

# State of Charge Estimation in Li-ion Batteries Using an Isothermal Pseudo Two-Dimensional Model

R. Bhushan Gopaluni<sup>\*,\*\*</sup>

*\* Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z4, Email: bhushan.gopaluni@ubc.ca.*

Richard D. Braatz<sup>\*\*</sup>

*\*\* Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, United States of America 02142, Email: braatz@mit.edu.*

---

**Abstract:** The dynamics of Li-ion batteries are often defined by a set of coupled nonlinear partial differential equations called the pseudo two-dimensional model. It is widely accepted that this model, while accurate, is too complex for estimation and control. As such, the literature is replete with numerous approximations of this model. For the first time, an algorithm for state-of-charge estimation using the original pseudo two-dimensional model is provided. A discrete version of the model is reformulated into a state-space model by separating linear, nonlinear, and algebraic states. This model is high dimensional (of the order of tens to hundreds of states) and consists of implicit nonlinear algebraic equations. The degeneracy problems with high-dimensional state estimation are circumvented by developing a particle filter algorithm that sweeps in time and spatial coordinates independently. The implicit algebraic equations are handled by ensuring the presence of a ‘tether’ particle in the algorithm. The approach is illustrated through simulations.

*Keywords:* Li-ion Batteries, State Estimation

---

## 1 Li-ion Batteries

Li-ion batteries are widely used in a variety of low-power consumer gadgets and high-power automobile and aeronautical applications. Their popularity can be attributed to such properties as high energy density, slow material degradation, no memory effect, and low self-discharge. However, these batteries are known to become explosive when overcharged and at high coulomb rates due to overheating. For safe and reliable operation of Li-ion batteries, it is important to estimate the amount of charge remaining in the battery and the corresponding concentrations and potentials that it depends on, which is called the state of charge (SOC).

In Li-ion batteries, a complex combination of chemical kinetics, transport phenomena, and electrochemical reactions drive their dynamics. Only two variables, namely voltage and current, are measured. The state of charge is a function of the Li-ion concentration in the battery and therefore can not be easily estimated from direct voltage and current measurements. An obvious recourse for the estimation of SOC is to use a dynamic model.

Li-ion batteries have been modeled at different levels of abstraction from simple empirical models, such as equivalent circuit models, to complex molecular level models. As expected, the accuracy of these models roughly improves with their complexity. However, the more complex models

are difficult to simulate faster than real time and have not been very useful for estimation and control (Ramadesigan et al. (2012)). On the other hand, simple models are not accurate and are more weakly connected to the phenomena occurring in the battery. It has been shown in practice that the pseudo two-dimensional (P2D) model, developed by Doyle (1995), 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 but is still fairly complex for online simulation.

The P2D model consists of a set of nonlinear and coupled partial differential equations. A number of approximations of this model have been developed to convert the model into a simpler form 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 (Chaturvedi et al. (2010)). The only known high-speed implementation of P2D models is developed by Northrop et al. (2011). However, their method is not suitable for the application of standard estimation algorithms.

Approximate models have been used for state and parameter estimation in Li-ion batteries (Smith (2006)). Most notably, the extended Kalman filter has been widely used on approximate models (Plett (2004); Santhanagopalan and White (2006); Domenico et al. (2010); Charkhgard

and Farrokhi (2010)). In Klein et al. (2012) a nonlinear Luenberger observer was designed on an approximation of the P2D model that assumed uniform concentration across the battery. Despite a flurry of activity in the area of state estimation for P2D models, to the best knowledge of the authors, there is no known approach that can be applied on the P2D model without order-reducing model simplifications.

For the first time, an approach for state estimation in P2D models is developed without any simplifications—apart from the numerical approximations that are needed in any discrete time-based approach and one minor approximation on the solid concentration. The P2D model is reformulated as a state-space model with linear, nonlinear, and algebraic states. The uncertainty in the measurements and the model is characterized by introducing stochastic noise. The states in the models are estimated using a modified particle filtering algorithm that sweeps independently in time and spatial domains.

## 2 Pseudo two-dimensional model

A standard intercalation Li-ion battery consists of three main regions: a positive electrode, a separator, and a negative electrode. A thorough description of the various chemical, transport, and electrochemical phenomena that occur in this battery can be found in Nazri and Pistoia (2003). The positive and negative electrodes contain an electrolyte that transports the lithium ions and an active material that holds the lithium ions. The electrolyte and the active material is held together by fillers and other binding material. 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 on the positive electrode. This process reverses itself during a charging cycle. The dynamics of this process can be modelled by writing conservation of mass and conservation of charge equations. These equations can be written using the electrolyte concentration of lithium ions  $c_e(x, t) \in \mathbb{R}^+$ , the electrolyte potential  $\Phi_e(x, t) \in \mathbb{R}$ , the active material potential (also called the *solid potential*)  $\Phi_s(x, t) \in \mathbb{R}$ , and the concentration of lithium ion in the spherical particles of the active material  $c_s(x, r(x), t) \in \mathbb{R}^+$  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  $x$ .

### 2.1 The PDE model

The standard pseudo two-dimensional model can be derived using concentrated solution and porous electrode theories. The P2D model, with a minor modification to the solid diffusion equation, is shown compactly in Table 1 (Doyle (1995), Northrop et al. (2011)). This model consists of two coupled partial differential equations and an algebraic equation that define the conservation of mass in the three sections of the battery. The model consists of three partial differential equations that define the conservation of charge in these sections. Using  $i \in \{p, s, n\}$  to denote positive electrode, separator, and negative electrode,  $\epsilon_i \in (0, 1)$  and  $D_i \in \mathbb{R}^+$  are the porosity and the effective diffusion coefficient in section  $i$ ,  $\bar{c}_s(x, t) \in \mathbb{R}^+$

is the average Li-ion concentration in the solid particles,  $c_s^*(x, t) \in \mathbb{R}^+$  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,  $i_e(x, t) \in \mathbb{R}$  is the current in the electrolyte,  $\kappa_i(x) \in \mathbb{R}^+$  is the conductivity of electrolyte,  $R$  is the universal gas constant,  $T$  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 Li-ion flux at the surface of solid particles.

The Li-ion flux  $j_i$  plays an important role in determining the solution of these partial differential equations. This flux is usually given by Butler-Volmer kinetics:

$$j_i = 2k_i [c_e c_s^* (c_i^{max} - c_s^*)]^{0.5} \sinh \left( \frac{0.5}{F} RT (\Phi_s - \Phi_e - U_i) \right) \quad (1)$$

where  $k_i$  is a reaction constant,  $c_i^{max}$  is the maximum Li-ion concentration in the solid particles, and  $U_i(x, t) \in \mathbb{R}$  is the open-circuit voltage. A few important observations can be made about the PDEs 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. However, these boundary conditions can be enforced by finding a suitable solid potential  $\Phi_s$  and ionic flux  $j_i$ .

In Doyle (1995) the PDEs were iteratively solved using a first-order Taylor series approximation and in Northrop et al. (2011) the PDEs were solved using a coordinate transformation followed by collocation methods. Numerous numerical approximations can be derived; however, they are often not in the standard state-space form suitable for estimation and control algorithms.

A numerical approach is developed here by reformulating the discretized PDEs as a state-space model. The complete derivation is not provided due to space limitations, but a brief outline of the approximations and a pseudocode for its implementation are presented. The time and spatial coordinates are divided into  $M$  equal time intervals and  $N$  equal spatial intervals. The concentrations, potentials, and other variables at space-time coordinate  $(x, t)$  are denoted by the corresponding discrete coordinate  $(m, n)$  where  $m$  denotes discrete time and  $n$  denotes the discrete spatial location.<sup>1</sup> The bold-faced variables are obtained by concatenating their corresponding values at all the spatial locations. For instance,  $\mathbf{c}_e(m) = [c_e(m, 1), c_e(m, 2), \dots, c_e(m, N)]'$ . A Crank-Nicolson approximation of M1 leads to

$$\mathbf{c}_e(m, \boldsymbol{\theta}) = \mathbf{A}_c(\boldsymbol{\theta}) \mathbf{c}_e(m-1, \boldsymbol{\theta}) + \mathbf{B}_c(\boldsymbol{\theta}) \mathbf{j}_i(m, \boldsymbol{\theta}) \quad (2)$$

where  $\mathbf{A}_c(\boldsymbol{\theta})$  and  $\mathbf{B}_c(\boldsymbol{\theta})$  are appropriate matrices and  $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$  is a vector of parameters. The boundary conditions are numerically approximated and included in (2). Using first-order implicit numerical approximation, C2 and C3 can be written as

<sup>1</sup> Note that the same notation is used for continuous and discrete variables, with the meaning being clear from the context.

Table 1. Pseudo two-dimensional model. The positive electrode extends from  $x = 0$  ( $n = 1$ ) to  $x = l_p$  ( $n = N_p$ ), separator up to  $x = l_s$  ( $n = N_s$ ), and negative electrode up to  $x = l_n$  ( $n = N_n$ )

Conservation Equations	Boundary Conditions			
Mass	$x = 0$	$x = l_p$	$x = l_s$	$x = l_n$
(M1) $\epsilon_i \frac{\partial c_e}{\partial t} = \frac{\partial}{\partial x} \left( D_i \frac{\partial c_e}{\partial x} \right) + a_i(1-t_+)j_i$	$\frac{\partial c_e}{\partial x} = 0$	$-D_p \frac{\partial c_e}{\partial x} = -D_s \frac{\partial c_e}{\partial x}$	$-D_s \frac{\partial c_e}{\partial x} = -D_n \frac{\partial c_e}{\partial x}$	$\frac{\partial c_e}{\partial x} = 0$
(M2) $\frac{\partial \bar{c}_s}{\partial t} = -3 \frac{j_i}{R_i}$				
(M3) $c_s^* - \bar{c}_s = -\frac{R_i}{D_s} \frac{j_i}{5}$				
Charge				
(C1) $i_e = -\kappa_i \frac{\partial \Phi_e}{\partial x} + \frac{2\kappa_i RT}{F} (1-t_+) \frac{\partial \ln c_e}{\partial x}$	$\frac{\partial \Phi_e}{\partial x} = 0$	$-\kappa_p \frac{\partial \Phi_e}{\partial x} = -\kappa_s \frac{\partial \Phi_e}{\partial x}$	$-\kappa_s \frac{\partial \Phi_e}{\partial x} = -\kappa_n \frac{\partial \Phi_e}{\partial x}$	$\frac{\partial \Phi_e}{\partial x} = 0$
(C2) $\frac{\partial i_e}{\partial x} = a_i F j_i$	$i_e = 0$	$i_e = I$	$i_e = I$	$i_e = 0$
(C3) $\frac{\partial \Phi_s}{\partial x} = \frac{i_e - I}{\sigma_i}$				

$$\begin{pmatrix} \Phi_s(m, \theta) \\ \mathbf{i}_e(m, \theta) \end{pmatrix} = \mathbf{A}_\Phi(\theta) \mathbf{j}_i(m, \theta) + \mathbf{B}_\Phi(\theta) I(m), \quad (3)$$

where  $\mathbf{A}_\Phi(\theta)$  and  $\mathbf{B}_\Phi(\theta)$  are appropriate matrices. The PDE for the electrolyte potential in C1 is similarly approximated using first-order implicit equations to obtain

$$\Phi_e(m, \theta) = \mathcal{F}_\Phi(\mathbf{c}_e(m, \theta), \theta), \quad (4)$$

where  $\mathcal{F}_\Phi : \mathbb{R}_+^{n_p} \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ . The PDE for the average particle concentration M2 and the algebraic equation M3 can similarly be written as,

$$\bar{\mathbf{c}}_s(m, \theta) = \bar{\mathbf{c}}_s(m-1, \theta) + \bar{\mathbf{B}}(\theta) \mathbf{j}_i(m, \theta), \quad (5)$$

$$\mathbf{c}_s^*(m, \theta) = \bar{\mathbf{c}}_s(m, \theta) + \mathbf{B}^*(\theta) \mathbf{j}_i(m, \theta), \quad (6)$$

with appropriate matrices  $\bar{\mathbf{B}}(\theta)$  and  $\mathbf{B}^*(\theta)$ . The PDE model in Table 1 is represented by its discrete approximation through (2)–(6) and the flux vector based on (1):

$$\mathbf{j}_i = \mathcal{F}_j(\mathbf{c}_e(m, \theta), \mathbf{c}_s^*(m, \theta), \mathbf{i}_e(m, \theta), \Phi_s(m, \theta), \Phi_e(m, \theta), \theta) \quad (7)$$

where  $\mathcal{F}_j : \mathbb{R}_+^{n_p} \times \mathbb{R}^{5N} \rightarrow \mathbb{R}^N$ . The pseudocode in the next section solves these nonlinear implicit state-space equations.

## 2.2 Pseudo-Code for Numerical Solution

The numerical solution is based on the following observations of the PDEs in Table 1 : (i) M1, M2, M3, and C2 are linear PDEs if  $j_i(m, \theta)$  is known, (ii) C3 does not have a boundary condition and therefore an initial condition needs to be guessed, (iii) the first-order PDE in C2 can be treated as an initial value problem and solved iteratively until both boundary conditions are satisfied (this is a form of the shooting method to solve ordinary differential equations with two boundary conditions), and (iv)  $j_i(m, \theta)$  must satisfy (1). The PDEs are solved by starting with an initial guess for  $j_i(m, \theta)$ . Then  $\mathbf{c}_e(m, \theta)$  is obtained from (2).  $\bar{\mathbf{c}}_s$  and  $\mathbf{c}_s^*$  are obtained from (5) and (6). C3 does not have any boundary conditions and therefore  $\Phi_s(m, 1)$  and  $\Phi_s(m, N_s)$  are initially guessed and later updated based on Broyden's minimization algorithm. C2 is solved recursively as an initial value problem and  $j_i(m, \theta)$  is also updated using Broyden's algorithm until all of the boundary conditions are satisfied. This procedure is

shown in Algorithm 1 in the form of a pseudocode. ‘‘Broy’’

### Algorithm 1 Pseudocode

---

**Inputs:**

- 1:  $\theta, \Phi_s(0, 1), \Phi_s(0, N_s), \mathbf{c}_e, \bar{\mathbf{c}}_s, \mathbf{j}_i, I(m), \forall m$
- 2:  $T$  ▷ simulation time

**Initialize:**

- 3:  $\delta_1, \delta_2, \lambda_1, \lambda_2$  ▷ tolerance parameters
- 4:  $i_e(m, N_p), i_e(m, N_n), i_j(1), i_j(2)$
- 5: **for**  $m = 0$  to  $T$  **do** ▷ simulate for  $T$  seconds
- 6:   **if**  $m \geq 1$  **then**
- 7:      $\mathbf{j}_i(m, \theta) = \mathbf{j}_i(m-1, \theta)$  ▷ initialize flux
- 8:   **end if**
- 9:    $i_{bc}(1) \leftarrow i_e(m, N_p) - I(m)$  ▷ error in BC of C2
- 10:    $i_{bc}(2) \leftarrow i_e(m, N_n)$  ▷ error in BC of C2
- 11:   **while**  $|i_{bc}(1)| \geq \delta_1$  &  $|i_{bc}(2)| \geq \delta_2$  **do**
- 12:     ▷ ensure C2 BCs are met
- 13:      $i_j(1) \leftarrow i_e(m, N_p) - i_j(1)$
- 14:      $i_j(2) \leftarrow i_e(m, N_n) - i_j(2)$
- 15:     **while**  $|i_j(1)| \geq \lambda_1$  &  $|i_j(2)| \geq \lambda_2$  **do**
- 16:       **Solve:** equations (2),(5)
- 17:        $\kappa_i(x) \leftarrow \kappa_i(x)$
- 18:       **Solve:** equations (3), (4), (6)
- 19:        $\mathbf{j}_{temp}(m, \theta) \leftarrow$  right-hand side of (1)
- 20:       ▷ temporary variable
- 21:        $\mathbf{j}_i(m, \theta) \leftarrow$  Broy( $\|\mathbf{j}_i(m, \theta) - \mathbf{j}_{temp}(m, \theta)\|_2$ )
- 22:     **end while**
- 23:      $[\Phi_s(0, 1) \ \Phi_s(0, N_s)] =$  Broy( $|i_{bc}(1)| + |i_{bc}(2)|$ )
- 24:   **end while**
- 25: **end for**
- 26: **return**

---

is a function that implements derivative-free quasi-Newton steps to minimize its argument.

## 2.3 Discrete State-Space Reformulation

The discretized PDEs can be reformulated into a set of discrete state-space and algebraic equations. Define the following classes of states: (i)  $\mathbf{x}_m^e \triangleq \{\mathbf{c}_e(m, \theta), \bar{\mathbf{c}}_s\}$  are states that evolve linearly in time (with respect to  $\mathbf{j}_i$ , (ii)  $\mathbf{x}_m^{a1} \triangleq \{\Phi_s(m, \theta), \mathbf{i}_e(m, \theta)\}$  and  $\mathbf{x}_m^{a2} \triangleq \{\mathbf{c}_s^*(m, \theta)\}$

are states that follow an explicit linear algebraic equation, (iii)  $\mathbf{x}_m^{\text{a3}} \triangleq \{\Phi_e(m, \theta)\}$  are states that follow an explicit nonlinear algebraic equation, (iv)  $\mathbf{x}_m^{\text{n}} \triangleq \{\mathbf{j}_i(m, \theta)\}$  are states that follow the nonlinear algebraic equation in (7), and (v)  $u_m = \{I(m)\}$  is the input. With this notation, the state-space equation can be written as

$$\mathbf{x}_m^\ell = \begin{bmatrix} \mathbf{A}_c(\theta) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x}_{m-1}^\ell + \begin{bmatrix} \mathbf{B}_c(\theta) \\ \mathbf{B}(\theta) \end{bmatrix} \otimes \mathbf{x}_m^{\text{n}}, \quad (8)$$

$$\mathbf{x}_m^{\text{a1}} = \mathbf{A}_\Phi(\theta) \mathbf{x}_m^{\text{n}} + \mathbf{B}_\Phi u_m, \quad (9)$$

$$\mathbf{x}_m^{\text{a2}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x}_m^\ell + \mathbf{B}^*(\theta) \mathbf{x}_m^{\text{n}}, \quad (10)$$

$$\mathbf{x}_m^{\text{a3}} = \mathcal{F}_\Phi(\mathbf{x}_m^\ell, \theta), \quad (11)$$

$$\mathbf{x}_m^{\text{n}} = \mathcal{F}_j(\mathbf{x}_m^\ell, \mathbf{x}_m^{\text{a1}}, \mathbf{x}_m^{\text{a2}}, \mathbf{x}_m^{\text{a3}}, \theta). \quad (12)$$

The only measurements available for a battery are the current and voltage measurements. The voltage is given by

$$v(m) = \Phi_s(m, 0) - \Phi_s(m, N_n). \quad (13)$$

### 2.4 Uncertainty Characterization

The model in Table 1 is based on a number of theoretical and experimental approximations. It is natural to expect that this model will not represent the *exact* dynamics of a Li-ion battery. The discrepancy between the model and the true dynamics of the battery is *model uncertainty*, which can be due to structural errors in the model or due to parameter errors or both. The structural errors were accounted by adding random Gaussian noise to all of the states and the parameter errors were accounted by introducing Gaussian noise (or priors) on the parameters. A common practice in the state estimation literature is to characterize only structural uncertainty by arbitrarily adding Gaussian noise. However, a true characterization of uncertainty should include both structural and parameter errors. The parameter prior is assumed to be Gaussian and given by

$$p_\theta(\theta) = \mathcal{N}(\theta; \theta, \Sigma_\theta). \quad (14)$$

The stochastic states are defined by

$$\mathbf{X}_m^\ell | \mathbf{x}_m^\ell(\theta) \sim \mathcal{N}(\mathbf{0}, \Sigma_\ell). \quad (15)$$

$$\mathbf{X}_m^{\text{ai}} | \mathbf{x}_m^{\text{ai}}(\theta) \sim \mathcal{N}(\mathbf{0}, \Sigma_i) \quad \text{for } i = 1 \text{ to } 3. \quad (16)$$

$$\mathbf{X}_m^{\text{n}} | \mathbf{x}_m^{\text{n}}(\theta) \sim \mathcal{N}(\mathbf{0}, \Sigma_n). \quad (17)$$

and the stochastic measurements by

$$V_m | v_m \sim \mathcal{N}(\mathbf{0}, \Sigma_v) \quad (18)$$

where capital letters are used to denote random variables.

### 2.5 State of Charge

The state of charge is an important notion that quantifies the residual amount of charge in a battery. The residual charge can be quantified by the concentration of Li-ions in the solid particles. If  $c_{max}$  is the maximum possible concentration of Li-ions in the solid particles, then the bulk state of charge (SOC) in the positive electrode is

$$\mathcal{S}(t) = \frac{1}{l_p} \int_0^{l_p} \frac{\bar{c}_s(x, t)}{c_{max}} dx \quad (19)$$

and the corresponding approximate expected value is

$$\mathbb{E}[\mathcal{S}(m)] \approx \frac{\Delta x}{l_p c_{max}} \sum_{n=1}^{N_p} \int \bar{c}_s(m, n) p_{\bar{c}_s}(\bar{c}_s(m, n) | v_{1:m}) d\bar{c}_s. \quad (20)$$

where  $p_{\bar{c}_s}(\bar{c}_s(m, n) | v_{1:m})$  is the density function of average solid particle concentration.

## 3 Tethered Particle Filter

The state of charge is a function of other unmeasurable variables such as concentrations and potentials. These internal variables have to be estimated using only the current and voltage measurements. The related state estimation problem is rather challenged due to the nonlinear, implicit, and complex relations between the unmeasured variables. Particle filters are a class of approximate nonlinear estimators that can be made arbitrarily accurate by increasing the computational complexity. The traditional particle filtering algorithms are based on standard Markov-type state-space models. However, the model in (8)–(12) has a particular structure that includes implicit nonlinear equations. In a standard state-space model, the states are iteratively calculated at each time step with only one iteration per time step. The model (8)–(12) may require numerous iterations at each time step to solve the implicit nonlinear equations. Any particle approach will therefore require performing these iterations on each particle. Hence, a straightforward application of particle filtering algorithm, while possible, is computationally prohibitive. A novel particle filtering algorithm that reduces the computational complexity by utilizing the special structure of the state-space model (8)–(12) is presented next.

### 3.1 Decentralized and Tethered Particles

An inspection of the structure of the state-space model (8)–(12) indicates that the states are dependent on each other in a particular fashion that makes it possible to split the estimation problem into lower dimensional problems. First, define the state vector

$$\mathbf{x}_m = \{\mathbf{x}_m^\ell, \mathbf{x}_m^{\text{a1}}, \mathbf{x}_m^{\text{a2}}, \mathbf{x}_m^{\text{a3}}, \mathbf{x}_m^{\text{n}}\}. \quad (21)$$

The objective of the state estimation algorithm is to approximate the density function  $p_{x_m}(\mathbf{x}_m | v_{1:m})$ . The dimensionality of  $p_{x_m}(\mathbf{x}_m | v_{1:m})$  is  $7N$  where  $N$  is the number of discrete spatial locations. Due to the high dimensionality of the density function, any particle-based approach, with finite particles, will fail with degenerate particles. A common approach often used to avoid estimating high-dimensional density functions is to estimate marginal densities of the form  $p_{\bar{c}_s}(\bar{c}_s(m, n) | v_{1:m})$ . In fact, this marginal density is all that is needed to estimate the state of charge. The special model structure in (8)–(12) makes it possible to reduce the dimensionality by partitioning the states and splitting the filter density into a series of marginal density functions,

$$\begin{aligned} p_{x_m}(\mathbf{x}_m | v_{1:m}) &= p_{x_m^\ell}(\mathbf{x}_m^\ell | v_{1:m}, \mathbf{x}_m^{\text{n}}) p_{x_m^{\text{a1}}}(\mathbf{x}_m^{\text{a1}} | v_{1:m}, \mathbf{x}_m^{\text{n}}) \\ &\times p_{x_m^{\text{a2}}}(\mathbf{x}_m^{\text{a2}} | v_{1:m}, \mathbf{x}_m^\ell, \mathbf{x}_m^{\text{n}}) p_{x_m^{\text{a3}}}(\mathbf{x}_m^{\text{a3}} | v_{1:m}, \mathbf{x}_m^{\text{n}}), \end{aligned} \quad (22)$$

However, observing that C1, C2, and C3 have derivatives only in space, the dimensionality of the density functions can be further reduced. At any given time  $m$ , the model has a Markov structure in the spatial direction as well, so the marginal density functions in (22) can further be

reduced to marginal densities in the spatial direction. The optimal marginal estimators are shown in Table (2). For instance,  $D4$  is treated as a two-dimensional density function with states  $\Phi_e(m, n)$  and  $j_i(m, n)$  and a spatial filter sweep is performed using a modified particle filter algorithm.  $D1$ ,  $D2$ , and  $D3$  can be estimated rather easily with Kalman filters (see Table 2). A naive particle filter can theoretically be applied to  $D4$ . In addition to the large dimensionality of the states, the states are implicitly related and therefore a solution to the model at any given time will have to be recursive. A particle-based algorithm has to run these recursions as many times as the number of particles. To mitigate the computational complexity of this naive approach, the particles corresponding to the flux are not propagated; instead, a mean of the flux is propagated to the next time step. The mean flux particle (that is, the mean of the particles corresponding to  $\mathbf{x}_m^n$ ) is called a *tethered* particle.

**Definition 1. Tethered Particles:** Suppose that the state space consist of two sets of states,  $x_m$  and  $z_m$ , and let  $X_m^{(i)}$  and  $Z_m^{(i)}$  for  $i = 1$  to  $N$  be the corresponding particles at time  $m$ . The process of reducing the number of particles by collapsing  $Z_m^{(i)}$  into a predetermined particle  $\bar{Z}_m$  is called *tethering*.  $\bar{Z}_m$  is the *tethered particle*.

Once the flux particles are tethered, only one iteration is required to solve the implicit equations, which reduces the computational cost. The state-space model has a peculiar structure wherein all of the particles are expected to satisfy the boundary conditions exactly. This is achieved by modifying the importance-sampling density function of the 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 this approach for estimating the states is provided in Algorithm 2. The state of charge in (20) is estimated using a particle approximation of  $p_{\bar{c}_e}(\bar{c}_s(m, n)|V_{1:m})$  as follows,

$$\begin{aligned} \mathbb{E}[\mathcal{S}(m)] &\approx \frac{\Delta x}{l_p c_{max}} \sum_{n=1}^{N_p} \int \bar{c}_s(m, n) \hat{p}_{c_s}(c_s(m, n)|V_{1:m}) dc_s \\ &\approx \frac{\Delta x}{l_p c_{max}} \sum_{i=0}^P \sum_{n=1}^{N_p} \bar{c}_s(m, n)^{(i)} \end{aligned} \quad (23)$$

where  $P$  is the number of particles and  $\bar{c}_s(m, n)^{(i)}$  are the particle estimates of the average solid concentration.

## 4 Simulations

### 4.1 Simulation of Deterministic Model

The deterministic model is simulated using Algorithm 1 at a constant galvanostatic discharge current of  $I(m) = -30 \text{ A/m}^2$ . The initial guesses for the solid potential at its boundary conditions are chosen to be  $\Phi_s(0, 0) = 4.116 \text{ V}$  and  $\Phi_s(0, l_s) = 0.074 \text{ V}$ , 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 Li-ion fluxes leading to divergence of the proposed algorithm—in fact, no algorithm will converge under this scenario. The initial electrolyte concentration is assumed to be uniform across the battery at  $1000 \text{ mol/L}$ . The initial Li-ion flux was assumed to be  $10^{-6} \text{ mol/s-m}^2$ . The rest of the model parameters and empirical diffusion and conductivity

---

### Algorithm 2 Tethered Particle Filtering

---

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

- (1) Obtain  $\mathbf{x}_m^n$  by solving the deterministic model.
- (2) Estimate  $p_{x_m^\ell}(\mathbf{x}_m^\ell|v_{1:m}, \mathbf{x}_m^n)$  using a Kalman filter.

Generate  $P$  particles  $\mathbf{X}_m^{\ell, i}$  such that

$$\hat{p}_{x_m^\ell}(\mathbf{d}\mathbf{x}_m^\ell|v_{1:m}, \mathbf{x}_m^n) = \frac{1}{P} \sum_{i=1}^P \delta_{\mathbf{X}_m^{\ell, i}}(\mathbf{d}\mathbf{x}_m^\ell)$$

- (3) Estimate  $p_{x_m^{a_1}}(\mathbf{x}_m^{a_1}(j)|v_{1:m}, \mathbf{x}_m^n)$  and  $p_{x_m^{a_2}}(\mathbf{x}_m^{a_2}(j)|v_{1:m}, \mathbf{x}_m^n)$  using a spatial Kalman filter where  $\mathbf{x}_m^{a_1}(j)$  and  $\mathbf{x}_m^{a_2}(j)$  are the  $j$ th elements of the respective vectors. Generate  $P$  particles  $\mathbf{X}_m^{a_1, i}$  and  $\mathbf{X}_m^{a_2, i}$  such that

$$\hat{p}_{x_m^{a_1}}(\mathbf{d}\mathbf{x}_m^{a_1}(j)|v_{1:m}, \mathbf{x}_m^n) = \frac{1}{P} \sum_{i=1}^P \delta_{\mathbf{X}_m^{a_1, i}(j)}(\mathbf{d}\mathbf{x}_m^{a_1}(j))$$

$$\begin{aligned} \hat{p}_{x_m^{a_2}}(\mathbf{d}\mathbf{x}_m^{a_2}(j)|v_{1:m}, \mathbf{x}_m^n, \mathbf{X}_m^{\ell, i}) \\ = \frac{1}{P} \sum_{i=1}^P \delta_{\mathbf{X}_m^{a_2, i}(j)}(\mathbf{d}\mathbf{x}_m^{a_2}(j)) \end{aligned}$$

- (4) Estimate  $p_{x_m^{a_3}}(\mathbf{x}_m^{a_3}(j)|v_{1:m}, \mathbf{x}_m^n)$  using a particle filter where  $\mathbf{x}_m^{a_3}(j)$  is the  $j$ th element of  $\mathbf{x}_m^{a_3}$ . Generate  $N$  particles  $\mathbf{X}_m^{a_3, i}$  such that

$$\hat{p}_{x_m^{a_3}}(\mathbf{d}\mathbf{x}_m^{a_3}(j)|v_{1:m}, \mathbf{x}_m^n) = \frac{1}{P} \sum_{i=1}^P \delta_{\mathbf{X}_m^{a_3, i}(j)}(\mathbf{d}\mathbf{x}_m^{a_3}(j))$$

2: **end for**

---

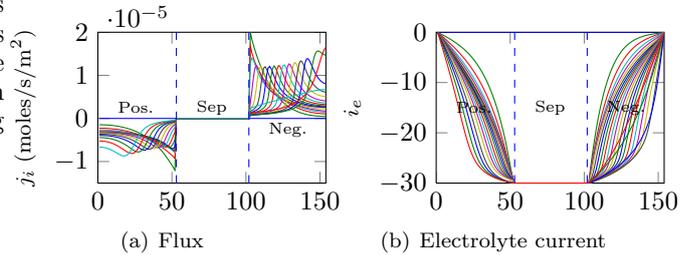


Fig. 1. (a) Li-ion flux, (b) electrolyte current at different times during the simulation along the length  $x$  of the battery: “Pos.” is the positive electrode, “Sep.” is the separator, “Neg.” is the negative electrode.

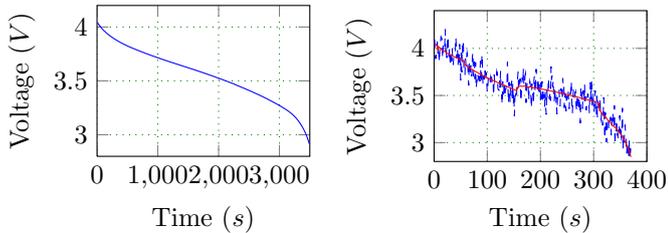
relations are taken from Northrop et al. (2011). The tolerance parameters are set to  $\delta_1 = \delta_2 = 0.1$  and  $\lambda_1 = \lambda_2 = 0.1$ . The proposed simulation algorithm runs through the discharge cycle of about 3500 s in real time in less than 3 s in Matlab (the run time of the algorithm depends on the resolution of the discrete spatial grid and the tolerance parameters). Figure 1 shows the evolution of fluxes and electrolyte current with time in the three sections of the battery. Figure 2a shows the discharge curve.

### 4.2 Simulation of Stochastic Model

The stochastic model is obtained by additive Gaussian noise as in (15)–(18). The standard deviation of the noise on the concentration is 1, on the potentials is 0.001, on the electrolyte current is 0.01, and on the voltage is 0.1. The discharge current is randomly switched between  $-35 \text{ A/m}^2$  and  $-25 \text{ A/m}^2$  with a Nyquist frequency of 0.01 Hz.

Table 2. Optimal estimators.

Density	Optimal Marginal Estimator	Dimension of Estimator	
		Full	Marginal
(D1) $p_{x_m^\ell}(\mathbf{x}_m^\ell   v_{1:m}, \mathbf{x}_m^n)$	Temporal & Spatial Kalman filter	$2N$	2
(D2) $p_{x_m^{a_1}}(\mathbf{x}_m^{a_1}   v_{1:m}, \mathbf{x}_m^n)$	Spatial Kalman filter	$2N$	2
(D3) $p_{x_m^{a_2}}(\mathbf{x}_m^{a_2}   v_{1:m}, \mathbf{x}_m^\ell, \mathbf{x}_m^n)$	Spatial Kalman filter	$N$	1
(D4) $p_{x_m^{a_3}}(\mathbf{x}_m^{a_3}   v_{1:m}, \mathbf{x}_m^\ell, \mathbf{x}_m^n)$	Spatial Particle Filter	$N$	1
(D5) $p_{x_m^n}(\mathbf{x}_m^n   v_{1:m})$	Spatial Particle Filter	$N$	1



(a) Deterministic simulation. (b) Stochastic simulation. (---) is the noisy measurement and (—) is the prediction.

Fig. 2. Discharge curves.

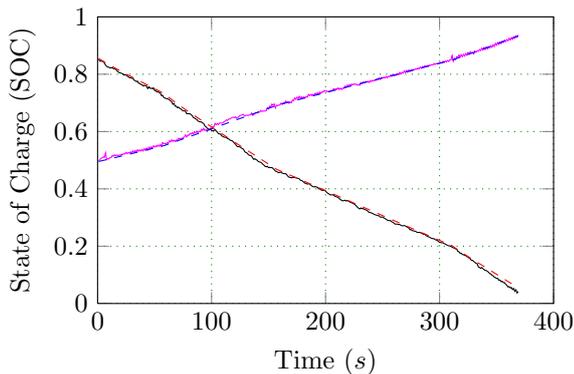


Fig. 3. The estimated mean of state of charge (—) vs. the true state of charge (- -).

Figure 2b shows the corresponding discharge curve and the predicted voltage from the estimator. The proposed approach is implemented with  $P = 2000$  particles. Figure 3 shows the estimated state of charge and the true state of charge obtained from the deterministic model. A common practice in the Li-ion battery literature is to assume Gaussian states and use an extended Kalman filter (EKF) for state-of-charge estimation. However, a plot of the time evolution of the state-of-charge density function clearly shows non-Gaussian behaviour (see Figure 4). These simulations suggest that, while EKF may work fine at low coulomb rates, at high coulomb rates the quality of SOC estimates may suffer unless powerful nonlinear estimators such as particle filters are used.

## 5 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

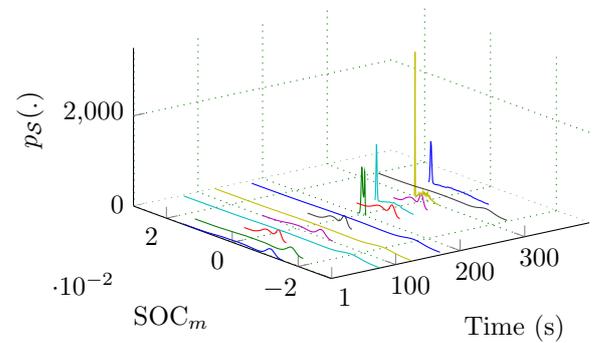


Fig. 4. The probability density function of the state of charge:  $SOC_m$  is the mean-removed state of charge.

algorithm. The algorithm uses a novel technique called ‘tethering’ to reduce computational complexity.

## References

- Charkhgard, M. and Farrokhi, M. (2010). State-of-charge estimation for lithium-ion batteries using neural networks and EKF. *IEEE Trans. on Industrial Electronics*, 57(12), 4178–4187.
- Chaturvedi, N., Klein, R., Christensen, J., Ahmed, J., and Kojic, A. (2010). Algorithms for advanced battery-management systems. *IEEE Control Systems Mag.*, 30(3), 49–68.
- Domenico, D.D., Stefanopoulou, A., and Fiengo, G. (2010). 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), 061302.
- Doyle, C.M. (1995). *Design and Simulation of Lithium Rechargeable Batteries*. Ph.D. thesis, University of California, Berkeley.
- Klein, R., Chaturvedi, N.A., Christensen, J., Ahmed, J., Findeisen, R., and Kojic, A. (2012). Electrochemical model based observer design for a lithium-ion battery. *IEEE Trans. on Control Systems Technology*, 21(99), 1–13.
- Nazri, G.A. and Pistoia, G. (eds.) (2003). *Lithium Batteries: Science and Technology*. Kluwer, Boston.
- Northrop, P.W.C., Ramadesigan, V., De, S., and Subramanian, V.R. (2011). Coordinate transformation, orthogonal collocation, model reformulation and simulation of electrochemical-thermal behavior of Lithium-ion battery stacks. *J. Electrochem. Soc.*, 158(12), A1461–A1477.
- Plett, G.L. (2004). Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *J. of Power Sources*, 134(2), 277–292.
- Ramadesigan, V., Northrop, P.W.C., De, S., Santhanagopalan, S., Braatz, R.D., and Subramanian, V.R. (2012). Modeling and simulation of lithium-ion batteries from a systems engineering perspective. *J. Electrochem. Soc.*, 159(3), R31–R45.
- Santhanagopalan, S. and White, R.E. (2006). Online estimation of the state of charge of a lithium ion cell. *J. of Power Sources*, 161(2), 1346–1355.
- Smith, K.A. (2006). *Electrochemical Modeling, Estimation and Control of Lithium Ion Batteries*. Ph.D. thesis, Pennsylvania State University, University Park.