Available online at www.sciencedirect.com
ScienceDirect
Journal of Power Sources 172 (2007) 749–759
JOURNAL OF POWER SOURCES
www.elsevier.com/locate/jpowsour

Neural network model for a commercial PEM fuel cell system

Anucha Saengrung [a,*], Amir Abtahi [b], Ali Zilouchian [a]

[a] Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA
[b] Department of Mechanical Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA

Received 28 March 2007; received in revised form 13 May 2007; accepted 15 May 2007
Available online 18 May 2007

Abstract

Performance prediction of a commercial proton exchange membrane (PEM) fuel cell system by using artificial neural networks (ANNs) is investigated. Two artificial neural networks including the back-propagation (BP) and radial basis function (RBF) networks are constructed, tested and compared. Experimental data as well as preprocess data are utilized to determine the accuracy and speed of several prediction algorithms. The performance of the BP network is investigated by varying error goals, number of neurons, number of layers and training algorithms. The prediction performance of RBF network is also presented. The simulation results have shown that both the BP and RBF networks can successfully predict the stack voltage and current of a commercial PEM fuel cell system. Speed and accuracy of the prediction algorithms are quite satisfactory for the real-time control of this particular application.

Keywords: Artificial neural network (ANN); Neural network; Proton exchange membrane fuel cell (PEMFC); Back-propagation (BP); Radial basis function (RBF) network; Modeling

1. Introduction

A fuel cell is an electrochemical device that can convert the chemical energy into electrical and thermal energy [1]. It is environment friendly as the only byproducts are water and heat. There are various types of fuel cells which can generate electrical power ranging from milliwatts to megawatts. It can be utilized in a construction of portable electronic equipment, vehicles, residential or even in distributed power systems [2].

A typical type of fuel cell is the proton exchange membrane fuel cell (PEMFC). There are several reasons for PEMFC to be a popular technology including its solid membrane and medium temperature range operation, which allow PEMFC to be operated in any orientation and easy start-up.

To design and control a fuel cell system for the maximum power performance, a designer needs to acquire sufficient knowledge pertaining to the physical process, the internal structure of the process, as well as the dominant input/output variables of the system. In sequel, an accurate mathematical representation (model) of the system may be developed with sufficient fidelity to accurately reflect the behavior of the physical process.

During the last decade, several one-dimensional (1D) and multi-dimensional (MD) models have been developed to explain the electrochemical and/or thermodynamic phenomena inside the fuel cells [3–9]. However, some of these models require specific knowledge of parameters, i.e., membrane thickness and resistance which are either unknown or only known to the manufacturers. Therefore, the availability of the electrochemical equations or models may not be sufficient to accurately design the fuel cell system for the optimum performances. In addition, these models as described above are commonly very complicated for large-scale fuel cell systems.

In the other hand, in most of control applications, the designer may be interested in relationship between inputs and outputs as well as the internal structure of the system. Such knowledge will provide the designers with the sufficient tool to control the inputs in order to reach the desired outputs, i.e., stack voltage and stack current for our particular application. Such a prediction may be performed by using artificial neural networks [10–13].

In this paper, we investigate the reliability of the BP and RBF networks for the output prediction of a 1.2 kW Nexa^{TM} fuel cell system [14]. First, the fuel cell system and load bank are briefly introduced. Second, the data collection and construction of the

* Corresponding author. Tel.: +1 561 395 4088; fax: +1 561 297 2336.
E-mail address: asaengru@fau.edu (A. Saengrung).

0378-7753/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2007.05.039

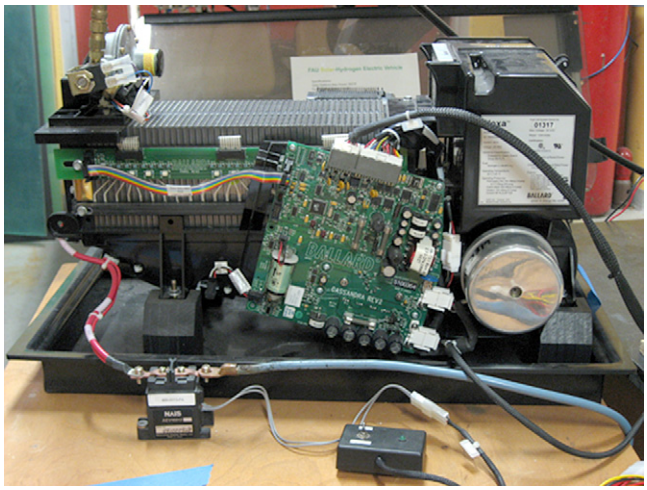

Fig. 1. The 1.2 kW Nexa™ fuel cell.

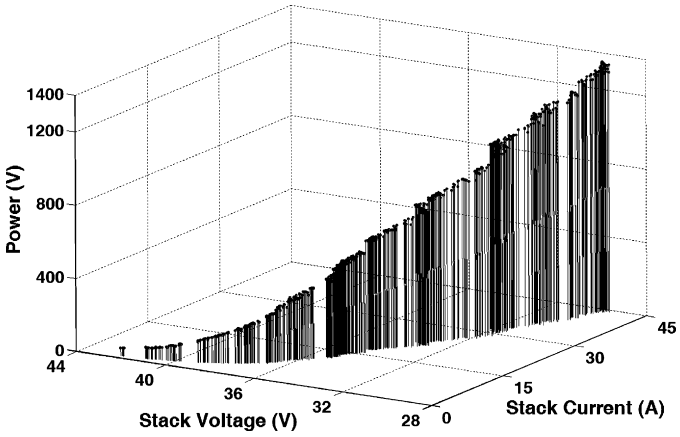

Fig. 3. Stack voltage, stack current and power of 1.2 kW Ballard™ fuel cell.

BP and RBF networks are described. Third, the reliability of the constructed neural networks to predict the performance of the fuel cell system is investigated. In addition, a comparison of the BP and RBF networks in term of error goal, training algorithms and network architectures have been investigated. Finally, the results and conclusions are discussed.

2. System description

2.1. Fuel cell system

A 1.2 kW Nexa™ fuel cell system as shown in Fig. 1, contains a BALLARD® fuel cell stack as well as all the ancillary equipment necessary for fuel cell operation. Ancillary subsystems include hydrogen delivery, oxidant air supply and cooling air supply. Onboard sensors monitor system performance and the control board fully automates operation [14].

The stack voltage and current of range from 42 V/1 A at idle to 30 V/45 A at full load. The system is capable of communicating with a computer via LabVIEW™ for data collection and system monitoring.

The system configuration, shown in Fig. 2, consists of seven control signals and nine sensors to monitor the significant variables. The hydrogen pressure is regulated by a regulator valve. The system controls the air compressor for air flow, and the cooling fan for stack temperature.

Figs. 3–5 depict the performance of this fuel cell with two input variables which are air flow and stack temperature. The 3D graph shows the visual relationship between inputs and outputs of the fuel cell.

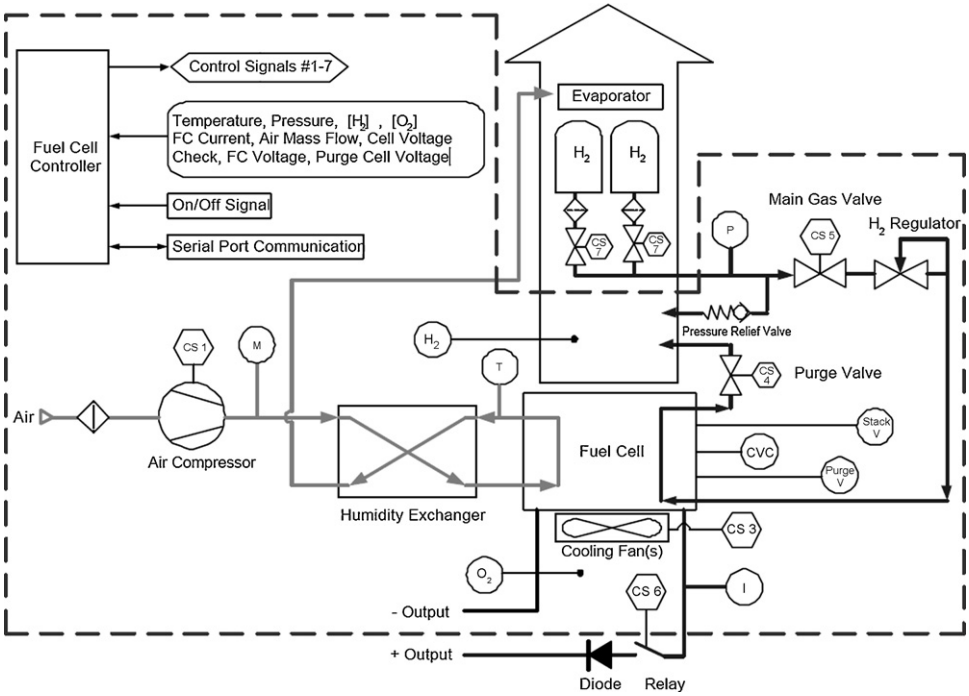

Fig. 2. Block diagram of the 1.2 kW Ballard™ fuel cell (Reprinted with permission – Ballard Power Systems Inc.).

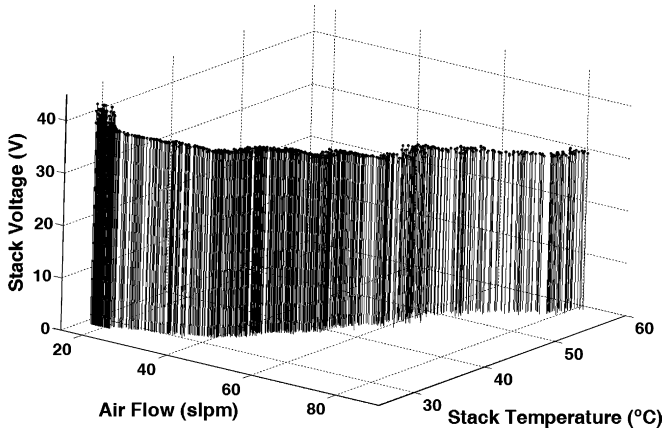

Fig. 4. The stack voltage according to air flow and stack temperature of 1.2 kW Ballard^{TM} fuel cell.

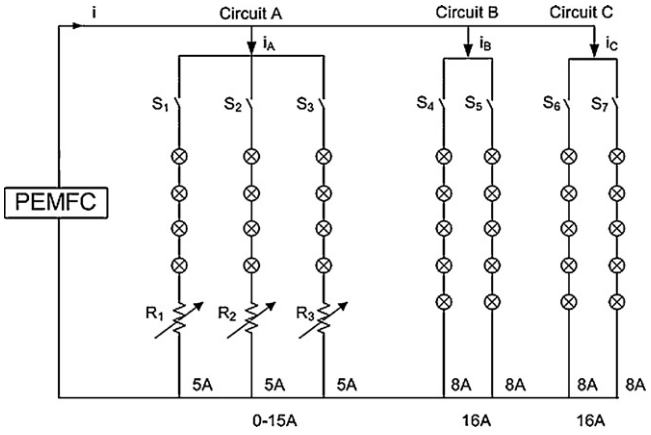

Fig. 6. In-house load bank configuration.

2.2. In-house load bank

To operate the fuel cell system at its rated power, a 1.2 kW load bank with a high cooling rate system is required. We initially used power MOSFETs as the load, but the generated heat from the load current was relatively high. The power MOSFETs failed frequently due to performance of the cooling system.

An alternative load bank has been developed which consists of three power resistors in combination with 32 12VDC light bulbs, as shown in Fig. 6. The load bank can increase the stack current up to 45 A. It is simple, inexpensive and more reliable. The solid-state relay switches are controlled by the LabVIEW^{TM}.

The load variation scheme applied throughout the investigation, shown in Fig. 7, can be explained as follows: First, three resistors are set to the maximum resistances. Then the first branch is switched on, the resistor R_1 is set to lower resistance and the stack current can be increased to 5 A which is the maximum current for the resistor.

When the second and third branches are switched on respectively, the stack current can be increased up to 15 A. In the next step, the first, second and third branches will be turned off and simultaneously, the fourth and fifth branches will be switched on to maintain the load current at about 16 A. In sequel, all resistors will be set to have maximum resistances again.

The procedure will be repeated again with the first, second and third branches to increase 0–15 A to the load; therefore, the load current now can be increased up to 30 A. The procedure will be repeated again to include the sixth and seventh branches so the stack current can be raised up to 45 A to obtain the rated power from the fuel cell system.

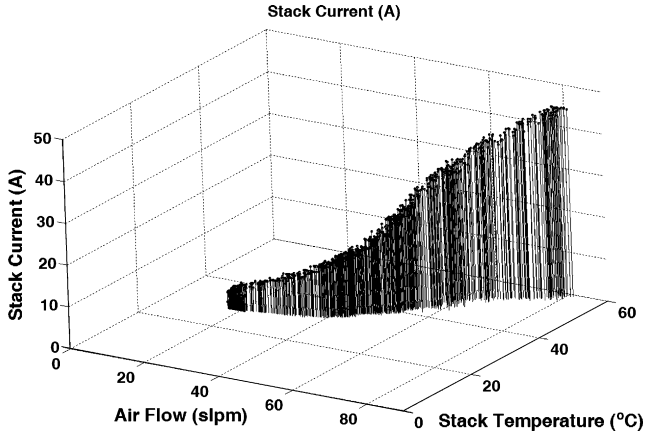

Fig. 5. The stack current according to air flow and stack temperature of 1.2 kW Ballard^{TM} fuel cell.

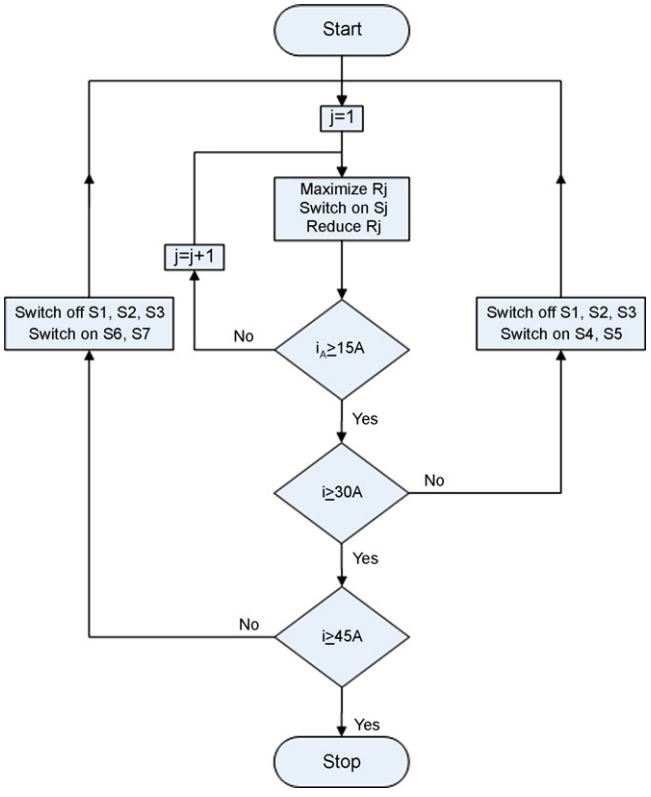

Fig. 7. Procedure to operate the load bank.

3. Data collection and analysis

3.1. Data collection

The fuel cell system was operated with the load bank up to maximum current of 45 A. After removing some data deemed unreliable, we had 883 data points to utilize for our proposed neural network algorithms.

The data sets collected for the modeling and prediction of the fuel cell include stack voltage (V), stack current (A), air flow (slpm), air temperature (°C), stack temperature (°C), fuel pressure (barg), fuel consumption (L), power (W), H_2 leak (%), O_2 concentration (%), purge cell voltage (V) and battery voltage (V).

3.2. Process variable selection

To train the network successfully and efficiently, one needs to select the appropriate variables as network inputs and outputs as well as the number of variables. To select the reliable variables, one needs to understand the process, how each variable affects the system performance then identify which variables are dominant and which one can be discarded. If there are too many variables as network inputs, the network will be unnecessary complicated and the training may be difficult or take too long time to succeed. In contrast, if the fewest dominant variables can be selected, the network will be small and can provide the fastest training and instant recall.

There are numerous methods for choosing appropriate input variables, i.e., using genetic algorithms [15], using reconstructability analysis [16], transform the input selection to model structure selection [17].

In our particular system, there are several available variables to be selected as inputs/outputs for our proposed neural networks. However, from manufacturer's system configuration viewpoint as shown in Fig. 2, the control board controls the air compressor for air flow and the cooling fan for stack temperature. Hydrogen pressure is kept constant by using a regulated valve. The humidity of air and hydrogen is managed by heat exchanger.

Therefore, we discard the other input variables which have been forced to be constant by the designer, and select only mass air flow (slpm) and stack temperature (°C) as the dominant input variables as specified by the manufacturer. Stack voltage (V) and current (I) are selected as the output variables due to their obvious relation to the fuel cell power generation.

3.3. Range of data

In general, neural network performs well in interpolation rather than extrapolation; therefore, in order to obtain a better prediction, the recall data should be in the range of training data. We randomly select the collected data to cover the range from minimum to maximum values as training data and the remaining as recall data. This approach will ensure that the recall data will always lie in the range of training data. The ranges of inputs/outputs data sets are as follows.

3.3.1. Ranges of inputs

Mass air flow range from 17 to 82 slpm, stack temperature range from 28 to 60 °C.

3.3.2. Ranges of outputs

Stack voltage range from 28 to 42 V, stack current (A) range from 1 to 45 A.

3.4. Size of training and recall data

When training data set is presented to the network, the weights and biases are updated on a pattern-by-pattern basis until the entire training data set is completed which is called one "epoch". This training phase is repeated until the network performs well according to the error goal as provided by the designer. In sequel, the recall data is presented to ensure that the network has learned the general patterns, not just simply has memorized the data set. If the network still performs well, the training is completed.

Training data set needs to be fairly large and contains variety of data in order to contain all the needed information. Therefore, in our investigation, from 883 data collected, 800 data is for training phase and 83 data is for recall phase. Figs. 8 and 9 show the graphs of inputs and outputs for training and recall phase, respectively.

3.5. Preprocess data

Instead of using only the raw data, we also preprocess data to have two additional data sets; normalized data and zero mean-unity standard deviation data. In the subsequent sections, we will investigate which data set provides a better and faster prediction.

The raw data can be normalized to have a range of [0,1] by using the following formula:

$$X_{\text{normalized}} = \frac{X - \min}{\max - \min} \tag{1}$$

Figs. 10 and 11 show the graphs of normalized data for the training and recall phase.

The zero mean-unity standard deviation data can be built from the raw data by using the following formula:

$$X_{\text{zu}} = \frac{X - \text{mean}}{\text{S.D.}} \tag{2}$$

where $\text{mean} = \sum_{i=1}^{n} Xi/n$ and $\text{S.D.} = \sqrt{\sum_{i=1}^{n} (Xi - \text{mean})^2/n}$

Figs. 12 and 13 show the graphs of zero mean-unity standard deviation data for the training and recall phase.

4. Construction of the BP and RBF networks

Inspired by the biological neural networks, an artificial neural network (ANN) is a massively parallel distributed processor made up of simple processing units, known as neurons. With ability to learn from input data with or without a teacher, neural

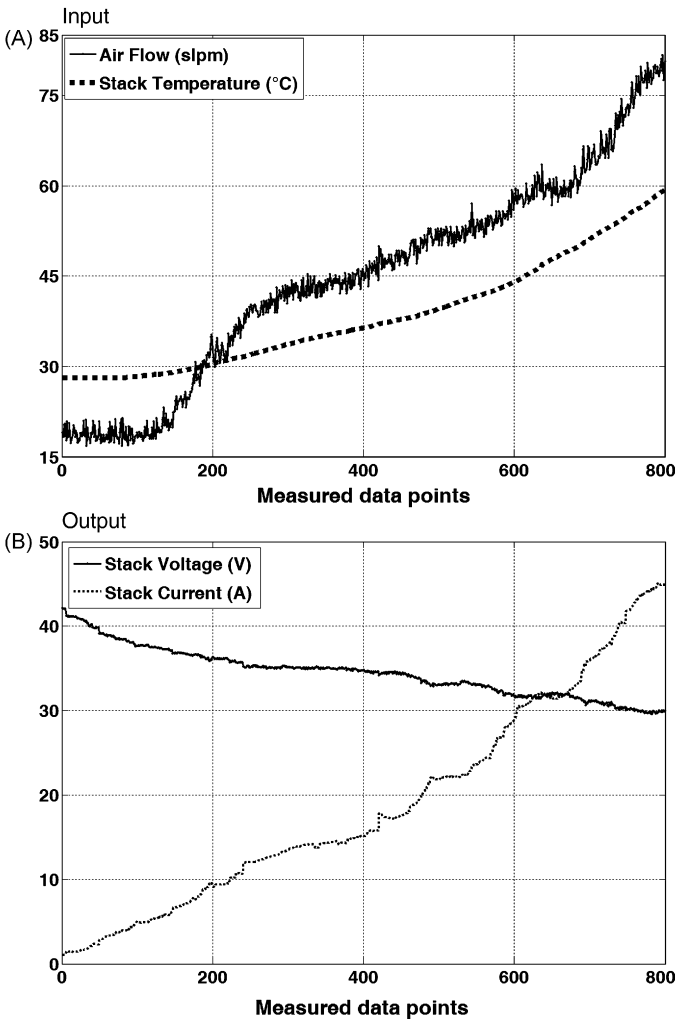

Fig. 8. Raw data for training phase.

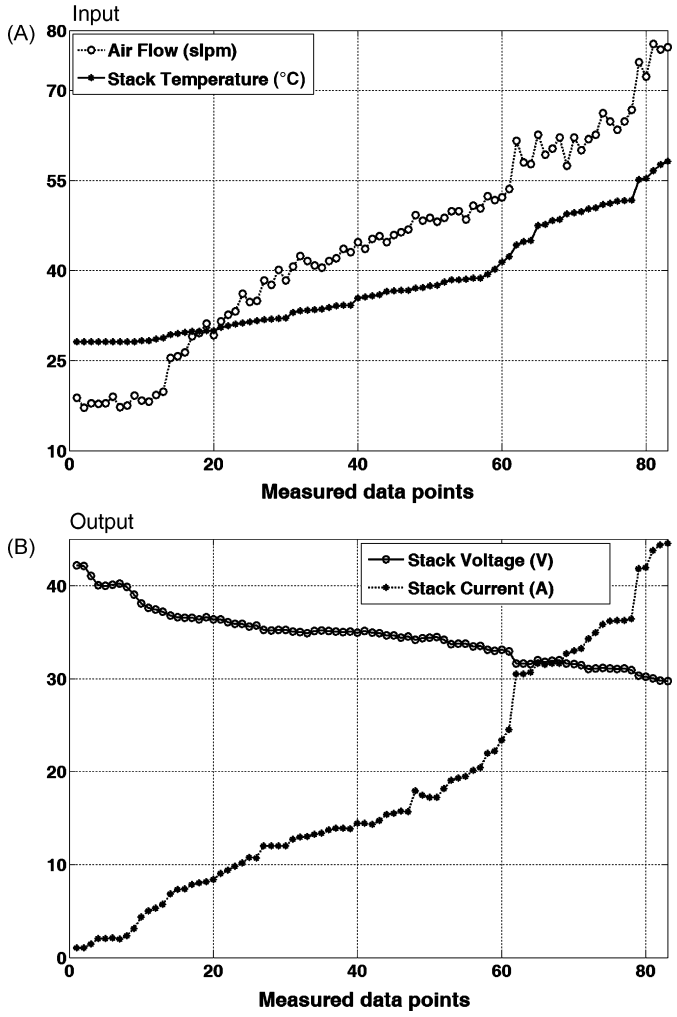

Fig. 9. Raw data for recall phase.

networks find applications in various fields, including modeling and control [12,13,15,18,19].

In this section, we construct the BP and RBF networks to investigate how well they can predict the performance of the 1.2 kW Nexa^{TM} fuel cell system.

4.1. Back-propagation networks

A BP network with two hidden layers, as shown in Fig. 14, is constructed. In each hidden layer, the number of neurons is varied to investigate the network prediction performance. The input layer has two input variables which are air flow and stack temperature. The output layer consists of two neurons for stack voltage and stack current. Subsequently, up to three hidden layers will be used to investigate the NN performances.

4.2. Radial basis function (RBF) network

The proposed RBF network consists of one hidden layer as shown in Fig. 15. The input and output layers are the same as in the BP network.

5. Results and discussions

The following three criteria are selected to investigate the prediction performances of the BP and RBF networks for 1.2 kW Nexa^{TM} fuel cell system:

(a) Number of epochs: indicates the training speed.
(b) Root mean square error (E_2): indicates average error of the prediction.

$$E_2 = \frac{\sqrt{\sum_{i=1}^{n} e_i^2}}{n} \tag{3}$$

(c) Absolute maximum error (E_∞): indicates the worst case error of the prediction.

$$E_\infty = \max_{i=1}^{n}(|e_i|) \tag{4}$$

5.1. Back-propagation results

5.1.1. Data set selection

When using raw data, normalized data and zero mean-unity standard deviation data, the performance predictions regarding

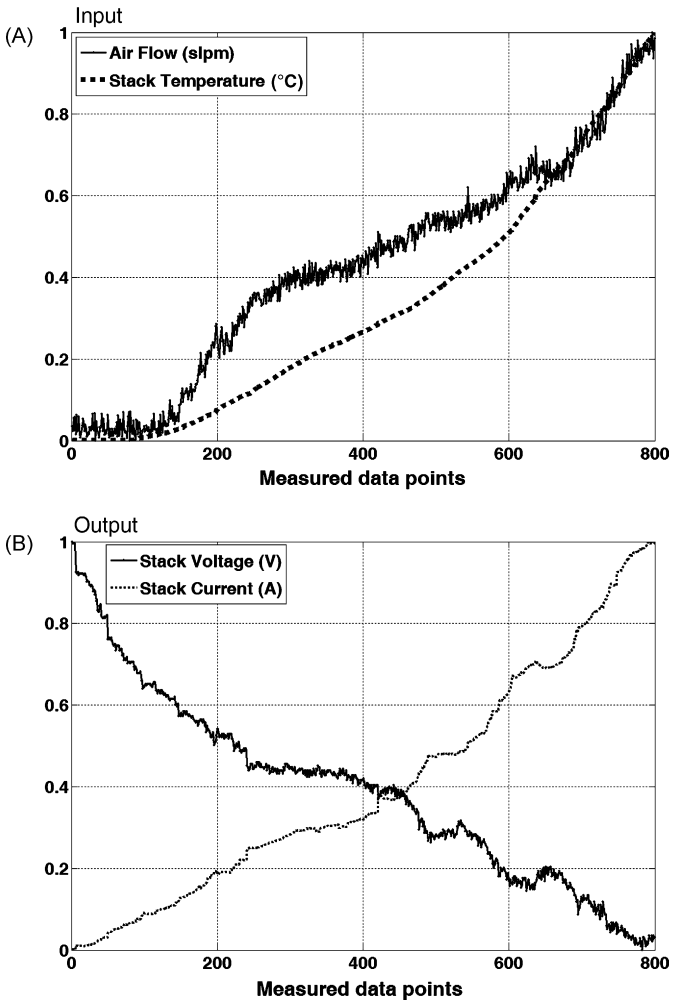

Fig. 10. Normalized data for training phase.

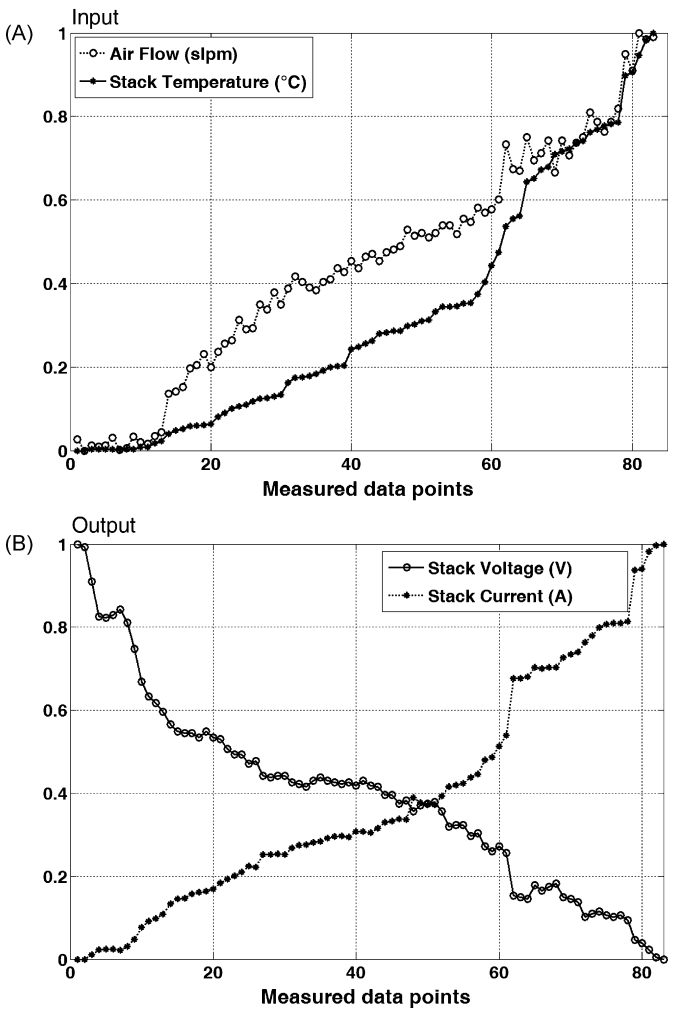

Fig. 11. Normalized data for recall phase.

to accuracy and time expenditure for the fuel cell system are shown in Tables 1 and 2.

These data sets are provided to the BP network whose weight and bias values are updated using the Levenberg–Marquardt optimization [20] with the error goal of 0.001.

Table 1
Training results from various data sets

Set of data	Epoch	E_2		E_∞	
		V	I	V	I
Raw	3000+	0.422	0.596	2.208	2.029
Normalized	21	0.518	0.736	2.584	3.089
Zero mean	100	0.426	0.746	2.211	2.892

Table 2
Recall results from various data sets

Set of data	E_2		E_∞	
	V	I	V	I
Raw	0.643	1.907	3.035	1.089
Normalized	0.474	1.126	1.984	2.951
Zero mean	0.977	0.991	4.377	3.286

From Table 1, it can be seen that the normalized data is the fastest in training phase. The average error (E_2) and maximum error (E_∞) using these three data sets are not much different. From Table 2, it shows that using the normalized data provides the best average error in the recall phase. Zero mean-unity standard deviation provides the worst prediction performances.The predictions of stack voltage and current for the fuel cell system are shown in Figs. 16 and 17. The results show the satisfactory predictions for the entire operating range except at the initial phase. From Figs. 8–13, it is observed that, at the starting phase of the fuel cell system, the inputs are almost constant even though the outputs are changing due to transient response of the feedback system at the initial phase. Therefore, the NN prediction does not perform well at the initial phase.

5.1.2. Error goals

In the training phase, the network will adjust its weights and biases until the output error reaches the designated error goal. If the selected error goal is relatively too large, then the training will be completed in a relative short time. However, a large error is obtained in the recall phase. On the other hand, if the error goal is selected to be too small, it will take a very long time for the

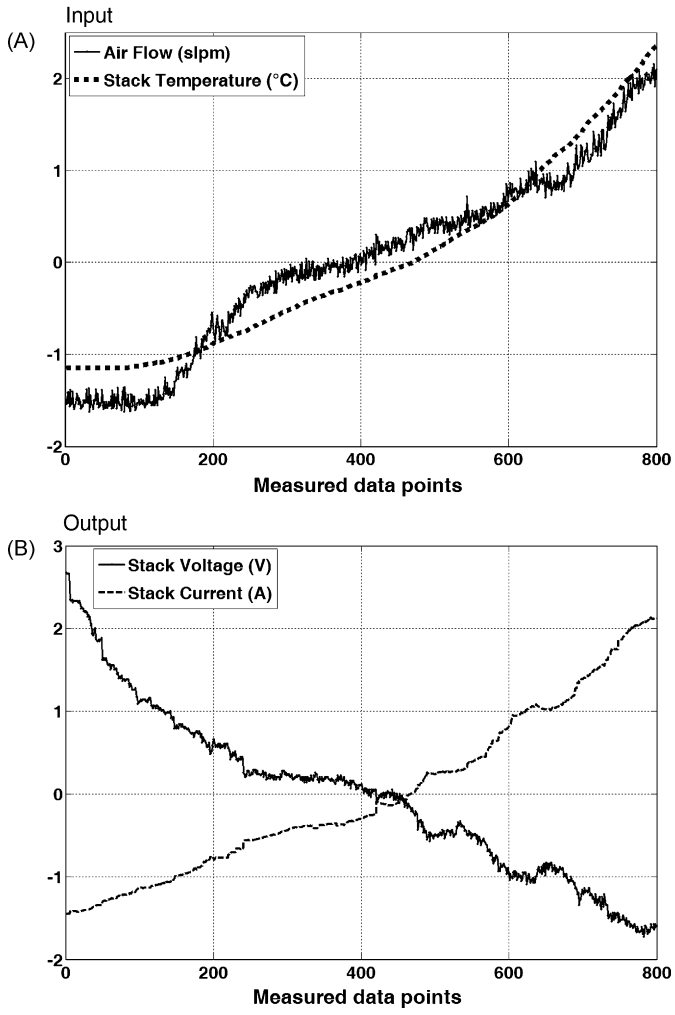

Fig. 12. Zero mean-unity standard deviation data for training phase.

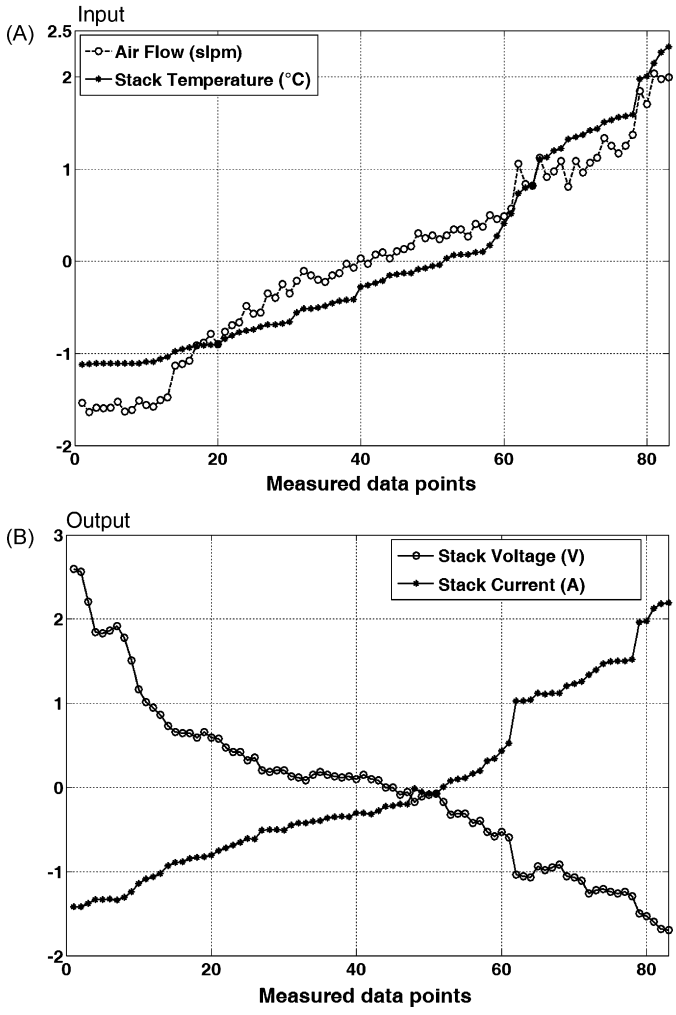

Fig. 13. Zero mean-unity standard deviation data for recall phase.

training to complete even though it provides a better prediction result.

The prediction results of various error goals are shown in Table 3 and Figs. 18 and 19. Epoch 1000+ indicates that the training cannot succeed in 1000 epochs. From the simulation results, it can be concluded that error goal of 0.001 provides the best prediction in term of speed and accuracy. Error goals of 0.1 and 0.01 are too large while the error of 0.0001 is too small.

Table 3
Prediction results for various error goals

Error goals		Epochs	E_2		E_∞	
			V	I	V	I
0.1	Training	1	2.915	4.501	6.414	9.999
	Recall		3.062	4.130	6.188	9.484
0.01	Training	1	1.583	1.981	4.305	4.720
	Recall		1.623	1.879	3.887	4.382
0.001*	Training	4	0.459	0.816	2.353	2.657
	Recall		0.494	0.801	2.232	1.791
0.0001	Training	1000+	0.3702	0.411	2.125	1.697
	Recall		0.440	0.726	2.019	1.975

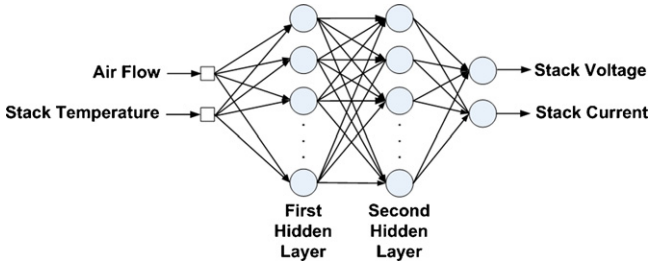

Fig. 14. Back-propagation network to predict the performance of the fuel cell system.

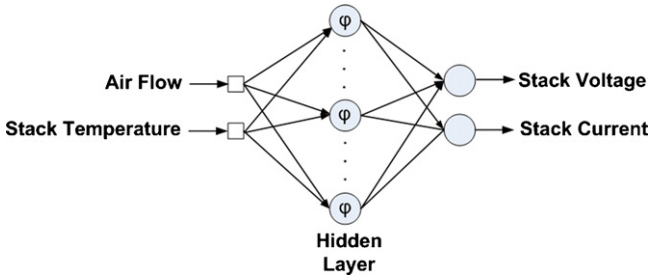

Fig. 15. RBF network to predict the performance of the fuel cell system.

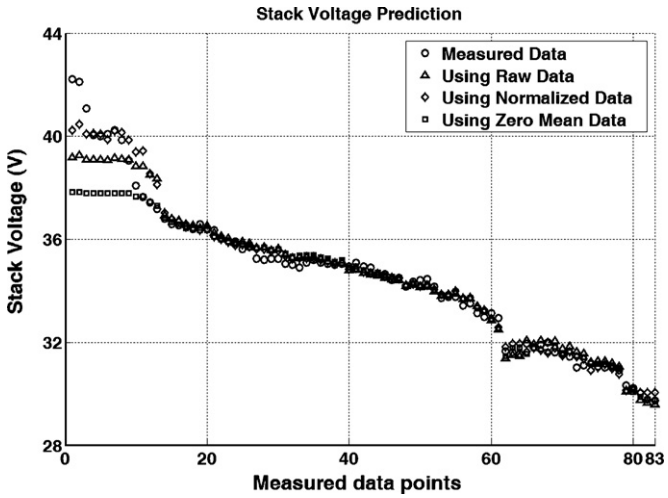

Fig. 16. Stack voltage predictions by using three data sets.

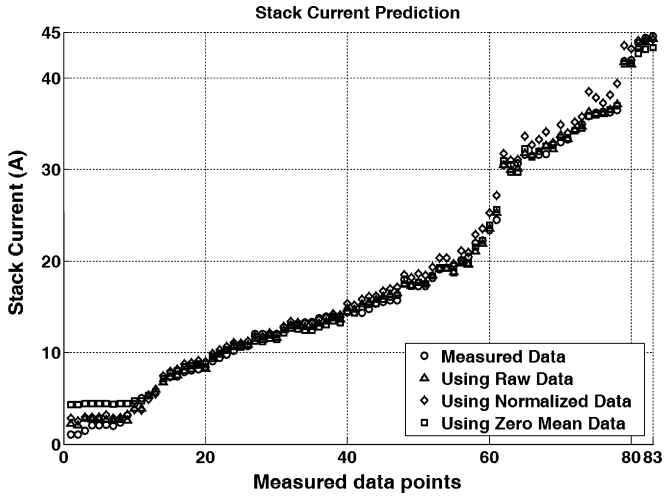

Fig. 17. Stack current predictions by using three data sets.

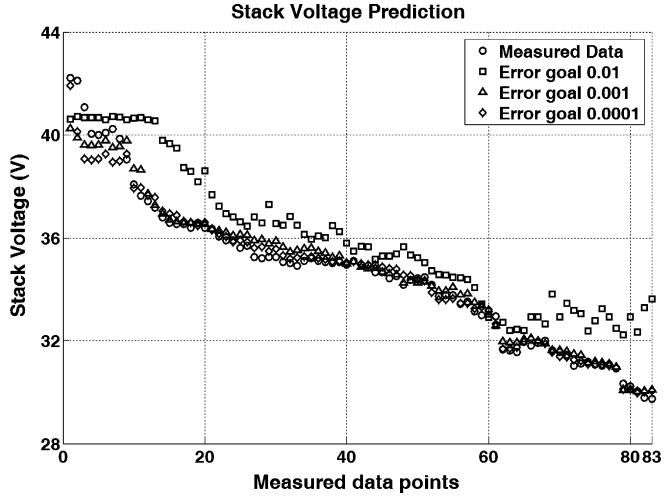

Fig. 18. Stack voltage predictions for various error goals.

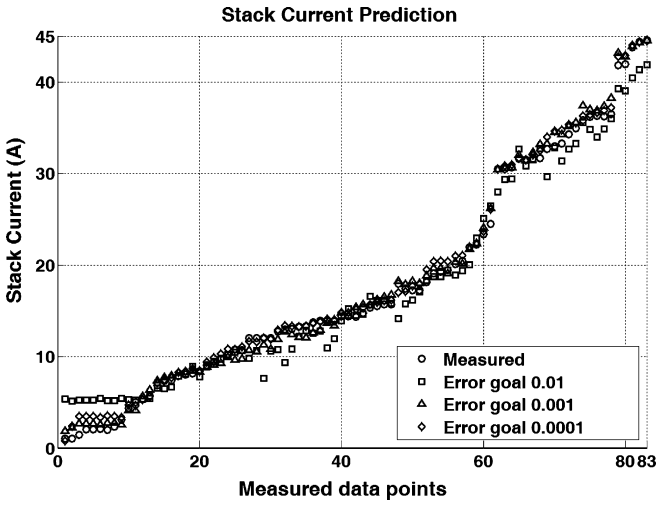

Fig. 19. Stack current predictions for various error goals.

5.1.3. Training algorithms

In this section, the investigation of various training algorithms using the normalized data and the desired error goal of 0.001 as obtained in the previous subsection for the BP network is presented. The details of these algorithms can be found in the references or MATLAB™ Help. The MATLAB™ commands in Table 4 are used to perform various training algorithms.

The predicting results from various training algorithms are shown in Table 5 and Figs. 20 and 21. Epoch 3000+ indicates that the training cannot succeed in 3000 epochs.

From the results, it can be concluded that training algorithm by using the Levenberg–Marquardt optimization provides the best prediction in term of speed and accuracy.

5.1.4. Network architectures

In this section, the normalized data, error goal of 0.001 and the training algorithm by using the Levenberg–Marquardt optimiza-

Table 4
Various training algorithms for the BP networks

Training algorithms	Network weights and bias values updated according to
traingd	Gradient descent [21]
traingdm	Gradient descent with momentum [21]
traingdx	Gradient descent momentum and an adaptive learning rate [21]
trainrp	Resilient back-propagation algorithm (RPROP) [22]
traincgf	Conjugate gradient back-propagation with Fletcher–Reeves updates [21,23]
traincgp	Conjugate gradient back-propagation with Polak–Ribiere updates [21,23]
traincgb	Conjugate gradient back-propagation with Powell–Beale restarts [24,25]
trainscg	Scaled conjugate gradient method [26]
trainbfg	BFGS quasi-Newton method [27]
trainoss	One step secant method [28]
trainlm	Levenberg–Marquardt optimization [20]
trainbr	Levenberg–Marquardt optimization with process called Bayesian regularization [29,30]

Table 5
Prediction results for various training algorithms

Training algorithm		Epoch	3–3–2			
			E_2		E_∞	
			V	I	V	I
traingd	Train	3000+	0.903	3.660	3.902	10.180
	Recall		1.119	4.650	3.528	8.780
traingdm	Train	3000+	1.432	4.275	5.562	9.905
	Recall		1.551	4.445	5.399	10.506
traingdx	Train	3000+	0.992	4.652	4.413	7.974
	Recall		0.650	0.991	2.837	2.360
trainrp	Train	3000+	0.697	1.077	3.573	3.140
	Recall		0.747	1.338	3.155	3.122
traincgf	Train	433	0.524	0.720	2.756	2.392
	Recall		0.548	0.873	2.478	2.566
traincgp	Train	968	0.587	0.801	3.456	3.031
	Recall		0.563	1.110	2.684	2.694
traincgb	Train	401	0.516	0.742	2.951	2.160
	Recall		0.530	0.858	2.412	2.383
trainscg	Train	1042	0.511	0.818	2.492	2.976
	Recall		0.537	1.280	2.356	4.158
trainbfg	Train	259	0.500	0.903	2.950	4.069
	Recall		0.527	1.439	2.393	3.285
trainoss	Train	701	0.522	0.722	2.903	2.656
	Recall		0.515	0.844	2.367	2.276
trainlm*	Train	11	0.479	0.779	2.450	2.239
	Recall		0.502	0.999	2.282	2.362
trainbr	Train	608	0.400	0.603	2.137	2.770
	Recall		0.478	0.892	2.193	2.549

tion are utilized. We investigate various network architectures by varying the number of neurons and hidden layers to determine which one tends to give a faster and better prediction performance.

The prediction results of various network architectures, as shown in Table 6 and Figs. 22 and 23, indicate that even the BP network with one hidden layer of 3 or 10 neurons can provide a good prediction in term of speed and accuracy. Therefore, a more complicated network may not necessarily perform better than a simpler one.

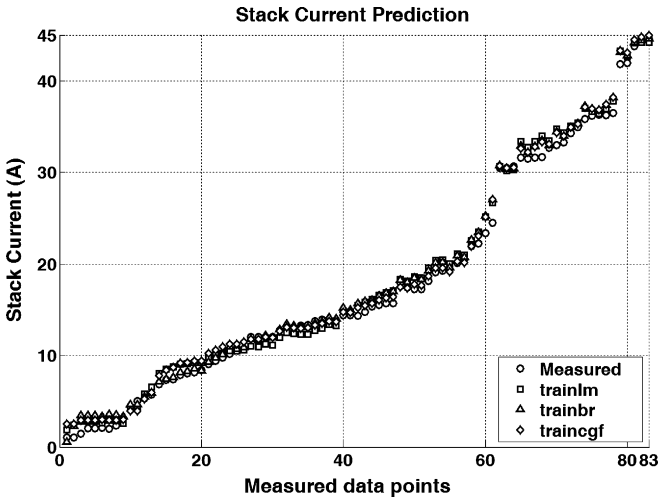

Fig. 21. Stack current predictions for various training algorithms.

Table 6
Prediction results for various network architectures

# Hidden layers		Epoch	E_2		E_∞	
			V	I	V	I
3	Train	11	0.473	0.769	2.459	2.529
	Recall		0.505	1.185	2.390	2.900
3–3	Train	11	0.464	0.907	2.291	3.342
	Recall		0.440	1.234	2.026	3.389
10	Train	6	0.449	0.960	2.300	3.056
	Recall		0.474	1.216	2.387	3.418
10–10	Train	19	0.496	0.792	2.693	3.503
	Recall		0.574	1.225	2.572	3.162
3–3–3	Train	18	0.462	1.082	2.381	5.484
	Recall		0.470	1.659	2.027	5.081

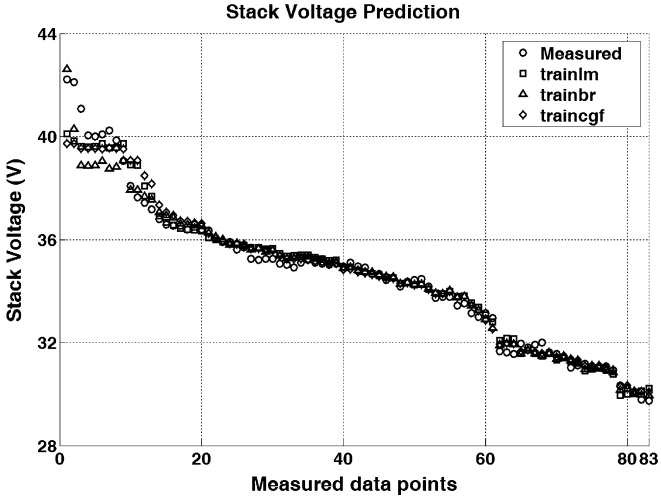

Fig. 20. Stack voltage predictions for various training algorithms.

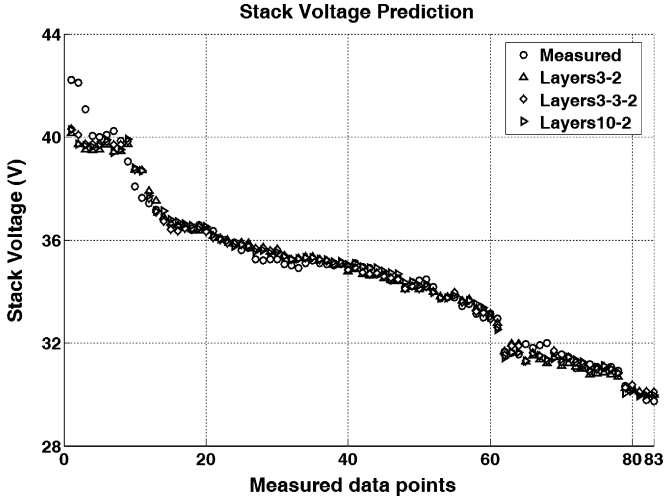

Fig. 22. Stack voltage predictions for various network architectures.

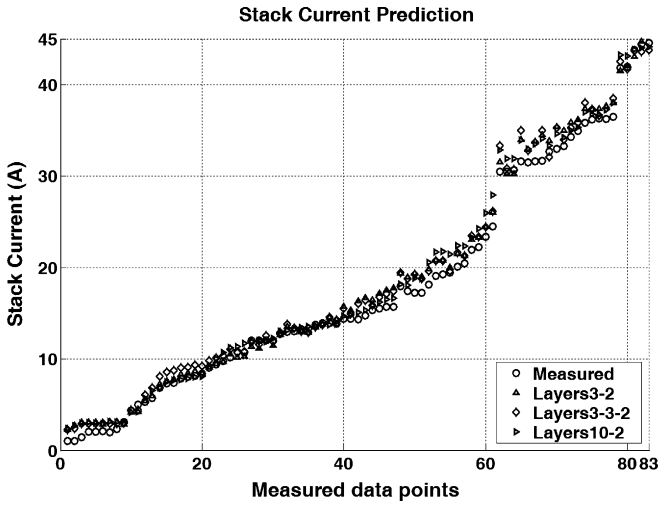

Fig. 23. Stack current predictions for various network architectures.

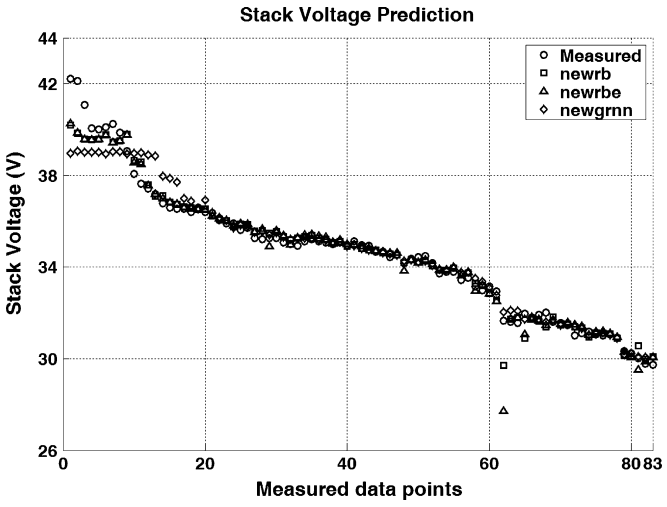

Fig. 24. Stack voltage predictions for various RBF networks.

5.2. Radial basis function (RBF) networks

Among three data sets, the normalized data provides the best result for the performance prediction, hence we use it in the subsequent RBF network investigation. Three algorithms are utilized to train the RBF networks. The algorithms are performed by using commands in MATLAB^{TM} which are "newrb", "newrbe" and "newgrnn". The details of the commands can be found in the Neural network toolbox manual [31].

In training phase, "newrb" and "newgrnn" are successful in a very short time (less than one second) while "newrbe" takes about five seconds and cannot succeed in 800 epochs. Therefore, from the prediction results shown in Table 7 and Figs. 24 and 25, it can be concluded that "newrb" provides the best predictions in term of speed and accuracy.

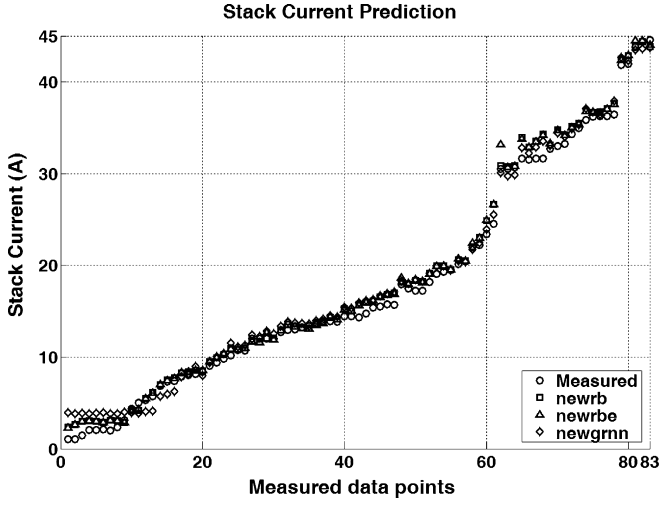

Fig. 25. Stack current predictions for various RBF networks.

5.3. Comparison between the BP and RBF networks

The best predictions from the BP and RBF networks are selected and compared as shown in Table 8 and Figs. 26 and 27. From the results, it can be concluded that the prediction performances of the BP and RBF networks are not much different in the term of speed and accuracy.

For the NN prediction of stack voltage, there are two points of RBF prediction that lie a little bit far from the measured data. This out-of-track prediction occurs where the voltage is rapidly changed (ripple points) due to load variation in load bank (Fig. 6). Since the RBF network has only one hidden layer while the BP network has two layers, it is likely that the BP network can smoothen the prediction better than the RBF network at this two particular ripple points. However, both the BP and RBF algorithms provide similar performances pertaining to remaining data sets. Therefore, it can be concluded that both BP and RBF networks can be similarly utilized for smooth data sets pertaining to fuel cell controller design and prediction applications.

Table 7
Prediction results for various RBF networks

Training algorithm		E_2		E_∞	
		V	I	V	I
newrb	Train	0.448	0.609	2.380	1.953
	Recall	0.522	0.888	2.282	2.653
newrbe	Train	0.442	0.603	2.360	2.407
	Recall	0.636	0.921	3.935	2.717
newgrnn	Train	0.689	1.008	3.501	3.179
	Recall	0.737	1.063	3.244	2.921

Table 8
Comparison of the prediction performances of the BP and RBF networks

Training algorithm		E_2		E_∞	
		V	I	V	I
BP	Train	0.479	0.779	2.450	2.239
	Recall	0.502	0.999	2.282	2.362
RBF	Train	0.448	0.609	2.380	1.953
	Recall	0.522	0.888	2.282	2.653

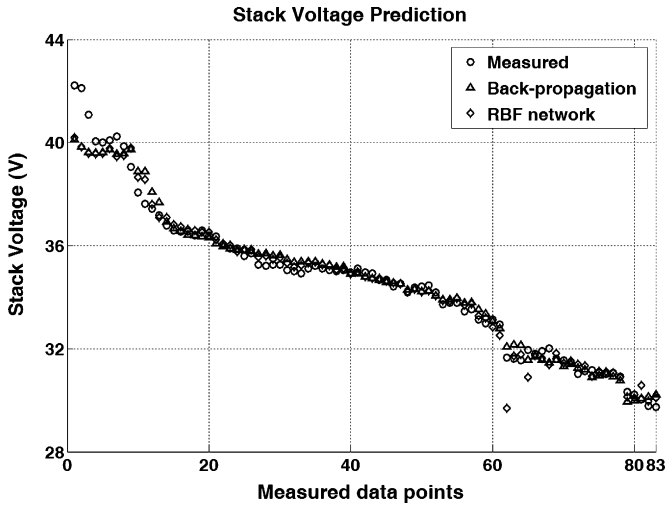

Fig. 26. Stack voltage predictions for the BP and RBF networks.

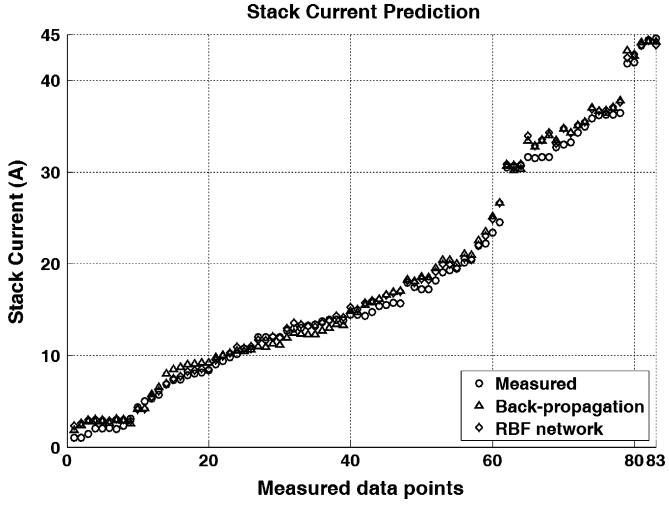

Fig. 27. Stack current predictions for the BP and RBF networks.

6. Conclusions

Performance prediction of a 1.2 kW Nexa^{TM} fuel cell system by using the BP and RBF networks is investigated. The simulation results indicate that both the BP and RBF networks can successfully predict the stack voltage and current of the fuel cell system by using only two input variables which are air flow and stack temperature. The speed of training and the prediction accuracy of NN are quite satisfactory.

Based on the results of this study, it can be concluded that if the BP or RBF network is constructed appropriately, i.e., proper input–output selection, network architecture, learning algorithms, error goal; then it is capable of predicting the performance of a particular fuel cell system with a satisfactory accuracy in a very short time period. Therefore, neural network can be an alternative approach to model a fuel cell system when the interest is in the input–output relationship, especially in the case that the physical models are not available. The derived model can be utilized for controller design for both classical [32,33] as well as soft computing methodologies [19,34,35].

References

[1] J.T. Pukrushpan, A.G. Stefanopoulou, H. Peng, Control of Fuel Cell Power Systems: Principles, Modeling, Analysis and Feedback Design, Springer, 2004.
[2] J. Larminie, A. Dicks, Fuel Cell System Explained, 2nd ed., John Wiley & Sons, 2003.
[3] J.J. Baschuk, X. Li, J. Power Sources 142 (2004) 134–153.
[4] R. Sousa Jr., E.R. Gonzalez, J. Power Sources 147 (2005) 32–45.
[5] G. Maggio, V. Recupero, L. Pino, J. Power Sources 101 (2001) 275–286.
[6] A. Rowe, X. Li, J. Power Sources 102 (2001) 82–96.
[7] S.R. Choudhury, M.B. Deshmukh, R. Rengaswamy, J. Power Sources 112 (2002) 137–152.
[8] X. Liu, W. Tao, Z. Li, Y. He, J. Power Sources 158 (2006) 25–35.
[9] K.W. Lum, J.J. McGuirk, J. Power Sources 143 (2005) 103–124.
[10] K. Bawazeer, A. Zilouchian, Neural Networks Int. Conf. 1 (1997) 157–162.
[11] S. Jemeï, D. Hissel, M.C. Péra, J.M. Kauffmann, Black-box Modeling of Proton Exchange Membrane Fuel Cell Generators, IEEE paper, 2002.
[12] S. Ou, L. Achenie, J. Power Sources 104 (2005) 319–330.
[13] A. Zilouchian, M. Jamshidi, Intelligent Control Systems Using Soft Computing Methodologies, CRC Press, 2000.
[14] Nexa^{TM} Power Module User's Manual, Ballard Power Systems Inc., 2003.
[15] Z. Guo, R.E. Uhrig, International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92) (1992) 223–234.
[16] S. Shervais, M. Zwick, Proceedings of the International Joint Conference on Neural Networks (2003) 3022–3026.
[17] K. Li, Proceedings of the 5th World Congress on Intelligent Control and Automation (WCICA) (3) (2004) 2018–2021.
[18] S.S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, Upper Saddle River, NJ, 1999.
[19] G.W. Irwin, K. Warwick, K.J. Hunt, Neural Network Applications in control, IEE (1995).
[20] M.T. Hagan, M. Menhaj, IEEE Trans. Neural Networks 5 (6) (1999) 989–993.
[21] M.T. Hagan, H.B. Demuth, M.H. Beale, Neural Network Design, PWS Publishing, Boston, MA, 1996.
[22] M. Reidmiller, H. Braun, Proceedings of the IEEE International Conference on NN (ICNN), San Francisco, 1993, pp. 586–591.
[23] R. Fletcher, C.M. Reeves, Computer Journal 7 (1964) 149–154.
[24] E.M.L. Beale, Numerical methods for nonlinear optimization, Academic Press, London, 1972.
[25] M.J.D. Powell, Math. Prog. 12 (1977) 241–254.
[26] M.F. Moller, Neural Networks 6 (1993) 525–533.
[27] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[28] R. Battiti, Neural Comput. 4 (2) (1992) 141–166.
[29] Foresee, Hagan, Proceedings of the International Joint Conference on Neural Networks, 1997.
[30] P.D. Wasserman, Advanced Methods in Neural Computing, Van Nostrand Reinhold, New York, 1993, pp. 155–161.
[31] Neural Networks Toobox, MATLAB^{TM}.
[32] S.S. Ge, J. Zhang, IEEE Trans. Neural Networks 14 (4) (2003) 900–918.
[33] H. Hu, P. Woo, IEEE Trans. Ind. Electron. 53 (3) (2006) 929–940.
[34] R. Fuller, Introduction to Neuro-Fuzzy Systems, Physica-Verlag, 2000.
[35] E. Czogala, J. Leski, Fuzzy and Neuro-Fuzzy Intelligent Systems, Physica-Verlag, 2000.