15th Triennial World Congress, Barcelona, Spain

www.elsevier.com/locate/ifac

SYSTEM IDENTIFICATION USING FUNCTIONAL - LINK NEURAL NETWORKS WITH DYNAMIC STRUCTURE

Letitia Mirea[1] and Teodor Marcu[2]

[1] *"Gh. Asachi" Technical University of Iaşi,*
Department of Automatic Control and Industrial Informatics
Blvd. D. Mangeron 53A, RO-6600 Iaşi, Romania
Fax: +40-32-214290, E-mail: lmirea@ac.tuiasi.ro

[2] *"Gerhard Mercator" University of Duisburg,*
Institute of Control Engineering (AKS)
Bismarckstrasse 81 (BB), D-47048 Duisburg, Germany
Fax: +49-203-379 2928, E-mail: t.marcu@uni-duisburg.de

Abstract: The paper considers the development of a new type of artificial neural network and its applicability to non-linear system identification. This is the functional-link neural network with internal dynamic elements. The net consists of a single layer where the non-linearity is firstly introduced by enhancing the input pattern with a functional expansion. The internal dynamic elements are auto-regressive moving average filters that implement local activation feedback and local output feedback, respectively. Experimental results demonstrate a better capability of generalisation of the suggested neural network in comparison with the functional-link net with static structure and external dynamic elements, used so far to perform system identification. *Copyright © 2002 IFAC*

Keywords: non-linear system identification, dynamic neural networks, functional-link net, three-tank system, evaporation process.

1. INTRODUCTION

Artificial Neural Networks (ANNs) have become a powerful tool for many complex applications such as function approximation, optimisation, non-linear system identification, and pattern recognition. This is due to the fact that they are capable to learn from examples and to perform non-linear mappings. Under certain assumptions, it is possible to identify dynamic non-linear systems using ANNs (Patra, *et al.*, 1999). The major problem that arises here is that of coping with dynamics. In this respect, the paper suggests a new dynamic structure based on the Functional-Link Neural Network (FLNN) and its application to system identification.

The FLNN has been developed as an alternative architecture to the multi-layer perceptron network with application to function approximation and pattern recognition. The FLNN is a flat network with a single neuron that has an increased input space given by the functional expansion of its initial inputs (Chen, *et al.*, 1999; Pao, *et al.*, 1994; Patra, *et al.*, 1999). The main advantage of the FLNN is the reduced computational cost in the training stage, while maintaining a good performance of approximation. The present paper suggests the introduction of dynamic elements within the FLNN structure. In this way, the approximation and the generalisation capabilities of the resulted neural network are improved, whilst the training time is still reduced.

The paper is organised as follows. In section 2, the architecture of the static functional-link net is presented. A new dynamic functional-link neural structure is presented in section 3. Three case studies, using the suggested dynamic neural architecture, are described in section 4. These refer to system identification of the following: a simulated system (Patra, *et al.*, 1999), a three-tank laboratory system (Amira, 1993), and an evaporation sub-process from a sugar factory (Bartys and Wasiewicz, 1998). Conclusions about the efficiency of the suggested approach and further research are given in section 5.

2. FLNN WITH STATIC STRUCTURE

The FLNN is a feed-forward single layer neural network with a number of enhancement nodes referred to as functional links. These are used as supplementary inputs within the network (Pao, *et al.*, 1994). Different types of non-linear enhancements have been investigated (Chen, *et al.*, 1999; Pao, *et al.*, 1994; Patra, *et al.*, 1999). A flat network results for which only the connection weights and the bias term must be learned. Thus, the back-propagation learning algorithm (Hagan, *et al.*, 1996), used for adapting the FLNN's parameters, becomes very simple.

The structure of a FLNN is depicted in Fig. 1. The initial N inputs of the net, $u_n, n = 1,..., N$, are functionally expanded to constitute the actual inputs of the neuron, $v_m, m = 1,..., N + M$. In the following, the functional expansion given by a sub-set of orthogonal trigonometric functions is considered. This provides a more compact representation of the function to be approximated, in the mean-square sense, than other orthogonal basis functions (Patra, *et al.*, 1999).

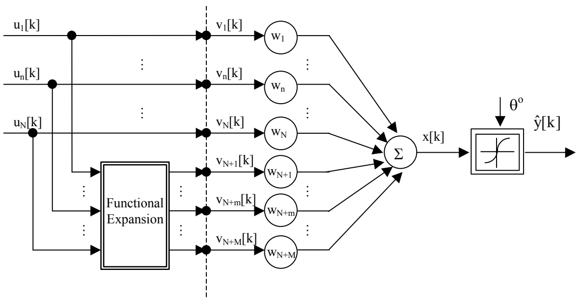


Fig. 1. The structure of a functional-link neural network.

For a pre-specified order of the functional expansion, S, the actual inputs of the neuron are given by the following set:

$$\{u_n, \{\cos(s\pi u_n), \sin(s\pi u_n)\}\};\ s = 1,...S;\ n = 1,..., N .$$

In this way, M=2·S·N supplementary inputs of the neuron are added to the initial N ones. The single neuron of the network is considered to have the activation function of hyperbolic-tangent type:

$$\hat{y}[k] = \frac{e^{z[k]} - e^{-z[k]}}{e^{z[k]} + e^{-z[k]}}, \quad z[k] = x[k] + \theta^o ,$$

where $\hat{y}$ is the output of the neuron, $[k]$ represents the sampling time instant k, θ^o is the bias term, and x is the sum of the original and expanded neuron's inputs using the connection weights w_m:

$$x[k] = \sum_{m=1}^{N+M} w_m \cdot v_m[k] .$$

In order to perform the identification of dynamic systems, the FLNNs have to be provided with dynamic elements and appropriate learning methods (Isermann, *et al.*, 1997). One way of doing this is by considering the FLNN with external dynamic elements. The implementation of the dynamic elements as simple tapped delay units is the most applied.

To model a non-linear process, the most suitable structure for the neural net is the input-output format. For the sake of simplicity, a Single-Input Single-Output (SISO) dynamic system is considered. In this case, the input-output model obtained using a neural network is:

$$\hat{y}[k] = f(u_{\mathrm{P}}[k-d],..., u_{\mathrm{P}}[k-d-k_u], \\ y_{\mathrm{P}}[k-1],..., y_{\mathrm{P}}[k-k_y]) \qquad (1)$$

where u_{P} denotes the process input, y_{P} represents the process output, and $\hat{y}$ denotes the approximated output given by the trained ANN. The maximum time delays k_u and k_y are the dynamic orders of the process, and d denotes the dead-time.

The number of time delay units requires that the system dynamics must be known beforehand. In practice, a trial-and-error tuning of these parameters is applied. The ANNs with internal dynamic elements overcome this drawback. Moreover, the dimension of the input space of the ANN with external dynamics increases, depending on the number of the used past values of the input and the output of the process to be modelled (Ayoubi, 1996).

3. FLNN WITH DYNAMIC STRUCTURE

Another way of providing ANNs with dynamic elements is achieved by including delay elements and recurrent connections within the structure of the net. This kind of networks does not require past values of the process measurements. The approximation of the output of a SISO process is therefore given by:

$$\hat{y}[k] = f(u_{\mathrm{P}}[k], y_{\mathrm{P}}[k-1]) . \qquad (2)$$

Lately, the locally recurrent globally feed-forward neural networks have been the subject of research (Ayoubi, 1996; Isermann, *et al.*, 1997). Their architecture is based on static feed-forward ANNs that have been extended with local recurrence. In the following, the FLNN is considered with a linear filter before the activation unit of the neuron and with a linear transfer function from the output to the input of the neuron. The architecture of the suggested FLNN with dynamic structure (DFLNN) is presented in Fig. 2. This integrates the structures suggested by Marcu, *et al.* (2001), where an Auto-Regressive Moving Average (ARMA) filter has been placed either before the activation function of the neuron or on the back connection from the output to the neuron's input.

The ARMA filter placed before the activation unit of the neuron implements a local activation feedback.

This acts as a memory of the past values of the network inputs. The ARMA filter placed on the back connection from the output of the neuron to its input implements a local output feedback that incorporates the neuron's static non-linearity. This acts as a memory of the past values of the network output. Both filters reduce the level of the noise that affects the network inputs.

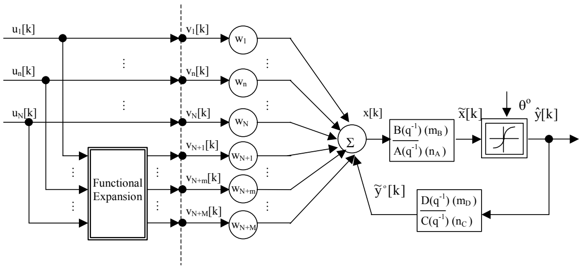


Fig. 2. The architecture of a FLNN with internal dynamic structure (DFLNN).

The output of the network, $\hat{y}$, is obtained using the following equations:

$$\hat{y}[k]=\frac{e^{z^o[k]}-e^{-z^o[k]}}{e^{z^o[k]}+e^{-z^o[k]}},\quad z^o[k]=\tilde{x}[k]+\theta^o,$$

where $\tilde{x}[k]$ is the output of the ARMA filter placed before the activation unit of the neuron:

$$\tilde{x}[k]=\sum_{j=0}^{m_B} b_j\cdot x[k-j]-\sum_{i=1}^{n_A} a_i\cdot\tilde{x}[k-i].$$

Above, n_A and m_B represent the filter's denominator and numerator orders, respectively, whilst $a_i, i=1,\dots,n_A$ and $b_j, j=0,\dots,m_B$ are the coefficients of the denominator and numerator, respectively. $x[k]$ represents the weighted sum of the neuron's inputs augmented with the output of the ARMA filter placed on the reaction from the output to the input of the neuron:

$$x[k]=\sum_{m=1}^{N+M} w_m\cdot v_m[k]+\tilde{y}^o[k],$$

where $\tilde{y}^o[k]$ is the output of the ARMA filter that implements the local output feedback:

$$\tilde{y}^o[k]=\sum_{j=1}^{m_D} d_j\cdot\hat{y}[k-j]-\sum_{i=1}^{n_C} c_i\cdot\tilde{y}^o[k-i].$$

In the previous equation, n_C and m_D represent the filter's denominator and numerator orders, respectively, whilst $c_i, i=1,\dots,n_C$ and $d_j, j=1,\dots,m_D$ are the coefficients of the denominator and numerator, respectively.

The parameters characterising the architecture of the network (S, n_A, m_B, n_C, m_D) are determined by a trial-and-error process in a pre-defined search space determined by the application to be developed. For a given architecture, the network's parameters are the connection weights, the bias term, and the filters' coefficients. These parameters, collected in a vector $\boldsymbol{\theta}$, are determined with an extended dynamic back-propagation algorithm. The objective is to ascertain an optimal parameter set $\boldsymbol{\theta}^*$ of the DFLNN that minimises a quadratic performance index J. The latter is defined based on the output prediction errors between the desired model output, $y_d[k]$, and the actual DFLNN output, $\hat{y}[k]$:

$$\boldsymbol{\theta}^*=\arg\min_{\boldsymbol{\theta}}\{J(\boldsymbol{\theta})\},\ J(\boldsymbol{\theta})=\frac{1}{2}\sum_{k=1}^{K}(y_d[k]-\hat{y}[k,\boldsymbol{\theta}])^2,$$

where K denotes the number of training data. Moreover, the batch learning mode is applied to, i.e. the net parameters are adapted after an entire pass of the training set through the network (one epoch). The mechanisms of variable parameter of learning rate and momentum term are also used (Hagan, *et al.*, 1996; Marcu, *et al.*, 2001).

4. APPLICATIONS

4.1 Simulation study

The difference equation of a simulated non-linear process (Patra, *et al.*, 1999) is given by:

$$\begin{aligned}y_P[k]=f_P(&u_P[k-1],u_P[k-2],\\&y_P[k-1],y_P[k-2],y_P[k-3])\end{aligned},\qquad(3)$$

where the non-linear function f_P is :

$$\begin{aligned}&f_P(g_1,g_2,g_3,g_4,g_5)=\\&[g_1g_2g_3g_5(g_3-1)+g_4]/(1+g_2^2+g_3^2)\end{aligned}.$$

In the performed experiments, the following input of the plant was considered (Patra, *et al.*, 1999):

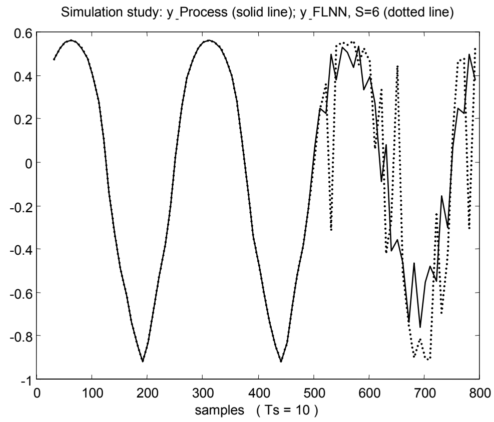
$$u_P[k]=\sin\frac{2\pi k}{250},\ k=1,\dots,500.$$

The output of the plant was approximated by both presented FLNN structures. The training data of the networks were selected using a sampling period T_S=10 (k = 1, 11, ...). To test the trained neural networks, the following input of the plant was considered (Patra, *et al.*, 1999):

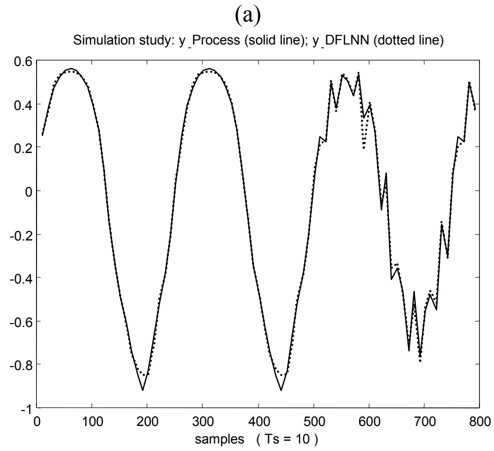
$$u_P[k]=\sin\frac{2\pi k}{250}+0.2\sin\frac{2\pi k}{25},\ k=501,\dots,800.$$

The training of the neural networks was performed in the batch learning mode for a number of 5000 epochs. The order of the functional expansion was varied from S = 1 to S = 15. The FLNN with external dynamic elements identified the series-parallel model (1) corresponding to the equation (3). The FLNN with internal dynamic elements identified the series-parallel model (2). In the latter case, the search space for the orders of the ARMA filters was considered in the set {0, 1, 2, 3} with $n_A + m_B > 0$ and $m_D + n_C \geq 1$ for each fixed value of the functional expansion order *S*.

The best results of system identification, using the FLNN with external dynamic elements, are presented in Fig. 3a. They correspond to the case when the order of the functional expansion is S = 6. One remarks the very good quality of approximation of the training data (*k*=1,…,500) and the poor results when untrained data are processed by the net (*k*=501,...,800).



(a)



(b)

Fig. 3. Results of system identification with the FLNN: with external dynamic elements (a), and with internal dynamic elements – DFLNN (b).

Fig. 3b shows the best results of system identification using the FLNN with internal dynamic elements. The results correspond to a net with the architecture characterised by S=2, m_B=3, n_A=3, m_D=1, n_C=1. The performance of approximating the training data is slightly better in the case of FLNN with external dynamic elements (*k*=1,…,500). However, the quality of approximation is better when untrained data are processed by a FLNN with internal dynamic structure (*k*=501,...,800).

The influence of the measurement noise was considered as well. Gaussian noise with zero mean, unitary variance and amplitude of a certain value that determines a specified Noise-to-Signal Ratio (NSR) was added to the process output. The considered NSR parameter was varied from 5% to 50%, being constant for a certain training experiment. The DFLNN was trained with noisy data. Knowing the added noise, the goal was to find out the network with the minimal architecture such that the sum of squared errors between the output of the net and the noise-free output of the process to be as small as possible. The results presented in Fig. 4 correspond to a net with the architecture characterised by S=2, m_B=1, n_A=1, m_D=1, n_C=1. The noise is filtered efficiently, while the true (noise-free) output of the system is acceptably reproduced.

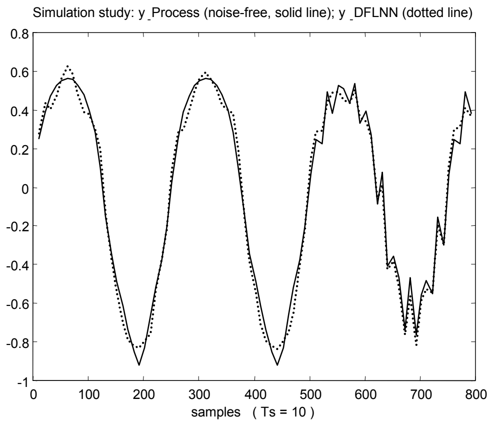


Fig. 4. Results of system identification with the FLNN with internal dynamic elements – DFLNN, trained in noisy environment (NSR=50%) and evaluated with noisy process data (NSR=100%).

4.2 Identification of a three-tank system

The experimental set-up "Three-Tank System" (Amira, 1993) consists of three cylindrical tanks with identical cross sections being filled with water. Circular pipes interconnect the tanks (Fig. 5).

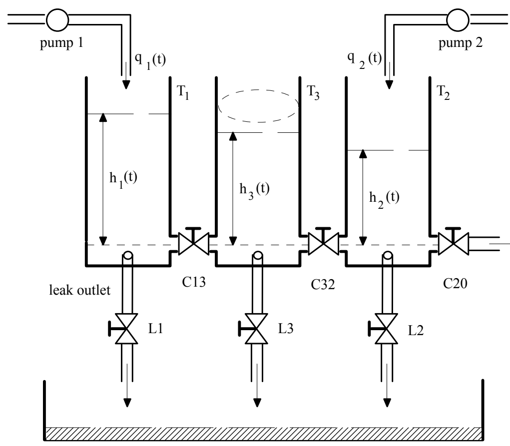


Fig. 5. The Three-Tank System DTS200.

An analytical model of the system is represented by three first-order non-linear differential equations. That model is used by an appropriate strategy to control the water inlet by two pumps. The volume flows $q_1(t)$ and $q_2(t)$ of lateral tanks T_1 and T_2, respectively, are controlled such that the level in the corresponding tanks, $h_1(t)$ and $h_2(t)$, are pre-assigned independently. The level $h_3(t)$, in the middle tank T_3, is uncontrollable. Here t stands for the time variable. The control strategy works at a sampling rate of 0.1 seconds. Although the dynamic modelling of the considered system is relatively simple, the resulted non-linear analytical model is a limited approximation (Marcu, *et al.*, 1999).

For the experiments, the reference values of the liquid levels in the lateral tanks were changed pulse-wise with different magnitude and duration for each controlled tank. A test period of 400 seconds was considered. The system data were sampled at every 5 seconds, due to the slow nature of the process. Thirty-five experiments were done in a period of a month in order to take into consideration the influence of the plant environment.

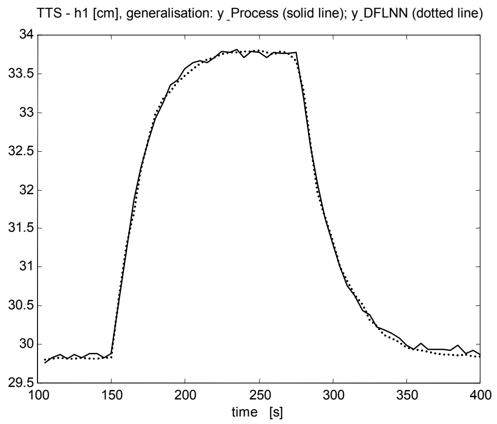


Fig. 6. Three-tank system, output h_1 of the process and of the identified DFLNN model (S=1, m_B=2, n_A=2, m_D=1, n_C=0) – generalisation.

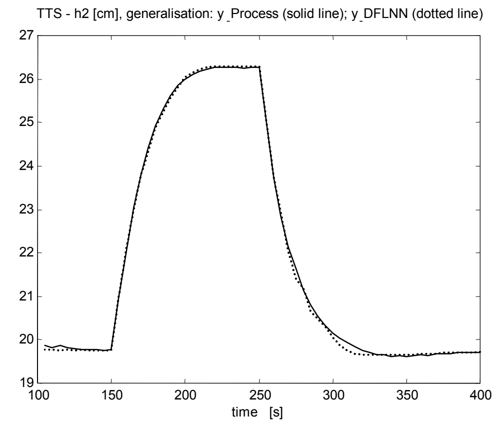


Fig. 7. Three-tank system, output h_2 of the process and of the identified DFLNN model (S=1, m_B=0, n_A=1, m_D=3, n_C=3) – generalisation.

The purpose was to obtain neural models for each system output, i.e. the measured liquid level in each tank:

$$\hat{h}_i[k] = f_i(\mathbf{q}[k], \mathbf{h}[k-1]), \quad i = 1,2,3, \quad k = 1,2,...$$
$$\mathbf{q}[k] = [q_1[k]\ q_2[k]],\ \mathbf{h}[k] = [h_1[k]\ h_2[k]\ h_3[k]].$$

The FLNNs with internal dynamic elements were used to determine the unknown functions f_i. The input-output data set that was used in the training stage, among all the 35 sets available, had the corresponding values for the steady-states closest to the mean values characterising all available data sets. A systematic search for the best approximation was carried out in a similar way to that presented in the previous simulation study. The training of the neural nets was done over 1000 epochs. Figures 6 and 7 illustrate some of the results obtained with data not used in the training (generalisation).

4.3 Identification of an evaporator

Real data from the Lublin sugar factory in Poland were considered (Bartys and Wasiewicz, 1998). The investigated process refers to the heater and the first section of the Evaporation Station (ES) illustrated in Fig. 8. That process is used to reduce the water contents of the sucrose juice. The latter goes through a series of five sections. In each passage the sucrose concentration increases and the steam recovered from one stage is used as a heating source for the next section. No analytical models are known from the sugar factory.

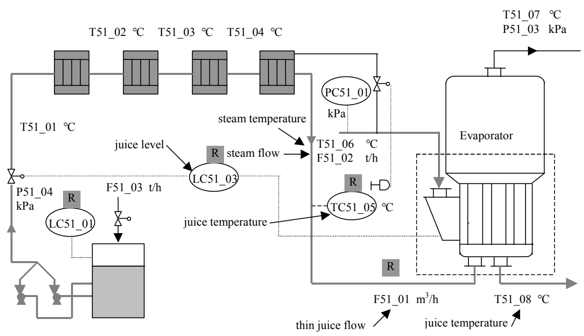


Fig. 8. The heater and first section of evaporation station of the Lublin sugar factory.

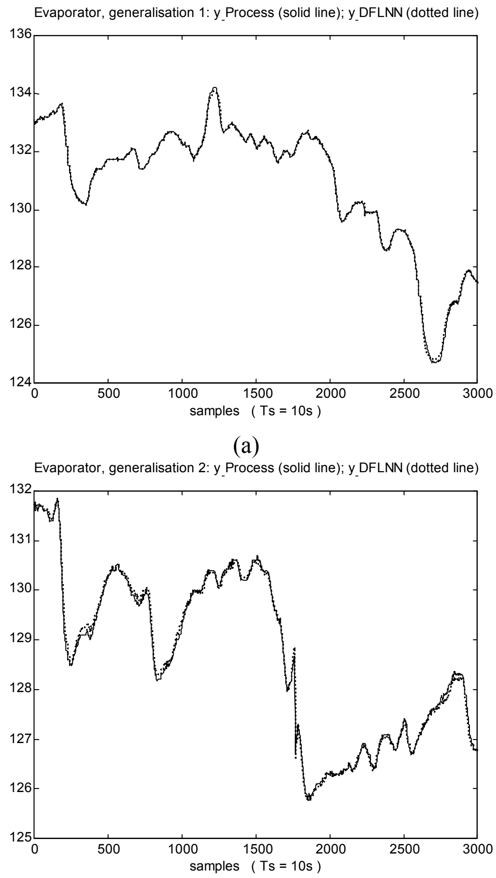
The results related to the identification of the "evaporator" sub-system of ES (Fig. 8) are presented in the following. The considered inputs of the process are: $u_{P,1}$ – the steam flow to the input of ES, $u_{P,2}$ – the steam temperature at the input of ES, and $u_{P,3}$ – the juice temperature after heater. The modelled output is y_P – the juice temperature after section 1 of ES. The data stored during one month, every T_S = 10 seconds, were used to develop a neural model based on DFLNN:

$$\hat{y}[k] := f(\mathbf{u}_P[k], y_P[k-1]); \mathbf{u}_P[k] := [u_{P,i}[k]]_{i=1,...,3}\ .$$

To design the model, a spectral analysis was performed. Based on this, a low-pass filtering by means of appropriate discrete-time Butterworth filters of 4th order was applied to reduce the noise.

This also allowed for the reduction of the amount of data used in the ANN learning. The training data containing 3000 rows of measurements, corresponding to a production shift of 8 hours, were decimated using each 10th sampled value.

The DFLNN was trained for 3000 epochs for each pre-defined value of the order of functional expansion S=1,...,3, and orders of ARMA filters in the set {0,1,2,3}. The best results of identification are presented in Fig. 9 and they correspond to a net with the architecture characterised by S=3, m_B=0, n_A=3, m_D=3, n_C=3.



(b)

Fig. 9. Evaporator, output of the process and of the identified model with the DFLNN: (a) expanded training data set of 3000 rows; (b) testing data set from a previous month of plant exploitation.

5. CONCLUSIONS

The present paper investigates the development and application to system identification of a functional-link neural network with internal dynamic elements. The experimental results obtained by using the suggested neural network reveal its good performances of approximation and generalisation, being characterised by a reduced training and evaluation time.

Further research will investigate the application of the suggested neural network to fault detection and isolation (Marcu, *et al*., 1999). Moreover, the genetic evolving of the presented dynamic neural net architecture is foreseen for an automatic procedure of design. This refers to the genetic selection of the functional expansion order, of the filters' orders, and the optimal placement of the ARMA filters: either before the activation unit of the neuron, or on the connection path from the output to the input of the neuron, or on both positions.

Acknowledgements. The second author would like to acknowledge the support for the completion of this work by the EU-IST-2000-30009 project "Multi-Agents-Based Diagnostic Data Acquisition and Management in Complex Systems" (MAGIC).

REFERENCES

Amira (1993). *Laboratory Experiment Three-Tank System*. Amira GmbH, Duisburg, Germany.

Ayoubi, M. (1996). *Non-linear System Identification Based on Neural Networks with Locally Distributed Dynamics and Application to Technical Processes*. Fortschritt-Berichte VDI, Reihe 8, Nr.591, VDI-Verlag, Düsseldorf.

Bartys, M. and P. Wasiewicz (1998). Description of Sugar Technology Process and Artificial Fault Generation. Prep. *EC INCO - Copernicus Workshop IQ2FD*, Kazimierz, Poland, pp.17-32.

Chen, C.L.P. and J.Z. Wan (1999). A Rapid Learning and Dynamic Stepwise Updating Algorithm for Flat Neural Networks and the Application to Time-Series Prediction. *IEEE Trans. Systems, Man, and Cybernetics – part B*, **29** (1), 62-72.

Hagan. M.T., H.B. Demuth and M. Beale (1996). *Neural Network Design*. PWS Publishing Company, Boston, MA.

Isermann, R., S. Ernst and O. Nelles (1997). Identification with Dynamic Neural Networks: Architectures, Comparisons, Applications. Prep. *IFAC Symposium on System Identification*, Fukuoka, Japan, Vol.3, pp. 997-1022.

Marcu, T., L. Mirea and P.M. Frank (1999). Development of Dynamic Neural Networks with Application to Observer-based Fault Detection and Isolation. *Int. Journal of Applied Mathematics and Computer Science*, **9** (3), 547-570.

Marcu, T., L. Mirea, P.M. Frank and H.D. Kochs (2001). System Identification and Fault Diagnosis using Dynamic Functional-Link Neural Networks. Proc. *European Control Conference,* Porto, Portugal, CD-ROM, pp. 1618-1623.

Pao, Y.H., G.H. Park and D.J. Sobajic (1994). Learning and Generalisation Characteristics of the Random Vector Functional-Link Net. *Neurocomputing*, **6**, 163-180.

Patra, J.C., R.N. Pal, B.N. Chatterji and G. Panda (1999). Identification of Non-linear Dynamic Systems Using Functional-Link Artificial Neural Networks. *IEEE Trans. Systems, Man and Cybernetics – part B*, **29** (2), 254-262.