

Short communication

Modeling a SOFC stack based on GA-RBF neural networks identification

Xiao-Juan Wu*, Xin-Jian Zhu, Guang-Yi Cao, Heng-Yong Tu

Institute of Fuel Cell, Shanghai Jiao Tong University, Shanghai 200030, China

Received 11 December 2006; received in revised form 14 January 2007; accepted 25 January 2007

Available online 6 February 2007

Abstract

In this paper, a nonlinear offline model of the solid oxide fuel cell (SOFC) is built by using a radial basis function (RBF) neural network based on a genetic algorithm (GA). During the process of modeling, the GA aims to optimize the parameters of RBF neural networks and the optimum values are regarded as the initial values of the RBF neural network parameters. Furthermore, we utilize the gradient descent learning algorithm to adjust the parameters. The validity and accuracy of modeling are tested by simulations. Besides, compared with the BP neural network approach, the simulation results show that the GA-RBF approach is superior to the conventional BP neural network in predicting the stack voltage with different temperature. So it is feasible to establish the model of SOFC stack by using RBF neural networks identification based on the GA.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Solid oxide fuel cells (SOFCs); Radial basis function (RBF); Neural networks; Genetic algorithms; Identification

1. Introduction

Unlike other fuel cells, the solid oxide fuel cell (SOFC) is entirely solid state with no liquid components and works in a complicated high-temperature (600–1000 °C) environment. Due to its higher energy conversion efficiency than those conventional heat engine systems and other types of fuel cells, the SOFC is the promising candidate for future energy conversion systems.

In the last several decades, fruitful results from SOFC stack modeling have been obtained. However, most of the existing models focus on the design of the SOFC instead of its applications. What matters most to SOFC users, however, are not its relevant internal details but its performance under different operating conditions. What they really need are behavioral models, with which they can predict the SOFC behavior under various operating conditions.

Motivated by this need, we decided to make a modeling study of the SOFC system by using a radial basis function (RBF) neural network based on a genetic algorithm (GA). Neural networks are considered as an attractive structure to establish the

mathematical relationship of the dynamic system based on the input–output data. A RBF neural network is a feed-forward neural network with one hidden layer and can uniformly approximate any continuous function to a prospected accuracy [1]. However, a key problem by using the RBF neural network approach is about how to choose the optimum initial values of the following three parameters: the output weights, the centers and widths of the hidden unit. If they are not appropriately chosen, the RBF neural network may degrade validity and accuracy of modeling. To assure the optimal performance of the RBF neural network approach for SOFC modeling, we consider applying a genetic algorithm to optimize the RBF neural network parameters in this study. Genetic algorithms are a kind of self-adaptive global searching optimization algorithm based on the mechanics of natural selection and natural genetics [2]. Different from conventional optimization algorithms, genetic algorithms are based on population, in which each individual is evolved parallel, and the ultimate result is included in the last population.

This paper is organized as follows. Section 2 briefly discusses various existing SOFC models. In Section 3, a RBF neural network based on a genetic algorithm for nonlinear system modeling is explained. In Section 4, identification structure of the SOFC stack and the processes of training and testing

* Corresponding author. Tel.: +86 21 34203657; fax: +86 21 62932154.
E-mail address: xj_wu@sjtu.edu.cn (X.-J. Wu).

the GA-RBF model are presented in detail. Conclusions and suggestions for future work are summarized in the last section.

2. Existing SOFC models

In this section, we briefly review the various existing SOFC models and discuss their advantages and disadvantages. A SOFC consists of an interconnected structure and a three-layer region composed of two ceramic electrodes, anode and cathode, separated by a dense ceramic electrolyte. In the fuel, the oxygen ions formed at the cathode migrate through the ion-conducting solid ceramic electrolyte to the anode/electrolyte interface where they react with the hydrogen and carbon monoxide contained in the fuel, producing water and carbon dioxide while liberating electrons that flow back to the cathode/electrolyte interface via an external circuit [3]. A single cell produces an open-circuit voltage of approximately 1 V, so cells have to be connected together in a series arrangement to form a stack.

Some analytical SOFC models have been put forward, including, simplified zero-dimensional [4] and one-dimensional models [5–8], two-dimensional models [9–12] and more complex three-dimensional models [13,14]. Although these models can help analyze and optimize the SOFC, they have some limits. Most of the models are based on mass, energy and momentum conservation laws, so their expressions are too complicated to be suitable for engineering applications.

An empirical modeling approach will be more practical for SOFC users, from which they can deduce SOFC stack responses without knowledge of the internal details. To meet the demands, some researchers have attempted to establish novel SOFC models. Arriagada et al. [15] utilized artificial neural network (ANN) methodology to derive a SOFC model. Highly efficient as the model is, however, its practical design suffers from some drawbacks, such as the existence of local minima and over-fitting, choice of the number of hidden units, etc. A least squares support vector machine (LS-SVM) was used to build the model of a SOFC stack by Huo et al. [16]. LS-SVM is a modification of SVM and possesses many advantages. However, this paper only considered the fuel utilization, which had an effect on the cell voltage, and did not think about the other factors.

In this work, a RBF neural network based on a genetic algorithm is employed to establish a black-box model for the SOFC. In the following sections, the GA-RBF modeling method will be presented in detail.

3. GA-RBF neural network for nonlinear system modeling

A RBF neural network has an input layer, a nonlinear hidden layer and a linear output layer. The nodes within each layer are fully connected to the previous layer nodes. The input variables are each assigned to nodes in the input layer and connected directly to the hidden layer without weights. The hidden layer nodes are RBF units. The nodes calculate the Euclidean distances between the centers and the network input vector, and pass the results through a nonlinear function [17]. The output layer nodes are weighted linear combinations of the RBF in hid-

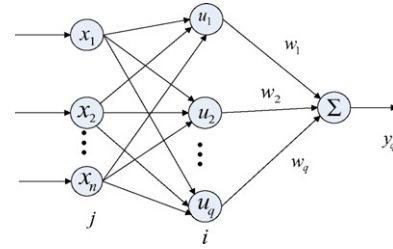


Fig. 1. The structure of RBF neural networks.

den layer. The structure of a RBF neural network with n inputs, one output and q hidden nodes is given in Fig. 1.

Where, input $x = [x_1, x_2, \dots, x_n]^T$ and $w = [w_1, w_2, \dots, w_q]^T$ is the neural network weight. u_i is a nonlinear function and here, it is chosen as a Gaussian activation function

$$u_i = \exp \left[-\frac{(x - c_i)^T(x - c_i)}{2b_i^2} \right] \quad (i = 1, 2, \dots, q) \quad (1)$$

where $c_i = (c_{i1}, c_{i2}, \dots, c_{in})^T$, $j = 1, 2, \dots, n$, is the center of the i th RBF hidden unit, and b_i is the width of the i th RBF hidden unit. Then the i th RBF network output can be represented as a linearly weighted sum of q basis functions

$$y_q(k) = \sum_{i=1}^q w_i u_i = \sum_{i=1}^q w_i \exp \left[-\frac{(x - c_i)^T(x - c_i)}{2b_i^2} \right] \quad (2)$$

Let $y(k)$ represent the target value of the network at time k . The error of the network at time k is defined as:

$$e(k) = y(k) - y_q(k) \quad (3)$$

The cost function of the network is the squared error between the target and the predicted values, which is given by the following equation:

$$E(k) = \frac{1}{2} [e(k)]^2 \quad (4)$$

The learning algorithm aims to minimize the squared error using a gradient descent procedure. Hence, the change of the output weight w_i , the centers c_{ij} and the widths b_i is determined according to the following equation:

$$w_i(k) = w_i(k-1) + \eta \Delta w_i + \alpha (w_i(k-1) - w_i(k-2)) \quad (5)$$

$$c_{ij}(k) = c_{ij}(k-1) + \eta \Delta c_{ij} + \alpha (c_{ij}(k-1) - c_{ij}(k-2)) \quad (6)$$

$$b_i(k) = b_i(k-1) + \eta \Delta b_i + \alpha (b_i(k-1) - b_i(k-2)) \quad (7)$$

where α is the momentum term and η is the learning rate, $\alpha \in [0, 1]$, $\eta \in [0, 1]$. The term Δw_i , Δc_{ij} and Δb_i are defined as:

$$\Delta w_i = \frac{\partial E}{\partial w_i} = (y(k) - y_q(k)) u_i = (y(k) - y_q(k)) \times \exp \left[-\frac{(x - c_i)^T(x - c_i)}{2b_i^2} \right] \quad (8)$$

$$\Delta c_{ij} = \frac{\partial E}{\partial c_{ij}} = (y(k) - y_q(k))w_i \frac{x_j - c_{ij}}{b_i^2} \quad (9)$$

$$\Delta b_i = \frac{\partial E}{\partial b_i} = (y(k) - y_q(k))w_i u_i \frac{\|x - c_i\|^2}{b_i^3} \quad (10)$$

When we program to realize the RBF algorithm, how to choose the optimum initial values of the following three parameters in Eqs. (5)–(7): the output weight w_i , the centers c_{ij} and the widths b_i , is very important. If they are not appropriately chosen, the RBF neural network may degrade validity and accuracy of modeling. So a genetic algorithm is used to optimize the RBF neural network parameters.

A genetic algorithm is an interactive procedure that maintains a population of strings which constitute the set of candidate solutions to the specific problem [18]. During each generation, the strings in the current population are rated for their effectiveness as solutions. On the basis of these evaluations, a new population of candidate solutions is formed by using genetic operations, such as selection, crossover and mutation. There are four major steps required to use the genetic algorithm to solve a problem, include, coding, evaluation of fitness, genetic operations and the terminate criterion.

4. Modeling SOFC by GA-RBF

For a given SOFC stack, the relation between terminal voltage U and current density I is influenced by many operating parameters, such as cell temperature, air flow rate, hydrogen flow rate, air pressure, hydrogen pressure, etc. However, due to the high number of operating variables, a complete experimental database of SOFC under the different operating conditions is difficult to obtain and no data are available in the open literature yet [19]. Up to now, almost no model has ever been able to accommodate all these operating variables. Our GA-RBF model is no exception. Temperature is one of the most operating parameters for the fuel cell and has a significant effect on the fuel cell. In order to analyze the effects of different temperatures on output voltage, we choose current density I , which is decided by the uncontrollable load, and cell temperature T as variables.

In general, a wide class of nonlinear systems can be described by nonlinear autoregressive model with exogenous inputs (NARX) [20]. So in this paper the SOFC nonlinear system with two inputs and one output can be described as follows:

$$U(k+1) = f[U(k), U(k-1), \dots, U(k-n), I(k), I(k-1), \dots, I(k-m), T(k)] \quad (11)$$

Supposing there is a series of inputs $I(k), I(k-1), \dots, I(k-m), T(k)$ and outputs $U(k), U(k-1), \dots, U(k-n)$. The identification structure based on GA-RBF is shown in Fig. 2, where TDL is the tapped delay line. The aim of our study is thus, to find an GA-RBF model that approximates Eq. (11). And it requires three steps to build an efficient GA-RBF model: preparing training data, training the data to obtain a GA-RBF model and predicting the new input data with the obtained model.

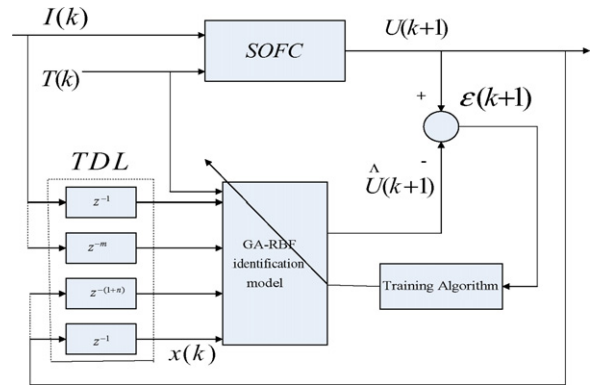


Fig. 2. Identification structure of SOFC stack based on GA-RBF.

4.1. Preparing training data

In our study, a model in Ref. [21] is used to generate the data required for the training of the GA-RBF model. Here, two groups of current density and cell voltage data at 800 °C and 1000 °C are chosen as training data, and each group has 701 pairs of data. Main operational parameters of SOFC are varied, such as temperatures (600–1000 °C), stack current density (0–700 mA cm⁻²). In most cases, training data should be scaled, normally linearly, to [0, 1] or [-1, +1]. An example of scaled current density is shown in Table 1. Scaling can increase the training speed and assist in selecting GA-RBF parameters. In this paper, all the data, including, cell voltage, current density and temperature, are scaled to [0, 1] by Eq. (12).

$$x' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad (12)$$

4.2. Selection of the optimal GA-RBF parameters

In order to reduce the number of the parameters and improve the speed of program debug, the hidden layer of the RBF neural network is chosen 3 nodes. There are two inputs (current density I and cell temperature T) and one output (voltage U), so the structure of the RBF neural network is chosen 2-3-1. i.e. let the RBF neural network consists of input layer with 2 nodes, 1 hidden layer with 3 nodes and output layer with 1 node.

4.2.1. Coding structure

Coding aims to build the relationship between the problem and the individual in genetic algorithms. If the problems are expressed by coding strings, these strings are called an indi-

Table 1
An example of cell current density scaled to [0, 1]

Unscaled (mA cm ⁻²)	Scaled
0	0
100	0.143
150	0.214
300	0.429
450	0.643
550	0.786
700	1

vidual or a chromosome. A population contains a number of individuals. Generally, the population size n is chosen from 30 to 100. In order to save the running time, here, the size n is chosen 30. Each individual represents a variable or a part of the problem which is needed to be optimized. In this paper, the parameters of RBF neural networks are needed to be optimized by the GA. So these individuals represent the widths and centers of the Gaussian function and the output weights, and the representation of an individual is

$$p = [b_1 \ b_2 \ b_3 \ c_{11} \ c_{12} \ c_{13} \ c_{21} \ c_{22} \ c_{23} \ w_1 \ w_2 \ w_3] \quad (13)$$

Here, each individual consists of 12 parameters, i.e. 3 widths and 6 centers of the hidden unit of the RBF neural network and 3 connection weights. Each parameter in the individual is expressed by a two-decimal coding of 10 bits.

All the width of the RBF hidden are chosen on the interval [0.1, 3] and the centers of the RBF hidden unit are chosen on the interval [−3, 3]. And all weights are chosen on the interval [−1, 1].

4.2.2. Fitness function evaluation

All individuals of one generation are evaluated by a fitness function. When using a genetic algorithm to solve a problem, the problem is represented by a string and an evaluation function is defined. The evaluation function uses the value of the string as a parameter to evaluate the results of the problem. Each string is evaluated through the evaluation function and the new generation is formed by using the specific genetic operators. Here, a RBF neural network is used to model a SOFC stack. The value of the goal function is littler, and then the precision is higher. To get a higher regression precision, the goal function is defined as follows:

$$J = 50 \sum_{i=1}^N |e(i)| \quad (14)$$

Here, $e(i)$ is the error between the experimental output and the model output. Generally, the fitness function is defined as the reciprocal of the goal function, so we adopt the fitness function as below:

$$f = \frac{1}{J} = \frac{1}{50 \sum_{i=1}^N |e(i)|} \quad (15)$$

4.2.3. Genetic operations

There are mainly three genetic operations, including, selection, crossover and mutation operations. These genetic operations have key effects on the performances of the genetic algorithm. In this study, we use the roulette-wheel selection method—a simulated roulette is spun—for this selection pro-

cess. The response fitness value of every individual is p_i ($i = 1, 2, \dots, N$). According to the p_i , a roulette wheel is divided into N parts. In the selection operation, spinning the roulette wheel, if a consulted point lies in the i th sector, we will choose the i th individual. Obviously the area of the sector is larger, and then the probability that the consulted point lies in the sector, is more. This indicates that the better an individual's fitness is, the more likely it is to be selected. An individual is probabilistically selected from the population on the basis of its fitness and the selected individual is then copied into the next generation of the population without any change.

Selection directs the search toward the best existing individuals but does not create any new individuals. In nature, an offspring has two parents and inherits genes from both. The main operator working on the parents is the crossover operator, the operation of which occurred for a selected pair with a crossover rate p_c that was set to 0.8 in this study. In each new population, there are $p_c \times n$ individuals which are needed crossover operations. Here, n is the population size. In the crossover step, we also keep the same number of chromosomes for each group. After this operation, the individuals with poor performances are replaced by the newly produced offspring.

Although selection and crossover will produce many new strings, they do not introduce any new information to the population at the site of an individual. Mutation is an operator that randomly alters the allele of a gene. With mutation, new genetic materials can be introduced into the population. In each new population, there are $p_m \times n \times L$ individuals which are needed mutation operations. Here, p_m is the mutation probability, n the population size and L is the string length. According to the above analyses, we know that every individual consists of 12 parameters and each parameter in the individual is expressed by a two-decimal coding of 10 bits. So L is 120. In the paper, mutation probability p_m is chosen $0.003 - [1:1:\text{size}] \times 0.003/\text{size}$. This indicates that the more little an individual's fitness is, the more likely it is to be mutated.

4.2.4. The terminate criterion

There are usually two criterions for terminating a run. The first criterion is deciding the maximum generation previously, and the second is that the process continues until the fitness function has no change. Here, we choose the first criterion and the maximum generation is chosen 100.

After 100 times genetic, the optimized initial values of the parameters are shown in Table 2. The optimization process of the goal function J in Eq. (14) is shown in Fig. 3 and the optimal value J is 121.3089.

After the optimized initial values of the three parameters are obtained, we utilize the gradient descent learning algorithms to adjust them, which can be seen in Eqs. (5)–(10). By tuning, the

Table 2
The optimized widths and centers of the Gaussian function and the optimized output weights

b_1	b_2	b_3	c_{11}	c_{12}	c_{13}	c_{21}	c_{22}	c_{23}	w_1	w_2	w_3
1.9880	1.5146	2.8724	0.3724	−0.8416	−0.5073	−1.7918	1.6276	−2.7889	−0.4174	0.8397	0.8690

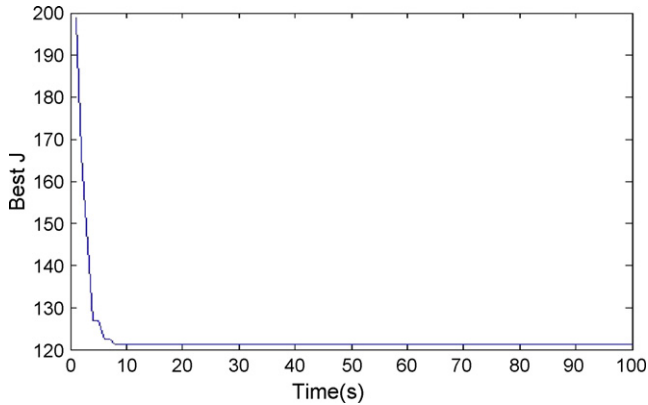


Fig. 3. The goal function curve of the best individual with the population evolving.

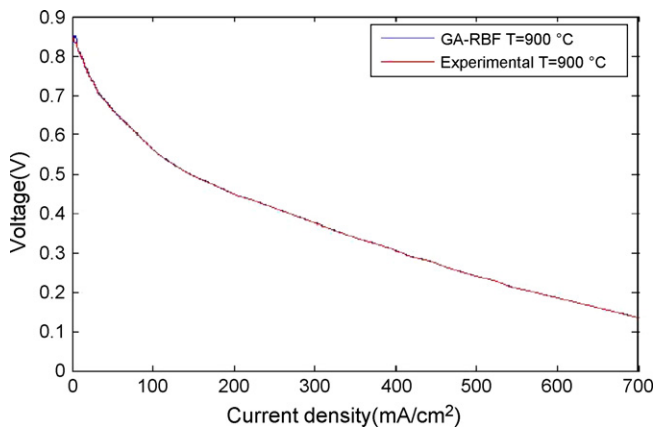


Fig. 4. Voltage–current density characteristics: predicted by GA-RBF model and experimental at $T=900\text{ }^{\circ}\text{C}$.

momentum term α is chosen 0.6 and the learning rate η is chosen 0.53.

4.3. Predicting with the GA-RBF model

After training, a GA-RBF model is obtained, which can be used to predict new input dates. In our study, the testing data is also chosen from the above-mentioned model in Ref. [21]. The cell voltage at $900\text{ }^{\circ}\text{C}$ with the current density in the range from

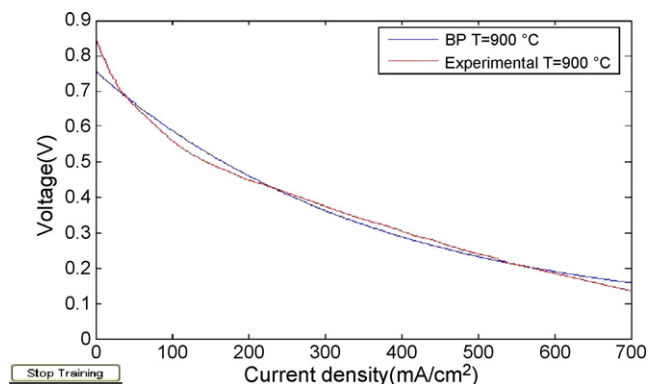


Fig. 5. Voltage–current density characteristics: predicted by BP model and experimental at $T=900\text{ }^{\circ}\text{C}$.

0 to 700 mA cm^{-2} is predicted. And a comparison between the predicted data and the experimental data is made to evaluate the model's prediction precision, which is shown in Fig. 4. At the same time, a BP neural network model is also used to predict the stack voltage at $900\text{ }^{\circ}\text{C}$. Via the toolbox of MATLAB 7.0, the predicted result is shown in Fig. 5. Comparing Figs. 4 and 5, we can see clearly that the precision is greatly improved. It indicates that GA-RBF is a powerful tool for modeling SOFC and our GA-RBF model presented in this paper is accurate and valid.

5. Conclusions

An offline modeling study of a SOFC stack using a GA-RBF neural network is reported in this paper. It is shown that the GA-RBF model is an attractive modeling solution in that it avoids using complicated differential equations to describe the stack, and the input–output characteristics can be achieved quickly by GA-RBF estimation. In our study, training a GA-RBF with the optimized parameters only needs 10 s on a PIII 256 MHz computer and the time for prediction is no more than 5 s. Besides, compared with the BP neural network approach, the simulation results show that the GA-RBF approach yields higher prediction accuracy. Hence, it is feasible to establish the model of the SOFC by using GA-RBF.

Among all the operating parameters that have an effect on the SOFC performance, only current density and temperature are included in our model. In the future we will incorporate other operating parameters into the GA-RBF model, and based on this GA-RBF model, some control scheme studies, such as predictive control and robust control will be developed.

Acknowledgement

This work is supported by National 863 Scientific Project Development Funds (no. 2003AA517020), PR China.

References

- [1] K. Warwick, Proceedings of the 35th Conference on Decision and Control, Kobe, Japan, December, 1996, pp. 464–469.
- [2] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
- [3] J.R. Ferguson, J.M. Fiard, R. Herbin, J. Power Sources 58 (1996) 109–122.
- [4] P. Lunghi, U. Ubertini, Proceedings of the Seventh International Symposium on Solid Oxide Fuel Cells (SOFC-VII), Tsukuba, Ibaraki, Japan, 2001, pp. 254–264.
- [5] T. Ota, M. Koyama, C. Wen, K. Yamada, H. Takahashi, J. Power Sources 118 (1–2) (2003) 430–439.
- [6] P.W. Li, K. Suzuki, J. Electrochem. Soc. 151 (4) (2004) A548–A557.
- [7] M. Yokoo, T. Take, J. Power Sources 137 (2) (2004) 206–215.
- [8] R. Bove, P. Lunghi, N.M. Sammes, Int. J. Hydrogen Energy 30 (2) (2005) 189–200.
- [9] Bharadwaj S A., D.H. Archer, E.S. Rubin, J. Fuel Cell Sci. Technol. 2 (1) (2005) 38–44.
- [10] A. Bharadwaj, D.H. Archer, E.S. Rubin, J. Fuel Cell Sci. Technol. 2 (1) (2005) 52–59.
- [11] E. Arato, P. Costa, Mathematical modeling of monolithic SOFC, SOFC Stack Design Tool, International Energy Agency Final Report, 1992.
- [12] P. Nehter, J. Power Sources 157 (1) (2006) 325–334.

- [13] K.P. Recknagle, R.E. Williford, L.A. Chick, D.R. Rector, M.A. Khaleel, J. Power Sources 113 (1) (2003) 109–114.
- [14] D. Herbstritt, A. Weber, E. Ivers-Tiffée, J. Eur. Ceram. Soc. 21 (10–11) (2001) 1813–1816.
- [15] J. Arriagada, P. Olausson, A. Selimovic, J. Power Sources 112 (2002) 54–60.
- [16] H.-B. Huo, X.-J. Zhu, G.-Y. Cao, J. Power Sources (2006).
- [17] A. Ai-Amoudi, L. Zhang, IEE Pro.-Gener. Transm. Distrib. 147 (5) (2000) 310–316.
- [18] Y. Gao, L. Shi, P.J. Yao, Proceedings of the 3rd World Congress on Intelligent Control and Automation June 28–July 2, Hefei, PR China, 2000.
- [19] P. Costamagna, L. Magistri, A.F. Massardo, J. Power Sources 96 (2001) 352–368.
- [20] J. Sjöberg, Q.H. Zhang, L. Ljung, et al., Automatica 31 (12) (1995) 1691–1724.
- [21] F. Calise, M. Dentice d'Accadia, A. Palombo, L. Vanoli, Energy 31 (2006) 3278–3299.