© Strojniški vestnik 47(2001)8,411-416
ISSN 0039-2480
UDK 536.2
Izvirni znanstveni članek (1.01)

© Journal of Mechanical Engineering 47(2001)8,411-416
ISSN 0039-2480
UDC 536.2
Original scientific paper (1.01)

ONLINE IDENTIFICATION OF HEAT DISSIPATERS USING ARTIFICIAL NEURAL NETWORKS

Sylvain LALOT[1] and Stéphane LECOEUCHE[2]
M.E.T.I.E.R., Ecole d'Ingénieurs du Pas-de-Calais,
Campus de la Malassise, B.P. 39,
62967 Longuenesse Cedex
France
Phone: +33 321 388 510
Fax: +33 321 388 505
Emails: (1) slalot@eipc.fr (2) slecoeuche@eipc.fr

ABSTRACT

This paper focuses on the feasibility of online identification of thermal systems. The transfer function is not looked for, but a black box model is obtained. In the first part, the principles of online identification are reminded. This leads to the definition of the regression vector and of the regressors. Then these principles are applied to neural based techniques which are adapted from standard ARX (AutoRegressive structure with eXtra inputs) and OE (Output-Error) models. For the Neural Network ARX (NNARX) model, only one example is given, which leads to a not fully satisfactory identification. This identification is based on the response of the system to random heat rates during random times. The validation is based on the response to another set of random heat rates and on the response of the system to a step function. For Neural Network OE (NNOE) model, the influence of the number of regressors is presented along with the influence of the number of neurons on the hidden layer. It is shown that many architectures lead to a good identification, but that some particular models may lead to a very poor result. To make the comparison possible between the proposed models, a distance criterion is computed. This leads to the choice of the best adapted architecture.

INTRODUCTION

Heat transfer from electronic devices such as transistors may be enhanced by inserting them in an aluminum sleeve or by mounting them on a heat dissipater. The temperature during steady states is easy to determine, as soon as the thermal resistance of the dissipater and the convection coefficient are known. It is less easy to determine the temperature during transient states if the Biot number is higher than 0.1 (Incropera and de Witt, 1996). This happens when the cooling of the dissipater is obtained using a fan. As it is not always possible to use standard offline identification techniques (e.g. response to a step function), other methods must be applied. These methods will lead to an online identification. Generally speaking, this will allow detection of drifts in the behavior of a thermal system (e.g. loss of efficiency of a fan, evolution of the fouling factor in a heat exchanger, modification of the thermal resistance between two solids, surveying of the aging of electrical heating elements). This would represent an improvement in the supervision possibilities.

As neural based techniques are more and more popular and has proved to be efficient in identification of thermal systems (Lalot, 2000), (Lalot and Lecoeuche, 2000), (Kalogirou, 1999), (Kalogirou, 2000), two of them, adapted from the ARX model and the Output Error Model (Ljung, 1999), are tested.

To show the feasibility of online identification, a heat dissipater is used as thermal system. Figure 1 shows a schematic of the system.

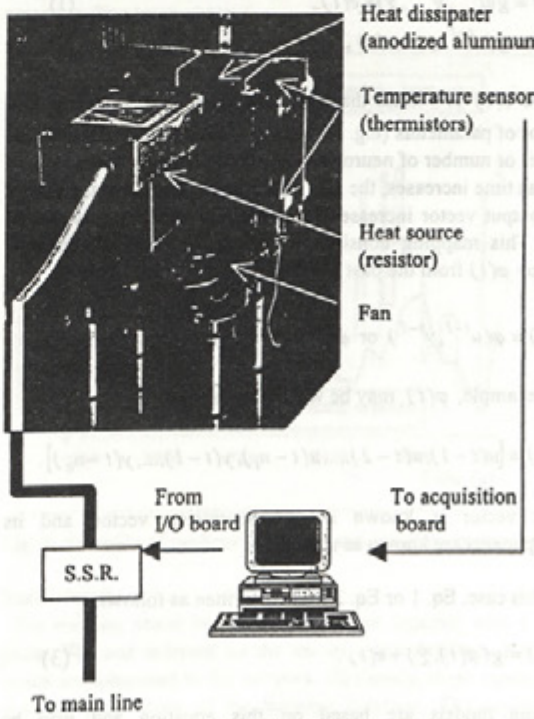

Fig. 1: Schematic of the studied thermal system

As the thermal system is quite slow, the command of the solid-state relay coming from the I/O board does not need to be

fast. A period of 4 seconds has been chosen. So, it is possible to make the heat rate vary from 0 to 100% by 0.5% steps (50 cycles per second in France).

PRINCIPLES OF ONLINE IDENTIFICATION

Usually, the identification of a system is carried out offline. Numerous methods may be applied (e.g. Strejc, Ziegler-Nichols, ...). But to be offline means that it is necessary to take the studied system away from its environment. This is avoided by online identification techniques that take the actual service parameters as inputs. In order to simulate service duties, random inputs have been chosen here.

The online identification may be applied to white box models (i.e. models are perfectly known), to gray box models (i.e. models are partly known through physics), and to black box models (i.e. no physical properties of the models can help the identification); it has been chosen here to consider a particular thermal system as a black box.

The identification problem may be described in different ways. In any case, for given inputs $u^t = [u(1),u(2),...,u(t)]$, outputs $y^t = [y(1).y(2).....y(t)]$ are measured. Then two possibilities occur. Either a relationship between past observations $[u^{t-1}.y^{t-1}]$ and future outputs $y(t)$ is looked for (Eq. 1) or only previous inputs are taken into account (Eq. 2).

$$y(t) = g\left(u^{t-1}.y^{t-1}.\chi\right) + e(t), \quad (1)$$

$$y(t) = g(u^{t-1}.\chi) + e(t), \quad (2)$$

where $e(t)$ is a noise that may disturb the system and χ is a vector of parameters (e.g. weights of the connections of a neural model or number of neurons on the hidden layer). It can be seen that as time increases, the size of the input vector and the size of the output vector increase. So, generally, a mapping is carried out. This mapping consists in defining a finite dimensional vector $\varphi(t)$ from the past observations:

$$\varphi(t) = \varphi(u^{t-1}.y^{t-1}) \text{ or } \varphi(t) = \varphi(u^{t-1}).$$

For example, $\varphi(t)$ may be written as follows:

$$\varphi(t) = [u(t-1),u(t-2),...,u(t-n_i),y(t-1),....,y(t-n_o)].$$

This vector is known as the regression vector, and its components are known as regressors.

In this case, Eq. 1 or Eq. 2 may be written as follows:

$$y(t) = g(\varphi(t).\chi) + e(t). \quad (3)$$

Some models are based on this equation and may be represented (Ljung, 1999) by a generalized equation:

$$A(z)y(t) = \frac{B(z)}{F(z)}u(t) + \frac{C(z)}{D(z)}e(t), \quad (4)$$

where $X(z)f(t) = x_0 f(t) + x_1 f(t-1) + x_2 f(t-2) + ...$

The Box-Jenkins model (BJ model) is characterized by $A = 1$; the ARMAX model is characterized by $F = 1$ and $D = 1$; the Output-Error model (OE model) is characterized by $A = 1$, $C = 1$, and $D = 1$; the ARX model is characterized by $F = 1$ and $D = 1$.

These models may be implemented using neural techniques. During the last five years, thanks to research works (Sjöberg, 1996) (Norgaard, 1997) these online identification neural techniques have been improved by using neural networks stemming from classical backpropagation neural networks used for modelling the state-space of the system.

The first step of the identification process is to choose the model. The second step consists in the determination of the size of the regression vector. Then the learning phase can begin. A test phase has to be carried out to verify the agreement between the model and the actual system. It may be necessary to come back to the choice of the model if the tests are not satisfactory. Finally, a exploitation phase comes during which failures or drifts may be detected, new service conditions may be investigated, ... Figure 2 summarizes the identification process:

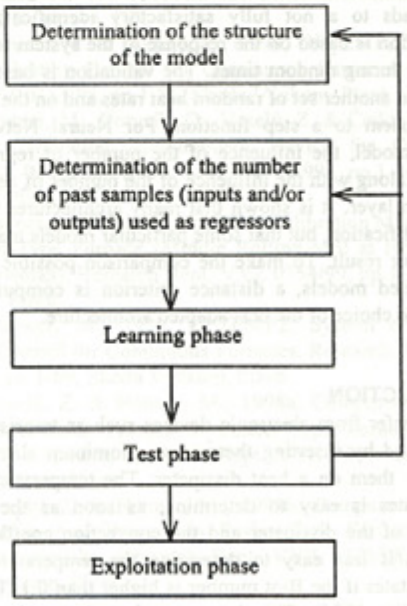

Fig. 2: Schematic of the identification process

Each step of the identification process is detailed below for the identification of heat dissipaters.

APPLICATION TO HEAT DISSIPATERS

Structure of the model and number of regressors

The guideline is here to test most simple models first. So, it is recommended (Sjöberg, 1996) to first test the ARX model, and if the results of the test phase are not satisfactory, to test the OE model. If none leads to good results, it may be generally necessary to move to non-linear models. So, two types of architecture are tested using the Matlab Toolbox "Neural Network Based System Identification" (Norgaard, 1997). Figure 3 shows the representation of the general architecture.

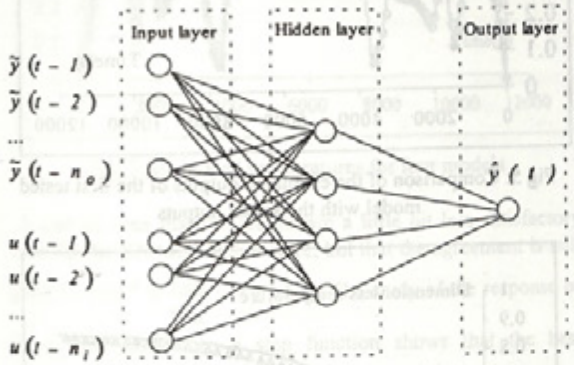

Fig. 3: General architecture of the tested networks

The input vector consists in actual input and, for the NNARX architecture in actual outputs $(\tilde{y}(t) = y(t))$, and for the NNOE architecture, in estimated outputs $(\tilde{y}(t) = y(t))$. The output is always the estimated value. The n_h neurons of the hidden layer, as well as the output neuron, are characterized by a transfer function; a linear transfer function and a hyperbolic tangent transfer function have been tested. The number of neurons on the hidden layer, and the transfer function types influence the quality of the identification and the computing time. Only experiments lead to the good choice.

There is no rule to determine the efficient number of regressors. If only input past samples are used as regressors, it can be considered that the regressors should cover the duration of a complete stabilization of the system. So, if the time response of the system (the time that is necessary to get close to the asymptote) is t_{stab} and if the sampling period that is used to measure the output is t_{samp}, the number of regressors should be at least $n_r = \frac{t_{stab}}{t_{samp}}$. But this could lead to a too large number of regressors. To reduce the number or regressors, it is possible to introduce output past samples as regressors. But this can lead to instabilities during the exploitation phase. Nevertheless, this combination should be tested first.

Learning phase

For any neural technique, a learning phase is necessary. It is based on the use of a learning database. This database consists of known inputs coupled with known outputs. The input vector is the regression vector $\varphi(t)$, and the corresponding output vector is the known output $y(t)$. The aim of the learning phase is to adapt the architecture of the network. This is done by the modification of the connection weights. The calculation of this modification is based on the differential between the known outputs and the estimated outputs. It is here necessary to use an adequate learning rule. This phase leads to the determination of the number of necessary learning steps and to the determination of the adequate number of inputs (i.e. the size of the learning database) (Kosko, 1992).

An example of a database consists of random heat rates q that have been generated during random times in a heat dissipater and of the corresponding temperature T_{dis} of one point of the heat dissipater. In order to get significant temperature variations, the following ranges have been chosen:

10 W < heat rate < 90 W,
120 s < stabilization time < 600 s.

To present the results in a dimensionless way, a maximum heat rate is determined q_{max}=100W , that corresponds to the maximum temperature $T_{dis\,max}$. This leads to the definition of a dimensionless temperature:

$$\theta = \frac{T_{dis} - T_a}{T_{dis\,max} - T_a}.$$

The heat rate will also be given in a dimensionless way:

$$\bar{q} = \frac{q}{q_{max}}$$

Figure 4 shows an example of heat rates and of the corresponding temperatures.

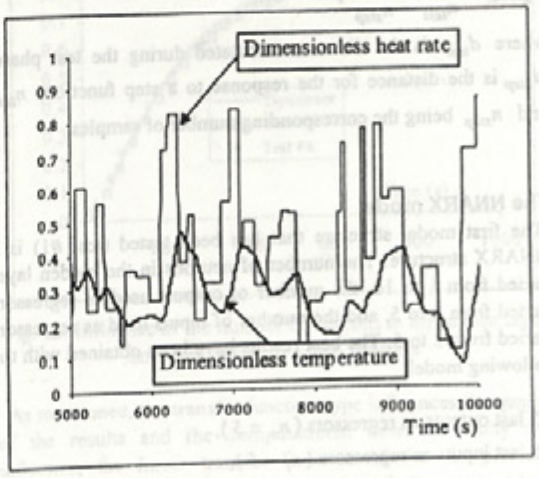

Fig. 4: Example of random heat rates and resulting temperatures

Test phase

The learning phase has to be carried out together with a test phase. The test is based on the use of a test database. Known inputs are submitted to the network. Obviously, these inputs are different from those of the learning database. The estimated outputs are compared to the known outputs. If there is a good agreement the architecture may be declared as well adapted. On the contrary, if there is a large differential, the learning phase has to be pursued. If there is no longer a evolution of the

weights of the connections, either the architecture of the neural network has to be modified or the model itself has to be changed.

Exploitation phase

As it has already been mentioned, the exploitation phase is the final goal of the identification process. It is during this phase that long term evolution such as fouling can be detected.

EXPERIMENTS AND RESULTS

As it has already been mentioned, two models have been tested. For both cases, numerous configurations have been tested. The number of regressors has been modified, the number of neurons in the hidden layer has been changed, ... The validation of the architecture is based on the comparison of the model outputs with the test database inputs (referred as test phase), and on the comparison of the response of the model to a step function with the actual response.

To appreciate the quality of the identification, a criterion has to be established. This criterion is based on the computation of the distance between the estimated response and the actual response:

$$d_{mod,dis} = \sqrt{\sum \left[\theta_{mod}(t) - \theta_{dis}(t)\right]^2} .$$

To take into account short stabilization times (as observed during the test phase), and long stabilization times (as simulated in the response to a step function), a global distance is computed as follows:

$$d_{global} = \frac{d_{test}}{n_{test}} + \frac{d_{step}}{n_{step}},$$

where d_{test} is the distance calculated during the test phase, d_{step} is the distance for the response to a step function, n_{test} and n_{step} being the corresponding number of samples.

The NNARX model

The first model structure that has been tested (test #1) is a NNARX structure. The number of neurons in the hidden layer varied from 1 to 10, the number of outputs used as regressors varied from 2 to 5, and the number of inputs used as regressors varied from 2 to 5. The best results have been obtained with the following model:

- 5 last outputs as regressors ($n_o = 5$),
- 5 last inputs as regressors ($n_i = 5$),
- 6 neurons in the hidden layer having a linear transfer function.

Although the learning is quite perfect (Fig. 5), $\frac{d_{test}}{n_{test}} = 6.7\,10^{-4}$, the response to a step function is not fully satisfactory (Fig. 6), $\frac{d_{step}}{n_{step}} = 1.98\,10^{-3}$.

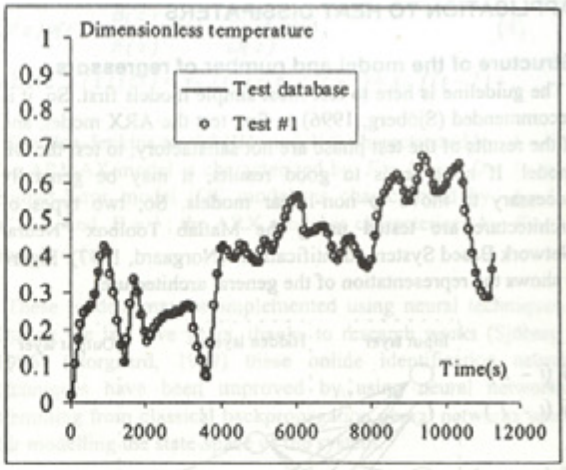

Fig 5: Comparison of the estimated outputs of the first tested model with the actual outputs

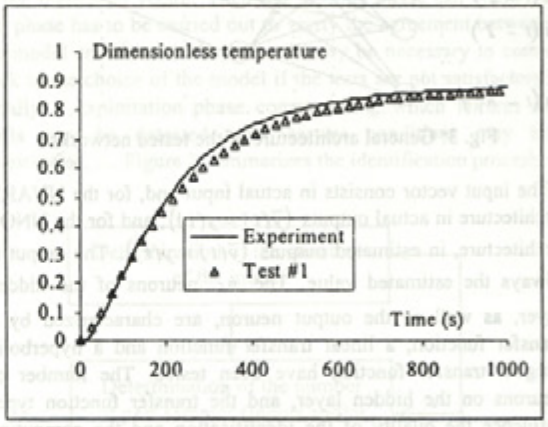

Fig. 6: Comparison of the estimated response to a step function with the actual response

It can be seen that the estimated response is close to the response of a first order system.

The NNOE model

The second studied architecture is a NNOE architecture. The first four models have the following characteristics:

Model n°	Test #2	Test #3	Test #4	Test #5
n_h	6	6	6	6
n_o	2	2	1	3
n_i	2	1	1	2
Transfer function	Linear	Linear	Linear	Linear

Figure 7 shows the comparison of the estimated outputs and of the actual outputs of the test database, for the four architectures.

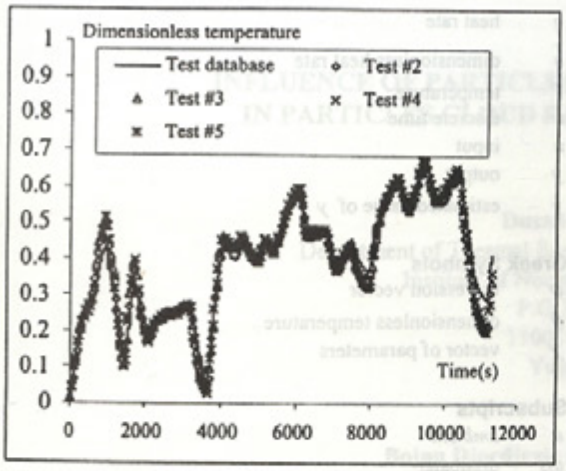

Fig. 7: Estimated temperatures for four models

It can be seen that the results are a little bit less satisfactory than for the NNARX architecture, but that the agreement is still good ($7.1\,10^{-4} \le \frac{d_{test}}{n_{test}} \le 9.3\,10^{-4}$). The study of the response of these four models to a step function shows that the best estimated responses are closer ($0.97\,10^{-3} \le \frac{d_{tstep}}{n_{step}} \le 2.6\,10^{-3}$) to the experimental one than with the NNARX model (Figure 8).

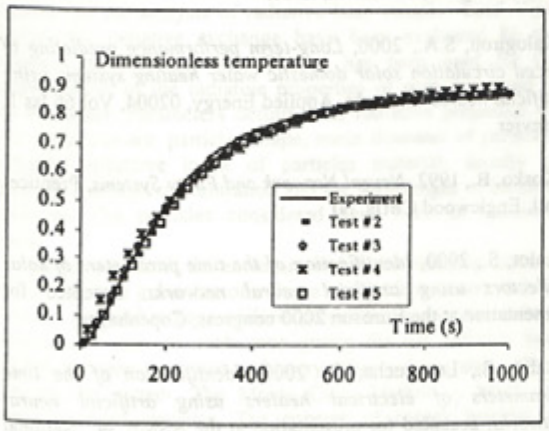

Fig. 8: Response of four models to a step function

A look at the beginning of the curves shows that the response of the model number 4 is close to the response of a first order system. This can be explained by the fact that having only one predicted output used as regressor, the neural network is unable to predict the behavior of systems of order higher than 1.

On the other hand, the response of model number 5 is close to the response of model number 2. This means that it is not necessary, in our case, to increase the number of predicted past outputs. In fact, a system having 12 predicted past outputs used as regressors has been tested. No significant difference has appeared.

As it has been mentioned, the influence of the number of neurons in the hidden layer has to be studied. This is done by comparing the model #5 to a model having only 2 neurons in the hidden layer (all other characteristics being equal). The results presented in Fig. 9 and Fig. 10 show that a good agreement is reached even with a small number of neurons.

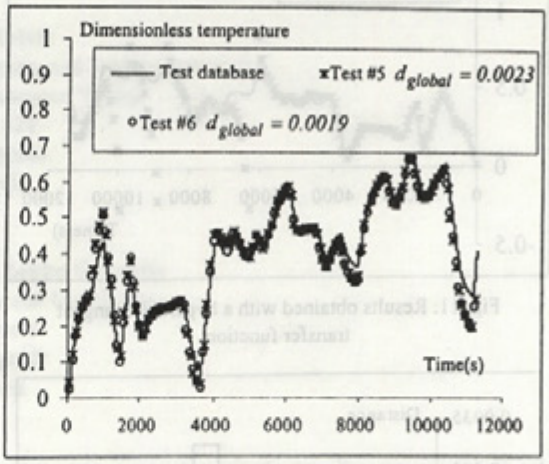

Fig. 9: Influence of the number of neurons in the hidden layer

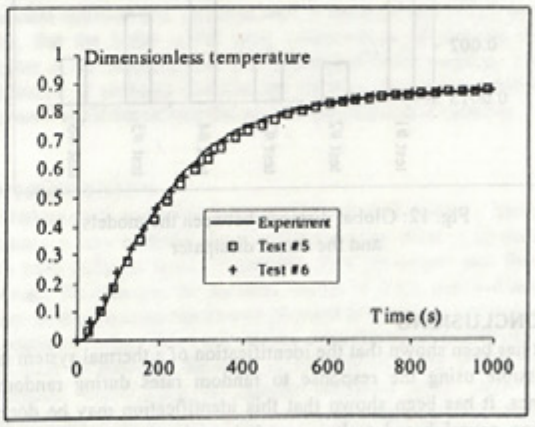

Fig. 10: Influence of the number of neurons in the hidden layer on the response to a step function

As mentioned, the transfer function type influences the quality of the results and the computational time. To study this influence, the linear transfer function of the six previous examples has been changed to a hyperbolic type. Many architectures has been tested, none of them leading to good results. Figure 11 shows the results for an architecture similar to the architecture of model #5 ($n_h = 6, n_o = 3, n_i = 2$).

All results are summarized in Fig.12. This shows that model number 2 may be considered as the best model for the tested dissipater, and that a simple two neurons in the hidden layer architecture (model #6) may lead to good identification. This is interesting when the computational time is critical; reducing the number of neurons decreases the number of weights to be calculated.

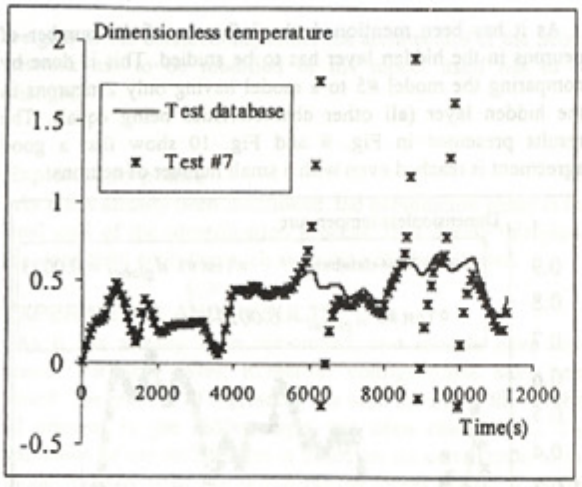

Fig. 11: Results obtained with a hyperbolic tangent transfer function

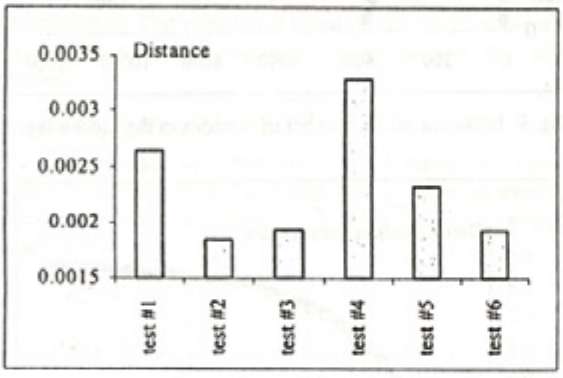

Fig. 12: Global distance between the models and the actual dissipater

CONCLUSIONS

It has been shown that the identification of a thermal system is possible using the response to random rates during random times. It has been shown that this identification may be done using neural based techniques. It has also been shown that, although it is possible to use many architectures, the neural model has to be carefully chosen. This leads to the conclusion that the online identification is efficient.

Future studies will address the survey of the evolution of the connection weights to detect drifts or failures of the thermal system.

NOMENCLATURE

A	polynomial function in equation (4)
B	polynomial function in equation (4)
C	polynomial function in equation (4)
D	polynomial function in equation (4)
d	distance
e	noise
F	polynomial function in equation (4)
n	number of samples or of regressors
q	heat rate
$\bar{q}$	dimensionless heat rate
T	temperature
t	discrete time
u	input
y	output
$\hat{y}$	estimated value of y

Greek Symbols

φ	regression vector
θ	dimensionless temperature
χ	vector of parameters

Subscripts

a	*ambient*
dis	dissipater
h	in the hidden layer
i	in the input layer
max	maximum
mod	model
o	in the output layer
$step$	for the response to a step function
$test$	during the test phase

REFERENCES

Kalogirou, S.A., 1999, *Performance prediction of a solar water heater using artificial neural networks*, ISES 1999 Solar World Congress, Jerusalem, Israel

Kalogirou, S.A., 2000, *Long-term performance predicting of forced circulation solar domestic water heating systems using artificial neural networks*, Applied Energy, 02004, Vol 66 Iss 1, Elsevier

Kosko, B., 1992, *Neural Network and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs, NJ

Lalot, S., 2000, *Identification of the time parameters of solar collectors using artificial neural networks*, accepted for presentation at the Eurosun 2000 congress, Copenhagen

Lalot, S., Lecoeuche, S., 2000, *Identification of the time parameters of electrical heaters using artificial neural networks*, accepted for presentation at the ASME 34th National Heat Transfer Conference, Pittsburgh (PA)

Ljung, L., 1999, *System identification – Theory for the user*, 2nd edition, PTR Prentice Hall, Upper Saddle River (NJ).

Norgaard, M., 1997, *Neural Network Based System Identification*, Technical Report 97-E-851, Technical University of Denmark, Department of Automation

Sjöberg, J., 1996, *Non-linear System identification with Neural Networks*, Ph.D. Thesis, Dept of Electrical Engineering, Linköping University, Sweden