf{c}}_{s,i}(1)$, $\mathbf{j}_i(1)$ for $i=\{p,s,n\}$, and $I(m)$ for $m=1,2,\ldots,M$
Initialize: δ_1, δ_2, $\mathbf{w}$, $\mathbf{i}_{e,i}(m)$ for $m=1,2,\ldots,M$ and for $i=\{p,s,n\}$.
for $m=1$ to M **do** ▷ simulate for M time points
  $\mathbf{j}(m) \leftarrow [\mathbf{j}_p(m), \mathbf{j}_s(m), \mathbf{j}_n(m)]^{\mathrm{T}}$.
  if $m \geq 2$ **then**
    $\mathbf{j}(m) = \mathbf{j}(m-1)$. ▷ initialize flux
  end if
  $\mathbf{i}_{\mathrm{bc}} \leftarrow \begin{bmatrix} i_{e,p}(m,N_p) - I(m) \\ i_{e,n}(m,1) - 0 \end{bmatrix}$. ▷ error in BC of (C2)
  $[i_{\mathrm{bc},1}, i_{\mathrm{bc},1}]^{\mathrm{T}} \leftarrow \mathbf{i}_{\mathrm{bc}}$.
  while $(|i_{bc}(1)| \geq \delta_1$ and $|i_{bc}(2)| \geq \delta_2)$ **do**
    Solve: (2) and (5) for $i=\{p,s,n\}$.
    $\kappa_i \leftarrow \kappa_i(\mathbf{c}_{e,i}(m))$ for $i=\{p,s,n\}$.
    Solve: (3), (4) and (6) for $i=\{p,s,n\}$.
    $\mathbf{j}_{\mathrm{temp},i} \leftarrow$ Solve (1) for $i=\{p,s,n\}$.
    $\mathbf{j}_{\mathrm{temp}} \leftarrow [\mathbf{j}_{\mathrm{temp},p}, \mathbf{j}_{\mathrm{temp},s}, \mathbf{j}_{\mathrm{temp},n}]^{\mathrm{T}}$.
    $\begin{bmatrix} \mathbf{j}(m) \\ \Phi_{s,p}(m,1) \\ \Phi_{s,n}(m,1) \end{bmatrix} \leftarrow \operatorname{argmin} J\big(\mathbf{j}(m), \mathbf{j}_{\mathrm{temp}}, \mathbf{i}_{bc}\big)$.
  end while
  Solve: (2) and (5) for $i=\{p,s,n\}$. ▷ update value
end for
return

(*ii*) equation (*C*3) does not have any specified boundary conditions, and therefore, initial conditions for $\mathbf{\Phi}_{s,p}$ and $\mathbf{\Phi}_{s,n}$ need to be guessed; and
(*iii*) the first-order PDE in (*C*2) can be treated as an initial value problem and solved iteratively until both the boundary conditions are satisfied (this is a form of the shooting method to solve ordinary differential equations with two boundary conditions).

The PDEs in Table 1 are solved by starting with an initial guess for $\mathbf{j}_i(m)$. Then $\mathbf{c}_{e,i}(m)$ is obtained from (2), and $\bar{\mathbf{c}}_{s,i}(m)$ and $\mathbf{c}^*_{s,i}(m)$ are obtained from (5) and (6), respectively. Since the PDE (*C*3) does not have specified boundary conditions, the potential at the interface of current collector/positive electrode and the interface of separator/negative electrode denoted by $\Phi_{s,p}(m,1)$ and $\Phi_{s,n}(m,1)$ in the discretized-space, respectively, are initially guessed and later updated based on an optimization algorithm. Equation (*C*2) is solved recursively as an initial value problem and $\mathbf{j}_i(m)$ is updated using an optimization algorithm until all of the boundary conditions for (*C*2) are satisfied. This procedure is shown in Algorithm 1 in the form of a pseudocode. In Algorithm 1, *J* is the objective function defined as

$$J \equiv w_1 ||\mathbf{j}_i - \mathbf{j}_{\mathrm{temp}}||_2 + w_2 |i_{bc,1}| + w_3 |i_{bc,2}|, \tag{8}$$

where $\|\cdot\|_2$ is the two-norm, $|\cdot|$ is an absolute value function, and $\mathbf{w} = [w_1, w_2, w_3]^{\mathrm{T}}$ are the user-defined weights in the optimization problem. Typically, w_2, w_3 are set to some large value to ensure accurate estimation of $\Phi_{s,p}(m,1)$ and $\Phi_{s,n}(m,1)$. This is because accurate estimation of $\Phi_{s,i}$ is crucial since in (1), as *F* is large (see Table 1), the hyperbolic sine function is sensitive to small changes in $\Phi_{s,i}$, which in turn result in large fluctuations in j_i. Therefore, the optimization problem in Algorithm 1 is nonlinear, and possibly non-convex thereby requiring an efficient optimization method.

Example 1. (*Deterministic P2D Model Simulation*) The modified P2D Li-ion battery model is simulated using Algorithm 1 at a constant galvanostatic discharge current of $I(m) = -30$ A/m^2 at a discharge rate of 1.0 C. The *x*-direction is discretized with $N_p = N_n = 52$ and $N_s = 50$. The initial guesses for the solid potential at its boundary conditions are chosen to be $\Phi_{s,p}(1,1) = 4.116$ V and $\Phi_{s,n}(1,1) = 0.074$ V, which were obtained through a trial-and-error process for quick initialization. Note that rational but arbitrary initialization of these boundary conditions can generate extremely large lithium ion fluxes leading to divergence of the proposed algorithm—in fact, no algorithm will converge under this scenario. The initial electrolyte concentration, $\mathbf{c}_{e,i}(1,\boldsymbol{\theta})$ is assumed to be uniform across the battery length at 1000 mol/L. The initial average lithium ion concentration in solid particles, $\bar{\mathbf{c}}_{s,i}(1,\boldsymbol{\theta})$ is assumed to be uniform at 2.554×10^4 mol/m^3 in the positive electrode, 0 mol/m^3 in the separator, and 2.612×10^4 mol/m^3 in the negative electrode. The initial lithium ion flux, $\mathbf{j}_i(1)$ was assumed to be in the order of magnitude 10^{-6} mol/s-m^2 in both positive and negative electrodes. The rest of the model parameters and empirical diffusion and conductivity relations are taken from Ref. [15]. The tolerance parameters are set to $\delta_1 = \delta_2 = 0.1$ and weights to $\mathbf{w} = [1,1,1]^{\mathrm{T}}$. The SNOPT Optimization Suite [35] is used to solve the optimization in Algorithm 1. The proposed algorithm runs through the discharge cycle of about 3500 s with cut-off voltage of 2.6 V.

Fig. 3 shows the evolution of concentrations, potentials, flux and current with time in the three sections of the battery during a discharge cycle. Fig. 3(a) shows the electrolyte concentration increasing across the length of the battery from the positive electrode to the negative electrode. Fig. 3(b) shows average

lithium ion concentration in solid particles during a discharge cycle. The lithium metal stored in the negative electrode comes out of the active solid particle and reacts at the surface to produce lithium ions. As more lithium ion are released at the negative electrode and absorbed in the active material in the positive electrode, the average concentration of lithium ions in the solid

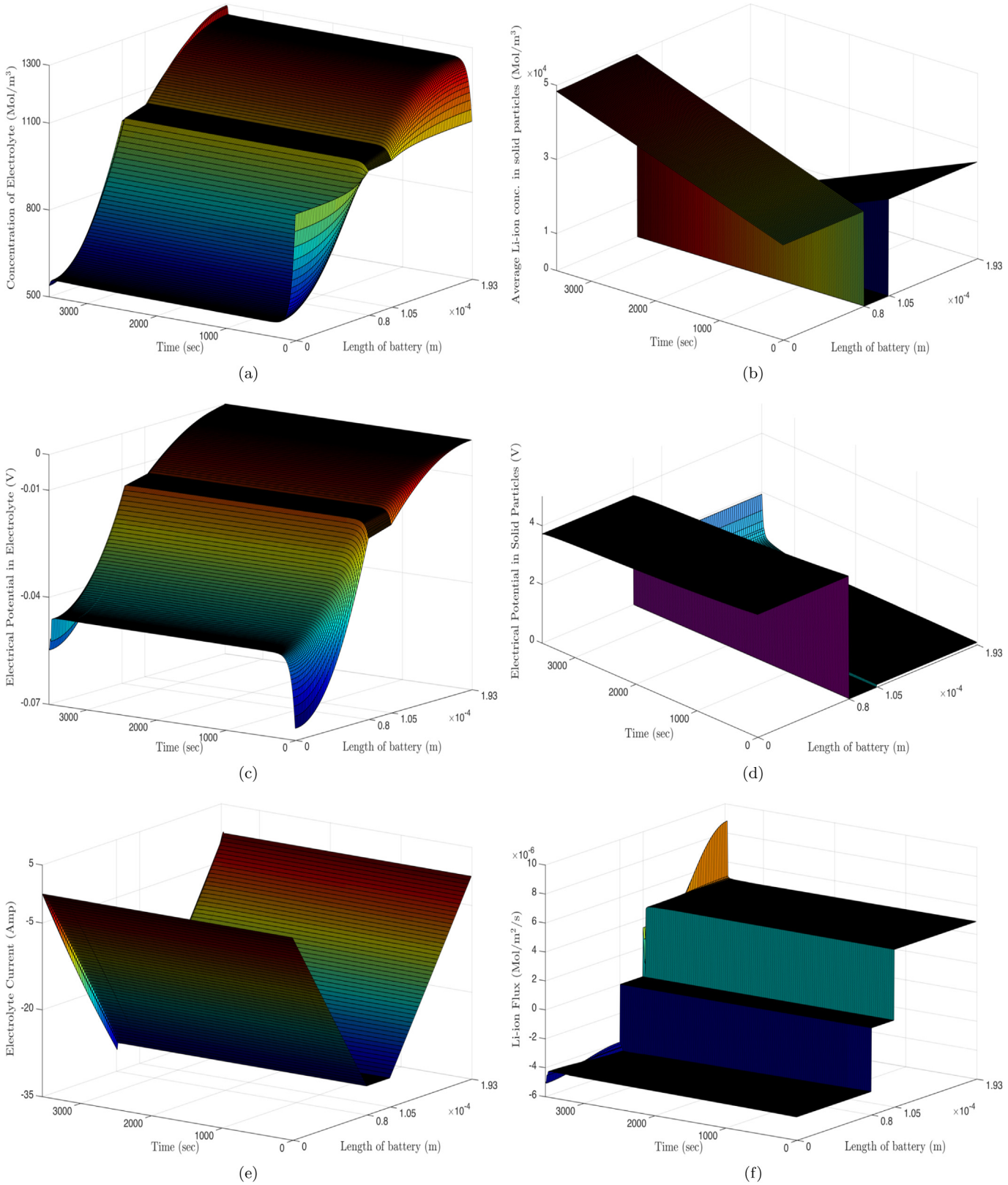

Fig. 3. Simulation of the deterministic P2D model using Algorithm 1. The subfigures show the dynamics of (a) concentration of electrolyte; (b) average lithium ion concentration in solid particles; (c) electrical potential in electrolyte; (d) electrical potential in solid particles; (e) current in the electrolyte; and (f) lithium ion flux at different times along the length of the battery. The ticks along the battery length correspond to the length of the positive electrode, separator and negative electrode.

particles in the positive electrode increases with discharge time. Fig. 3(c) through 3(f) show other states of the Li-ion battery. Fig. 4 shows discharge curves for the lithium ion battery at discharge rates of 1.0 C and 1.5 C.

4. Stochastic P2D state space model (SSM)

The P2D model in Table 1 is based on a number of theoretical and experimental approximations. It is natural to expect that the P2D model simulation in Algorithm 1 will only approximately represent the *exact* dynamics of a Li-ion battery. The discrepancy between the model predictions and the true dynamics of the battery is commonly referred to as the *model uncertainty*. An efficient approach to address model uncertainty is to introduce stochasticity or randomness in Algorithm 1 that accounts for all the Li-ion battery dynamics not captured by the P2D model.

In this section, the procedure to reformulate the P2D model simulation in Algorithm 1 into a set of discrete-time stochastic nonlinear state-space and algebraic equations is discussed. This reformulation not only accounts for model uncertainty, but also yields a representation that enables application of the proposed simulation model in real-time estimation, control and optimization. This is done as follows. For $i=\{p,s,n\}$, define the five classes of variables: (*i*) $\mathbf{x}^{l}_{t,i}\equiv[\mathbf{x}^{l,1}_{t,i},\mathbf{x}^{l,2}_{t,i}]^{\mathrm{T}}=[\mathbf{c}_{e,i}(t),\bar{\mathbf{c}}_{s,i}(t)]^{\mathrm{T}}$ are states that evolve linearly in time (*ii*); $\mathbf{x}^{a_1}_{t,i}\equiv[\mathbf{x}^{a_1,1}_{t,i},\mathbf{x}^{a_1,2}_{t,i}]^{\mathrm{T}}=[\mathbf{\Phi}_{s,i}(t),\mathbf{i}_{e,i}(t)]^{\mathrm{T}}$ and $\mathbf{x}^{a_2}_{t,i}\equiv\mathbf{c}^{*}_{s,i}(t)$ are linear algebraic states (*iii*); $\mathbf{x}^{a_3}_{t,i}\equiv\mathbf{\Phi}_{e,i}(t)$ are nonlinear algebraic states (*iv*); $\mathbf{x}^{n}_{t,i}\equiv\mathbf{j}_i(t)$ are nonlinear algebraic states; and (*v*) $u_t\equiv I(t)$ is the exogenous input (or manipulated variable). With this new notation, the P2D model simulation discussed in Section 3.1 can be represented as follows:

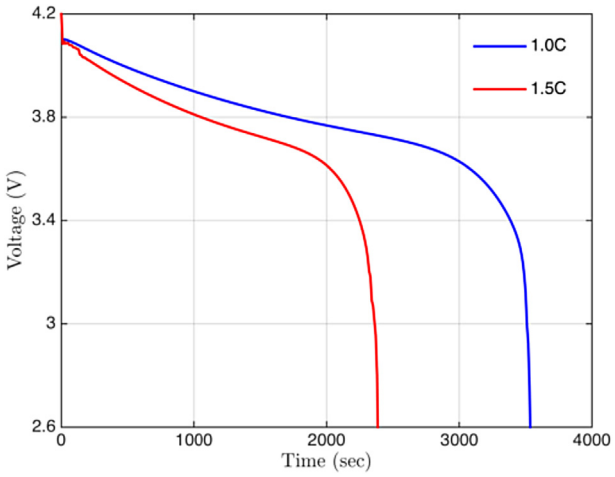

Fig. 4. Discharge curves for higher discharge rates.

$$\mathbf{X}^{l}_{t,i}=\begin{bmatrix}\mathbf{A}_{c,i} & 0\\ 0 & \mathbf{I}\end{bmatrix}\mathbf{X}^{l}_{t-1,i}+\begin{bmatrix}\mathbf{B}_{c,i}\\ \mathbf{B}_{\mathrm{avg},i}\end{bmatrix}\otimes\mathbf{X}^{n}_{t,i}+\varepsilon^{l}_{t,i}; \tag{9a}$$

$$\mathbf{X}^{a_1}_{t,i}=\mathbf{A}_{\Phi,i}\,\mathbf{X}^{n}_{t,i}+\mathbf{B}_{\Phi,i}\,u_t+\varepsilon^{a_1}_{t,i}; \tag{9b}$$

$$\mathbf{X}^{a_2}_{t,i}=[0\quad\mathbf{I}]\mathbf{X}^{l}_{t,i}+\mathbf{B}_{\mathrm{surf},i}\,\mathbf{X}^{n}_{t,i}+\varepsilon^{a_2}_{t,i}; \tag{9c}$$

Algorithm 2 Stochastic Li-ion battery simulation

Inputs: $x^{a_1,1}_{1,p}(1)$, $x^{a_1,1}_{1,n}(1)$; $\mathbf{x}^{l}_{1,i}$, $\mathbf{x}^{n}_{1}$, for $i=\{p,s,n\}$; u_t for $t=1,2,\ldots,M$; and $\mathbf{\Sigma}^{l}_{t,i}$, $\mathbf{\Sigma}^{a_1}_{t,i}$, $\mathbf{\Sigma}^{a_2}_{t,i}$, $\mathbf{\Sigma}^{a_3}_{t,i}$, $\mathbf{\Sigma}^{n}_{t,i}$, Σ^{y}_{t} for $i=\{p,s,n\}$ and for $t=1,2,\ldots,M$

Initialize: δ_1, δ_2, $\mathbf{w}$, and $\mathbf{x}^{a_1,2}_{t,i}$ for $i=\{p,s,n\}$ and for $t=1,2,\ldots,M$.

for $t=1$ to M **do** ▷ simulate for M time points
  $\mathbf{x}^{n}_{t}\leftarrow[\mathbf{x}^{n}_{t,p},\mathbf{x}^{n}_{t,s},\mathbf{x}^{n}_{t,n}]^{\mathrm{T}}$.
  if $t\geq 2$ **then**
    $\mathbf{x}^{n}_{t}=\mathbf{x}^{n}_{t-1}$. ▷ initialize flux
  end if
  $\mathbf{i}_{\mathrm{bc}}\leftarrow\begin{bmatrix}x^{a_1,2}_{t,p}(N_p)-u_t\\ x^{a_1,2}_{t,n}(1)-0\end{bmatrix}$. ▷ error in BC of (C2)
  $[i_{\mathrm{bc},1},i_{\mathrm{bc},1}]^{\mathrm{T}}\leftarrow\mathbf{i}_{\mathrm{bc}}$.
  while $(|i_{bc}(1)|\geq\delta_1$ and $|i_{bc}(2)|\geq\delta_2)$ **do**
    Solve: (9a) for $i=\{p,s,n\}$.
    $\kappa_i\leftarrow\kappa_i(\mathbf{x}^{l,1}_{t,i})$ for $i=\{p,s,n\}$.
    Solve: (9b), (9c) and (9d) for $i=\{p,s,n\}$.
    $\mathbf{x}^{n}_{\mathrm{temp},i}\leftarrow$ Solve (9e) for $i=\{p,s,n\}$.
    $\mathbf{x}^{n}_{\mathrm{temp}}\leftarrow[\mathbf{x}^{n}_{\mathrm{temp},p},\mathbf{x}^{n}_{\mathrm{temp},s},\mathbf{x}^{n}_{\mathrm{temp},n}]^{\mathrm{T}}$.
    $\begin{bmatrix}\mathbf{x}^{n}_{t}\\ x^{a_1,1}_{t,p}(1)\\ x^{a_1,1}_{t,n}(1)\end{bmatrix}\leftarrow\operatorname{argmin} J\left(\mathbf{x}^{n}_{t},\mathbf{x}^{n}_{\mathrm{temp}},\mathbf{i}_{bc}\right)$.
  end while
  Solve: (9a) for $i=\{p,s,n\}$. ▷ update value
  Compute: (10). ▷ measurement
end for
return

$$\mathbf{X}_{t,i}^{a_3} = \mathcal{F}_{\Phi}\left(\mathbf{X}_{t,i}^{l}, \mathbf{X}_{t,i}^{n}\right) + \varepsilon_{t,i}^{a_3}; \tag{9d}$$

$$\mathbf{X}_{t,i}^{n} = \mathcal{F}_{j}\left(\mathbf{X}_{t,i}^{l}, \mathbf{X}_{t,i}^{a_1}, \mathbf{X}_{t,i}^{a_2}, \mathbf{X}_{t,i}^{a_3}\right) + \varepsilon_{t,i}^{n}, \tag{9e}$$

for all $i = \{p,s,n\}$. The notation $\otimes$ is the Kronecker product, and $\varepsilon_{t,i}^{l}, \varepsilon_{t,i}^{a_1}, \varepsilon_{t,i}^{a_2}, \varepsilon_{t,i}^{a_3}$, and $\varepsilon_{t,i}^{n}$ are independent multivariate Gaussian random noise sequences. These noise sequences model the uncertainty in the dynamics of the states – $\mathbf{X}_{t,i}^{l}$, $\mathbf{X}_{t,i}^{a_1}$, $\mathbf{X}_{t,i}^{a_2}$, $\mathbf{X}_{t,i}^{a_3}$, and $\mathbf{X}_{t,i}^{n}$. The states are stochastic and defined as sequences of random variables. It is important to highlight that while the state $\mathbf{X}_{t,i}^{l}$ in (9a) is a first-order Markov process in the time direction (recall that a first-order temporal discretization of ($M1$), and ($M2$) is used, see (2) and (5)), the states $\mathbf{X}_{t,i}^{a_1}$, $\mathbf{X}_{t,i}^{a_2}$, $\mathbf{X}_{t,i}^{a_3}$, and $\mathbf{X}_{t,i}^{n}$ in (9b) through (9e) evolve algebraically in the time direction, but are Markov processes in the spatial direction; wherein, $\mathbf{X}_{t,i}^{a_1}$ and $\mathbf{X}_{t,i}^{a_3}$ are first-order Markov processes, and $\mathbf{X}_{t,i}^{a_2}$ and $\mathbf{X}_{t,i}^{n}$ are zeroth-order Markov processes. In a typical Li-ion battery, the measurements for the states in (9) are generally not available. In other words, the states in (9) are *unobserved* or *latent*. The variables that are typically measured for a battery are the current and voltage. Assuming voltage measurements across the battery are available.

$$Y_t = X_{t,p}^{a_1,1}(1) - X_{t,n}^{a_1,1}(N_n) + \varepsilon_t^{y}, \tag{10}$$

where $Y_t \in \mathbb{R}$ is the voltage across the battery, $\varepsilon_t^{y} \in \mathbb{R}$ is a random Gaussian sensor noise, and $X_{t,p}^{a_1,1}(1)$ and $X_{t,n}^{a_1,1}(1)$ are in vectors $\mathbf{X}_{t,p}^{a_1,1} \equiv [X_{t,p}^{a_1,1}(1), X_{t,p}^{a_1,1}(2), \ldots, X_{t,p}^{a_1,1}(N_p)]^{\mathrm{T}}$ and $\mathbf{X}_{t,n}^{a_1,1} \equiv [X_{t,n}^{a_1,1}(1), X_{t,n}^{a_1,1}(2), \ldots, X_{t,n}^{a_1,1}(N_n)]^{\mathrm{T}}$, respectively. Together, Equations (9a–e) and (10) represent a stochastic nonlinear P2D state-space model. Assuming: $\varepsilon_{t,i}^{l} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{t,i}^{l})$; $\varepsilon_{t,i}^{a_1} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{t,i}^{a_1})$; $\varepsilon_{t,i}^{a_2} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{t,i}^{a_2})$; $\varepsilon_{t,i}^{a_3} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{t,i}^{a_3})$; $\varepsilon_{t,i}^{n} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{t,i}^{n})$; and $\varepsilon_t^{y} \sim \mathcal{N}(0, \Sigma_t^{y})$ are perfectly known in their distribution class, (9) and (10) can be compactly represented as a probabilistic model defined as follows:

$$\mathbf{X}_{t,i}^{l} \left| \left(\mathbf{x}_{t-1,i}^{l}, \mathbf{x}_{t,i}^{n}\right) \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{t,i}^{l}\right); \right. \tag{11a}$$

$$\mathbf{X}_{t,i}^{a_1} \left| \left(\mathbf{x}_{t,i}^{n}, u_t\right) \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{t,i}^{a_1}\right); \right. \tag{11b}$$

$$\mathbf{X}_{t,i}^{a_2} \left| \left(\mathbf{x}_{t,i}^{l}, \mathbf{x}_{t,i}^{n}\right) \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{t,i}^{a_2}\right); \right. \tag{11c}$$

$$\mathbf{X}_{t,i}^{a_3} \left| \left(\mathbf{x}_{t,i}^{l}, \mathbf{x}_{t,i}^{n}\right) \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{t,i}^{a_3}\right); \right. \tag{11d}$$

$$\mathbf{X}_{t,i}^{n} \left| \left(\mathbf{x}_{t,i}^{l}, \mathbf{x}_{t,i}^{a_1}, \mathbf{x}_{t,i}^{a_2}, \mathbf{x}_{t,i}^{a_3}\right) \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{t,i}^{n}\right); \right. \tag{11e}$$

$$Y_t \left| \left(x_{t,p}^{l}(1), x_{t,n}^{l}(N_n)\right) \sim \mathcal{N}(0, \Sigma_t^{y}), \right. \tag{11f}$$

for all $i = \{p,s,n\}$ and $t = 1,2,\ldots,M$. The zero mean of the multivariate Gaussian distribution in (11) is generically denoted as a vector $\mathbf{0}$. Note that the vector $\mathbf{0}$ in (11) is of appropriate dimension as defined based on the state and the section of the battery. In (11), the lowercase variables denote random realizations of the states. Finally, Algorithm 2 gives the pseudocode for simulating the stochastic P2D model in (9a) through (9e).

Example 2. (*Stochastic P2D Model Simulation*) In this simulation example, Example 1 is reconsidered, but under stochastic settings. The stochastic P2D model is obtained by adding zero mean Gaussian noise sequences in (9a) through (9e). The covariances for the Gaussian noise added to the stochastic model are:

$$\boldsymbol{\Sigma}_{t,i}^{l} = \begin{bmatrix} 0.01^2 & 0 \\ 0 & 2.0^2 \end{bmatrix}; \tag{12a}$$

$$\boldsymbol{\Sigma}_{t,i}^{a_1} = \begin{bmatrix} 0.02^2 & 0 \\ 0 & 0.2^2 \end{bmatrix}; \tag{12b}$$

$$\boldsymbol{\Sigma}_{t,i}^{a_2} = 2^2; \tag{12c}$$

$$\boldsymbol{\Sigma}_{t,i}^{a_3} = 0.0020^2; \tag{12d}$$

$$\boldsymbol{\Sigma}_{t,i}^{n} = 4 \times 10^{-14}; \tag{12e}$$

$$\Sigma_t^{y} = 0.1^2, \tag{12f}$$

for all $i = \{p,s,n\}$ and $t = 1,2,\ldots,M$. The covariances in (12a) through (12f) are set based on the absolute values of the states during a standard discharge cycle. This allows to maintain a *high* signal-to-noise ratio, which is crucial for efficient state estimation (discussed in Section 5).

Algorithm 2 is used to simulate the stochastic P2D model. Fig. 5 shows the simulation of the stochastic P2D model. The concentration of the electrolyte, electrical potential in the electrolyte, current in the electrolyte, and the lithium ion flux are shown in Fig. 5(a) through 5(d). In contrast to the deterministic P2D simulation in Fig. 3, the stochastic case in Fig. 5 is relatively non-smooth, but captures the overall dynamics inside the Li-ion battery. For instance, Fig. 5(a) indicates that during a discharge cycle the concentration of electrolyte in the positive electrode decreases while the concentration on the negative electrode increases. Also, Fig. 5(b) through 5(d) conform to the typical behavior of other internal states of the Li-ion battery during a discharge cycle. The simulation of the electrolyte potential in solid particles and average lithium ion concentration in solid particles under the stochastic case are not shown here for the sake of brevity. The stochastic simulation is performed to ensure that the boundary conditions are satisfied. For instance, in Fig. 5(c), (d) the electrolyte current and the lithium ion flux in the separator are zero, as required by the boundary conditions. Finally, Fig. 6 shows the discharge curve at 1C for the stochastic P2D model. Similar discharge curves were obtained for higher discharge rates.

5. Decentralized state estimation

A recursive approach to estimate the latent states in the stochastic P2D model is discussed in this section. This is commonly referred to as the *state estimation* problem. Computing the latent states in (9a) through (9e) and (10) in real-time is a nontrivial problem, and is challenged due to the nonlinear, implicit, and complex relations between the unmeasured states. This problem is further exacerbated due to the lack of sufficient measurements. The only measurements available are the current (u_t) and voltage (y_t). Representing the states in (9a) through (9e) as

$$\mathbf{X}_{t,i} = \left[\mathbf{X}_{t,i}^{l}, \mathbf{X}_{t,i}^{a_1}, \mathbf{X}_{t,i}^{a_2}, \mathbf{X}_{t,i}^{a_3}, \mathbf{X}_{t,i}^{n}\right]^{\mathrm{T}}, \tag{13}$$

for all $i = \{p,s,n\}$, the objective is to estimate $\mathbf{X}_{t,i}$ given $\mathbf{z}_{1:t}$, where $\mathbf{z}_{1:t} = \{(u_1, y_1), (u_2, y_2), \ldots, (u_t, y_t)\}$ is a sequence of input and output measurements until time t. A nonlinear state estimation problem is typically solved in the Bayesian framework by

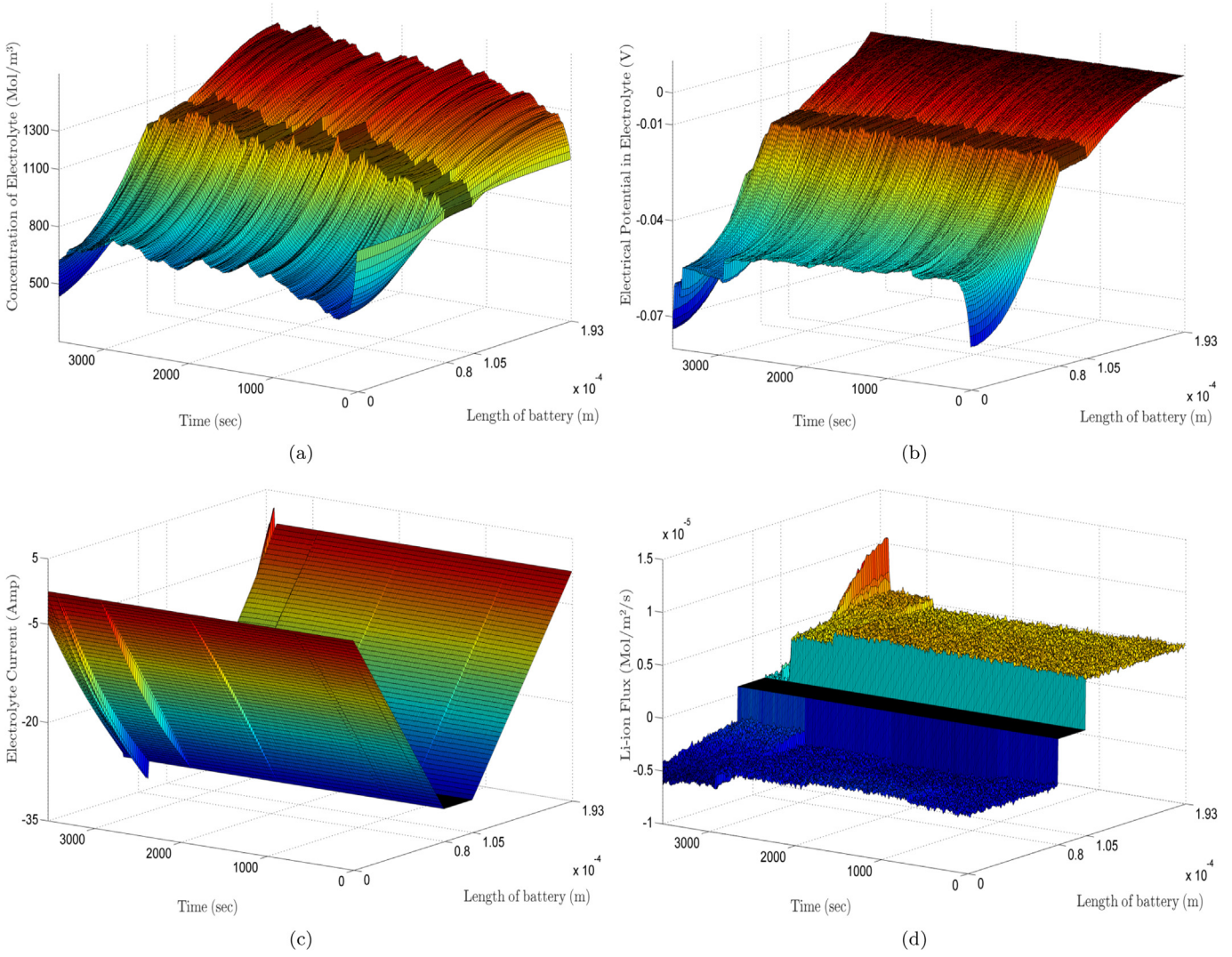

Fig. 5. Simulation of the stochastic P2D model using Algorithm 2. The subfigures show the dynamics of (a) concentration of electrolyte; (b) electrical potential in electrolyte; (c) current in the electrolyte; and (d) lithium ion flux at different times along the length of the battery. The ticks along the battery length correspond to the length of the positive electrode, separator and negative electrode.

formulating it as a filtering problem. For example, if $\widehat{\mathbf{x}}_{t,i}$ denote an estimate of $\mathbf{X}_{t,i}$, then in the Bayesian framework, $\widehat{\mathbf{x}}_{t,i}$ is calculated as

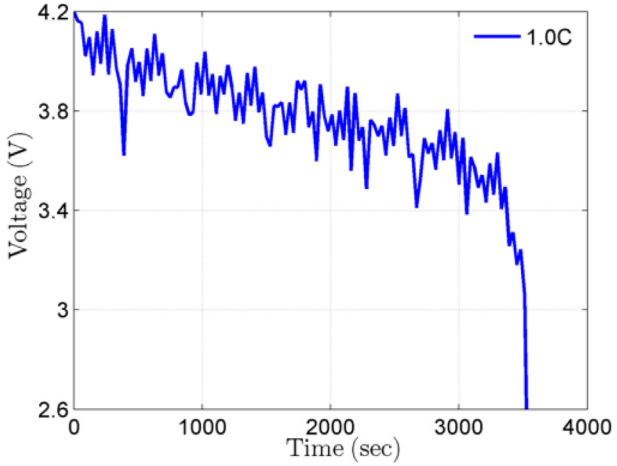

Fig. 6. Discharge curve at 1C for Stochastic P2D model.

the mean (or median, mode) of $\mathbf{X}_{t,i}|\mathbf{z}_{1:t} \sim p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t})$ for all $t = 1,2,\ldots,M$ and $i=\{p,s,n\}$. Here $p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t})$ is referred to as the *posterior* or *filtering* density. A posterior density encapsulates any statistical information on $\mathbf{X}_{t,i}$ given $\mathbf{z}_{1:t}$. The posterior density is computed using a class of methods called *filters*. While the Bayes' theorem provides an approach to recursively compute $p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t})$, it does not lend itself to a closed form solution for nonlinear systems. In other words, an optimal nonlinear filter that solves $p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t})$ exactly for the stochastic P2D model is not tractable in finite computational time. Many advanced nonlinaer filtering methods exist to approximate the optimal nonlinear filter. The quality of the state estimates obtained with these approximate nonlinear filters depends on the underlying numerical and statistical approximation techniques used in their designs. The idea of Bayesian nonlinear filtering methods is well established, and for the sake of brevity it is not discussed here, but can be found discussed in Refs. [36,37,38] and the references cited therein.

For state estimation in the stochastic P2D model, the objective is to recursively compute $\mathbf{X}_{t,i}|\mathbf{z}_{1:t} \sim p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t})$, where $\mathbf{X}_{t,i}$ has a dimension $7N_i$ (see (9a) through (9e)). Now for a fine spatial discretization mesh (*i.e.,* for a *relatively large* N_i), computing $p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t})$ becomes nontrivial. Generally, most of the nonlinear filters are

inefficient in high-dimensional spaces. This is due to the problem commonly referred to as the *curse of dimensionality*. A common approach to beat the *curse of dimensionality* is to perform state estimation on a reduced dimensional space instead. While existing nonlinear filters are efficient in a low dimensional space, computing the reduced space for complex systems, including the stochastic P2D model is nontrivial, and is a subject of discussion in this section.

An inspection of the structure of the stochastic P2D model in (9a) through (9e) indicates that the latent states are dependent on each other in a particular fashion that makes it possible to split the state estimation problem into lower dimensional problems. This is done by using the law of conditional probability, and rewriting $p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t})$ as

$$p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t}) = p\left(\mathbf{x}_{t,i}^{a_1}\middle|\mathbf{x}_{t,i}^{a_2},\mathbf{x}_{t,i}^{a_3},\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)p\left(\mathbf{x}_{t,i}^{a_2}\middle|\mathbf{x}_{t,i}^{a_3},\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) \times p\left(\mathbf{x}_{t,i}^{a_3}\middle|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)p\left(\mathbf{x}_{t,i}^{l}\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)p\left(\mathbf{x}_{t,i}^{n}\middle|\mathbf{z}_{1:t}\right). \tag{14}$$

In (14), the posterior density is decomposed into a product of five conditional density functions. It is possible to further simplify (14) by observing that the conditional density $p(\mathbf{x}_{t,i}^{a_1}|\mathbf{x}_{t,i}^{a_2},\mathbf{x}_{t,i}^{a_3},\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) = p(\mathbf{x}_{t,i}^{a_1}|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$ (see (9b)). Similarly, (9c) yields the relation $p(\mathbf{x}_{t,i}^{a_2}|\mathbf{x}_{t,i}^{a_3},\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) = p(\mathbf{x}_{t,i}^{a_2}|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$. Substituting them back into (14) yields

$$p(\mathbf{x}_{t,i}|\mathbf{z}_{1:t}) = p\left(\mathbf{x}_{t,i}^{l}\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)p\left(\mathbf{x}_{t,i}^{a_1}\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)\times p\left(\mathbf{x}_{t,i}^{a_2}\middle|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)p\left(\mathbf{x}_{t,i}^{a_3}\middle|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)p\left(\mathbf{x}_{t,i}^{n}\middle|\mathbf{z}_{1:t}\right), \tag{15}$$

where $\mathbf{X}_{t,i}^{l}\big|(\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) \sim p(\mathbf{x}_{t,i}^{l}|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$, $\mathbf{X}_{t,i}^{a_1}\big|(\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) \sim p(\mathbf{x}_{t,i}^{a_1}|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$, $\mathbf{X}_{t,i}^{a_2}\big|(\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) \sim p(\mathbf{x}_{t,i}^{a_2}|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$, $\mathbf{X}_{t,i}^{a_3}\big|(\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) \sim p(\mathbf{x}_{t,i}^{a_3}|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$, and $\mathbf{X}_{t,i}^{n}\big|\mathbf{z}_{1:t} \sim p(\mathbf{x}_{t,i}^{n}|\mathbf{z}_{1:t})$ are conditional states of dimensions $2N_i$, $2N_i$, N_i, N_i, and N_i, respectively. With the conditional states and their densities defined in (15), standard nonlinear filters can be used for state estimation in the reduced space. The procedure of estimating states in the reduced space is called *decentralized* state estimation in this article. Recall that since $\mathbf{X}_{t,i}^{l}$ in (9a) is a first-order Markov process in the temporal direction, a temporal filter can be used to compute $p(\mathbf{x}_{t,i}^{l}|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$ in (15). This temporal filter performs a $2N_i$-dimensional temporal sweep. Similarly, since $\mathbf{X}_{t,i}^{a_1}$, $\mathbf{X}_{t,i}^{a_2}$, $\mathbf{X}_{t,i}^{a_3}$, and $\mathbf{X}_{t,i}^{n}$ in (9b) through (9e) are Markov processes in the spatial direction, a spatial filter can be used to compute the conditional densities in (15).

It is possible to further reduce the dimension of the conditional densities in (15) in the spatial direction. Observe that the charge conservation equations (*C*1), (*C*2), and (*C*3) in Table 1 only have spatial derivatives. This fact is exploited to further reduce the dimension of the density functions in (15) in the spatial direction. For instance, at any given time t, the conditional density $p(\mathbf{x}_{t,i}^{a_1}|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$ is

$$p\left(\mathbf{x}_{t,i}^{a_1}\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) = p\left(x_{t,i}^{a_1}(1),x_{t,i}^{a_1}(2),\ldots,x_{t,i}^{a_1}(N_i)\right)\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) \tag{16}$$

Using the law of conditional probability, (16) is given by

$$p\left(\mathbf{x}_{t,i}^{a_1}\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) = p\left(x_{t,i}^{a_1}(1)\middle|x_{t,i}^{a_1}(2),\ldots,x_{t,i}^{a_1}(N_i),\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) \times p\left(x_{t,i}^{a_1}(2)\middle|x_{t,i}^{a_1}(3)\ldots,x_{t,i}^{a_1}(N_i),\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right)\times\ldots \times p\left(x_{t,i}^{a_1}(N_i)\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right). \tag{17}$$

Using the first-order Markov property of $\mathbf{X}_{t,i}^{a_1}$ in the spatial direction, (17) can be further simplified and written as

$$p\left(\mathbf{x}_{t,i}^{a_1}\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) = p\left(x_{t,i}^{a_1}(N_i)\middle|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) \times\prod_{k=1}^{N_i-1} p\left(x_{t,i}^{a_1}(k)\middle|x_{t,i}^{a_1}(k+1),\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right). \tag{18}$$

Observe that in (18), the conditional density is defined for a 2-dimensional state $X_{t,i}^{a_1}(k)$. As compared to the density $p(\mathbf{x}_{t,i}^{a_1}|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$ in (15), the decomposition in (18) reduces the dimension for state estimation from $2N_i$ to 2. Now using similar arguments, as in (18), other densities in (15) can be reduced in the spatial direction to obtain:

(a) conditional density for $\mathbf{X}_{t,i}^{a_2}\big|(\mathbf{x}_{t,i}^{a_3},\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) \sim$

$$p\left(\mathbf{x}_{t,i}^{a_2}\middle|\mathbf{x}_{t,i}^{a_3},\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) = \prod_{k=1}^{N_i} p\left(x_{t,i}^{a_2}(k)\middle|\mathbf{x}_{t,i}^{a_3},\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right); \tag{19}$$

(b) conditional density for $\mathbf{X}_{t,i}^{a_3}\big|(\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}) \sim$

$$p\left(\mathbf{x}_{t,i}^{a_3}\middle|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) = p\left(x_{t,i}^{a_3}(N_i)\middle|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right) \times\prod_{k=1}^{N_i-1} p\left(x_{t,i}^{a_3}(k)\middle|x_{t,i}^{a_3}(k+1),\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t}\right); \tag{20}$$

(c) conditional density for $\mathbf{X}_{t,i}^{n}\big|(\mathbf{z}_{1:t}) \sim$

$$p\left(\mathbf{x}_{t,i}^{n}\middle|\mathbf{z}_{1:t}\right) = \prod_{k=1}^{N_i} p\left(x_{t,i}^{n}(k)\middle|\mathbf{z}_{1:t}\right). \tag{21}$$

The conditional densities in (15), and further dimension reduction in the spatial direction in (18) though (21) dramatically reduce the dimension of the problem for state estimation. Table 3 gives a list of density functions to be approximated using a filter. For instance (*D*2), is a two-dimensional density function that is approximated through a spatial filter sweep. Further, from Table 3 it is clear that as compared to the original problem defined in dimension $7N_i$ the proposed decentralized method reduces the dimension to $2N_i+5$. This leads to a huge improvement in the overall state estimation problem formulation. In the next section, a nonlinear filtering method is discussed to compute the conditional densities given in Table 3.

6. Particle methods for state estimation

In a linear stochastic models, the state posterior density is typically Gaussian and can be exactly represented by the Kalman filter (KF) using a finite number of moments (e.g., mean, variance). In nonlinear stochastic systems, the posterior density is typically non-Gaussian, and at least in theory, an infinite number of moments

Table 3
Decentralized state estimation in stochastic P2D model.

Density	Filtet Type	Dimension	
		Full	Reduced
(D1) $p(\mathbf{x}_{t,i}^{l}\|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$	Temporal	$2N$	$2N$
(D2) $p(\mathbf{x}_{t,i}^{a_1}\|\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$	Spatial	$2N$	2
(D3) $p(\mathbf{x}_{t,i}^{a_2}\|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$	Spatial	N	1
(D4) $p(\mathbf{x}_{t,i}^{a_3}\|\mathbf{x}_{t,i}^{l},\mathbf{x}_{t,i}^{n},\mathbf{z}_{1:t})$	Spatial	N	1
(D5) $p(\mathbf{x}_{t,i}^{n}\|\mathbf{z}_{1:t})$	Spatial	N	1

are required for the exact representation of the density [39]. Thus, with finite computing capabilities, an optimal state filter for nonlinear systems, including the stochastic P2D model in (9a) through (9e) is not realizable. In the last few decades, several approximate nonlinear state filters based on statistical and analytical approximations of the optimal nonlinear filter have been developed for state estimation in nonlinear systems [40,41,42]. Most of these nonlinear filters can be classified as either Kalman-based filters or sequential Monte-Carlo (SMC)-based filters [43]. Both the Kalman and SMC-based nonlinear filters are tractable in finite computational time and can be used for state estimation in general or specific types of nonlinear systems. A detailed exposition of nonlinear filtering methods and related approximations is not included here, but can be found in the handbook on nonlinear filtering [44].

The class of SMC-based filtering methods, popularly referred to as *particle filters* is an importance class of filtering methods for nonlinear systems. The basic idea behind particle filters is to approximate the posterior density by generating random samples (or *particles*) from it. Since generating direct random particles distributed according to the posterior density is nontrivial for nonlinear systems, an importance density is emploued that is easier to generate particles from. Once particles are generated from the importance density, appropriate particle weights are assigned based on the how likely it is for the particle to be classified as a particle from the posterior density. Some of the popular particle filtering algorithms, include sampling importance resampling (SIR) filter, auxiliary SIR (ASIR) filter, and Rao-Blackwellized particle filter (RBPF). The rest of this section assumes a familiarity with the theory and implementation of particles filters due to space limitations and because many well-written publications on particle filters are available in the open literature. Readers not familiar with the subject are referred to [45] and the references cited therein.

An attractive feature of particle filters is that the posterior density approximation can be made arbitrarily accurate by simply

Algorithm 3 Tethered Particle Filtering

1: **for** $t = 1$ to T **do**

2: Obtain $\mathbf{x}^n_{t-1,i}$ by solving Algorithm 2 and calculate $p(\mathbf{x}^n_{t,i}|\mathbf{z}_{1:t})$ using a spatial particle filter, such that

$$\tilde{p}(d\mathbf{x}^n_{t,i}|\mathbf{z}_{1:t}) = \frac{1}{P}\sum_{k=1}^{P}\delta_{\mathbf{X}^{n,k}_{t,i}}(d\mathbf{x}^n_{t,i}), \tag{22}$$

where $\{\mathbf{X}^{n,k}_{t,i}\}_{k=1}^{P} \sim p(\mathbf{x}^n_{t,i}|\mathbf{z}_{1:t})$ are particles distributed according to $p(\mathbf{x}^n_{t,i}|\mathbf{z}_{1:t})$.

3: **Compute:** state estimate

$$\widehat{\mathbf{x}}^n_{t,i} = \frac{1}{P}\sum_{k=1}^{P}\mathbf{X}^{n,k}_{t,i}. \tag{23}$$

4: Calculate $p(\mathbf{x}^l_{t,i}|\widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t})$ using a temporal particle filter and represent the distribution

$$\tilde{p}(d\mathbf{x}^l_{t,i}|\widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t}) = \frac{1}{P}\sum_{k=1}^{P}\delta_{\mathbf{X}^{l,k}_{t,i}}(d\mathbf{x}^l_{t,i}). \tag{24}$$

5: Calculate $p(\mathbf{x}^{a_1}_{t,i}|\widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t})$ and $p(\mathbf{x}^{a_2}_{t,i}|\mathbf{x}^l_{t,i}, \widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t})$ using a spatial particle filter, such that

$$\tilde{p}(d\mathbf{x}^{a_1}_{t,i}|\widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t}) = \frac{1}{P}\sum_{k=1}^{P}\delta_{\mathbf{X}^{a_1,k}_{t,i}}(d\mathbf{x}^{a_1}_{t,i}), \tag{25}$$

$$\tilde{p}(d\mathbf{x}^{a_2}_{t,i}|\mathbf{x}^l_{t,i}, \widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t}) = \frac{1}{P}\sum_{k=1}^{P}\delta_{\mathbf{X}^{a_2,k}_{t,i}}(d\mathbf{x}^{a_2}_{t,i}). \tag{26}$$

6: Calculate $p(\mathbf{x}^{a_3}_{t,i}|\mathbf{x}^l_{t,i}, \widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t})$ using a spatial particle filter and represent the distribution

$$\tilde{p}(d\mathbf{x}^{a_3}_{t,i}|\mathbf{x}^l_{t,i}, \widehat{\mathbf{x}}^n_{t,i}, \mathbf{z}_{1:t}) = \frac{1}{P}\sum_{k=1}^{P}\delta_{\mathbf{X}^{a_3,k}_{t,i}}(d\mathbf{x}^l_{t,i}). \tag{27}$$

7: **Compute:** state estimates –

$$\widehat{\mathbf{x}}^l_{t,i} = \frac{1}{P}\sum_{k=1}^{P}\mathbf{X}^{l,k}_{t,i}, \tag{28a}$$

$$\widehat{\mathbf{x}}^{a_1}_{t,i} = \frac{1}{P}\sum_{k=1}^{P}\mathbf{X}^{a_1,k}_{t,i}, \tag{28b}$$

$$\widehat{\mathbf{x}}^{a_2}_{t,i} = \frac{1}{P}\sum_{k=1}^{P}\mathbf{X}^{a_2,k}_{t,i}, \tag{28c}$$

$$\widehat{\mathbf{x}}^{a_3}_{t,i} = \frac{1}{P}\sum_{k=1}^{P}\mathbf{X}^{a_3,k}_{t,i}. \tag{28d}$$

8: **end for**

increasing the number of particles. The traditional particle filtering algorithms are based on the standard temporal first-order Markov state-space model assumption; however, as discussed in Section 4, the stochastic P2D model in (9a)–(9e) includes implicit nonlinear equations that leads to states that are Markov processes in either temporal or spatial direction. In fact the model in (9a)–(9e) includes both zero and first order Markov processes in the spatial direction, which makes any direct use of particle filters a challenging problem. Further, in standard state space models, the states are calculated at each time step with only one iteration per time step; however, the optimization step in Algorithm 2 to simulate the stochastic P2D model may require numerous iterations at each time step to solve the implicit nonlinear equations. A standard particle filtering method will therefore, require performing these iterations on each particle. Hence, a straightforward application of particle filtering algorithm, while technically feasible, is computationally prohibitive.

6.1. Tethered particle filters

As discussed in Sections 5 and 6, in addition to the challenges posed by the large dimensionality of the states in the stochastic P2D model in (9a) through (9e), some of the states are implicitly related and therefore, can not be simply propagated to a future time. From Algorithm 2, it is clear that the states have to be estimated by solving an optimization routine that requires solving a set of implicit equations at each time. Under this setting, a standard particle filter algorithm would require solving the implicit equations for propagating each particle to a future time, thereby increasing the computational burden dramatically.

To mitigate the computational complexity of this naive approach, the flux particles from $p(\mathbf{x}_{t,i}^n|\mathbf{z}_{1:t})$ are not propagated forward; and instead, the estimate of the flux, denoted by $\widehat{\mathbf{x}}_{t,i}^n$ for $t = 1,2,\ldots,M$ and for $i\in\{p,s,n\}$, is propagated to the next time step. The estimate of the flux, $\widehat{\mathbf{x}}_{t,i}^n$, is called a *tethered* particle, and the filter is called a *tethered particle filter*. Once the flux particles are tethered around $\widehat{\mathbf{x}}_{t,i}^n$, only one iteration is required to solve the implicit equations, which reduces the computational cost. Further, to ensure that the generated particles satisfy the boundary conditions exactly at each time, the choice of a importance density function is crucial. In tethered particle filter, a Dirac delta importance function is chosen at the boundary conditions so that the particles collapse to the exact boundary condition. A pseudocode for the tethered particle filter based state estimation in stochastic P2D model is given in Algorithm 3.

Example 3. (*State Estimation*) In this example the efficacy of the proposed tethered particle filter based state estimation is discussed for the stochastic P2D model is demonstrated. Consider the simulation scenario in Example 2. The objective is to estimate the states using Algorithm 3. Here Algorithm 3 is implemented with P=2000 particles. Fig. 7 gives the error in estimating the electrical potential and current in the electrolyte using Algorithm 3. The error in estimating the electrical potential in the electrolyte (see Fig. 7(a)) is in the order of magnitude 10^{-3} while the error in estimating the current in the electrolyte (see Fig. 7(b)) is within the same order of magnitude for all sampling times, and along the entire length of the battery. Other latent states were also accurately estimated; however, for the sake of brevity the results are not shown here. Fig. 8(a) shows the estimated discharge curve compared against the true discharge curve. It is evident that Algorithm 3 is successful in filtering out excessive sensor noise in the measurements, while keeping track of the underlying true discharge curve. All in all, Figs. 7 and 8(a) suggest that Algorithm 3 is efficient in estimating the latent states of the stochastic P2D model and the discharge curve.

7. State-of-Charge

The state-of-charge (SOC) is an important notion that quantifies the residual amount of charge in a battery based on the concentration of lithium ions in the solid particles in each electrode. For example, if $c_p^{\max}$ is the maximum possible concentration of lithium ions in the solid particles in the positive electrode, then the bulk SOC is defined as

$$\mathcal{S}_p(t) \equiv \frac{1}{l_p c_p^{\max}} \int_0^{l_p} \bar{c}_{s,p}(t,x)dx, \tag{29}$$

where $S_p\in\mathbb{R}^+$ is the bulk SOC in the positive electrode. During a discharge cycle, as the average lithium ion concentration in the solid particles in the positive electrode, denoted by $\bar{c}_{s,p}$ in (29),

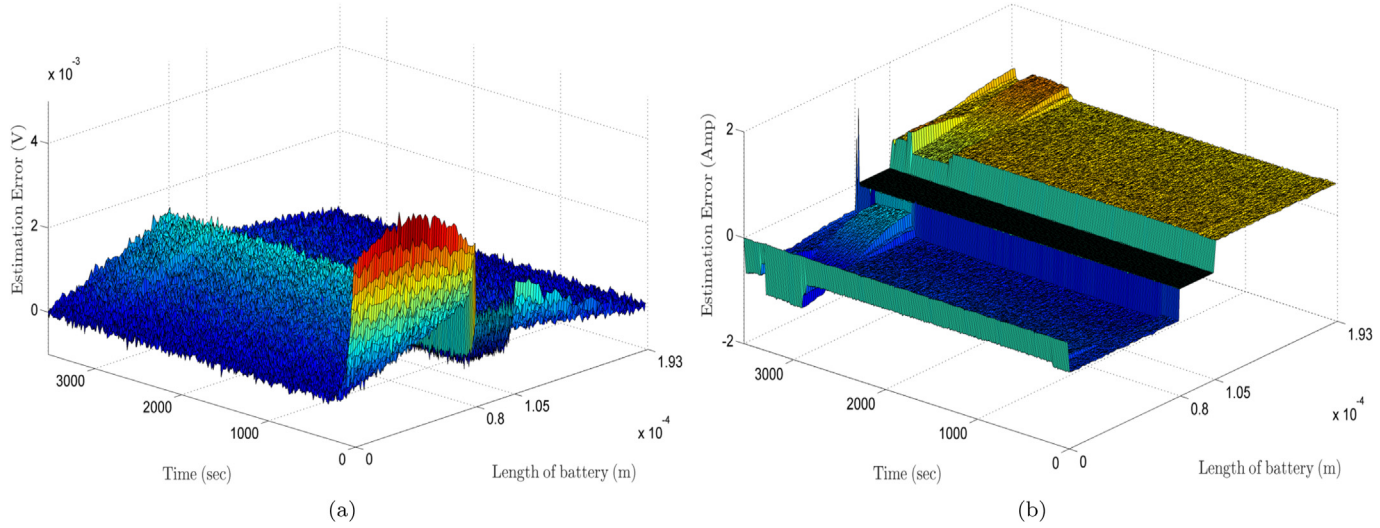

Fig. 7. State estimation in stochastic P2D model using Algorithm 4. The subfigures show the error in estimating (a) electrical potential in electrolyte, and (b) current in the electrolyte at different times along the length of the battery. The ticks along the battery length correspond to the length of the positive electrode, separator and negative electrode.

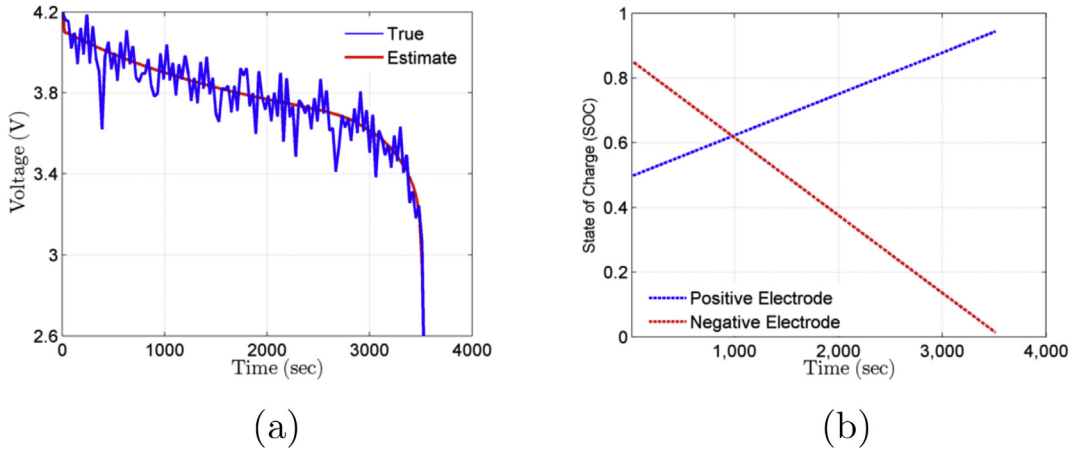

Fig. 8. (a) Discharge curve at 1C for Stochastic P2D model. (b) Estimated expected bulk SOC in the positive and negative electrodes.

increases, S_p increases as well. The SOC in the negative electrode, denoted by S_n can be defined likewise using $c_n^{\max}$ and the average lithium ion concentration in the solid particles in the negative electrode, $\bar{c}_{s,n}$. During a discharge cycle, S_n decreases as a function of time as $\bar{c}_{s,n}$ decreases in the negative electrode. The SOC in the separator is zero as the average lithium ion concentration in the solid particles in the separator is zero. The typical behavior of the average lithium ion concentration in the solid particles in the positive and negative electrodes during a discharge cycle is given in Fig. 3(b).

While (29) gives an expression to compute SOC in the positive electrode, it does not lend itself to a closed form solution. This is because as discussed in Section 3, there is no closed form solution for $\bar{c}_{s,p}$. From the numerical solution in Section 3, the SOC in (29) can be written as

$$\overline{\mathcal{S}}_p(t) = \frac{1}{l_p c_p^{\max}} \sum_{m=1}^{N_p} \bar{c}_{s,p}(t,m)\Delta x_p, \tag{30a}$$

$$= \frac{\Delta x_p}{l_p c_p^{\max}} \sum_{m=1}^{N_p} \bar{c}_{s,p}(t,m), \tag{30b}$$

where $\overline{\mathcal{S}}_p$ is an approximation of $\mathcal{S}_p$ in (29), which can also be written as

$$\overline{\mathcal{S}}_p(t) = \frac{\Delta x_p}{l_p c_p^{\max}} \sum_{m=1}^{N_p} X_{t,p}^{l,2}(m). \tag{31}$$

The SOC, $\overline{\mathcal{S}}_p$ in (31) is a random variable since it is a function of $\mathbf{X}_{t,p}^{l,2}$. Further, recall from Section 4, that $\mathbf{X}_{t,p}^{l,2}$ is a latent state. In other words, $\overline{\mathcal{S}}_p$ in (31) is also a latent state, that needs to be estimated using available measurements. Interestingly, the bulk SOC can be estimated from an estimate of lithium ion concentration in the active material and no other estimates are directly required. This observation is used to simplify SOC estimation. First, a good measure of (31) is its expected value, which is

$$\mathbb{E}\left[\overline{\mathcal{S}}_p(t)\right] = \mathbb{E}\left[\frac{\Delta x_p}{l_p c_p^{\max}} \sum_{m=1}^{N_p} X_{t,p}^{l,2}(m)\right], \tag{32}$$

where $\mathbb{E}[\overline{\mathcal{S}}_p(t)]$ is the expected bulk SOC, with expectation defined over $p(\mathbf{x}_{t,p}^{l,2}|\mathbf{x}_{t,p}^{n}, z_{1:t})$, such that

$$\mathbb{E}\left[\overline{\mathcal{S}}_p(t)\right] = \frac{\Delta x_p}{l_p c_p^{\max}} \int \left[\sum_{m=1}^{N_p} x_{t,p}^{l,2}(m)\right] p\left(d\mathbf{x}_{t,p}^{l,2}\middle|\mathbf{x}_{t,p}^{n}, z_{1:t}\right), \tag{33}$$

where $p(d\mathbf{x}_{t,p}^{l,2}|\mathbf{x}_{t,p}^{n}, z_{1:t}) \equiv p(\mathbf{x}_{t,p}^{l,2}|\mathbf{x}_{t,p}^{n}, z_{1:t}) d\mathbf{x}_{t,p}^{l,2}$. Equation (33) can be further simplified and written as follows

$$\mathbb{E}\left[\overline{\mathcal{S}}_p(t)\right]$$

$$= \frac{\Delta x_p}{l_p c_p^{\max}} \sum_{m=1}^{N_p} \int x_{t,p}^{l,2}(m) p\left(d\mathbf{x}_{t,p}^{l,2}\middle|\mathbf{x}_{t,p}^{n}, z_{1:t}\right), \tag{34a}$$

$$= \frac{\Delta x_p}{l_p c_p^{\max}} \sum_{m=1}^{N_p} \int x_{t,p}^{l,2}(m) p\left(dx_{t,p}^{l,2}(m)\middle|\mathbf{x}_{t,p}^{n}, z_{1:t}\right). \tag{34b}$$

Observe that the integral in (34b) is with respect to the conditional density $p(x_{t,p}^{l,2}(m)|\mathbf{x}_{t,p}^{n}, z_{1:t})$. A Monte-Carlo method to approximate the integral in (34b) requires a set of random particles from the density $p(x_{t,p}^{l,2}(m)|\mathbf{x}_{t,p}^{n}, z_{1:t})$. A procedure to compute $p(\mathbf{x}_{t,p}^{l,2}(m)|\mathbf{x}_{t,p}^{n}, z_{1:t})$ using the proposed tethered particle filter is discussed in Algorithm 3. Assuming $\{X_{t,p}^{l,2,i}(m)\}_{i=1}^{P} \sim p(x_{t,p}^{l,2}(m)|\mathbf{x}_{t,p}^{n}, z_{1:t})$ represents a pool of P random particles approximately distributed according to $p(x_{t,p}^{l,2}(m)|\mathbf{x}_{t,p}^{n}, z_{1:t})$, and obtained from Algorithm 3, then $p(dx_{t,p}^{l,2}(m)|\mathbf{x}_{t,p}^{n}, z_{1:t})$ is represented as

$$p\left(dx_{t,p}^{l,2}(m)\middle|\mathbf{x}_{t,p}^{n}, z_{1:t}\right) = \frac{1}{P}\sum_{i=1}^{P} \delta_{X_{t,p}^{l,2,i}(m)}\left(dx_{t,p}^{l,2}(m)\right), \tag{35}$$

where $\delta_{X_{t,p}^{l,2,i}(m)}(\cdot)$ is a Dirac delta measure at particle location $X_{t,p}^{l,2,i}(m)$. Now substituting (35) into (34b) yields

$$\mathbb{E}\left[\overline{\mathcal{S}}_p(t)\right]$$

$$\approx \frac{\Delta x_p}{P l_p c_p^{\max}} \sum_{m=1}^{N_p} \int x_{t,p}^{l,2}(m) \sum_{i=1}^{P} \delta_{X_{t,p}^{l,2,i}(m)}\left(dx_{t,p}^{l,2}(m)\right), \tag{36a}$$

$$= \frac{\Delta x_p}{Pl_p c_p^{\max}} \sum_{m=1}^{N_p} \sum_{i=1}^{P} X_{t,p}^{l,2,i}(m), \tag{36b}$$

where (36b) is a P-particle Monte-Carlo approximation of (34b). The expected bulk SOC in the negative electrode can be similarly approximated and computed as:

$$\mathbb{E}\left[\overline{\mathcal{S}}_n(t)\right] \approx \frac{\Delta x_n}{Pl_n c_n^{\max}} \sum_{m=1}^{N_n} \sum_{i=1}^{P} X_{t,n}^{l,2,i}(m), \tag{37}$$

where $\{X_{t,n}^{l,2,i}(m)\}_{i=1}^{P} \sim p(x_{t,n}^{l,2}(m)\big|\mathbf{x}_{t,n}^{n}, z_{1:t})$ represents a pool of P random particles approximately distributed according to $p(x_{t,n}^{l,2}(m)\big|\mathbf{x}_{t,n}^{n}, z_{1:t})$ and obtained from Algorithm 3. Finally, as discussed in Section 6, the quality of the expected SOC in the positive and negative electodes computed in (36b) and (37), respectively, depend on the number of particles used, and can be made arbitrarily accurate by simply increasing P.

Example 4. (*State-of-Charge Estimation*) This example deals with the SOC estimation in a Li-ion battery using the ideas discussed in this section. The particles required to compute the expected bulk SOC in the positive and negative electrodes are computed using the tethered particle filters discussed in Algorithm 3. Fig. 8(b) shows the estimated expected bulk SOC in the positive and negative electrodes as a function of time. As one would expect, the expected bulk SOC increases in the positive electrode as the average lithium ion concentration in solid particles increases on the positive side during a discharge cycle. Similarly, as the average lithium ion concentration in solid particles depletes in the negative electrode, the expected bulk SOC in the negative electrode drops. Fig. 8(b) demonstrates that the proposed method is efficient in accurately estimating the expected bulk SOC both in the positive and negative electrodes of a Li-ion battery.

8. Conclusions

The complex nonlinear PDEs that define the dynamics of a standard Li-ion battery are discretized and reformulated as a large dimensional state-space model. The state of charge and other battery properties that depend on unmeasured state variables such as concentrations and potentials are estimated using a modified particle filtering algorithm. The algorithm uses a novel technique called 'tethering' to reduce computational complexity. A simulation example shows that the proposed algorithm provides accurate estimation of the state of charge in the presence of significant state and measurement noise. The P2D model as described in this article is one of the more sophisticated models, however, it does not account for the health of the battery as it changes with charge and discharge cycles. The proposed algorithm can be easily adapted to integrate state of health equations into the P2D model.

References

[1] G.L. Plett, Extended kalman filtering for battery management systems of lipb-based hev battery packs: Part 2. modeling and identification, J. Power Sources 134 (2) (2004) 262–276.
[2] M. Dubarry, B.Y. Liaw, Development of a universal modeling tool for rechargeable lithium batteries, J. Power Sources 174 (2) (2007) 856–860.
[3] D. Andre, M. Meiler, K. Steiner, H. Walz, T. Soczka-Guth, D. Sauer, Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. ii: Modelling, J. Power Sources 196 (12) (2011) 5349–5356.
[4] X. Hu, S. Li, H. Peng, A comparative study of equivalent circuit models for Li-ion batteries, J. Power Sources 198 (2012) 359–367.
[5] S. Santhanagopalan, Q. Guo, P. Ramadass, R.E. White, Review of models for predicting the cycling performance of lithium ion batteries, J. Power Sources 156 (2) (2006) 620–628.
[6] N. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, A. Kojic, Algorithms for advanced battery-management systems, IEEE Control Syst. Mag. 30 (3) (2010) 49–68.
[7] J. Newman, W. Tiedemann, Porous-electrode theory with battery applications, AIChE J. 21 (1) (1975) 25–41.
[8] C.M. Doyle, Design and Simulation of Lithium Rechargeable Batteries, Ph.D. thesis, University of California, Berkeley (, 1995.
[9] S. Renganathan, G. Sikha, S. Santhanagopalan, R.E. White, Theoretical analysis of stresses in a lithium ion cell, J. Electrochem. Soc. 157 (2) (2010) A155–A163.
[10] D.E. Stephenson, E.M. Hartman, J.N. Harb, D.R. Wheeler, Modeling of particle-particle interactions in porous cathodes for lithium-ion batteries, J. Electrochem. Soc. 154 (12) (2007) A1146–A1155.
[11] S. Garofalini, Molecular dynamics simulations of li transport between cathode crystals, J. Power Sources 110 (2) (2002) 412–415.
[12] V. Ramadesigan, P.W.C. Northrop, S. De, S. Santhanagopalan, R.D. Braatz, V.R. Subramanian, Modeling and simulation of lithium-ion batteries from a systems engineering perspective, J. Electrochem. Soc. 159 (3) (2012) R31–R45.
[13] Y. Liang, K. Wen, Y. Mao, Z. Liu, G. Zhu, F. Yang, W. He, Shape and size control of lifepo4 for high-performance lithium-ion batteries, Chem. Electro. Chem. 2 (9) (2015) 1227–1237.
[14] S. Yang, B. Zhou, Z. Ding, H. Zheng, L. Huang, J. Pan, W. Wu, H. Zhang, Tetragonal hematite single crystals as anode materials for high performance lithium ion batteries, J. Power Sources 286 (2015) 124–129.
[15] P.W.C. Northrop, V. Ramadesigan, S. De, V.R. Subramanian, Coordinate transformation, orthogonal collocation, model reformulation and simulation of electrochemical-thermal behavior of Lithium-ion battery stacks, J. Electrochem. Soc. 158 (12) (2011) A1461–A1477.
[16] K.A. Smith, Electrochemical Modeling, Estimation and Control of Lithium Ion Batteries, Ph.D. thesis, Pennsylvania State University, University Park, 2006.
[17] J. Chiasson, B. Vairamohan, Estimating the state of charge of a battery, IEEE Trans. Control Syst. Technol. 13 (3) (2005) 465–470.
[18] I.-S. Kim, The novel state of charge estimation method for lithium battery using sliding mode observer, J. Power Sources 163 (1) (2006) 584–590.
[19] Y. Hu, S. Yurkovich, Battery cell state-of-charge estimation using linear parameter varying system techniques, J. Power Sources 198 (2012) 338–350.
[20] G.L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation, J. Power Sources 134 (2) (2004) 277–292.
[21] S. Santhanagopalan, R.E. White, Online estimation of the state of charge of a lithium ion cell, J. Power Sources 161 (2) (2006) 1346–1355.
[22] D.D. Domenico, A. Stefanopoulou, G. Fiengo, Lithium-ion battery state of charge and critical surface charge estimation using an electrochemical model-based extended Kalman filter, J. Dyn. Syst. Meas. Control 132 (6) (2010) 061302.
[23] M. Charkhgard, M. Farrokhi, State-of-charge estimation for lithium-ion batteries using neural networks and EKF, IEEE Trans. Ind. Electron. 57 (12) (2010) 4178–4187.
[24] H. Fang, Y. Wang, Z. Sahinoglu, T. Wada, S. Hara, State of charge estimation for lithium-ion batteries: an adaptive approach, Control Eng. Pract. 25 (2014) 45–54.
[25] S.J. Moura, N.A. Chaturvedi, M. Krstic, Adaptive PDE observer for battery SOC/SOH estimation via an electrochemical model, *ASME J. Dyn. Syst. Meas. Control.*
[26] R. Klein, N.A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, A. Kojic, Electrochemical model based observer design for a lithium-ion battery, IEEE Trans. Control Syst. Technol. 21 (99) (2012) 1–13.
[27] S. Santhanagopalan, R.E. White, State of charge estimation using an unscented filter for high power lithium ion cells, Int. J. Energy Res. 34 (2) (2010) 152–163.
[28] R.B. Gopaluni, R.D. Braatz, State of charge estimation in Li-ion batteries using an isothermal pseudo two-dimensional model, in: Proceedings of DYCOPS, Mumbai, India, 2013, pp. 135–140.
[29] G.-A. Nazri, G. Pistoia (Eds.), Lithium Batteries: Science and Technology, Kluwer, Boston, 2003.
[30] M. Doyle, T.F. Fuller, J. Newman, Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell, J. Electrochem. Soc. 140 (6) (1993) 1526–1533.
[31] T.F. Fuller, M. Doyle, J. Newman, Simulation and optimization of the dual lithium ion insertion cell, J. Electrochem. Soc. 141 (1) (1994) 1–10.
[32] V.R. Subramanian, V.D. Diwakar, D. Tapriyal, Efficient macro-micro scale coupled modeling of batteries, J. Electrochem. Soc. 152 (10) (2005) A2002–A2008.
[33] J. Newman, W. Tiedemann, Porous-electrode theory with battery applications, AIChE J. 21 (1) (1975) 25–41. Wiley Online Library.
[34] V.R. Subramanian, V. Boovaragavan, V.D. Diwakar, Toward real-time simulation of physics based lithium-ion battery models, Electrochem. Solid-State Lett. 10 (11) (2007) A255–A260.
[35] P.E. Gill, W. Murray, M.A. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization, SIAM Rev. 47 (1) (2005) 99–131.
[36] S.C. Patwardhan, S. Narasimhan, P. Jagadeesan, R.B. Gopaluni, S.L. Shah, Nonlinear bayesian state estimation: a review of recent developments, Control Eng. Pract. 20 (10) (2012) 933–953.
[37] A. Tulsyan, B. Huang, R.B. Gopaluni, J.F. Forbes, On simultaneous on-line state and parameter estimation in non-linear state-space models, J. Process Control 23 (4) (2013) 516–526.

[38] A. Tulsyan, J.F. Forbes, B. Huang, Designing priors for robust Bayesian optimal experimental design, J. Process Control 22 (2) (2012) 450–462.
[39] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, Boston, MA, 2004. Ch. A tutorial on Particle Filters.
[40] H.W. Sorenson, On the development of practical non-linear filters, Inf. Sci. 7 (C) (1974) 253–270.
[41] P.S. Maybeck, Stochastic Models, Estimation and Control Vol. 2, Academic Press, New York, 1982.
[42] A. Tulsyan, B. Huang, R.B. Gopaluni, J.F. Forbes, A particle filter approach to approximate posterior Cramér-Rao lower bound: the case of hidden states, IEEE Trans. Aerosp. Electron. Syst. 49 (4) (2013) 2478–2495.
[43] A. Tulsyan, B. Huang, R.B. Gopaluni, J.F. Forbes, Performance assessment, diagnosis, and optimal selection of non-linear state filters, J. Process Control 24 (2) (2014) 460–478.
[44] D. Crisan, B. Rozovskii, The Oxford Handbook of Non-linear Filtering, Oxford University Press, Oxford, 2011.
[45] A. Tulsyan, R.B. Gopaluni, S.R. Khare, Particle Filtering without Tears: a Primer for Begineers, Computers & Chemical Engineering, 2016. Accepted.