Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Power Sources 141 (2005) 351–358
JOURNAL OF POWER SOURCES
www.elsevier.com/locate/jpowsour

Support vector based battery state of charge estimator

Terry Hansen, Chia-Jiu Wang*

Department of Electrical and Computer Engineering, University of Colorado at Colorado Springs, 1420 Austin Bluffs Parkway, P.O. Box 7150, Colorado Springs, CO 80933, USA

Received 12 September 2004; accepted 24 September 2004
Available online 30 November 2004

Abstract

This paper investigates the use of a support vector machine (SVM) to estimate the state-of-charge (SOC) of a large-scale lithium-ion-polymer (LiP) battery pack. The SOC of a battery cannot be measured directly and must be estimated from measurable battery parameters such as current and voltage. The coulomb counting SOC estimator has been used in many applications but it has many drawbacks [S. Piller, M. Perrin, Methods for state-of-charge determination and their application, J. Power Sources 96 (2001) 113–120]. The proposed SVM based solution not only removes the drawbacks of the coulomb counting SOC estimator but also produces accurate SOC estimates, using industry standard US06 [V.H. Johnson, A.A. Pesaran, T. Sack, Temperature-dependent battery models for high-power lithium-ion batteries, in: Presented at the 17th Annual Electric Vehicle Symposium Montreal, Canada, October 15–18, 2000. The paper is downloadable at website http://www.nrel.gov/docs/fy01osti/28716.pdf] aggressive driving cycle test procedures. The proposed SOC estimator extracts support vectors from a battery operation history then uses only these support vectors to estimate SOC, resulting in minimal computation load and suitable for real-time embedded system applications.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Support vector machine (SVM); Support vector regression; State of charge (SOC); US06

1. Introduction

As electric vehicles (EVs) gain popularity, the importance of correct estimation of the current state of charge (SOC) of a battery increases. The owner of a gasoline-powered car needs to know how much gasoline is in the tank. In the same way, the owner of an EV needs to know how much energy is in the battery at any time. Unfortunately, SOC cannot be measured directly but must be estimated based on measurable battery parameters such as current and voltage.

Many battery management systems (BMSs) estimate SOC with some variation of the basic coulomb counting method [1]. In coulomb counting, the current entering and leaving the battery is measured periodically, then SOC of the battery is updated by adding or subtracting the last period's net cumulative charge. Support vector machines (SVMs) [3] have been successfully applied for classification and regression especially in highly nonlinear systems. In the work presented here, we build a support vector machine (SVM) using a lithium-ion-polymer (LiP) battery pack testing data covering the expected range of operation. The SVM is then optimized by comparing its output with data obtained during a simple SOC test of a LiP battery pack. Finally, the SVM is tested with US06 [2] dynamic operational data from U.S. Department of Energy's Hybrid Electrical Vehicle program. In the dynamic tests, the SVM estimates battery SOC with a root-mean-squared error of 5%, 5.76% and 2.54% for three cases, respectively.

There has been much research into the applications of SVMs, as well as into new methods in estimating battery SOC. However, there has been little work in applying SVMs to estimate battery SOC. This paper presents a new SOC method based on support vector regression approach.

* Corresponding author. Tel.: +1 719 2623495; fax: +1 719 2623589.
E-mail address: cwang@eas.uccs.edu (C.-J. Wang).

0378-7753/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2004.09.020

2. Existing and proposed methods for estimating SOC

A review of existing methods for estimation of battery SOC is presented in the following. The advantages and disadvantages of the traditional coulomb counting method and advanced extended Kalman filtering (EKF) estimator [4,5] are discussed. The proposed support vector based battery SOC estimator is presented in detail.

2.1. Coulomb counting approach

The most straightforward method to estimate battery SOC is the coulomb counting method. We use the following example to explain the method and discuss the inherent drawbacks in using the simple coulomb counting approach for real-time applications.

If a battery's capacity is 6.4 ampere-hours (Ah) and it starts at 50% SOC, the battery is said to contain 3.2 Ah of charge. If it is then discharged at 36 A for 10 s, its new capacity is calculated as follows:

- $3.2\ \text{Ah} - (36\ \text{A} \times 10\ \text{s}/3600\ \text{s h}^{-1}) = 3.2 - 0.1 = 3.1\ \text{Ah}$.
- 3.1 Ah is 48.4% of the battery's original capacity of 6.4 Ah.
- The new SOC = 48.4%.

If a sufficiently accurate current sensor is used, a coulomb counter is reasonably accurate and inexpensive to implement. However, it suffers the following drawbacks.

(1) The coulomb counter is an open loop SOC estimator. Errors in the current detector are accumulated by the estimator. The longer the estimator is operated, the larger the cumulative error becomes. Also, the worse the error in the current detector, the faster the estimator produces an incorrect result.
(2) The coulomb counter does not take into account changes in the battery's capacity as the battery ages. Characteristics of aging can be estimated ahead of time and saved in the software. But the coulomb counter cannot detect and account for these factors in real time. In consequence, if the battery's aging does not follow the expected course, SOC estimation becomes more incorrect as the battery becomes older.
(3) The coulomb counter must estimate the starting SOC by measuring battery pack voltage.

For many SOC levels, this estimation carries an average ±15% error. Temperature, charging history and relaxation time can be saved and used to refine the starting SOC estimate. But with a coulomb counter, whatever error is contained in the starting estimate will be carried forward. The coulomb counter cannot detect and fix the starting error.

2.2. Kalman filtering approach

An early paper on the use of a Kalman filter (KF) to estimate SOC was published in 2001 by Pang et al. [6]. The authors discuss the application of KF modeling to an SOC system. Their report correctly points out that the KF system can overcome an incorrect initial SOC value. They also recognize that the KF model could detect and model cell aging. A more recent paper on the use of extended Kalman filter (EKF) to estimate SOC was presented by Barbarisi et al. at an IFAC symposium in April 2004 [7]. It is based on Nickel Metal Hydride (NiMH) battery technology and uses modeling data specific to that technology. Neural networks and fuzzy logic techniques are also used by several authors to build SOC estimators [8].

Plett [4,5] has proposed an EKF system for estimating SOC. The system consists of an inner filter that adjusts the SOC estimate and an outer filter that adjusts the underlying battery model. The inner filter takes the measured current and proposes a corresponding voltage based on SOC and the system model. The proposed voltage is compared with the measured voltage and the SOC is adjusted. Thus, the system feedback is voltage and its output is SOC. The outer filter monitors system current and voltage trends over long time periods. It slowly adjusts the parameters of the underlying system model so that aging and other lifetime effects are detected and incorporated into the model in real time.

The final result of the EKF approach is a highly accurate SOC estimator that maintains better than ±5% accuracy over all ranges of operation. The EKF SOC estimator maintains this level of accuracy whether the system is changing slowly or is operating in an extremely dynamic range. It compensates for inaccuracies in system detectors. If the starting SOC is incorrect, it will detect that and correct it in a short period of time. As the battery ages, the system detects capacity changes and compensates for them. Even changes in individual cells can be sensed and incorporated into the system model. The drawback of the EKF SOC estimator is its high computational complexity and implementation cost. A full implementation requires a 32-bit processor operating at 40 MHz with an on-chip hardware floating-point coprocessor.

2.3. The SVM niche

The SVMs [3] have been applied for classification in various domains of pattern recognition. However, it can also be applied to regression problems, although regression is inherently more difficult than classification. The SVM used as a nonlinear estimator is more robust than a least-squares estimator because it is insensitive to small changes. The ε-insensitivity loss function as shown in Eq. (1) proposed by Vapnik [3] is used in support vector regression

$$L_\epsilon(y, f(\mathbf{x})) = \begin{cases} |y - f(\mathbf{x})| - \varepsilon & \text{for } |y - f(\mathbf{x})| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where ε is a prescribed parameter, y is the desired response, $f(\mathbf{x})$ is the estimated output, and $\mathbf{x}$ is an input vector. The function $L_\epsilon(y, f(\mathbf{x}))$ is called the ε-insensitive loss function. The loss is equal to zero if the difference between the estimated

$f(\mathbf{x})$ and the desired response y is less than ε. The support vector (SV) regression model is to approximate the function $f(\mathbf{x})$ as the following equation where K is the kernel function

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) \quad (2)$$

The values of α_i and α_i^* are selected during the training process to minimize the loss function under the constraints:

$$\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0, \quad 0 \le \alpha_i \le C, \quad 0 \le \alpha_i^* \le C \quad (3)$$

Both C and ε parameters must be tuned simultaneously to have the best solution.

After training the SVM, we have values of α_i and α_i^*, which are both zero if $\mathbf{x}_i$ does not contribute to the loss function. Only support vectors have nonzero either α_i or α_i^*. For a new input vector $\mathbf{z}$, $f(\mathbf{z})$ is predicted as

$$f(\mathbf{z}) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{z}) \quad (4)$$

A properly optimized SVM can condense thousands of training points to a manageable number of SVs. After learning the SVs, an SVM usually does not require matrix inversions and calls to computationally intensive math functions, which are required by the EKF approach.

In the field of SOC estimation, an SVM could be designed to incorporate thousands of training data points and reduce them to a set of SVs that can be manipulated by an inexpensive 8-bit processor. If the SVM is correctly optimized, it might offer accuracy comparable to the EKF system at a price typical of a simple coulomb counter. The key to leveraging the power of SVs is to use the right training data and proper kernel functions [9,10]. The resultant SVM occupies minimal memory and calculates an SOC in minimum time. We use SVMlight [11] to determine the SVs with the source code modified so that its input parameters can be automatically varied and the resulting SVM can be tested for accuracy.

3. Experimental procedures and results

Steps used in training an SVM for SOC estimation are presented first followed by the testing procedures and results. Training steps include: choose and preprocess the training data, find the optimal SVM parameters, and choose and preprocess the testing data.

3.1. Choose and preprocess training data

Good training data should meet several criteria as follows:

(1) Training data should be different from the data that will be used for testing. If training data and testing data are identical, then the SVM is just interpolating points on a line—which is not what we expect the SVM to do. Also, the resulting SVM is not likely to perform well when given real world data that are different from the training data.
(2) Training data should cover the expected range of operation of the final SVM. For this work, training data should cover SOC from 20% to 80%, currents from +50 to −50 A and voltages in ranges that correspond to those SOC and current values. Of course, real world battery system values will sometimes exceed these limits. But an SVM that predicts SOC within these limits would certainly demonstrate that such an approach is feasible.
(3) Training data should be compatible to a real world SOC estimator and represent a continuous flow of measured battery data. Training data for the work presented in this paper are taken from cell testing data for a large scale LiP cell that is designed for use in vehicle propulsion systems. Thus, the training data is very closely related to the real data that the SVM SOC estimator will be working with. The original cell testing data covers operation from 100% SOC to 0% SOC and back up to 100% SOC. Current goes from −80 to −1 A and then from +80 to +1 A. Training data for this research are obtained by using all data points that are less than 86% SOC and greater than 16% SOC. Current goes from −75 to −10 A and then from +10 to +75 A. Several plateaus with 0 A current and steady state SOC are also included. Figs. 1 and 2 show the SOC, current and voltage ranges of the training data. Current, voltage and ideal SOC data are obtained once a second for a total of almost one hour of data points.

Data preprocessing turns out to be the key to getting the SVM to converge. Without preprocessing, very few SVM training runs converge. SVM test runs are monitored with a timer that will stop SVM training processes after 30 min of trying to find a solution. With preprocessing, almost all SVM runs converge within 1–2 min.

Preprocessing consists of scaling the data so that all input vector elements are in the range of 0.0–1.0. In addition, one additional element is added to the training vector. This element is the change in voltage in the last 1 s of operation. This element is added because voltage data changes very rapidly during pack operation. This can be observed in Fig. 2. Each training datum thus consists of a four-element vector containing current, voltage, SOC at the end of the last second (SOC_{t-1}) and the change in voltage during the last second (ΔV). This datum is then scaled to the range of 0.0–1.0. A representative training vector is shown in Table 1.

Table 1
An example of training data vectors

Element	Current	Voltage	SOC_{t-1}	ΔV
Unscaled	−20.0102	152.909	40.87806	−0.12244
Scaled	0.374936	0.245451	0.408781	0.489797

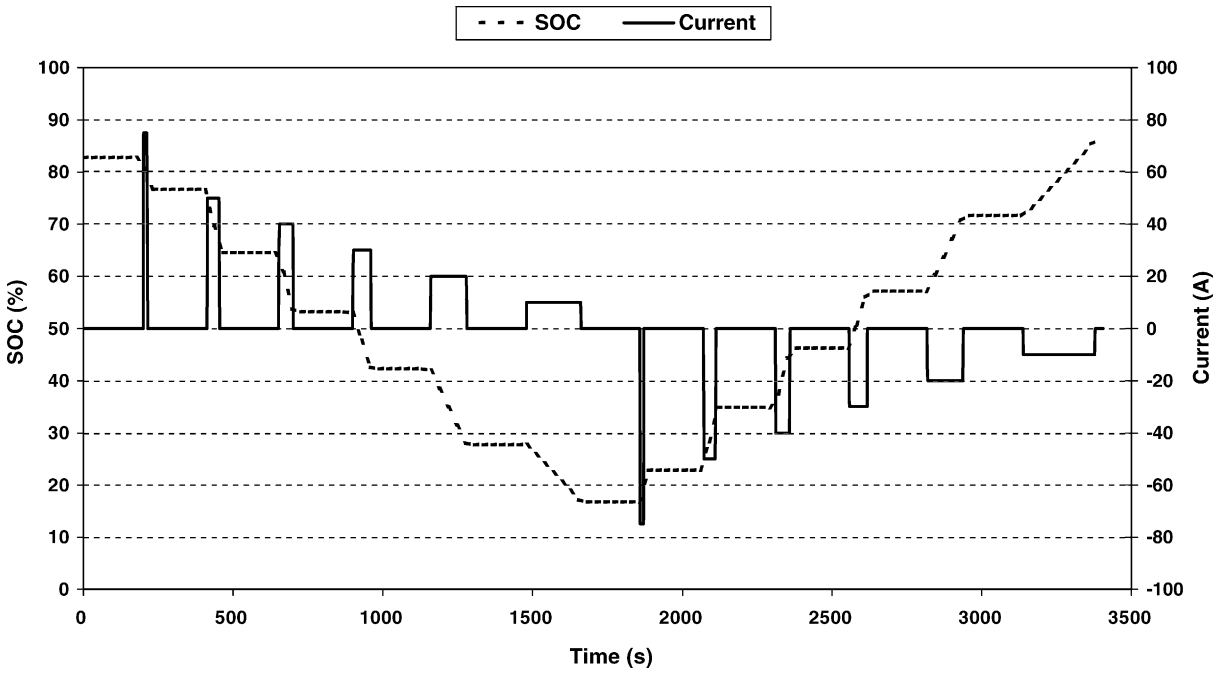

Fig. 1. SOC and current in training data.

3.2. Find the optimal SVM parameters

The SVM parameters are the constant C, the size of the error tube ε and the type of kernel function K. An increase in C penalizes larger errors and leads to a decrease in the approximation error. The best kernel is found to be the second-degree polynomial for our application. Using the second-degree polynomial, the optimal value of C is found to be between 13.86 and 14.14. In the final optimal SVM a value of 13.9 is used for C. At the same time, a value of 0.0001 is used for ε.

Within the polynomial kernel, two additional parameters can be used to fine tune the SVM. The second-degree polynomial kernel function $K(\mathbf{a}, \mathbf{b})$ is evaluated as:

$$K(\mathbf{a}, \mathbf{b}) = s \times (\mathbf{a} \cdot \mathbf{b})^2 + r \tag{5}$$

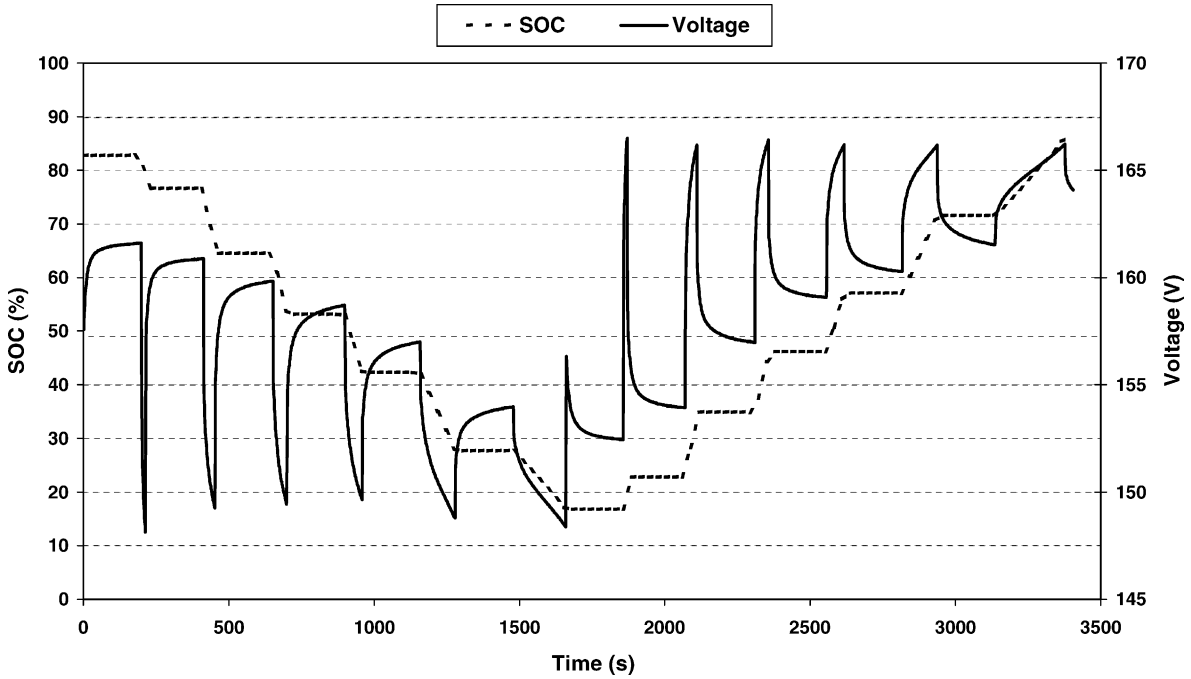

Fig. 2. SOC and voltage in training data.

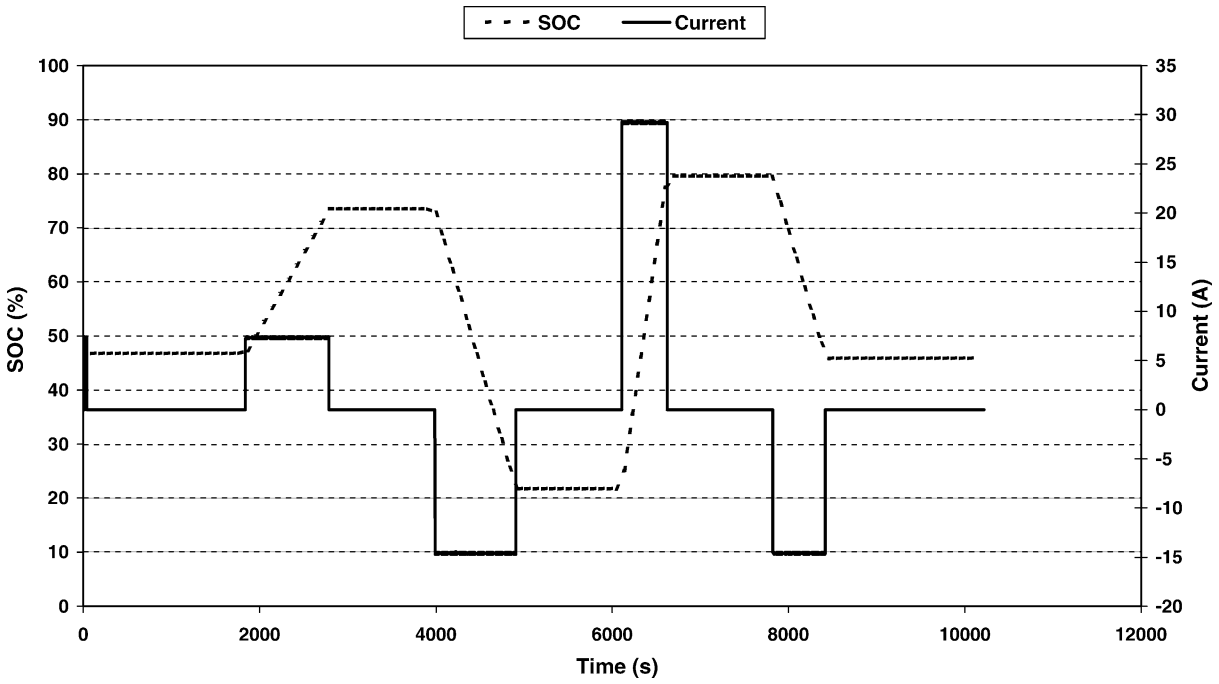

Fig. 3. SOC and current in the simple SOC test.

where **a** and **b** are vectors, s is the linear factor and r is a constant factor. The selection of s and r is purely empirical. In the final optimal SVM, $s = 7.3$ and $r = 19.7$ are used.

3.3. Choose and preprocess testing data

The target application for this SOC estimator is an EV. Unlike a hybrid electric vehicle (HEV), battery charge and discharge transients in an EV tend to be relatively mild. Currents are still large but they do not change as rapidly as in an HEV application. In order to obtain robust testing results, the testing data chosen for this work are obtained from running simple SOC tests as well as dynamic SOC tests typically used in HEVs.

A graph of typical SOC and current values for the simple SOC test is shown in Fig. 3. Fig. 4 shows typical SOC and voltage values of the simple SOC test. Preprocessing of test

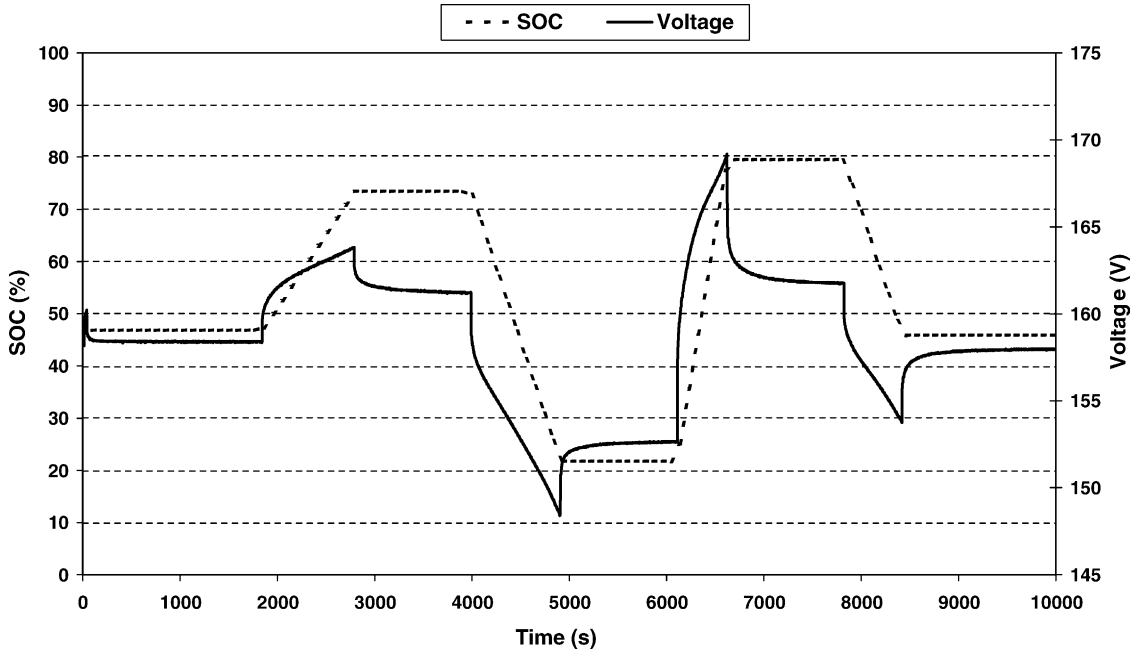

Fig. 4. SOC and voltage in the simple SOC test.

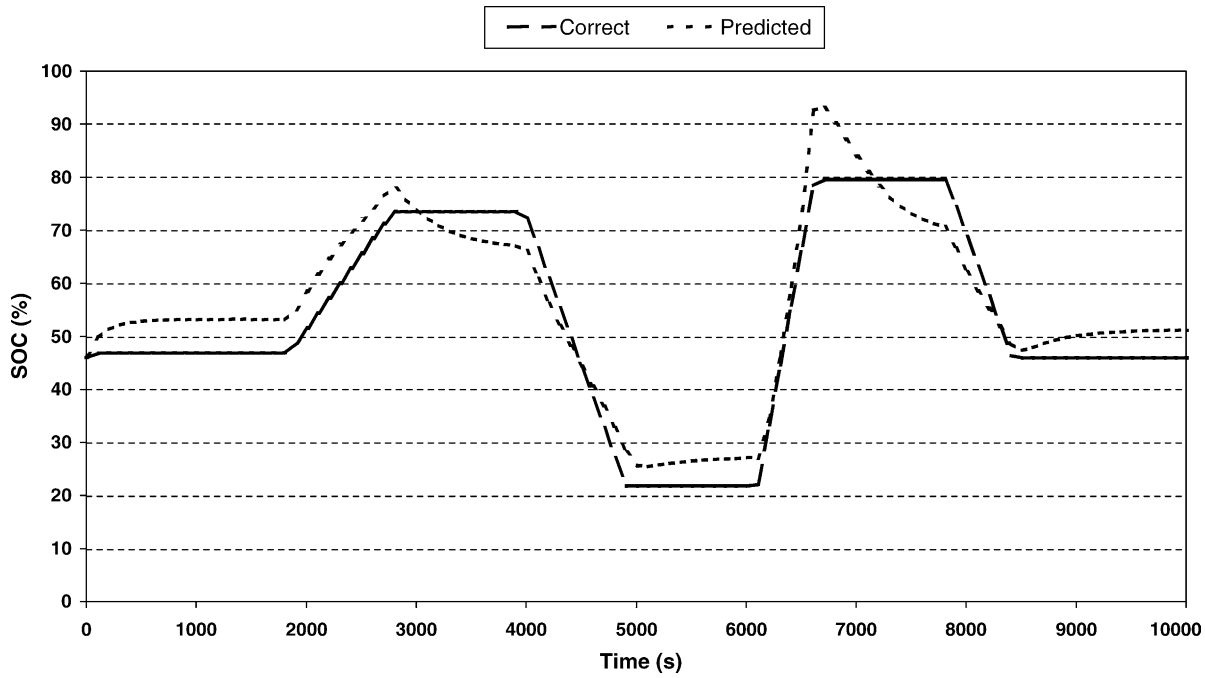

Fig. 5. Predicted SOC by optimized SVM vs. actual SOC.

data is done in the same way as training data. Thus, each test vector is a four-element vector.

3.4. Test results using the optimal SVM

3.4.1. Simple SOC test

The optimal SVM is used to predict SOC with simple SOC test inputs and the results are shown in Fig. 5.

The error between correct SOC and estimated SOC in the above test is shown in Fig. 6. The root-mean-squared error is 5% over the whole test. The maximum positive error is +16% and the maximum negative error is −9%. There is no drift over time in SOC estimation by the SVM while output drift is a major problem to the coulomb counting method. The SVM estimates SOC with the largest error when current is zero. The system seems to seek out certain preferred SOC levels

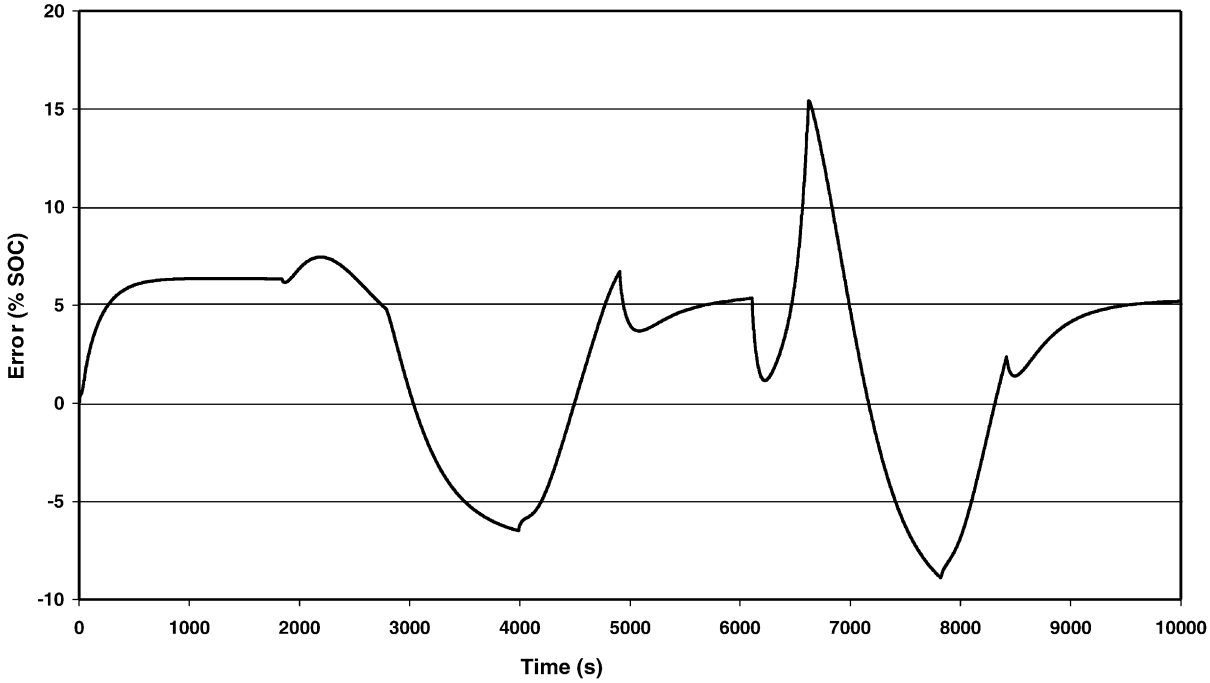

Fig. 6. Error between optimized SVM and ideal SOC.

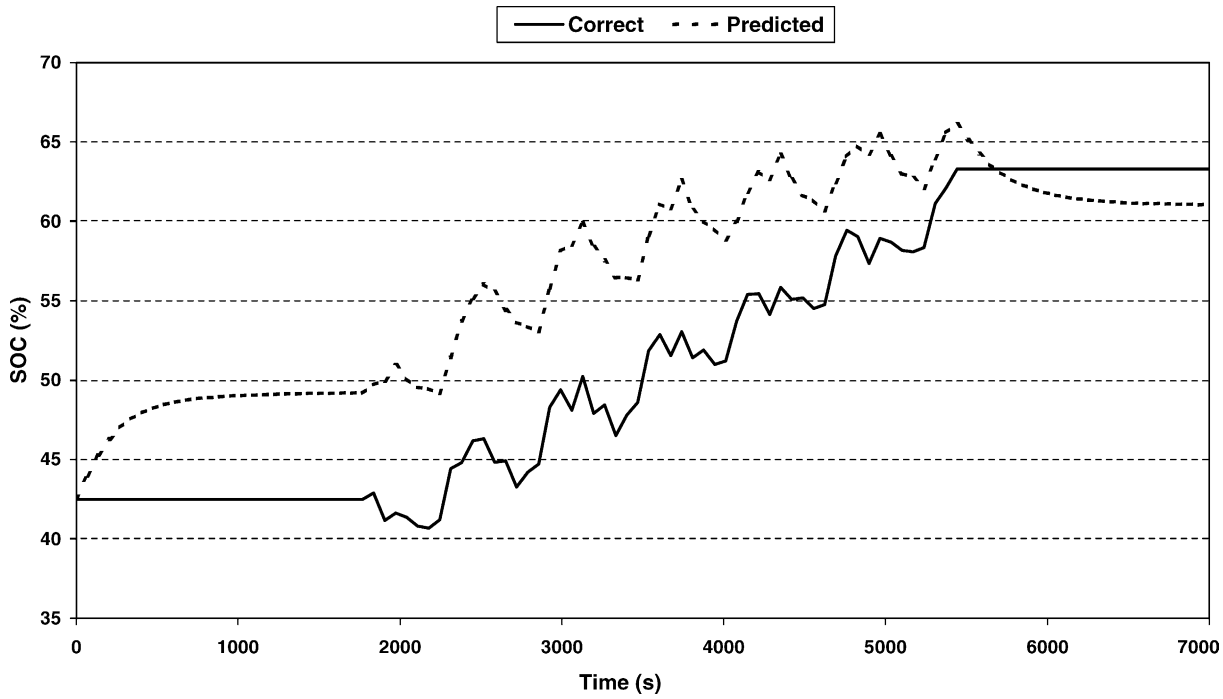

Fig. 7. Optimized SVM predicts dynamic SOC.

when in steady state. This may be an imitation of the training data, which has several well-defined plateaus where current was zero.

3.4.2. Dynamic SOC test

The optimal SVM is also tested with US06 [2], an aggressive driving cycle provided by U.S. Department of Energy's Hybrid Electrical Vehicle program. US06 data are obtained by driving an instrumented HEV on US highway 6 near Boulder, CO. The resulting current, voltage and SOC data are available to companies doing battery research and development.

US06 data are taken once a second and vary widely from 1 s to the next. Current can vary as much as 80 A in 1 s. On an average current varies by 3.5 A s^{-1}. In the simple SOC test, current does not vary from 1 s to the next except at the

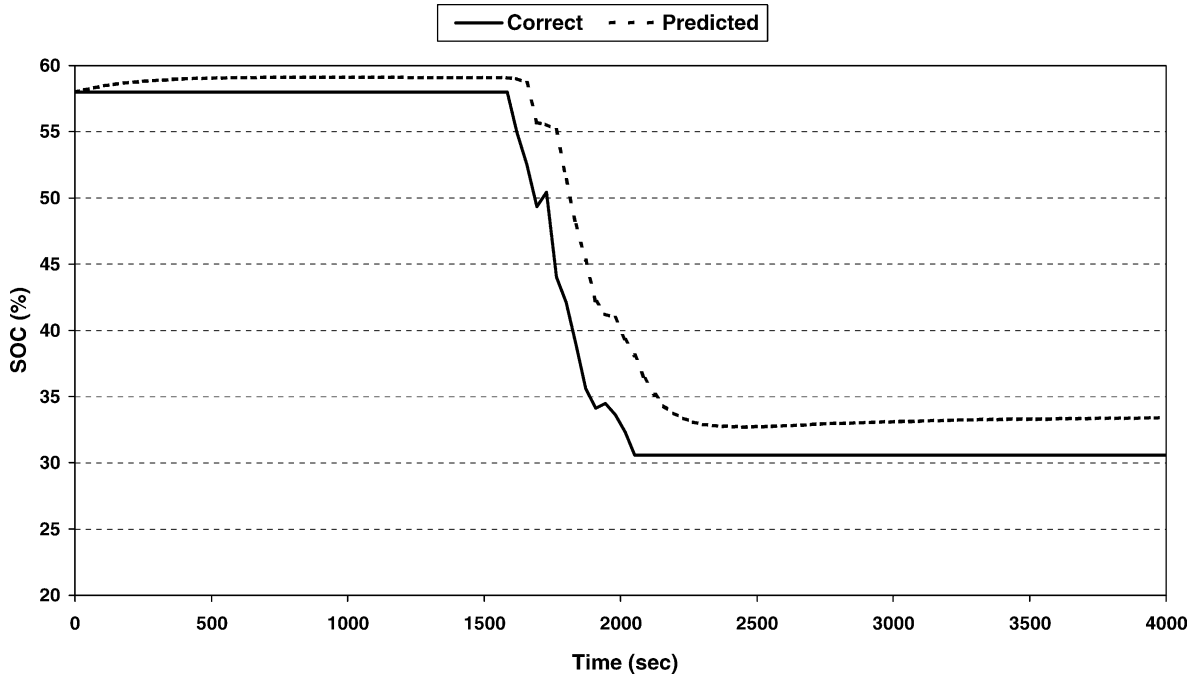

Fig. 8. Second US06 test with the optimized SVM.

step change points where current jumps from one steady state value to another. The US06 test is an especially demanding test of the SVM SOC estimator because it is trained with steady state data but is being asked to predict very dynamic outputs. Final testing data are scaled using the same formulas that are used on the training data. When the optimal SVM is run with US06 test data as input, the results are surprisingly good, as shown in Fig. 7.

In this test run, the root-mean-squared error is 5.76% and the maximum positive error is +12% and the maximum negative error is −2%. On the whole, this is impressive performance for an SVM trained on slowly changing data and being tested on very dynamic data.

Another US06 test is also used to test the optimized SVM. This test is even more dynamic than the previous test. In this test, SOC changes by 25% in just 7.5 min. For comparison, the first US06 test changes SOC by 20% over 63 min. The results of this test are shown in Fig. 8. In this test, the root-mean-squared error is just 2.5% with a maximum error of +13%.

4. Microcontroller implementation

One of the objectives of this research is to implement the optimized SVM in an inexpensive 8-bit microcontroller commonly used in many embedded systems. This is done on a Microchip PIC18F8720 running at 20 MHz with an instruction cycle time of 200 ns. The PIC18F8720 uses a modified Harvard architecture and has a two-stage pipeline with a built-in 8-bit × 8-bit single-cycle multiplier. It has 4K bytes of data RAM and 128K Bytes of FLASH program memory. The entire optimized SVM takes 8K bytes of program memory, including library routines added to the code to facilitate floating point math functions. After reading new data for pack current and voltage, the SVM computes a predicted SOC in 54 ms.

The same code requires 26 ms execution time for each SOC estimation when run on an Infineon C16716-bit microcontroller at 20 MHz. The program takes up 8K bytes of program memory. The EKF SOC program does not fit on either of the above inexpensive microcontrollers. A simplified version of the EKF SOC estimator takes up 12K bytes of memory in the C167 and executes in 345 ms. A full version of the EKF estimator will require a 32-bit microprocessor and larger memory space.

5. Conclusions and future work

In this research, we have demonstrated that the optimized SVM is attractive because it offers the accuracy of the EKF SOC estimator at the price of a coulomb counter. The SVM tested above is optimized with less than 4000 iterations through possible ranges of parameter values. An improved SOC estimator by SVM can be found by doing more iteration during training.

Temperature and charging history influence SOC, but the history can be incorporated in the training data vector. In the case of temperature, it can be included as an element of the input vector. Charging history is important only when the estimator is first starting up. A record of how long ago the battery pack has been shut down and whether it was charging or discharging helps estimate the starting SOC. The charging history can be embedded into the training data used by the SVM SOC estimator during the training process. Using the SVM approach, a low cost 8-bit microcontroller is sufficient to estimate battery SOC with high accuracy in real time.

References

[1] S. Piller, M. Perrin, A. Jossen, Methods for state-of-charge determination and their application, J. Power Sources 96 (2001) 113–120.
[2] V.H. Johnson, A.A. Pesaran, T. Sack, Temperature-dependent battery models for high-power lithium-ion batteries, in: Presented at the 17th Annual Electric Vehicle Symposium Montreal, Canada, October 15–18, 2000, The paper is downloadable at website http://www.nrel.gov/docs/fy01osti/28716.pdf.
[3] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[4] G.L. Plett, Advances in EKF SOC estimation for LiPB HEV battery packs, in: Proceedings of the EVS 20 Symposium, Long Beach, 2003.
[5] G.L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 1. Background, J. Power Sources 134 (No. 2) (2004) 252–261; Part 2. Modeling and identification, J. Power Sources 134 (No. 2) (2004) 262–276; Part 3. State and parameter estimation, J. Power Sources 134 (No. 2) (2004) 277–292.
[6] S. Pang, J. Farrell, J. Du, M. Barth, Battery state-of-charge estimation, in: Proceedings of the American Control Conference, Arlington, 2001.
[7] O. Barbarisi, L. Glielmo, F. Vasca, Automotive NiMH battery SOC estimation via an extended Kalman filter, in: Proceedings of the IFAC Symposium, Salerno, 2004.
[8] C.H. Cai, D. Du, Z.Y. Liu, Battery state-of-charge (SOC) estimation using adaptive neuro-fuzzy inference system (ANFIS), in: Poster Presentation at the IEEE International Conference on Fuzzy Systems, St. Louis, 2003.
[9] V. Kecman, Learning and Soft Computing, MIT Press, Cambridge, 2001.
[10] S. Haykin, Neural Networks, 2nd ed., Prentice-Hall, New Jersey, 1999.
[11] T. Joachims, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Making Large-Scale SVM Learning Practical. Advances in Kernel Methods—Support Vector Learning, MIT Press, 1999.