Journal of Power Sources 289 (2015) 105–113

Contents lists available at ScienceDirect

Journal of Power Sources

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

Online estimation of lithium-ion battery capacity using sparse Bayesian learning

Chao Hu*, Gaurav Jain, Craig Schmidt, Carrie Strief, Melani Sullivan

Medtronic Energy and Component Center, Brooklyn Center, MN 55430, USA

HIGHLIGHTS

- We propose a sparse Bayesian learning method for battery capacity estimation.
- The method is applicable to Li-ion batteries used in implantable medical device.
- Five features indicative of battery capacity are extracted from charge curves.
- RVM regression approximates mapping from feature space to capacity state space.
- Cycling data from lab and field cells are used to verify the performance.

ARTICLE INFO

Article history:
Received 4 February 2015
Received in revised form
20 April 2015
Accepted 28 April 2015

Keywords:
Sparse Bayesian learning
Relevance vector machine
Capacity estimation
Lithium-ion battery

ABSTRACT

Lithium-ion (Li-ion) rechargeable batteries are used as one of the major energy storage components for implantable medical devices. Reliability of Li-ion batteries used in these devices has been recognized as of high importance from a broad range of stakeholders, including medical device manufacturers, regulatory agencies, patients and physicians. To ensure a Li-ion battery operates reliably, it is important to develop health monitoring techniques that accurately estimate the capacity of the battery throughout its life-time. This paper presents a sparse Bayesian learning method that utilizes the charge voltage and current measurements to estimate the capacity of a Li-ion battery used in an implantable medical device. Relevance Vector Machine (RVM) is employed as a probabilistic kernel regression method to learn the complex dependency of the battery capacity on the characteristic features that are extracted from the charge voltage and current measurements. Owing to the sparsity property of RVM, the proposed method generates a reduced-scale regression model that consumes only a small fraction of the CPU time required by a full-scale model, which makes online capacity estimation computationally efficient. 10 years' continuous cycling data and post-explant cycling data obtained from Li-ion prismatic cells are used to verify the performance of the proposed method.

1. Introduction

Capacity, as an important indicator of the state of health (SOH) of Li-ion battery [1,2], measures the maximum amount of electric charge that a fully charged battery can deliver. Online estimation of battery capacity raises awareness of the present battery health condition and enables early detection of an incipient fault and timely implementation of maintenance actions. Recent literature reports a variety of approaches to estimating the capacity of Li-ion battery. In general, these approaches can be categorized into the adaptive filtering approach [1–6], the coulomb counting approach [7–9], the neural network (NN) approach [10–12] and the kernel regression approach [13–17].

Joint/dual extended Kalman filter (EKF) [1] and unscented Kalman filter [2,3] were employed to estimate the state of charge (SOC), capacity and/or resistance of Li-ion battery. To improve the performance of joint/dual estimation, adaptive measurement noise models of Kalman filter were developed to separate the sequence of SOC and capacity estimation [4]. A multiscale scheme with EKF [5] was developed that decouples the SOC and capacity estimation with respect to both the measurement- and time-scales and employs a state projection schedule for accurate and stable capacity estimation. Most recently, a data-driven multi-scale EKF algorithm was developed that leverages the fast-varying characteristic of SOC

* Corresponding author.
E-mail addresses: chao.x.hu@medtronic.com, huchaostu@gmail.com (C. Hu).

http://dx.doi.org/10.1016/j.jpowsour.2015.04.166
0378-7753/

and the slowly-varying characteristic of capacity, with an aim to achieve accurate SOC and capacity estimation in real-time [6].

The coulomb counting approach estimates the capacity by a simple integration of current over time. An enhanced coulomb counting approach was developed to estimate the capacity of a Li-ion cell with dynamic re-calibration after the cell is fully charged or discharged [7]. The coulomb counting approach offers a simple way to compute the capacity. But it requires accurate current measurement and, often, a full charge/discharge cycle to be exercised. This approach is typically used, in a well-controlled experiment, to provide a benchmark for evaluating a more sophisticated capacity estimation approach. Two recently developed approaches employed the coulomb counting approach to estimate the battery capacity based on the difference in the SOC values before and after partial charge/discharge [8,9].

The NN approach builds a network structure of interconnected "neurons" to model the dependency between the input features (e.g., cell terminal voltage, current and temperature) and the output (i.e., cell capacity). The recurrent NN were employed to estimate the two SOH-related parameters, namely the capacity and equivalent series resistance, of a high-power-density Li-ion cell based on the temperature, current, SOC variations and historical cell behavior [10] and achieved good accuracy in SOH estimation over hundreds of accelerated ageing cycles. The Hamming NN was applied to identifying the representative capacity pattern (from a set of training cells with known capacities) that most closely matches that of a testing cell whose capacity is unknown and to be estimated [11]. A very recent study developed a data-driven approach that integrates NN with dual EKF for online estimation of Li-ion battery [12]. In this approach, an NN model was built as a battery dynamic model that utilizes the SOC, current and capacity to estimate the terminal voltage, and dual EKF was used to jointly estimate both the SOC and capacity.

The kernel regression approach models the non-linear relationship between the measureable features and the cell capacity by way of kernel functions. Kernel regression techniques that were employed to estimate the capacity of Li-ion battery include support vector machine (SVM) [13,14], relevance vector machine (RVM) [15,16] and k-nearest neighbor (kNN) regression [17], all of which are machine learning techniques. SVM was used to predict the SOC, capacity fade and power loss of Li-ion battery based on the baseline data collected from reference performance tests [13]. In a more recent study, SVM was used in combination with load cycle counting to estimate the capacity of high-power Li-ion battery used in hybrid electric vehicles [14]. The performance of the developed capacity estimation approach was verified by performing a six-month cycling test with real-world driving profiles. RVM is a sparse Bayesian approach to kernel regression and performs regression in a probabilistic manner. The extreme sparsity of the RVM regression model allows one to make estimations for new observations in a highly efficient manner. An intelligent RVM-based method was proposed to estimate the SOH of Li-ion battery based on the sample entropy feature extracted from the discharge voltage measurements [15]. A Bayesian framework combining RVM and particle filter was proposed for tracking the capacity fade and predicting the remaining useful life of Li-ion battery [16]. Unlike SVM and RVM, kNN regression is a non-parametric learning technique. The technique possesses the unique property that no explicit training step is required. Recently, kNN regression was employed to capture the complex dependency of the capacity on the charge-related features, and particle swarm optimization was adapted to finding the optimal combination of feature weights for creating a kNN regression model with the minimum estimation error [17].

As mentioned above, researchers have developed a wide range of methods to estimate the capacity of Li-ion battery. However, further research is still needed to develop efficient, accurate and robust methods that track the capacity fade of Li-ion battery based on readily available measurements (i.e., voltage, current and temperature). This paper aims to apply a statistical learning method, RVM, to the task of estimating the capacity of Li-ion battery based on the voltage and current measurements during charge. This application involves two main steps, which are presented in Section 2 and summarized as follows:

1. First, five characteristic features that are indicative of the capacity are extracted from the charge curves. These features can be easily computed based on the voltage and current measurements during a charge cycle, where a battery is fully charged from a partially discharged state. See Section 2.1 for details.
2. Then, RVM is used to learn the relationship between the capacity of a battery and its charge-related features. A RVM regression model, after being trained offline, is used to infer the unknown capacity of a battery online from a set of charge-related features. Two desirable properties that RVM possesses are (i) the generalization, i.e., the over-fitting is avoided during offline training, and (ii) the sparsity, i.e., only a sparse set of training points, namely relevance vectors, are used for online inference. The generalization property ensures good accuracy in online inference, while the sparsity property improves computational efficiency. To the best of our knowledge, the present study is the first to investigate the use of the sparse Bayesian learning method to infer the battery capacity from the charge data. See Section 2.2 for details.

The experimental verification of this application was accomplished by analyzing data from (i) a 10 years' continuous cycling test on eight Li-ion cells that were manufactured in 2002 and (ii) a post-explant cycling test on twenty-three Li-ion cells with 4–7 implant years. Section 3 presents and discusses the verification results. The paper is concluded in Section 4.

2. Technical approach

The aim of the method described in this section is to estimate the capacity of a Li-ion battery cell based on the basic measurements (i.e., voltage and current) collected from the cell during charge. Specifically, we intend to convert the voltage and current measurements during a charge cycle to a feature vector, $\mathbf{x}$, and use a trained RVM expert to infer the cell capacity based on the feature vector. Section 2.1 briefly describes the composition of the feature vector. Section 2.2 discusses the use of RVM to build a sparse kernel regression model that approximates the non-linear mapping from the multi-dimensional feature space to the one-dimensional capacity state space.

2.1. Feature extraction

Typical voltage and current curves of a Li-ion battery cell during a charge cycle is shown in Fig. 1 [17]. The cell enters the charge stage after being partially discharged to a certain SOC level. A standard charge protocol comprises of two charge steps, the constant current (CC) charge step and the constant voltage (CV) charge step. In the CC charge step, a charge current is kept at a constant until the cell terminal voltage rises up to a predefined voltage limit, V_{max}. Immediately after the cell terminal voltage reaches V_{max}, the charge process transitions from the CC charge step to the CV charge step. In the CV charge step, the cell terminal voltage is kept unchanged at V_{max} until a predefined time limit is reached. Accordingly, the total charge capacity consists of two parts, the CC charge

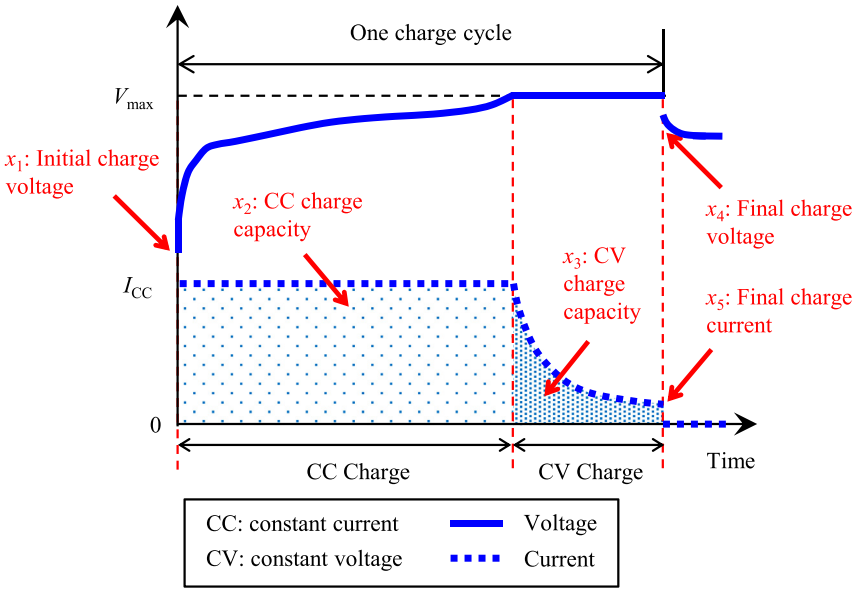

Fig. 1. Five charge-related features in an illustrative charge cycle [17].

capacity and the CV charge capacity. It is noted that, in practice, a Li-ion battery cell may not experience a complete charge (i.e., after the cell is full discharged). If the user charges the cell before its complete depletion, the cell then starts charge from a partially discharged state, and the charge cycle shown in Fig. 1 becomes a partial charge cycle.

Five charge-related features (see Fig. 1) that are indicative of the cell capacity are extracted from the voltage and current curves and are fed as inputs to the RVM regression model. These five features, namely the initial charge voltage (x_1), the CC charge capacity (x_2), the CV charge capacity (x_3), the final charge voltage (x_4) and the final charge current (x_5), are the elements of a feature vector, $\mathbf{x}$. Detailed explanations of the five features can be found in Ref. [17].

2.2. Relevance vector machine

RVM was developed by Tipping [18] as a sparse Bayesian version of kernel regression. This technique adopts a Bayesian framework to probabilistically train a kernel regression model, which leads to a sparse probabilistic regression model. In this study, we train a RVM regression model such that it learns a mapping from the multi-dimensional feature space to the one-dimensional capacity state space.

2.2.1. Model description

Suppose we have a training data set $\{\mathbf{x}_i, y_i\}$, $i = 1, 2, \ldots, M$, sampled from a scalar-valued function with additive zero mean Gaussian noise ε with a variance σ^2. This training data set possesses M feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$, with each being an N-dimensional vector, and M responses, $y_1, y_2, \ldots, y_M$. Note that, in the context of capacity estimation of Li-ion battery, y represents the capacity of a battery cell at a specific charge/discharge cycle. RVM is a special case of a linear model to approximate this data set, expressed as [18]

$$h(\mathbf{x}) = \sum_{i=1}^{M} \omega_i K(\mathbf{x}, \mathbf{x}_i) + \omega_0 \tag{1}$$

where $\mathbf{x}$ is a feature vector, h is the scalar-valued function that approximates the relationship between the response and the feature vector, $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_M)^T$ is a kernel weight vector, and $K(\mathbf{x}, \mathbf{x}_i)$ is a kernel function typically centered at the training point $\mathbf{x}_i$. The selection of an appropriate kernel parameter is an important step that ensures good prediction accuracy of a trained RVM model. For most kernel types, the key parameter is the so-called kernel width, a smoothing parameter that has a dramatic impact on the prediction accuracy and model sparsity. For example, the kernel width of a Gaussian kernel function employed in this study is the standard deviation parameter of a Gaussian distribution function, as shown in the following expression:

$$\phi(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{r^2} \|\mathbf{x} - \mathbf{x}_i\|^2 \right) \tag{2}$$

where r is the kernel width, the value of which was empirically set in this study by maximizing the prediction accuracy of a trained RVM model on the testing data set (see Section 3.4.1). It can be observed from Eq. (2) that the Gaussian kernel function decays as an exponential function of the Euclidean distance between two input vectors $\mathbf{x}$ and $\mathbf{x}_i$, and the decay rate is controlled by the kernel width r.

Training a RVM regression model determines the weight vector $\boldsymbol{\omega}$ of the model. Assuming the independence of $\{y_i\}$, $i = 1, \ldots, M$, we can express the likelihood of the observed data as

$$p\left(\mathbf{y} \middle| \boldsymbol{\omega}, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-M/2} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\omega}\|^2 \right) \tag{3}$$

where $\mathbf{y} = (y_1, \ldots, y_M)^T$, and $\boldsymbol{\Phi}$ is an $M \times (M + 1)$ design matrix constructed with the training input vectors. $\boldsymbol{\Phi}_{ij} = 1$ for $j = 1$, and $\boldsymbol{\Phi}_{ij} = K(\mathbf{x}_i, \mathbf{x}_{j-1})$ for $j = 2, \ldots, M + 1$.

To evaluate the unknown parameters in Eq. (3) from a Bayesian perspective, a zero-mean Gaussian prior distribution can be specified over $\boldsymbol{\omega}$, in such a way that a unique variance parameter is assigned to each weight, as

$$p(\boldsymbol{\omega}|\boldsymbol{\alpha}) = \prod_{i=0}^{M} N\left(\omega_i \middle| 0, \ \alpha_i^{-1}\right) \tag{4}$$

where $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_M)^T$ is a vector consisting of $M + 1$ hyper-

parameters that are treated as independent random variables. To specify this hierarchical Bayesian inference model, prior distributions for $\boldsymbol{\alpha}$ and the noise variance σ^2 must be defined. For the scale parameters $\boldsymbol{\alpha}$ and σ^2, it is common to use Gamma prior distributions as

$$p(\boldsymbol{\alpha}) = \prod_{i=0}^{M_s} \text{Gamma}(\alpha_i|a,\ b)$$
$$p(\beta) = \text{Gamma}(\beta|c,\ d) \tag{5}$$

where $\beta = \sigma^{-2}$, $\text{Gamma}(\alpha|a,b) = \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-b\alpha}$ with $\Gamma(a)$ being the gamma function, and a, b, c and d are the hyper-parameters and are set to small values to form a flat Gamma prior.

With the prior defined, the posterior distribution over the weights is given by the Bayesian inference as

$$p\left(\boldsymbol{\omega}\middle|\mathbf{y},\boldsymbol{\alpha},\sigma^2\right) = (2\pi)^{-(M+1)/2}|\boldsymbol{\Sigma}|^{-1/2} \times \exp\left(-\frac{1}{2}(\boldsymbol{\omega}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\omega}-\boldsymbol{\mu})\right) \tag{6}$$

where the posterior mean vector of the weights $\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}\mathbf{h}$, and the covariance matrix $\boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}+\mathbf{A})^{-1}$ with $\mathbf{A} = \text{diag}(\alpha_0, \ldots, \alpha_M)$. Appropriate iterative optimization methods [18], such as marginal likelihood optimization, expectation maximization (EM) algorithms, and incremental optimization algorithms, can be employed to find the hyper-parameter posterior modes or most probable values $\boldsymbol{\alpha}_{MP}$ and σ^2_{MP} that maximize $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{h}) \propto p(\mathbf{h}|\boldsymbol{\alpha}, \sigma^2)\, p(\boldsymbol{\alpha})\, p(\sigma^2)$. It is often found in practice that the optimal values of many hyper-parameters α_i are infinite. From Eq. (6), this leads to the consequence that the posterior probability distributions $p(\omega_i|\mathbf{y},\alpha_i,\sigma^2)$ of the associated weights are highly peaked at zero, thus effectively 'pruning' the corresponding training input vectors and deeming them as 'irrelevant' vectors. The remaining input vectors with non-zero weights are termed *relevance vectors* which form the final sparse regression model. Often, the number of the relevance vectors M_R is far less than the total number M of the training input vectors, i.e., $M_R << M$.

2.2.2. Capacity estimation using RVM

In this study, the output of a RVM regression model is the capacity estimate h of a battery cell and the inputs are the five charge-related features $\mathbf{x}$ (see Section 2.1) of the cell. The RVM regression model, after being trained, approximates the non-linear mapping from the multi-dimensional feature space to the one-dimensional capacity state space.

The process of capacity estimation with a trained RVM regression model is shown in Fig. 2. First, a testing vector $\mathbf{x}_t$, which consists of the five features described in Section 2.1, is extracted from the voltage and current measurements during a partial charge cycle. Then, this testing vector is compared to the relevance vectors, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathrm{X}_{\mathrm{M_R}}$, using the kernel function K (i.e., the Gaussian kernel function). Finally, the comparison results are combined in a weighted-sum form to give a capacity estimate h. From Eq. (1), the capacity estimate can be expressed as

$$h(\mathbf{x}_t) = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{K} \tag{7}$$

where $\boldsymbol{\omega}$ is a vector of the weights of the relevance vectors, and $\mathbf{K}$ is a vector of the kernel evaluations. As shown in Eq. (6), the posterior distributions of the weights $\boldsymbol{\omega}$ are Gaussian with the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, both of which can be computed once $\boldsymbol{\alpha}_{MP}$ and σ^2_{MP} are found. Thus, $h(\mathbf{x}_t)$ is also Gaussian distributed with the mean and variance expressed as

$$\mu_h = \boldsymbol{\mu}^{\mathrm{T}}\mathbf{K} \tag{8}$$

$$\sigma_h^2 = \sigma^2_{MP} + \mathbf{K}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{K} \tag{9}$$

It can be observed from the above equations that (i) the mean of the capacity estimate is computed by weighting the kernel evaluations by the posterior mean weights and (ii) the variance of the estimate consists of two components: the first being the estimated variance of the data noise and the second resulting from the variance in the estimation of the weights. It is noted that RVM doesn't just estimate the mean of the capacity but generates a probability distribution or probability density function (PDF) of the capacity, which is one of the principal advantages of RVM over SVM, kNN regression and other non-probabilistic machine learning techniques.

3. Experimental results

Section 2 describes the proposed method to estimate the capacity of a Li-ion battery cell based on the readily available measurements (i.e., voltage and current) acquired from the cell during a partial charge cycle. The basic idea is to extract the five characteristic features (see Section 2.1) from the measurements and to train an RVM regression model (see Section 2.2) such that the model learns the relationship between the features and the capacity. The performance of the proposed method is evaluated based on data from (i) a 10 years' continuous cycling test (i.e., repeated charge/discharge cycles) on eight Li-ion prismatic cells and (ii) a post-explant cycling test (i.e., single charge/discharge cycle) on twenty-three field cells. In this evaluation, the capacity estimate of a cell at a specific charge/discharge cycle is obtained through two sequential steps: (i) the voltage and current measurements from the cell during a partial charge cycle are converted into a feature vector ($\mathbf{x}$); and (ii) the feature vector is fed into a trained RVM model, which then generates the capacity estimate at the specific cycle.

This section reports the results of this evaluation. Section 3.1 presents the test procedure along with the cycling performance of these cells. Section 3.2 describes the generation of partial charge curves from the full charge curves of these cells, and the extraction of feature data from the generated partial charge curves. Section 3.3 defines two error measures for the performance evaluation. The capacity estimation results by the method are presented and discussed in Section 3.4.

3.1. Test procedure and cycling data

Li-ion cells were constructed in hermetically sealed prismatic cases between 2002 and 2012 and subjected to full depth of charge/discharge cycling under 37 °C in the lab [9,17]. The cycling test was conducted with the following parameter settings: (i) the charge rate (I_{CC}) for the CC charge was $C/6$; (ii) the charge cutoff voltage (V_{max}) was 4.075 V; (iii) the time duration of the CV charge was 30 min; and (iv) the discharge rate was $C/150$ or a nominally weekly discharge rate. The discharge capacities of these lab cells are plotted against the time on test in Fig. 3. Please note that, for confidentiality reasons, the discharge capacity of a cell in Fig. 3 and in the discussions thereafter is presented after being normalized by the beginning-of-life (BOL) discharge capacity of the cell. This plot also includes the mean curve of capacity fade as predicted by a battery longevity model [19]. Two observations can be made from the cycling data acquired from the lab cells: (i) the eight 2002 lab cells still maintained about 80% of their initial capacities after 10 years of repeated cycling; and (ii) despite the manufacturing variability over

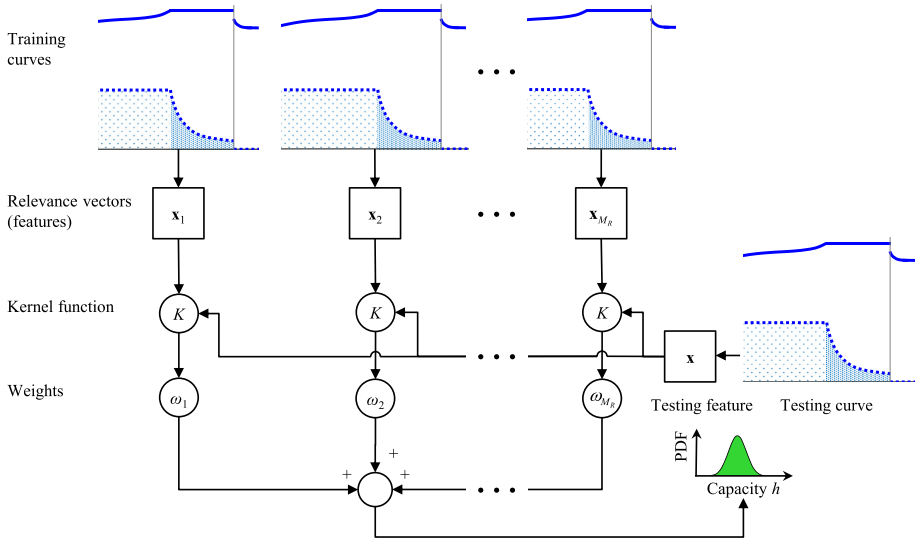

Fig. 2. Capacity estimation process with a trained RVM regression model.

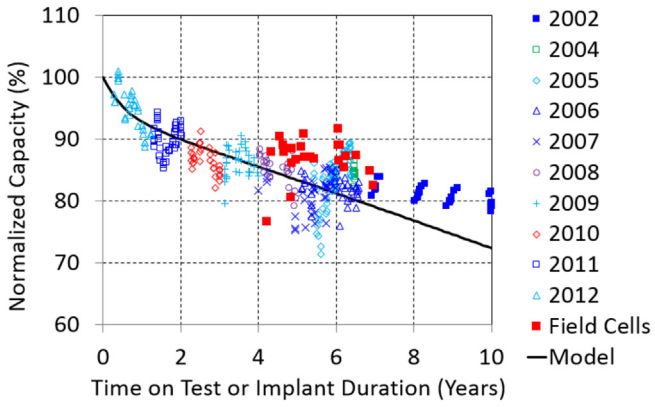

Fig. 3. Normalized capacity vs. time on test or implant duration for lab cells, field cells, and model mean.

the years (2002–2012), the capacity fade performance of these cells appears to show reasonably good agreement with the model prediction.

Also plotted in Fig. 3 are the discharge capacities of twenty-three field cells from explanted devices. The estimated implant durations of these devices range from 4 years to 7 years. After being extracted from the devices, these field cells were cycled using the same test procedure as the lab cells, which allowed the discharge capacities of these field cells to be characterized. Overall speaking, the cycling performance of the field cells in the real application matches the lab cells and the model.

3.2. Feature data generation

The 10 years' cycling data from the eight 2002 lab cells and the post-explant cycling data from the twenty-three field cells were used to verify the effectiveness of the proposed method in capacity estimation. In order to simulate actual use cases where a cell starts charge at a partially discharged state and ends up being fully charged, we generated a partial charge curve from a full charge curve by truncating the full charge curve below a pre-assigned initial charge voltage. The five features detailed in Section 2.1 were then extracted from the partial charge curves that had been generated. The initial charge voltage pre-assigned to a charge cycle was randomly generated from a uniform distribution between a lower bound V_{LB} and an upper bound V_{UB}. Two settings were considered for the bounds of the uniform distribution: (i) Setting 1 – Low initial SOC ($V_{LB} = 3.65$ V or roughly 3% SOC, and $V_{UB} = 3.80$ V or roughly 23% SOC) and (ii) Setting 2 – High initial SOC ($V_{LB} = 3.80$ V or roughly 23% SOC, and $V_{UB} = 3.85$ V or roughly 43% SOC). The two SOC ranges (3%–23% and 23%–43%) were chosen to investigate how the initial SOC level affects the accuracy in capacity estimation. Note that Setting 2 represents the scenario where a cell undergoes a shallower discharge, as compared to the scenario in Setting 1, before a subsequent charge starts. As shown in Fig. 4, three initial voltage values (i.e., V_1, V_2 and V_3) randomly sampled from the uniform distribution result in the generation of three partial charge curves from the same charge curve, and the subsequent extraction of three feature vectors (i.e., $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$), accompanied by the target capacity values (i.e., y_1, y_2, and y_3). Note that $y_1 = y_2 = y_3$, since the three partial charge curves are generated from the same charge curve. In addition to the two different settings in the initial SOC distribution, this study also considered two scenarios with respect to the current sensor error: (i) Scenario 1 – No bias in current measurement and (ii) Scenario 2 – 2% positive bias in current measurement. In Scenario 1, no change was made to the feature vectors, whereas in Scenario 2, the two capacity-related feature vectors, the CC and CV charge capacities (x_2 and x_3), were artificially increased by 2% to account for the effect of the current sensor error in the capacity calculation. The two scenarios were employed to study how the current sensor error affects the accuracy in capacity estimation. The two settings and two scenarios resulted in four different combinations (or cases) for investigation, listed as follows:

- Case 1 – Low initial SOC (3%–23%), no bias in current measurement;
- Case 2 – Low initial SOC (3%–23%), 2% positive bias in current measurement;
- Case 3 – High initial SOC (23%–43%), no bias in current measurement;
- Case 4 – High initial SOC (23%–43%), 2% positive bias in current measurement.

For each of the four cases, the feature data were generated using

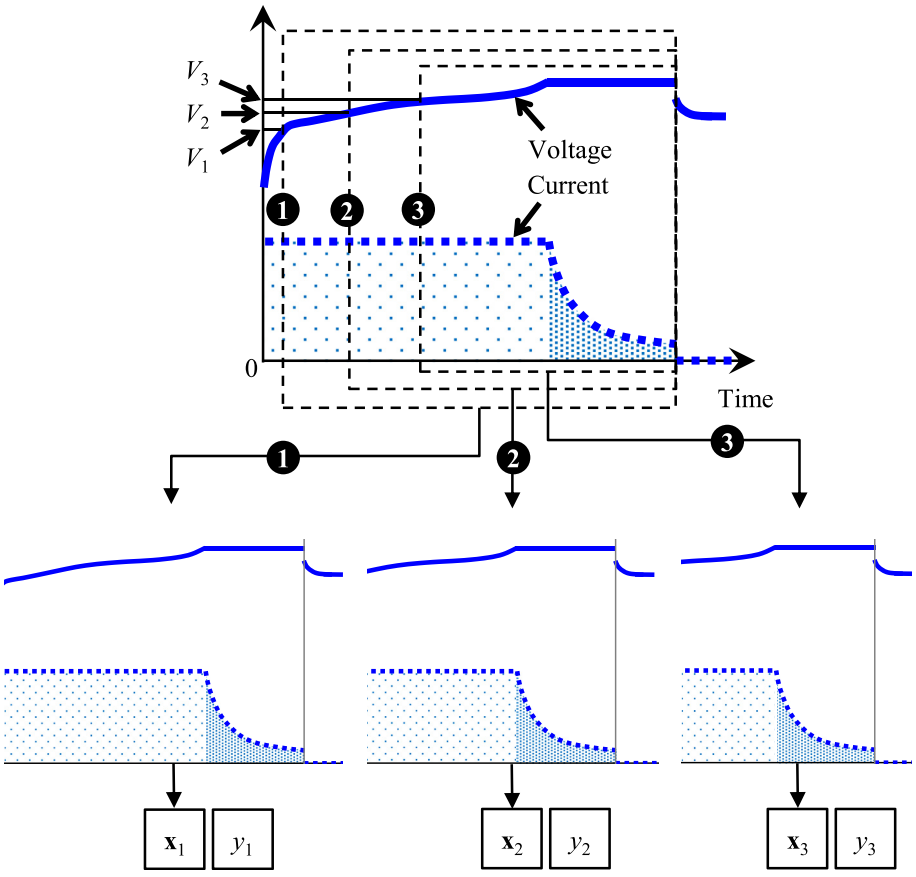

Fig. 4. Generation of partial charge curves and subsequent extraction of feature vectors. From the full charge curve of a cell with a known capacity y, different partial charge curves may be randomly sampled and converted to feature vectors ($\mathbf{x}_i$), accompanied by the target displacements $y_i\,(=y)$. As a tutorial example, this graph shows the process of generating three different partial curves from a single full charge curve.

the charge curves at every 20 charge/discharge cycles from each of the eight 2002 lab cells. Table 1 summarizes the numbers of charge/discharge cycles, charge cycles used for feature data generation and extracted feature vectors from the lab cells. For training a RVM regression model (i.e., determining the unknown weight vector $\boldsymbol{\omega}$), the M extracted feature vectors, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$, and their target capacity values, $y_1, y_2, \ldots, y_M$, from the training cells constitute a training data set $\{\mathbf{x}_i, y_i\}$, $i = 1, 2, \ldots, M$. The training data set was used to build the likelihood function in Eq. (3), one critically important element for RVM training. For testing a RVM regression model (i.e., evaluating the accuracy of a trained RVM model), the extracted feature vector of a testing cell are fed into the trained RVM model, which then produces a capacity estimate of the cell. Before applying the RVM regression method, the feature data were normalized to [0,1].

3.3. Definition of error measures

RVM training was implemented to learn the posterior distributions of the kernel weights $\boldsymbol{\omega}$ from the generated feature data (see Section 3.2) and the corresponding capacity values. The accuracy of a trained RVM regression model was evaluated by using the k-fold cross validation (CV). In this study, the complete feature data set $\mathbf{X}$ consists of eight mutually exclusive subsets or folds $\mathbf{X}_1$, $\mathbf{X}_2, \ldots, \mathbf{X}_8$ that were respectively obtained from the eight lab cells. In each CV trial, of the eight subsets, one was used as the test set and the other seven subsets were put together as a training set. The CV process was performed eight times (i.e., the total number of CV trials is eight), with each of the eight subsets left out exactly once as the test set. Thus, all the data points in the complete data set were used for both training and testing. Let $\mathbf{I}_l = \{i\colon \mathbf{x}_i \in \mathbf{X}_l\}$, $l = 1, 2, \ldots, 8$, denote the index set of the data points that construct the subset $\mathbf{X}_l$. Then the CV root mean square (RMS) error is computed as the root square of the average error over all the eight CV trials, expressed as

$$\varepsilon_{RMS} = \sqrt{\frac{1}{U}\sum_{l=1}^{8}\sum_{i\in\mathbf{I}_l}\left(\mu_{h,\mathbf{X}\backslash\mathbf{X}_l}(\mathbf{x}_i) - y(\mathbf{x}_i)\right)^2} \tag{10}$$

where U is the number of feature vectors for the CV, $\mu_{h,\mathbf{X}\backslash\mathbf{X}l}$ is the mean predicted capacity by the RVM regression model built with

Table 1
Numbers of charge/discharge cycles, charge cycles used for feature data generation and feature vectors from eight 2002 lab cells.

	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8
No. of cycles	717	701	714	704	718	713	715	727
No. of cycles used for feature generation	36	35	36	35	36	36	36	37
No. of feature vectors	108	105	108	105	108	108	108	111

the complete data set $\mathbf{X}$ excluding the subset $\mathbf{X}_l$, and $y(\mathbf{x}_i)$ is the measured capacity of $\mathbf{x}_i$. The CV maximum (Max) error is computed as the maximum absolute error over all the eight CV trials, expressed as

$$\varepsilon_{Max} = \max_{1\le l\le 8}\left(\max_{i\in \mathbf{I}_l}\left|\mu_{h,\mathbf{X}\setminus\mathbf{X}_l}(\mathbf{x}_i) - y(\mathbf{x}_i)\right|\right) \quad (11)$$

The error formulae in Eqs. (10) and (11) indicate that all the U feature vectors in the complete data set $\mathbf{X}$ are used for both training and testing, and each feature vector is used for testing exactly once and for training seven times.

3.4. Results and discussion

3.4.1. Capacity estimation results on lab cells

To study whether the use of RVM truly improves the accuracy in capacity estimation over existing methods, we compared the performance of the proposed method to that of the coulomb counting method. As aforementioned in the introduction, the coulomb counting method [7–9] determines the amount of charge released from or stored in a cell by a simple integration of current. A recently developed coulomb counting method [9] estimates the battery capacity based on two pieces of information: (i) the difference in the SOC values before and after a partial charge/discharge cycle [9], and (ii) the net flow charge during the partial cycle that is directly calculated by coulomb counting. This coulomb counting method is employed in the comparative study. The capacity estimate by the method can be mathematically expressed as [9]

$$h = \frac{\int_{t_1}^{t_2} I(t)dt}{SOC_2 - SOC_1} \quad (12)$$

where SOC_1 (SOC_2) is the SOC level at the very beginning (end) of the partial cycle, I is the current, and t_1 (t_2) is the time stamp at the very beginning (end) of the partial cycle. The way to compute each of the terms in Eq. (12) is summarized as follows:

- SOC_1 was empirically derived based on the initial charge voltage (x_1) and the cubic spline interpolation with a set of known charge voltage and SOC values (see the measurement points and interpolated V-SOC curve in Fig. 5). The charge voltage and SOC values were obtained from cell 1 at an early charge/discharge cycle (i.e., cycle 21).
- SOC_2 was set equal to 100%, since each cell is fully charged during each partial charge cycle.
- The time integral in Eq. (12), or the net flow charge, was computed as a simple summation ($x_2 + x_3$) of the CC and CV charge capacities.

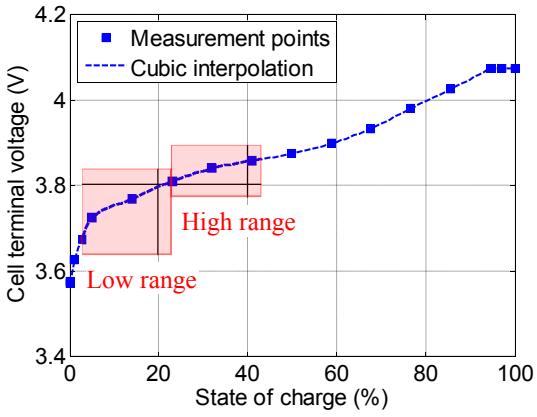

Fig. 5. Plot of cell terminal voltage as a function of SOC during charge. The voltage and SOC measurements were obtained from cell 1 at the 21st charge/discharge cycle. Low initial SOC range (3%–23%) and high initial SOC range (23%–43%) are highlighted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In the RVM regression model, the Gaussian kernel function was employed as the kernel function with the kernel width r set to 0.8, and the maximum number of RVM training iterations was set to 500. The kernel type and the value of the kernel width were empirically determined based on the accuracy of a trained RVM model in capacity estimation. It was observed that the chosen kernel type (Gaussian kernel) and width (0.8) resulted in a trained RVM model with a best accuracy in capacity estimation. The empirical study that determined the value of the kernel width will be detailed in Section 3.4.2.

Table 2 summarizes the capacity estimation errors, namely the RMS error and the Max error, by the proposed and coulomb counting methods. Three important observations can be made from the results and are listed as follows:

- First, the RMS errors by the proposed method are less than 1.00% for Cases 1 and 3, where no positive bias exists in the current measurement, and the Max errors less than 3.00%. These results suggest that the proposed method is capable of producing accurate capacity estimation under both deep (low initial SOC 3%–23%) and shallow (high initial SOC 23%–43%) discharge conditions. The good accuracy can likely be attributed to two facts: (i) the lab cells exhibit a somewhat homogenous capacity fade behavior (see Figs. 3 and 6) throughout the cycling test, which attempts to simulate a use condition similar to patient use in medical applications; and (ii) since the fade behavior is fairly consistent among the lab cells, a training data set, which carries the information about the aging behavior of 7 training cells, is capable of capturing the aging behavior of the testing cell. In non-medical applications (e.g., hybrid and electric vehicles, and consumer electronics) where harsher and more inhomogeneous aging scenarios are often encountered, it is likely that the training data set doesn't fully cover the way a testing cell degrades, and in such cases, the proposed method could produce inaccurate and biased capacity estimates.
- Second, neither the RMS error nor the Max error by the proposed method showed a large increase when a 2% positive bias was artificially added to the current measurement (Case 1 versus Case 2, and Case 3 versus Case 4). This is not surprising, since RVM can adapt the regression model to a change in the input feature vectors, as long as the change applies consistently to both the training and testing data sets (as in this study).
- Third, a comparison of the RMS and Max errors between the two methods for Case 1 suggests that the coulomb counting method is less accurate than the proposed method and, as a positive bias was introduced to the current measurement (Cases 2 and 4) and/or the initial SOC range was elevated to a higher level (Cases 3 and 4), the performance difference became more pronounced. Although easy to implement, the accuracy of the coulomb counting method is known to be negatively affected by two error sources, the current sensor error (I) and the error in the initial SOC estimate (SOC_1). The impacts of these error sources on the capacity estimation accuracy are elaborated as follows: (i) the current sensor error, which accumulates over time, affects the numerator in Eq. (12), thereby increasing the error in capacity estimation; and (ii) the error in SOC_1 affects the denominator in Eq. (12), and a higher initial SOC (Cases 3 and 4) calls

Table 2
Capacity estimations on eight lab cells by the coulomb counting and proposed methods.

Case number	Initial SOC range	Bias in current measurement	RMS (%)		Max (%)	
			Coulomb counting method	Proposed method	Coulomb counting method	Proposed method
Case 1	3%–23%	None	2.29	0.51	9.54	1.67
Case 2	3%–23%	2% (+)	3.69	0.51	11.73	1.96
Case 3	23%–43%	None	5.06	0.87	16.80	2.10
Case 4	23%–43%	2% (+)	6.61	0.90	19.13	2.30

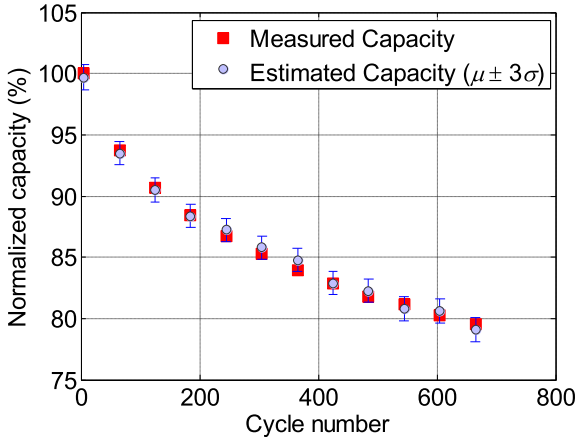

Fig. 6. Capacity estimation results of first lab cell (cell 1) by the proposed method for Case 1. Results are plotted every 60 cycles for the ease of visualization. Error bars indicate ± three standard deviations in capacity estimates.

for a more flat region in the V-SOC curve (see Fig. 5) to be exercised for inferring SOC_1 from x_1, resulting in a larger error in SOC_1 inference and thus a lower accuracy in capacity estimation. It is also noted that the error in SOC_1 inference may increase over repeated cycling, since the V-SOC curve evolves over the course of the cycling test likely due to impedance increase, and as a result, the V-SOC curve at the early cycle (i.e., cycle 21) cannot accurately represent the V-SOC relationship at a later cycle. In contrast, the proposed method is capable of learning the evolution of the V-SOC curve and adapting the regression model to accommodate the evolution, and thus achieved more accurate capacity estimation.

The capacity estimation result of the first lab cell (cell 1) by the proposed method for Case 1 is shown in Fig. 6. The measured capacity for a charge/discharge cycle was calculated using the coulomb counting method, which integrates the discharge current over time for the entire discharge cycle. Note that the measured capacities of the first lab cell are also plotted, against the time on test instead of the cycle number, in Fig. 3. It can be observed from Fig. 6 that the sparse Bayesian learning method closely tracks the capacity fade trend throughout the 10 years' cycling test. For most of the cycles shown in Fig. 6, the ± 3σ error bars in the capacity estimates contain the measured capacities. This suggests that the proposed method, as a statistical learning technique, is capable of archiving robust capacity estimation by accounting for the uncertainty in both the feature data and kernel weight estimates (see Section 2.2.2). Similar observations can be made on the other seven lab cells (cells 2–8).

3.4.2. Effect of kernel width

A parametric study was performed to empirically investigate the effect of the kernel width r on the accuracy of the RVM regression model. For each of the four cases (Cases 1–4), we conducted 9 CV runs, each of which consists of 8 CV trials, with r set to 9 unique values evenly spaced in the range [0.2, 3.2]. A plot of the RMS errors for these values of r was produced to graphically summarize the results for Cases 1 and 3 and is shown in Fig. 7. It can be observed that, for both cases, the mean of the RMS error monotonically decreases with r when $r \leq 0.8$ and increases with k when $r \geq 1.6$. The results suggest that (i) the kernel width r, as a smoothing parameter in the Gaussian kernel function, has a dramatic impact on the prediction accuracy and (ii) an inappropriate selection of the kernel width may result in an unacceptably low accuracy. The three values of r between 0.8 and 1.6 produce relatively small errors for both cases. Among the three values, an r value of 0.8 produces a slightly smaller error than the other two for Case 1 and very comparable errors for Case 3. This value was thus chosen as the smoothing parameter. The results dealt earlier (Fig. 6, and Tables 1 and 2) were derived using this chosen value of r. Finally, we note that the optimal value of the kernel width heavily depends on the specific data set and is likely to vary from one data set to another. A similar simulation study can be performed to determine the optimal kernel width for a specific data set.

3.4.3. Sparsity property of RVM model

As aforementioned in Section 2.2.2, one of the desirable properties that RVM possesses is sparsity, i.e., only a subset of training data points is retained in the trained regression model. Those retained data points are termed *relevance vectors* that form the final sparse solution. For example, among the 750 feature vectors in the training data set for CV trial 8, only 23 (i.e., 3.07%) were chosen as relevance vectors, and these 12 relevance vectors appear to be the most extreme vectors in the feature space. Table 3 summarizes the numbers of feature vectors and relevance vectors in each of the 8 CV trials for Case 1. It can be observed that the ratios of relevance vectors to feature vectors are less than 4.00% for all the CV trials.

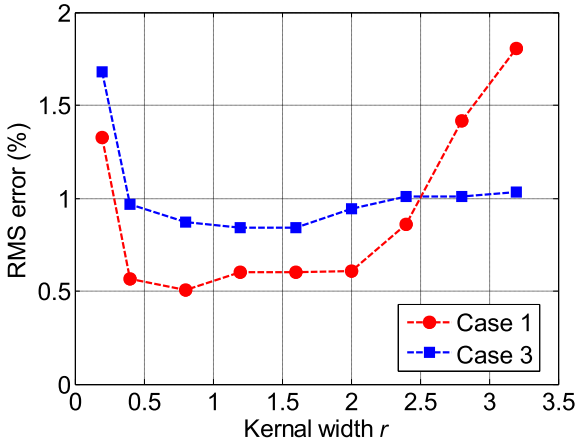

Fig. 7. RMS errors for different kernel width r values for Cases 1 and 3. RMS error curves for Cases 2 and 4 are found to be similar with those for Case 1 and 3, and thus are not shown in the graph.

Table 3
Numbers of feature vectors and relevance vectors in each CV trial for Case 1.

	CV trial 1	CV trial 2	CV trial 3	CV trial 4	CV trial 5	CV trial 6	CV trial 7	CV trial 8
No. of feature vectors	753	756	753	756	753	753	753	750
No. of relevance vectors	26	25	28	29	27	23	27	23
Ratio of relevance vectors[a] (%)	3.45	3.31	3.72	3.84	3.59	3.05	3.59	3.07

[a] Ratio of relevance vectors = No. of relevance vectors/No. of feature vectors × 100.

This suggests that the sparsity property of RVM enables the generation of a reduced-scale regression model that utilizes only a small fraction of the training data set, thereby improving computational efficiency in capacity estimation.

3.4.4. Capacity estimation results on field cells

Twenty-three Li-ion prismatic cells were extracted from the devices after 4–7 implant years. These twenty-three field cells were then cycled using the same test procedure as the lab cells. The proposed method described in Section 2 were used to estimate the discharge capacities of the field cells based on the post-explant cycling data. The RVM regression model was trained using the feature data generated from the eight lab cells for Case 1. The capacity estimation results on these field cells are graphically summarized in Fig. 8. Both the measured and estimated discharge capacities presented in Fig. 8 are normalized by an estimated BOL discharge capacity. It can be observed that the capacity estimates closely match the measured values over a wide range of normalized capacities (i.e., from 76.8% to 91.8%) and that, for most of the cells, the ± 3σ error bars contain the measured capacities. The overall RMS and Max errors are respectively 0.92% and 2.07%, which appear to be comparable to the estimation errors on the lab cells (see Table 1). The satisfactory estimation accuracy on real field cells further verify that the proposed method is effective in estimating the capacity of Li-ion battery.

4. Conclusion

This paper presents a sparse Bayesian learning method for capacity estimation of Li-ion battery based on the charge voltage and current curves. The sparse Bayesian learning method is used to train a RVM regression model which approximates the complex relationship between the capacity of a battery and the characteristic features extracted from the charge curves of the battery. The sparsity property of RVM makes the proposed method very suitable for onboard estimation devices that require computationally efficient estimation methods. The performance of the method is verified using 10 years' continuous cycling data and post-explant cycling data acquired from Li-ion batteries that are designed for use in implantable medical devices. The verification reveals that the method achieves satisfactory accuracy and measures the uncertainty in capacity estimation, suggesting that the method is a promising tool for online health management of Li-ion batteries used in these medical devices.

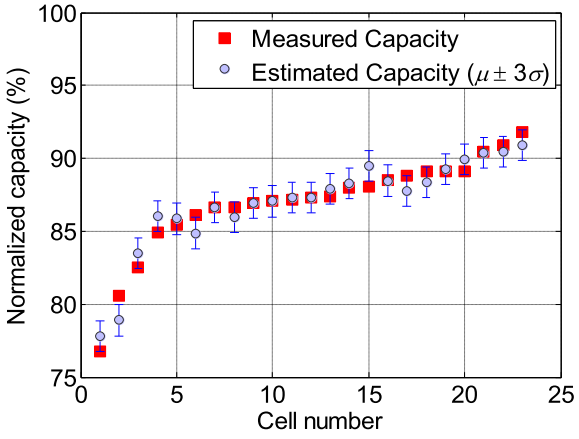

Fig. 8. Capacity estimation results on twenty-three field cells for Case 1. Cells are sorted by the normalized capacity in an ascending order. Error bars indicate ± three standard deviations in capacity estimates.

References

[1] 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.
[2] G.L. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2: simultaneous state and parameter estimation, J. Power Sources 161 (2) (2006) 1369–1384.
[3] W. He, N. Williard, C. Chen, M. Pecht, State of charge estimation for electric vehicle batteries using unscented Kalman filtering, Microelectron. Reliab. 53 (6) (2013) 840–847.
[4] S. Lee, J. Kim, J. Lee, B.H. Cho, State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge, J. Power Sources 185 (2) (2008) 1367–1373.
[5] C. Hu, B.D. Youn, J. Chung, A multiscale framework with extended Kalman filter for lithium-ion battery SOC and capacity estimation, Appl. Energy 92 (2012) 694–704.
[6] R. Xiong, F. Sun, Z. Chen, H. He, A data-driven multi-scale extended Kalman filtering based parameter and state estimation approach of lithium-ion polymer battery in electric vehicles, Appl. Energy 113 (2014) 463–476.
[7] K.S. Ng, C.S. Moo, Y.P. Chen, Y.C. Hsieh, Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries, Appl. Energy 86 (9) (2009) 1506–1511.
[8] W. Waag, D.U. Sauer, Adaptive estimation of the electromotive force of the lithium-ion battery after current interruption for an accurate state-of-charge and capacity determination, Appl. Energy 111 (2013) 416–427.
[9] C. Hu, G. Jain, P. Tamirisa, T. Gorka, Method for estimating capacity and predicting remaining useful life of lithium-ion battery, Appl. Energy 126 (2014) 182–189.
[10] A. Eddahech, O. Briat, N. Bertrand, J. Deletage, J. Vinassa, Behavior and state of health monitoring of Li-ion batteries using impedance spectroscopy and recurring neural networks, Int. J. Electr. Power & Energy Syst. 42 (1) (2012) 487–494.
[11] J. Kim, S. Lee, B.H. Cho, Complementary cooperation algorithm based on DEKF combined with pattern recognition for SOC/capacity estimation and SOH prediction, IEEE Trans. Power Electron. 27 (1) (2012) 436–451.
[12] G. Bai, P. Wang, C. Hu, M. Pecht, A generic model-free approach for lithium-ion battery health management, Appl. Energy 135 (2014) 247–260.
[13] B. Pattipati, C. Sankavaram, K. Pattipati, System identification and estimation framework for pivotal automotive battery management system characteristics, IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev. 41 (6) (2011) 869–884.
[14] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, K. Dietmayer, Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods, J. Power Sources 239 (2013) 680–688.
[15] A. Widodo, M.C. Shim, W. Caesarendra, B.S. Yang, Intelligent prognostics for battery health monitoring based on sample entropy, Expert Syst. Appl. 38 (9) (2011) 11763–11769.
[16] B. Saha, K. Goebel, S. Poll, J. Christophersen, Prognostics methods for battery health monitoring using a Bayesian framework, IEEE Trans. Instrum. Meas. 58 (2) (2009) 291–296.
[17] C. Hu, G. Jain, P. Zhang, C. Schmidt, P. Gomadam, T. Gorka, Data-driven approach based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery, Appl. Energy 129 (2014) 49–55.
[18] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.
[19] E. Scott, J. Brown, C. Schmidt, W. Howard, A practical longevity model for lithium-ion batteries: de-coupling the time and cycle-dependence of capacity fade, in: 208th ECS Meeting, Abstract MA2005–02 239, 2006.