IDENTIFICATION OF THE NON-LINEAR SYSTEMS USING INTERNAL RECURRENT NEURAL NETWORKS

Gheorghe Puscasu, Bogdan Codres, Alexandru Stancu

"Dunărea de Jos" University Galați, <u>Faculty of Computer Science</u>, Romania

Abstract: In the past years utilization of neural networks took a distinct ampleness because of the following properties: distributed representation of information, capacity of generalization in case of uncontained situation in training data set, tolerance to noise, resistance to partial destruction, parallel processing. Another major advantage of neural networks is that they allow us to obtain the model of the investigated system, systems that is not necessarily to be linear. In fact, the true value of neural networks is seen in the case of identification and control of nonlinear systems. In this paper there are presented some identification techniques using neural networks.

Keywords: identification, recurrent neural networks, training.

1. INTRODUCTION

In the field of the identification of the systems there are two main research directions: the processing of the information through neural networks by neurons and by the types of neural networks structures.

The following neural networks are among neural networks that process the information in a distinct manner:

- neural networks with memory neurons (S. P. Satry, et. al., 1994);
- Elman neural networks (D. Pham and L. Xiug, 1995; P. Gil et al., 1999);
- Jordan neural networks (D. Pham and L. Xiug, 1995).

Also starting from mathematical representation of the systems new neural networks structures were designed (E. D. Rumelhart et. al., 1995). A part of these structures use the state representation of the systems, that in discreet representation are described by the following equations

$$\begin{cases} x(\,k\,)= f(\,x(\,k-1\,),u(\,k-1\,)) \\ y(\,k\,)= g(\,x(\,k\,),u(\,k\,)) \end{cases} \quad (1)$$

where:

$x(k)$ – the state vector;
$u(k)$ – the command vector;
$y(k)$ – the output vector.

Other representations of the system that is used in identification with neural network are Volterra and Wiener models. In this case the neural RBF (Radial Basis Function) has good results.

Usually, experimental identification procedure implies a stage of *data forming* and another stage of *data processing* to adjust network parameters.

During identification it is necessary to obtain as many information as possible regarding the studied process (nonlinearities, type of nonlinearities, degree of system). After obtaining this information it is necessary to establish the type and structure of the network and after that to go over the stage of weights adjustment. The results obtained will be validated through the analysis of residuum properties and the behavior of the obtained model when another set of input signal is used.

For the success of the identification it is necessary to know the aspects regarding the process and engineer experience and intuition.

2. GENERAL ASPECTS REGARDING A SYSTEM IDENTIFICATION BY NEURAL NETWORKS

For a clear presentation of the system identification by neural networks it necessary to establish: the inputs, the outputs and the perturbations of the system's model. In figure no.1 it is presented the structural block schema of the system. In this paper it is supposed that the system contains two sub-systems connected in cascade.
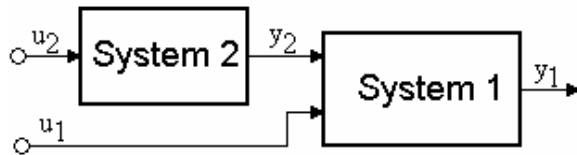
Figure no.1 The block schema of the system

The block schemas represented in the figures above suggest two neural networks structures which will be used at the identification of a system. In the following, the two neural networks structures will be presented.
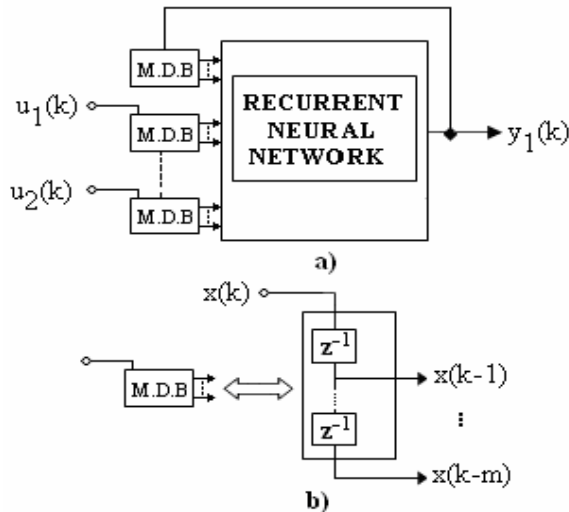

a)

b)

Figure no.2: u1 – the first input; u2 – the second input (y=y1) - the system output
M. D. B. - Memory and Delay Block

For the identification of the system represented in figure no. 1 it is used a recurrent multilayer neural network. In figure no. 2 it is presented the recurrent neural network that is used to identify the system
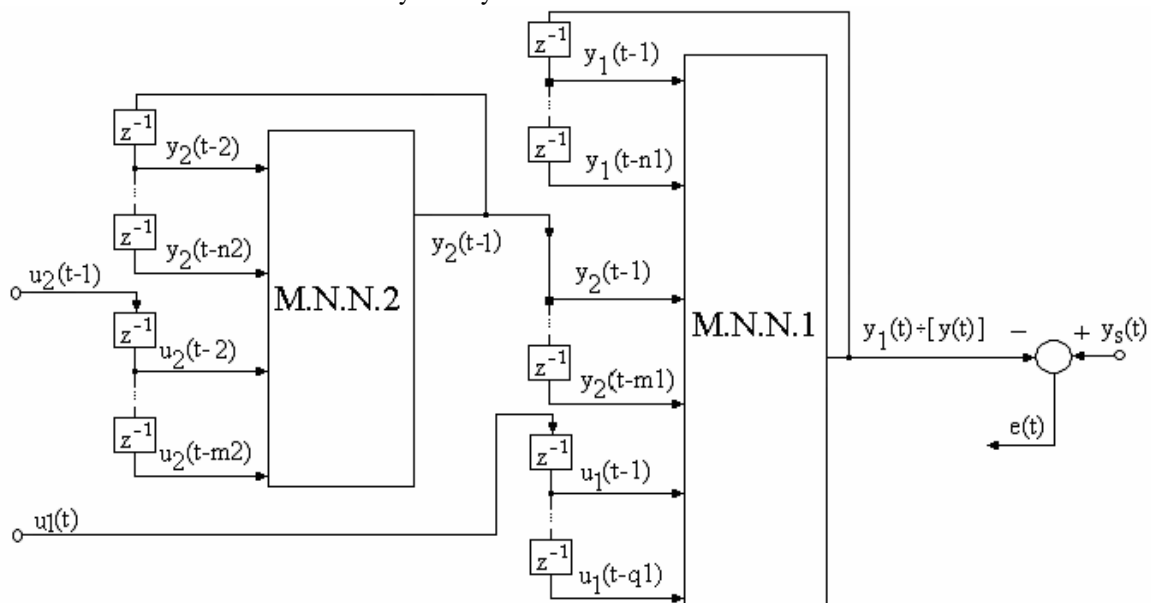
mentioned above.

The neural network from the figure above has the advantage that the necessary steps for the implementation of the neural network structure are easily run through. The major disadvantage of this structure consists in the fact that it requires a large neural network.

The disadvantage mentioned before can be eliminated by using two neural networks, one network for each output. In this case the identification of the system is minimized to the learning of two neural networks.

The block schema of the system represented in figure no.1 suggests the following neural structure (figure no.3) for the system's identification.

Because the suggested structure contains internal elements of delay and memorising, the neural network will be called *internal recurrent neural network*. The two neural networks will identify the subsystems 1 (*M.N.N.1*) and the subsystem (*M.N.N.2*).

Using a neural network with a structure identical to the system, the neural network is able to capture the dynamics of the given system.

Another advantage of this neural structure (figure no.3) is that it allows the determination of the neural models of the component sub-systems. The suggested neural structure can estimate the internal variables of the system without measuring them.

If it is known the neural model of a sub-system, then this can be used in the learning stage of the other neural sub-systems.

In the following paragraph there are presented the aspects regarding the identification of the systems using internal recurrent neural networks.

Figure no.3: u1 – the first input; u2 – the second input; y1-the subsystem1 output;   y2-the subsystem2 output; (y=y1)-the system output .

3. INTERNAL RECURRENT NEURAL NETWORKS

In this section there are presented the aspects regarding the learning of a neural network which identifies a system with a single input and a single output. To simplify, it is considered that the system contains two subsystems.

In figure no. 5 it is shown the schema of an internal recurrent neural network which contains two recurrent neural networks.

Each neural network approximates a nonlinear function given by an expression such as:

$$y_n = f_n(x_1, x_2,...,x_r) \text{, where:}$$

$$(x_1, x_2,...,x_{n1+m1+q1}) = (y_1(t-1),...,$$

- $y_1(t-n1) \; y_2(t-1),..., y_2(t-m1) \;;$

$$u_1(t-1),...,u_1(t-q1))$$

$$y_n(t) = y_1(t) \text{ for neural network } MNN1,$$

- $(x_1, x_2,...,x_{n2+m2}) = (y_2(t-2),...,$

$$y_2(t-n2) \, u_1(t-2),...,u_1(t-m2))\; ;$$

$$y_n(t) = y_2(t-1) \quad \text{for neural network } M.N.N.2.$$

The equations used to learn the *M.N.N.1* and *M.N.N.2* neural networks are the same with equations used to learn the back propagation neural networks (E. D. Rumelhart et. al., 1995).

To exemplify the relations which adjusts the parameters of the two networks it is considered the neuron output *m* which belongs to layer *q* of neural network *k* (k=1 for neural network 1 or k=2 for neural network 2). This output is calculated by the following formula:

$$Ak^q(m) = Fk^q(\sum_{n=1}^{Sk_q} wk^q(m,n) \cdot Ak^{q-1}(n), Bk^q(m)) \quad (2)$$

where:

$Ak^{q-1}(n)$ represents the neuron output *n* which belongs to layer *(q-1)* from network *k* ;

$wk^q(m,n)$ represents the connection between neuron *m* from layer *q* and neuron *n* from layer *(q-1)* of network *k*;

$Bk^q(m)$ represents the bias of neuron *m* from layer *q* of network *k*;

$Fk^q(.,.)$ represents the activation function of neurons from layer *q* of network *k*;

Sk_q represents the number of neurons from layer *q* of network k.

For the parameters adjustment it is used the method of decreasing gradient which minimises the following error criterion:

$$Er = (y(t) - ys(t))^2 \quad (3)$$

With these notations and considering the relation (2) are obtained the following equations which adjusts the parameters of the network:

$$\begin{cases} w1^{qkp}(i,j) = w1^{qkp-1}(i,j) - \eta1(kp) \cdot 2 \cdot e1^{q+1}(j) \cdot A1_i^q \\ B1^{qkp}(i) = B1^{qkp-1}(i) - \eta1(kp) \cdot 2 \cdot e1^{q+1}(j) \end{cases}$$

$$(4)$$

where

$$\begin{cases} e1^n(j) = F1^n{}'(X1^n(j), B1^n(j)) \cdot (y(j) - ys(j)) \\ e1^q(j) = F1^q{}'(X1^q(j), B1^q(j)) \cdot \sum_{p=1}^{S1_{q+1}} e^{q+1}(p) \cdot w1^q(j,p) \end{cases}$$

$$1 \le q < n$$

$$(5)$$

$$X1^q(m) = \sum_{n=1}^{S1_{q-1}} w1^q(m,n) \cdot A1^{q-1}(n) , \quad m = \overline{1, S1_q}$$

$\eta1(kp)$ learning rate for neural network 1.

$$\begin{cases} w2^{qkp}(i,j) = w2^{qkp-1}(i,j) - \eta2(kp) \cdot 2 \cdot e2^{q+1}(j) \cdot A2_i^q \\ B2^{qkp}(i) = B2^{qkp-1}(i) - \eta2(kp) \cdot 2 \cdot e2^{q+1}(j) \end{cases}$$

$$(6)$$

where

$$\begin{cases} e2^n(j) = F2^n{}'(y2(t-1), B2^n(j)) \cdot \sum_{p=1}^{S1_1} e1^1(p) \cdot w1^1(p,i) \\ e2^q(j) = F2^q{}'(X2^q(j), B2^q(j)) \cdot \sum_{p=1}^{S2_{q+1}} e2^{q+1}(p) \cdot w2^q(j,p) \end{cases}$$

$$1 \le q < n$$

$$(7)$$

$$X2^q(m) = \sum_{n=1}^{S1_{q-1}} w2^q(m,n) \cdot A2^{q-1}(n) , \quad m = \overline{1, S2_q}$$

$\eta2(kp)$ learning rate for neural network 2.

Observations

The *M.N.N.2* neural network can have as many inputs and outputs and command variables as are specific to the analysed sub-system. In this situation, the dimension of the input vector increases, while the scientific approach remains unchanged.

In this case, besides the fact that *u(t)* and *ys(t)* are known, it can be measured the output of the sub-system that is approximated to the neural network 2 (*M.N.N.2*). The identification of the system through the neural structure (see figure no. 3) is basically a separate learning of the two networks (Chung-Ming Kuan, 1995; J.Liang and M. M. Gupta, 1999; B. Srinivasan , et. al., 1994). If the output of the sub-

system that is approximated to the neural network 2 (*M.N.N.2*) is not known, then the learning of the two networks must be done simultaneously.

In order to present the way of processing the data and the way of the propagation of the error through the internal recurrent neural structure (figure no.4 and figure no.5), the following hypotheses are presumed:

- the *M.N.N.1* and *M.N.N.2* neural networks have a single neuron of the output layer with the

- linear activation function;
- the connection between the two networks is in fact the connection between the output of the *M.N.N.2* network and the input on the position *i - {x(i)}* of the *M.N.N.1* network;
- the input layer of the *M.N.N.1* network has *q* neurons.

The way of the information processing forward through the network and the way of the backward error propagation is illustrated in the following figures.
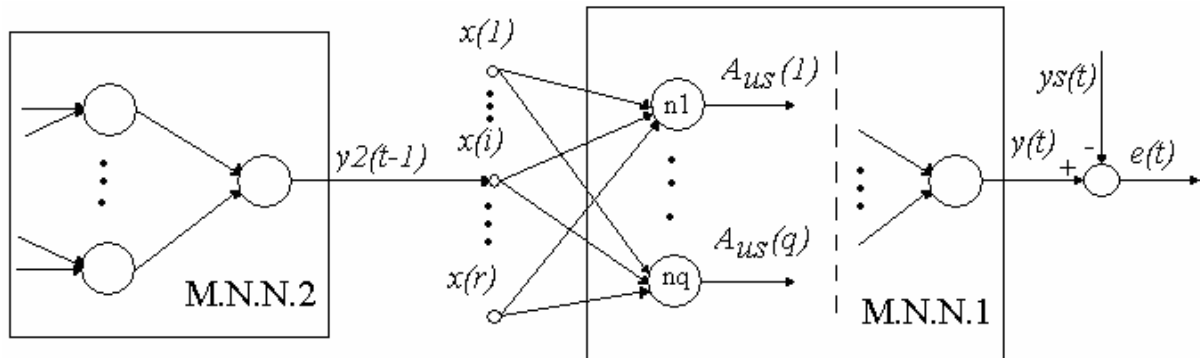

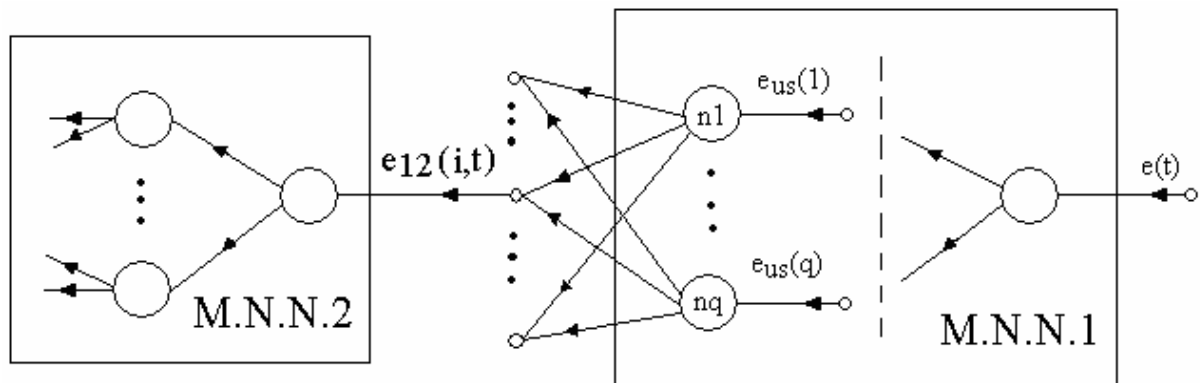Figure no.4 - The information processing forward

Figure no.5 - The backward error propagation

In the figures above there are represented only the variables that are needed in the internal recurrent neural networks presentation. The significations of the variables are:

- *y2(t-1)* output of the *M.N.N.2* that becomes the input *{x(i)}* of the *MNN1* neural network;
- $(x(1), x(2),..., x(r))$ the inputs of the *M.N.N.1* neural network ;
- $(A_{us}(1), A_{us}(2),..., A_{us}(q))$ the neurons outputs of the last layer of the *M.N.N.1* neural network;
- *y(t)* ,*ys(t)* and *e(t)* represent the network output, the system output and the error;
- *e12(i,t)* represents the error *e(t),* that it is propagated through the *M.N.N.1* neural network;

- $\{e_{us}(j)\}_{j=1,q}$ represents the error *e(t),* that is the backward propagation through neuron *j* from the last layer of the *M.N.N.1* neural network.

The error which is propagating through these neural networks *(M.N.N.1 and M.N.N.2)* is:

- $e(t) = (y(t) - ys(t))$ for *M.N.N.1* neural network;

$$e12(i,t) = F^{nern2}\,'(\,y2(t-1),B^{nern2}\,)\cdot$$

- $\sum_{p=1}^{q} e_{us}(p)\cdot w^{us}(p,i)$ for *MNN1* neural network.

In the equation of the error $e12(i,t)$, the variables: $F^{nern2}, B^{nern2}, w^{us}(p,i)$ have the

following significations:

- F^{nern2} - represents the activation function of the output neuron of the *M.N.N.2* network ;

- B^{nern2} - the bias of the output layer neuron of the *MNN2* network;

- $w^{us}(p,i)$ - represents the weight between input i [$x(i)$] and p neuron of the *M.N.N.1* network   last layer.

The neural structure from figure no. 3 has two major advantages:

- the possibility to configure the neural network depending   on the system structure. Also the neural structure used allows the application of the input variables to the hidden layer;

- the possibility of the neural structure learning without knowing the output variable of the sub-systems. This advantage allows to design an estimator of the sub-system output;

The neural structure suggested in figure no. 3 needs a specific algorithm, because the input of the *M.N.N.2*   network $(y2(t-2),...,y2(t-n2))$ modifies itself with each learning stage. In this situation, after each adjustment of the *M.N.N.2* network weigths, the input vector of the neural *M.N.N.2* will be modified.

4. IDENTIFICATION OF THE SYSTEM USING INTERNAL RECURRENT NEURAL NETWORKS

In this paragraph there are presented the practical aspects regarding the identification of the system with internal recurrent neural networks.

For the case study there are used the following equations which describe the systems represented in figure no.1

$$y_1(k) = 4 \cdot \frac{y_1(k-1)}{1+1.2 \cdot y_1^2(k-1)} + y_2(k-1) +$$

$$+(y_1(k-1)+u_1(k-1)) \cdot \frac{1+y_1(k-1)}{1+y_1^2(k-1)}$$

$$(8)$$

$$y_2(k) = 0.5 \cdot [-y_2(k-1)+u_2^2(k-1)+$$

$$+2 \cdot u_2(k-1) \cdot \sqrt{(1+|y_2(k-1)|}]$$

$$.$$

The learning data are presented in figure no. 6 and the data set which is used to learn the internal recurrent neural network is given below.

$$P1 = \{(0 \quad u1(i-1) \quad y1(i-1))_{i=3,nl}^t \}$$

$$P2 = \{(0 \quad u2(i-2))_{i=3,nl}^t \}$$

$$T = \{(y1(i))_{i=3,nl} \}$$

Initially, the first line of the P2 matrix is zero, and the values of the first line will be modified after each learning step. Also, the values of the first line of the P1 matrix are zero initially, but after the learning stage these are set to the values of the *M.N.N.2* output.
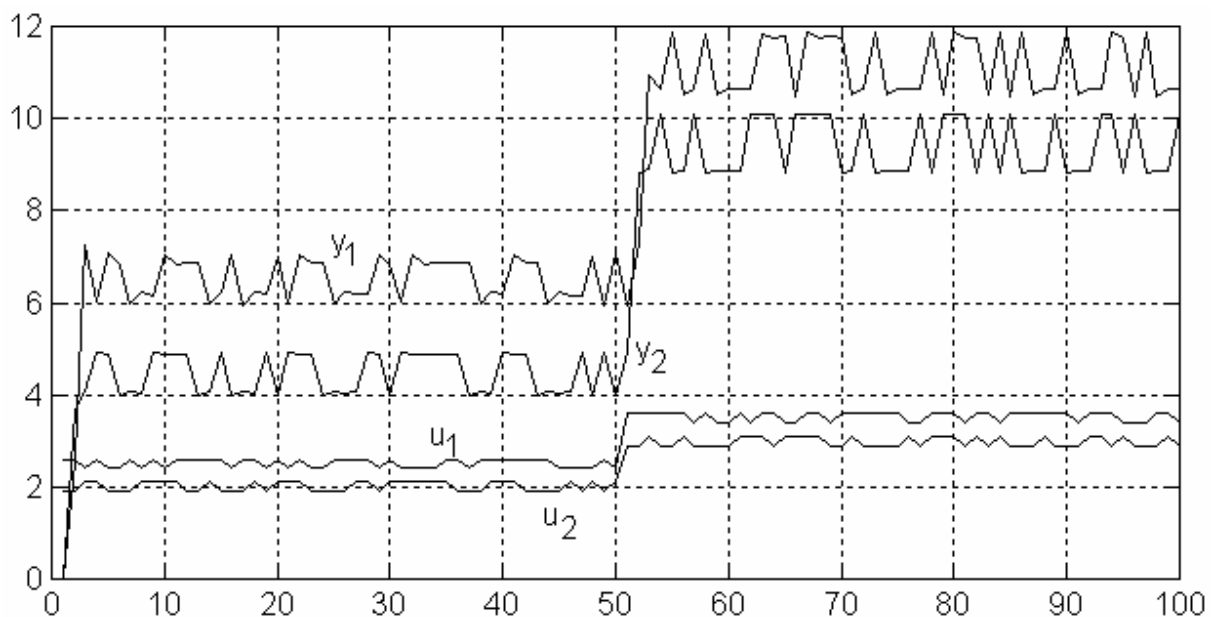

Figure no.6 The learning datas

The structures of these two neural networks are:

The M.N.N.2 neural network has three layers:
- layer 1- 3 neurons and activation function tan-sigmoid,
- layer 2- 7 neurons and activation function tan-sigmoid,
- layer 3- 1 neuron and activation function linear.

The M.N.N.1 neural network has three layers, too:

- layer 1- 5 neurons and activation function tan-sigmoid,
- layer 2- 9 neurons and activation function tan-sigmoid,
- layer 3- 1 neuron and activation function linear.

The parameters of the neural networks are initialized with random sequence.

In the following, there are presented the computed steps of the neural network learning stage.

The learning algorithm

1. It is initialized the j counter with 1;
2. It is computed the M.N.N.2 (y2(j)) output network with the following:

$$y2(j) = f_{RN2}[(P2(1, j)\ P2(2, j)\ P2(3, j))^t\ ]$$

where f_{RN2} represents the non-linear function identified by M.N.N.2 neural network;

3. It is brought up to date the learning lot by the next step :

$$P2(1, j + 1) = y_2(j),\quad P1(2, j) = y_2(j);$$

4. It is computed the output MNN1 network:

$$y(j) = f_{RN1}[(P1(k, j)_{\overline{k=1,10}})^t\ ]$$

where f_{RN1} represents the non-linear function identified by M.N.N.1 neural network;

5. It is computed the error e(j)=T(j)-y(j);
6. It is adjusted the M.N.N.1 network parameters by backwards propagation of the error e(j);
7. It is computed the error   e12(2,j) propagated through MNN1 neural network and through the connection from the MNN2 network output and the M.N.N.1 network input-2.
8. It is adjusted the M.N.N.2 network parameters by backwards propagation of the error e12(2,j);
9. It is increased the j counter j=j+1;
10. If   j<nl (nl-the size of the learning data), then jump to step 2, otherwise continue;
11. STOP.

This algorithm is different from the classic algorithm because it contains the 3 and 7 steps. Through these steps, the learning data are continuously modified.

After the learning stage, the output of the *M.N.N.1* neural network and the output system are illustrated in the figure no.7.

In order to verify the capacity of generalisation, it is used a data set different from the learning data. It can be noticed the fact that the network output (figure no.8) estimates very well the output of sub- system 1.

To highlight the fact that *M.N.N.2* captures the dynamics of the sub-system 2 in figure no.9, there are represented the *M.N.N.2* output and sub-system 2 output. It is observed the fact that those outputs have a higher degree of correlation which suggests the idea that *M.N.N.2* output could be used like an estimator for sub-system 2 output.
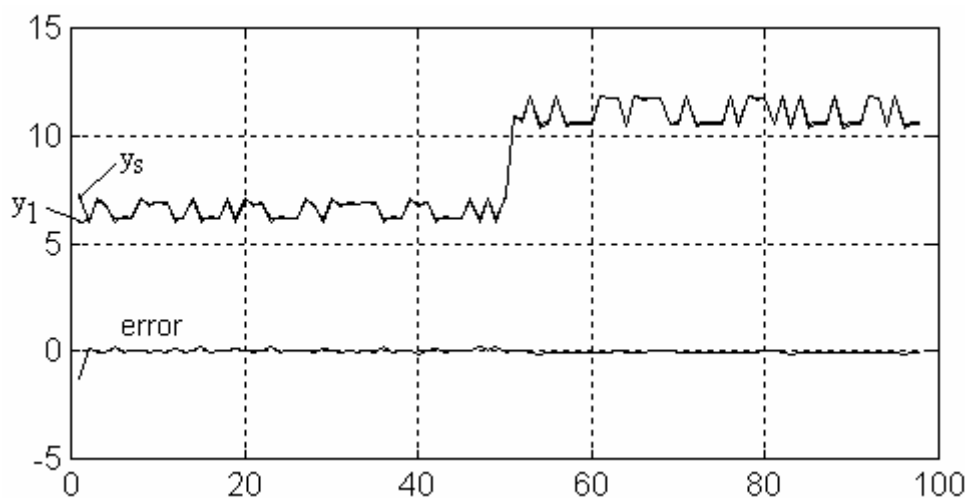

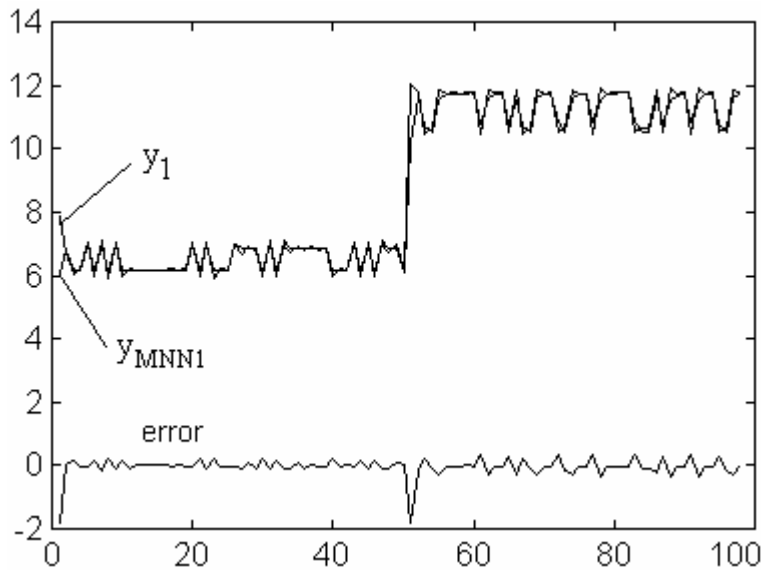Figure no.7 Output of the *M.N.N.1* and system's output

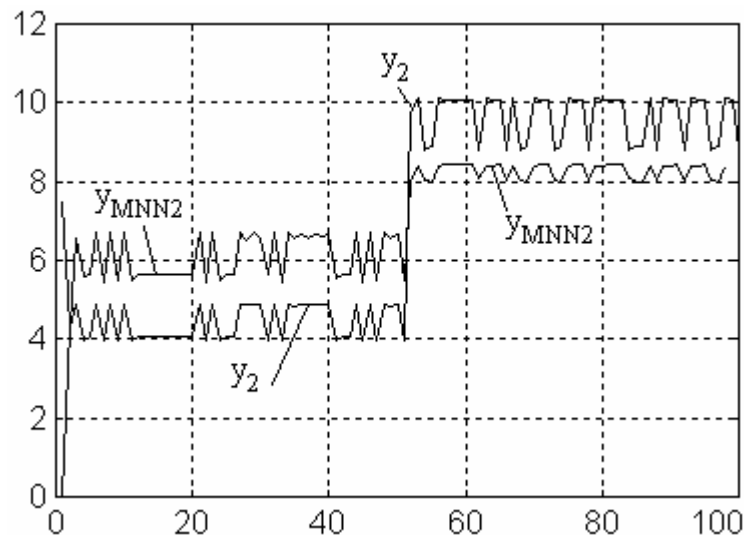Figure no.8 The *M.N.N.1* output and sub-system 1 output

Figure no.9 The *M.N.N.2* output and sub-system 2 output

5. CONCLUSION

Internal recurrent neural network structures suggested in this paper represent a contribution to the identification system fields with neural networks.

This approach has the following features:

- using a neural network with a structure identical to the system's structure, the neural network is able to capture the dynamics of the given system.
- the possibility of independently learning, every neural network if the data learning is available;
- the neural network structure can estimate the internal variables of the analysed system;
- if it is known the neural model of a sub-system, then this can be used in the learning stage of the other neural sub-systems.

6. REFERENCES

S. P. Satry, et. al. (1994) , " *Memory Neuron Networks for Identification and Control of Dynamical Systems* " , IEEE Neural Networks **Vol. 5** No. 2.

D. Pham and L. Xiug, (1995), "*Dynamic System Identification Using Elman and Jordan Networks, Neural Networks for Chemical Engineers*", Editor A. Bulsori, Chap. 23.

P. Gil, J. Henriques, A. Dourado and H. Duarte - Ramos , (1999) " *Non - linear Predictive Control Based on a Recurrent Neural Network*", Proc. ESIT99, June 3-4, Crete, Greece.

E. D. Rumelhart, E. G. Hinton, and J. R. Williams, (1995) " *Learning Internal Representation by Error Propagation* " , in Parallel Distributed Processing. **Vol. 1**, MIT Press, Cambridge.

Chung-Ming Kuan, (1995), "*A Recurrent Newton Algorithm and Its Convergence Properties*" IEEE on Neural Networks **Vol. 6** No. 3.

J.Liang and M. M. Gupta (1999), "*Stable Dynamic Backpropagation Learning in Recurrent Neural Networks*" IEEE on Neural Networks **Vol. 10** No. 6.

B. Srinivasan , et. al. (1994) , " Back Propagation Through Adjoints for the Identification of Non-linear Dynamic Systems Using Recurrent Neural Models ", IEEE Neural Networks **Vol. 5** No. 2,