Applied Energy 155 (2015) 778–790

Contents lists available at ScienceDirect

Applied Energy

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

Dynamic data-driven identification of battery state-of-charge via symbolic analysis of input–output pairs ☆

Yue Li, Pritthi Chattopadhyay, Asok Ray *

Mechanical & Nuclear Engineering Department, Pennsylvania State University, University Park, PA 16802, USA

HIGHLIGHTS

- Symbolic Time Series Analysis (STSA) has been used for low-complexity feature extraction.
- Discrete wavelet transformation (DWT) has been used for data segmentation.
- Algorithms have been validated on experimental data of pairs of current and voltage data from a lead-acid battery.

ARTICLE INFO

Article history:
Received 26 April 2015
Received in revised form 15 June 2015
Accepted 17 June 2015

Keywords:
State-of-charge
Symbolic dynamics
Pattern classification

ABSTRACT

This paper presents a dynamic data-driven method of pattern classification for identification of the state-of-charge (SOC) parameter in battery systems for diverse applications (e.g., plug-in electric vehicles and hybrid locomotives). The underlying theory is built upon the concept of symbolic dynamics, which represents the behavior of battery system dynamics at different levels of SOC as probabilistic finite state automata (PFSA). In the proposed method, (finite-length) blocks of battery data are selected via wavelet-based segmentation from the time series of synchronized input–output (i.e., current–voltage) pairs in the respective two-dimensional space. Then, symbol strings are generated from the segmented time series pairs in the sense of maximum entropy partitioning and a special class of PFSA, called the D-Markov machine, is constructed to extract the features of the battery system dynamics for pattern classification. To deal with the uncertainties due to the (finite-length) approximation of symbol sequences, combinations of (a priori) Dirichlet and (a posteriori) multinomial distributions are respectively adopted in the training and testing phases of pattern classification. The proposed concept of pattern classification has been validated on (approximately periodic) experimental data that have been acquired from a commercial-scale lead-acid battery.

1. Introduction

The state-of-charge (SOC) is a crucial parameter for operation of battery systems, which depicts the battery system's current capacity (i.e., the maximum charge that can be drawn from its fully charged condition). Accurate estimates of SOC mitigate the risk of battery cells being over-charged and over-discharged. Many applications (e.g., plug-in electric vehicles and hybrid locomotives) require large battery packs that may contain several hundreds of battery cells to meet the large and dynamic power demands. From these perspectives, real-time identification of battery SOC would enhance the efficiency of power and energy allocation within the battery packs.

The current state-of-the-art methods of battery parameter identification can be divided into three broad categories: (i) empirical modeling; (ii) physics-based modeling; and (iii) dynamic data-driven analysis, all of which need data bases of experimental measurements for validation. Empirical modeling methods (e.g., impedance measurements and open circuit voltage testing [1]) often employ dedicated hardware and/or software and require tested battery cells, and are essentially off-line. Although these empirical methods provide good estimates in specific cases, they are time-consuming and cost-prohibitive for general applications. Physics-based modeling methods have been extensively used for identification of battery parameters; examples are reduced-order system identification, linear switch models [2], and Kalman filtering [3]. However, physics-based modeling methods require thorough knowledge of the electrochemical

☆ This work has been supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-12-1-0270. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

* Corresponding author.
E-mail addresses: yol5214@psu.edu (Y. Li), pxc271@psu.edu (P. Chattopadhyay), axr2@psu.edu (A. Ray).

http://dx.doi.org/10.1016/j.apenergy.2015.06.040
0306-2619/

characteristics of the battery cells to arrive at appropriate model structures for parameter identification at different operating points of the nonlinear battery dynamics. In contrast, dynamic data-driven methods (that do not explicitly rely on physical phenomena of battery electrochemistry) could be more efficient in terms of computation time and memory costs if adequate training data are available. Several data-driven methods have been reported in literature for battery SOC identification based on different concepts of machine learning, which include artificial neural networks (ANNs) [4], fuzzy logic [5], support vector regression [6], and symbolic dynamics [7]. Changes in the battery SOC (e.g., frequent discharge by acceleration and regeneration by deceleration in the operations of electric vehicles and locomotives) may take place over small time-windows, the estimated values should be updated frequently. Therefore, a crucial evaluation factor for dynamic data-driven applications is to generate robust and accurate identification of SOC in real time with limited lengths of test data. Very few data-driven methods, available in current literature, have addressed this particular issue. The reason being that most of the data-driven methods require adequate length of training and test data to make extracted features accurate and robust to corresponding system operating condition, which usually lead to a delayed SOC identification.

Li et al. [7] have reported a dynamic data-driven method, as a feasible alternative to model-based methods, for identification of SOC by using the time series of the battery voltage time series. The underlying concept is built upon the theory of Symbolic Time Series Analysis (STSA) [8–10] that extracts the dynamic information as low-dimensional features by symbolization of sensor time series and subsequently generation of probabilistic finite state automata (PFSA). The performance of SOC identification in this context has been studied under different training and test data lengths. In 2015, Li et al. [11] have extended their earlier work in [7] by making use of an ensemble of pairs of synchronized battery input/output (i.e., current/voltage) time series for identification of the state-of-health (SOH) parameter. In 2013, Wen et al. [12] presented a general framework for statistically quantifying the uncertainties of the extracted PFSA features due to finite-length data in both training and testing phases of pattern classification.

As an extension of the authors' earlier work [7,11], this paper develops a method for identification of battery SOC as a pattern classification problem by making use of ensemble of time series pairs of synchronized battery inputs (i.e., charging/discharging current) and battery outputs (i.e., voltage) for information compression and feature extraction. The uncertainties due to finite lengths of both training and testing data are modeled as Dirichlet and multinomial distributions respectively, based on the earlier work of Wen et al. [12]. The proposed method has been validated with experimental data of a (commercial-scale) lead-acid battery under varying input–output (e.g., current–voltage) conditions.

The application part of this paper focuses on lead-acid batteries that are widely used in automobiles and electric locomotives [13]. Nevertheless the proposed method of SOC identification can be easily extended to other battery types (e.g., lithium-ion) and identification of additional important operating parameters (e.g., state-of-health) [7]. Significant contributions of the present paper over the work reported in open literature and especially the authors' earlier work [7,11] on development of dynamic data-driven application systems (DDDAS) [14] for battery parameter identification are delineated below.

- *Characterization of the battery dynamics based on the synchronized input–output time series via symbolic dynamic analysis*: This representation of (possibly nonlinear) input–output characteristics is analogous to a transfer function realization and alleviates the need for linearization of the system dynamics.
- *Wavelet-based segmentation of time series for separation of the active parts (e.g., charge/discharge pulses) from the inactive parts (i.e., constant charge/discharge)*: While battery systems may be operated under constant load (i.e., when there are no significant fluctuations) over a time span, the active dynamic responses of the battery can be localized based on the information derived in the time–frequency domain [15].
- *Information compression in the form of low-dimensional features extracted from symbolized time series of two-dimensional data*: Feature extraction is achieved by partitioning of (synchronized) input–output pairs in the sense of maximum entropy and subsequent analysis of the symbolized data in the setting of probabilistic finite state automata (*PFSA*) via D-Markov modeling [8,10].
- *Incorporation of the effects of finite-length symbol strings on pattern classification*: The uncertainties due to finite-length symbol strings are quantified by as a combination of *a priori* (training phase) and a posteriori (testing phase) are quantified by Dirichlet and multinomial distribution models, respectively [12]. Experimental results have provided an insight for selection of the lengths of training and testing data (e.g., number of consecutive cycles of charging followed by discharging) to achieve the desired performance.

This paper is organized into six sections, including the present section, and one appendix. Section 2 briefly describes the underlying definitions and pertinent concepts of battery dynamics, wavelet-based segmentation, symbolic dynamics, along with a framework for online identification of battery state-of-charge (SOC). Section 2 provides a very brief conceptual background of battery dynamics and summarizes the fundamental concepts and synopses of previous and related work that lead to the proposed method for battery SOC estimation. Section 3 summarizes the general framework of online pattern classification by symbolic analysis of synchronized time-series pairs in the input–output space. Section 4 presents the procedure of experimental data collection as well as the data pre-processing procedure. Section 5 presents the results of the application to a commercial-scale lead-acid battery. Section 6 summarizes and concludes this paper along with recommendations for future research. Appendix A lists the algorithms that are developed in the main body of the paper to generate the results.

2. Background and pertinent concepts

This section presents the underlying definitions and pertinent concepts of battery dynamics, wavelet-based segmentation, partitioning of segmented data for symbolization, and symbolic dynamics-based information compression, along with a framework for online pattern classification.

2.1. Battery system parameters

This subsection introduces standard definitions of pertinent battery parameters, at a given ambient temperature [7,16].

Definition 2.1 (*Battery capacity*). The capacity $C(t)$ of a battery at time t is its maximum charge (in units of ampere-hours) that can be drawn from its fully charged condition at a rate $C(t)/30$ (in units of amperes).

Definition 2.2 (*SOH*). Let a new battery be put into service at time t_0. The state-of-health $SOH(t)$ of the (possibly used) battery at the current time t, where $t \geqslant t_0$, is defined to be the ratio of the battery capacities at time epochs t and t_0, i.e.,

$$SOH(t) = \frac{C(t)}{C(t_0)} \quad \text{for all time } t \geqslant t_0 \tag{1}$$

Definition 2.3 (*DOD and SOC*). Let a battery be fully charged at time t and let $I(\tau)$ be the applied current (in units of amperes) at time τ. Then, depth of discharge (*DOD*) and state-of-charge (*SOC*) at time $t + \Delta t$ are respectively defined as

$$DOD(t+\Delta t) = \frac{1}{C(t)} \int_t^{t+\Delta t} I(\tau)\, d\tau \text{ for } \Delta t \geqslant 0 \tag{2}$$

$$SOC(t+\Delta t) = 1 - DOD(t+\Delta t) \text{ for } \Delta t \geqslant 0 \tag{3}$$

Remark 2.1. It is noted that $SOH \in [0,1]$ and $SOC \in [0,1]$ for all time $t \geqslant t_0$, where t_0 is the time of putting a new battery into service.

2.2. Wavelet-based time series segmentation

Recently Li et al. [11] proposed a wavelet-based segmentation method to extract dynamic data segments. The objective here is to capture the dynamic characteristics of the battery system from the time series of mixed operating modes consisting of idle operation and frequent discharge & regeneration in the battery operation. In the time–frequency analysis of wavelet transform [15], the coefficients of chosen scales corresponding to frequency points in interest reflects the dynamic level of the system at each point in the time domain. Details of wavelet-based segmentation are reported in [11] and the underlying algorithm is listed as Algorithm 1 in Appendix A.

The pertinent steps of the wavelet-based segmentation procedure, depicted in Fig. 1, are as follows.

- Step 1: *Collect (finite) time series data* $x[n],\ n = 1,2,\ldots,N$ *for a given sampling interval* Δ. It is noted that the length N of the time series is user-selectable.
- Step 2: *Computation of the discrete Fourier transform* $\hat{x}[k],\ k = 1,2,\ldots,N$ *and the corresponding PSD* $S_x[k],\ k = 1,2,\ldots,N$.
- Step 3: *Identify the frequency points of interest,* $\varphi_m,\ m = 1,2,\ldots,M$. In this step, the set $\{S_x[\varphi_m]\}$ is formed in terms of the M points with highest values of the PSD $S_x[k],\ k = 1,2,\ldots,N$. [Note: M is usually significantly less than N].
- Step 4: *Compute the corresponding wavelet scales* $\alpha_m,\ m = 1,2,\ldots,M$. This step follows $\alpha_m = \frac{f_C}{\varphi_m \cdot \Delta}$ with the corresponding frequency points φ_m and the central frequency f_c of the chosen wavelet basis function.
- Step 5: *Compute the wavelet coefficients for each scale and get the total summation*. In this step, the values of the wavelet coefficients is obtained $\tilde{x}_{\alpha_m}[\tau_\ell^m] = \tilde{x}[\alpha_m, \tau_\ell^m],\ \ell = 1,2,\ldots,N$ at each scale α_m. Then the total summation of wavelet coefficients at each time shift indices $\tilde{x}[\tau_\ell]$ is computed.
- Step 6: *Identification of the set, T, of all segmented time indices*. This step is completed by selecting the time shift indices τ_ℓ as the corresponding summated wavelet coefficients $\tilde{x}[\tau_\ell]$ exceeding the pre-defined threshold ξ.

2.3. Maximum entropy partitioning and symbolization

This subsection addresses the partitioning of the input–output space for symbolization of the time series into a mutually exclusive and exhaustive set of finitely many segments. In this setting, a symbol string is generated from the (finite-length) time series by assigning a unique symbol to each segment of the input–output space.

The maximum-entropy partitioning (MEP) [9] of the time-series data has been adopted in this paper to construct the symbol alphabet Σ and to generate symbol strings. In this partitioning, the information-rich regions of the data set are partitioned finer and those with sparse information are partitioned coarser to maximize the Shannon entropy of the generated symbol string from the reference data set. The MEP procedure is presented as Algorithm 2 in Appendix A.

Fig. 2 depicts the underlying concept of symbolization of a 2-dimensional time series, where each segment in the top plot is labeled by a unique symbol and Σ denotes the alphabet of these symbols. The segment, visited by the time series plot takes a symbol value from the alphabet Σ. For example, having $\Sigma = \{\alpha, \beta, \gamma, \delta\}$ in Fig. 2, a time-series $x_0x_1x_2\ldots$ generates a string of symbols in the symbol space as: $s_0s_1s_2\ldots$, where each $s_i,\ i = 0,1,2,\ldots$, takes a symbol value from the alphabet Σ. This mapping is called symbolic dynamics as it attributes a (physically admissible) symbol string to the dynamical system starting from an initial state (for example, see the symbol string at the middle row in Fig. 2).

Li et al. [11] used four alternative types of partitioning in the input–output space of battery response:

- Partitioning Type 1 (Cartesian coordinates): First partition in the input axis (e.g., abscissa), and then partition in the output axis (e.g., ordinate) at individual input segments.
- Partitioning Type 2 (Cartesian coordinates): First partition in the output axis, and then partition in the input axis at individual output segments.
- Partitioning Type 3 (Polar coordinates): First partition in the magnitude, and then partition in the phase at individual magnitude segments.
- Partitioning Type 4 (Polar coordinates): First partition in the phase, and then partition in the magnitude at individual phase segments.

Each time series of the input–output (i.e., current–voltage) pair is partitioned in the associated two-dimensional space to construct a symbolic string.

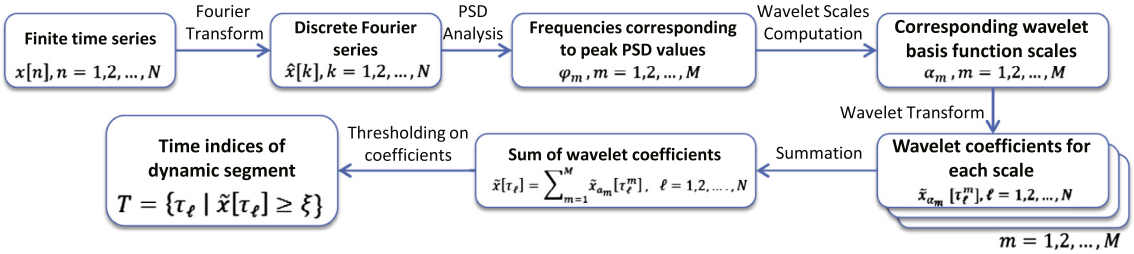


Fig. 1. Procedure for wavelet-based segmentation.

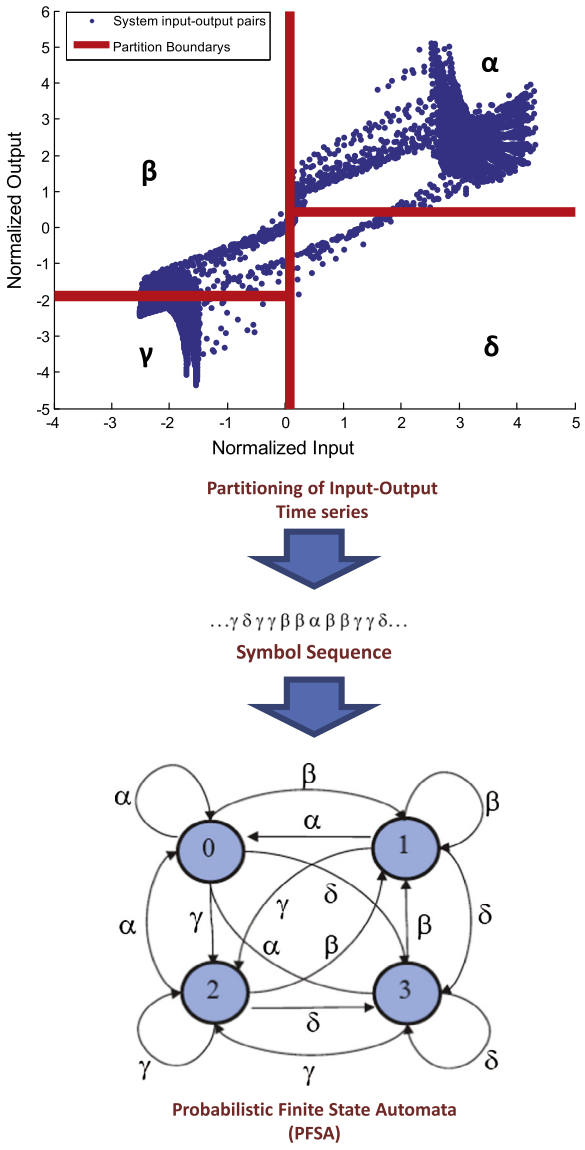


Fig. 2. Construction of finite state automata (FSA) from time series.

2.4. Symbolic Time Series Analysis (STSA)

This subsection briefly describes the underlying concept of Symbolic Time Series Analysis (STSA) upon which the proposed dynamic-data-driven tool of battery parameter identification is constructed; STSA encodes the behavior of (possibly nonlinear) dynamical systems from the observed time series by symbolization and construction of state machines (i.e., probabilistic finite state automata (PFSA)) [8]. This is followed by computation of the state emission matrices that are representatives of the evolving statistical characteristics of the dynamical system.

The core assumption in the STSA analysis for construction of probabilistic finite state automata (PFSA) from symbol strings is that the symbolic process under both nominal and off-nominal conditions can be approximated as a Markov chain of order D, called the D-Markov machine, where D is a positive integer. While the details of the D-Markov machine construction are reported in [8,10], the pertinent definitions and their implications are succinctly presented below.

Definition 2.4 (*DFSA*). A deterministic finite state automaton (DFSA) is a 3-tuple $G = (\Sigma, Q, \delta)$ where:

(1) Σ is a non-empty finite set, called the symbol alphabet, with cardinality $1 < |\Sigma| < \infty$;
(2) Q is a non-empty finite set, called the set of states, with cardinality $1 < |Q| < \infty$;
(3) $\delta : Q \times \Sigma \to Q$ is the state transition map;

and $\Sigma^\star$ is the collection of all finite-length strings with symbols from Σ including the (zero-length) empty string ϵ, i.e., $|\epsilon| = 0$.

Remark 2.2. It is noted that Definition 2.4 does not make use of an initial state, because the purpose here is to work in a statistically stationary setting, where no initial state is required as explained by Adenis et al. [17].

Definition 2.5 (*PFSA*). A probabilistic finite state automaton (PFSA) is constructed upon a DFSA $G = (\Sigma, Q, \delta)$ as a pair $K = (G, \pi)$, i.e., the PFSA K is a 4-tuple $K = (\Sigma, Q, \delta, \pi)$, where:

(1) Σ, Q, and δ are the same as in Definition 2.4;
(2) $\pi : Q \times \Sigma \to [0, 1]$ is the probability emissivity function that satisfies the condition $\sum_{\sigma \in \Sigma} \pi(\sigma|q) = 1 \ \ \forall q \in Q$. Denoting π_{ij} as the probability of occurrence of a symbol $\sigma_j \in \Sigma$ at the state $q_i \in Q$, the $(|Q| \times \Sigma|)$ probability emissivity matrix (also known as emission matrix) is obtained as $\Pi = [\pi_{ij}]$.

Definition 2.6 (*D-Markov*). A D-Markov machine [8] is a PFSA in which each state is represented by a (nonempty) finite string of D symbols where

- D, a positive integer, is the depth of the Markov machine;
- Q is the finite set of states with cardinality $|Q| \leqslant |\Sigma|^D$. The states are represented by equivalence classes of symbol strings of maximum length D, and each symbol in the sting belongs to the alphabet Σ;
- $\delta : Q \times \Sigma \to Q$ is the state transition map that satisfies the following condition if $|Q| = |\Sigma|^D$: There exist $\alpha, \beta \in \Sigma$ and $s \in \Sigma^\star$ such that $\delta(\alpha s, \beta) = s\beta$ and $\alpha s, s\beta \in Q$.

Remark 2.3. It follows from Definition 2.6 that a D-Markov chain is treated as a statistically stationary stochastic process $S = \cdots s_{-1} s_0 s_1 \cdots$, where the probability of occurrence of a new symbol depends only on the last D symbols, i.e., $P[s_n | \cdots s_{n-D} \cdots s_{n-1}] = P[s_n | s_{n-D} \cdots s_{n-1}]$.

The construction of a D-Markov machine is based on: (i) state splitting that generates symbol blocks of different lengths according to their relative importance; and (ii) state merging that assimilates histories from symbol blocks leading to the same symbolic behavior. Words of length D on a symbol string are treated as the states of the D-Markov machine before any state-merging is executed. Thus, on an alphabet Σ, the total number of possible states becomes less than or equal to $|\Sigma|^D$; and operations of state merging may significantly reduce the number of states [10].

The PFSA states represent different combinations of blocks of symbols on the symbol string. In the graph of a PFSA, the directional edge (i.e., the emitted event) that interconnects a state (i.e. a node) to another state represents the transition probability between these states. Therefore, the "states" denote all possible symbol blocks (i.e., words) within a window of certain length, and the set of all states is denoted as $Q = \left\{q_1, q_2, \ldots, q_{|Q|}\right\}$ and $|\ Q\ |$ is the number of (finitely many) states. The procedure for estimation of the emission probabilities is presented next.

Given a (finite-length) symbol string S over a (finite) alphabet Σ, there exist several PFSA construction algorithms to discover the underlying irreducible PFSA model K of S, such as causal-state splitting reconstruction (CSSR) [18] and D-Markov [8,10]. These algorithms start with identifying the structure of the PFSA $K \triangleq (Q, \Sigma, \delta, \pi)$. To estimate the state emission matrix, a $|Q| \times |\Sigma|$ count matrix C is constructed and each element c_{kj} of C is computed as:

$$c_{kj} \triangleq 1 + N_{kj} \tag{4}$$

where N_{kj} denotes the number of times that a symbol σ_j is generated from the state q_k upon observing the symbol string S. The maximum a posteriori probability (MAP) estimates of emission probabilities for the PFSA K are computed by frequency counting as

$$\pi(\sigma_j|q_k) \triangleq \frac{c_{kj}}{\sum_\ell c_{k\ell}} = \frac{1 + N_{kj}}{|\Sigma| + \sum_\ell N_{kj}} \tag{5}$$

The rationale for initializing each element of the count matrix C to 1 is that if no event is generated at a state $q \in Q$, then there should be no preference to any particular symbol and it is logical to have $\pi(\sigma|q) = \frac{1}{|\Sigma|}\ \forall \sigma \in \Sigma$, i.e., the uniform distribution of event generation at the state q. The above procedure guarantees that the PFSA, constructed from a (finite-length) symbol string, must have an (elementwise) strictly positive emissivity map Π and that the state transition map δ in Definitions 2.4 and 2.6 is a total function.

Having computed the emission probabilities $\pi(\sigma_j|q_k)$ for $j \in \{1, 2, \ldots, |\Sigma|\}$ and $k \in \{1, 2, \ldots, |Q|\}$, the estimated emission probability matrix of the PFSA is obtained as:

$$\Pi \triangleq \begin{bmatrix} \pi(\sigma_1|q_1) & \cdots & \pi(\sigma_{|\Sigma|}|q_1) \\ \vdots & \ddots & \vdots \\ \pi\left(\sigma_1|q_{|Q|}\right) & \cdots & \pi\left(\sigma_{|\Sigma|}|q_{|Q|}\right) \end{bmatrix}. \tag{6}$$

Bahrampour et al. [19] presented a comparative evaluation of Cepstrum, Principal Component Analysis (PCA) and Symbolic Time Series Analysis (STSA) as feature extractors for target detection and classification. The underlying algorithms of feature extraction were executed in conjunction with three different pattern classification algorithms, namely, support vector machines (SVM), k-nearest neighbor (k-NN), and sparse representation classifier (SRC). The results of comparison show consistently superior performance of STSA-based feature extraction over both Cepstrum-based and PCA-based feature extraction in terms of successful detection, false alarm, and wrong detection and classification decisions. The procedure of STSA for feature extraction, which makes use of the estimated emission probability matrix Π, is presented as Algorithm 3 in Appendix A.

2.5. Online pattern classification

This subsection presents a general framework for online pattern classification problems in the symbolic domain; the patterns are constructed from (finite-length) symbol strings as probabilistic finite state automata (PFSA) with (possibly) diverse algebraic parameters (e.g., alphabet size $|\Sigma|$ and state cardinality $|Q|$). The uncertainties due to the finite length of the symbol string in both training and testing phases, which could influence the final classification decision, are formulated as (a priori) Dirichlet and (a posteriori) multinomial distributions [20]. While the details are reported by Wen et al. [12] and there is an application fault detection in gas turbine by Sarkar et al. [21], the essential concepts and procedures of the proposed method are succinctly summarized below for completeness of the paper.

Let there be L classes of symbolic systems of interest, denoted by $C_1, C_2, \ldots, C_L$, over the same symbol alphabet Σ. During the training phase, a symbol string $S^i \triangleq s^i_1 s^i_2 \ldots s^i_{N_i}$ of (finite) length N_i is obtained from each class C_i. Then, a PFSA $K^i = (Q^i, \Sigma, \delta^i, \pi^i)$ is obtained for each class, whose structures (i.e., Q^i and δ^i) may not necessarily be the same [12]. Thus the only indeterminate π^i (i.e., the numerically generated estimate of the emission matrix) is selected as the feature vector for classification. The distribution of π^i for each class i is fitted as a Dirichlet distributions in the training phase. In the testing phase, the probability of an observed symbol string $\widetilde{S}$ belonging to the i^{th} class of PFSA is modeled as a multinomial distribution that is fitted based on the (numerically generated) emission probability matrix Π^i and the classification decision is made by choosing the class that maximizes that the posterior probability.

In the training phase, each row $\Pi^i_m (m = 1, 2, \ldots, |Q|)$ of the emission probability matrix Π^i is treated as a Dirichlet distribution conditioned on a symbol string S^i:

$$f_{\Pi^i_m|S^i}(\boldsymbol{\theta}^i_m) \sim Dirichlet(\boldsymbol{N}^i_m + \mathbf{1}) \tag{7}$$

where $\boldsymbol{\theta}^i_m$ is a realization of the random vector $\boldsymbol{\Pi}^i_m$ as:

$$\boldsymbol{\Pi}^i_m = \begin{bmatrix} \Pi^i_{m1} & \Pi^i_{m2} & \ldots & \Pi^i_{m|\Sigma|} \end{bmatrix}$$

and the vector $\boldsymbol{N}^i_m$ in Eq. (4) is generated as:

$$\boldsymbol{N}^i_m = \begin{bmatrix} N^i_{m1} & N^i_{m2} & \ldots & N^i_{m|\Sigma|} \end{bmatrix}$$

By the Markov property of the PFSA K^i, the $(1 \times |\Sigma|)$ row-vectors, $\{\Pi^i_m\}, m = 1, \ldots |Q|$, are statistically independent of each other. Therefore, the *a priori* density $f_{\Pi^i|S^i}$ of the emission probability matrix Π^i, conditioned on the symbol string S^i, is given as

$$f_{\Pi^i|S^i}(\boldsymbol{\theta}^i) = \prod_{m=1}^{|Q^i|} f_{\Pi^i_m|S^i}\left(\boldsymbol{\theta}^i_m\right) = \prod_{m=1}^{|Q^i|} \left(\left(N^i_m + |\Sigma| - 1\right)! \prod_{n=1}^{|\Sigma|} \frac{(\boldsymbol{\theta}^i_m)^{N^i_{mn}}}{(N^i_{mn})!} \right) \tag{8}$$

where $\boldsymbol{\theta}^i \triangleq \begin{bmatrix} (\boldsymbol{\theta}^i_1)^T & (\boldsymbol{\theta}^i_2)^T & \ldots & (\boldsymbol{\theta}^i_{|Q|})^T \end{bmatrix}^T \in [0,1]^{|Q| \times |\Sigma|}$. The details of derivation of Eq. (8) can be found in [12].

In the testing phase, let $\widetilde{N}^i_{mn}$ be the number of times the symbol σ_n is emanated from the state $q^i_m \in Q^i$ in the test symbol string $\widetilde{S}$, which is similar to N^i_{mn}, defined earlier for S^i. The probability mass function $\Pr(\widetilde{N}^i_m|\Pi^i_m)$ of the random row-vector $\widetilde{\boldsymbol{N}}^i_m$ is modeled as a multinomial distribution conditioned on a given state and an emissivity function $\boldsymbol{\Pi}^i_m$:

$$\widetilde{\boldsymbol{N}}^i_m = \begin{bmatrix} \widetilde{N}^i_{m1} & \widetilde{N}^i_{m2} & \ldots & \widetilde{N}^i_{m|\Sigma|} \end{bmatrix} \sim Multi(\boldsymbol{\Pi}^i_m) \tag{9}$$

and the conditional probability is

$$Pr(\widetilde{\boldsymbol{N}}^i_m|\Pi^i_m) = (\widetilde{N}^i_m)! \prod_{n=1}^{|\Sigma|} \frac{(\Pi^i_{mn})^{\widetilde{N}^i_{mn}}}{(\widetilde{N}^i_{mn})!} \tag{10}$$

where $\widetilde{N}^i_m \triangleq \sum_{n=1}^{|\Sigma|} \widetilde{N}^i_{mn}$ and $\Gamma(\bullet)$ is the standard gamma function with $\Gamma(n) = (n-1)!\ \forall n \in \mathbb{N}_1$. The details of derivation of Eq. (10) can be found in [12].

Similar to the argument provided in the training phase, all row vectors in the emission matrix for the testing phase are also statistically independent of each other. Therefore, the probability of observing $\widetilde{S}$ conditioned on the emission probability matrix Π^i is given as:

$$\Pr\left(\widetilde{S}|\Pi^i\right) \triangleq \prod_{m=1}^{|Q^i|} \Pr(\widetilde{\boldsymbol{N}}^i_m|\Pi^i_m) \tag{11}$$

$$= \prod_{m=1}^{|Q^i|} \left((\widetilde{N}^i_m)! \prod_{n=1}^{|\Sigma|} \frac{\left(\Pi^i_{mn}\right)^{\widetilde{N}^i_{mn}}}{(\widetilde{N}^i_{mn})!} \right) \tag{12}$$

The results, derived in the training phase and the testing phase, are now combined. Given a symbol string S^i in the training phase, the probability of observing a symbol string $\widetilde{S}$ in the testing phase is

$$\Pr(\widetilde{S}|S^i) = \prod_{m=1}^{|Q^i|} \frac{(\widetilde{N}^i_m)!\left(N^i_m + |\Sigma| - 1\right)!}{\left(\widetilde{N}^i_m + N^i_m + |\Sigma| - 1\right)!} \times \prod_{n=1}^{|\Sigma|} \frac{(\widetilde{N}^i_{mn} + N^i_{mn})!}{(\widetilde{N}^i_{mn})!(N^i_{mn})!} \tag{13}$$

where $\Omega_{|\Sigma|} \triangleq [0,1]^{|\Sigma|}$ is the sample space of dimension $|\Sigma|$. The details of derivation of Eq. (13) can be found in [12].

The posterior probability of the observed symbol string $\widetilde{S}$ belonging to the class C_i is denoted as $\Pr(C_i|\widetilde{S})$ and is given as

$$\Pr(C_i|\widetilde{S}) = \frac{\Pr(\widetilde{S}|S^i)\Pr(C_i)}{\sum_{j=1}^{L}\Pr(\widetilde{S}|S^j)\Pr(C_j)}, \quad i = 1,2,\ldots,L \tag{14}$$

where $\Pr(C_i)$ is the known prior distribution of the class C_i. Then, the classification decision is made as follows.

$$D_{class} = \arg\max_i \Pr(C_i|\widetilde{S}) = \arg\max_i \left(\Pr(\widetilde{S}|S^i)\Pr(C_i)\right) \tag{15}$$

If no prior information on $\Pr(C_i)$ is available, it is logical to assume a uniform distribution over the classes. In that case, the rule of classification decision becomes

$$D_{class} = \arg\max_i \Pr(\widetilde{S}|S^i) \tag{16}$$

Remark 2.4. If the information on N^i_{mn}'s and $\widetilde{N}^i_{mn}$'s are available, no other information is needed to obtain the statistics of the symbol strings S^i's and $\widetilde{S}$. Therefore, N^i_{mn}'s and $\widetilde{N}^i_{mn}$'s are sufficient statistics of S^i's and $\widetilde{S}$, respectively.

3. Framework of pattern classification

This section summarizes the general framework of online pattern classification by symbolic analysis of synchronized time-series pairs in the input–output space.

The flow chart in Fig. 3 summarizes the basic procedures of the proposed information compression method for two-dimensional time series of synchronized input–output data. First, the raw data are normalized such that they have the properties of zero-mean and unit-variance. Second, the segments that contain dynamic information of the system are extracted from the normalized data and are then concatenated after wavelet-based segmentation in the time–frequency domain. Third, the concept of Maximum Entropy Partitioning (MEP) [9] is adopted to partition the concatenated dynamic data set into multiple states in the 2-dimensional input–output data space. Finally, symbol strings are generated from the partitioned data sets by labeling each partition segment with assigning a unique symbol. The collection of these symbols is the alphabet of the PFSA.

The proposed sequential online identification method consists of an off-line training phase and an online testing phase, as depicted in Fig. 4. All training and testing data sets are symbolized distinctively and, in the same way, the classification process is conducted based on the generated symbol strings. In the training phase, probabilistic finite state automata (PFSA) K^i, $i = 1,2,\ldots,L$, are constructed from the symbol strings S^i for every class. Then, the probability density function $f_{\Pi^i|S^i}$ of the emission probability matrix Π^i, conditioned on the symbol string S^i, is fitted as a Dirichlet distribution [22] based on the emission counts $\{N^i_{mn}\}$. In the testing phase, a symbol string $\widetilde{S}$ of finite length is fitted into every PFSA K^i and the associated emission counts $\{\widetilde{N}^i_{mn}\}$ are computed accordingly for each class. Then, the probability $\Pr(\widetilde{N}^i|\Pi^i)$ of observing the test symbol string $\widetilde{S}$ conditioned on a given state Q^i and emission matrix Π^i is modeled as a multinomial distribution [22]. Thus, the probability $\Pr(\widetilde{S}|S^i)$ of observing a symbol string $\widetilde{S}$ in the testing phase given a symbol string S^i in the training phase is computed by combining $\Pr(\widetilde{N}^i|\Pi^i)$ and $f_{\Pi^i|S^i}$ (see Eq. (13)). A classification decision is made by choosing the class, corresponding to the maximum of posterior probability $\Pr(C_i|\widetilde{S})$ (see Eq. (14)).

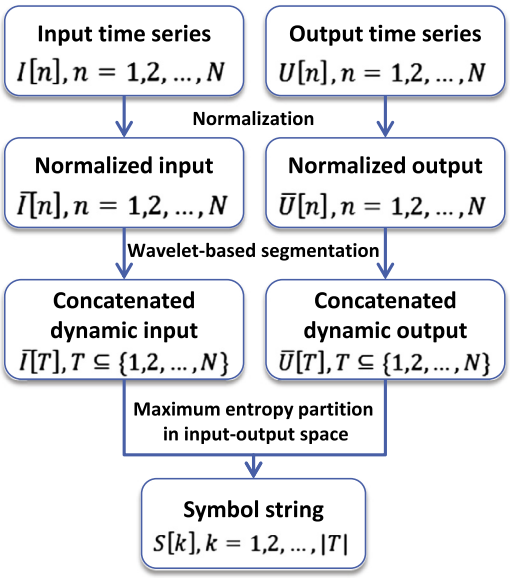


Fig. 3. Flow chart for parameter identification via symbolization of input–output data.

4. Experimental validation

This section validates the algorithms of battery parameter identification with an ensemble of experimental data that have been collected from a commercial-scale lead-acid battery.

4.1. Data acquisition and pre-processing

A fresh (12*V* AGM VRLA with 56 Ah capacity) lead-acid battery has been used in the experiments. The battery was charged/discharged according to given input (current) profiles at room temperature and an ensemble of synchronized time-series of the input charge/discharge current and output voltage responses has been collected at the sampling frequency of 1 Hz. A typical input current profile for this experiment is shown in Fig. 5.

The input profile are repeated "hotel-pulses" cycles. Each individual "hotel-pulses" cycle (i.e., duration of ~120 s) consists of a "hotel" load (i.e., relatively steady discharge due to "hotel" needs like lighting and other electrical equipments) and a discharge pulse (i.e., acceleration) followed by a charge (i.e., regenerative braking) pulse [23], as shown in Fig. 5. The amplitude of the "hotel" load and the discharging & charging pulses are numerically fluctuating in the experiment, which made each cycle different from others. This pattern of input cycles largely simulates a real-time working condition for an electric locomotive.

Remark 4.1. In many instances of industrial applications, the designer may not have the detailed knowledge of the anticipated load profile to which the battery system will be subjected. This

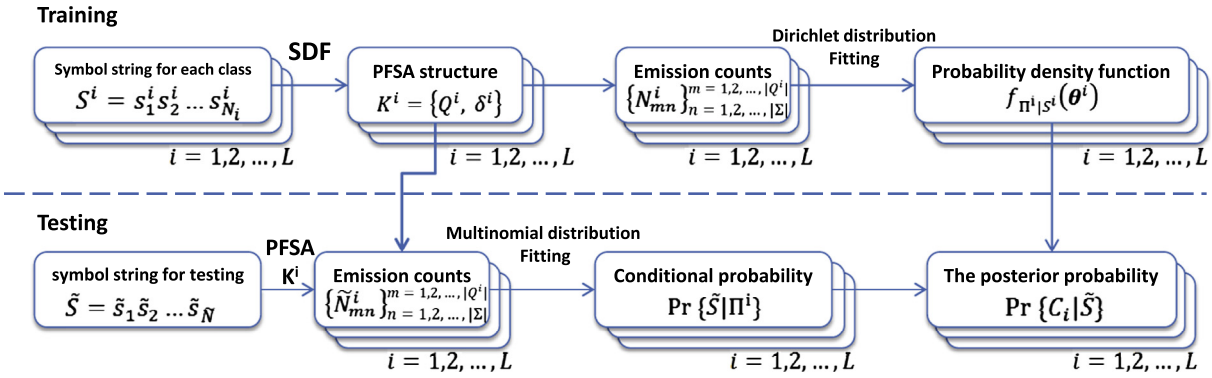


Fig. 4. Flow chart of the proposed method training and testing phases.

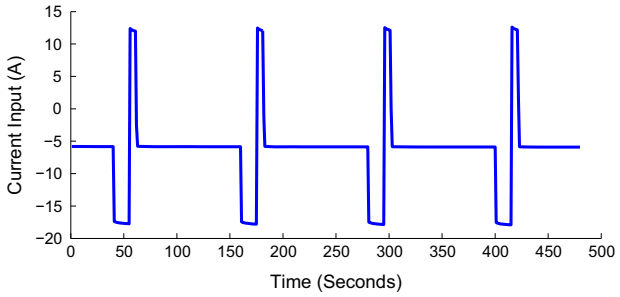


Fig. 5. Profile of input current data for the battery experiment.

procedure of battery testing under a "hotel-pulses" load is analogous to pseudo-random excitation [24] that is a standard practice for system identification of electromechanical systems in various industrial applications when testing in the actual operating environment may not be feasible. The rationale for this procedure is that a "hotel-pulses" signal could be viewed as a combination of different types of excitation signal that the battery could be subjected to. Since the proposed algorithm is executed on the synchronized pair of input (current) and output (voltage) time series, testing in an industrial setting is expected to yield satisfactory results.

This experiment has been conducted over a wide range of SOC at different battery aging stages (i.e., different values of the battery state-of-health (SOH)). Table 1 presents the coverage of battery operating conditions within the operating range of the experiment.

The raw time series of input current and output voltage are individually normalized, followed by wavelet-based segmentation based on the time series of output voltage. While segmentation extracts the relevant segments of normalized data based on the information of their frequency contents, normalization involves time translation and (down or up)-scaling of the raw data for conversion into zero-mean unit variance time series as:

$$x[n] = \frac{x_{raw}[n] - \mu_N[n]}{\sigma_N[n]} \tag{17}$$

Table 1
Experiments at different SOH stage and SOC range.

SOH stage	SOC range
1 (new)	0.53–0.99
0.92	0.50–0.99
0.86	0.49–0.99
0.82	0.47–0.99
0.80	0.42–0.99

where $\mu_N[n]$ is the mean value and $\sigma_N[n]$ is the standard deviation of the time series over a time span of N data points centered at the time index n. The experimental data of each duty cycle (i.e., both input current and output voltage) are normalized in this moving average fashion. First, the data set is smoothed under moving average with a shift window of 240 s (i.e., 240 consecutive data points). Then, each element of that data set is divided by its own standard deviation. Finally, the normalized data turn out to be zero-mean and unit-variance time series. Fig. 6(b) and (e) provide two examples of normalized current and voltage data.

Fig. 7 elucidates the notion of wavelet-based segmentation over a small window of normalized output. It depicts the normalized output voltage, the segmented normalized output, and the corresponding normalized sum of absolute values of wavelet coefficients that are obtained from the chosen scales. The dynamic characteristics of the tested battery are identified from the segments of discharging and charging pulses in each "hotel-pulses" cycle, which are in the domain of absolute values of wavelet coefficients. Using the wavelet-based segmentation algorithm (see Section 2.2), dynamic segments of the output time series are drawn from the original data and an appropriate choice of the threshold is made for the summed wavelet coefficients. Then, the information-relevant segments are concatenated together, tail to head, in the order of the original data. The resuslting segmented time series sets contain only "hotel-pulses" dynamic parts (i.e., the parts of the original data containing low-amplitude noise, which carry no significant information, are discarded).

Fig. 6(c) and (f) exhibit typical results for synchronized input current and output voltage after segmentation based on the voltage data only. It is noted that the data length after wavelet-based segmentation is much shorter than the original data length.

Fig. 8 exemplifies typical profiles of normalized and segmented input–output pairs from the ensembles of "hotel-pulses" cycles at different charging conditions, where the dynamic responses under the same input current pattern are significantly distinguishable at different levels of SOC. Fig. 9 presents the symbol strings generated corresponding to four different types of partitioning as specified in Section 2.3, where the identical alphabet size of 4 was chosen for all (i.e., $|\Sigma|_{in} = |\Sigma|_{out} = 4$ and $|\Sigma|_{magnitude} = |\Sigma|_{phase} = 4$, and total alphabet size $|\Sigma| = 4 \times 4 = 16$). It is observed from Fig. 9 that the symbol strings generated from different partitioning are different as the symbols represent regions in the input–output space.

5. Results and discussion

This section presents the details of the proposed SOC identification scheme that is built upon online pattern classification and STSA-based feature extraction from time series pairs of input–output (i.e., battery current–voltage) data. The performance and

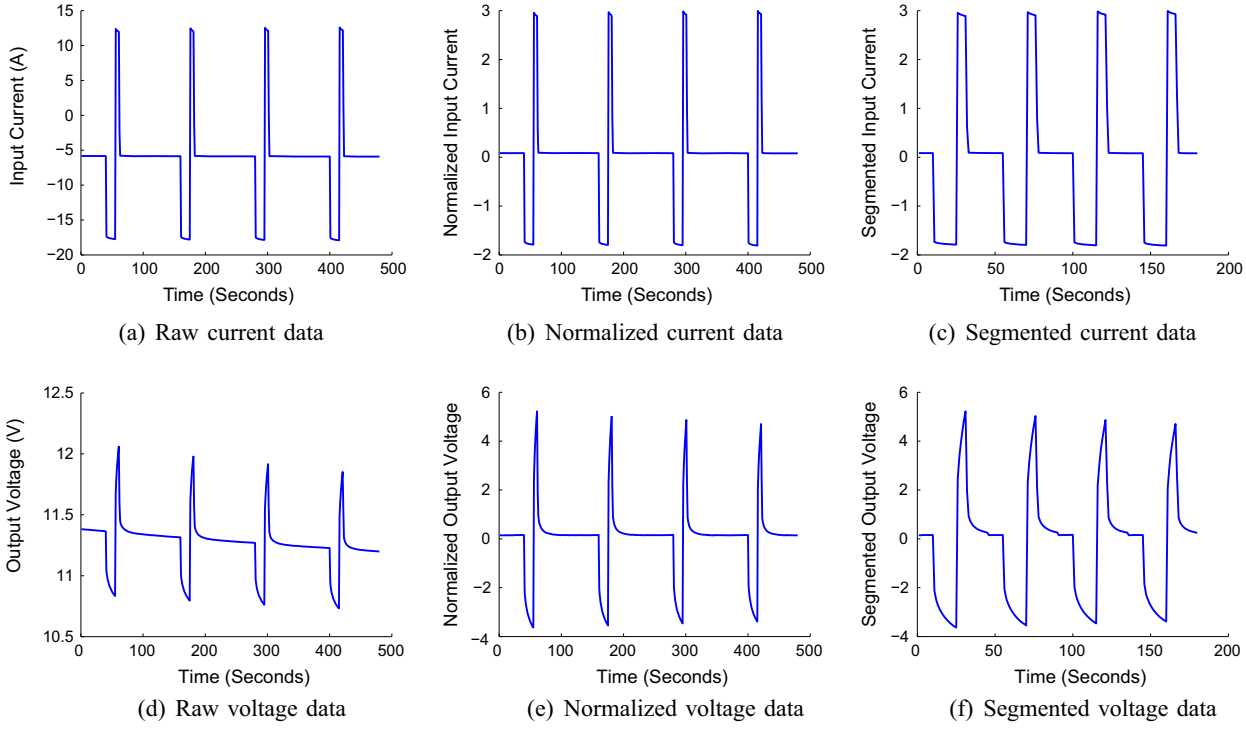


Fig. 6. Typical examples of normalization and segmentation of synchronized input current and output voltage data.

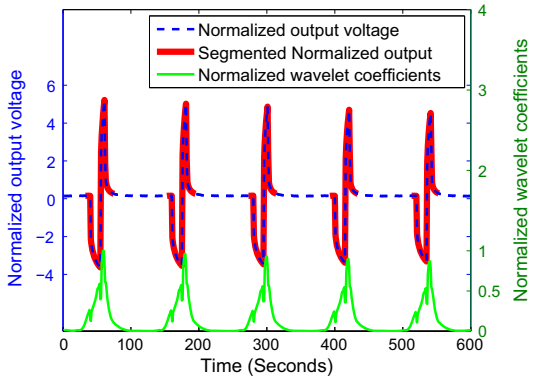


Fig. 7. Wavelet-based segmentation for normalized output.

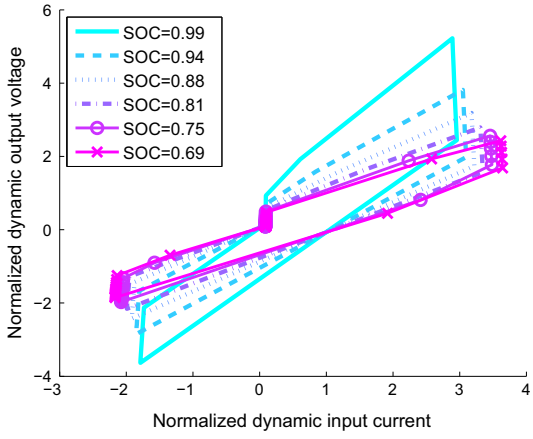


Fig. 8. Normalized input–output pairs at different SOC conditions.

robustness of SOC identification are investigated by cross validation [25] at different operating conditions of the battery and parameter settings in proposed method.

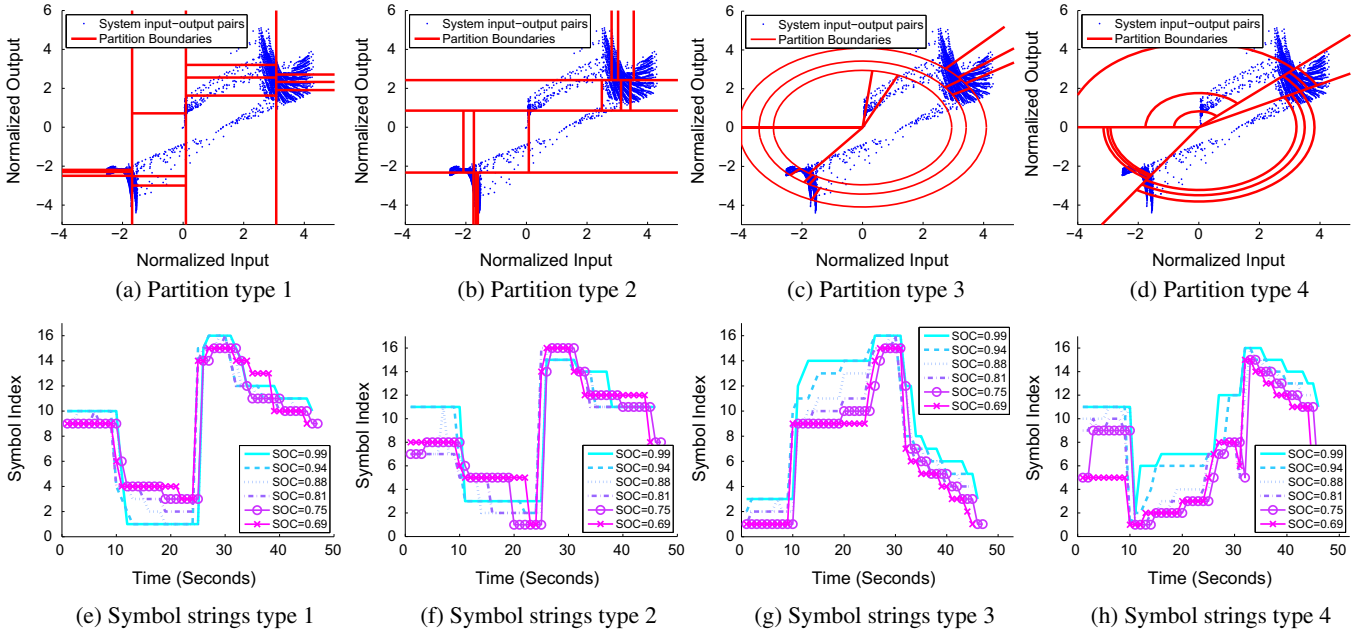
5.1. Performance of SOC class identification

This subsection presents the parameters for feature extraction (via STSA analysis) and pattern classification to generate the results.

- The partition type I in Cartesian coordinates is applied. The alphabet size for input is $|\Sigma|_{in} = 3$ and alphabet size for output is $|\Sigma|_{out} = 5$, and the total alphabet size is $|\Sigma| = |\Sigma|_{out} \times |\Sigma|_{in} = 15$.
- The depth in the D-Markov machine is set at $D = 1$, which implies that $|\Sigma| = |Q|$.
- The training/tested data are collected at the battery operating condition at $SOH = 1$ (new) and the SOC range of 0.53–0.99.
- The number of SOC classes assigned for classification problem is $NC_{SOC} = 8$. The "hotel-pulses" cycles are equally assigned to each class. The SOC range for each class is demonstrated in Table 2.
- The length of training data is $L_{train} = 5000$.

Since the number of "hotel-pulses" cycles are required to be the same for each class, the SOC range for each class may differ from each other (see Table 2) due to uneven data collection at different SOC range.

Fig. 10 shows typical profiles of posterior probability of each class as functions of the length of observed test data at three different values of SOC. Fig. 10(a) exhibits correct classification even for very short data lengths (e.g., ~150), which implies that the algorithm is able to correctly predict the class after observing test data that are of very short length. Fig. 10(b) shows a more general scenario, where modestly longer (e.g., ~200) test data are needed for



(a) Partition type 1 (b) Partition type 2 (c) Partition type 3 (d) Partition type 4

(e) Symbol strings type 1 (f) Symbol strings type 2 (g) Symbol strings type 3 (h) Symbol strings type 4

Fig. 9. Examples of symbol strings for four different types of partitioning; $|\Sigma|_{in} = |\Sigma|_{out} = 4$ and $|\Sigma|_{magnitude} = |\Sigma|_{phase} = 4$.

Table 2
SOC range for different classes.

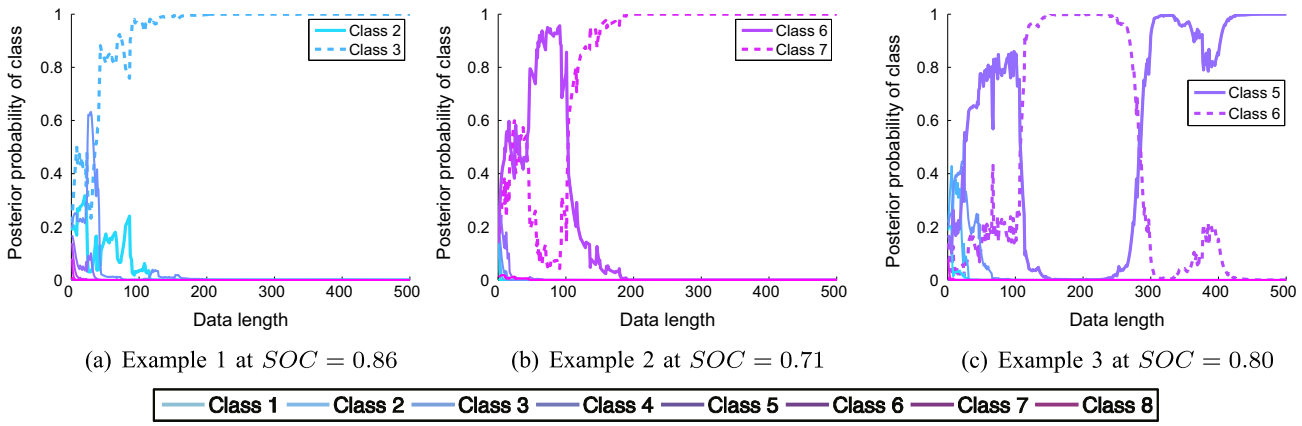
Class index	SOC range	Number of cycles
1	0.53–0.63	230
2	0.63–0.68	232
3	0.68–0.72	232
4	0.72–0.76	232
5	0.76–0.80	231
6	0.80–0.84	232
7	0.84–0.88	232
8	0.88–0.99	233

correct classification. Fig. 10(c) portrays a worse scenario, which shows the need of having significantly longer test data to achieve acceptable levels of accurate classification.

5.2. Robustness of SOC class identification

To evaluate the robustness of SOC class identification at different operating conditions and parameter settings, cross validation testing [25] is introduced by randomly choosing training/testing data sets from the ensemble of experimental data. For each class at corresponding SOC range, 50 training/testing processes have been conducted by randomly selecting training data of required data length. Presented below are the cross validation results of average misclassification rate for SOC identification at different lengths of test data. (Note that, under different situations, same parameters settings and the operating conditions have been used as provided in the previous subsection unless any changes are specifically mentioned.)

Fig. 11 presents the results of cross validation under partition type 1 with varying alphabet size $|\Sigma|_{out}$ for output while the alphabet sizes of the input held fixed at $|\Sigma|_{in} = 3$. As $|\Sigma|_{out}$ is increased, dynamic characteristics of the output are expected to be captured to a larger extent through the symbolization process, which leads to reduction of the misclassification rate as seen in plots of Fig. 11. The misclassification rate with $|\Sigma|_{out} = 6$ is approximately half the value with $|\Sigma|_{out} = 3$ when the length of the tested data is of 400 and larger. The classification accuracy is also improved as more testing data is used. The improvement in performance due to increased length of test data is more significant for the first 10 cycles of observed test data.



(a) Example 1 at $SOC = 0.86$ (b) Example 2 at $SOC = 0.71$ (c) Example 3 at $SOC = 0.80$

Fig. 10. Profiles of classification accuracy at different values of SOC.

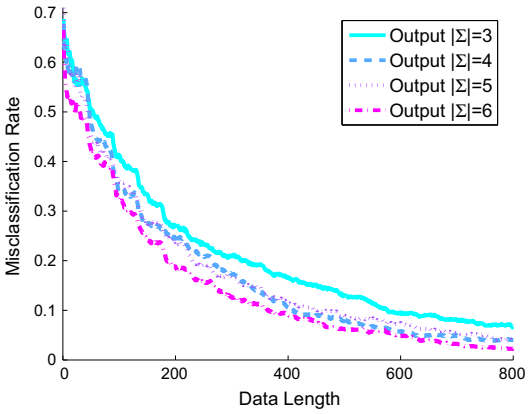


Fig. 11. Cross validation with different output alphabet size $|\Sigma|$.

Fig. 12 presents the cross validation results under different partition types. These results have been generated for alphabet sizes assigned as: $|\Sigma|_{in} = |\Sigma|_{out} = 4$ for partition types 1 and 2; and $|\Sigma|_{magnitude} = |\Sigma|_{phase} = 4$ for partition types 3 and 4. The misclassification rates for all four partition types become increasingly similar as the length of test data is increased. It is noted that the results for partition type 1 presented in Fig. 12 have better performance as compared to those in Fig. 11 for $|\Sigma|_{out}$ in the range of 3–5. Increasing the size $|\Sigma|_{in}$ of the input alphabet also improves the classification accuracy.

Fig. 13 presents the results of cross validation for different lengths of training data. For each training/testing process in cross validation, training data are randomly selected from all available data for each class. The results present the average misclassification rate from all 50 training/testing processes. The SOC parameters identified in the training phase become more accurate as the training data length is increased, which reduces the misclassification rate. It is observed that the improvement in accuracy by increasing the length of training data beyond 4000 is insignificant.

Fig. 14 presents the results of cross validation at different battery aging stages (i.e., different SOH values). Since the dynamic characteristics of the battery evolve its age (i.e., SOH) [7,11], the training data are selected distinctively at each SOH stage. The results shows that the proposed method has consistent performance at different battery aging stages, expect from the noticeable difference when the battery is quite aged (i.e., SOH = 0.80). This robustness property increases the reliability of predicted performance at different battery operating conditions, which is determined by two major factors: SOC and SOH.

Fig. 15 presents the results of cross validation with assignment of different SOC classes. As the number of classes is increased, the average resolution of the SOC range for each class becomes finer, as shown in Table 3. Keeping the length of training data fixed, as the number of SOC classes is increased, it is possible to use shorter test data for each class. However, for a larger number of SOC classes, the misclassification rate tends to increase significantly and the convergence rate of the algorithms becomes slower.

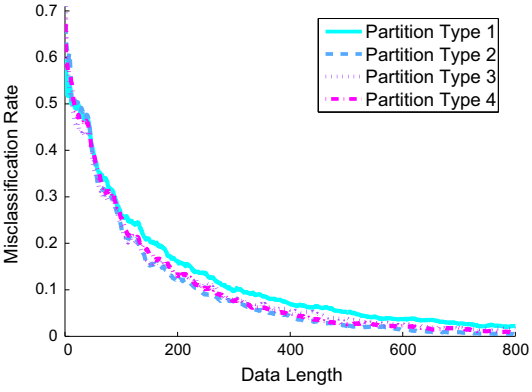


Fig. 12. Cross validation with different types of partitioning.

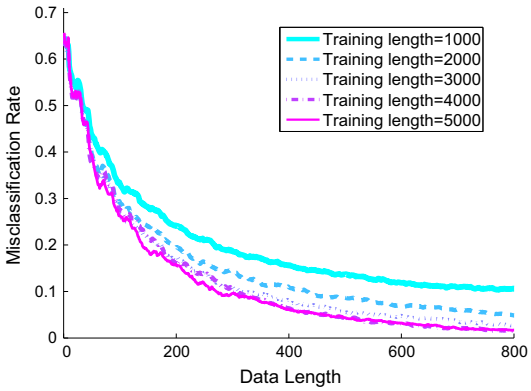


Fig. 13. Cross validation with different lengths of training data.

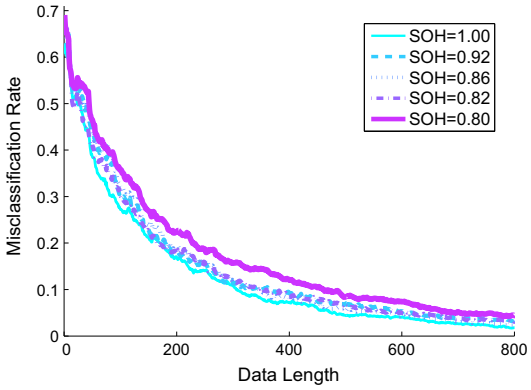


Fig. 14. Cross validation at different aging stages (i.e., different SOH).

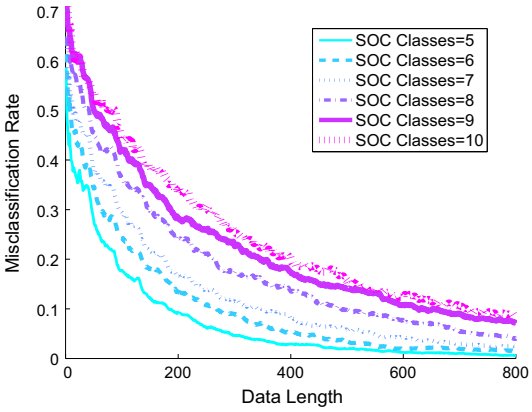


Fig. 15. Cross validation with different number of SOC classes.

5.3. Computational costs

This subsection presents a statement of the computational costs (i.e., execution time) of the SOC parameter identification algorithm.

Table 3
Average SOC resolution for different class assignments.

Number of classes	Average SOC resolution
5	0.082
6	0.068
7	0.058
8	0.051
9	0.045
10	0.041

Table 4
Execution time for test data with a length of 1000 under different conditions.

Time unit: second	Number of SOC classes (NC_{SOC})					
$\lvert\Sigma\rvert_{in} \times \lvert\Sigma\rvert_{out}$	5	6	7	8	9	10
$3 \times 3 = 9$	10.11	12.21	14.23	15.89	17.82	20.44
$3 \times 4 = 12$	13.73	16.35	19.06	21.49	24.06	26.58
$3 \times 5 = 15$	17.70	21.01	24.51	27.79	31.27	34.40
$3 \times 6 = 18$	22.76	26.46	30.87	35.20	39.71	43.82

In this paper, all results have been generated on a single core 3.6 GHz CPU with 12 GB RAM.

Li et. al. [11] addressed the issue of computation cost of a similar feature extraction tool with a much longer data set. It shows that the proposed STSA tool for online feature extraction requires very small execution time and memory. Since the training phase is conducted off-line and the length of training data for SOC identification is much less than that for SOH identification in [11], the computation cost of the proposed SOC identification does not have a major significance in the training phase.

In the testing phase, when a new set of input–output pair is acquired, the posterior probabilities for all classes need to be updated in real time. For example, if the sampling frequency is 1*Hz* in the experiment, then the execution time of the online algorithm should in the order of a fraction of a second. There are two major parameters in the feature extraction process that may affect the execution time: (i) the number of SOC classes NC_{SOC} and (ii) total alphabet size $|\Sigma|$. Table 4 presents the required execution time for analyzing test data with a length of 1000 under different combinations of the two parameters, NC_{SOC} and $|\Sigma|$. The proposed method is capable of analyzing test data of length 1000 well within 50 s for the combinations shown in Table 4, so that the updating of posterior probabilities is capable of safely accommodating a sampling frequency as high as 20 Hz. This observation satisfies the requirement of online execution for a vast majority of practical applications.

6. Summary, conclusions, & future work

The proposed method of pattern classification for battery state-of-charge (SOC) identification is built upon the concept of Symbolic Time Series Analysis (STSA) [8]. In this setting, time series of synchronized pairs of battery input (i.e., current) nd output (i.e., voltage) has been analyzed for information compression and feature extraction via D-Markov machine modeling. The online battery SOC identification is posed as a pattern classification problem that is addressed in terms of the Dirichlet/multinomial distribution for training/testing data. The performance and robustness of proposed method has been validated on experimental data of a commercial scale lead-acid battery. The pertinent conclusions drawn from the work reported in this paper are summarized below.

- Symbolic Time Series Analysis (STSA), as a low-complexity feature extraction tool, is capable of real-time execution on in-situ computational platforms (e.g., sensor nodes of battery cells). It provides a computationally efficient and electrochemistry-independent method of identifying the battery SOC parameter as an alternative to physics-based modeling analysis.
- Extracted STSA features capture the information, embedded in the input–output time series, for SOC identification. The underlying software can be implemented on a large scale package of battery cells.
- By combined Dirichlet/multinomial distribution fittings in the training/testing phases, the proposed identification method provides good and consistent performance with short-length test data under different operating conditions. For Kalman filtering based approaches, the converge time from initial SOC estimation.

For model-based SOC estimation, equivalent-circuit models make use of extensive empirical data for parametrization [26], while electrochemical models are usually partially observable [27] and their embedded nonlinear and coupled nature requires certain model simplification (e.g., model order reduction) before they can be used for industrial application. On the other hand, Kalman filtering for SOC estimation is prone to the problem of non-convergence of the estimation error. Although a data-driven approach may require adequate data length for both training and testing phases, the proposed method provides quantitative evaluation of the average estimation accuracy that is dependent on training and testing data lengths. More importantly, the proposed method is robust to measurement noise [8] and there is no issue of overfitting that is very common in data-driven methods [4–6].

Further theoretical and experimental research is needed to resolve several issues before the proposed method can be applied in practice. For example, all experimental data used in this paper are generated from the same battery. Although the proposed method has been demonstrated to be robust under different training and testing data sets via cross validation process, it is desirable to generate training and testing data from different battery cells with similar specifications to evaluate the performance of the proposed method. Nevertheless, authors suggest the following topics of future research for application of the proposed method for SOC identification.

- Usage of D-Markov machines [10] with $D \geqslant 1$ to accommodate longer memory of synchronized symbolic input–output time series.
- Extension of the proposed method to achieve robust performance for changing patterns of the input profiles (e.g., for stochastic nature of the charging and discharging current inputs) in actual operating environments.
- Investigation of the impact of temperature changes on battery dynamics for SOC identification.
- Validation of the proposed method for other types of batteries (e.g., Li-ion and Ni-MH) as well as for different discharge or charge cycle patterns.

Acknowledgement

The authors wish to thank Professor Christopher D. Rahn who provided the experimental data used in this paper.

Appendix A. Algorithms

This appendix lists three algorithms that are used in the main body of the paper.

Algorithm 1. Wavelet-based Segmentation

Require: Perform the following:
Select the sampling time period Δ.
Collect finite-length time series $x[n], n = 1, \ldots c, N$.
Select a wavelet basis function ψ with central frequency f_c; and the (scalar) threshold parameter ξ_T for selection of wavelet coefficients for segmentation.

Ensure: From the collected time series, execute the following:
Compute power density $S_x(f)$ of the time series $x[n]$ for the frequency window $f \in [0, \frac{1}{2\Delta}]$ at N discrete points. Choose a sequence of frequency points $\{\varphi_m\}_{m=1}^{M}$ at local peaks of the energy density coefficients.

1: **for** $m = 1$ to M **do**
2: Compute the corresponding wavelet scale α_m by substituting frequency points φ_m.
3: Obtain the wavelet coefficients for the scale α_m as translated versions along time series $\tilde{x}_{\alpha_m}[\tau_\ell^m] = \tilde{x}[\alpha_m, \tau_\ell^m]$.
4: **end for**
5: Compute the summation of the wavelet coefficients at each time shift indices $\tilde{x}[\tau_\ell]$
6: Identify the set of time indices T by thresholding on the summated wavelet coefficients $T = \{\tau_\ell | \tilde{x}[\tau_\ell] > \xi\}$.

Algorithm 2. Maximum Entropy Partitioning for 2-dimensional Time Series

Require: A 2-dimensional string $X[n] = [x_1[n] x_2[n]]$ for $n = 1, 2, 3, \ldots, N$ of synchronized and normalized time series data set; Alphabet size $|\Sigma_1|$ for the first coordinate of the 2-dimensional data and alphabet size $|\Sigma_2|$ for the second coordinate of the 2-dimensional data.

Ensure: Partition vector $\wp_1 \in \mathbb{R}^{|\Sigma_1|+1}$ for first coordinate data; Partition Matrix $\wp_2 \in \mathbb{R}^{|\Sigma_1| \times (|\Sigma_2|+1)}$ for the 2-dimensional data set.

Assign $\wp_1(1) = -\infty$, i.e., the minus infinity
Assign $\wp_2(m, 1) = -\infty$, for $m = 1, 2, \ldots, |\Sigma_1|$
Assign $\wp_1(|\Sigma_1| + 1) = \infty$, i.e., the positive infinity
Assign $\wp_2(m, |\Sigma_2| + 1) = \infty$, for $m = 1, 2, \ldots, |\Sigma_1|$
Sort the data string x_1 in the ascending order as x_1^s
Let $K = length(x_1^s)$
for $i = 1$ to $|\Sigma_1|$ **do**
  if $i \neq |\Sigma_1|$ **then**
    $\wp_1(i+1) = x_1^s \left[ceil\left(\frac{i \times K}{|\Sigma_1|}\right) \right]$
  end if
  Define $x_2^i \triangleq \{x_2[n] | \wp_1(i) < x_1[n] < \wp_1(i+1)\}$
  Sort the data string x_2^i in the ascending order as x_2^{is}
  Let $L = length(x_2^{is})$
  for $j = 1$ to $|\Sigma_2| - 1$ **do**
    $\wp_2(i, j+1) = x_2^{is} \left[ceil\left(\frac{j \times L}{|\Sigma_2|}\right) \right]$
  end for
end for

Algorithm 3. Symbolic Time Series Analysis (STSA) for Feature Extraction

Require Symbolic strings of length N obtained at I different operating conditions of system under analysis: $S_i = \{s_i^1, s_i^2, s_i^3, \ldots, s_i^N\}, i = 0, 1, \ldots, I-1$; the alphabet size $|\Sigma| = |\Sigma_1| \times |\Sigma_2|$; the depth D of Markov machine; and number of states $|Q|$, where $|Q| \leqslant |\Sigma|^D$.

Ensure Extracted emission matrices $\Pi_i \in \mathbb{R}^{|\Sigma|^D \times |\Sigma|}, i = 0, 1, \ldots, I-1$ for each symbol string and the feature divergences $\{m_i\}_{i=0}^{I-1}$.

1: Initialize $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_{|\Sigma|}\}$, and $Q = \{q_1, q_2, \ldots, q_{|Q|}\}$.
2: **for** $i = 0$ to $I - 1$ **do**
3: **for** $k = 1$ to $|Q|$ **do**
4: **for** $j = 1$ to $|\Sigma|$ **do**
5: Count the number of event that symbol σ_j occurs after symbol (combination) of $q_k = \{\sigma_1^k \ldots \sigma_{|D|}^k\}$, denoted as $N(\sigma_j, q_k)$, from the symbol string S_i.
6: **end for**
7: **for** $j = 1$ to $|\Sigma|$ **do**
8: compute the estimated emission probability $\pi(\sigma_j | q_k)$ (see Eq. (5)).
9: **end for**
10: **end for**
11: Construct the estimated emission matrix Π_i for symbol string S_i (see Eq. (6)).
12: **end for**

References

[1] Xing Y, He W, Pecht M, Tsui KL. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. Appl Energy 2014;113:106–15.
[2] Wang Y, Zhang C, Chen Z. A method for state-of-charge estimation of li-ion batteries based on multi-model switching strategy. Appl Energy 2015;137:427–34.
[3] He H, Xiong R, Guo H. Online estimation of model parameters and state-of-charge of lifepo 4 batteries in electric vehicles. Appl Energy 2012;89(1):413–20.
[4] Weigert T, Tian Q, Lian K. State-of-charge prediction of batteries and battery–supercapacitor hybrids using artificial neural networks. J Power Sources 2011;196(8):4061–6.
[5] Malkhandi S. Fuzzy logic-based learning system and estimation of state-of-charge of lead-acid battery. Eng Appl Artif Intell 2006;19(5):479–85.
[6] Shi Q-S, Zhang C-H, Cui N-X. Estimation of battery state-of-charge using v-support vector regression algorithm. Int J Autom Technol 2008;9(6):759–64.
[7] Li Y, Shen Z, Ray A, Rahn C. Real-time estimation of lead-acid battery parameters: a dynamic data-driven approach. J Power Sources 2014;268:758–64.
[8] Ray A. Symbolic dynamic analysis of complex systems for anomaly detection. Signal Process 2004;84(July):1115–30.
[9] Rajagopalan V, Ray A. Symbolic time series analysis via wavelet-based partitioning. Signal Process 2006;11(November):3309–20.
[10] Mukherjee K, Ray A. State splitting and merging in probabilistic finite state automata for signal representation and analysis. Signal Process 2014;104(November):105–19.
[11] Li Y, Chattopadhyay P, Ray A, Rahn C. Identification of the battery state-of-health parameter from input–output pairs of time series data. J Power Sources 2015;285:235–46.
[12] Wen Y, Mukherjee K, Ray A. Adaptive pattern classification for symbolic dynamic systems. Signal Process 2013;93(1):252–60.
[13] Shen Z, Gou J, Rahn C, Wang C-Y. Ritz model of a lead-acid battery with application to electric locomotives. In: ASME 2011 dynamic systems and control conference and Bath/ASME symposium on fluid power and motion control. American Society of Mechanical Engineers; 2011. p. 713–20.
[14] Darema F. Dynamic data driven applications systems: new capabilities for application simulations and measurements. In: 5th International conference on computational science – ICCS 2005, Atlanta, GA (United States); 2005.

[15] Mallat S. A wavelet tour of signal processing. 3rd ed. Amsterdam (The Netherlands): Elsevier; 2009.
[16] Rahn C, Wang C-Y. Battery systems engineering. New York, NY (USA): John Wiley; 2012.
[17] Adenis P, Wen Y, Ray A. An inner product space on irreducible and synchronizable probabilistic finite state automata. Math Control Signals Syst 2012;23(1):281–310.
[18] Shalizi C, Shalizi K. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In: AUAI '04: proceedings of the 20th conference on uncertainty in artificial intelligence. Arlington, VA (USA): AUAI Press; 2004. p. 504–11.
[19] Bahrampour S, Ray A, Sarkar S, Damarla T, Nasrabadi NM. Performance comparison of feature extraction algorithms for target detection and classification. Pattern Recogn Lett 2013;34(16):2126–34.
[20] Ferguson T. A Bayesian analysis of some nonparametric problems. Ann Stat 1973:209–30.
[21] Sarkar S, Mukherjee K, Sarkar S, Ray A. Symbolic dynamic analysis of transient time series for fault detection in gas turbine engines. J Dynam Syst Measure Control 2013;135(1):014506.
[22] Wilks S. Mathematical statistics. New York, NY (USA): John Wiley; 1963.
[23] Shen Z, Cao L, Rahn CD, Wang C-Y. Least squares galvanostatic intermittent titration technique (ls-gitt) for accurate solid phase diffusivity measurement. J Electrochem Soc 2013;160(10):A1842–6.
[24] Ljung L. System identification. UpperSaddle River, NJ (USA): Prentice Hall; 1999.
[25] Bishop C. Pattern recognition and machine learning 1. New York, NY (USA): Springer; 2006.
[26] Cho S, Jeong H, Han C, Jin S, Lim JH, Oh J. State-of-charge estimation for lithium-ion batteries under various operating conditions using an equivalent circuit model. Comput Chem Eng 2012;41:1–9.
[27] Tanim TR, Rahn CD, Wang C-Y. State of charge estimation of a lithium ion cell based on a temperature dependent and electrolyte enhanced single particle model. Energy 2015;80:731–9.