Identification of Dynamic Processes with Artificial Neural Networks

Jordan Badev[1] and Ivan Maslinkov[2]

Abstract – **In this writing are studied the options for application of Artificial Neural Networks (ANN) in dynamic processes modelling, and mainly in identification and modelling of cases that conventionally are expressed with differential (difference) equations. Special attention has been paid to models identification that is further transformed into ANN training to experimental results of the actual physical model.**

Keywords – **Dynamic Processes Modelling, Artificial Neural Networks, Recursive Perceptron, Discrete AR Models**

I. INTRODUCTION

In A basic approach in designing automation systems is the modelling of separate elements and/or the entire structure of the signal network for process values control. The end results of automation greatly depend on the type and quality of the models employed. The engineering of models that are relevant to the physical processes is done at the beginning of the design stage and is known as identification. Most often subject to identification is the physical process to be controlled, the so called Object of Control. As it is known, ANN in terms of calculation are "heavy" models, since they require computing resources greater than those necessary for other types of analytical models, however they have certain advantages – teach method similar to that of the human beings, solution supply guaranteed, input data noise resistance, "firm" structures (parallel structures) suitable for non-linear and MIMO models, etc. With the massive employment of high capacity computing equipment in the design, the ANN shortcomings are practically insignificant.

For the dynamic control static and dynamic models are used. The output values with static models depend only on the input current values, while with the dynamic models they depend both on current and prior input values. In this writing, for the purpose of clarity, only SISO models are studied, the results, however, being in the greater part relevant for the MIMO models.

It is known and proven that the straight ANN of definite number of neurons in the hidden layers are universal approximators of static functions (models) [1]. **The aim of the present writing** is to explore the ANN capabilities for dynamic processes modelling.

[1]Jordan Badev, D-r, Assistant in the Engineering Faculty of the University of Food Technologies, bul. Mariza 26, Plovdiv, 4000, Bulgaria, E-mail: j_badev@abv.bg

[2]Ivan Maslinkov, D-r, Ass. Prof. in the Engineering Faculty of the University of Food Technologies, bul. Mariza 26, Plovdiv, 4000, Bulgaria, E-mail: imm@uft-plovdiv.bg

II. MODELLING OF DYNAMIC PROCESSES WITH ARTIFICIAL NEURAL NETWORKS

Since the experimental data identification problems are discrete in their nature, herein discrete models and methods are studied.

If the discrete object input signal is $u(k)$ and the discrete object response is $y(k)$, then the general equation that is the discrete image of the continuous differential equation may be presented as:

$$
\begin{aligned}
& y(k)+a_1 * y(k-1)+a_2(k-2)+\ldots+a_n y(k-n)= \\
& b_1 u(k-1)+b_2 u(k-2)+\ldots+b_m u(k-m)
\end{aligned} \tag{1}
$$

where $n \geq m$.

Without losing totality, it is studied the type where the values of $u(k)$ and $y(k)$ participate in a linear way, i.e. it easy to model an expression with these values but of non-linear type. Often in identification these values are called regressors and they are assumed to be known values (measurable).

The finding of an object relevant model of the above type means to determine the coefficients

$$[a_1, a_2, \ldots a_n, b_1, b_2, \ldots b_m] \tag{2}$$

so that when calculating the response $y(k)$ by the expression (model) above given for $k \geq n$, the difference with the measured value $y_o(k)$ to be minimal, i.e. $e(k)=y_0(k)-y(k) \approx 0$.

That problem is easy to solve provided that the necessary experiments are conducted with the object; it is far more difficult if the model is non-linear to the regressors.

If we assume that through ANN processes in the object could be modelled (the dependency $y_0(k)=f(u(k))$), the process of selecting a relevant model can be presented as the chart in Fig.1.

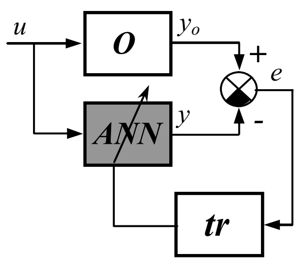

Fig.1 Problem of the identification whit ANN

In this figure:
O - object of identification (control);
ANN - Neural network;
r - ANN training algorithm.

It is seen in the chart that: the object output coincides with the ANN output, i.e. *y(k)*; the process of finding the coefficients Eq.(2) is transformed into a training process of the network with the object's input-output data $[u(k),\ y_o(k)]$; the identification pattern is one and the same for identification of static, dynamic, linear, non-linear etc. models.

The model type is determined by the ANN model and parameters, and the input-output data for training.

The very nature of dynamic models supposes the use of more complex than the strait ANN structures, namely the ANN recurrent structures [2]. Characteristic of these structures is modelling of earlier (following) values of the quantities through unitary elements (z^{-1}), for retaining the information in one stroke (discretization period - T_0) .

For example: Eq.(1) could be presented with the structure in Fig.2.

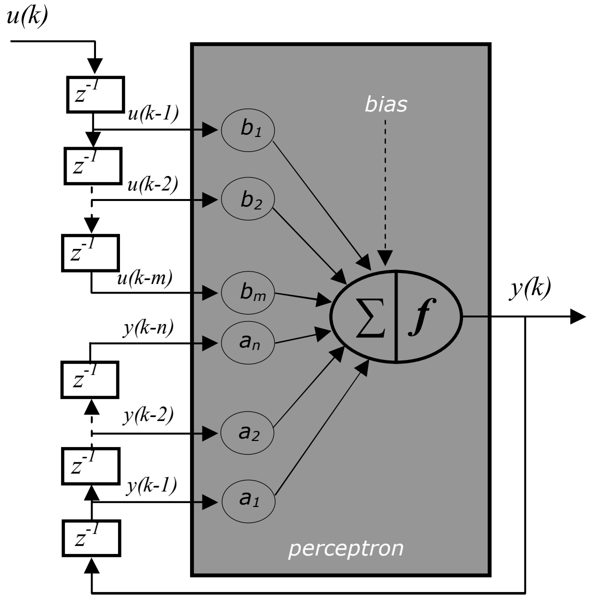

Fig.2 Linear ARX model

Obviously the chart in the grey (big) square is a linear perceptron that is a unitary structure building the ANN. It is so because Eq.(1) is of the autoregression type (AR). In this case it is evident that coefficients in Eq.(1) coincide with the connection weights at the perceptron input, so it is natural that the process of finding the coefficient Eq.(2) be transformed into an ANN training process.

With more complex equations the chart shall contain a greater number of and/or with non-linear perceptrons, possibly distributed in different layers (multilayer), in which case there is no congruence between connection weights and coefficients and the obtained model is of the "black box" type, namely ANN.

It is not hard to present the models in the space state.

If is writen the following for this type SISO models in general:

$$\begin{aligned} x(k+1) &= f_1(x(k),u(k)) \\ y(k) &= f_2(x(k)) \end{aligned} \tag{3}$$

In this model *x(k)* is *n*–dimensional vector and the neural model is demonstrated with the chart in fig. 3

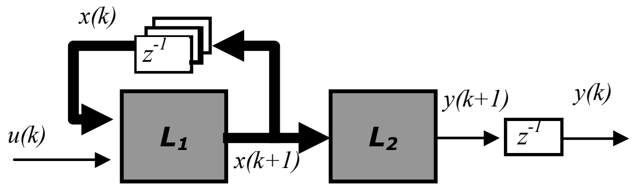

Fig.3 Neural model in the space state.

In this figure the blocks L_1 and L_2 perform respectively the functions f_1 and f_2 that generally can be implemented by a certain number and type of perceptrons distributed in two layers.

If it is a linear object, the functions are linear, and so are the perceptron activating functions in the two layers. This is the case applicable for the AR models, most widely used in practice. Here the required computing capacity is minimal and the training process is the fastest. In many cases the non-linear models are modelled with only the first, L_1 neural layer non-linear.

III. Experimental Data and Results

The identification of actual physical objects is frequently done by experimental transient characteristics that contain the basic information of the static and dynamic properties, and therefore are used for objective evaluation of the control quality. The models are relevant if they produce one and the same response to various input signals.

In this writing, for better flexibility is haven used data simulated with known discrete transfer function (TF) - *W(z)*. The validation of the neural model is through visual comparison between the model response and the neural model response.

The sufficient ANN for modelling linear models is a single-layer linear neuron (one linear neuron) of one output and inputs depending on the model's order – Fig. 2. The training process is up to the level of the mean square difference (MSE) between the network output $1*10^{-10}$, by the Levenberg-Marquardt recursion (*trainlm*).

Training set is known with: the response values $y(k)$; the amplitude of input signal $u_{tr}(k)$; is the measuring period T_0. u_{test} is the testing of neural model input signal.

In this writing, through ANN have been modelled TF of a periodic second order object -Eq.4 and fifth-order oscillatory object – Eq.5.

$$W(s) = \frac{0.003121*z^{-1} + 0.001*z^{-2}}{1 - 1.01*z^{-1} + 0.08208*z^{-2}} \tag{4}$$

$t(0)=0;\ t(end)=4$ sec.; $T_0=0.05$ sec.

The ANN is taught with $u_{tr}=1(k)$, for 5 epochs and the following difference is achieved MSE=$4.25*10^{-12}$.

The test results with $u_{test}=1(t)=u_{tr}$ are given in Fig.4.

The test results with $u_{test}=1.2*1(t)$ are given in Fig .5.

The test results with $u_{test}=5*sin(t)$ are given in Fig.6.

In all graphs:

1 - object response (TC);

2 - neural model response.

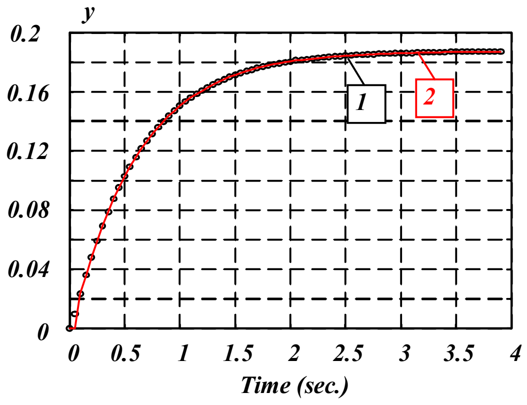

Fig.4 Response with $u_{test}=u_{tr}=1(t)$

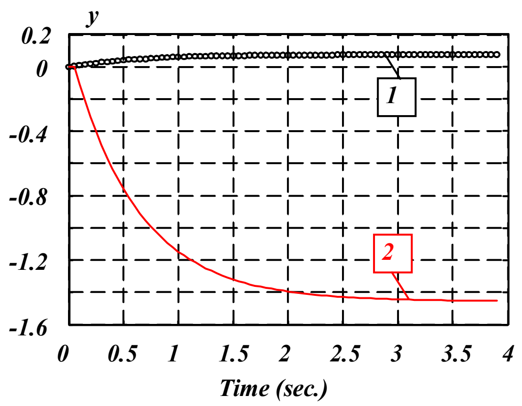

Fig.5 Response with $u_{test}=1.2*1(t)$

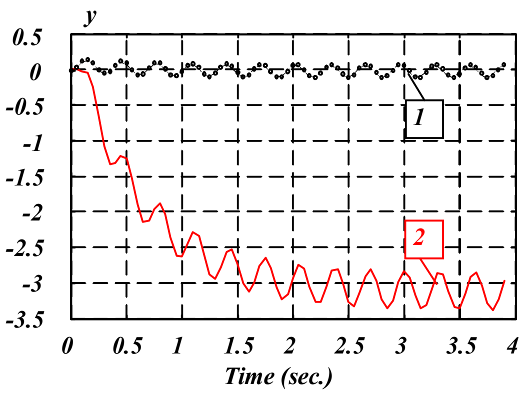

Fig.6 Response with $u_{test}=5*sin(t)$

In fig.7 the response of the ANN that is taught and tested with the same ramp input, i.e. $u_{trt}=1*t=u_{test}$.

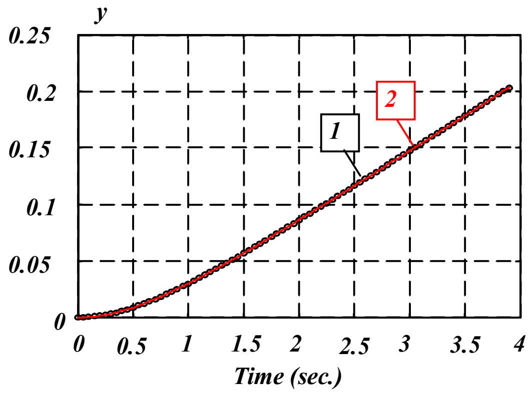

Fig.7 Response with $u_{tr}=u_{test}=t$

The oscillatory object is modeled by TF - Eq.(5).

$$W(s)=\frac{0.02056*z^{-1}+0.07613*z^{-2}+0.0177z^{-3}}{1-2.662*z^{-1}+2.417*z^{-2}-0.7408z^{-3}} \tag{5}$$

$t(0)=0$; $t(end)=160$ sec.; $T_0=1$ sec.

The ANN is taught with $u_{tr}=5*1(t)$ for 3 epochs and a difference of MSE=$4.25*10^{-10}$ is achieved.

The test results with $u_{test}=5*1(t)=u_{tr}$ are given in Fig .8.

The test results with $u_{test}=1.2*1(t)$ are given in Fig .9.

The test results with $u_{test}=5*sin(t)$ are given in Fig .10.

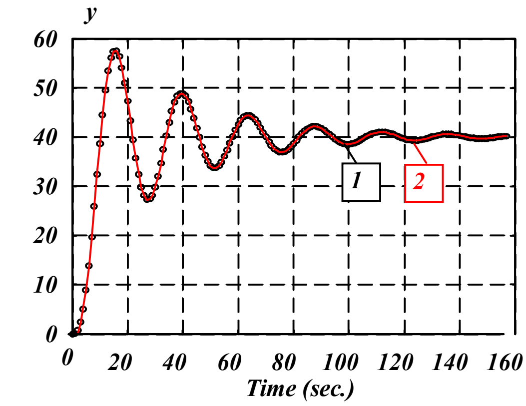

Fig.8 Response with $u_{tr}=u_{test}=5*1(t)$

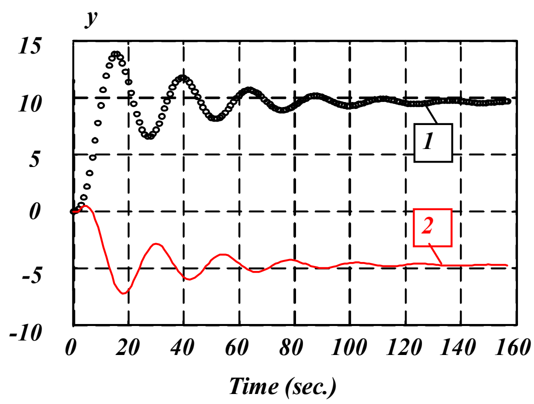

Fig.9 Response with $u_{test}=1.2*1(t)$

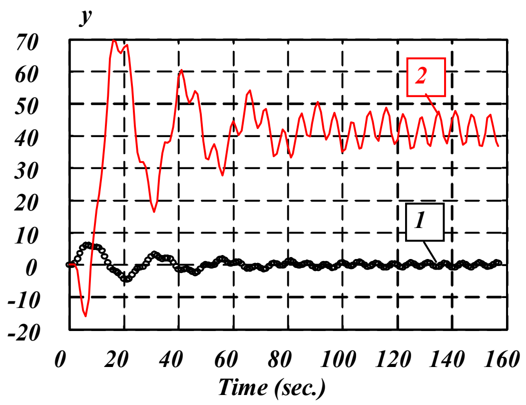

Fig.10 Response with $u_{test}=5*sin(t)$

Applying the same method, it has been performed training and testing of other neural models intended for modelling other functions, the results of which are skipped herein, since they are similar.

IV. CONCLUSIONS AND RECOMMENDATIONS.

All results lead to the following major conclusions:

1) Through recursive ANN it is possible to model linear dynamic functions of random order only for the input signal with which they have been taught - Figs. 4, 7 and 8.

2) The input training signal may be of arbitrary type (step, ramp, sinusoidal, etc.) - Figs. 4, 7 and 8.

3) For linear objects, relevant neural models may be obtained also for input signals different from the amplitude training step ones, provided that the ANN is taught by the standard response characteristic, and the amplitude Δu is accounted separately as a static amplification coefficient of object, i.e.

if for $u=1(t)$, the ANN response is $y(t)$, then for $u=\Delta u*1(t)$, the neural model response is $y(t)=\Delta u*y(t)$

4) ANN with single layer of neurons are preferred with modelling linear dynamic functions due to greater accuracy and faster teaching.

5) With non-linear functions it is preferable to use double-layer structures of first layer non-linear neurons, and one linear in the second layer.

6) With ANN taught by the classical method (back-propagation; Levenberg-Marquardt, etc.) dynamic functions of input signal different from the training one can not be modelled – Figs. 5, 6, 9 and 10. This is explained with the fact that networks store their inputs. They find one of the possible solutions.

7) Taking into account the above examples and the arguments, obviously with ANN modelling we must consider the fact that the “black box” models have inaccessible state variables (unknown underlying structure), which to some extent limits their application.

REFERENCES

[1] K. Hornik, M. Stinchcombe, H. M&White, Multilayer feedforward networks are universal approximators (Discussion paper 88-45), San Diego, CA, Department of economics, University of California, 1988.

[21] S. Haykin, Neural Networks (sec.edition), Prentice Hall, Pearson education, 1999.