Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Power Sources 124 (2003) 479–486
JOURNAL OF POWER SOURCES
www.elsevier.com/locate/jpowsour

On-board fuel cell power supply modeling on the basis of neural network methodology

S. Jemeï*, D. Hissel, M.C. Péra, J.M. Kauffmann

Laboratory of Electrical Engineering and Systems, UTBM-UFC Research Unit Associated to the INRETS, UTBM Bat. F technopole, 90010 Belfort, France

Received 24 January 2003; received in revised form 9 June 2003; accepted 30 June 2003

Abstract

Proton exchange membranes are one of the most promising fuel cell technologies for transportation applications. Considering the aim of transportation applications, a simulation model of the whole fuel cell system is a major milestone. This would lead to the possibility of optimizing the complete vehicle (including all ancillaries, output electrical converter and their dedicated control laws). In a fuel cell system, there is a strong relationship between available electrical power and actual operating conditions: gas conditioning, membrane hydration state, temperature, current set point . . . Thus, a "minimal behavioral model" of a fuel cell system able to evaluate the output variables and their variations is highly interesting. Artificial neural networks (NN) are a very efficient tool to reach such an aim. In this paper, a proton exchange membrane fuel cell (PEMFC) system neural network model is proposed. It is implemented on Matlab/Simulink® software and will be integrated to a complete vehicle powertrain. Thus, it will be possible to carry out the development and the simulation of the control laws in order to drive energy transfers on-board fuel cell vehicles.

Keywords: Fuel cell modeling; Artificial neural network; PEM fuel cell; Automotive applications

1. Introduction

Proton exchange membrane fuel cells (PEMFC) are well-known candidates for low-emission vehicle propulsion and are currently being researched extensively around the world [1]. Independent of the problems of integration and cost, the use of PEMFC in a vehicle requires full control of its behavior and thus a highly efficient model.

Modeling is usually done with complex models based on the knowledge of physicochemical phenomena. These models require a good knowledge of the process' parameters [2,3]. In most cases, these parameters are difficult to determine for fuel cell systems. However, it is possible to achieve behavioral modeling and to avoid having to identify all these parameters by using "black-box" models.

The aim of this paper consists in proposing such a model of a fuel cell system based on proton exchange membrane technology by using neural networks (NN). In our case, a 500 W fuel cell is considered but the proposed methodology can easily be extrapolated to more powerful fuel cell systems.

In the first part of this paper, a general description of the architecture of a PEMFC system [4] is presented. In the second part, the neural network modeling applied to fuel cell systems is proposed. Finally, simulation results obtained using the proposed NN model are compared to experimental findings.

2. PEM fuel cell systems

The fuel cell system considered consists of a proton exchange membrane fuel cell that can be operated with pure or reformed hydrogen on the anode side and with air or oxygen on the cathode side. The experimental stack is here only a low-power stack (500 W). This stack is made up of 20 cells (active area 100 cm^2) and is able to supply electrical power of up to 500 W under 12 V. The operating temperature ranges from 20 to 80 °C. Additionally to the anode and cathode gas circuits, a coolant deionized water circuit is used to extract the calories from the stack. Fig. 1 shows a view of this stack.

As is well known, many different kinds of physical phenomena (electrochemical, thermodynamical, thermal, etc.) are involved in the performance of the stack. Furthermore,

* Corresponding author. Tel.: +33-3-84-58-36-28; fax: +33-3-84-58-36-36.
E-mail addresses: samir.jemei@utbm.fr (S. Jemeï), daniel.hissel@utbm.fr (D. Hissel), marie-cecile.pera@utbm.fr (M.C. Péra), jean-marie.kauffman@univ-fcomte.fr (J.M. Kauffmann).

0378-7753/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0378-7753(03)00799-7

Fig. 1. PEM fuel cell stack.

the electrical response of the fuel cell depends not only on the stack but also on the response of the ancillaries that are located around the stack (Fig. 2).

Considering the air circuit, a compressor must be placed at the inlet of the fuel cell so as to provide oxygen for the electrochemical reaction and to raise its efficiency by increasing the air pressure. Considering the hydrogen circuit, hydrogen can be stored directly on-board or supplied through a reforming system. Special care must also be taken with the water management system. In fact, the proton exchange membrane must always be maintained in a well-hydrated state so as to ensure the migration of protons H^+ from the anode to the cathode. Moreover, in most cases, there is a special water coolant circuit within the stack. The aim is to obtain a completely self-sufficient system with respect to water. Finally, an electrical converter must be placed at the outlet of the fuel cell, providing the electrical power to the dc electrical bus of the vehicle [5].

All these fuel cell ancillary systems have to be taken into account by the modeling. This results in a high complexity level model based on a huge number of parameters that have to be evaluated for the real system. Moreover, some parameters cannot be easily measured on the real system in experimental tests. For example, fuel cell behavior is greatly dependent on size and shape of gas channels. These two parameters can reduce or increase the gas diffusion (O_2, H_2 and H_2O) ability, and therefore the achievable power level of the fuel cell system. Another important example is the membrane hydration level state. When this membrane dries out, the available electrical power decreases rapidly; when it is flooded, the result is exactly the same. The above mentioned parameters, despite their important influence on the fuel cell system behavior, are not only difficult to evaluate, but are also not systematically provided by the stack manufacturers, as a part of their own know-how.

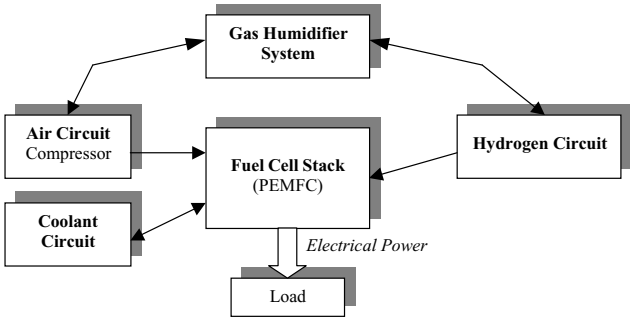

Fig. 2. General design of a PEM fuel cell system.

In conclusion, many fuel cell stack parameters are unknown for the user but they must be taken into account when considering analytical behavioral models.

Modern non-linear modeling methods are efficient tools to overcome this difficult identification step for both the system and its components. Among them, black-box models are used for small and powerful modeling systems. These are also accurate enough for simulation of the whole system for automotive applications, and for implementation on hardware or on microcontroller-based structures.

The modeling approach of the whole PEM fuel cell generator consists in proposing only a black-box model constituted on the basis of a neural network. Thus, it will be possible to carry out the development and the simulation of the command laws intended to manage the energy transfers on-board a fuel cell vehicle.

3. Fuel cell neural network model

3.1. Neural network modeling

Artificial neural networks are models of information processing inspired by the biology of the brain. Their basic unit is the artificial neuron. The neuron receives numerical information from input nodes, processes it internally, and gives out an output. The processing is done in two stages: first, each unit input is weighted and added, and then the result is used as the argument of a linear or non-linear activation function f. For the model proposed in this paper, logistic activation functions ($f(x) = 1/(1+e^{-x})$) are used for hidden layer neurons and identity activation function for the output neuron ($f(x) = x$) [6]. This choice results from various tests carried out with different activation functions on each layer.

The topology of the network is defined by the organization of the neurons. The most commonly used topology is the multilayer perceptron (MLP) type, in which the neurons are organized in layers [7,8]. In the case of a feed-forward architecture, the outputs of a layer are the inputs of the following one. Thus, the outputs of the network are those of the last layer. The layers between the input nodes and the output layer are called hidden layers. Fig. 3 shows an example of the topology of the neural network. It is composed of series of input nodes, two hidden layers and one output layer.

If logistic functions are used for the activation of the hidden layers and linear functions for the activation of the output layer, this network (Fig. 3) is equivalent to the model given by (1). It shows how complex but also how flexible a network can be [6,8],

$$y = \sum_{m=1}^{1} \left[\sum_{k} \left[W^{o}_{m,k} \frac{1}{1+e^{-A_1}} \right] \right] + b_m \quad (1)$$

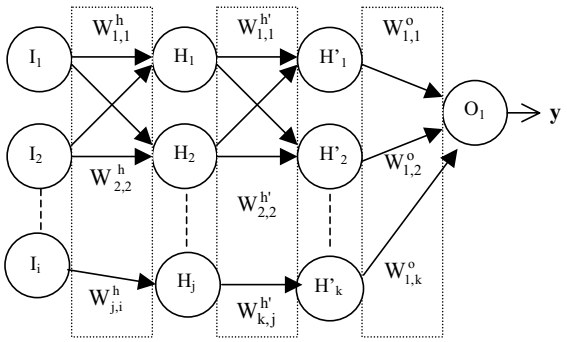

Fig. 3. Example of a feed-forward neural network. I: inputs, H and H′: hidden neurons, O: output neurons, $W^{h}_{j,i}$: weights between hidden neuron j and input i, $W^{h'}_{k,j}$: weights between hidden neuron j and hidden neuron k, $W^{o}_{m,k}$: weights between hidden neuron k and output neuron m.

with

$$A_1 = \left[\sum_j \left[W^{h'}_{k,j} \frac{1}{1 + e^{(-\sum_i W^{h}_{j,i} I_i + b_j)}} \right] \right] + b_k \qquad (2)$$

where

- $W^{h}_{k,j}$ is the matrix containing the weights that connect neuron j to input i;
- $W^{h'}_{k,j}$: weights between neuron k and neuron j;
- $W^{o}_{m,k}$: weights between neuron k and output m;
- b_j, b_k, b_m are bias vectors (not shown in Fig. 3).

These different matrix and vectors represent the different parameters of the network that are to be tuned. This is done by the training process.

In our case, it is done by minimizing the following quadratic cost function of the output error (3). This error is calculated by comparing the output values of the network and the desired output values (obtained by experimental tests with the real system).

$$J(W) = \frac{1}{2} \sum_{N=1}^{M} [T_N - Y_N(W)]^2 \qquad (3)$$

T is the target output value; Y the output value; M the number of patterns; W is the weight value.

Many optimization methods have been adapted for this task but the most commonly used is the back propagation algorithm. It is an optimization technique designed to minimize the cost function. It uses a steepest descent method that performs stochastic gradient descent on the error surface (defined by (3)) with respect to the network parameters [9].

3.2. Neural network used for fuel cell generator modeling

The low-power fuel cell test bench (Fig. 4) as described in the previous part of this paper is used for the experimental validation of our neural model. It has to be underlined that the test bench presented in Fig. 4 is, of course, only a research laboratory test bench, with many more measurement facilities (pressures, temperatures, gas flows, humidity level, single cell voltages, ...) than would be really required for a commercial fuel cell vehicle.

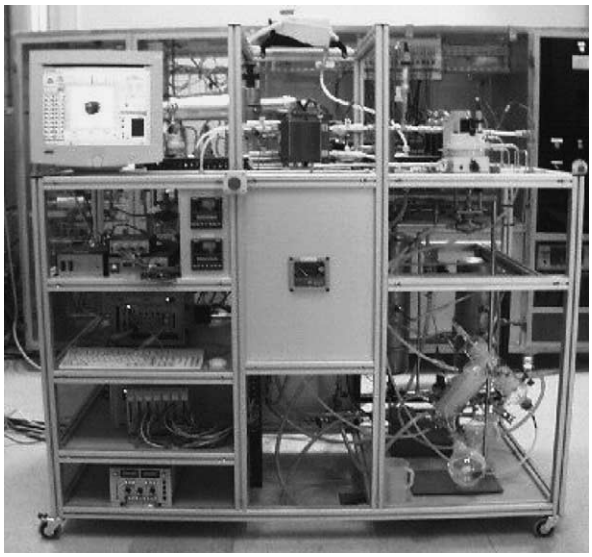

Fig. 4. Structure of the test bench.

For the training process of the proposed NN model, a set of different voltage–current $U(I)$ characteristics of the fuel cell system are used.

3.2.1. Data processing

To carry out good training, it is necessary that the training examples are well distributed throughout the operation range. The training patterns must also be appropriately selected. In our system, the values do not vary on the same scale. For example, the pressure variations are measured in millibars and the current variations in ten amperes. Thus, it is advisable to standardize (between −1 and +1 or 0 and 1) the inputs presented to the network in order to facilitate training. Furthermore, processing data permits a better homogeneity of the parameters and avoids neglecting values which may be decisive [10].

3.2.2. Designing the network

As mentioned above, the topology of the neural network (NN) that has been chosen is a feed-forward one (Fig. 3). One must then decide the number of hidden layers and the number of neurons on these hidden layers. This aspect is always the subject of intense search, but as yet no theory enables us to determine the optimal number of neurons and hidden layers for a correct modeling of a process.

After selecting the number of hidden layers, one must choose the network inputs. Some knowledge about the behavior of the system is required here (experts are needed) and especially about the factors that condition the system output.

3.2.3. Inputs and output choice

The model uses an NN which has four input nodes and a linear output neuron. This single output neuron is used here

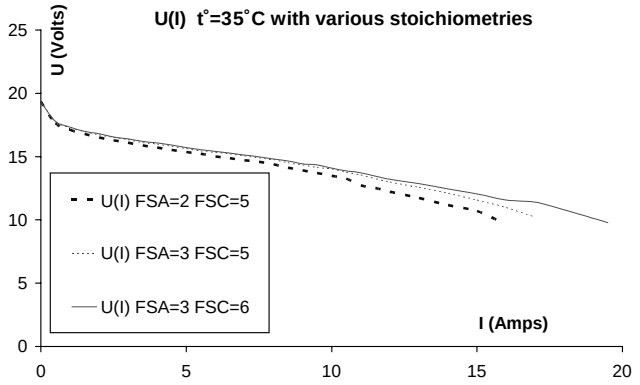

Fig. 5. $U(I)$ measured for different stoichiometries.

for evaluating the fuel cell stack voltage. The aim of our NN model is also to estimate the voltage considering the different operating conditions and the delivered current.

On the basis of the various tests carried out on the test bench with dry gases, it has been established that the influence of the input gas stoichiometry and the stack temperature are decisive for the stack behavior. Fig. 5 presents the experimental evolution of the fuel cell static characteristic at 35 °C for different stoichiometric conditions (FSA is the stoichiometry factor on the anode side (hydrogen), FSC is the stoichiometry factor on the cathode side (air)). Fig. 6 presents the evolution of the $U(I)$ static characteristic under fixed stoichiometry conditions and for different stack temperatures. Considering these experimental results, the two gas flows values and the stack temperature were chosen in addition to the current as inputs for the proposed NN model.

Indeed, considering Figs. 5 and 6, the influence of the stoichiometry and the temperature on the stack is very clear. It can be seen that the fuel cell power increases with the temperature and the input gas flows. Nevertheless, the best fuel cell static characteristic is obtained for a 60 °C stack temperature and for a stoichiometry factor of 3 on the anode side and 6 on the cathode side. A higher operating temperature or different gas flow values do not improve the system behavior; by using dry gases, no additional water is supplied to the fuel cell, so the membrane dries out.

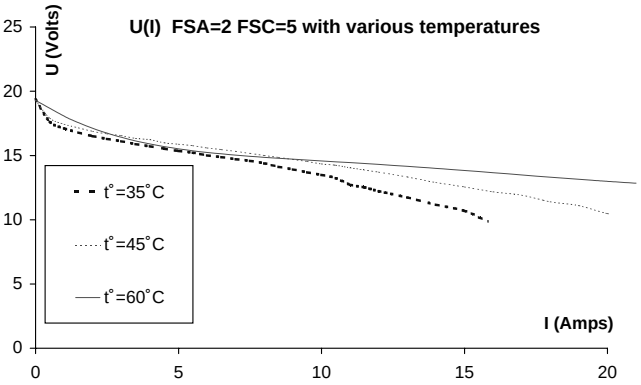

Fig. 6. $U(I)$ measured for different temperatures.

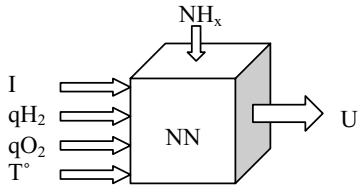

Fig. 7. NN parameters. I: stack current, U: stack voltage, T^o: stack temperature qH_2: hydrogen flow, qO_2: oxygen flow NH_x: number of neurons on hidden layer x.

Fig. 7 presents the general scheme of the proposed PEM fuel cell system neural network model. It must be underlined that the number of inputs in this model has been progressively increased in order to quantify the influence of each considered parameter. Thus, the most significant parameters were maintained in order to obtain an excellent model requiring as low a number of measurements as possible. Indeed, the main objective here is to propose a model able to represent the system behavior as perfectly as possible with the smallest number of sensors on the vehicle. The anodic and cathodic pressure values are not, in our case, significant parameters as the stack must be operated under atmospheric pressure. Furthermore, these pressures are held constant in case of flow and temperature variations.

3.2.4. Training with back propagation

The hidden layer(s) have no target values. Therefore, a procedure is used to back propagate the output layer errors to the hidden layer neurons in order to modify their weights to minimize the error.

Training is started with a gradient algorithm. When we approach a minimum, this gradient takes increasingly low values and convergence towards this minimum is greatly retarded. Then, to speed up the convergence, we use the quasi-Newton algorithm [10]. Before starting training, weights are initialized to small random values to reduce the chance of prematurely saturating the logistical neurons and thus reducing the training speed.

The common principle of second-order algorithms is to compute a descent direction obtained by a linear transformation of the cost function gradient. For the gradient algorithm, the weights are updated at each step according to:

$$W(t+1) = W(t) + \Delta W \tag{4}$$

where

$$\Delta W = -\alpha \frac{\partial J}{\partial W(t)}$$

In this paper, we also use the quasi-Newton algorithm with an approximation of the Hessian H_W of the cost function [11]. In this case, the weight update equation is given by (5) (α is the learning rate).

$$W(t+1) = W(t) - \alpha [H_W(t)]^{-1} \frac{\partial J}{\partial W(t)} \tag{5}$$

where

$$[H_W(t)] = \frac{\partial^2 J}{\partial W(t)^2}, \qquad W(t+1) = W(t) - \alpha \frac{\partial J}{\partial W(t)} \frac{\partial W(t)^2}{\partial^2 J}$$

α can be thought as the size of the step down the error gradient direction. If very small steps are considered, an error minimum is obviously found, but this minimum research procedure may take a very long time. Larger steps may result in unstable learning since this could lead to pass the researched minimum. To speed up training and still ensure stability, the following heuristic method is used to determine the step size.

The heuristic rule states are:

- If training "went well" (error decreased) then increase the step size.

  $\alpha = \alpha 1.1$

- If training is "poor" (error increased) then decrease the step size.

  $\alpha = \alpha 0.5$

- Only update the weights if the error decreased.

The training is stopped after a fixed number of iterations or after the mean square error on the training set (TMSE) goes below a specified tolerance.

3.2.5. Overfitting detection strategy

The problem of overfitting is frequently mentioned in the NN literature. It is caused by the capacity of an NN, with a number of memorizing units higher than necessary (one also speaks of "over-parameterization"), to learn the examples of the training sequence perfectly. In many cases, this also leads to very poor generalization ability. In order to stop the training before the overfitting phenomenon occurs, the evolution of the TMSE and the mean square error on a performance estimation set (PMSE) are plotted on the same graph (Fig. 8).

One way to avoid overtraining is to use cross-validation. The sample set is spilt into a training set (estimation of the TMSE) and a validation set (estimation of the PMSE) [8]. The NN parameters are estimated on the training set, and the performance of the model is tested on the validation set.

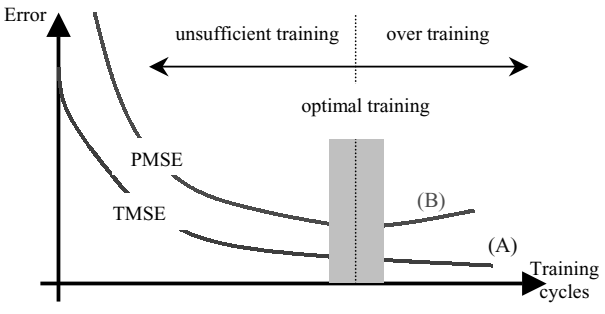

Fig. 8. Evolution of the TMSE and the PMSE.

Thus, training is stopped when the PMSE (curve B) reaches a minimum and the last set of parameters that has been computed is used to produce the forecast. The TMSE and PMSE are obtained with the aid of Eq. (3).

3.3. Training procedure and results

In order to have both a training set and a validation set relatively significant to validate the proposed NN model, several tests are carried out using the 500 W fuel cell test bench. These tests are done considering a quasi-static operating mode. In that case, the only variable parameters are the stoichiometry factors (input gas flows), the stack temperature (controlled through an external cooling water circuit) and, of course, the fuel cell current and voltage. Each test is carried out with a fixed stack temperature and also fixed stoichiometry factors on both the anode side (FSA) and cathode side (FSC).

3.3.1. Training set

The training set is carried out on the experimental curves obtained at a 35 °C stack operating temperature and for various stoichiometries. The following stoichiometry conditions were considered for the characteristic $U(I)$ (Fig. 9): FSA = 2, FSC = 4; FSA = 2, FSC = 5; FSA = 3, FSC = 5; and FSA = 3, FSC = 6.

3.3.2. Testing set

The testing set covers all the effective data of two other experimental $U(I)$ characteristics: FSA = 2, FSC = 6 and FSA = 3, FSC = 4. To obtain efficient training of the network, it is computed with a fixed number of iterations.

3.3.3. Training of the network

In practice, the TMSE and the PMSE values are computed at each iteration. Of course, as already said, training could be stopped as soon as a minimum of PMSE is reached but it is not sure that this value is effectively the global minimum of the PMSE. Therefore, training is carried out until the fixed number iteration is really reached. The adopted strategy consists in memorizing the network parameters when a minimum of PMSE is reached. The neural weight values used to calculate the network output are those obtained with the global minimum of PMSE.

Figs. 10 and 11 illustrate the interest to continue the training even if a minimum of PMSE is met. Indeed, whereas the TMSE value clearly decreases monotonously, the PMSE value goes through two successive local minima. To be sure of obtaining the global minimum, it would be necessary to continue the training with more iteration. But on the basis of the different tests carried out with the model, we concluded that longer training will not give better results. Moreover, the calculated time required would be too long. By means of this strategy, we obtain very interesting results (Fig. 12). It has to be underlined that these results are only obtained on the basis of the forecasting ability of the proposed neural

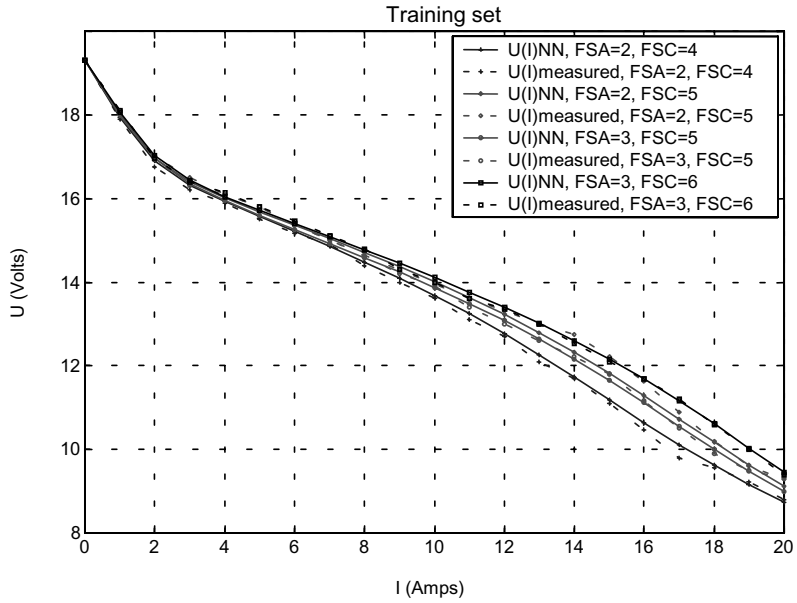

Fig. 9. $U(I)$ measured and $U(I)$ obtained with NN for different stoichiometries: training set.

network model. In fact, considering the two curves presented in Fig. 12, only the open circuit voltage and the maximal load voltage were introduced in the training set. All data on these characteristics, also excluding no load and maximal load voltage, are predicted by the proposed network.

The PMSE values of the two calculated curves are relatively weak (Table 1), which enables us to conclude the quality of the training and the capacity for generalization. If the number of neurons increases on hidden layers, the PMSE value and the differences between experimental and calculated curves increase. In the same way, the fact of carrying out training on a more significant number of iterations does not improve the generalization ability of the model.

Table 1 lists the mean square errors and the maximum difference voltages between the calculated points and the experimental points that result from Fig. 12. As can be seen in Fig. 12 and Table 1, the proposed neural network model of the fuel cell generator can correctly interpolate between patterns not used in training. The difference between the estimated voltage and the real voltage is very low (<1.5%).

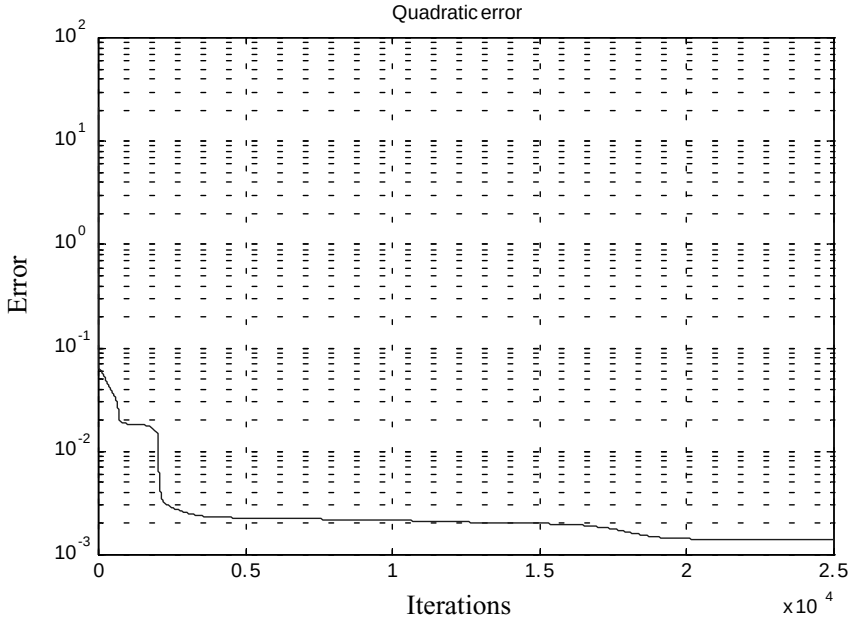

Fig. 10. Training set mean square error.

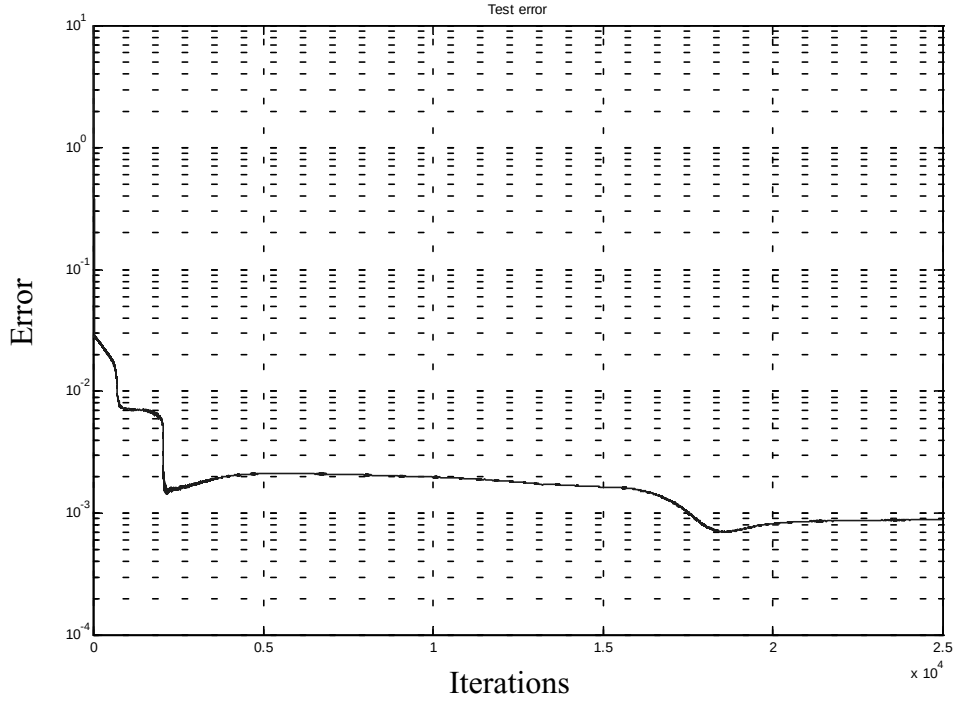

Fig. 11. Validation set mean square error.

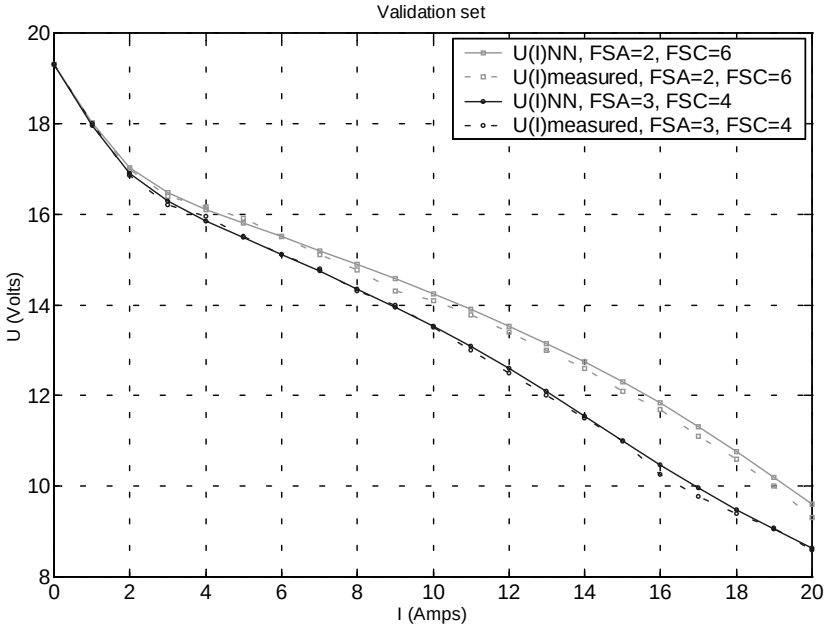

Fig. 12. Predicted and experimental $U(I)$ characteristics for different stoichiometry conditions.

Table 1
Error statistics

$U(I)$	TMSE	PMSE	FC static curve maximum difference
SFA = 2, SFC = 6	1×10^{-3}	0.11	237 mV, 1.23%
SFA = 3, SFC = 4	1×10^{-3}	0.28	254 mV, 1.32%

Therefore, only a very small number of training patterns are necessary to perform such a powerful neural network model of a fuel cell system. In this paper, only few results were presented. It has to be underlined that the capacity of generalization is the same as mentioned above for different operating conditions (lower or higher temperatures, different gas flows, i.e. different stoichiometric factors . . .).

4. Conclusion

Modeling fuel cell generators dedicated to automotive applications are a very important milestone in the integration of these power sources into vehicles. Conventional analytical modeling strategies have difficulties in fulfilling this purpose. Indeed, physicochemical phenomena take place in fuel cell systems and the parameters involved in these phenomena are very numerous and difficult to evaluate with satisfactory precision. Other modeling strategies, presenting efficiency but also the tuning ability with as few experimental tests on the real system as possible, should also be considered. As has been presented in this paper, neural networks provide an interesting and powerful solution.

This work enables an efficient static model of the fuel cell system to be obtained. Due to the training strategy and the topology of the NN adopted, we are able to make predictions in static mode. Moreover, only four experimental learning patterns are necessary for the training stage of the network. This results in a very powerful fuel cell modeling methodology considering automotive applications. The proposed model can be implemented without any problem in a complete vehicle powertrain simulation as experimental and simulated results are very close.

Finally, if the training has been done off-line, an on-line training sequence of the model can also be considered. In fact, on-line integration can be easily carried out due to the fact that the calculating time required for the model is very short. Once training has been carried out, the calculating time of the model on a Pentium 866 MHz is less than 1 s. This time is therefore completely compatible with the requirements of an automotive powertrain simulation.

Acknowledgements

This work is supported by PSA PEUGEOT CITROEN.

References

[1] K. Kordesch, G. Simader, Fuel Cells and Their Applications, VCH, Weinheim, 1996.
[2] A. Amphlett, R.F. Mann, B.A. Peppley, P.R. Roberge, A. Rodrigues, A model predicting transient responses of proton exchange membrane fuel cells, J. Power Sources 61 (1996) 183–188.
[3] M. Wöhr, K. Bolwin, W. Schnurnberger, M. Fischer, W. Neubrand, G. Eigenberger, Dynamic modelling and simulation of a polymer membrane fuel cell including mass transport limitation, Int. J. Hydrogen Energy 23 (1998) 213–218.
[4] X. François, M.C. Péra, D. Hissel, J.M. Kauffmann, Design of a test bench for low power PEMFC, in: Proceedings of the European Polymer Electrolyte Fuel Cell Forum, Lucerne, Switzerland, 2001, pp. 491–499.
[5] M.C. Péra, D. Hissel, J.M. Kauffmann, Fuel cell systems for electrical vehicles: an overview, IEEE Vehicular Technol. Soc. News 49 (1) (2002) 9–14.
[6] H.S. Hippert, C.E. Pedreira, R.C. Souza, Neural networks for short-term load forecasting: a review and evaluation, IEEE Trans. Power Syst. 16 (1) (2001) 44–55.
[7] C.M. Bishop, Neural Networks for Pattern Recognition, Claredon Press, Oxford, 1997.
[8] S. Haykin, Neural Networks, A Comprehensive Foundation, second ed., Prentice Hall, Upper Saddle River, NJ, 1999.
[9] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: the state of the art, Int. J. Forecast. 14 (1998) 35–62.
[10] L. Constant, Modélisation de dispositifs électriques par réseaux de neurones en vue de l'émulation en temps réel, Ph.D. notes, Institut National Polytechnique de Toulouse, 2000.
[11] G. Cybenko, Continuous value neural networks with two hidden layers are sufficient, Math Control Signals Syst. 2 (1989) 303–314.