System Identification of Nonlinear State-Space Battery Models

Wei He

weihe@calce.umd.edu

Advisor: Dr. Chaochao Chen

Department of Mechanical Engineering
University of Maryland, College Park

Background

- Lithium-ion batteries are power sources for electric vehicles (EVs).
- State of charge (SOC) estimation of batteries is important for the optimal energy control and residual range prediction of EVs.
- SOC is the ratio between the remaining charge (Q_{remain}) and the maximum capacity of a battery (Q_{max})

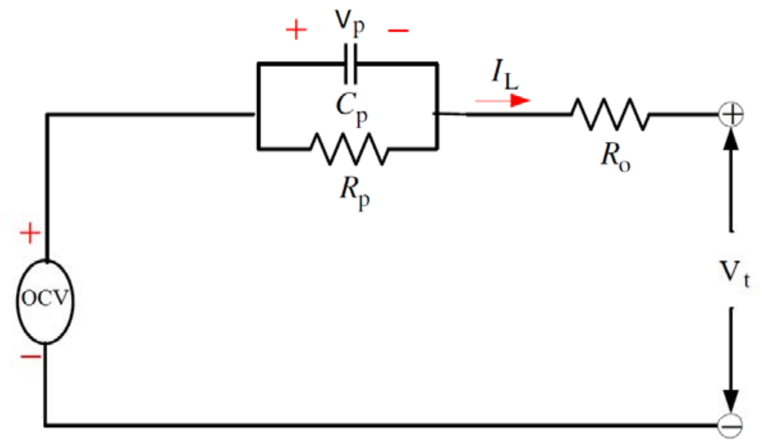
$$SOC = \frac{Q_{remain}}{Q_{max}}$$

Full: SOC = 100% Empty: SOC=0%

Equivalent Circuit Model of Batteries

- Equivalent circuit models have been used to model the relationship between SOC and the measurable battery parameters: current I_L and voltage U_t [1-3].

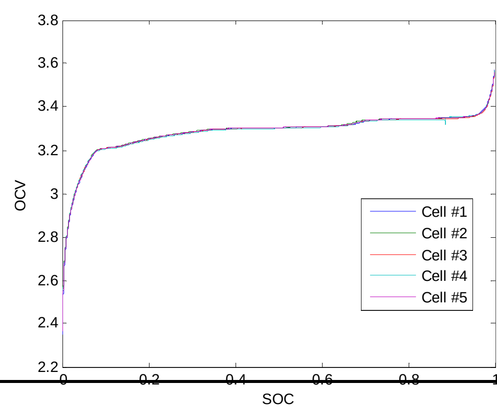
$$\begin{cases} \dot{V}_p = \dfrac{I_L}{C_p} - \dfrac{V_p}{C_p R_p} \\ V_t = OCV(SOC) - V_p - I_L R_0 \end{cases}$$



(Courtesy: Ref. [1])

where OCV is the open circuit voltage as a function of SOC, which can be determined by battery tests.

OCV-SOC



State-Space Representation

- Process function:

$$SOC_k = SOC_{k-1} - \frac{I_{L,k}\Delta t}{Q_{\max}}$$

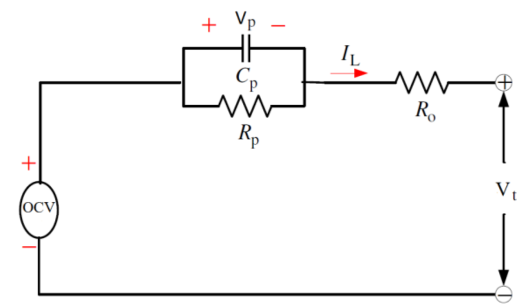
$$V_{p,k+1} = \exp\left(-\frac{\Delta t}{C_p R_p}\right) V_{p,k} + R_p\left[1-\exp\left(-\frac{\Delta t}{C_p R_p}\right)\right] I_{L,k}$$

- Measurement function:

$$V_{t,k} = OCV(SOC_k) - R_0 I_{L,k} - V_{p,k}$$

- Measured Signals:

$$\begin{bmatrix}\tilde{I}_L \\ \tilde{V}_t\end{bmatrix} = \begin{bmatrix} I_L \\ V_t \end{bmatrix} + \begin{bmatrix}\omega_I \\ \omega_2\end{bmatrix} \qquad \begin{matrix}\omega_I \sim N(0,\sigma_I) \\ \omega_V \sim N(0,\sigma_V)\end{matrix}$$



$$\begin{cases} \dot{V}_p = \dfrac{I_L}{C_p} - \dfrac{V_p}{C_p R_p} \\ V_t = OCV(SOC) - V_p - I_L R_0 \end{cases}$$

$$\theta = \left[R_p, C_p, R_0, Q_{\max}, \sigma_I, \sigma_v\right]$$

The model parameters will change with loading conditions and battery aging. Updating of the model parameters is necessary to ensure the accurate SOC estimation

Problem Formulation

- Estimate the unknown parameters θ in

$$x_{t+1} = f_t\left(x_t, u_t, \upsilon_t, \theta\right)$$
$$y_t = h_t\left(x_t, u_t, e_t, \theta\right)$$

based on the information in the measured input-output responses

$$U_N \triangleq \left[u_1, \ldots, u_N\right], Y_N \triangleq \left[y_1, \ldots, y_N\right]$$

using a maximum likelihood framework

$$\hat{\theta} = \arg\max_{\theta} p_\theta\left(Y_N\right)$$
$$\triangleq \arg\max_{\theta} L_\theta\left(Y_N\right)$$

Expectation Maximization (EM)

- **Expectation step (E step):** calculate the expected value of the log likelihood function, with respect to the conditional distribution of X_N given Y_N under the current estimate of the parameters θ_k [4]

$$Q(\theta,\theta_k) = E_{\theta_k}\left[L_\theta(X_N, Y_N) \mid Y_N\right] = \int L_\theta(X_N, Y_N) p_{\theta_k}(X_N \mid Y_N) dX_N$$

- **Maximization step (M step):** find the parameter that maximizes this quantity:

$$\theta_{k+1} = \arg\max_{\theta} Q(\theta,\theta_k)$$

If not converged, update k->k+1 and return to step 2

It has been approved in Ref. [4] that

$$L_{\theta_{k+1}}(Y_N) - L_{\theta_k}(Y_N) \geq Q(\theta_{k+1},\theta_k) - Q(\theta_k,\theta_k)$$

Expectation Maximization (EM)

$$Q(\theta,\theta_k) = E_{\theta_k}\left[L_\theta(X_N,Y_N)\,|\,Y_N\right] = \int L_\theta(X_N,Y_N)\,p_{\theta_k}(X_N\,|\,Y_N)\,dX_N$$

$$L_\theta(X_N,Y_N) = \log p_\theta(Y_N,X_N) = \log p_\theta(Y_N\,|\,X_N) + \log p_\theta(X_N)$$
$$= \log p_\theta(x_1) + \sum_{t=1}^{N-1}\log p_\theta(x_{t+1}\,|\,x_t) + \sum_{t=1}^{N}\log p_\theta(y_t\,|\,x_t)$$

Take $E_{\theta_k}[\cdot\,|\,Y_N]$

$$Q(\theta,\theta_k) = I_1 + I_2 + I_3$$

where

$$I_1 = \int \log p_\theta(x_1)\log p_{\theta_k}(x_1\,|\,Y_N)\,dx_1$$

$$I_2 = \sum_{t=1}^{N-1}\int\int \log p_\theta(x_{t+1}\,|\,x_t)\,p_{\theta_k}(x_{t+1},x_t\,|\,Y_N)\,dx_t\,dx_{t+1}$$

$$I_3 = \sum_{t=1}^{N}\int \log p_\theta(y_t\,|\,x_t)\,p_{\theta_k}(x_t\,|\,Y_N)\,dx_t$$

The particle smoother provides approximations for I_1 and I_3:

$$p_{\theta_k}(x_t\,|\,Y_N) = \sum_{i=1}^{N}\omega_{t|N}^{i}\,\delta\left(x_t - x_t^{i}\right)$$

Expectation Maximization (EM)

$$I_2 = \sum_{t=1}^{N-1} \int\int \log p_\theta\left(x_{t+1} \mid x_t\right) \underline{p_{\theta_k}\left(x_{t+1}, x_t \mid Y_N\right)} dx_t dx_{t+1}$$

using Bayesian and Markov property

$$p_{\theta_k}\left(x_{t+1}, x_t \mid Y_N\right) = p_{\theta_k}\left(x_t \mid x_{t+1}, Y_N\right) p_{\theta_k}\left(x_{t+1} \mid Y_N\right)$$

$$= \frac{p_{\theta_k}\left(x_{t+1} \mid x_t\right) p_{\theta_k}\left(x_t \mid Y_t\right)}{p_{\theta_k}\left(x_{t+1} \mid Y_t\right)} p_{\theta_k}\left(x_{t+1} \mid Y_N\right)$$

$$= \frac{p_{\theta_k}\left(x_{t+1} \mid x_t\right) p_{\theta_k}\left(x_t \mid Y_t\right)}{\int p_{\theta_k}\left(x_{t+1} \mid x_t\right) p_{\theta_k}\left(x_t \mid Y_t\right) dx_t} p_{\theta_k}\left(x_{t+1} \mid Y_N\right)$$

State equation

Particle filter

$$p_{\theta_k}\left(x_t \mid Y_t\right) = \sum_{i=1}^{N} \omega_t^i \delta\left(x_t - x_t^i\right)$$

Particle Smoother

$$p_{\theta_k}\left(x_{t+1} \mid Y_N\right) = \sum_{i=1}^{N} \omega_{t+1|N}^i \delta\left(x_{t+1} - x_{t+1}^i\right)$$

Expectation Maximization (EM)

- Particle smoothing approximations

$$Q\left(\theta,\theta_k\right) = I_1 + I_2 + I_3$$

$$I_1 \approx \widehat{I_1} = \sum_{i=1}^{M} w_{1|N}^{i} \log p_{\theta_k}\left(\tilde{x}_1^{i}\right)$$

$$I_2 \approx \widehat{I_2} = \sum_{t=1}^{N-1}\sum_{i=1}^{M}\sum_{j=1}^{M} \log w_{t|N}^{j}\, p_{\theta_k}\left(\tilde{x}_{t+1}^{j} \mid \tilde{x}_t^{i}\right)$$

$$I_3 \approx \widehat{I_3} = \sum_{t=1}^{N-1}\sum_{i=1}^{M} w_{t|N}^{i} \log p_{\theta_k}\left(y_t \mid \tilde{x}_t^{i}\right)$$

Particle EM Algorithm [4]

1. Set $k = 0$ and initialize θ_k
2. Expectation (E) Step:
   a) Run particle filter and particle smoother
   b) Calculate $Q_M\left(\theta,\theta_k\right) = \widehat{I}_1 + \widehat{I}_2 + \widehat{I}_3$
3. Maximization (M) Step:
   Compute: $\theta_{k+1} = \arg\max_{\theta} \widehat{Q_M}\left(\theta,\theta_k\right)$
4. Check the non-termination condition $Q\left(\theta_{k+1},\theta_k\right) - Q\left(\theta_k,\theta_k\right) > \varepsilon$
   If satisfied update $k \rightarrow k+1$ and return to step 2, otherwise terminate.

Particle Filter Algorithm [4-5]

1. Initialize particles, $\{x_0^i\}_{i=1}^{M} \sim P_\theta(x_0)$ and set $t = 1$.

2. Predict the particles by drawing M i.i.d samples according to

$$\tilde{x}_t^i \sim P_\theta\left(\tilde{x}_t \mid \tilde{x}_{t-1}^i\right),\ i = 1, \ldots, M$$

3. Compute the importance weights $\left\{w_t^i\right\}_{i=1}^{M}$

$$w_t^i = w\left(\tilde{x}_t^i\right) = \frac{P_\theta\left(y_t \mid \tilde{x}_t^i\right)}{\sum_{j=1}^{M} P_\theta\left(y_t \mid \tilde{x}_t^j\right)}, i = 1, \ldots, M$$

4. For each $j = 1, \ldots, M$ draw a new particle $\tilde{x}_t^i$ with replacement (resample) according to

$$P(x_t^j = \tilde{x}_t^j) = w_t^i, i = 1, \ldots, M$$

5. If $t < N$ increment $t \to t + 1$ and return to step 2, otherwise terminate.

Particle Smoother Algorithm [4-6]

1. Run the particle filter and store the predicted particles $\{x_t^i\}_{i=1}^M$ and their weights $\left\{w_t^i\right\}_{i=1}^M$, for $t = 1,\ldots,N$.

2. Initialize the smoothed weights to be the terminal filtered weights $\left\{w_t^i\right\}$ at time $t = N$.

$$w_{N|N}^i = w_N^i, \quad i = 1,\ldots,M$$

   and set $t = N\text{-}1$.

3. Compute the smoothed weights $\left\{w_{t|N}^i\right\}_{i=1}^M$ using the filtered weights $\left\{w_t^i\right\}_{i=1}^M$ and particles $\{\tilde{x}_t^i, \tilde{x}_{t+1}^i\}_{i=1}^M$ via

$$w_{t|N}^i = w_t^i \sum_{k=1}^M w_{t+1|N}^k \frac{P_\theta(\tilde{x}_{t+1}^k \mid \tilde{x}_t^i)}{v_t^k} \qquad where \;\; v_t^k = \sum_{k=1}^M w_t^i P_\theta(\tilde{x}_{t+1}^k \mid \tilde{x}_t^i)$$

4. Update $t \to t-1$. If t > 0 return to step 3, otherwise terminate.

Implementation

- Hardware
  - Dell Laptop with a 2.67G Hz Intel Core i7 CPU and 4 GB of RAM
- Software
  - Matlab

Validation

- Simulated data will be used to validate each component: the particle filter, particle smoother and the particle EM.
- Simulated data will be generated with the assumed exact values for the model parameters and states.
- Validation of particle filter and smoother
  - Model parameters are assumed to be known
  - State filtering and smoothing results will be compared with the true state values to verify the algorithm.
- Validation of particle EM
  - Model parameters and states are assumed be unknown

Project Schedule and Milestones

- Project proposal: October 5 2012
- Algorithm Implementation:
  - Particle filter and smoother: December 1 2012
  - The full algorithm (particle EM): February 1 2012
- Validation: March 15 2012
- Testing: April 15 2012
- Final Report: May 1 2012

Deliverables

- Codes
- Simulated data sets
- Presentations and reports

References

1. H. He, R. Xiong, and H. Guo, *Online estimation of model parameters and state-of-charge of LiFePO4 batteries in electric vehicles. Applied Energy*, 2012. ***89(1): p. 413-420.***
2. C. Hu, B.D. Youn, and J. Chung, *A Multiscale Framework with Extended Kalman Filter for Lithium-Ion Battery SOC and Capacity Estimation. Applied Energy, 2012*. ***92: p. 694-704.***
3. H.W. He, R. Xiong, and J.X. Fan, *Evaluation of Lithium-Ion Battery Equivalent Circuit Models for State of Charge Estimation by an Experimental Approach. Energies, 2011*. ***4(4): p. 582-598.***
4. T.B. Schön, A. Wills, and B. Ninness, *System identification of nonlinear state-space models. Automatica, 2011*. ***47(1): p. 39-49.***
5. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. Signal Processing, IEEE Transactions on, 2002*. ***50(2): p. 174-188.***
6. A. Doucet and A.M. Johansen, *A tutorial on particle filtering and smoothing: fifteen years later. Handbook of Nonlinear Filtering, 2009: p. 656-704.*