batteries

MDPI

Article

Real-Time State-of-Charge Estimation via Particle Swarm Optimization on a Lithium-Ion Electrochemical Cell Model

Arun Chandra Shekar [1] **and Sohel Anwar** [2,*]

[1] KPIT Technologies, Inc., Columbus, IN 47201, USA; arunhsc@gmail.com
[2] Department of Mechanical and Energy Engineering, Purdue School of Engineering and Technology, IUPUI, Indianapolis, IN 46202, USA
* Correspondence: soanwar@iupui.edu; Tel.: +1-317-274-7640

Received: 23 October 2018; Accepted: 26 November 2018; Published: 4 January 2019

Abstract: With the ever-increasing usage of lithium-ion batteries, especially in transportation applications, accurate estimation of battery state of charge (SOC) is of paramount importance. A majority of the current SOC estimation methods rely on data collected and calibrated offline, which could lead to inaccuracies in SOC estimation under different operating conditions or when the battery ages. This paper presents a novel real-time SOC estimation of a lithium-ion battery by applying the particle swarm optimization (PSO) method to a detailed electrochemical model of a single cell. This work also optimizes both the single-cell model and PSO algorithm so that the developed algorithm can run on an embedded hardware with reasonable utilization of central processing unit (CPU) and memory resources while estimating the SOC with reasonable accuracy. A modular single-cell electrochemical model, as well as the proposed constrained PSO-based SOC estimation algorithm, was developed in Simulink©, and its performance was theoretically verified in simulation. Experimental data were collected for healthy and aged Li-ion battery cells in order to validate the proposed algorithm. Both simulation and experimental results demonstrate that the developed algorithm is able to accurately estimate the battery SOC for 1C charge and 1C discharge operations for both healthy and aged cells.

Keywords: state of charge; particle swarm optimization; real-time estimation; single-cell model; Simulink©

1. Introduction

Human-generated greenhouse gases are believed to be the primary contributor to the climate change observed in the mid 20th century [1]. The United States (US) Environmental Protection Agency (EPA) estimated that electricity generation and transportation contributed to 56% of greenhouse gases in 2014. The EPA recommends several means of reducing these emissions, one of them being fuel switching. The use of public transport fueled by compressed natural gas (CNG), and the increased use of hybrid and electric vehicles are some of the examples of fuel switching. With this directive, one can expect a dramatic increase in the usage of portable energy sources, such as high-energy-density traction batteries, in the field of transportation. Batteries have become an integral part of our daily lives. They are used in a wide variety of products, ranging from consumer electronic devices, such as cell phones, laptops, etc., to safety critical applications in the field of medicine. The effective utilization and management of battery energy is of paramount importance to improving the efficiency of such energy storage devices.

Battery state of charge (SOC) is one of the most important parameters that must be estimated by the battery management system (BMS). It is comparable to the fuel gauge in an internal combustionengine-powered car. SOC indicates the amount of usable energy left in the battery at a given time. Accurate knowledge of SOC offers the following benefits, to name a few:

- Effective battery operation and management;
- Reliable diagnostics;
- Safety in operation.

Unfortunately, SOC cannot be measured directly in real time and must be estimated using the data of other measurable variables such as voltage, current, battery temperature, etc. A wide range of methods were researched to estimate SOC depending on the application [2]. A number of online estimation techniques were proposed for the estimation of SOC. The Kalman filter method and its variants were one of the important methods used in literature. A series of articles by Plett [3–5] proposed the extended Kalman filter method to estimate battery states including SOC, power fade, capacity fade, and instantaneous available power. Kurzweil and Shamonin investigated SOC monitoring using impedance spectroscopy during long-term self-discharge of supercapacitors and lithium-ion batteries [6]. The main focus of this paper was to study the self-discharge behavior of Li-ion batteries and supercapacitors in the frequency domain. Panchal et al. [7,8] developed a mathematical model to predict the transient temperature distributions of a large-sized $LiFePO_4$ prismatic battery (20 Ah) at different C-rates. Their study showed that the increased C-rates resulted in increased temperatures on the principle surface of the battery.

In this paper, the particle swarm optimization (PSO) method was leveraged as an estimation technique to explore SOC estimation under varying operating conditions of the battery. Another important objective of this work was to demonstrate that the developed battery and estimator models can run on a real-time embedded device with good performance.

2. Battery Model

A wide variety of battery models were proposed in the literature. The appropriateness of a battery model depends on the application and its performance requirements. One of the most popular battery models used is the equivalent circuit model. Here, the battery is modeled using a set of resistors (R), capacitors (C), and other circuit elements to emulate the battery behavior. The accuracy of the model depends on the order of the model reflected by the number of RC elements in the circuit. The higher the order of the model, the greater is the ability of the model to account for different physicochemical processes in the battery operation [9].

Another battery modeling technique is called empirical modeling, which is purely mathematical. Here, the experimentally obtained historical data are used to realize a model where mathematical relationships such as exponentials, polynomials, etc. are developed using the different battery parameters and variables. These models claim to be computationally efficient. However, they must be re-tuned to capture different operating conditions, which means that the data for such scenarios must be available.

Fuzzy logic and neural network techniques were also used to model battery behavior [10–12]. These techniques involve a learning process where the models are trained using experimental data to learn the relationship between the different battery parameters and variables. These models are then used to predict various battery states, such as voltages, SOC, etc. Again, the limitation of such models includes the lack of adaptability as the battery behavior changes over time (aging).

Electrochemical or physics-based models incorporate both the chemical and the electrical behaviors to accurately model a battery. There are a number of such models available with varying degree of complexities. The single-particle model (SPM) models the battery as constituent particles of the active material within the electrodes. An electrochemical battery model considering the diffusion phenomenon of such particles and intercalation within the particle was developed by Zhang et al. [13]. Doyle et al. [14] developed a more comprehensive model based on concentrated solution theory. This model, also known as the pseudo two-dimensional (P2D) model, aims to represent the behavior of a wide range of lithium-ion batteries with different combinations of electrolytes, separators, anodes, and cathodes.

2.1. Battery Chemistry

Lithium (Li)-ion-based chemistry has an unique combination of being lightweight (Li atomic weight of 6.94 g) and high capacity (3.86 Ah/g) making Li-ion chemistry one of the most popular chemistries used, especially in the transportation industry [15]. Lithium cobalt oxide (LCO) chemistry was chosen in this work for its high-energy-density properties. The overall electrochemical reaction in an LCO cell is shown as follows:

$$C + LiCoO_2 \leftrightarrow LiC_6 + Li_{0.5}CoO_2; \tag{1}$$

At the cathode:

$$LiCoO_2 - Li^+ - e^- \leftrightarrow Li_{0.5}CoO_2 \Rightarrow 143 \text{ mAh/g}; \tag{2}$$

At the anode:

$$6C + Li^+ + e^- \leftrightarrow LiC_6 \Rightarrow 372 \text{ mAh/g}. \tag{3}$$

The principal objective of this research was to develop a PSO-based SOC estimation algorithm for Li-ion batteries which would efficiently and seamlessly work under different operating conditions while being computationally efficient enough to run on an embedded device in real time. While the equivalent circuit model can be computationally efficient, it requires re-tuning for numerous operating conditions. The data-driven models (e.g., empirical, fuzzy logic, neural network, etc.) need a lot of experimental data which may not be practical in many cases. The electrochemical models, on the other hand, capture the battery physics well to give a more accurate estimation of the states and the parameters. However, some of these models, such as the one developed by Doyle et al. [14], are not suitable for running in real time due to their complex set of partial differential equations. Other such examples include the model developed by Subramanian et al. [16], derived from first principles as an isothermal pseudo two-dimensional model. A control-oriented one-dimensional (1D) electrochemical model was developed by Smith et al. [17], which is of particular interest due to its real-time computational adaptability. Based on this 1D model, Sourav and Anwar [18] developed and validated a discretized model using the finite difference method (FDM). This model demonstrates both the details of battery physics and the computational efficiency to run in real time. This model was chosen to be used in this research work.

2.2. Constituent Equations

The governing equations of the battery model in [19] were adopted here and are as shown below. This model has six states which are as follows:

Electrolyte lithium concentration $c_e(x, t)$
Electrode lithium concentration $c_s(x, r, t)$ of both anode and cathode
Electrolyte potential $\varphi_e(x, t)$
Electrode potential $\varphi_s(x, t)$ of both anode and cathode
Electrolyte ionic current $i_e(x, t)$
Molar ionic flux $j_n(x, t)$

The constituent equations are given by the following equations [14–20]:

$$\frac{\partial}{\partial t}\bar{c}_s{}^{\pm}(x,t) = -\frac{3}{R_p{}^{\pm}} j_n{}^{\pm}(x,t); \tag{4}$$

$$\varepsilon_e \frac{\partial c_e(x,t)}{\partial t} = \frac{\partial}{\partial x}(\varepsilon_e D_e \frac{\partial c_e(x,t)}{\partial x} + \frac{1 - t_c{}^0}{F} i_e(x,t)); \tag{5}$$

$$\frac{\partial c_{s,i}(x,r,t)}{\partial t} = \frac{1}{r^2}\frac{\partial}{\partial r}(D_{s,i}r^2\frac{\partial c_{s,i}(x,r,t)}{\partial r}). \tag{6}$$

These are further simplified as follows:

$$\frac{\partial}{\partial t}\bar{c}_s{}^{\pm}(t) = -\frac{3}{R_p{}^{\pm}}j_n{}^{\pm}(t); \tag{7}$$

$$\frac{\partial \varphi_e(x,t)}{\partial x} = -\frac{i_e(x,t)}{\kappa} + \frac{2RT}{F}(1-t_c{}^0)(1+\frac{d\ln f_{\frac{c}{a}}}{d\ln c_e(x,t)})\frac{\partial c_e(x,t)}{\partial x}; \tag{8}$$

$$\frac{\partial \varphi_s(x,t)}{\partial x} = \frac{i_e(x,t)-I(t)}{\sigma}; \tag{9}$$

$$\frac{\partial i_e(x,t)}{\partial x} = \frac{3\varepsilon_s}{R_p}Fj_n(x,t); \tag{10}$$

$$j_n(x,t) = \frac{i_0(x,t)}{F}(e^{\frac{\alpha_a F\eta\ (x,t)}{RT}} - e^{\frac{-\alpha_C F\eta\ (x,t)}{RT}}). \tag{11}$$

In the last equation above, the exchange current density $i_0(x,t)$ and the over-potential $\eta(x,t)$ for the main reaction are modeled as follows:

$$i_0(x,t) = r_{eff}c_e(x,t)^{\alpha_a}(c_s{}^{\max} - c_{ss}(x,t))^{\alpha_c}c_{ss}(x,t)^{\alpha_C}; \tag{12}$$

$$\eta(x,t) = \phi_s(x,t) - \phi_e(x,t) - U(c_{ss}(x,t)) - FR_f j_n(x,t); \tag{13}$$

where $c_{ss}(x,t) \approx c_s(x,R_p,t)$, $U(c_{ss}(x,t))$ is the open-circuit potential of the active materials, and c_s^{max} is the maximum concentration in the active material of each electrode. The internal temperature of the battery is described by

$$\rho^{avg}c_p\frac{dT(t)}{dt} = h_{cell}(T_{amb}-T(t)) + I(t)V(t) - \int_{0^-}^{0^+}\frac{3\varepsilon_s}{R_p}Fj_n(x,t)\Delta U(x,t)dx, \tag{14}$$

where

$$\Delta U(x,t) \triangleq U(\bar{c}_s(x,t)) - T(t)\frac{\partial U(\bar{c}_s(x,t))}{\partial T},$$

where T_{amb} is the ambient temperature, and $\overline{c_s}(x,t)$ is the volume-averaged concentration of a particle in the solid phase which is defined as

$$\bar{c}_s(x,t) \triangleq \frac{3}{R_p{}^3}\int_0^{R_{p,i}} r^2 c_s(x,r,t)dr. \tag{15}$$

The initial conditions of the battery model are given by

$$c_s{}^{\pm}(x,r,0) = \bar{c}_{s,0}{}^{\pm}(x,r),\ c_e{}^{\pm}(x,0) = \bar{c}_{e,0}{}^{\pm}(x),\ \text{and } T(0) = T_0, \tag{16}$$

and the boundary conditions are given by the following equations:

$$\frac{\partial c_e(0^-,t)}{\partial x} = \frac{\partial c_e(0^+,t)}{\partial x} = 0; \tag{17}$$

$$c_e(L^-,t) = c_e(0^{sep},t); c_e(L^{sep},t) = c_e(L^+,t); \tag{18}$$

$$\varepsilon_e^- D_e\frac{\partial c_e(L^-,t)}{\partial x} = \varepsilon_e^{sep}D_e\frac{\partial c_e(0^{sep},t)}{\partial x}; \tag{19}$$

$$\varepsilon_e^{sep} D_e \frac{\partial c_e(L^{sep},t)}{\partial x} = -\varepsilon_e^{+} D_e \frac{\partial c_e(L^{+},t)}{\partial x}; \tag{20}$$

$$\frac{\partial c_s(x,0,t)}{\partial r} = 0; \frac{\partial c_s(x,R_p,t)}{\partial r} = -\frac{j_n(x,t)}{D_s}; \tag{21}$$

$$\phi_e(L^-,t) = \phi_e(0^{sep},t); \phi_e(L^{sep},t) = \phi_e(L^+,t); \phi_e(0^+,t) = 0; \tag{22}$$

$$i_e(0^-,t) = i_e(0^+,t) = 0; i_e(x^{sep},t) = -I(t); \tag{23}$$

$$i_e(L^-,t) = -i_e(L^+,t) = -I(t); \tag{24}$$

where $x^{sep} \in [0^{sep}, L^{sep}]$ represents the entire separator domain of the battery.

In the above equations ε_e, ε_s, σ, R, R_p, F, α_a, α_c, ρ_{avg}, c_p, h_{cell}, and t_c^0 are model parameters and are constant in each region of the cell. κ, fc/a, and D_e are known functions of the electrolyte concentration.

The terminal voltage is given by

$$V(t) = \varphi_s(0^+,t) - \varphi_s(0^-,t). \tag{25}$$

This battery has a nominal capacity of 3.5 Ah. The SOC is directly calculated from the lithium-ion concentration as shown below.

$$SOC = \frac{\overline{C_s}}{C_{smax}}, \tag{26}$$

where

$$\overline{C_s} = \frac{\sum_{i=1}^{N} C_s(i,t)}{N}, \tag{27}$$

where N is the number of nodes in the electrode.

The lithium-ion concentration of the negative electrode was used to estimate the SOC. It was observed that the lithium-ion concentration in the negative electrode demonstrated the range and direction of the actual SOC, meaning that the SOC increases with charge and decreases with discharge. The lithium-ion concentration of the positive electrode lacks the range and exhibits reverse directional characteristics.

2.3. Reformulation of Battery Model

The two main goals of this work were the following:

- Application of PSO in SOC estimation for improved accuracy under different operating conditions;
- Ability to run the developed estimation algorithm in real time on an embedded target.

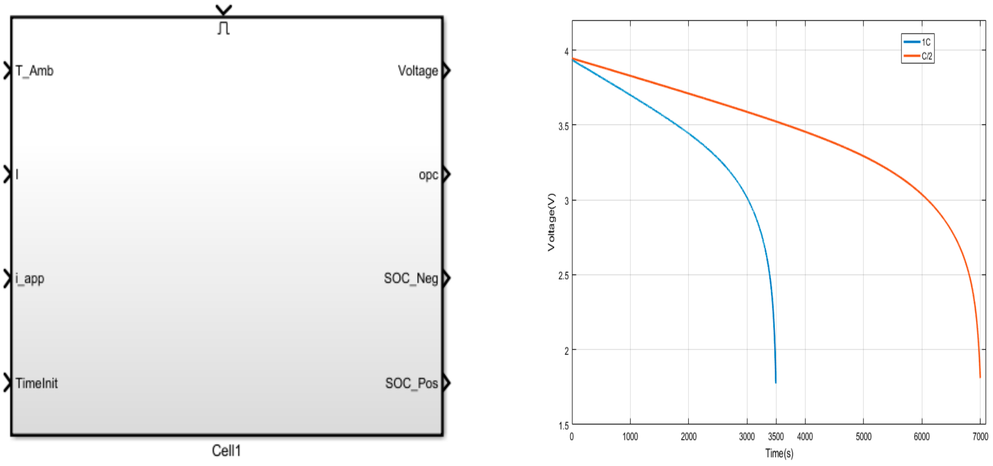
To achieve the above goals, the model needed to be re-formulated. Such a reformulation and the requirement for each of the above goals are discussed here.

A search-based optimization algorithm, such as PSO, evaluates the objective function by varying a set of tunable parameters to arrive at the optimal solution. This process could be repeated at every time step in a discrete time implementation. In the case of a battery model, it would have to be run with a set number values for a given state, evaluating the fitness function for each of these values and comparing to the given optimization criteria. This requirement demands the battery model to be modular, where it can be run repeatedly within a given time step.

The equations for the positive and negative electrodes, i.e., Equations (7), (9), and (11)–(13), are similar. Hence, all these equations can grouped together into a function. Similarly, the equations for the electrolyte and battery temperature can be grouped into their respective functions. Also, to run the battery model on an embedded target, it is preferred to be written in a high-level language, such as C or C++, where the code can be compiled into real-time executables.

Matlab©/Simulink© offers a platform where the modular model can not only be developed, but also simulated along with analysis of the results. Additionally, it offers rapid prototyping on an embedded hardware, such as Raspberry Pi. Once the modular battery model and PSO algorithm are developed and validated in Simulink©, its auto code generator can produce the real-time code that is capable of running on a target-embedded device.

The top-level modular Simulink implementation of a single-cell model is shown in Figure 1a. which is based on Equations (4) through (27) with the suggested grouping as stated above. It takes the ambient temperature (T_{Amb}) and the measured current (I) as inputs. Terminal voltage, open-circuit voltage (OCV), state of charge at the negative electrode (SOC_Neg), and state of charge of positive electrode (SOC_Pos) are the model outputs. The states are calculated and updated internally. This modular approach makes the Model efficient for prototyping and analysis.



(**a**) (**b**)

Figure 1. (**a**) Single-cell battery model; (**b**) discharging simulation of the single-cell battery model.

2.4. Theoretical Simulation Validation

Charge and discharge rates are often expressed in C-rate, defined as the charge or discharge current divided by the battery's capacity to store an electrical charge. For example, for a battery with a capacity of 1000 mAh, a discharge rate of 1500 mA corresponds to a C-rate of 1.5 (per hour). This model was subjected to standard discharge rates of 1C (30 A/m^2) and 0.5C (15 A/m^2) in order to verify the standard battery operation via simulation. The battery parameters used in this work are listed in the Table 1. The output voltage response shown in Figure 1b indicates that, at 1C, the battery discharges completely in 3500 s, while, at 0.5C, it takes 7000 s to completely discharge. This is in tune with the expected response of a battery with a nominal capacity of 3.5 Ah.

Table 1. Battery parameters [16].

Parameter	Units	Negative Electrode	Separator	Positive Electrode
L	m	88×10^{-6}	35×10^{-6}	80×10^{-6}
N	-	88	35	80
R_i	m	2×10^{-6}	-	2×10^{-6}
E_s	-	0.4824	-	0.5
E_e	-	0.485	0.724	0.385
σ	$\Omega^{-1}\mathrm{m}^{-1}$	100	-	100
t_+^0	-	0.363	0.363	0.363
$C_{s_{max,i}}$	$\mathrm{mol\,m}^{-3}$	30,555	-	51,554
$c_{e,i}$	$\mathrm{mol\,m}^{-3}$	1000	1000	1000
α_a, α_c	-	0.5, 0.5	-	0.5, 0.5
R	$\mathrm{jmol}^{-1}\mathrm{K}^{-1}$	8.314	8.314	8.314
F	$\mathrm{C\,mol}^{-1}$	96,487	96,487	96,487
A	m^2	30	-	30

3. Particle Swarm Optimization

The particle swarm optimization (PSO) is a population-based stochastic optimization technique. It was invented by Dr. Kennedy and Dr. Eberhart in 1995. Similar to the movement of a flock of birds, the algorithm mimics the flock's flight pattern for the optimal path to locate food. PSO has several advantages over genetic algorithm (GA), such as needing fewer parameters to tune, being computationally efficient, and having a higher degree of convergence.

PSO is used in a variety of industries for a wide range of applications, such as robot control optimization [21], optimization of global solar radiation estimation for regions where they cannot be measured [22], detection of ovarian cancer [23], and optimization of machine loads [24]. In the automotive applications, one of the most popular uses of PSO is in battery applications with respect to identifying battery parameters [25–27]. Another popular application is in optimal energy management for hybrid electric powertrains [28–30].

Ismail [31] presented a PSO-based SOC estimation of Li-ion batteries utilizing an equivalent-circuit-type battery model to estimate SOC of a $LiFePO_4$ battery. This paper focuses on PSO-based SOC estimation using the electrochemical model of a single cell of a battery with lithium cobalt oxide (LCO) cathode chemistry.

4. PSO Algorithm

In a global PSO algorithm, which is used in this work, a swarm of solutions, called particles, are initialized randomly in the solution space. These particles are then flown through the solution space based on the optimality conditions. The particles keep track of their coordinates with respect to the best solution in the problem space. This is termed as *pbest* which is a particle's best position in solution space. Similarly, a global best, termed as *gbest*, is tracked, which is the best location or solution obtained by any particle so far. At every time step, the velocities of the particles are updated toward the *pbest* and the *gbest* values until the required optimization criteria are met or the algorithm termination condition is reached.

An outline of the PSO algorithm is presented below.

Step 1: Initialize the particles

Initialize the position array with random numbers having uniform distribution.

$$X = Urand\left(r_{lowerlim}, r_{upperlim}\right). \tag{28}$$

Assign this initial position to best known position array.

$$P = X. \tag{29}$$

Initialize particle velocity.

$$V = X. \tag{30}$$

If the number of particles is Num_p then, X is a Num_p-size array of particle position; similarly, P is a Num_p-size array of *pbest* positions, and V is a Num_p-size array of particle velocities.

Step 2: Evaluate the optimization fitness function.

$$Ex = F(X)\ and\ Ep = F(P)\ and\ eg = f\ (gbest), \tag{31}$$

where Ex and Ep are the fitness evaluation array for X and P, respectively, and eg is the function evaluation at *gbest*.

Step 3: Update *pbest* value for each particle of the population.

$$\text{if } Ex(i) < Ep(i) \text{ then } P(i) = X(i), \tag{32}$$

where $i = 1, 2, \ldots$ *Nump*.

Step 4: Update *gbest* value for the entire population.

$$\text{if } Ep(i) < eg \text{ then } gbest = P(i). \tag{33}$$

Step 5: Update the velocity and position of the particles.

$$V(i+1) = wV(i) + c_1 u_{rand}(0,1)(P(i) - X(i)) + c_2 u_{rand}(0,1)(gbest - X(i)), \tag{34}$$

$$X(i+1) = X(i) + V(i+1), \tag{35}$$

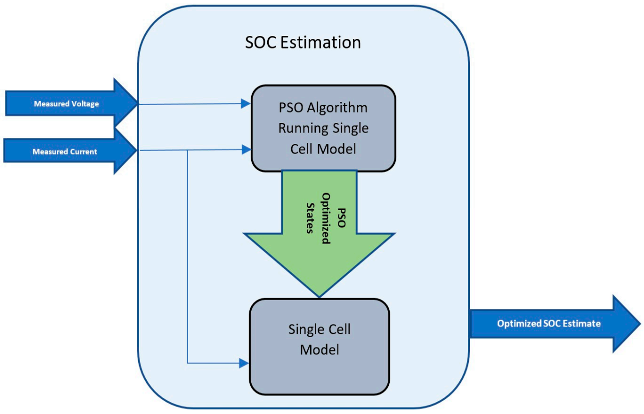
where w is the inertial weight, c_1 is the cognitive parameter, and c_2 is the social parameter.

Step 6: If criteria are met, then exit; otherwise, loop to Step 2. Exit criteria are usually a fitness threshold or maximum number of iterations completed.

5. SOC Estimation Approach

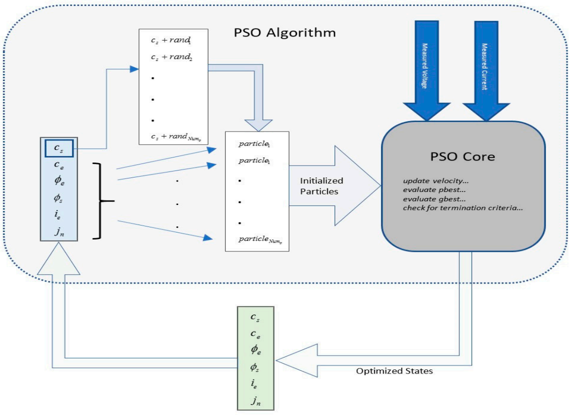
An overview of the approach of SOC estimation using PSO is show in Figure 2. The proposed algorithm takes measured voltage and current as inputs, and outputs the estimated SOC. The approach is to optimize the states with the goal of minimizing the error between the measured voltage and estimated voltage. Once the optimized states are obtained, these states are used again in the single-cell model to estimate the SOC. Although the battery model has six states, the scope of this work is limited to optimizing one state, which is $C_s(x, r, t)$, the lithium-ion concentration in the negative electrode. This is done to keep the algorithm simple and efficient. The rationale behind choosing C_s is that the lithium-ion concentration in the electrode intuitively represents the SOC of the cell, which is also directly proportional to the open-circuit voltage. This means that the C_s shall be used as the particles of the PSO algorithm which are initialized to uniformly random values around the previous C_s, and the terminal voltage is estimated and optimized based on these states for the measured current value. A pictorial view of this process is shown in Figure 2. According to the PSO algorithm, the initial value of the particles can be randomly initialized; however, for faster convergence, it is proposed that the best known value that would represent the current battery condition be used. A linear interpolation function is applied where the measured voltage and current are inputs, and the stoichiometric value of the concentration is the output. This value is then used in the calculation of C_s as shown below. This is done only once at the beginning of the algorithm. For subsequent time steps, the optimized states, including C_s, from the previous step are fed back as shown in Equation (36).

$$C_s = \textit{stoichiometric value} \times C_s \textit{max}, i. \tag{36}$$



(a)

Figure 2. *Cont.*



(**b**)

Figure 2. (**a**) State of charge (SOC) estimation overview; (**b**) schematic of the particle swarm optimization (PSO)-based SOC estimation algorithm.

O Algorithm Parameters

From Equation (34), it is clear that the PSO algorithm uses only three parameters, c_1 and c_2, which are learning coefficients of the stochastic acceleration terms, and w, which is the inertial coefficient.

The values of the PSO parameters w, c_1, and c_2, as well as the number of particles and maximum iterations, chosen for this work are shown in the Table 2. A constriction factor introduced by Clerc [32] was used to determine c_1 and c_2. Additionally, a method proposed by Eberhart and Shi [33] for application in dynamic systems was adopted to determine the inertial weight w.

Table 2. Particle swarm optimization (PSO) parameters.

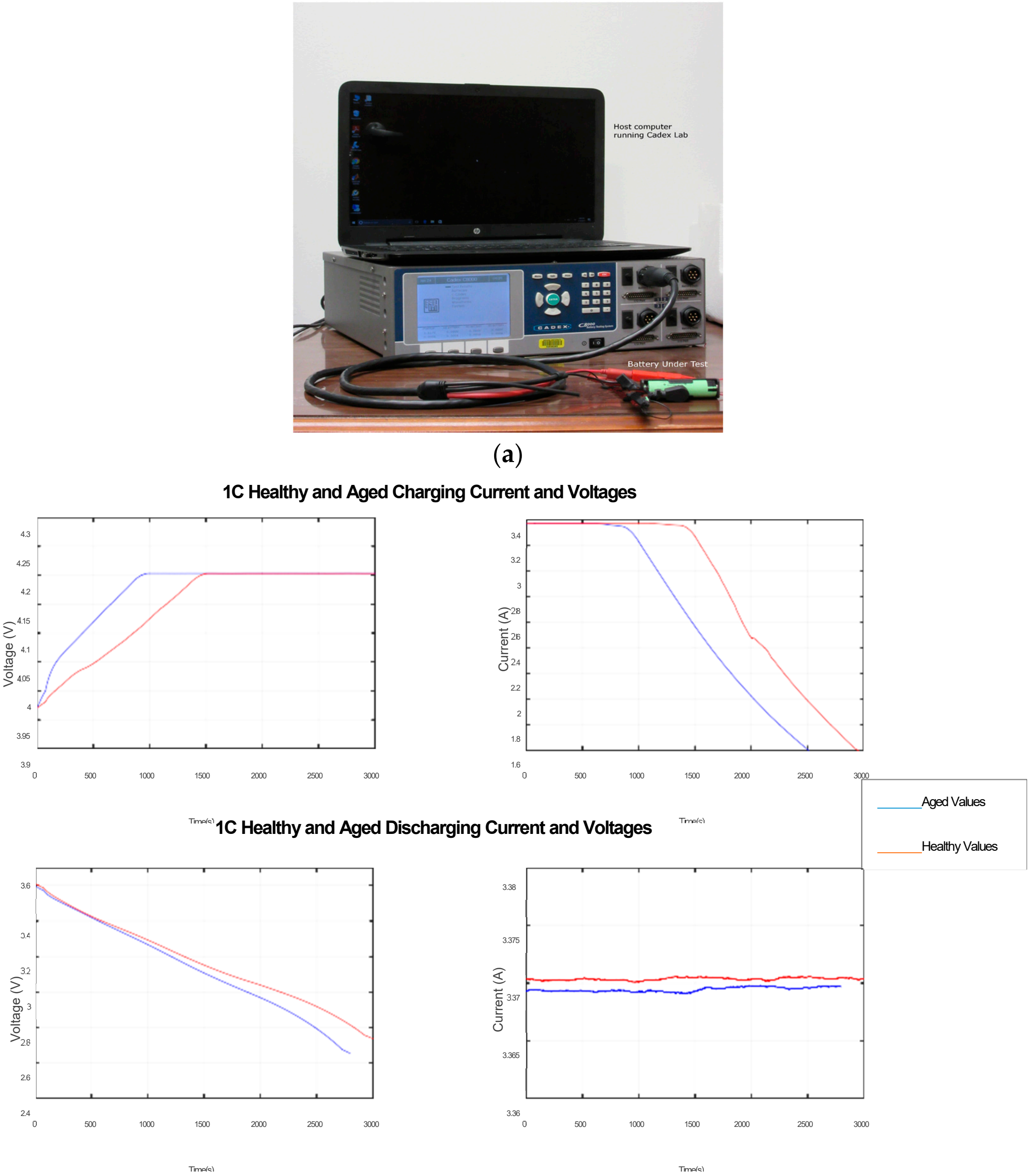
Parameter	Value
w	$(0.5 + u_{rand}/2)$
c_1	1.494
c_2	1.494
Number of particles	8
Maximum number of iterations	20

6. Algorithm Validation and Results

6.1. Experimental Set-Up

The battery used was a Panasonic NCR 18650B with $LiCoO_2$ (LCO) cathode chemistry. The details of the battery are as follows: rated capacity, 3200 mAh (at nominal voltage); nominal voltage, 3.6 V. Experimental data were collected by subjecting this LCO battery to various test cycles. Cadex C8000 was the test equipment used for this experiment. The set-up is shown in Figure 3a. The battery was subjected to constant current charging and discharging at 1C. This cycle was repeated for both new and aged batteries. A scheme of 100-cycle aging was used to age the new battery. A built-in profile of NCR18650B in the CADEX software was used for both charging and discharging profiles. The discharging was done at a constant current of 3.3 A. The data were logged at 1-Hz sampling rate.

Figure 3b shows the charging and discharging voltage and current of both healthy and aged batteries. These graphs clearly show the degradation of battery performance with aging.



(**a**)

(**b**)

Figure 3. (**a**) CADEX C8000 experimental set-up; (**b**) experimental aged and healthy charging voltage and current.

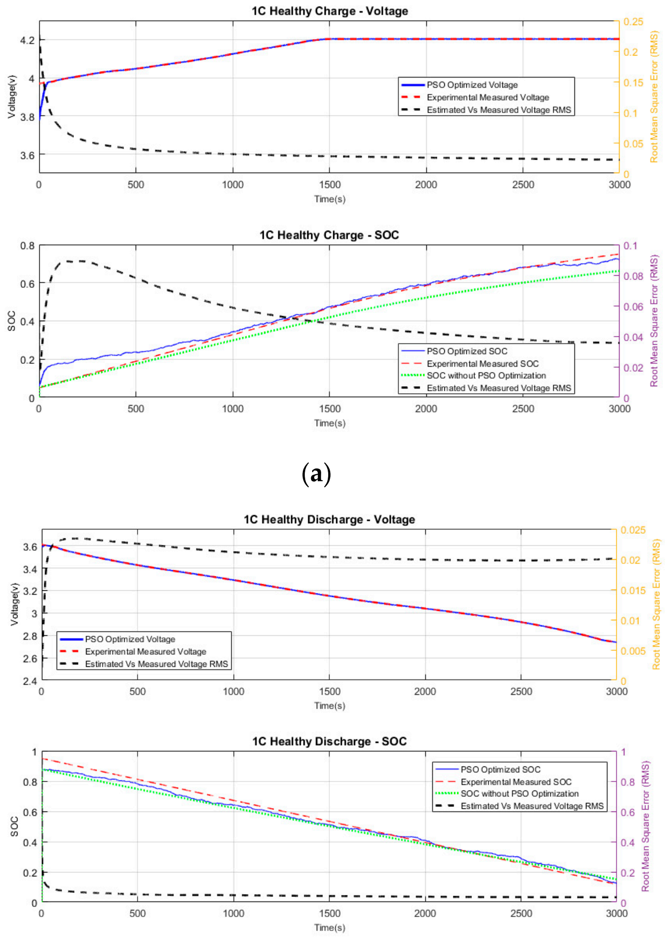
6.2. Simulation Validation

In order to validate the proposed algorithm, the experimental data were run on the Simulink model. The model was set up with a discrete solver with a step time of 200 ms. The model was set to "normal mode" in order to verify the algorithm in simulation. Root-mean-square error (RMS) as per Equation (37) was used to measure the performance of the proposed algorithm, where *xi* is the difference between the measured and PSO estimated states. The simulation data for 1C charging and discharging for both healthy and aged batteries are presented in this section.

$$RMS = \sqrt{\frac{\sum_{i=1}^{M} |x_i|^2}{M}}. \tag{37}$$

6.2.1. Healthy Charging

Figure 4a shows the measured terminal voltage against the PSO-estimated voltage in 1C charging. Although the measured and the estimated voltages start with different values, the PSO algorithm quickly converges to the measured voltage in less than 45 s. The average RMS error between the estimated and measured voltages is 0.0221. The graph of running RMS of PSO voltage and measured voltage is shown in Figure 4a (top plot).



(**a**)

(**b**)

Figure 4. Healthy battery experimental vs. PSO voltage: (**a**) charging; (**b**) discharging.

Figure 4a (bottom plot) compares the experimental SOC with the PSO-estimated SOC. The experimental SOC value is directly obtained from the CADEX battery tester. The SOC values directly computed from the single-cell battery model (without PSO) are also plotted to provide a reference for analysis purposes. The estimated SOC converges within 2% of the experimental SOC in the first 800 s, and then continues to follow the experimental SOC with a running RMS of 0.0355, as shown in Figure 4a. It is clear from the graph that the computed SOC without PSO diverges from the experimental SOC.

6.2.2. Healthy Discharging

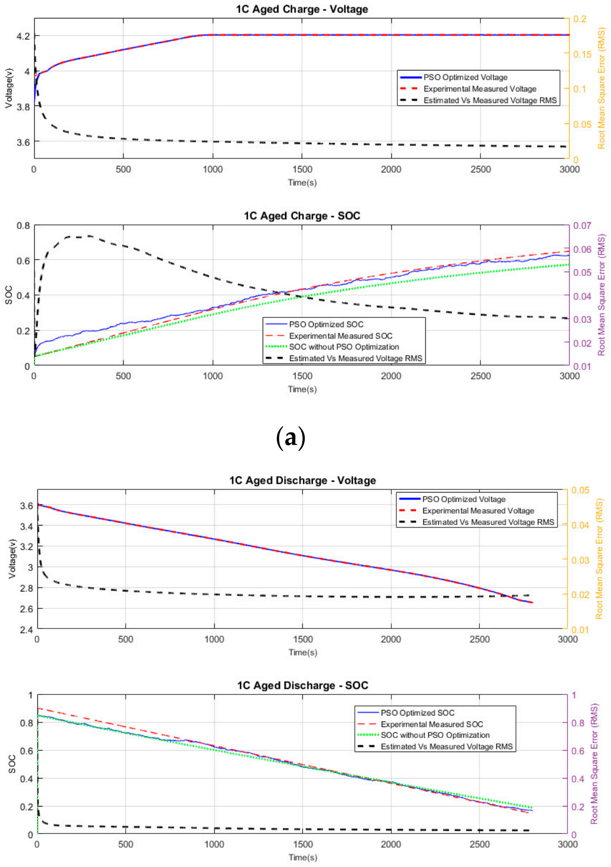
Figure 4b shows the measured voltage against the PSO-estimated voltage for discharging. The measured and the estimated voltages start with different values, but the PSO algorithm quickly converges to the experimental voltage in less than 15 s. The short convergence time can be attributed to the initial value of PSO voltage being close to measured voltage, with a 0.025-V voltage difference.

The average RMS error between the estimated and measured voltages is 0.0200. The running RMS of PSO-estimated and measured voltages is shown in Figure 4b (top plot).

Figure 4b (bottom plot) shows the experimental SOC against the PSO-estimated SOC. The computed SOC from the single-cell battery model (without optimization) is also plotted to provide a reference for analysis purposes. The estimated SOC converges within 2% of the experimental SOC within the first 1160 s, and then continues to follow the experimental SOC with a running RMS of 0.03186, as shown in Figure 4b. The computed SOC of the single cell, the estimated SOC with PSO, and the experimental SOC converge to within 3% absolute difference in 1500 s.

6.2.3. Aged Charging

Figure 5a illustrates the measured voltage against the PSO-estimated voltage for charging. Although the measured and the estimated voltages start with different values, the PSO algorithm converges to the measured voltage in less than 30 s. The average RMS error between the estimated and measured voltages is 0.01729. The running RMS of PSO-estimated voltage and measured voltage is shown in Figure 5a (top plot).



(**a**)

(**b**)

Figure 5. Aged battery experimental vs. PSO voltage: (**a**) 1C charging; (**b**) 1C discharging.

Figure 5a (bottom plot) illustrates the experimental SOC against the PSO-estimated SOC. The computed SOC from the single-cell battery model (without optimization) is also plotted to provide a reference for analysis purposes. Similar to the healthy charging, the estimated SOC converges within an absolute difference of less than 2% within the first 800 s, and then continues to follow the experimental SOC with a running RMS of 0.03016, as shown in Figure 5a. As in the case of the healthy battery, it is clear from the graph that the computed SOC without PSO diverges from the experimental SOC.

6.2.4. Aged Discharging

Figure 5b shows the measured voltage against the PSO-estimated voltage for discharging. The measured and the estimated voltages start with different values, but the PSO algorithm converges to the measured voltage in less than 12 s. The short convergence time is due to the initial value of the PSO voltage being closer to measured voltage, with an absolute difference of 0.0212 V for the given conditions. The average RMS error between the estimated and measured voltages is 0.0196. The running RMS of PSO voltage and measured voltage is shown in Figure 5b (top plot).

Figure 5b (bottom plot) compares the experimental SOC against the PSO-estimated SOC. The computed SOC from the single-cell battery model (without optimization) is also plotted to provide a reference for analysis purposes. The estimated SOC converges within an absolute difference of less than 1% within the first 830 s, and then continues to follow the experimental SOC with a running RMS of 0.02515 as shown in Figure 5b. The computed SOC of the single-cell model without optimization, the estimated SOC with PSO, and the experimental SOC converge to within 3% absolute difference within 1500 s. However, it is clear that the PSO-estimated SOC closely follows the experimentally calculated SOC.

The following observations were made:

- The SOC estimation accuracy for charging of a healthy battery was within 0.0355 or 3.55%;
- The SOC estimation accuracy for discharging of a healthy battery was within 0.03186 or 3.19%;
- The SOC estimation accuracy for charging of an aged battery was within 0.03016 or 3.02%;
- The SOC estimation accuracy for discharging of an aged battery was within 0.0251 or 2.51%.

6.3. Real-Time Validation

A real-time version of the PSO estimator was run on a Raspberry Pi 3 embedded hardware using the external mode in Simulink. Raspberry Pi 3 is a single-board computer with a 64-bit quad core processor and 1 GB internal random-access memory (RAM). Being a low-cost device, it is ideal for prototyping computationally intensive algorithms like the PSO estimator presented in this work. Simulink provides a seamless interface to Raspberry Pi hardware. Once the models are validated in pure simulation mode, the same models can be run on Raspberry Pi by changing the simulation mode to "external mode" in the model. Figure 6 shows a picture of the Raspberry Pi 3 set-up.

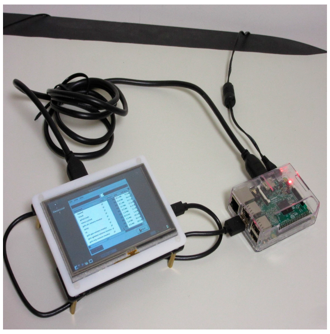


Figure 6. Raspberry Pi 3 experimental set-up.

Figures 7 and 8 show the comparisons of voltage and SOC obtained from simulation and from the real-time processor (Raspberry Pi 3) for charging and discharging, respectively. The results from simulation and real-time hardware match completely, which shows that the PSO estimator is efficient on the real-time processor, offering the exact same results. The central processing unit (CPU) utilization time of Raspberry Pi was collected to evaluate the performance. A maximum of 26% CPU utilization was observed at PSO estimator initialization. During convergence, it ran at 3% average utilization for the rest of the execution.

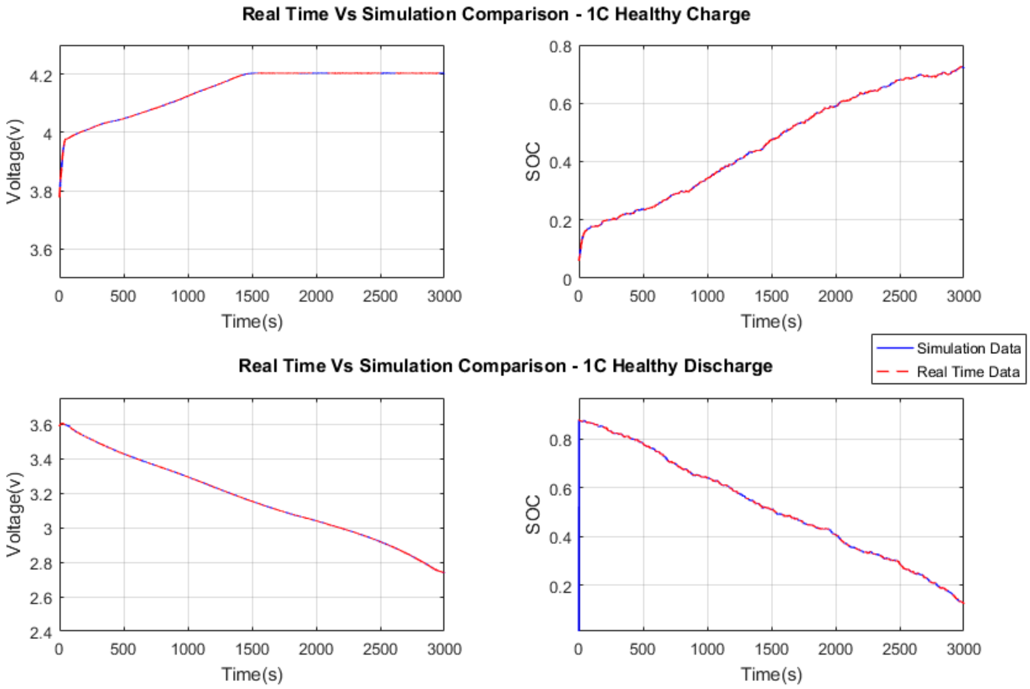


Figure 7. Simulated and real-time voltages and SOCs for the healthy battery.

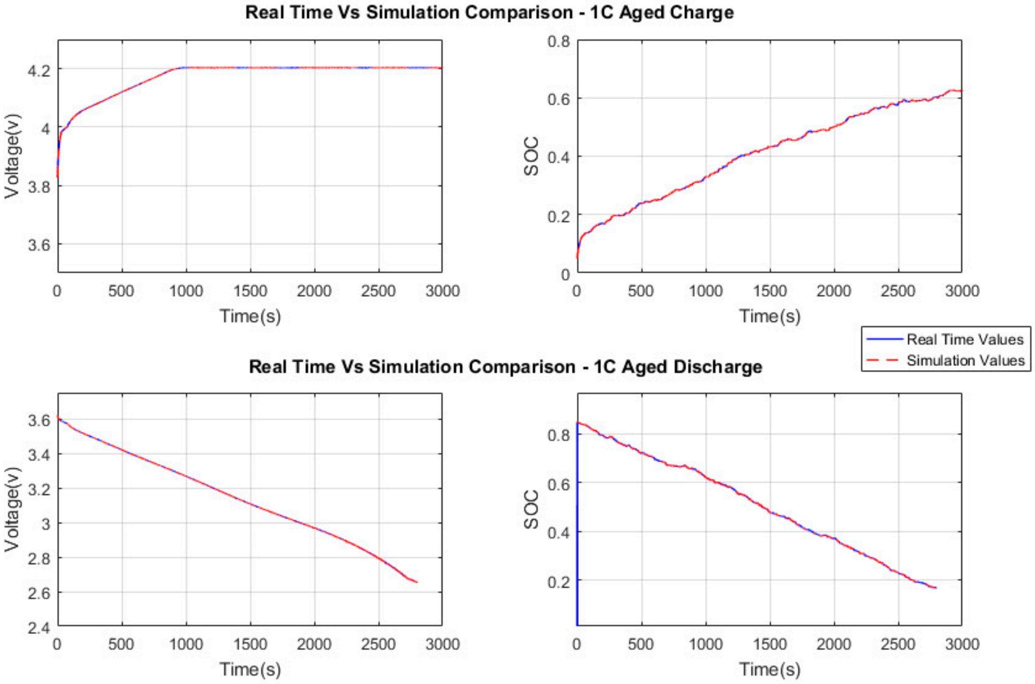


Figure 8. Simulated and real-time voltages and SOCs for the aged battery.

7. Conclusions

The main goal of this work was to develop a PSO algorithm based on an electrochemical battery model to accurately estimate the SOC, which would also be capable of running in real time. It was demonstrated that the proposed particle swarm optimization (PSO)-based estimation algorithm developed in the work was able to accurately estimate of state of charge (SOC) based on an electrochemical battery model of a lithium-ion cell. The results show a close agreement between the PSO-estimated and experimental SOC values for both healthy and aged battery cells under charging and discharging conditions. It was demonstrated that the proposed real-time implementation of the

PSO estimation algorithm is capable of effectively running on Raspberry Pi 3 embedded hardware in real time with good performance. It is to be noted that initial conditions need to be closer to the actual value while starting the estimation to ensure faster convergence. Also, the single-cell battery models were not optimized against the experimental data.

Author Contributions: Conceptualization, S.A.; methodology, S.A. and A.C.S.; software, A.C.S.; validation, A.C.S.; investigation, S.A.; resources, S.A.; Writing—Original Draft preparation, A.C.S.; Writing—Review and Editing, S.A.; supervision, S.A.; project administration, S.A.; funding acquisition, S.A.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

c_e—Lithium ion concentration in the electrolyte phase

c_s—Lithium ion concentration in the active materials in both electrodes

$\bar{c}_{s,i}$—Volume-averaged concentration of a single particle

D_e—Diffusivity at electrolyte phase

D_s—Diffusivity at solid phase

$f_{c/a}$—Mean molar activity coefficient

F—Faraday constant

i_e—Current in the electrolyte phase

i_0—Exchange current density

I—Load current

j_n—Molar ion fluxes between the active materials in electrodes and the electrolyte

L^-—Length of negative electrode

L^+—Length of positive electrode

N—Number of nodes in the electrode

R—Universal gas constant

R_p—Radius of the spherical particles

${t_c}^0$—Transference number

T—Average internal temperature

U—Open-circuit potential

V—Cell voltage

$_a$—Charge-transfer coefficient in anode

$_c$—Charge-transfer coefficient in cathode

γ—Observer gain constant

$_e$—Potential at electrolyte phase

$_s$—Potential at solid phase

$_e$—Volume fraction at electrolyte phase

$_s$—Volume fraction at solid phase

η—Over-potential for the reactions

ρ^{avg}—Average density

k—Intercalation/de-intercalation reaction rate

References

1. Climate Change Indicators. Available online: https://www.epa.gov/climate-indicators (accessed on 17 November 2018).
2. Moo, C.S.; Ng, K.S.; Chen, Y.P.; Hsieh, Y.C. State-of-charge estimation with open-circuit-voltage for lead-acid batteries. In Proceedings of the IEEE Power Conversion Conference-Nagoya, PCC'07, Nagoya, Japan, 2–5 April 2007; pp. 758–762.
3. Plett, G.L. Extended kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background. *J. Power Sources* **2004**, *134*, 252–261. [CrossRef]

4. Plett, G.L. Extended kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification. *J. Power Sources* **2004**, *134*, 262–276. [CrossRef]
5. Plett, G.L. Extended kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *J. Power Sources* **2004**, *134*, 277–292. [CrossRef]
6. Kurzweil, P.; Shamonin, M. State-of-charge monitoring by impedance spectroscopy during long-term self-discharge of supercapacitors and Lithium-Ion batteries. *Batteries* **2018**, *4*, 35. [CrossRef]
7. Panchal, S.; Mathew, M.; Fraser, R.; Fowler, M. Electrochemical thermal modeling and experimental measurements of 18650 cylindrical lithium-ion battery during discharge cycle for an EV. *Appl. Therm. Eng.* **2018**, *135*, 123–132. [CrossRef]
8. Panchal, S.; Dincer, I.; Agelin-Chaab, M.; Fraser, R.; Fowler, M. Transient electrochemical heat transfer modeling and experimental validation of a large sized LiFePO4/graphite battery. *Int. J. Heat Mass Transf.* **2017**, *109*, 1239–1251. [CrossRef]
9. Paganelli, G.; Guezennec, Y.G.; Kim, H.; Brahma, A. Battery dynamic modeling and real-time state-of-charge estimation in hybrid electric vehicle application. In Proceedings of the ASME International Mechanical Engineering Congress and Exposition, New York, NY, USA, 11–16 November 2001; Volume 2, pp. 1101–1107.
10. Salkind, A.J.; Fennie, C.; Singh, P.; Atwater, T.; Reisner, D.E. Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology. *J. Power Sources* **1999**, *80*, 293–300. [CrossRef]
11. Bhatikar, S.R.; Mahajan, R.L.; Wipke, K.; Johnson, V. *Artificial Neural Network Based Energy Storage System Modeling for Hybrid Electric Vehicles*; SAE Technical Paper No. 2000-01-1564; SAE Future Car Congress: Arlington, VA, USA, 2000.
12. Singh, P.; Reisner, D. Fuzzy logic-based state-of-health determination of lead acid batteries. In Proceedings of the International Telecommunications Energy Conference, Montreal, QC, Canada, 29 September–3 October 2002; pp. 583–590.
13. Zhang, D.; Popov, B.N.; White, R.E. Modeling lithium intercalation of a single spinel particle under potentiodynamic control. *J. Electrochem. Soc.* **2000**, *147*, 831–838. [CrossRef]
14. Doyle, M.; Fuller, T.F.; Newman, J. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *J. Electrochem. Soc.* **1993**, *140*, 1526–1533. [CrossRef]
15. Reddy, T.B. (Ed.) *Linden's Handbook of Batteries*, 4th ed.; McGraw-Hill Education: New York, NY, USA, 2010.
16. Subramanian, V.R.; Boovaragavan, V.; Ramadesigan, V.; Arabandi, M. Mathematical model reformulation for lithium-ion battery simulations: Galvanostatic boundary conditions. *J. Electrochem. Soc.* **2009**, *156*, 260–271. [CrossRef]
17. Smith, K.A.; Rahn, C.D.; Wang, C.-Y. Control oriented 1d electrochemical model of lithium ion battery. *Energy Convers. Manag.* **2007**, *48*, 2565–2578. [CrossRef]
18. Pramanik, S.; Anwar, S. Electrochemical model based charge optimization for lithium-ion batteries. *J. Power Sources* **2016**, *313*, 164–177. [CrossRef]
19. Klein, R.; Chaturvedi, N.A.; Christensen, J.; Ahmed, J.; Findeisen, R.; Kojic, A. Electrochemical model based observer design for a lithium-ion battery. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 289–301. [CrossRef]
20. Klein, R.; Chaturvedi, N.A.; Christensen, J.; Ahmed, J.; Findeisen, R.; Kojic, A. Optimal charging strategies in lithium-ion battery. In Proceedings of the American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 382–387.
21. Das, M.T.; Dulger, L.C.; Das, G.S. Robotic applications with particle swarm optimization (pso). In Proceedings of the 2013 International Conference on Control, Decision and Information Technologies (CoDIT), Hammamet, Tunisia, 6–8 May 2013.
22. Mohandes, M.A. Modeling global solar radiation using particle swarm optimization (pso). *Solar Energy* **2012**, *86*, 3137–3145. [CrossRef]
23. Meng, Y. A swarm intelligence based algorithm for proteomic pattern detection of ovarian cancer. In Proceedings of the 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, Toronto, ON, Canada, 28–29 September 2006.
24. Zhao, F.; Zhu, A.; Yu, D.; Yang, Y. A hybrid particle swarm optimization(pso) algorithm schemes for integrated process planning and production scheduling. In Proceedings of the World Congress on Intelligent Control and Automation (WCICA), Dalian, China, 21–23 June 2006.

25. Rahman, M.A.; Anwar, S.; Izadian, A. Electrochemical model based fault diagnosis of a lithium ion battery using multiple model adaptive estimation approach. In Proceedings of the IEEE International Conference on Industrial Technology, Seville, Spain, 17–19 March 2015.
26. Afshari, H.H.; Attari, M.; Ahmed, R.; Farag, M.; Habibi, S. Modeling, parameterization, and state of charge estimation of li-ion cells using a circuit model. In Proceedings of the 2016 IEEE Transportation Electrification Conference and Expo (ITEC), Dearborn, MI, USA, 27–29 June 2016.
27. Kim, T.; Qiao, W.; Qu, L. Real-time state of charge and electrical impedance estimation for lithium-ion batteries based on a hybrid battery model. In Proceedings of the 2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC), Long Beach, CA, USA, 17–21 March 2013.
28. Samanta, C.K.; Padhy, S.K.; Panigrahi, S.P.; Panigrahi, B.K. Hybrid swarm intelligence methods for energy management in hybrid electric vehicles. *IET Electr. Syst. Transp.* **2013**, *3*, 22–29. [CrossRef]
29. Chen, Z.; Xiong, R.; Wang, K.; Jiao, B. Optimal energy management strategy of a plug-in hybrid electric vehicle based on a particle swarm optimization algorithm. *Energies* **2015**, *8*, 3661–3678. [CrossRef]
30. Chen, Z.; Xiong, R.; Wang, C.; Cao, J. An on-line predictive energy management strategy for plug-in hybrid electric vehicles to counter the uncertain prediction of the driving cycle. *Appl. Energy* **2017**, *185*, 1663–1672. [CrossRef]
31. Ismail, N.H.F.; Toha, S.F. State of charge estimation of a lithium-ion battery for electric vehicle based on particle swarm optimization. In Proceedings of the 2013 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Kuala Lumpur, Malaysia, 25–27 November 2013; pp. 1–4.
32. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
33. Eberhart, R.C.; Shi, Y. *Computational Intelligence—Concepts to Implementations*, 1st ed.; Morgan Kaufmann: San Francisco, CA, USA, 2007.