



UNIVERSITY OF LEEDS

This is a repository copy of *Compact Neural Modeling of Single Flow Zinc-Nickel Batteries Based on Jaya Optimization*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/141010/>

Version: Accepted Version

Proceedings Paper:

Zhang, L, Li, K, Yang, Z et al. (4 more authors) (2018) Compact Neural Modeling of Single Flow Zinc-Nickel Batteries Based on Jaya Optimization. In: Proceedings of CEC 2018. 2018 IEEE Congress on Evolutionary Computation (CEC), 08-13 Jul 2018, Rio de Janeiro, Brazil. IEEE , pp. 851-856. ISBN 978-1-5090-6017-7

<https://doi.org/10.1109/CEC.2018.8477707>

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Compact Neural Modeling of Single Flow Zinc-Nickel Batteries Based on Jaya Optimization

Li Zhang, Kang Li, Zhile Yang, Xiang Li, Yuanjun Guo, Dajun Du, Chikong Wong

Abstract—As a novel family member of the redox flow batteries (RFBs), the single flow zincnickel battery (ZNB) without ion exchange membranes has attracted a lot of interests in recent years due to the high charging and discharging efficiencies. To understand the electrical behaviour is a key for proper battery management system. Unlike the electrochemical mechanism models and equivalent circuit models, the neural network based black-box model does not need knowledge about the electrochemical reactions and is a promising and adaptive approach for the ZNB battery modelling. In this paper, a compact radial basis function neural network is developed using a two-stage layer selection strategy to determine the network structure. While Jaya optimization is utilized to determine the non-linear parameters in the selected hidden nodes of the resultant RBF neural network (RBF-NN) model. The proposed method is implemented to model the ZNB to capture the non-linear electric behaviours through the readily measurable input signals. Experimental results manifest the accurate prediction capability of the resultant neural model and confirm the effectiveness of the proposed approach.

I. INTRODUCTION

The Redox flow batteries (RFBs) have been widely studied in recent years due to their impressive capacity to efficiently store large amounts of electrical energy with relatively lower cost. It is promising to store excessive intermittent renewable energy such as solar and wind power generation [1].

Since first proposed by Thaller [2] in 1976, many variants of redox flow batteries have been developed such as the all-vanadium system (Pinnacle VRB Ltd.), polysulphidebromine system (Regenesys Technologies Ltd.) and zincbromine system (ZBB Energy Ltd.). In 2004, Pletcher et al. proposed a novel redox lead-acid flow battery system with a single electrolyte [3]. In addition, another single electrolyte system using conventional zincnickel battery was proposed by Cheng et al. [4] in 2007, without any separators in the mechanical configurations for cost reduction.

In order to safely and efficiently use RFBs, accurate estimation of the state-of-charge (SOC) for battery cell is essential. Model-based methods are extensively applied in internal state estimation given that they are close-loop methods and less sensitive to the measurement errors if the model is designed properly. In this case, an accurate battery model, which is crucial to the model-based method, should therefore be built

Li Zhang and Dajun Du are with School of Mechatronics and Automation, Shanghai University, Shanghai 200072, China.

Kang Li and Xiang Li are with School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK.

Zhile Yang and Yuanjun Guo are with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China.

Chikong Wong is with the University of Macau, Department of Electrical and Computer Engineering, Macau, China.

first, based on the measured data such as the battery terminal voltage and load current. However, the battery behaviours are normally highly non-linear and non-stationary because of the significantly complex internal chemical reactions. In [5], a mathematical model was proposed based on the electrochemical principle with a number of complex equations and unmeasured quantities to be determined. An equivalent circuit model (ECM) was presented in [6] to describe the battery non-linear behaviours using electronic components. But both the two models are related to the internal reactions. Thus, a black box neural network model, which does not have to rely on the knowledge of the internal reaction, was applied to model ZNBs in [7]. In [7] the influence of the structure of the neural network was however neglected.

In order to model the non-linear behaviours of zincnickel batteries, a compact RBF-NN model is proposed in this paper. It is known that the number of hidden nodes and choice of suitable values for nonlinear parameters in the hidden nodes such as the centers and the widths, as well as the output weights [8] are important factors in constructing a good RBF-NN model. According to the principle of parsimony, a compact network is preferable to a complex one under the similar approximation and generalization performance. Li et al. proposed a fast recursive algorithm (FRA) [9] and a fast two-stage selection algorithm (TSS) [10] combining forward selection and backward refinement for building linear-in-the-parameter models for a wide class of nonlinear systems.. In this paper, the TSS model selection method is used to refine the structure of the RBF-NN model. With these novel algorithms, the number of hidden nodes and the weights relating to the output nodes are determined simultaneously. In addition, a recently efficient meta-heuristic algorithm namely Jaya [11] is applied to optimize the centers and the widths of the hidden nodes to improve the model accuracy.

The reminder of the paper is organized as follows. A brief introduction of the compact RBF neural network is presented in Section II, followed by Section III that introduces the Jaya algorithm. Then the procedure to built an optimal compact RBF-NN model is detailed in IV. Further, the experimental and simulation results are analyzed in the Section V. Finally, Section VI concludes this paper.

II. CONSTRUCTION OF COMPACT RBF NETWORKS USING TSS ALGORITHM

According to [9], [10], [12], the RBF-NN structure makes it possible to formulate its construction as a linear-in-the-parameters structure. Based on this formulation, a compact

RBF-NN can be built using the two-stage stepwise identification method. All candidate neurons are reviewed in the two stages to select the most significant hidden nodes for building the compact model. Meanwhile, the weights in relation to the output nodes are estimated based on the least square solution.

A. RBF Neural Networks

Consider a multi-input-single-output (MISO) RBF network is used to approximate a nonlinear system, it can be formulated as a linear-in-the-parameters model as follows:

$$y(t) = \sum_{k=1}^n \theta_k \varphi_k(X(t); c_k; \sigma_k) + \varepsilon(t) \quad (1)$$

where $y(t)$, $X(t) \in R^m$ and $\varepsilon(t)$ are output, input and model error at time instant t respectively. Herein, m and n denote the number of inputs and hidden nodes respectively. And $\varphi_k(X(t); c_k; \sigma_k)$ is the Gaussian function as the activation function for the hidden nodes. $c_k \in R^m$ is the center vector and $\sigma_k \in R^1$ denotes the RBF width. θ_k represents the output linear weights.

Suppose N samples such as the $X(t)$, $y(t)$, $t = 1, \dots, N$ are used as the training data for the RBF-NN construction. Therefore, Equ. (1) can be reformulated in a matrix form as:

$$\mathbf{y} = \Phi \theta + \mathbf{e} \quad (2)$$

where $\Phi = [\phi_1, \dots, \phi_n]^T \in R^{N \times n}$ is known as the output matrix of the hidden nodes. $\phi_i = [\varphi_i(X(1), \dots, \varphi_i(X(N))]^T$, $i = 1, \dots, n$, $\mathbf{y} = [y(1), \dots, y(N)]^T \in R^N$ is the output vector. Besides, $\theta = [\theta_1, \dots, \theta_n]^T \in R^n$ denotes the output weights.

B. Two Stage Stepwise Method

In order to construct a compact RBF-NN model with a minimal number of hidden nodes, a two-stage stepwise method [10] is employed in this paper. During the first stage, the FRA is used to select significantly non-linear regressors to build the parsimonious RBF-NN model [9]. Then, the already selected important model terms (nodes) are reviewed in the second stage. After term refinement, the weights of the model are estimated using the update recursion formulation. The optimized model usually has a better generalization capability.

a) Stage 1: Forward Selection: A compact RBF-NN model has been preliminarily formed during the forward selection by selecting the largest net contribution neurons, storing in a regression matrix \mathbf{P} and rearranging in Φ . Consider a linear-in-the-parameter RBF-NN with randomly generated hidden neurons, supposing i^{th} model terms (neurons), the corresponding optimal linear parameters (e.g. output weights in the RBF-NN) are given below using the least square algorithm (LS).

$$\hat{\theta}_i = (\Phi_i^T \Phi_i)^{-1} \Phi_i^T \mathbf{y} \quad (3)$$

Thus the cost function with the optimal output weights can be formulated as

$$\mathbf{J}_i = \mathbf{y}^T \mathbf{R}_i \mathbf{y} \quad (4)$$

where $\mathbf{R}_i = I - \Phi_i [\Phi_i^T \Phi_i]^{-1} \Phi_i^T$ is called the residue matrix. Then, the net contribution of a new model term φ_{i+1} at the $i + 1^{th}$ iteration is expressed as

$$\Delta \mathbf{J}_{i+1}(\varphi_{i+1}) = ((\mathbf{y}^{(i)})^T \cdot \varphi_{i+1}^{(i)})^2 / (\varphi_{i+1}^{(i)})^T \cdot \varphi_{i+1}^{(i)} \quad (5)$$

where φ_{i+1} is the reordered one, $\mathbf{y}^{(i)} = \mathbf{R}_i \mathbf{y}$, $\varphi_{i+1}^{(i)} = \mathbf{R}_i \varphi_{i+1}$, $k = 0, \dots, n - 1$.

If the k^{th} term is selected to construct the compact model, the intermediate matrices $A = [a_{i,j}]_{k \times n}$, $A_y = [ay_i]_{n \times 1}^T$ and $B = [b_i]_{n \times 1}$ are defined as

$$\begin{cases} a_{i,j} = 0, & j < i \\ a_{i,j} = (\varphi_i^{(i-1)})^T \varphi_j^{(i-1)} = (\varphi_i)^T \varphi_j - \sum_{h=1}^{i-1} a_{h,i}^2 / a_{h,h}, & j = i \\ a_{i,j} = (\varphi_i^{(i-1)})^T \varphi_j^{(i-1)} = (\varphi_i)^T \varphi_j - \sum_{h=1}^{i-1} (a_{h,i} a_{h,j}) / a_{h,h}, & j > i \end{cases} \quad (6)$$

$$\begin{cases} ay_i = \varphi_i^T \mathbf{R}_{i-1} \mathbf{y} = \varphi_i^T \mathbf{y} - \sum_{h=1}^{i-1} (a_{h,i} a_{y,h}) / a_{h,h}, & 1 \leq i < k \\ ay_i = \varphi_i^T \mathbf{R}_k \mathbf{y} = \varphi_i^T \mathbf{y} - \sum_{h=1}^{k-1} (a_{h,i} a_{y,h}) / a_{h,h}, & k \leq i \leq n \end{cases} \quad (7)$$

$$\begin{cases} b_i = (\varphi_i^{(i-1)})^T \varphi_i^{(i-1)} = (\varphi_i)^T \varphi_i - \sum_{h=1}^{i-1} a_{h,i}^2 / a_{h,h}, & 1 \leq i < k \\ b_i = \varphi_i^T \mathbf{R}_k \varphi_i = (\varphi_i)^T \varphi_i - \sum_{h=1}^{k-1} a_{h,i}^2 / a_{h,h}, & k \leq i \leq n \end{cases} \quad (8)$$

Therefore, the net contribution of the $k + 1^{th}$ node can be expressed as

$$\Delta \mathbf{J}_{k+1}(\varphi_{i+1}) = \frac{(\mathbf{y}^T \varphi_{i+1} - \sum_{h=1}^k (a_{y,h} a_{h,i+1} / a_{h,h}))^2}{(\varphi_{i+1})^T \varphi_{i+1} - \sum_{h=1}^k a_{h,i+1}^2 / a_{h,h}} \quad (9)$$

$$k \leq i \leq n$$

b) Stage 2: Backward Refinement: The backward refinement stage is used to re-examine the model constructed by the forward selection approach. The insignificant ones are replaced by the more significant nodes using the recalculated net contributions through the changes of the nodes position in regression matrix. In brief, the unselected items in the initial set of nodes which have more significant contributions will

be adopted in the new compact RBF-NN model after the backward refinement.

The forward regression matrix comprising k^{th} selected nodes is denoted as $\mathbf{P}_k = [p_1, \dots, p_k]$, and the adjacent terms are interchanged ($\hat{p}_q = p_{q-1}$ & $\hat{p}_{q-1} = p_q$). Therefore, some changes occur in the corresponding residue matrix and intermediate matrices which constitute the regression context defined in [10]. The residue matrix $\hat{\mathbf{R}}_q$ is rewritten as

$$\begin{aligned} \hat{\mathbf{R}}_q &= \mathbf{R}(p_1, \dots, p_{q-1}, \hat{p}_q) \\ &= \mathbf{R}_{q-1} - \mathbf{R}_{q-1} \cdot \hat{p}_q \cdot \hat{p}_q^T \cdot \mathbf{R}_{q-1}^T / (\hat{p}_q^T \cdot \mathbf{R}_{q-1} \cdot \hat{p}_q) \end{aligned} \quad (10)$$

then the interchanged $A_{i,q;q+1}$ are given by

$$\begin{cases} \hat{A}(1 : q - 1, q) = A(1 : q - 1, q + 1) \\ \hat{A}(1 : q - 1, q + 1) = A(1 : q - 1, q) \end{cases} \quad (11)$$

and the position interchanging in the row is described as

$$\begin{cases} \hat{A}(q, q) = A(q+1, q+1) + A^2(q, q+1) / A(q, q) \\ \hat{A}(q, q+2:n) = A(q+1, q+2:n) + \frac{A(q, q+1)A(q, q+2:n)}{A(q, q)} \\ \hat{A}(q+1, q+1) = A(q, q) - A^2(q, q+1) / \hat{A}(q, q) \\ \hat{A}(q+1, q+2:n) = A(q, q+2:n) - \frac{A(q, q+1)\hat{A}(q, q+2:n)}{\hat{A}(q, q)} \end{cases} \quad (12)$$

and $B(q : q + 1)$ can be reformulated as

$$\hat{B}(q) = \hat{A}(q, q), \hat{B}(q + 1) = \hat{A}(q + 1, q + 1) \quad (13)$$

Similarly, $A_y(q : q + 1)$ is updated as

$$\begin{cases} \hat{A}_y(q) = A_y(q + 1) + A(q, q + 1)A_y(q) / A(q, q) \\ \hat{A}_y(q + 1) = A_y(q) - A(q, q + 1)\hat{A}_y(q) / \hat{A}(q, q) \end{cases} \quad (14)$$

The remaining candidates will be shifted to the k^{th} position, i.e. $\hat{p}_k = \varphi_i$, of which the net contribution is calculated as follows

$$\Delta \mathbf{J}_k(\varphi_i) = \frac{(\mathbf{y}^T \varphi_i - \sum_{h=1}^{k-1} \hat{a}_y \hat{a}_{h,i} / \hat{a}_{h,h})^2}{(\varphi_i)^T \varphi_i - \sum_{h=1}^{k-1} \hat{a}_{h,i}^2 / \hat{a}_{h,h}} \quad (15)$$

where $\hat{\mathbf{R}}_{k-1}$, $\hat{a}_{\cdot, \cdot}$ and \hat{a}_y are the $(k - t)^{th}$ updated values in Step 1, and t is the initial position index of the reviewed elements.

Subsequently, the corresponding weights will be estimated as

$$\hat{\theta}_j = (a_{j,y} - \sum_{h=j+1}^k \hat{\theta}_h a_{j,h}) / a_{j,j}, \quad j = k, k - 1, \dots, 1 \quad (16)$$

III. JAYA ALGORITHM

The selection of centers and the widths of the hidden nodes directly determine the performance of the RBF-NN model. Due to the highly non-linear nature of the zincnickel batteries model, the optimal centers and the widths of the hidden nodes should be identified. Jaya algorithm is a recently proposed

meta-heuristic algorithm with no parameters to tune and high competitive performance [11]. The evolutionary process is fairly straightforward: approaching the best and departing from the worst. The both ideas are merged within a single phase denoted as

$$X_{j,k,i}^{new} = X_{j,k,i} + rand_{1,j,i}(X_{j,best,i} - |X_{j,k,i}|) - rand_{2,j,i}(X_{j,worst,i} - |X_{j,k,i}|) \quad (17)$$

where $X_{j,k,i}$ is the j^{th} variable of k^{th} particle in the i^{th} iteration. $X_{j,best,i}$ and $X_{j,worst,i}$ denote the best and worst candidates of corresponding position. $X'_{j,k,i}$ is the updated value of $X_{j,k,i}$ and $rand_{1,j,i}$ and $rand_{2,j,i}$ are the two random numbers ranging within (0, 1). It could be observed that the term ' $(X_{j,best,i} - |X_{j,k,i}|)$ ' denotes the tendency of the solution to approach the best solution and the term ' $-(X_{j,worst,i} - |X_{j,k,i}|)$ ' indicates the tendency of the solution to disembarass the worst solution. $X'_{j,k,i}$ is accepted if it gives better function value.

The algorithm has been well adopted in solving engineering problems [13], [14] and achieved good results. In this paper, the non-linear parameters in the RBF-NN model formulated in (1) is optimized by Jaya, and the detailed procedure is demonstrated in the following section.

IV. MODELLING USING A COMPACT RBF NEURAL NETWORK OPTIMIZED BY JAYA

Due to the highly nonlinear nature of the ZNB electrical behavior, a method combining the procedures of the TSS and Jaya algorithm is employed to build the RBF-NN model (TSS_Jaya_RBF). In this model, a dynamic optimizer of the nonlinear parameters and a dynamic structure selector is designed.

To build the electrical behavior of the ZNB, according to [7], the past terminal voltage measurements $V(t-1)$, $V(t-5)$, $V(t-10)$ at time instants $t-1$, $t-5$, $t-10$ and the current measurements $I(t)$ and $I(t-7)$ at time instants t , $t-7$ are selected as the model inputs. The present voltage $V(t)$ at time instants t is the model output. A compact RBF-NN model below is used to capture their nonlinear relationships

$$V(t) = \sum_{i=1}^k \theta_i \exp(-\|u(t) - uc_i\|^2 / (2\sigma_i)^2) \quad (18)$$

where $u(t) = [V(t-1), V(t-5), V(t-10), I(t), I(t-7)]$ is the model input vector, the Gaussian function $\exp(-\|u(t) - uc_i\|^2 / (2\sigma_i)^2)$ is the output of the i^{th} hidden nodes with the center vector uc_i and the width vector σ_i , k is the selected number of hidden nodes in the RBF neural model, and θ_i is the hidden output weights.

To identify the model described in (18), the number of the hidden nodes (k) and the weights θ_i is determined using TSS method, and the center vector uc_i and the width vector σ_i is optimized using Jaya algorithm. Thus the procedures of the TSS_Jaya_RBF model is described as follows:

Step 1. Initialization:

- a) collect the data samples and generate the candidate RBF regression matrix $\Phi = [\Phi_1, \dots, \Phi_M]$, $M = N$.
- b) set the desired number of model terms k , the size of the population, the initial population, the upper/lower bounds and the iteration number l .

Step 2. Forward selection:

- a) Set the model size $k = 0$.
- b) At the k^{th} , $1 \leq k \leq M$ step, calculate $a_{j,j}$ and ay_j ($j = 1, \dots, M$) using (6) and (7). Then compute their net contributions to the cost function using (9) based on the widths and centers selected by Jaya.
- c) Find the candidate regression terms with the maximum net contribution and the minimal net contribution, then update the widths and centers using (17).
- d) If the iteration number l is reached, move to 2(e). Otherwise, set $k = k + 1$, and go back to 2(b).
- e) Construct the candidate RBF regression vector φ_{nn} using the selected widths and centers based on Jaya algorithm and save it into the regression matrix \mathbf{P} .
- f) If the desired number of model terms (k) is reached, move to Step 3. Otherwise, set $nn = nn + 1$, and go back to 2(b).

Step 3. Backward model refinement:

- a) Interchange the positions of p_{nn} and p_{nn+1} ($nn = n-1, \dots, 1$), and update the related terms in A, A_y and B using (11), (12), (13) and (14).
- b) Repeat 3(a) until the regressor p_{nn} shifted to the k^{th} position.
- c) Calculate the new net contribution of the reviewed term and the term (ϕ_i , $i = n+1, \dots, m$) remaining in the candidate pool.
- d) Find the candidate regression terms with the maximum net contribution and the minimal net contribution, then update the widths and centers using (17).
- e) If the iterative l is reached, move to 3(f). Otherwise, set $nn = nn + 1$, and go back to 3(c).
- f) Construct the candidate RBF regression vector ϕ_i using the selected widths and centers based on the Jaya algorithm.
- g) If the net contribution of the shifted term is less than that of a candidate, replace p_{nn} with ϕ_i , and move p_{nn} back to the candidate pool. Otherwise, set $nn = nn - 1$, if $nn > 1$, then go to step 3(a), and if $nn = 1$, go to the next step.

Step 4. calculate the weights:

- a) Estimate the weights using (16).

The schematic is illustrated in Fig.1.

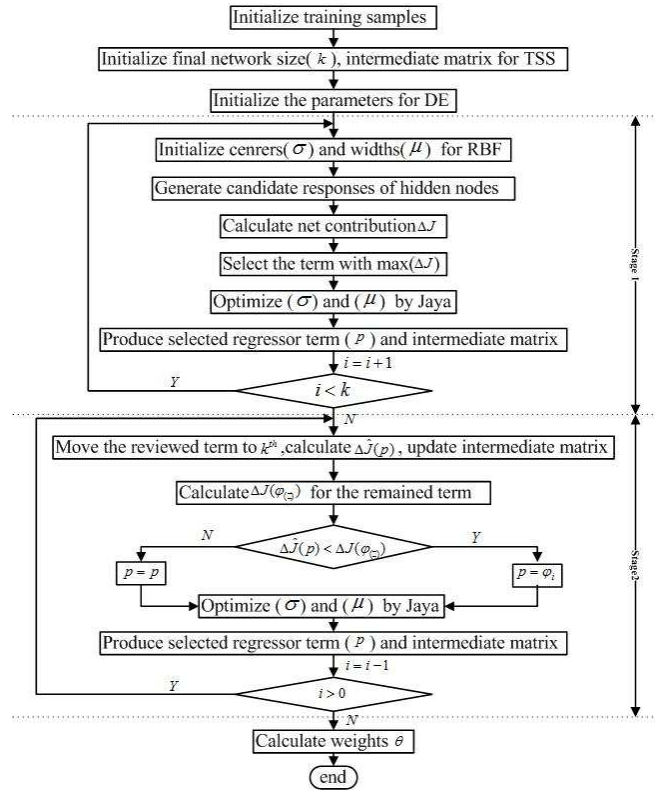


Fig. 1. Modeling flowchart diagram for ZNBs using Jaya trained compact RBF neural network

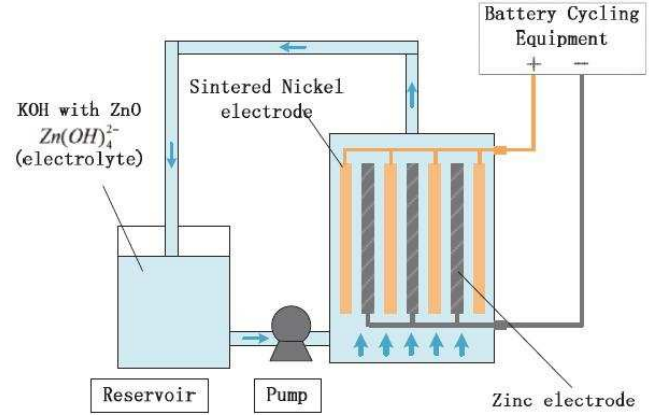


Fig. 2. Schematic diagram of single ZNB system.

V. SIMULATION AND ANALYSIS

In this paper, a ZNB flow battery system includes a stack, an electrolyte reservoir, several positive electrodes with nickel oxides and several negative electrodes with zinc as illustrated in Fig.2. The electrolyte employing the concentrated solutions of ZnO is circulated from the reservoir by a pump to the electrode cell.

Based on the ZNB battery system shown in Fig. 2, sample data was collected from a bench-scale 1.85 Ah ZNB

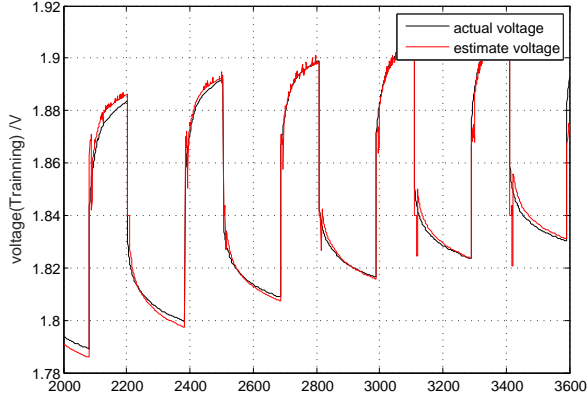


Fig. 3. Training Voltage using the compact RBF-NN model optimized by Jaya

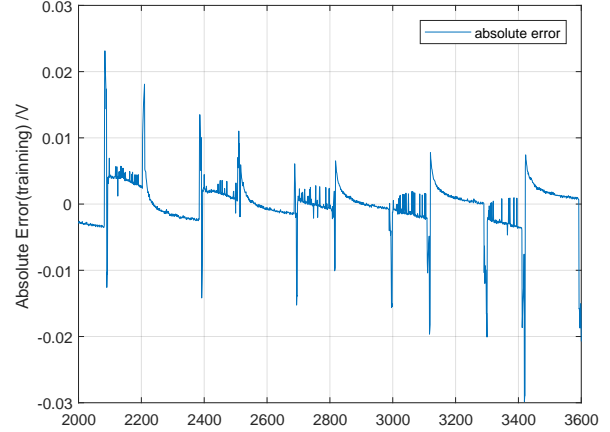


Fig. 4. Training Error using the compact RBF-NN model optimized by Jaya

using 0.5C charging/discharging current to test the electrical behavior of the system. The battery tests were performed using a battery testing system CT-3008W (ShenZhen Neware Corp., China). During the testing procedure, the temperature was fixed as the room temperature. In order to capture and demonstrate the non-linear relationship, 10 intervals representing different state of charge (SOC) points were used in the experiments. Eight points between SOC 10% and 90% were evenly chosen during charging processing. In the charging cycle, a duration period of relaxation was up to 15 min after each 10 min pulse current charging.

The maximum number of generations and the population size of the Jaya method are set as 10 and 60 respectively. Moreover, the upper/lower bounds of the center is the 1.5 times of the input maximum/minimum, and the upper and lower bounds of the width is 0.1 and 8 respectively. The simulation test using the optimized compact RBF-NN is shown in Fig. 3. The absolute errors range from 0.03 to 0.03, with the majority of them falling within ± 0.01 as shown in Fig. 4. Besides, the number of the hidden nodes is selected as 6. It is clear that a highly accurate model is built using the proposed method.

In addition, another set of ZNB test data is collected for model validation under the same testing environments and the same charging current. The validation results are illustrated in Fig. 5, and it is noted that the majority of absolute errors are within ± 0.02 as shown in Fig. 6.

In order to illustrate the effectiveness of Jaya method, we compare the compact RBF-NN model optimized by the conventional PSO method [15], with the same maximum iteration and number of particles. The results are illustrated in Fig. 7. It is shown that the compact RBF-NN model optimized by Jaya is more accurate, in particular outperforms the PSO in the charging process.

VI. CONCLUSION AND FUTURE WORKS

In this paper, an accurate ZNBs black box model to capture the electrical dynamics has been built using an compact RBF-NN algorithm. The two stage selection method is adopted to

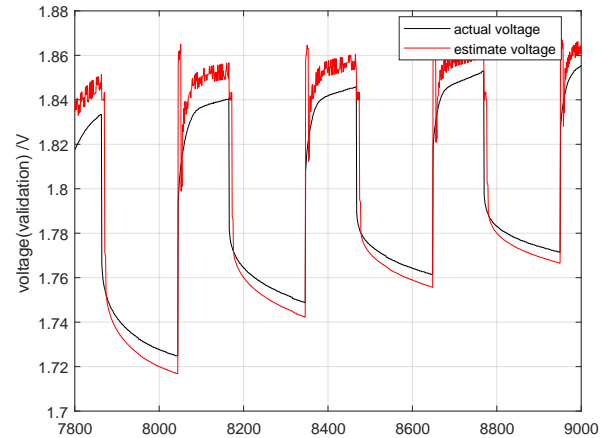


Fig. 5. Validation Voltage using the compact RBF-NN model optimized by Jaya

determine the network structure and establish a compact RBF-NN with limited number of hidden nodes. In order to improve the accuracy of the model, an elegant optimizer Jaya method is employed to adjust the non-linear parameters in the radial basis functions. The simulation result demonstrates that both the modelling performance and computational efficiency are significantly improved with the TSS selected structure and Jaya optimized nonlinear parameters. Future work will be addressing the SOC estimation based on the model proposed in this note.

ACKNOWLEDGMENTS

This paper was partially funded by the NSFC under grant 51607177 and 61673256. State Key Laboratory of Alternate Electrical Power System with Renewable Energy Source under grant LAPS17018. L. Zhang, X. Li and C. K. Wong would like to thank the Macao Science and Technology Development

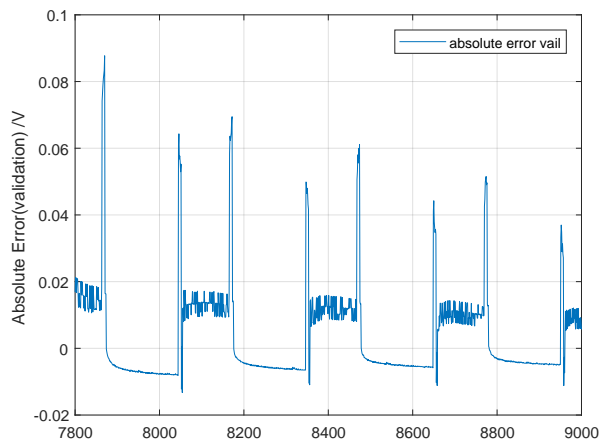


Fig. 6. Validation Error using the compact RBF-NN model optimized by Jaya

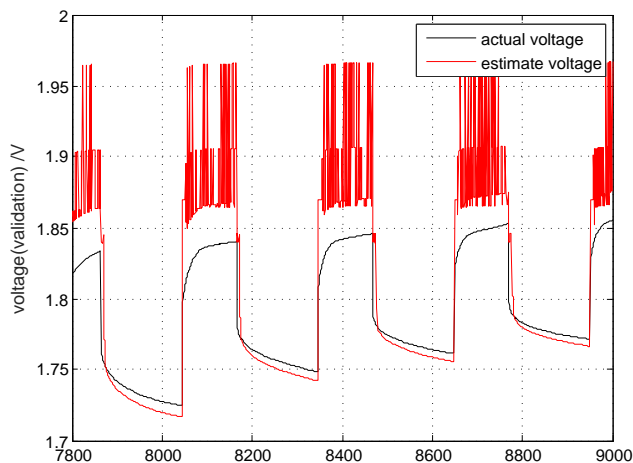


Fig. 7. Validation Error using the compact RBF-NN model optimized by PSO

Fund (FDCT) 's support with the project (111/2013/A3)-Flow Battery Storage System Study and Its Application in Power System.

REFERENCES

- [1] A. Z. Weber, M. M. Mench, J. P. Meyers, P. N. Ross, J. T. Gostick, and Q. Liu, "Redox flow batteries: a review," *Journal of Applied Electrochemistry*, vol. 41, no. 10, p. 1137, 2011.
- [2] L. H. Thaller, "Electrically rechargeable redox flow cell," Dec. 7 1976, uS Patent 3,996,064.
- [3] A. Hazza, D. Pletcher, and R. Wills, "A novel flow battery: A lead acid battery based on an electrolyte with soluble lead (ii) part i. preliminary studies," *Physical Chemistry Chemical Physics*, vol. 6, no. 8, pp. 1773–1778, 2004.
- [4] J. Cheng, L. Zhang, Y.-S. Yang, Y.-H. Wen, G.-P. Cao, and X.-D. Wang, "Preliminary study of single flow zinc-nickel battery," *Electrochemistry Communications*, vol. 9, no. 11, pp. 2639–2642, 2007.
- [5] P. M. Gomadam, J. W. Weidner, R. A. Dougal, and R. E. White, "Mathematical modeling of lithium-ion and nickel battery systems," *Journal of power sources*, vol. 110, no. 2, pp. 267–284, 2002.

- [6] Y. X. Li, M. C. Dong, P. C. Zhao, and Y. D. Han, "Modeling of single flow zinc-nickel battery for system efficiency improvement," *Applied Mechanics & Materials*, vol. 716-717, pp. 94–97, 2014.
- [7] X. Li, K. Li, Z. Yang, and C. Wong, "A novel rbf neural model for single flow zinc nickel batteries," in *Advanced Computational Methods in Energy, Power, Electric Vehicles, and Their Integration*. Springer, 2017, pp. 386–395.
- [8] J. Deng, K. Li, and G. W. Irwin, "A two-stage algorithm for automatic construction of rbf neural models," in *MELECON 2010-2010 15th IEEE Mediterranean Electrotechnical Conference*. IEEE, 2010, pp. 166–171.
- [9] K. Li, J.-X. Peng, and G. W. Irwin, "A fast nonlinear model identification method," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, 2005.
- [10] K. Li, J.-X. Peng, and E.-W. Bai, "A two-stage algorithm for identification of nonlinear dynamic systems," *Automatica*, vol. 42, no. 7, pp. 1189–1197, 2006.
- [11] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.
- [12] J. Deng, K. Li, E. Harkin-Jones, M. Fei, and S. Li, "A novel two stage algorithm for construction of rbf neural models based on a-optimality criterion," in *Natural Computation (ICNC), 2013 Ninth International Conference on*. IEEE, 2013, pp. 1–7.
- [13] K. Yu, X. Chen, X. Wang, and Z. Wang, "Parameters identification of photovoltaic models using self-adaptive teaching-learning-based optimization," *Energy Conversion and Management*, vol. 145, pp. 233–246, 2017.
- [14] L. Wang, Z. Zhang, C. Huang, and K. L. Tsui, "A gpu-accelerated parallel jaya algorithm for efficiently estimating li-ion battery model parameters," *Applied Soft Computing*, 2017.
- [15] J. Deng, K. Li, G. W. Irwin, and M. Fei, "Two-stage rbf network construction based on particle swarm optimization," *Transactions of the Institute of Measurement and Control*, vol. 35, no. 1, pp. 25–33, 2013.