A new predictive model for the state-of-charge of a high power lithium-ion cell based on a PSO optimized multivariate adaptive regression splines approach

J. C. Álvarez Antón, *Member, IEEE*, P. J. García Nieto, E. García Gonzalo, J. C. Viera Pérez, *Member, IEEE, M. González Vega, Member, IEEE* and C. Blanco Viejo.

Abstract—**Batteries play a key role in achieving the target of universal access to reliable affordable energy. Despite their relevant importance, many challenges remain unsolved as regards the characterization and management of batteries. One of the major issues in any battery application is the estimation of the state-of-charge (SoC). SoC, expressed as a percentage, indicates the amount of energy available in a battery. An accurate SoC estimation under realistic conditions improves battery performance, reliability and lifetime. This paper proposes a SoC estimation method based on a new hybrid model that combines multivariate adaptive regression splines (MARS) and particle swarm optimization (PSO). The proposed hybrid PSO-MARS-based model uses data obtained from a high power load profile (Dynamic Stress Test) specified by the United States Advanced Battery Consortium (USABC). The results provide comparable accuracy to other, more sophisticated techniques, but at a lower computational cost.**

Index Terms— **Lithium batteries; Nonlinear estimation; State-of-charge (SoC); Multivariate adaptive regression splines (MARS); Particle Swarm Optimization (PSO).**

I. Introduction

THE rechargeable battery industry is experiencing significant growth driven by an upsurge in portable battery-powered devices, electric vehicles and other industrial applications. A number of different battery chemistries, such as lead-acid, nickel-metal-hydride and lithium-ion, among others, are used in these applications. One of the most popular types of rechargeable battery technologies is the lithium-ion battery. Its chemistry provides a high cell voltage, high energy density, long lifespan and exceptional cyclability. All lithium-ion battery applications, especially those used in electric vehicles, require a Battery Management System (BMS). The main objective of a BMS is to maintain the health of all the cells in the battery within the manufacturer's recommended operating conditions in order to prolong the lifespan of the battery pack. One of the most important BMS functional requirements is that of estimating the State-of-Charge (SoC) of the battery or of the individual cells in the battery pack. SoC is an expression of the current battery capacity as a percentage of its maximum capacity. The BMS needs to estimate the SoC in order to report the capacity left in the battery, typically called the "gas gauge". Several applications require accurate measurement of SoC to give users an indication of available runtime. The SoC is also needed to control the battery charging or discharging process. This control can avoid situations such as over-discharging or over-charging, which lead to premature wear-out of the battery. The lithium-ion chemistry operates safely within the designed operating voltages; however, the battery becomes unstable and may pose a safety hazard if overcharged. Over-charging stresses the battery and may lead to damage. Over-discharging also stresses the battery and reduces its lifespan.

The available capacity of a battery depends on several factors such as cell chemistry, charging rates, discharging rates and temperature. All of these factors need to be considered when calculating the SoC. Many methods for estimating the SoC have been reported in the literature [1–20]. For instance, measuring cell voltage to calculate the SoC can work well for lead-acid battery chemistries, for which the cell voltage and SoC are fairly linearly related, but is ineffective for lithium-ion, for which the voltage is mostly flat over the battery discharge curve (except at the extremes, where it is non-linear). Coulomb counting is one of the popular methods for determining the SoC. Battery capacity is determined by calculating the integral over time of the current being delivered or received by the battery. However, several factors affect the Coulomb counting accuracy, such as battery age, discharge rate and errors in the current detector. Other artificial intelligence methods based on adaptive systems have been proposed (neural networks, fuzzy logic, support vector machines, and the Kalman filter) offering different degrees of accuracy and computational complexity [15].

Manuscript received April 30, 2015; revised July 20, 2015. This work was supported in part by the Spanish Science and Innovation Ministry and the Regional Ministry of Principality of Asturias under Grants MINECO-13-DPI2013-46541-R, FC-15-GRUPIN14-073 and MINECO-15-TIN2014-56967-R

J. C. Álvarez Antón, J. C. Viera, M. González and C. Blanco Viejo are with the Electrical Engineering Department, University of Oviedo, Campus de Viesques, Gijón, Spain 33204 (+34 985-182553; e-mail: anton@uniovi.es).

P. J. Garcia Nieto and E. García Gonzalo are with the Mathematics Department, University of Oviedo, Spain 33007 (e-mail: pjgarcia@uniovi.es).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2504933, IEEE Transactions on Vehicular Technology

Statistical learning techniques are a novel alternative for SoC prediction. Statistical learning theory addresses the problem of finding a predictive function based on data. The goal is to learn a general rule that maps input variables to outputs. In [20], a statistical learning method called MARS was applied to estimate the SoC of a high capacity battery cell for a simple data profile (constant current-constant voltage charge and constant current discharge) and for a limited range of SoC. These limitations are partially the consequence of the fact that MARS parameters are difficult to estimate. In this paper, the MARS technique is applied to estimate SoC in a more useful and realistic scenario. A special regimen, called the Dynamic Stress Test (DST), is thus used as the data profile. The DST is a complex, high power load profile, like those required for electric vehicles, specified by the United States Advanced Battery Consortium (USABC) [21]. It has been developed to simulate real driving conditions and several manufacturers have incorporated these profiles into their test equipment. Furthermore, the MARS technique is combined in this paper with the particle swarm optimization (PSO) approach in order to find the optimal parameters of the MARS model. PSO is a population-based stochastic optimization technique inspired by the social conduct of bird flocking. The system is initialized with a population of random solutions and searches for optima by updating generations. PSO learned from the scenario and used it to solve the optimization problem.

No previous work has been reported on the application of this hybrid PSO-MARS-based model to estimate the SoC of a battery cell using data provided from a DST test. Furthermore, this flexible modeling technique is applicable to a wide variety of data analyses. Hence, its applications can be easily extended, particularly to those problems in which interaction effects between variables are relevant.

II. Materials and methods

A. Experimental dataset

The experimental dataset was obtained from a lithium iron manganese phosphate (LiFeMnPO4) battery cell with a nominal capacity of 60 Ah. This cell constitutes a new generation of lithium-ion rechargeable battery that uses LiFeMnPO4 as the cathode material. These cells possess good thermal stability, excellent safety properties and good cyclic performance. This technology has become the top choice in terms of cathode materials for power batteries in electric vehicles. The main characteristics of the cell are given in Table I [22].

First, the battery is charged and temperature stabilized following the battery manufacturer’s recommendations. Starting from full charge, the battery is discharged by applying the scaled DST power profile.

This variable power discharge regimen represents the best available simulation of the actual power requirements of an electric vehicle. Fig. 1 shows a graphical representation of the DST test profile used in this paper. The DST is scaled to a percentage of the maximum rated power. The corresponding DST values are given in Table II. In Fig. 1, a negative current value represents a battery discharge current; a positive current value means a charge current. This discharge regimen is continued until either reaching the end-of-discharge point specified in the test plan or until it is no longer possible to follow the test profile within the battery limit, whichever occurs first [23].

TABLE I
Listing of DST profile

Parameter	Value
Manufacturer	GBS Energy Co.
Model	GBS-LFP60AH
Technology	LiFeMnPO4
Nominal capacity	60 Ah
Nominal voltage	3.2 V
Charging/Discharging cut-off voltage	3.55 V / 2.8 V
Recommend charging current	0.25 C 15 A
Max. discharge current (t<10s)	600 A
Max. discharge current	180 A
Cycle life (at 80% DoD, 0.3 C)	2000 cycles
Operating temperature (charge/discharge)	0 ºC to 45 ºC / −25 ºC to 65 ºC

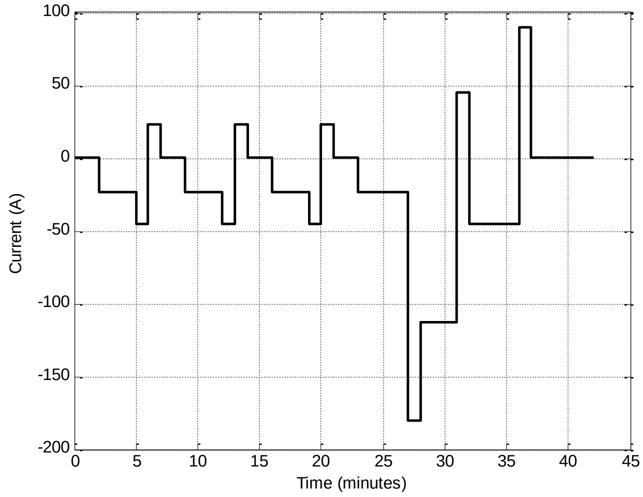


Fig. 1. Dynamic stress test (DST profile).

Fig. 2 shows the experimental data used for training the MARS model. In addition to the current, the variables measured during the DST test are the cell voltage and cell temperature. These variables are used as input variables for the model.

Four consecutive DST cycles are performed until the cell reaches the cut-off voltage (2.8 V) as shown in Fig. 2.

TABLE II
LISTING OF DST PROFILE

Step No.	**Duration (minutes)**	**Amplitude [%]**	**Current [A]**
1	2	0.0%	0
2	3	−12.5%	−23
3	1	−25.0%	−45
4	1	12.5%	23
5	2	0.0%	0
6	3	−12.5%	−23
7	1	−25%	−45
8	1	12.5%	23
9	2	0.0%	0
10	3	−12.5%	−23
11	1	−25%	−45
12	1	12.5%	23
13	2	0.0%	0
14	4	−12.5%	−23
15	1	−100%	−180
16	3	−62.5%	−113
17	1	25%	45
18	4	−25%	−45
19	1	50%	90
20	5	0.0%	0

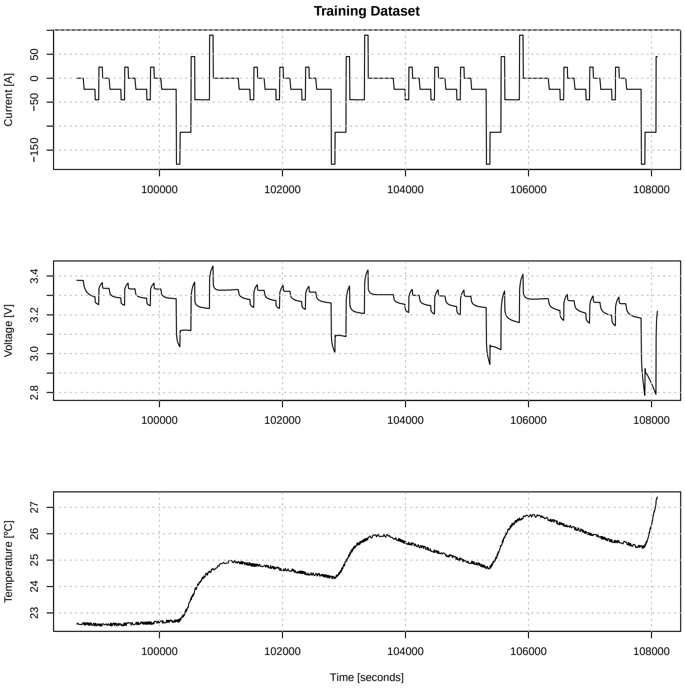


Fig. 2. Experimental dataset.

If the programmed current value for any step of the test profile cannot be performed within the discharge voltage limit, the discharge is terminated at that point. During these cycles, the training current ranges between 90 A and −180 A. Several plateaus with 0 A current and a steady-state SoC were included as training for the model. The experimental SoC to be estimated by the MARS model is shown in Fig. 3 and it ranges from 100% to 0% .

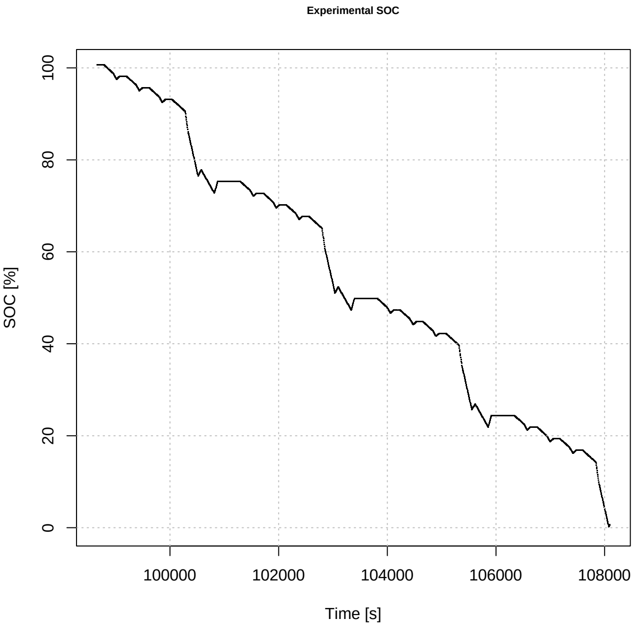


Fig. 3. Experimental SoC.

In this paper, the estimation of the experimental SoC is accurately calculated by the Coulomb counting method. This method is critical in verifying the accuracy of estimated results from other methods. In fact, the discharge test with Coulomb counting presents a standard reference for assessing the accuracy of other techniques. The accuracy of the Coulomb counting method relies primarily on accurate measurement of the battery current during the charging/discharging process. In this research, the battery current is calculated using the internal current meters provided by the power supply (during charge) and the electronic load (during discharge) of the battery workbench used during test. The battery workbench is based on programmable standalone instruments. An Agilent 6050A multiple module Electronic Load Mainframe is used for battery discharging purposes. Each load module (60504B) allows up to 600 W, with a current range of 0 to 120 A. The internal meter of the electronic load is used to measure the battery current with an accuracy of ±0.1% ±110 mA. An AMETEK Sorensen DHP series DC high power programmable supply is used for battery charging purposes. The power supply can be configured in Constant Current (CC) or Constant Voltage (CV) operation mode with an automatic crossover feature. It can deliver up to 60 V and 220 A. The power source has a built-in internal meter used for current measurements (accuracy of 0.3% + 0.3% of full-scale output current). The HP34970A is a modular data acquisition instrument. This module is used to acquire battery voltage and temperature. A total accuracy of 260 µV is obtained for dc measurements. A RTD sensor placed on the battery chassis is used to measure temperature. As battery performance largely depends on temperature, an electronically controlled environmental chamber is used. All workbench instruments are linked to the computer using a GPIB bus. High-level software based on NI LabVIEW™ control the workbench operations [24]. Fig. 4 shows the battery workbench. A detailed description of the test battery workbench operation is given in [25].

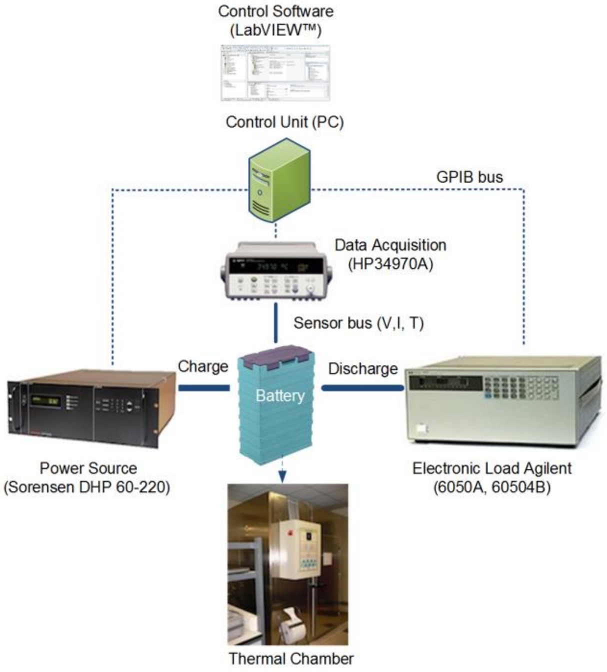


Fig. 4. Battery workbench.

The PSO-MARS technique was applied to a LiFeMnPO4 cell, although its use could extend to a pack of cells provided the measurements are carried out on the pack as a whole. The possibility also exists of measuring each individual cell in the pack, subsequently obtaining the joint response.

B. Multivariate adaptive regression splines method (MARS)

Developed in 1991 by Friedman [26], MARS is an adaptive procedure and multivariate non parametric classification / regression technique for solving high dimensional problems. Non-parametric models differ from parametric models in that the model structure is not defined a priori, but is constructed according to the information derived from the data. This does not mean that such models completely lack parameters, but rather that the number and nature of the parameters are flexible and not fixed in advanced. In general, non-parametric methods make fewer assumptions, their applicability is much wider and they are more robust than the corresponding parametric methods. The MARS technique is able to deal with multidimensional data and examine individual features and interactions between them. The method automatically selects the predictors that take part in the final model and deletes the predictors that do not contribute sufficiently to the performance of the final model. Furthermore, MARS makes no assumption about the underlying functional relationship between the dependent and independent variables. The relationship between variables is totally data driven. The algorithm builds the model from input variables in order to make data-driven predictions of the response variable (SoC), instead of following strictly static program instructions. This means that several effects as hysteresis and relaxation can be modeled if that data is provided to the model. Data provided within the DST test can thus be used for this purpose.

MARS uses a stepwise procedure to introduce and delete explanatory variables while also taking into account transformations and interactions between variables. This means that the predictive model can take the form of a product of functions. These multiplicative terms are used to form non-linear functions that keep the computational complexity within reasonable bounds. Besides accuracy, the primary advantage of MARS is its robustness. MARS tends to be resistant to moderate to heavy contamination by bad measurements (outliers) of the predictors and/or the responses, missing values and to the inclusion of potentially large numbers of irrelevant predictor variables that have little or no effect on the response.

In this study, the response variable (y-variable) is the SoC and the predictors (x-variables) are cell voltage, cell current and cell temperature. The dataset consists of a collection of N previously solved cases or samples: $\{y_i, x_{i1},..., x_{in}\}_{i=1}^{N}$. In each sample (from 1 to N), a response variable and a set of predictor variables (from 1 to n) are obtained. The predictive model takes the following abstract form in MARS:

$$\hat{y} = \hat{f}\left(x_1,..., x_n\right) \tag{1}$$

where $\hat{f}$ is a prediction rule that maps a set of predictor variable values to a response value. The goal is to use the data to produce an accurate mapping. To check the goodness of fit, several measurements can be considered. In general, lack of accuracy is defined in terms of a distance measure between the values predicted by the MARS technique, $\hat{y}_i$, the experimental value, y_i , and the mean of the N observed data, $\bar{y}$.

The coefficient of determination is used in the context of statistical models whose main purpose is the prediction of future outcomes based on other, related information. It is the proportion of variability in a dataset that is taken into account by the statistical model. It thus provides a measure of how well future outcomes are likely to be predicted by the model. The coefficient of determination ranges from 0 to 1, where 1 indicates a perfect fit between observed and modeled values.

This statistic was chosen in the present research study to estimate the goodness of fit of the MARS model.

The MARS model of a response variable, y, can be written as [27,28]:

$$\hat{y} = \hat{f}_M(x) = \hat{f}_M(x_1,..., x_n) = \sum_{m=1}^{M} c_m B_m(x_1,..., x_n) \tag{2}$$

where $\hat{y}$ is the variable predicted by the MARS model, $B_m(x_1,..., x_n)$ is the m-th basis function, and c_m is the coefficient of the m-th basis functions. In this respect, the MARS technique automatically models nonlinearities and interactions between variables as a weighted sum of basis functions. Basis functions can take the form of a constant, 1, a hinge function, or a product of two or more hinge functions in order to form non-linear functions. Hinge functions take the form of $h(x-c)$ or $h(c-x)$, where c is a constant called the knot, and $h(z)$ is a function that returns 0 if $z<0$ and z if $z>0$. A hinge function is illustrated graphically in Fig. 5, which depicts the shape of two simple hinge functions, $h(3.5-x)$ and $h(x-3.5)$, with a knot at the point 3.5. MARS automatically selects predictor variables and values of these variables for knots of the hinge functions. One of the advantages of MARS lies in its ability to estimate the contributions of the basis functions, thus allowing both the additive and the interactive effects of the predictors to determine the response variable.

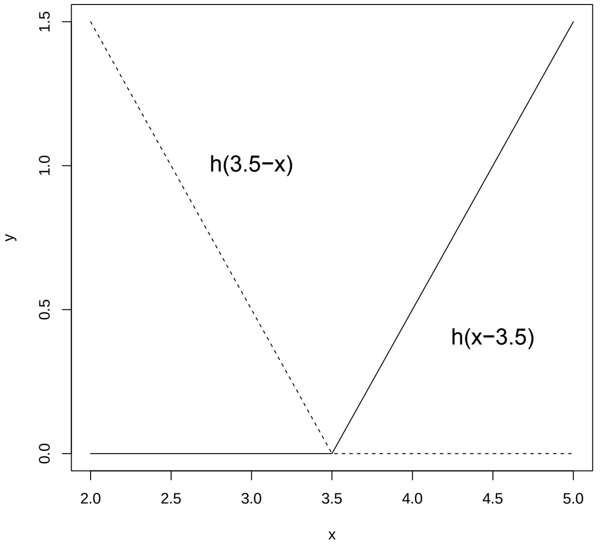


Fig. 5. A mirrored pair of hinge functions.

The MARS model is selected in a two-phase process: forward selection and backward deletion. Fig. 6 shows a flowchart of the main process followed by the MARS technique. With regard to the forward phase, the X-matrix contains the variables that are referred to as predictor variables. Each column of the X-matrix is one predictor variable (from 1 to n predictors), while each row is the set of observations of the predictors (from 1 to N observations). The Y-column vector is the response variable (SoC). The algorithm starts in the forward phase with a model consisting of just the intercept term (a constant) and interactively adds the reflected terms of the basis functions. The reflected terms of the basis functions are identical except for the fact that a different side of a mirrored hinge function is used for each function, as shown in Fig. 5. MARS repeatedly adds basis functions in pairs to the model. In each step, it finds the pair of basis functions that provides the maximum reduction in RSS error. A predictor variable and a knot define a hinge function. In order to add a new basis function, MARS must search over all combinations of the following: (a) existing terms, (b) all predictor variables, and (c) all values of each variable for the knot of the new hinge function. This process of adding terms continues until the change in residual error is too small to continue or until the maximum number of terms is reached.

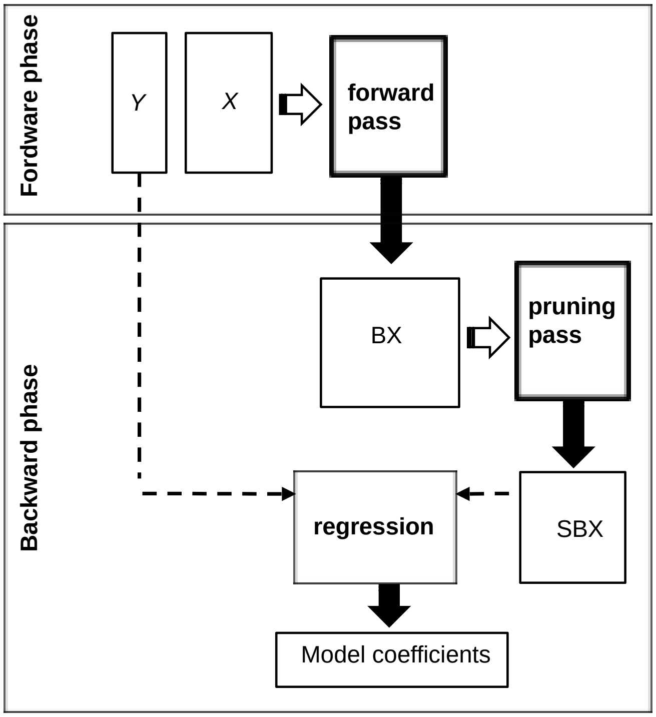


Fig. 6. The MARS flowchart: its stages.

The basis functions BX-matrix is the result of the forward pass (see Fig. 6). The BX-matrix has a row for every observation and a column for each basis function, also known as a MARS term. At the end of the forward phase, we have a large model that typically overfits the data. Due to this problem, a backward deletion phase is implemented via which the algorithm prunes the model. It removes terms one by one, deleting the least effective term in each step until it finds the best sub-model. The forward pass adds terms in pairs, but the backward pass typically discards one side of the pair. Hence, terms are often not seen in pairs in the final model. The pruning pass is handed the set of terms (columns) of the BX-matrix created by the forward pass. The pruning pass determines the subset of terms in the BX-matrix with the lowest RSS for each model size. It then calculates the Generalized Cross Validation (GCV) with a penalty for each entry of the RSS subsets. The backward job is thus to find the subset of those terms that provides the lowest GCV. At the end of the backward phase, one model with the lowest GCV value is selected from these "best" models of each size and outputted as the final one. This phase yields the SBX-matrix, which keeps only the selected terms. After the pruning pass, MARS determine the coefficients (cm) of the basis functions by regressing the response vector, Y, on the SBX-matrix (see Fig. 6).

GCV is a form of regularization: it trades off goodness of fit against model complexity. The raw RSS on the training data is inadequate for comparing models, because the RSS always increases as MARS terms are dropped. In other words, if the RSS were used to compare models, the backward pass would always choose the largest model. However, the largest model generally does not have the best generalization performance [20]. The GCV formula thus adjusts (i.e. increases) the raw training RSS to take into account the flexibility of the model [20, 27]. GCV is the mean squared residual error (MSE) divided by a penalty that depends on the complexity of the model. The GCV for a model, also known as the lack-of-fit criterion, is calculated as follows [27, 28]:

$$GCV = \frac{MSE}{\left(1 - \frac{enp}{N}\right)^2} \tag{3}$$

where N is the number of data samples in the training data, and *enp* is the effective number of parameters [27]. The *enp* term can be expressed as:

$$enp = M + d\,\frac{(M-1)}{2} \tag{4}$$

where M is the number of basis functions in the model (including the intercept term), and d is a penalty parameter. It should be noted that $(M-1)/2$ is the number of hinge functions knots. Therefore, the GCV formula penalizes not only the number of basis functions, but also the number of knots by means of the penalty parameter.

The penalty parameter, the maximum number of basis functions and the maximum interaction level are usually estimated empirically. For instance, with regard to the GCV penalty (d parameter), theory suggests values ranging from about 2 to 4 [26]. In the present study, the d parameter, the maximum number of basis functions and the maximum interaction level were calculated using the particle swarm optimization (PSO) algorithm.

C. The particle swarm optimization (PSO) algorithm

The particle swarm optimization (PSO) algorithm is an evolutionary optimization algorithm that was presented in 1995 [29], inspired by the social behavior of bird flocking [30]. The PSO has been adapted to solve different kind of problems such as discrete or multi-objective problems. One of the reasons for its success is that the PSO algorithm is relatively easy to implement, understand and modify. The PSO algorithm belongs to the group of evolutionary algorithms, in which a population of individuals, or proposed solutions, changes with time. Genetic algorithms [31], differential evolution [32] and the ant colony system [33] belong to this group, in which the evolution of the population has a stochastic component. Depending on the algorithm, different strategies are used to update a new population from the previous one.

In the case of the PSO algorithm, the new population is considered the same as in the previous iteration, except that the individuals have changed position. The movement of each individual is influenced by its own history and the experience of its neighbors.

PSO is an optimization algorithm that does not use gradient, so the fitness function, that describes the problem to be optimized, can be partially irregular, and it needs to make few or no assumptions about it. It also can search with success in very large spaces of potential solutions and, compared with other methods, has few parameters to adjust.

Each particle, or individual in the swarm, is a vector, x_i, which contains the parameters whose values we are attempting to determine in order to optimize the objective function. The particle length is the dimension of this function. Its position, x_i, and velocity, v_i, are randomly initialized in a space of possible solutions. The objective function value is then calculated for each particle and velocities and positions are updated taking into account these values. The algorithm updates the positions, X_i^k, and the velocities, V_i^k, of the particles according to the following equations:

$$V_i^{k+1} = \omega\, V_i^k + \phi_1\left(g^k - X_i^k\right) + \phi_2\left(I_i^k - X_i^k\right) \tag{5}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{6}$$

The velocity of each particle, at iteration *k*, depends on three components:

- The previous step velocity term, V_i^k, affected by the constant inertia weight, ω.
- The cognitive learning term, which is the difference between the particle's best position found so far (called I_i^k, the local best) and the particle's current position, X_i^k.
- The social learning term, which is the difference between the global best position found so far (called g^k, the global best) and the particle's current position, X_i^k.

These two last components are affected by $\phi_1 = c_1 r_1$ and $\phi_2 = c_2 r_2$, where r_1 and r_2 are random numbers distributed uniformly within the interval [0,1], and c_1 and c_2 are constants.

The swarm topology defines how particles are connected to one another so as to exchange information with the global best. All particles are informed in a star topology, but a different topology can be defined in which only some particles receive the information. In the actual Standard PSO [34], for instance, each particle informs only k particles, usually three.

The particles in the swarm make up a cloud that covers the entire search space in the initial iteration and gradually contracts in size as the iterations advance performing the exploration. The algorithm thus performs an exploration searching for plausible zones in the initial stages, with the best solution being improved in the last iterations.

III. Results and discussion

A. Model parameters Equations

The MARS model was obtained with the ARESLab library [35]. The modified ARESLab parameters are:

a) MaxFuncs: the maximum number of basis functions included in the model in the forward phase. It includes the intercept term.
b) Penalty: the generalized cross-validation (GCV) penalty per knot.
c) Type of spline function: whether to use piecewise-cubic (1) or piecewise-linear (0) as interpolation functions.
d) Interactions: the maximum degree of interactions between input variables.
e) MaxFinalFuncs: the maximum number of basis functions (including the intercept term) in the pruned model.

B. Optimized parameters using the PSO algorithm

The PSO version used in this paper is the Standard PSO 2011 [34]. The flowchart of the hybrid PSO-MARS-based model is shown in Fig. 7. The parameters optimized with the PSO technique are MaxFuncs, Penalty and Interactions. The parameter search space is shown in Table III.

TABLE III
Parameter search space

ARESLab parameter	Lower limit	Upper limit
MaxFuncs	2	100
Penalty	2	5
Interactions	2	4

In each iteration, particles are attracted to their own personal best position so far and to the best-known position in their neighborhood, which depends on the value of the topology. The PSO parameters are set to the values defined in the Standard PSO 2011:

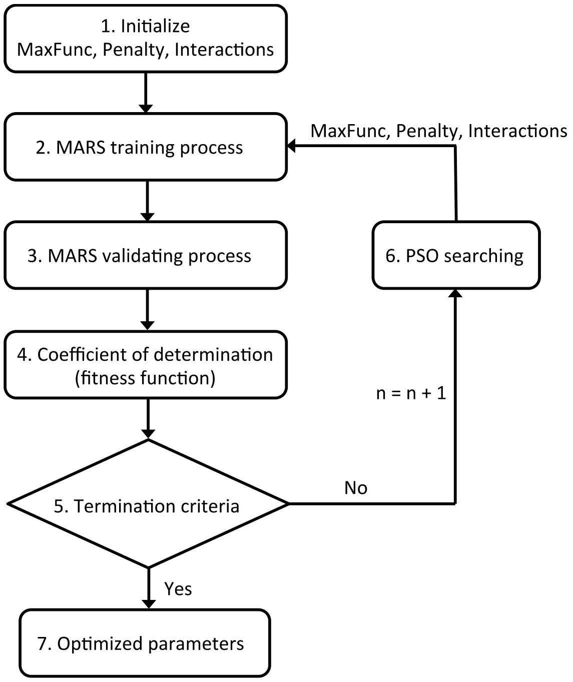


Fig. 7. Flowchart of the hybrid PSO-MARS-based model.

$$\omega = \frac{1}{2\ln 2} \quad \text{and} \quad c_1 = c_2 = 0.5 + \ln 2 \qquad (7)$$

PSO algorithm begins selecting ten (the number of particles) randomly chosen sets of the parameters in Table III X_i^0 within the parameter search space, also in Table III. For each one of these sets, the corresponding cross-validation R^2 is found using the MARS algorithm. From this starting point, and following the governing equations (5) and (6) new sets of parameters X_i^k (k is the number of iteration) are obtained. The R^2 is improved through this process and an optimal parameter set is found (see Table IV).

TABLE IV
Optimal parameter values

Parameter	Optimal value
MaxFuncs	58
Penalty	5
Interactions	2
R	0.9832

C. The hybrid PSO-MARS-based analytical model

The results are reported in Table V, which shows a list of the main basis functions and their coefficients. A second-order PSO-MARS-based model is obtained using a piecewise-linear regression with a maximum of 30 basis functions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2504933, IEEE Transactions on Vehicular Technology

The cell input variables are represented as: x_1 (cell current), x_2 (cell voltage) and x_3 (cell temperature).

TABLE V
BASIS FUNCTIONS AND COEFFICIENTS OF THE SOC MODEL

Basis functions	Coefficients
Intercept	– 3.92
$BF1 = h(0, 24.496 - x_3)$	– 54.38
$BF2 = h(0, x_3 - 24.496) \times h(0, x_2 - 3.241)$	268.53
$BF3 = h(0, x_3 - 24.496) \times h(0, 3.241 - x_2)$	326.40
$BF4 = h(0, x_1 + 0.006)$	4.09
$BF5 = BF4 \times h(0, 3.285 - x_2)$	14.59
$BF6 = h(0, x_2 - 3.06)$	544.00
$BF7 = h(0, x_3 - 24.496) \times h(0, x_1 + 44.953)$	0.19
$BF8 = h(0, x_2 - 3.331)$	– 835.84
$BF9 = h(0, -0.006 - x_1) \times h(0, 3.292 - x_2)$	0.49
$BF10 = h(0, x_2 - 2.887) \times h(0, 23.048 - x_1)$	1551.73
$BF11 = h(0, x_2 - 2.887) \times h(0, 23.03 - x_1)$	1550.98
$BF12 = BF6 \times h(0, 23.228 - x_3)$	143.59
$BF13 = h(0, -0.006 - x_1) \times h(0, x_3 - 24.361)$	0.23
$BF14 = BF6 \times h(0, x_3 - 25.512)$	– 206.69
$BF15 = BF6 \times h(0, 25.512 - x_3)$	– 142.05
$BF16 = BF4 \times h(0, x_2 - 3.25)$	7.40
$BF17 = BF4 \times h(0, 3.25 - x_2)$	– 33.53
$BF18 = h(0, 3.331 - x_2) \times h(0, x_3 - 24.465)$	301.57
$BF19 = h(0, 3.331 - x_2) \times h(0, 24.465 - x_3)$	206.24
$BF20 = BF6 \times h(0, x_1 + 44.953)$	22.32
$BF21 = BF4 \times h(0, \mathrm{x3} - 24.644)$	– 0.29
$BF22 = BF4 \times h(0, 24.644 - x_3)$	0.18
$BF23 = h(0, x_1 + 112.807) \times h(0, 25.531 - x_3)$	0.41
$BF24 = h(0, x_1 + 112.807) \times h(0, x_2 - 3.226)$	12.43
$BF25 = h(0, x_1 + 112.807) \times h(0, 3.226 - x_2)$	– 8.36
$BF26 = h(0, 3.331 - x_2) \times h(0, \mathrm{x1} + 23.01)$	44275.41
$BF27 = h(0, -0.006 - x_1) \times h(0, x_2 - 3.121)$	– 16.49
$BF28 = h(0, 3.331 - x_2) \times h(0, x_1 + 23.007)$	– 44272.29
$BF29 = h(0, x_2 - 2.887) \times h(0, 25.25 - x_3)$	93.77

The analytical expression of the SoC derived from Table V is:

$$
\begin{aligned}
SoC_i &= -3.92 - 54.38 \times BF1 + 268.53 \times BF2 - 326.40 \times BF3 \\
&+ 4.09 \times BF4 + 14.59 \times BF5 + 544.004 \times BF6 + 0.19 \times BF7 \\
&- 35.84 \times BF8 + 0.49 \times BF9 + 1551.73 \times BF10 - 1550.98 \times BF11 \\
&+ 143.59 \times BF12 - 0.23 \times BF13 - 206.69 \times BF14 - 142.05 \times BF15 \\
&+ 7.40 \times BF16 - 33.53 \times BF17 + 301.57 \times BF18 + 206.24 \times BF19 \\
&- 22.32 \times BF20 - 0.29 \times BF21 - 0.18 \times BF22 + 0.41 \times BF23 \\
&+ 12.43 \times BF24 - 8.36 \times BF25 + 4427541 \times BF26 - 16.49 \times BF27 \\
&- 4427229 \times BF28 + 93.77 \times BF29
\end{aligned}
\tag{8}
$$

This mathematical expression is not computationally intensive and is straightforward to implement using a microcontroller unit (MCU). Microcontrollers are well suited to algorithms that are executed sequentially with a rate that falls within the microcontroller processor's capability. This feature makes the PSO-MARS-based model suitable for implementation on a low cost microcontroller due to its reduced computational load.

D. Importance of variables

The importance of variables is a measure of the effect that observed changes to the variable have on the observed response. Estimating predictor importance is generally a controversial issue. There is usually no completely reliable way to estimate the importance of the variables in a standard MARS model. In this paper, the importance of the variables is obtained from the ANOVA decomposition of the model [26], as shown in Table VI. "STD", "GCV", "#basis" and "#params" can be used as a criterion for estimating the importance of the input variable or set of input variables in the model.

TABLE VI
ANOVA DECOMPOSITION OF THE SOC MODEL

Func.	STD	GCV	#basis	#params	variables
1	66.85	5717.78	1	3.5	Current
2	40.79	12803.69	2	7.0	Voltage
3	40.27	2394.27	1	3.5	Temperature
4	94.80	18604.76	12	42.0	Current ×Voltage
5	47.17	3427.39	5	17.5	Current ×Temp
6	21.16	992.16	8	17.5	Voltage ×Temp

The first column in Table VI lists the ANOVA function number. The second column gives the standard deviation of the function values predicted using a reduced model. This reduced model only considers the terms in Table V that contain the variable or variables specified in the last column (variable column) of Table VI. For Func. 1, for instance, the reduced model includes only the intercept and the term BF4.

For Func. 5, the reduced model includes the intercept and the terms BF7, BF13, BF21, BF22 and BF23. This standard deviation provides an indication of its (relative) importance to the overall model and can be interpreted in a similar way to a standardized regression coefficient in a linear model. The third column gives the GCV score for the reduced model, obtained by removing the variable terms in Table VI. For instance, for Func. 1, the reduced model includes all the terms in Table V except for the term BF4. For Func. 5, the reduced model includes all the terms in Table V except for the terms BF7, BF13, BF21, BF22 and BF23. The quantitative evaluation equation of GCV is given by Equations (3) and (4). The GCV score is a measure of the loss suffered by the model in terms of efficiency when excluding these variables from the proposed model. That is, it is the cost of omission. The cost of omission shows the unrecoverable loss of model performance due to variable exclusion from the existing model [36]. This can be used to judge whether this ANOVA function is making an important contribution to the model or whether it just helps to improve the global GCV score slightly. The fourth column gives the number of basis functions comprising the ANOVA function. For instance, for Func. 1, there is one basis function (BF4), for Func. 5, there are five basis functions (BF7, BF13, BF21, BF22 and BF23). Variables that are included in more basis functions are considered more important. The fifth column provides the effective number of parameters for the reduced model. This expression is given by Equation (4). The last column gives the particular predictor variables associated with the ANOVA function [26].

If we normalize the GCV column by dividing all its values by the highest value, multiplying them by 100 and sorting them, we obtain the results shown in Table VII, i.e., the normalized GCV values. This table provides us with a criterion to evaluate the relative importance of the variables in the model.

TABLE VII
VARIABLE IMPORTANCE ACCORDING TO GCV

Normalized *GCV*	Variables
100	Current × Voltage
69	Voltage
31	Current
18	Current × Temperature
13	Temperature
5	Voltage × Temperature

According to the results shown in Table VII, the interaction represented by Current × Voltage is found to be the most important variable in estimating the SoC. In such a situation, this product, associated with the electrical power delivered by the cell, explains the non-linear relationship between the input variables and the SoC estimated by the model better than other model variables. The cell voltage is the next most significant variable in this study. The importance of other variables is shown in Fig. 8.

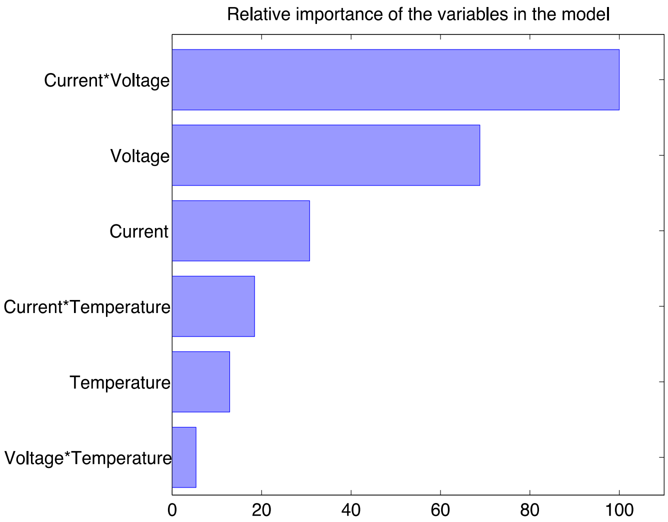
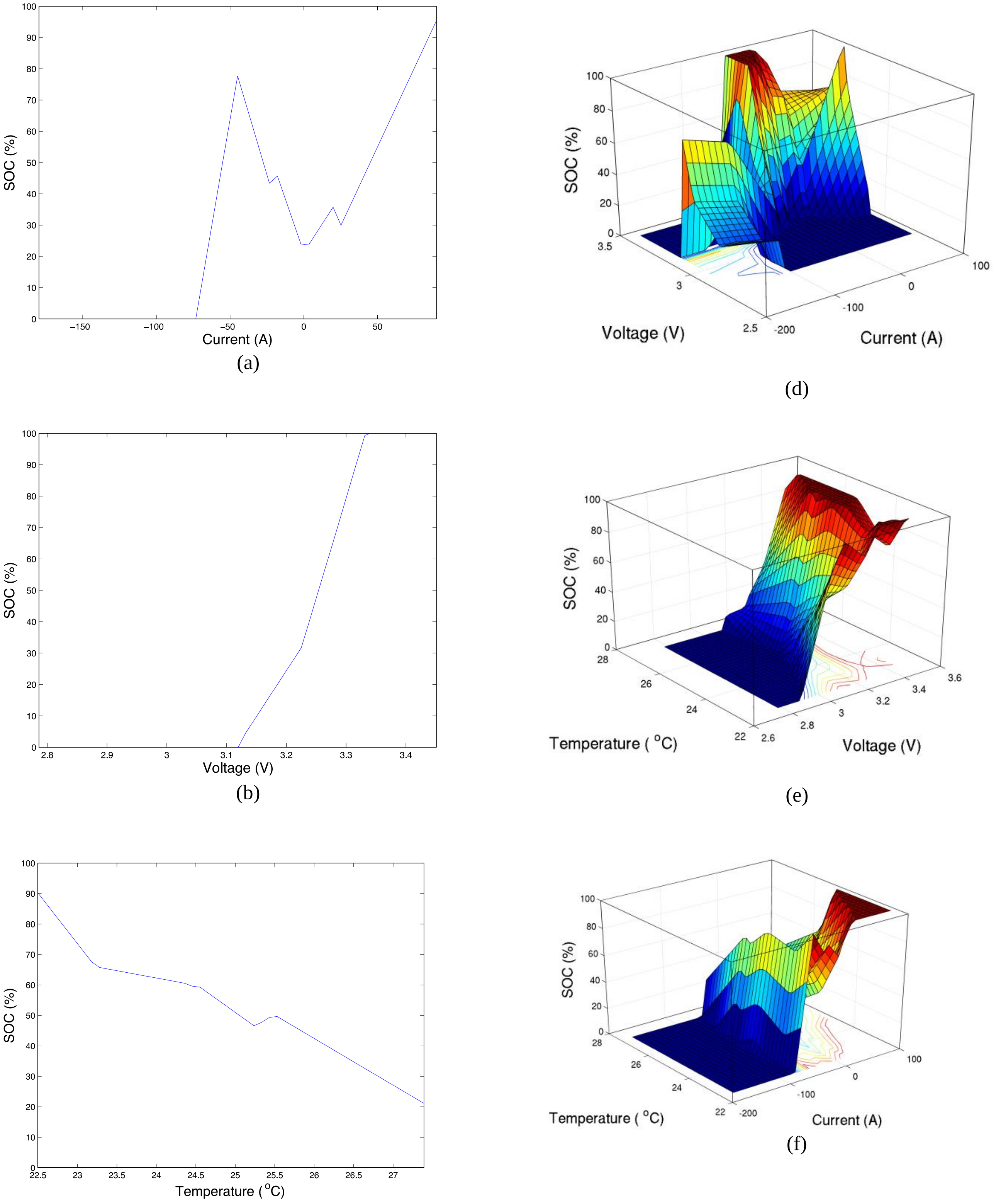


Fig. 8. Evaluation of the importance of the input variables that make up the SoC model.

After identifying the most relevant predictor variables, the next step is to get an idea of the dependence of the MARS model on each of them. Fig. 9 shows the graphical representation of the terms that constitute the hybrid PSO-MARS-based model. Fig. 9 is a plot of the response of the model (according to Equation 8) when varying one predictor while maintaining the other variables predictors constant at their median values. Figs. 9(a), 9(b) and 9(c) respectively indicate the predicted SoC as current, voltage and temperature vary, with the other variables fixed at their median values (m*Current* = −23.01 A, m*Voltage* = 3.26 V and m*Temperature* = 24.91 ºC). It should be borne in mind that each graph shows only a thin slice of the data, as most variables are fixed. Of course, we must be aware of this fact when interpreting the graphs, as the tendency to over-interpret can lead to meaningless results. A non-linear, uneven curve, i.e., one that is not flat, suggests that the independent variable (current in Fig 9a) affects the dependent variable (SOC, y-axis). Figs. 9(d), 9(e) and 9(f) shows the model response when two variables (current and voltage; voltage and temperature; and current and temperature, respectively) vary while the remaining variables are kept constant at their median values. If the shape of the dependence on either variable is unaffected by the value of the other variable, this suggests that there is no interaction between the two variables. Flexible surfaces are observed in Figs 8d-e-f, especially in Fig 8d, which suggests that considerable interaction exists between voltage and current.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2504933, IEEE Transactions on Vehicular Technology



(a)

(d)

(b)

(e)

(f)

(c)

Fig. 9. Graphical representation of the terms that constitute the hybrid SoC PSO-MARS-based model: (a) first-order term of the predictor variable Current; (b) first-order term of the predictor variable Voltage; (c) first-order term of the predictor variable Temperature; (d) second-order term of the predictor variables Current and Voltage; (e) second-order term of the predictor variables Temperature and Voltage; and (f) second-order term of the predictor Temperature and Current.

Finally, a comparison between the predicted and experimental SoC is shown in Fig. 10. The result confirms the model fit with a coefficient of determination of 0.98.

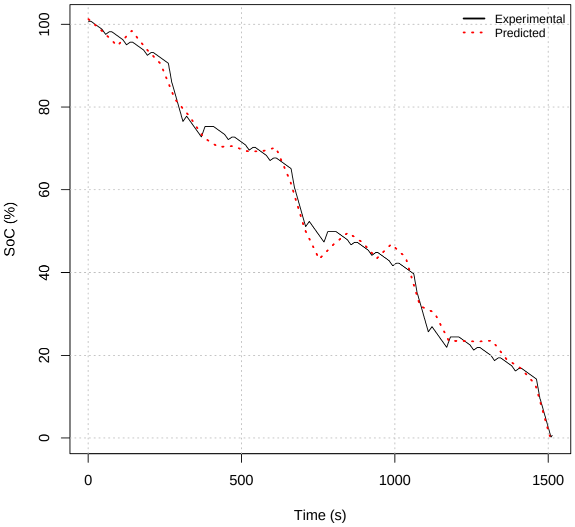


Fig. 10. SoC predicted by the hybrid PSO-MARS-based model vs. the experimentally-determined SoC.

IV. Conclusions

The growing use of batteries has given rise to the need for more accurate, faster methods to estimate battery SoC. In this paper, the MARS technique was used to estimate the SoC of a high capacity lithium (LiFeMnPO4) battery cell in combination with the particle swarm optimization (PSO) technique. Battery data were obtained from a Dynamic Stress Test (DST) specified by the United States Advanced Battery Consortium (USABC). The optimal parameters of the MARS model were found using the PSO algorithm. The performance of the SoC estimation using this hybrid technique can be efficiently improved by selecting: a) the optimal GCV penalty parameter, b) the maximum number of basis functions, and c) the maximum degree of interaction between input variables. The PSO search reduces the time needed to train MARS and avoids searching the parameters in a large search space. Moreover, the PSO-MARS-based technique can handle nonlinear relationships and interactions between these independent variables, providing an interpretable model. We thus found the interaction represented by Current Voltage to be the most important factor in the SoC estimation using this modeling technique.

The results confirm the model fit and quick estimation of SoC with a coefficient of determination of 0.98 using cell voltage, current and temperature as input variables. Furthermore, this hybrid PSO-MARS-based model is not computationally intensive and is straightforward to implement. This feature makes the hybrid PSO-MARS-based model suitable for implementation on a low cost microcontroller due to its reduced computational load.

Finally, the model can be easily modified to fit data from different batteries successfully, as hybrid PSO-MARS-based models are more flexible and accurate than linear regression models. It is particularly useful for successfully tackling highly nonlinear problems.

The proposed approach may constitute a useful methodology for process optimization with major applications in other fields.

Acknowledgment

The authors wish to acknowledge the computational support provided by the Department of Mathematics at the University of Oviedo. Finally, we would like to thank Paul Barnes for his revision of English grammar and spelling of the manuscript.

References

[1] D. Andre, C. Appel, T. Soczka-Guth, D. Sauer, “Advanced mathematical methods of SOC and SOH estimation for lithium-ion,” *J. Power Sources*, vol. 224, pp. 20–27, Feb. 2013.

[2] V. Pop, H. Bergveld, J. Veld, P. Regtien, D. Danilov, P. Notten, “Modeling battery behavior for accurate state–of–charge indication,” *J. Electrochem. Soc.*, vol. 153, no. 11, pp. A2013–A2022, Sep. 2006.

[3] C. Blanco, L. Sánchez, J.C. Álvarez Antón, V. García, M. González, J.C. Viera, “A variable effective capacity model for LiFePO4 traction batteries using computational intelligence techniques,” *IEEE Trans. Ind. Electron.*, vol 62, no. 1, pp. 555–563, Jan. 2015.

[4] W. Waag, C. Fleischer, D.U. Sauer, “Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles,” *J. Power Sources*, vol. 258, pp. 321–339, Jul. 2014.

[5] S. Lee, J. Kim, J. Lee, B.H. Cho, “State-of-charge and capacity estimation of lithium ion battery using a new open-circuit voltage versus state-of-charge,” *J. Power Sources*, vol. 185, no. 2, pp. 1367–1373, Dec. 2008.

[6] M. A. Roscher, D.U. Sauer, “Dynamic electric behavior and open-circuit-voltage modeling of LiFePO4-based lithium ion secondary batteries,” *J. Power Sources*, vol. 196, no. 1, pp. 331–336, Jan. 2011.

[7] W. X. Shen, C.C. Chan, E.W.C. Lo, K.T. Chau, “Adaptive neuro-fuzzy modeling of battery residual capacity for electric vehicles,” *IEEE Trans. Ind. Electron.*, vol. 49, no. 3, pp. 677–684, Jun. 2002.

[8] Y. S. Lee, W.Y. Wang, T.Y. Kuo, “Soft computing for battery state-of-charge (BSOC) estimation in battery string systems,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 229–239, Jan. 2008.

[9] S.M. Rezvanizaniani, Z. Liu, Y. Chen, J. Lee, “Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (EV) safety and mobility,” *J. Power Sources*, vol. 256, pp. 110–124, Jun. 2014.

[10] T. Hansen, C. J. Wang, “Support vector based battery state of charge estimator,” *J. Power Sources*, vol. 141, no. 2, pp. 351–358, Mar. 2005.

[11] M. Charkhgard, M. Farrokhi, “State-of-Charge estimation for Lithium-ion batteries using neural networks and EKF,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4178−4187, Dec. 2010.

[12] H. Dai, X. Wei, Z. Sun, J. Wang, W. Gu, “Online cell SOC estimation of Li-ion battery packs using a dual time-scale Kalman filtering for EV applications,” *Appl. Energ.*, vol. 95, pp. 227−237, Jul. 2012.

[13] J. Kim, S. Lee, B. H. Cho, “Complementary Cooperation Algorithm Based on DEKF Combined with Pattern Recognition for SOC/Capacity Estimation and SOH Prediction,” *IEEE Trans. Power Electron.*, vol. 27, no. 1, pp. 436−451, Jan. 2012.

[14] J. N. Hu, J.J. Hu, H.B. Lin, X.P. Li, C.L. Jiang, X.H. Qiu, W.S. Li, "State of charge estimation for battery management system using optimized support vector machine for regression," *J. Power Sources*, vol. 269, pp. 682−693, Dec. 2014.

[15] Wen-Yeau Chang, "The State of Charge Estimating Methods for Battery: A Review," *ISRN Applied Mathematics.*, vol. 2013, pp. 1−7, Jul. 2013.

[16] M. Verbrugge, "Adaptive, multi-parameter based state estimator with optimized time-weighting factors," *J. Appl. Electrochem.*, vol 37, no. 5, pp. 605–616, Feb. 2007.

[17] Y. Hu, S. Yurkovich, "Battery cell state-of-charge estimation using linear parameter varying system techniques," *J. Power Sources*, vol. 198, no. 15, pp. 338–350, Jan. 2012.

[18] I. Kim, "Nonlinear state of charge estimator for hybrid electric vehicle battery," *IEEE Trans. Power Electron.*, vol. 23, no. 4, pp. 2027–2034, Jul. 2008.

[19] V. Pop, H.J. Bergveld, P. Notten, J. Veld, P. Regtien, "Accuracy analysis of the State-of-Charge and remaining run-time determination for lithium-ion batteries," *Measurement*, vol. 42, no. 8, pp. 1131–1138, Oct. 2009.

[20] J. C. Alvarez Antón, P.J. Garcia Nieto, F.J. de Cos Juez, F. Sánchez Lasheras, C. Blanco, N. Roqueñí, "Battery state-of-charge estimator using the MARS technique," *IEEE Trans. Power Electron.*, vol. 28, no. 8, pp. 3798–3805, Aug. 2013.

[21] *Electric Vehicle Battery Test Procedures Manual*, U.S. Advanced Battery Consortium (USABC), Southfield, MI, USA, 1996, pp. 15–16.

[22] GBS-LFP60Ah Datasheet. [Online]. Available: http://www.gbsystem.com/products-show-en.asp?id=45

[23] J. C. Viera, M. González, B.Y. Liaw, F.J. Ferrero, J.C. Álvarez Antón, J. C. Campo, C. Blanco, "Characterization of 109 Ah NiMH batteries charging with hydrogen sensing termination," *J. Power Sources*, vol. 171, no. 2, pp. 1040–1045, Sep. 2007.

[24] J. C. Álvarez Antón, J. C. Viera, F. Ferrero, M. González, C. Blanco, *Instrumentación Virtual con LabVIEW*. Oviedo, Spain: University of Oviedo Editions, 2009.

[25] J. C. Álvarez Antón, C. Blanco, F.J. Ferrero, J.C. Viera, M. González, M. Valledor, "Open Architecture Workbench for Testing Medium-High Capacity Batteries," in *Proc. SAEEI Conference*, Tanger, Morocco, 2014, pp. 1–6.

[26] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, Mar. 1991.

[27] J. H. Friedman, C. B. Roosen, "An introduction to multivariate adaptive regression splines," *Stat. Methods Med. Res.*, vol. 4, no. 3, pp. 197–217, Sep. 1995.

[28] P. J. García Nieto, J.C. Álvarez Antón, "Application of a MARS-Based Regression Model to the Air Quality Study at Local Scale in the Gijón Urban Area (Northern Spain)," in *Air Quality*, Arthur Hermans, Ed. New York: Nova Science Publishers, 2013, pp. 43–72.

[29] J. Kennedy, R. Eberhart, "Particle swarm optimization," in Proc. *IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, 1995, pp. 1942–1948.

[30] C.W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. 14th Annual Conference on Computer graphics and Interactive Techniques (SIGGRAPH '87)*, 1987, pp. 25–34.

[31] M. Mitchell, *An Introduction to Genetic Algorithms.* Massachusetts: Bradford Publisher, 1998.

[32] R. Storn, K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Nov. 1997.

[33] M. Dorigo, T. Stützle, *Ant Colony Optimization.* Massachusetts: Bradford Publisher, 2004.

[34] A. Subasi, "Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders," *Comput. Biol. Med.*, vol. 43, no. 5, pp. 576–586, Feb. 2013.

[35] G. Jekabsons. (2015, Oct.). ARESLab: Adaptive Regression Splines toolbox. [Online]. Available: http://www.cs.rtu.lv/jekabsons/

[36] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York: Springer, 2009.

Juan Carlos Álvarez Antón (M'08) was born in Veracruz, Mexico. He received the M.Sc. degree in Computer Engineering from the University of Valladolid (Spain) in 1996. He got his Ph.D. degree in 2007 from the University of Oviedo (Spain). He is currently with the Department of Electrical Engineering, University of Oviedo, as an Associated Professor. His research interests include lighting, electronic instrumentation systems and battery modeling.

Paulino García Nieto was born in Oviedo, Asturias, Spain, in 1965. He received the Bachelor's degree in mining engineering with specialization in fuels and energy, and the Master's and Ph.D. degrees all in mining engineering from the University of Oviedo, in 1989, 1990, and 1994, respectively. Since 1996, he has been with the Department of Mathematics, University of Oviedo, where he is currently a Professor. He was involved in research on numerical simulation as well as the application of the finite-element methods in numerous physical and engineering problems. He is currently involved in the application of the statistical learning and data mining to several biological, electrical, and physical datasets in order to obtain predictive models.

Esperanza García Gonzalo was born in Soria, Spain, in 1964. She received the M.Sc. degree in mining engineering in 1990 and the Ph.D. degree in mathematics from Oviedo University, Asturias, Spain, in 2010. She is an associate professor at the Mathematics Department (Oviedo University, Spain, 1999). Her research interests include global stochastic optimization methods, evolutionary algorithms, particle swarm optimization, regression methods and machine learning.

Juan Carlos Viera Pérez was born in Havana, Cuba, in 1969. He received the M.Sc. degree in Electrical Engineering from the University of Technology, Havana, in 1992 and the Ph.D. degree in Electrical Engineering from the University of Oviedo, Spain, in 2003. He is currently an Assistant Professor in the Dep. of Elect. and Elec. Eng. at the University of Oviedo. His research interests include battery testing, fast charging and battery management systems.

Manuela Gonzalez Vega received the M.Sc. and Ph.D. degree in electrical engineering from the University of Oviedo, Spain, in 1992 and 1998, respectively. She is the Founder and Head of the Battery Research Laboratory, Dep. of Elec. and Elec. Eng., University of Oviedo, where she is also currently an Associate Professor. Her research interests include battery management systems for new battery technologies and fast chargers for traction applications.

Cecilio Blanco Viejo was born in Lada, Spain. He received the M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Oviedo, Spain, in 1989 and 1996, respectively. In 1989 he joined the Electrical and Electronic Department of the University of Oviedo, where he is currently an Associated Professor. His research interests include high-frequency electronic ballast, discharge lamp and battery modeling.