Journal of Power Sources 196 (2011) 2913–2923

Contents lists available at ScienceDirect

Journal of Power Sources

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

Linear parameter varying battery model identification using subspace methods

Y. Hu, S. Yurkovich[*]

Center for Automotive Research, The Ohio State University, 930 Kineear Rd, Columbus, OH 43212, United States

ARTICLE INFO

Article history:
Received 31 August 2010
Received in revised form 22 October 2010
Accepted 26 October 2010
Available online 9 November 2010

Keywords:
Subspace identification
Linear parameter varying systems
Battery modeling
System identification
P/HEV
State space

ABSTRACT

The advent of hybrid and plug-in hybrid electric vehicles has created a demand for more precise battery pack management systems (BMS). Among methods used to design various components of a BMS, such as state-of-charge (SoC) estimators, model based approaches offer a good balance between accuracy, calibration effort and implementability. Because models used for these approaches are typically low in order and complexity, the traditional approach is to identify linear (or slightly nonlinear) models that are scheduled based on operating conditions. These models, formally known as linear parameter varying (LPV) models, tend to be difficult to identify because they contain a large amount of coefficients that require calibration. Consequently, the model identification process can be very laborious and time-intensive. This paper describes a comprehensive identification algorithm that uses linear-algebra-based subspace methods to identify a parameter varying state variable model that can describe the input-to-output dynamics of a battery under various operating conditions. Compared with previous methods, this approach is much faster and provides the user with information on the order of the system without placing an *a priori* structure on the system matrices. The entire process and various nuances are demonstrated using data collected from a lithium ion battery, and the focus is on applications for energy storage in automotive applications.

1. Introduction

The recent push for better fuel economy has prompted many automakers to develop hybrid powertrains as a means of improving powertrain efficiency. The battery pack, a critical technology for such advanced powertrains, facilitates the storage of allows electrical energy so that the overall vehicle efficiency is improved. In order to function correctly, the battery pack must have a well designed battery management system (BMS). Among other things, a BMS must be able to estimate various conditions of the battery in real time so that the energy management system (EMS) of the vehicle can best decide on the optimal operating strategy [1]. In particular, this includes estimating the state of charge (SoC), state of health (SoH) and state of life (SoL) of the pack. A growing number of results have appeared in the literature on various methodologies for designing these estimators [2–9]. Within these results, the various model-based approaches (such as [6,9]) have shown good balance between accuracy, calibration effort, and implementability. For these methods to be applicable, however, a control-oriented model of the battery cell must be available; that is, such a model typically possesses good accuracy and is composed of lower order ordinary differential equations (ODEs) or difference equations, preferably linear. A side benefit of having a model of this type is that it can be easily implemented in a vehicle simulator and can be used in the design of vehicle architecture and operating strategies. In addition, it can be used to study cell interactions inside a pack [10] or evaluate and design other BMS functions such as charge balancing [11].

Generating a control-oriented model that can describe the input-to-output dynamics of a battery is a challenging problem. A primary reason for this is that battery dynamics vary significantly with operating conditions, and are typically nonlinear in nature. This comes from the fact that the macroscopic behavior of the battery is actually the result of a series of complex electrochemical processes, such as charge transfer and diffusion, that are affected by operating conditions [12,13]. The most important operating conditions are temperature, SoC, and current demand. For example, temperature can affect the rate of electrochemical reactions; SoC determines the available reaction components; and, current demand (in particular, the direction) determines the types of reactions. When these operating conditions are held approximately constant, by lumping the distributed spatial dynamics, the overall input-output dynamics can be approximated by electrical circuits composed of linear elements such as resistors and capacitors as well as possibly with nonlinear elements such as Warburg impedance. By further linearizing any remaining nonlinearities, a linear system of ODEs can be used to describe the localized

[*] Corresponding author. Tel.: +1 614 352 6605; fax: +1 614 688 4111.
E-mail addresses: yiran.hu@gmail.com (Y. Hu), yurkovich.1@osu.edu (S. Yurkovich).

0378-7753/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2010.10.072

dynamics. A common approach to extend the validity of this local model is to generate a family of local models corresponding to various portions of the operating space. Then this family can be interpolated to form a parameter varying model that describes the dynamics over a large portion of the operating space. In this sense, the simplicity in the model structure comes at the cost of having parameter-dependent model coefficients.

There are two traditional approaches to identifying a parameter varying battery model composed of ODEs. The first approach starts by obtaining localized operating data for the battery over a range in the operating space of interest. Then linear ODEs that are often derived from an equivalent circuit are identified using localized data. The model parameters are then interpolated with respect to the operating condition to generate the parameter varying model described previously. The drawback of this approach is that it is rather difficult to obtain localized data, because for a dataset to be attributed to one type of operating condition, the battery must only be charging or discharging during the entire dataset. Such a process limits the length of the dataset, because a lengthy dataset results in significant changes in the SoC.

The second approach is to obtain a scheduled model directly from a comprehensive dataset that spans across various operating conditions. Such datasets are much easier to engineer and are more practical to collect. As reported in previous work [14,15,19], the identification can be done using an optimization based procedure. In this process, several datasets are taken, where inside each dataset the battery operates at a constant temperature. The temperatures for the overall process are chosen so that all appropriate temperatures of interest are represented, and the dataset appropriately populated. Each isothermal dataset contains asymmetrical steps that allow the SoC of the battery to travel through the range of interest for the SoC while exciting battery dynamics (both charging and discharging) throughout the entire SoC range. To model the battery, a Randle equivalent circuit is used that contains an open circuit voltage, an internal resistance, and parallel RC circuits to approximate the dynamics. Then the model identification is done in a layered fashion, beginning with identification of a constant parameter model using each isothermal dataset, including the open circuit voltage as a function of the SoC and other circuit elements. Following this, using the constant model as an initial guess, the model is re-identified with the assumption that the circuit elements are functions of the SoC and current direction. When this is done for every isothermal dataset, the circuit elements are then interpolated with respect to temperature so that they become functions of temperature, SoC and current direction. Compared with the aforementioned interpolation approach, this approach is much more efficient in terms of human interaction and effort. Nevertheless, it still suffers in terms of computational requirements. The step of generating isothermal parameter varying models consists of a relatively large optimization problem where each model could contain tens, if not hundreds of unknown coefficients. Even on a reasonable-sized computer cluster, each identification can take several hours. Because it is often the case during development that the model structure is changed, or conditions of the identification are altered, requiring repetitious applications of the optimization process, any reduction in time required to generate these isothermal models represents a large improvement.

Aside from the open circuit voltage (OCV), the remainder of the model is in the form of a linear parameter varying (LPV) state variable system. For some time now, subspace identification methods have been used effectively for generating multi-input, multi-output discrete linear time invariant (LTI) state variable models using only the input and output data [16]. In recent years, a version of the subspace method that applies to certain forms of LPV systems has also appeared [17,18]. Subspace methods have several distinct advantages over optimization based methods. First and foremost, they are much faster than optimization based routines. The model coefficients are *computed* using linear algebra tools applied to input and output data; relatively speaking, therefore, the process is instantaneous. Second, this class of methods can be used to identify a state variable representation in a completely general form. In the context of battery modeling, this means the user would not have to assume *a priori* an equivalent circuit representation. Third, the concept of a dataset initial condition is irrelevant. In optimization based routines that require the simulation of the model over the dataset, unknown non-zero initial conditions can cause inaccurate identifications. Finally, the linear algebra analysis of the input and output data also provides the user with the approximate order of the system. Experiments may still be required to ascertain an appropriate model order that works to trade off simplicity for accuracy, but having a guideline that is based on the data can make this process very intuitive. Given these advantages, when the subspace method is applied to the battery modeling problem effectively, the overall modeling process is improved significantly. However, such an application requires an innovative restructuring of the battery identification problem, with additional innovation to existing subspace identification methodologies for LPV systems.

In this paper, a comprehensive LPV battery model identification methodology is described that uses a subspace method as the primary identification tool. The subspace method used is an extension of the LTI subspace method given in [16] using techniques and formulations from [17,18]. The resulting method rivals the optimization method from previous work [14,15,19] in terms of accuracy, but is faster and more user-friendly. The overall method is illustrated using data from a lithium ion battery.

2. Battery model

The basic model structure has current as input and the terminal voltage as output. A major component of the battery terminal voltage is the OCV, which is a static function of the SoC. Because the SoC is essentially a scaled integral of the current, the OCV must have marginally stable dynamics. Subspace based methods are not effective at identifying such equations because small numerical inaccuracies can significantly influence the fit. Therefore the output voltage is the difference between the measured terminal voltage and the OCV, which is commonly known as the over-voltage of the battery. This essentially implies that prior to applying any model identification algorithm, the OCV as a function of the SoC and temperature must be obtained. Nonetheless, since OCV is a variable that is generally measured during battery characterization, computing the over-voltage is generally not problematic.

The dynamics of a battery are most heavily influenced by three operating conditions: current direction (i_d), temperature (T), and state of charge (z). Ignoring effects of uncertainty or measurement noise for simplicity, the most generic model that can be formed is

$$\begin{aligned} x[k+1] &= A(T,z,i_d)x[k] + B(T,z,i_d)u[k] \\ y[k] &= C(T,z,i_d)x[k] + D(T,z,i_d)u[k], \end{aligned}$$

where y is the over-voltage, u is input current, and the matrices A, B, C, D are of appropriate dimension. A simplification that can be made immediately is to assume that C is independent of the parameters. This is because as long as the system is observable, a parameter dependent similarity transformation with the parameter dependent C matrix as a row can remove the parametric dependence on C completely. Observability can be assumed because the model is constructed with the purpose of representing the input to output dynamics of a battery. Therefore the model becomes

$$\begin{aligned} x[k+1] &= A(T,z,i_d)x[k] + B(T,z,i_d)u[k] \\ y[k] &= Cx[k] + D(T,z,i_d)u[k]. \end{aligned}$$

This is essentially in the same format that appeared in [17], with the minor exception that in [17] the matrix D is assumed to be independent of the parameters; a modification to this idea allows D to be parameter dependent.

The algorithm proposed in [17] is not applicable, for two main reasons. First, in order to model the dependence of system matrices on the parameters, nonlinear functions must be used. Writing these functions into the form needed by [17] requires rewriting the systems parameters, namely T, i_d, and z into a much larger vector of parameters. As a result, the data matrices that must be analyzed for the subspace routine are too large for reasonable computations, even on a supercomputer. A matrix reduction algorithm can be used to reduce the size of the problem at the cost of reduced identification accuracy; however, that algorithm is itself very time consuming, which in the end negates the speed advantage of the subspace identification method. Second, the algorithm requires that the system be exponentially stable with small enough time constant so that the effect of any initial conditions disappears within a number of steps (selected by the user). If this condition is not met, there will most likely be errors in the identification. If the user selects a very large number for the number of steps to ensure that effects of initial conditions disappear, the matrix dimension problem is exacerbated. Previous modeling experience has shown that even when the battery dynamics are exponentially stable, the time constants tend to be very long compared to the sampling period, thereby requiring a large number of steps. Therefore, this approach is not suitable for the battery model identification problem. In [18], a different method is used to analyze the matrices generated by the algorithm so that a much smaller matrix (square and the size of the dataset) is analyzed. Nevertheless, the initial condition problem persists. Furthermore, depending on the dataset size, matrices analyzed are still relatively large and because this analysis is in itself an approximation to the previous analysis, additional inaccuracies are introduced into the identification. Given these concerns, the LPV subspace algorithm as it appears in [18] is not adopted for this problem.

The approach that we take is to break the problem down further so that the subspace method for linear time invariant systems can be applied. First, battery modeling datasets are typically collected in isothermal conditions to ensure that the measured temperature is reliable (because measuring the internal temperature of a battery is not practical). Therefore the temperature dependence of the system matrices can be removed if isothermal models are identified. A generic isothermal model can be written as

$$\begin{aligned} x[k+1] &= A(z, i_d)x[k] + B(z, i_d)u[k] \\ y[k] &= Cx[k] + D(z, i_d)u[k]. \end{aligned}$$

To solve the matrix size problem, the dependence of the A matrix on z and i_d is removed. This results in a model form given as

$$\begin{aligned} x[k+1] &= Ax[k] + B(z, i_d)u[k] \\ y[k] &= Cx[k] + D(z, i_d)u[k]. \end{aligned} \tag{1}$$

As will be seen in the next section, because the parameters only modify the input in (1), an LTI subspace method can be used to perform the identification. One question that might arise is whether this process simplifies the dynamics too much to be useful. To answer this, first note that whether A depends on the parameters or not, is independent of the D matrix structure. Secondly, when the isothermal models are interpolated, the temperature dependence for A is captured. Thirdly, an iterative subspace method can be used to recover some of these parameter dependencies in certain cases. If need be, an optimization based procedure (such as in [19]) can be used to iteratively optimize the model to add the parametric dependence to A using (1) as a starting point. With a good starting point, the optimization procedure should be able to quickly improve the fit. As the results give later demonstrate, it is often that case that models without parametric dependence on the A matrix provide performance as good as that for models with this dependence. Therefore isothermal models of the form (1) are often sufficient.

3. Subspace identification algorithm

In this section, a subspace identification algorithm that can be used to identify (1) is discussed. To begin, write the model into the following form:

$$\begin{aligned} x[k+1] &= Ax[k] + B\begin{bmatrix} u[k] \\ p[k] \otimes u[k] \end{bmatrix} \\ y[k] &= Cx[k] + D\begin{bmatrix} u[k] \\ p[k] \otimes u[k] \end{bmatrix} \end{aligned} \tag{2}$$

where $p \in \Re^s$ is a parameter vector and $\otimes$ represents the Kronecker product. As an illustrative reference, if A and B are matrices defined as

$$A = \{a_{ij}\} \in \Re^{m \times n}, \quad B = \{b_{ij}\} \in \Re^{l \times k},$$

then A Kronecker product B is given by

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}.$$

Note that in writing the system this way, it is assumed that the parameters affect the parameter dependent coefficients in an affine or linear way. To see this, partition B as

$$B = \begin{bmatrix} B_0 & B_1 & \cdots & B_s \end{bmatrix},$$

where $B_i \in \Re^{n \times m}$ for $i = 0, 1, \ldots, s$. Then

$$\begin{aligned} B\begin{bmatrix} u[k] \\ p[k] \otimes u[k] \end{bmatrix} &= \begin{bmatrix} B_0 & B_1 & \cdots & B_s \end{bmatrix}\begin{bmatrix} u[k] \\ p[k] \otimes u[k] \end{bmatrix} \\ &= \left(B_0 + \sum_{i=1}^{s} B_i p_i[k] \right) u[k]. \end{aligned}$$

The term $B_0 + \sum_{i=1}^{s} B_i p_i[k]$ can be thought of as a parameter dependent B matrix. Clearly the dependence on p is affine (or linear if one augments p with 1). Consequently, (2) can represent *any* system where the dependence of B and D on the parameters is affine or linear. It is important to note that the parameters in the vector p may not be the same as the fundamental parameters on which the system depends. This aspect will be illustrated more clearly when the identification algorithm is applied to the battery model.

Now let m[k] be defined as

$$\mu[k] = \begin{bmatrix} u[k] \\ p[k] \otimes u[k] \end{bmatrix}.$$

In this way, μ can be thought of as the actual input to the system. Then (2) can be written as

$$\begin{aligned} x[k+1] &= Ax[k] + B\mu[k] \\ y[k] &= Cx[k] + D\mu[k]. \end{aligned} \tag{3}$$

Eq. (3) is an LTI system with μ as the input. Therefore the theoretical development from [16] can be applied directly to the identification problem. In fact, even though the system written here has no stochastic input representing uncertainty and noise, such quantities can be added easily without changing the solve-ability of the overall problem. The detailed identification algorithm and derivation can be found in [16] and is therefore not repeated here. The overall identification process can be summarized as follows:

1. Select the maximum order N of the system;
2. Form the data matrices (also known as Hankel matrices) using the input (μ) and output measurements;
3. Use linear algebra tools such as QR and SVD to analyze matrices generated from the data matrices to select the order of the system and to compute the state sequences and the system matrices (A, B, C, D).

One important issue that must be addressed is identifiability. In other words, under what conditions can algorithms such as [16] be applied with good results? The standard conditions under which the system is identifiable are:

1. If noise processes are present, then the input should be uncorrelated with the process noise and measurement noise.
2. The sequence $\mu[k]$ is persistently exciting of order $2N$, where N is the maximum order of the system selected by the user.
3. The input is not a function of the past states and output.

Among these conditions, the first is generically true for most systems. The third condition simply says that the data should be collected under open loop conditions, a process controlled by the user. Thus, the second condition is the only nontrivial condition. Define

$$U_{0|2N-1} = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{j-1} \\ \mu_1 & \mu_2 & \cdots & \mu_j \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N-1} & \mu_N & \cdots & \mu_{N+j-2} \\ \mu_N & \mu_{N+1} & \cdots & \mu_{N+j-1} \\ \mu_{N+1} & \mu_{N+2} & \cdots & \mu_{N+j} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{2N-1} & \mu_{2N} & \cdots & \mu_{2N+j-2} \end{bmatrix}$$

where j is a large integer that allows as much of the dataset to be used as possible. U is called the Hankel matrix of μ from 0 to $2N-1$. The input sequence μ is persistently exciting of order $2N$ if the covariance matrix between $U_{0|2N-1}$ and $U_{0|2N-1}^T$ is full rank. This straightforward condition is complicated by the fact that μ contains both u and p. For example, if $p_r[k]=p_q[k]$ for all k, then regardless of what the sequence $u[k]$ is, the covariance matrix will be rank deficient. Therefore, to be able to apply the identification algorithm, one must ensure that the persistence of excitation condition is met. Because the input is designed in open loop, this condition can be checked prior to the experiment to ensure that the dataset will be useful.

3.1. Iterative application

A recent study in [20] describes a Picard type iterative procedure that can be used to find the dependence of the A and C matrices on the parameter vectors. In particular, it is argued that for an LPV discrete state variable system with affine parameter dependence in all state matrices, if the parameters are excited using a white noise signal, then the parameter dependence of the A and C matrices can be obtained by repeated applications of the subspace method described in the previous section.

Consider the LPV system with affine parametric dependence for all system matrices, given by

$$\begin{aligned} x[k+1] &= A\begin{bmatrix} x[k] \\ p[k]\otimes x[k] \end{bmatrix} + B\begin{bmatrix} u[k] \\ p[k]\otimes u[k] \end{bmatrix} \\ y[k] &= C\begin{bmatrix} x[k] \\ p[k]\otimes x[k] \end{bmatrix} + D\begin{bmatrix} u[k] \\ p[k]\otimes u[k] \end{bmatrix}, \end{aligned} \tag{4}$$

where $x\in\Re^n$, $u\in\Re^m$, $y\in\Re^l$, $p\in\Re^s$ are the state, input, output and parameters, respectively, and A, B, C, D are the system matrices of appropriate dimensions. Partition the matrices A and C as

$$A = \begin{bmatrix} A_0 & A_1 & \cdots & A_s \end{bmatrix}, \tag{5}$$

$$C = \begin{bmatrix} C_0 & C_1 & \cdots & C_s \end{bmatrix}. \tag{6}$$

Rewrite (4) as

$$\begin{aligned} x[k+1] &= A_0x[k] + \begin{bmatrix} B & A_1 & \cdots & A_s \end{bmatrix}\begin{bmatrix} u[k] \\ p[k]\otimes u[k] \\ p[k]\otimes x[k] \end{bmatrix} \\ y[k] &= C_0x[k] + \begin{bmatrix} D & C_1 & \cdots & C_s \end{bmatrix}\begin{bmatrix} u[k] \\ p[k]\otimes u[k] \\ p[k]\otimes x[k] \end{bmatrix}. \end{aligned} \tag{7}$$

Writing the system in this manner shows that $p[k]\otimes x[k]$ can be considered as an additional input into the system if an estimated state sequence is available. Given this interpretation, the iterative subspace algorithm can now be described.

In the first step of the iterative algorithm, assume $p[k]\otimes x[k]=0$. Then (7) reduces to (2), so that the subspace method described previously can be used to find A_0, C_0, B, D, and the corresponding state estimates. In the next step of the iteration, the following system is identified

$$\begin{aligned} x[k+1] &= A_0x[k] + \begin{bmatrix} B & A_1 & \cdots & A_s \end{bmatrix}\begin{bmatrix} u[k] \\ p[k]\otimes u[k] \\ p[k]\otimes \hat{x}[k] \end{bmatrix} \\ y[k] &= C_0x[k] + \begin{bmatrix} D & C_1 & \cdots & C_s \end{bmatrix}\begin{bmatrix} u[k] \\ p[k]\otimes u[k] \\ p[k]\otimes \hat{x}[k] \end{bmatrix} \end{aligned} \tag{8}$$

where the state estimates obtained using the previously identified system matrices are now used as inputs to the system. From this, the estimates for A_i and C_i for $i=1, \ldots, s$ can be obtained. Then this process can be continued until convergence is achieved for the estimated system matrices.

The idea of this algorithm is that rather than considering the influence of the parameters on the A and C matrices all in one step, causing exponential growth in the data matrices, the parameter dependence is refined iteratively. In each step, the entire problem is manageable from a computational viewpoint. As the examples in [20] show, convergence is achieved in only a few iterations. Since each iteration can essentially be computed instantaneously, few iterations will not significantly increase the overall computational time. The drawback of this approach is that for general parameter trajectories (i.e. not white noise), there is no proof that the process converges. For most practical systems, it may be impossible to enforce the requirement that parameter trajectories be characterized as white noise due to physical constraints on the equipment. While anecdotal evidence using simple non-physical example systems suggest that this process could converge for general parameter trajectories as well, this method should be used with caution. Results from using this method are presented in Section 5.1.

4. Battery model identification algorithm

4.1. Model parameterization

In view of the subspace identification algorithm outlined in the previous section, the next innovation is to convert the battery model to a form which can be used for application. In other words, $B(z, i_d)u[k]$ and $D(z, i_d)u[k]$ must be expressed in a suitable matrix form. First, to express current direction, a binary representation can be used. Without loss of generality, consider the term $B(z, i_d)u[k]$, and write

$$B(z, i_d) = \begin{bmatrix} B_c(z) & B_d(z) \end{bmatrix}.$$

In other words, separate the charge and discharge components of B, so that each component depends only on the SoC. To accommodate the increased size of B, create the augmented input vector $\bar{u}$ as

$$\bar{u}[k] = \left(\begin{array}{ll} \begin{bmatrix} u[k] \\ 0 \end{bmatrix} & \text{if charge} \\ \begin{bmatrix} 0 \\ u[k] \end{bmatrix} & \text{if discharge} \end{array}\right.$$

With this notation, $B(z, i_d)u[k]$ can be replaced with the equivalent term $[B_c(z), B_d(z)]\bar{u}[k]$.

Because B_c and B_d are functions of SoC, a suitable functional form is needed. There are many choices for this functional form so that the result can be expressed in the matrix form desired. Whatever form is chosen, it should be flexible enough to place no *a priori* constraint on the function. For this reason, linear spline functions are used here. Because the domain for z is an interval (namely 0–100%), the linear spline function is well suited. To define a linear spline function, first define a partition for the range of SoC relevant to the problem (a subset of the interval from 0 to 100%). Call this partition $[0 \leq \rho_1, \rho_2, \ldots, \rho_s \leq 100]$. Define

$$L_{\rho_i}(z) = \left(\begin{array}{ll} z - \rho_i & \text{if} z > \rho_i \\ 0 & \text{else} \end{array}\right. \quad i = 1, 2, \ldots, s.$$

L_{ρ_i} are basis functions for linear spline functions defined using the partition $[\rho_1, \rho_2, \ldots, \rho_s]$. Thus

$$B_c(z) = B_{c_0} + \sum_{i=1}^{s} B_{c_i} L_{\rho_i}(z).$$

Consequently,

$$B_c(z)u[k] = \begin{bmatrix} B_{c_0} & B_{c_1} & \cdots & B_{c_s} \end{bmatrix} \begin{bmatrix} u[k] \\ \begin{bmatrix} L_{\rho_1}(z) \\ L_{\rho_2}(z) \\ \cdots \\ L_{\rho_s}(z) \end{bmatrix} \otimes u[k] \end{bmatrix}.$$

Similarly,

$$B_d(z)u[k] = \begin{bmatrix} B_{d_0} & B_{d_1} & \cdots & B_{d_s} \end{bmatrix} \begin{bmatrix} u[k] \\ \begin{bmatrix} L_{\rho_1}(z) \\ L_{\rho_2}(z) \\ \cdots \\ L_{\rho_s}(z) \end{bmatrix} \otimes u[k] \end{bmatrix}$$

Define the parameter vector $p \in \Re^s$ to be

$$p[k] = \begin{bmatrix} L_{\rho_1}(z[k]) \\ L_{\rho_2}(z[k]) \\ \cdots \\ L_{\rho_s}(z[k]) \end{bmatrix}.$$

Then define $\bar{B}_c$ and $\bar{B}_d$ as

$$\begin{aligned} \bar{B}_c &= \begin{bmatrix} B_{d_0} & B_{d_1} & \cdots & B_{d_s} \end{bmatrix} \\ \bar{B}_d &= \begin{bmatrix} B_{d_0} & B_{d_1} & \cdots & B_{d_s} \end{bmatrix} \end{aligned}$$

Consequently, the fully parameterized term $B(z, i_d)u[k]$ can be written as

$$B(z, i_d)u[k] = \left(\begin{array}{ll} \begin{bmatrix} \bar{B}_c & \bar{B}_d \end{bmatrix} \begin{bmatrix} \begin{bmatrix} u[k] \\ p[k] \otimes u[k] \end{bmatrix} \\ 0 \end{bmatrix} & \text{if charge} \\ \begin{bmatrix} \bar{B}_c & \bar{B}_d \end{bmatrix} \begin{bmatrix} 0 \\ \begin{bmatrix} u[k] \\ p[k] \otimes u[k] \end{bmatrix} \end{bmatrix} & \text{if charge} \end{array}\right. \tag{9}$$

This parameterization allows the dependence on SoC and current direction to be represented jointly and still retain the desired matrix form. Clearly the $D(z, i_d)u[k]$ term can be expressed in the exact same way. Using this representation, the identification algorithm outlined in the previous section can be applied.

The cost of this representation is that the size of the vector u has increased $2(s+1)$ times (recall, s is the size of the parameter vector). However, this increases the size of the entire problem only in a linear fashion. Therefore even problems with large datasets can still be handled quite easily on a standard desktop computer.

4.2. Standard model form

One important feature about subspace identification is that the system matrices identified may be in any basis. This poses a problem since the ultimate goal is to have the isothermal models work together to have a multi-temperature model. So in this section a standard model form is offered for the special case when the eigenvalues of A are positive, real and distinct. As it turns out, these three conditions are typically met for lower order model identifications in battery modeling problems. Therefore the form described here is not overly restrictive, and has practical value.

Suppose that the eigenvalues of A are real and distinct. Then there exists some similarity transformation that diagonalizes A. In other words,

$$A = V\bar{A}V^{-1}$$

where $\bar{A}$ is diagonal and whose diagonal is precisely the eigenvalues of A. Without loss of generality, assume that the diagonal values of A are arranged from the smallest to largest. Then

$$\begin{aligned} V^{-1}x[k+1] &= \bar{A}(V^{-1}x[k]) + V^{-1}Bu[k] \\ y[k] &= CV(V^{-1}x[k]) + Du[k] \end{aligned}.$$

Let $\bar{B} = V^{-1}B$ and $\bar{C} = CV$. Then the following system has the same input to output dynamics as the previous (although, note that x no longer refers to the same states):

$$\begin{aligned} x[k+1] &= \bar{A}x[k] + \bar{B}u[k] \\ y[k] &= \bar{C}x[k] + Du[k] \end{aligned}$$

Now apply another transformation $W = \text{diag}(-\bar{C})$. Note here that because $\bar{A}$ is diagonal and the system is observable, W is necessarily non-singular. Let $\tilde{A} = W\bar{A}W^{-1}$, $\tilde{B} = W\bar{B}$, and $\tilde{C} = \bar{C}W^{-1}$. Then the above system is equivalent in terms of input to output as:

$$\begin{aligned} x[k+1] &= \tilde{A}x[k] + \tilde{B}u[k] \\ y[k] &= \tilde{C}x[k] + Du[k] \end{aligned} \tag{10}$$

In this last form, the transformation W ensures that $\tilde{C} = [-1, \ldots, -1]$. Consequently, $y[k]$ is the negative sum of all the components of x plus $Du[k]$. In this form, $Du[k]$ can be thought of as the voltage drop due to an internal resistance; each component of x can be thought of as voltage increase/drop due to some dynamics. As

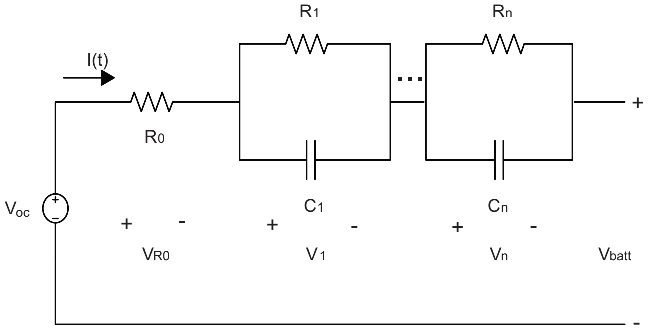

Fig. 1. Circuit model that is equivalent to the model identified.

such $y[k]$ is simply a sum of several voltage components. Furthermore, because each state of x is decoupled from the other states, each state has the effect of a first order lowpass filter on the current. Since the eigenvalues of the matrix A are positive, this model is equivalent to a discretized equivalent circuit model with an internal resistance and parallel RC circuits (as shown in Fig. 1). Note that if the eigenvalue happens to be negative, then there would be no continuous time equivalent because there would be oscillations in the response. However, in all the modeling done on batteries performed by the authors, this has not been the case.

Given this interpretation of the model identified, a continuous equivalent circuit model can actually be constructed using the system matrices identified, which would give the model physical meaning. More importantly, applying this transformation allows all the models to share the same structure, thereby allowing for interpolation of the model coefficients based on temperature.

4.3. Model identification algorithm

Now a detailed identification procedure for identifying a parameter dependent state space battery model can be stated.

1. Measure the open circuit voltage. Because our subspace method is not suitable for modeling the open circuit voltage, the OCV as a function of the SoC must be available *a priori*. In general, this should be done for every temperature at which modeling data is collected because the OCV varies with both SoC and temperature, although the effect of temperature is smaller than that of SoC. Typically the OCV is measured by either discharging the battery at a very small rate (C/20 for example) or discharging/charging to a SoC and then resting for a long period of time to allow the dynamics to settle. In this case, a best practice measurement technique is to charge the battery to 100%, soak the battery at various temperatures, record the OCV, then discharge the battery by 10% SoC increments, re-soaking at these temperatures. This is continued until the SoC reaches its lowest sustainable value (such as 10%).
2. Collect modeling data at a set of preselected temperatures. For example, if we are interested in modeling the battery behavior from −5 °C to 45 °C, then we can collect data at 5 °C increments. Each isothermal dataset must meet the persistence of excitation criterion discussed earlier. In particular, this necessarily means that the entire span of SoC is traversed through both charge and discharge. An approach that works well for fitting lower order models is to have multiple charging and discharging steps inside every 10% SoC zone and gradually move through the entire range of SoC. Because the current profile is designed prior to the test, the open loop condition is automatically satisfied.
3. Apply the subspace identification algorithm on each isothermal dataset. This provides a family of linear parameter varying models (parameters being SoC and current direction) that are themselves parameterized by the temperature.
4. Transform each isothermal model into the standard form described in the previous section. This assumes that the eigenvalues of each A matrix are positive, real and distinct. If this is not the case, then a more complicated procedure must be used for this conversion. Note again that we have not experienced such cases to date.
5. Interpolate the standardized isothermal models with respect to the temperature to arrive at an LPV model whose parameters are temperature, SoC and current direction.
6. (*Optional but recommended*) Use an optimization based procedure to optimize the interpolated model over all the temperature datasets jointly. This step may not be necessary if all of the model coefficients interpolated previously show acceptable smoothness properties. In general however, because of uncertainty and the low-order nature of the model, numerical peculiarities are typically inherent in the models identified. Therefore a best practice is generally to perform such a global optimization. How to do this is outside the scope of this paper; interested readers should pursue [15] for details.

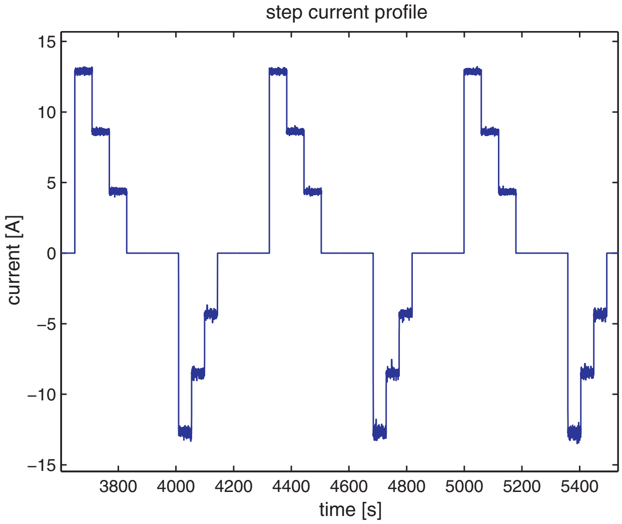

Fig. 2. Step current profile.

5. Modeling results

In this section, a complete modeling case study is presented for an A123 lithium ion iron-phosphate battery with nominal voltage of 3.2 V and nominal capacity of 2.3 AH. A total of seven datasets are used, collected at −5 °C, 0 °C, 5 °C, 15 °C, 25 °C, 35 °C and 45 °C (note that lower temperatures such as −15 °C are not used because in that condition the battery can only be operated in a limited range of SoC before experience under-voltage). For each temperature, a dataset consists of a series of asymmetrical steps used to excite the battery dynamics, designed to allow the battery to travel through as much of the 10–90% SoC region as possible while at the same time exciting dynamics of interest for a typical application. Figs. 2 and 4 show the asymmetrical step profile as well as the SoC trajectory achieved using this profile. Note here that the SoC is calculated via current integration after the current data is post-processed to remove noise and drifts. Figs. 3 and 5 show a pulse current profile and its SoC trajectory used to simulate battery response to higher magnitudes, but with shorter duration current pulses than those used in the step profile. Because a higher current rate is used, this profile is only executed for temperatures from 15 °C thru 45 °C. Furthermore, the SoC range is limited, from 50% to 90%, to avoid under or over-

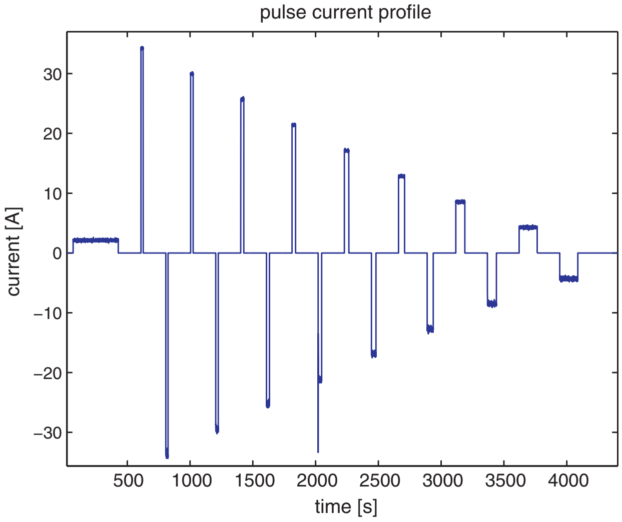

Fig. 3. Pulse current profile.

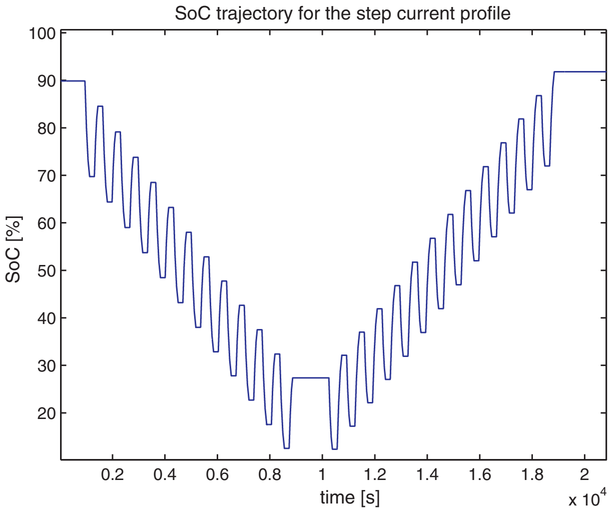

Fig. 4. SoC trajectory of the step profile.

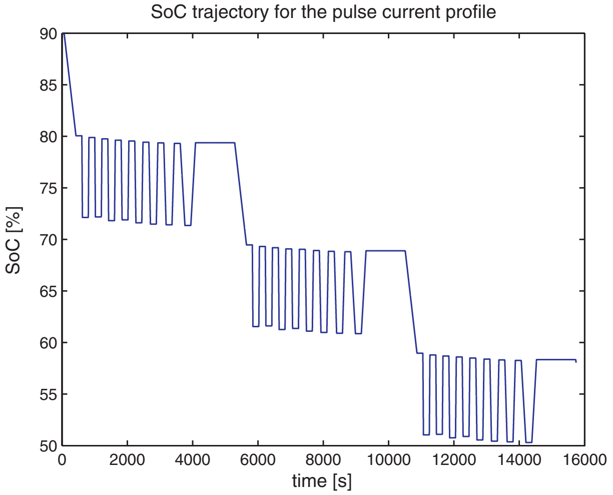

Fig. 5. SoC trajectory of the pulse profile.

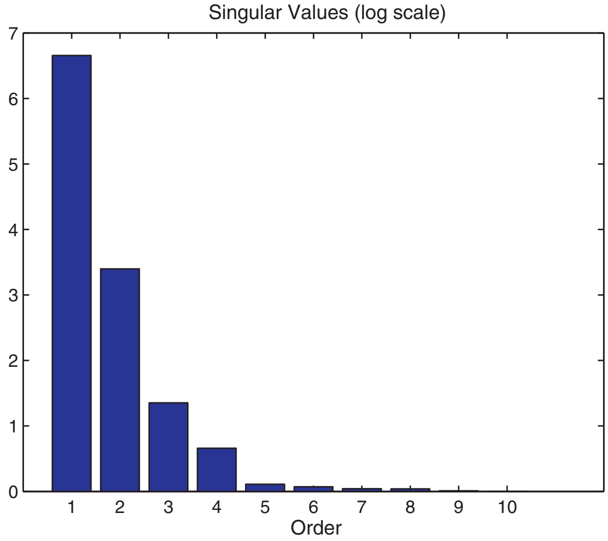

Fig. 6. Singular values analysis used to determine the model order.

voltage conditions on the battery. Here the pulse dataset is used for validation purposes.

To start, the OCV is measured at 25 °C. We make the assumption that this OCV is valid for all the other temperatures; while not completely valid, this assumption has little effect on the model fitting results because any errors manifest themselves as constant offset errors in some SoC regions. Since constant offsets *cannot* be modeled by a dynamic system with stable eigenvalues, they do not have a significant influence on the results of the fitting.

To illustrate the result of the identification, we first consider the 25 °C model. To find the order of the model, the maximum order of the model is selected to be 10, which is much larger than the order of the desired model. Subsequently, the subspace algorithm is used to generate the relevant data matrices whose nonzero singular values provide the order of the underlying system. As seen in Fig. 6, the relevant order of the system could be chosen in the range from 1st to 4th, because of the relative magnitudes of the first four singular values. In fact, we selected a second order because the fitting differences between 2nd, 3rd, and 4th order models were very small.

Once the order is selected, the model is identified according to methodology and algorithmic steps discussed above, and subsequently simulated in open loop using the dataset. The resulting RMS modeling error is only 7 mV for the step profile, as seen in Fig. 7, where a comparison is given between the model and the measurement. Clearly the model captures the dynamics exhibited in the data. As Fig. 8 shows, the performance over the pulse dataset is also very good (12 mV of RMS error).

To show the effect of parameter variation, the internal resistances for the 25 °C and 5 °C models are plotted in Figs. 9 and 10, respectively. In both cases, there is a difference between the charge and discharge internal resistances, confirming the need for parametric dependence on the current direction. For both temperatures, the internal resistance varies with respect to the SoC. In particular, as the SoC approaches high and low regions, the internal resistances increase; this phenomenon is consistent with the physical intuition. For 25 °C, the overall variation with regard to the SoC is small. But for 5 °C, this variation is significant, although occurring smoothly with respect to the SoC. This suggests that the model fit captures the physical characteristics of the battery. Furthermore, this also suggests that these quantities can be easily interpolated with respect to temperature.

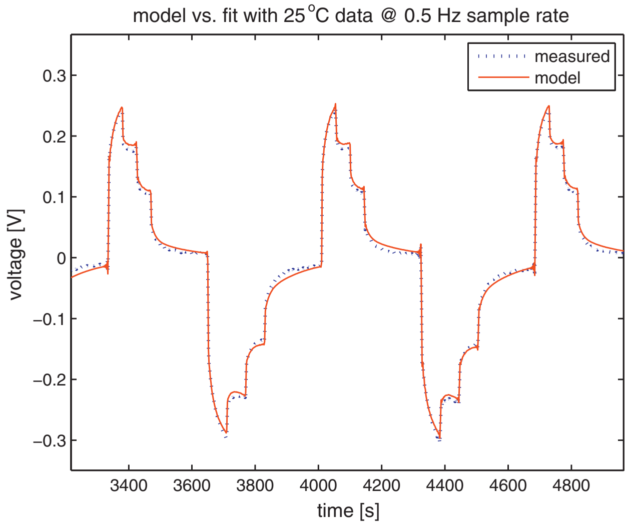

Fig. 7. Zoomed view of the fit for the step profile.

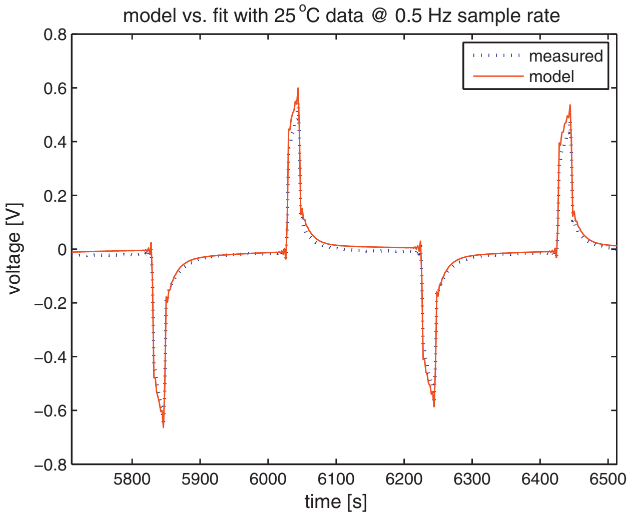

Fig. 8. Zoomed view of the fit for the pulse profile.

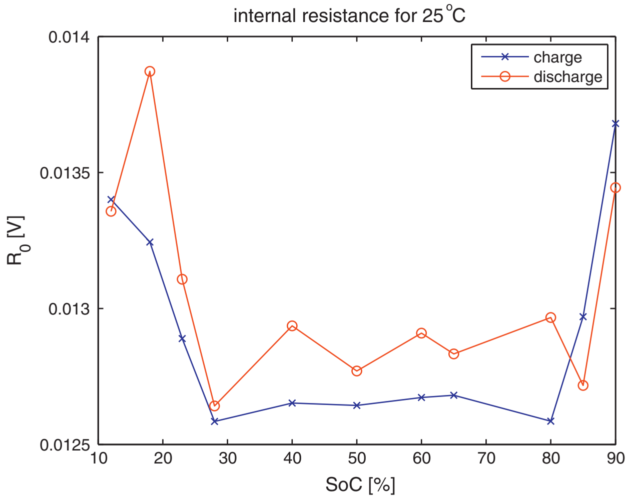

Fig. 9. Internal resistance of model found for 25 °C.

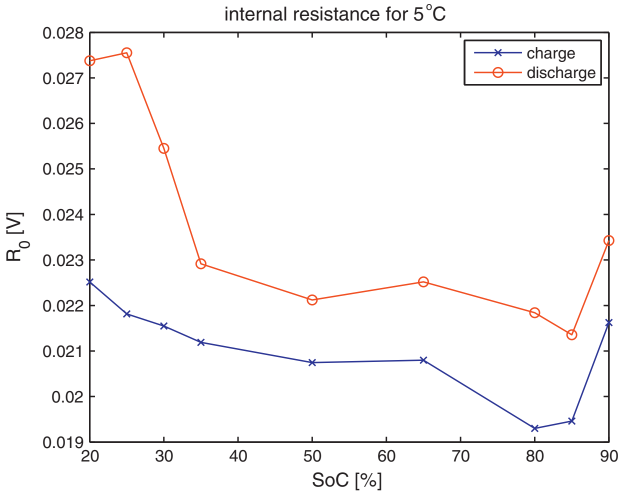

Fig. 10. Internal resistance of model found for 5 °C.

For other temperatures, the same model identification procedure is performed; Table 1 summarizes the fitting results. At higher temperatures, the RMS errors are reasonable, exhibited for both the step dataset used for modeling and the pulse dataset used for validation (wherever available). The fact that the model performs similarly for the validation dataset is a good indication of the validity of the identification. Because the battery dynamics become more complicated as the temperature lowers, at lower temperatures the fit degrades. But as seen in Fig. 11, the error is mostly an offset error that is the result of the inaccuracies in the OCV in the lower

Table 1
RMS error for datasets collected at different temperatures; M1 = subspace, M2 = optimization.

T	Step RMS [V]		Pulse RMS [V]	
	M1	M2	M1	M2
45 °C	0.0098	0.0800	0.0072	0.010
35 °C	0.0112	0.0820	0.0072	0.011
25 °C	0.0084	0.0101	0.0128	0.0150
15 °C	0.0117	0.0128	0.0140	0.0140
5 °C	0.0206	0.0194	N/A	N/A
0 °C	0.0340	0.0319	N/A	N/A
−5 °C	0.0518	0.0480	N/A	N/A

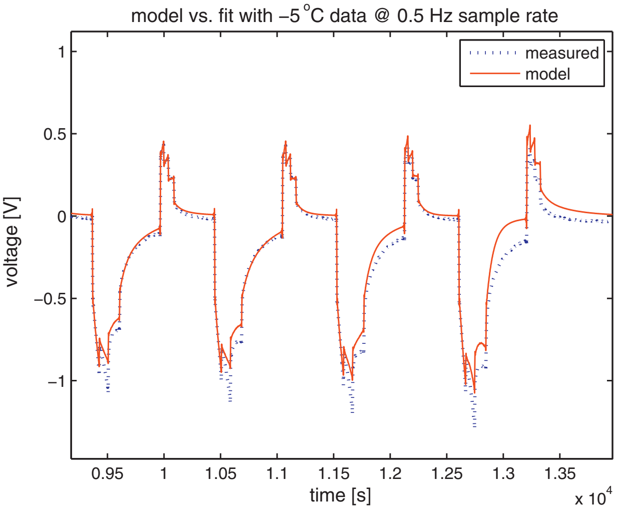

Fig. 11. Model fit at −5 °C.

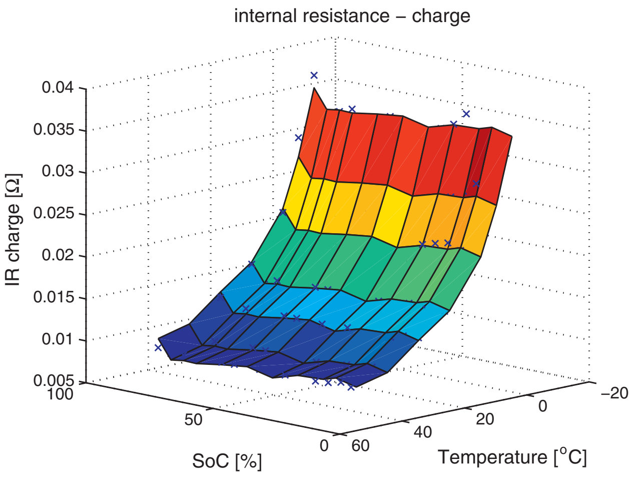

Fig. 12. Charging internal resistance for −5 °C to 45 °C.

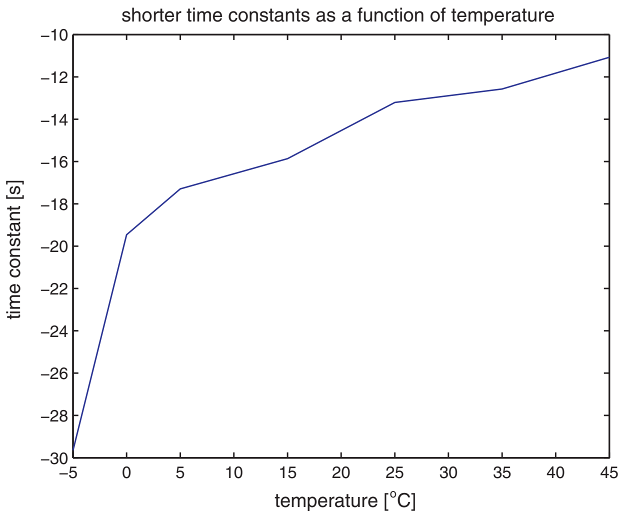

Fig. 14. Shorter time constant for −5 °C to 45 °C.

SoC regions; otherwise, dynamics are captured well. By using an OCV function that is specifically measured for −5 °C, the fit would improve significantly. Therefore, even at −5 °C, despite the higher error, the model predicts the battery behavior well and is therefore useful. Perhaps more importantly, the numbers here are similar to the RMS error that results when an optimization based routine is used to generate these models (see [15,19] for example). For illustration and comparison, a set of RMS errors using an optimization based routine is shown in Table 1. The numbers obtained between the two methods are comparable, further evidence that in this particular application, the scheduling of the A matrix with respect to the SoC and current direction is unnecessary.

To arrive at a multi-temperature model, the identified coefficients must be interpolated with respect to temperature, and all the models identified at constant temperatures must be converted to the standard form. Figs. 12 and 13 show the charge and discharge internal resistance as a function of the SoC and temperature after interpolation. Both plots show a very smooth inverse dependence on temperature (i.e. when temperature is lowered, the internal resistance is increased) which is consistent with the chemical properties of the battery. In addition, it is interesting to analyze the time constants (from the eigenvalues of the A matrix) as a function of the temperature. These time constants correspond to the dynamics of polarization and diffusion effects in the battery. To illustrate, Figs. 14 and 15 show the shorter and longer time constants in seconds (that is, a pole-zero mapping technique is used to transform the eigenvalues to continuous time). In both cases, the variation with respect to temperature is smooth, allowing for an easy interpolation as functions of temperature.

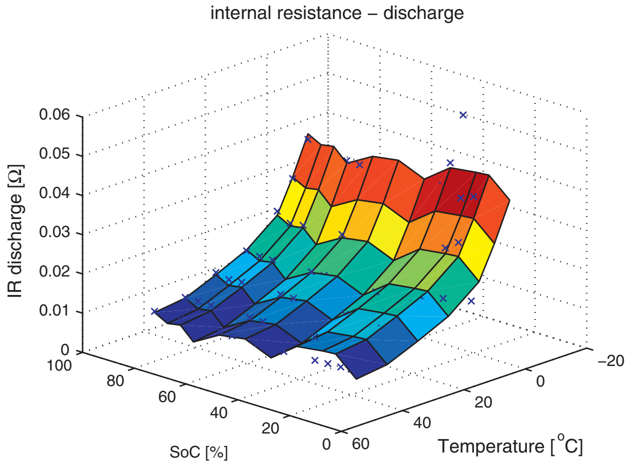

Fig. 13. Discharging internal resistance for −5 °C to 45 °C.

5.1. *Iterative identification results*

In Section 3.1, a method of iteratively computing an LPV discrete state space model that includes parameter scheduling on the A and C matrices is outlined. Because there is no guarantee of convergence, a feasibility test is conducted to test the viability of this method.

In the first portion of the test, the 25 °C step profile data is used to fit a model where B and D still depend on the current direction and SoC, but A and C now depend only on the current direction. To do this, first define an additional parameter variable p_{i_d} as

$$p_{i_d}[k] = \begin{cases} -1 & \text{charging} \\ 1 & \text{discharging} \end{cases}. \tag{11}$$

Then $A(p_{i_d}[k])$ and $C(p_{i_d}[k])$ are written as

$$\begin{aligned} A(p_{i_d}[k]) &= A_0 + p_{i_d}[k]A_1, \\ C(p_{i_d}[k]) &= C_0 + p_{i_d}[k]C_1. \end{aligned} \tag{12}$$

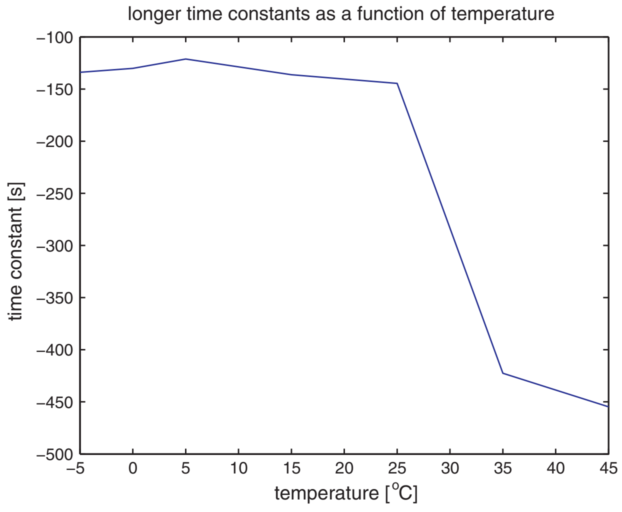

Fig. 15. Longer time constant for −5 °C to 45 °C.

In other words A and C take on the values $A_0 \pm A_1$ and $C_0 \pm C_1$ based on whether the battery is being charged or discharged. Using this setup, the approximate states computed using the model identified in the previous section are fed into the new identification process as input signals. In this case, the iterative algorithm converges; after transforming the resulting model into the standard form, the A matrix is given by

$$A = \begin{cases} \begin{bmatrix} 0.9892 & 0 \\ 0 & 0.8710 \end{bmatrix} & \text{charging} \\ \begin{bmatrix} 0.9892 & 0 \\ 0 & 0.8958 \end{bmatrix} & \text{discharging} \end{cases}. \tag{13}$$

The differences in the A matrix between charging and discharging are very minor, suggesting that the dependence on current direction could be neglected. Nevertheless, because the algorithm performs as desired, this step could be used in other battery modeling applications where the effect of the current direction may be larger.

Next, under the same conditions, the dependence of A and C on SoC is added. However, the model obtained in the second iteration is unstable. In particular, the parameter varying A matrix has eigenvalues that are outside of the unit circle for some values of the parameters (model simulation confirms this instability). Consequently, the inclusion of SoC as a scheduling variable cannot be handled by the iterative algorithm in this case.

To complete the study, several isothermal datasets are used at the same time to see if the temperature dependence could be obtained using the iterative method. Given the instability results found previously when SoC dependence is added, for the multiple temperatures case, only current direction and temperature dependence are imposed on the A matrix. In this example, the step profiles obtained at 0 °C, 5 °C, 15 °C and 25 °C are used to fit the model. A one-dimensional linear spline function is used to represent the dependence of the A and C matrices on temperature, while a two-dimensional linear spline function is used to represent the dependence of the B and D matrices on SoC and temperature. Then a setup similar to (12) is used to add the current direction dependence to A and C. For example, if the partition for the temperature is given as $[T_1, T_2, \ldots, T_s]$ and the linear spline basis functions for this partition are given as $U_{T_i}(\cdot)$ for $i = 1, 2, \ldots, s$, then A can be represented by

$$A(T[k], i_d[k]) \tag{14}$$

$$= A_0 + p_{i_d}[k]A_1 + p_c[k]\sum_{i=1}^{s-1} A_{i_c} U_{T_i}(T[k]) + p_d[k]\sum_{i=1}^{s-1} A_{i_d} U_{T_i}(T[k]), \tag{15}$$

where under charging conditions $p_c[k] = 1$ and $p_d[k] = 0$, and under discharging conditions $p_c[k] = 0$ and $p_d[k] = 1$. Without loss of generality, C can be represented in a similar manner.

The dependence of B and D on current direction can be described using the same setup described in Section 4.1. In this case, while the models obtained during the iterations do not have stability issues, the process does not converge. In particular, the models generated after the first iteration oscillate between two forms, neither of which are better than the model obtained after the first iteration. While this does not mean that the methodology is not feasible, it does mean that perhaps a different formulation is needed to ensure that this does not happen.

This feasibility study shows that the iterative subspace method could be used to identify constant temperature models where the A matrix depends on the current direction. If all constant temperature models are identified in this way, then the interpolation scheme described previously can be used to find a model where the A matrix is scheduled on temperature and current direction. Using this method for additional scheduling variables may be possible, but the formulation should be carefully examined to avoid the problems of instability, as well as for assuring model accuracy.

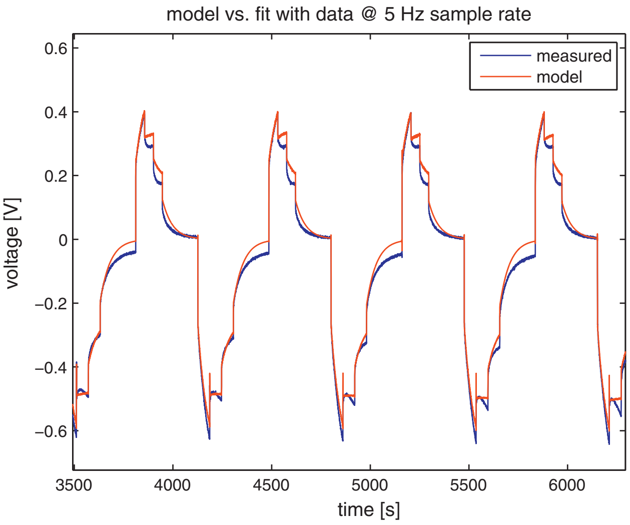

Fig. 16. Model fit using the 05 °C data at 5 Hz.

5.2. Data sampling

A subtle issue that appeared in this study is the dependence of the fit on the sampling time chosen for the dataset. Originally, the datasets were collected at a 10 Hz sampling rate, but using this dataset directly to compute a model produced poor results. The main reason for this is that the battery dynamics have a much slower time constant than the 10 Hz sampling rate. Consequently, the eigenvalues of the A matrix identified under this sampling rate are very near 1. As such, perturbations caused by noise and unmodeled dynamics can significantly influence the accuracy of the identification. For example, an eigenvalue of 0.99 gives rise to a dynamic behavior that is quite different from an eigenvalue of 0.995, even though numerically the difference is small. This suggests that a larger sampling time should be used. As a side benefit, using a larger sampling period also reduces the number of data points required.

To illustrate the effect of sampling time, two modeling exercises are performed for the same dataset, sampled at two different rates. The dataset used in this example is the asymmetrical step profile taken at 5 °C. First the identification is performed using the data at 5 Hz (10 Hz data was not used because of computer memory limitations). As Fig. 16 indicates, the algorithm and identification process adequately identified the D term, whereas the time constants are clearly not captured adequately. When the same identification is performed with data sampled at 0.5 Hz, as seen in Fig. 17, it is clear that the model performs much better. This suggests that using 0.5 Hz sampling rate is a better practice than using the 5 Hz rate. Note here that further down-sampling the dataset is not recommended, because with this dataset the two dominant time constants are approximately 15 s and 120 s. Therefore if the dataset is downsampled further, the dynamics related to the 15-s time constant would be filtered out, resulting in an inaccurate model. Consequently, this analysis suggests that for this application, a good sample rate is essential to the success of the identification. A good rule of thumb is that a sampling rate around 1 Hz is sufficient.

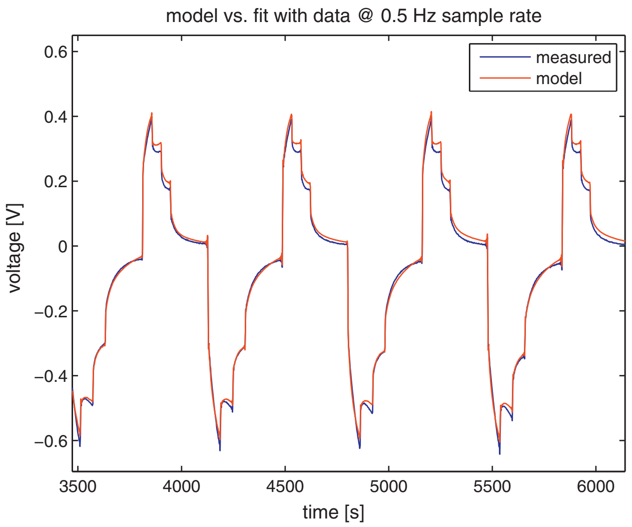

Fig. 17. Model fit using the 05 °C data at 0.5 Hz.

6. Conclusion

In this paper, a complete methodology for identifying a control oriented model that can describe the input to output dynamics of a battery is presented. This methodology differs from previous work in the literature in that a subspace method is used as the identification tool. Because the battery dynamics are inherently nonlinear and subspace methods only apply to linear systems, the first contribution of this work is to frame the identification problem in such a way that the subspace method is applicable. In particular, it is shown that if the open circuit voltage of the battery as a function of the SoC and temperature is removed from the battery terminal voltage, then the remaining voltage terms can be described adequately by an LPV model whose input is the current and parameters are the temperature and SoC. Then, a subspace methodology can be used to quickly and effectively compute the LPV model coefficients.

Compared with other identification procedures that have appeared in the open literature, the method discussed in this paper provides several important improvements. First, because the subspace method is an analytical tool, the time required to perform the identification is reduced drastically when compared with optimization based techniques. Secondly, because the identification process does not require simulation of the model, datasets that have nonzero initial conditions can also be handled. Thirdly, the matrix analysis used in the subspace method provides the user with an estimate of the system order, which reduces the need for selecting the order via trial and error. Lastly, a promising iterative subspace method is available that could further improve the identification accuracy, although preliminary results suggest that more work is needed to render this technique more useful. Nevertheless, even without using the iterative subspace method, the models identified show good accuracy and are also physically sound. Consequently, the process described in this paper provides a new paradigm for solving the problem of control-oriented modeling of battery cells.

References

[1] E. Meissner, G. Richter, Journal of Power Sources 144 (2005) 438–460.
[2] C. Cai, D. Du, Z. Liu, J. Ge, 7th World Congress on Intelligent Control and Automation, 2008.
[3] S. Malkhandi, Engineering Applications of Artificial Intelligence 19 (2006) 479–485.
[4] A. Salkind, S.C. Fennie, T.P. Atwater, D. Reisner, Journal of Power Sources 80 (1999) 293–300.
[5] M. Verbrugge, E. Tate, Journal of Power Sources 126 (2004) 236–249.
[6] G. Plett, Journal of Power Sources 134 (2004) 262–276.
[7] G. Plett, Journal of Power Sources 134 (2004) 277–292.
[8] I. Kim, Journal of Power Sources 163 (2006) 584–590.
[9] F. Zhang, G. Liu, L. Fang, Proceedings of the 7th World Congress on Intelligent Control and Automation, 2008.
[10] B.J. Yurkovich, Y. Guezennec, ASME Dynamic Systems and Control Conference, 2009.
[11] J. Welsh, A comparison of active and passive cell balancing techniques for series/parallel battery packs, M.S. thesis, Ohio State University, 2009.
[12] D. Linden, T.B. Reddy, Handbook of Batteries, third ed., McGraw Hill, 2001.
[13] E. Barsoukov, J.R. Macdonald, Impedance Spectroscopy: Theory, Experiment, and Applications, second ed., Wiley-Interscience, 2005.
[14] Y. Hu, S. Yurkovich, Y. Guezennec, B.J. Yurkovich, Control Engineering Practice 17 (2009) 1190–1201.
[15] Y. Hu, S. Yurkovich, Y. Guezennec, B.J. Yurkovich, ASME Dynamic Systems and Control Conference, 2009.
[16] P. van Overschee, M.B. De Moor, Subspace Identification for Linear Systems: Theory – Implementation – Applications, first ed., Springer, 1996.
[17] V. Verdult, M. Verhaegen, Automatica 38 (2002) 805–814.
[18] V. Verdult, M. Verhaegen, Automatica 41 (2005) 1557–1565.
[19] Y. Hu, S. Yurkovich, Y. Guezennec, B.J. Yurkovich, Journal of Power Sources 196 (2010) 449–457.
[20] P. Lopes dos Santos, J. Ramos, J. Martins de Carvalho, International Journal of Systems Science 39 (2008) 897–911.