DISCRETE FRACTIONAL ORDER ARTIFICIAL NEURAL NETWORK

Dominik SIEROCIUK*, Grzegorz SARWAS*, Andrzej DZIELIŃSKI*

*Institute of Control and Industrial Electronics, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland

dsieroci@isep.pw.edu.pl, sarwasg@isep.pw.edu.pl, adziel@isep.pw.edu.pl

Abstract: In this paper the discrete time fractional order artificial neural network is presented. This structure is proposed for simulating the dynamics of non-linear fractional order systems. In the second part of this paper several numerical examples are shown. The final part of the paper presents the discussion on the use of fractional or integer discrete time neural network for modelling and simulating fractional order non-linear systems. The simulation results show the advantages of the proposed solution over the classical (integer) neural network approach to modelling of non-linear fractional order systems.

1. INTRODUCTION

Extending a highly desirable genericity of linear dynamic systems models to non-linear systems has for quite some time occupied control theorist. The main reason of the problems with obtaining generic models for non-linear systems is the complex behaviour associated with nonlinearity and its intrinsic locality. Thus the search for a universal non-linear model is highly non-trivial, as is the underlying problem of classification of non-linear systems. An important feature of a candidate for such a model is that it be parameterised to make finite-dimensional identification techniques applicable. Moreover, the model should be tractable from the control point of view as it is only an auxiliary step in the overall closed-loop system design. In this context we attempt to analyse and extend the application of neural networks for control. The neural networks can be treated as candidates for a generic, parametric, nonlinear model of a broad class of non-linear plants (see e.g. Hunt et al. (1995); Żbikowski and Hunt (1996); Kalkkuhl et al. (1997); Nørgaard et al. (2000)). Neural networks have modelling capabilities to a desired accuracy, however it is not entirely clear how they represent the plant's system properties. A remarkable progress in the investigations on the representational capabilities of neural networks in recent years not only validate them as the models, but also give interesting and practical suggestions for further research. Boroomand and Menhaj (2009); and Benoit-Marand et al. (2006) present continuous time description of neural networks for modelling nonlinear fractional order systems. In this paper the discrete approach is considered.

In many cases the use of feedforward neural networks for non-linear control is based on the input-output discrete-time description of the systems

$$y_{k+n} = f(y_k, \dots, y_{k-n+1}; u_k, \dots, u_{k-m+1}). \tag{1}$$

However, this model has rather limited capabilities for modelling the fractional order systems. Thus, in this paper we suggest the use of the fractional order calculus to built a model of a non-linear system in the form

$$\Delta^{n\alpha} y_{k+n} = f(\Delta^{(n-1)\alpha} y_{k+n-1}, \dots, \Delta^{\alpha} y_{k+1}, y_k, u_k). \tag{2}$$

The model proposed may turn out to be of lower order and may better reflect the dynamic properties of the fractional order system modelled.

2. DISCRETE FRACTIONAL ORDER NON-LINEAR SYSTEM

To present fractional order discrete time neural network we have to introduce discrete fractional order non-linear system. In this paper the following definition of the fractional order difference is used (see e.g. Oldham and Spanier (1974), Podlubny (1999)):

Definition 1. Fractional order difference is given as follows:

$$\Delta^{\alpha} x_k = \sum_{j=0}^{k} (-1)^j \binom{\alpha}{j} x_{k-j} \tag{3}$$

where, $\alpha \in \boldsymbol{R}$ is a fractional order and $k \in \boldsymbol{N}$ is a number of sample for which the difference is obtained.

In our case the artificial neural network is used to model the fractional order non-linear systems. Using fractional order difference the following non-linear discrete fractional order system in the state-space description is defined:

Definition 2. The non-linear discrete fractional order system in a state-space representation is given by the following set of equations:

$$\Delta^{\alpha} x_{k+1} = f(x_k, u_k) \tag{4}$$

$$x_{k+1} = \Delta^{\alpha} x_{k+1} - \sum_{j=1}^{k+1} (-1)^j \binom{\alpha}{j} x_{k+1-j} \tag{5}$$

$$y_k = h(x_k) \tag{6}$$

The system which we take into consideration is given as the following relation:

$$\Delta^{n\alpha} y_{k+n} = g(\Delta^{(n-1)\alpha} y_{k+n-1}, \dots, \Delta^{\alpha} y_{k+1}, y_k, u_k) \tag{7}$$

which can be rewritten as:

$$\begin{aligned} \Delta^{\alpha} y_{1,k+1} &= x_{2,k} \\ \Delta^{\alpha} y_{2,k+1} &= x_{3,k} \\ &\vdots \\ \Delta^{\alpha} x_{n,k+1} &= g(y_k, x_{2,k}, \dots, x_{n,k}, u_k) \end{aligned} \tag{8}$$

This system can be modelled using the artificial neural network presented in the next section.

3. DISCRETE TIME FRACTIONAL ORDER NEURAL NETWORKS

Neural networks have good properties to model the dynamics of the non-linear systems. In fact they are treated as a candidate for a generic, parametric, non-linear model of a broad class of non-linear systems, because they have modelling capabilities to a desired accuracy. Irrespective of system order so far the system scientists have proposed the integer order neural network for modelling integer or non-integer order system. In case of using standard (integer order) neural network for fractional systems modelling the network structure is complicated and the accuracy can be insufficient. Better solution can be achieved using fractional order neural network of the form.

This structure is a combination of a standard neural network and a linear discrete fractional order state-space system (DFOSS) defined below.

Definition 3. Linear discrete fractional order system in the state-space representation is given as follows (see e.g. Sierociuk and Dzieliński (2006)):

$$\Delta^{\alpha} x_{k+1} = Ax_k + Bu_k \tag{9}$$

$$x_{k+1} = \Delta^{\alpha} x_{k+1} - \sum_{j=1}^{k+1} (-1)^j \binom{\alpha}{j} x_{k-j+1} \tag{10}$$

$$y_k = Cx_k + Du_k \tag{11}$$

where, $\alpha \in \boldsymbol{R}$ is a system order.

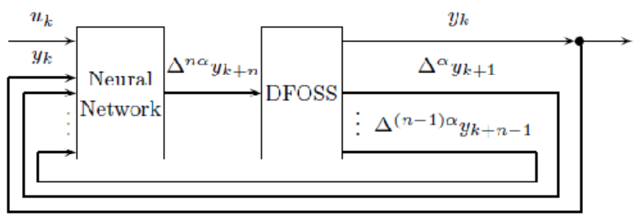

Fig. 1. Discrete time fractional neural network

Fig. 1 presents the architecture useful to simulate the fractional order neural network. It can be noticed, that the neural network is a traditional structure which choice is dependent on the modelled system. The neural network input signals are the system input and output data for the k sample (u_k, y_k) and the vector differences between previous outputs from $\Delta^{(n-1)\alpha} y_{k+n-1}$ to $\Delta^{\alpha} y_{k+1}$. In the output of the neural network we obtain the prediction of the next step difference $\Delta^{n\alpha} y_{k+n}$. Using this value DFOSS calculates the value of the system output and a new vector of differences. DFOSS blocks' sizes depend on the modelled system structure and the system matrices we can be obtained in the following way:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \mathrm{B}=\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, C = [I], D = [0]. \tag{12}$$

The structure discussed in this section can be used for offline simulation of the modelled non-integer order dynamics. In order to apply it to an on-line application in control one needs to use $\Delta^{n\alpha} y_{k+n}$ (output of the network) to calculate the system output using only previous signals samples which are available.

In the next section we present the numerical example which illustrates the operation of the proposed structure.

4. NUMERICAL EXAMPLE

For all the simulations two groups of input signals were prepared. The first group of four signals was meant for learning process and the second group of two signals was used for testing process. The learning and testing signals are presented in Fig. 2 and Fig. 3 respectively. Final results of neural modelling were obtained by on-line simulations in Simulink using Neural Network Toolbox and Fractional State-Space Toolkit (FSST) (see e.g. Sierociuk (2005)).

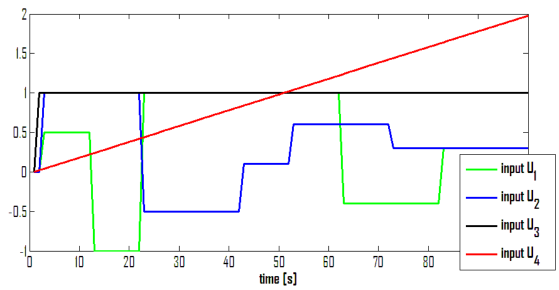

Fig. 2. Input learning signals $\boldsymbol{U_1}, \boldsymbol{U_2}, \boldsymbol{U_3}, \boldsymbol{U_4}$

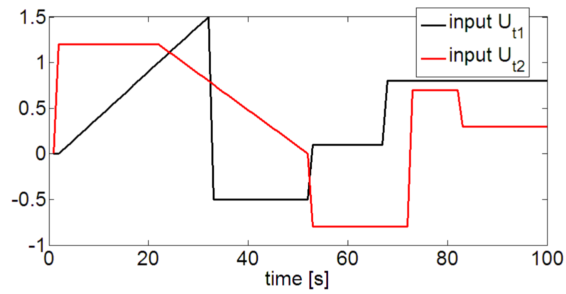

Fig. 3. Input testing signals $\boldsymbol{U_{t1}}, \boldsymbol{U_{t2}}$

Example 1. Modelling of the fractional system by a discrete fractional order neural network

The system is given by the following equation:

$$\Delta^{0.5} y_{k+1} = -0.1 y_k^3 + u_k \tag{13}$$

For modelling the non-linear function in this system a two-layer neural network with two inputs and one output was used. The network consists of three neurons with nonlinear (tansign) activation function in the input layer and one linear neuron in the output layer.

The input vector for the fractional neural network for this case has the following form:

$$P = \begin{bmatrix} u_0 & u_1 & \cdots & u_k \\ y_0 & y_1 & \cdots & y_k \end{bmatrix} \tag{14}$$

The output vector has the form:

$$T = [\Delta^{0.5} y_1 \ \Delta^{0.5} y_2 \ \cdots \Delta^{0.5} y_{k+1}] \tag{15}$$

The DFOSS block has the following matrices:

$$A = [0], B = [1], C = [1], D = [0]. \quad (16)$$

The order of this block is equal to $\alpha = 0.5$ and is the same as the order of the given system equation.

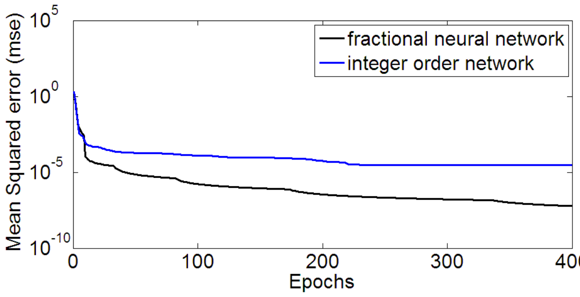

Fig. 4. Learning error for fractional and integer order neural network

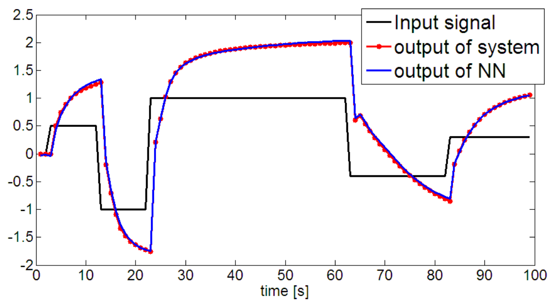

Fig. 5. Output of the fractional neural network for learning signal $\boldsymbol{U_1}$

For training the neural network the Levenberg-Marquardt and backpropagation algorithms (implemented in function TRAINLM in Neural Network Toolbox) were used for 400 epochs. The results of the performance of the network during learning process is presented in Fig. 4 (together with results of integer order neural network). As it could be seen the final error is very small, about 10^{-7}. Fig. 5 presents a comparison between the responses of the fractional neural network with original output for input signal U_1 from the group of learning signals. As it could be seen the accuracy of modelling is very high. Moreover, the Fig. 6 and Fig. 7 present analogical results for test signals U_{t1} and U_{t2} respectively. This results prove very high accuracy of modelling and confirm that the neural network has been properly taught. This also shows that the network is able to properly generalize data, which is the main feature of neural networks. Fig. 8 and Fig. 9 present results of neural modelling of the non-linear function. Fig. 8 presents the original non-linear function of the system equation, whereas Fig. 9 presents the modelling error, the difference between original function and the one modelled by neural network.

Example 2. Modelling of the same fractional system as in Example 1 by a discrete integer order neural network. In this example the traditional approach will be presented in which we try to model non-linear fractional order system by a non-linear integer order system with some (usually big) number of delays. Let us take into consideration the neural network with 5 inputs and one output. The neural network used has the following structure: input layer has 6 tansign neurons, output layer has one linear neuron. In this case the modelled equation has the form:

$$y_{k+4} = f(y_{k+3}, y_{k+2}, y_{k+1}, y_k, u_k) \quad (17)$$

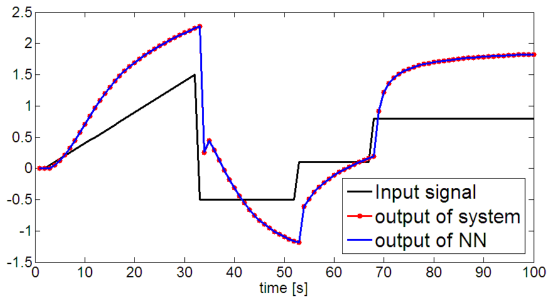

Fig. 6. Output of the fractional neural network for testing signal $\boldsymbol{U_{t1}}$

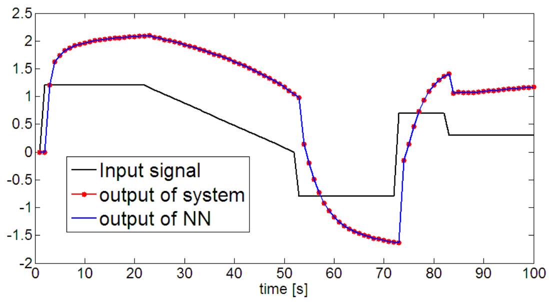

Fig. 7. Output of the fractional neural network for testing signal $\boldsymbol{U_{t2}}$

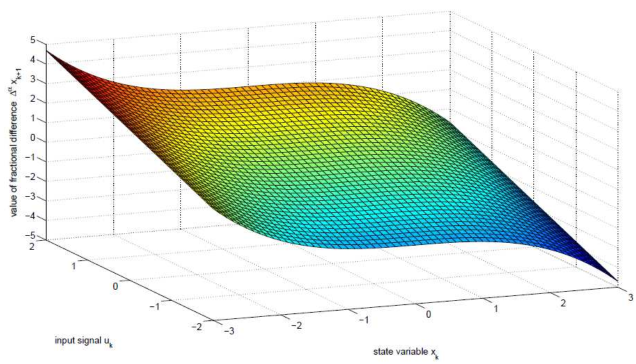

Fig. 8. Original non-linear function

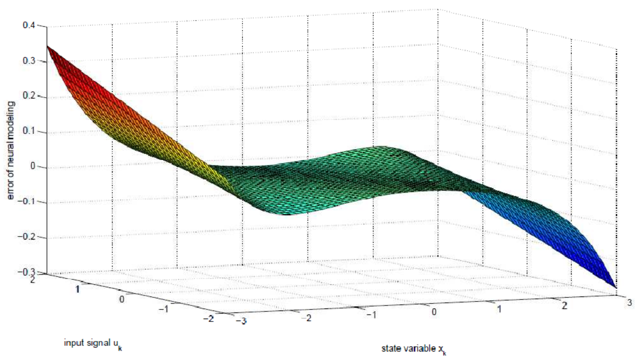

Fig. 9. Error of neural modelling

The results of the neural network performance during learning process are presented in Fig. 4. Fig. 10 presents results of simulation for one signal U_1 from learning group. As it may be seen the accuracy is acceptable. Additionally, the Fig. 11 and Fig. 12 present analogical results for the test signals U_{t1} and U_{t2} respectively. As it may be noticed, obtained results show unacceptable accuracy. In this case the integer order neural network is not able to properly generalize the data of the system, despite of the more complicated structure of a neural network. This justifies the main advantage of the proposed algorithm.

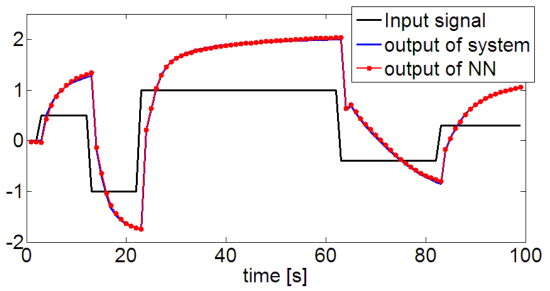

Fig. 10. Output of the integer order neural network for learning signal $\boldsymbol{U_1}$

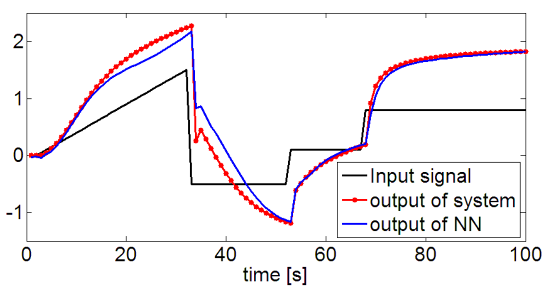

Fig. 11. Output of the integer order neural network for testing signal $\boldsymbol{U_{t1}}$

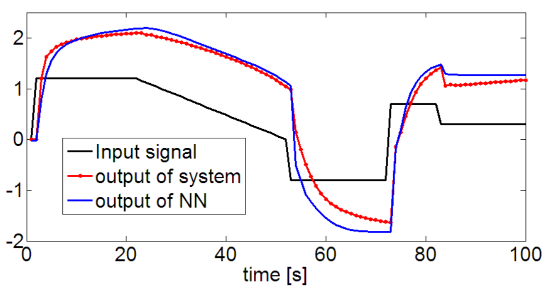

Fig. 12. Output of the integer order neural network for testing signal $\boldsymbol{U_{t2}}$

Example 3. Modelling of the fractional system with two state variables (one hidden) by the discrete fractional order neural network

Let us assume the system is given by the following equation:

$$\Delta^1 x_{1,k+2} = -0.1 x_{2,k}^3 + u_k \tag{18}$$

where

$$x_{2,k} = \Delta^{0.5} y_{1,k+2} \tag{19}$$

In this case for modelling the non-linear function the two layer neural network with 3 inputs and one output was used. This network consisted of 3 neurons with non-linear (tansign) activation function in input layer and one linear neuron in output layer.

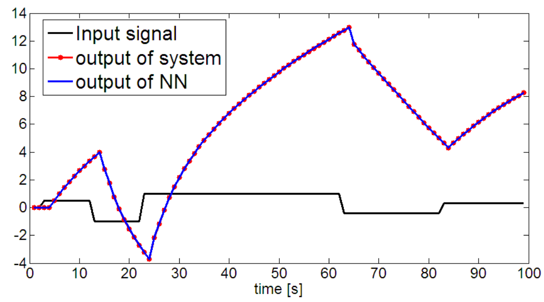

Fig. 13. Output of the fractional order neural network for learning signal $\boldsymbol{U_1}$

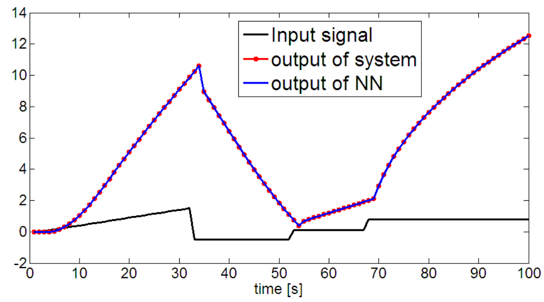

Fig. 14. Output of the fractional order neural network for testing signal $\boldsymbol{U_{t1}}$

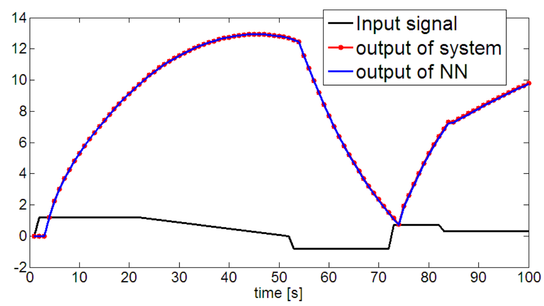

Fig. 15. Output of the fractional order neural network for testing signal $\boldsymbol{U_{t2}}$

The input vector for the fractional neural network has the following form:

$$P = \begin{bmatrix} u_0 & u_1 & \cdots & u_k \\ \Delta^{0.5} y_1 & \Delta^{0.5} y_2 & \cdots & \Delta^{0.5} y_{k+1} \\ y_0 & y_1 & \cdots & y_k \end{bmatrix} \tag{20}$$

The output vector has the form:

$$T = [\Delta^1 y_2 \; \Delta^1 y_3 \; \cdots \; \Delta^1 y_{k+2}] \tag{21}$$

The DFOSS block has the following matrices:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{22}$$

For training the neural network the same conditions as in previous examples were used. The results of an on-line simulation for one of the learning input signals is presented in Fig. 13, while Fig. 14 and Fig. 15 present results for the testing input signals. As it may be seen the accuracy of neural modelling is very high.

5. CONCLUSIONS

In the paper we proposed a discrete time fractional order neural network. The structures given can be used to model the nonlinear fractional order dynamic systems. We have shown the appropriateness of the approach by the numerical examples. Also the advantages in modelling the fractional order discrete-time dynamic systems with the structure proposed over the traditional neural network coupled with the tapped-delay line have been shown in several example cases. Further research is needed to show the theoretical properties and advantages (and limitations) of the approach suggested.

REFERENCES

1. **Benoit-Marand F., Signac L., Poinot T., Trigeassou J.C**. (2006), Identification of non linear fractional systems using continuous time neural networks, *Proceedings of 2nd IFAC Workshop on Fractional Differentiation and its Applications*, IFAC FDA'06.
2. **Boroomand A., Menhaj M. B.** (2009), *Fractional order hopfield neural networks*, Lecture Notes in Computer Science: Advances in Neuro-Information Processing, 5506, 883–8902.
3. **Hunt K., Irwin G. Warwick K.** (1995), *Neural Network Engineering in Dynamic Control Systems*, Springer.
4. **Kalkkuhl J., Hunt K., Zbikowski R., Dzieliński A.** (1997), *Applications of neural adaptive control technology*, World Scientific.
5. **Nørgaard M., Ravn O., Poulsen N., Hansen L.** (2000), *Neural Networks for Modelling and Control of Dynamic Systems*, Springer.
6. **Oldham K., Spanier J.** (1974), *The Fractional Calculus. Academic Press*, New York.
7. **Podlubny I.** (1999), *Fractional Differential Equations*, Academic Press, San Diego.
8. **Sierociuk D.** (2005). Fractional Order Discrete State-Space System Simulink Toolkit User Guide http://www.ee.pw.edu.pl/˜dsieroci/fsst/fsst.htm
9. **Sierociuk D., Dzieliński A.** (2006), Fractional Kalman filter algorithm for states, parameters and order of fractional system estimation, *International Journal of Applied Mathematics and Computer Science*, 16(1), 129–140.
10. **Żbikowski R,. Hunt K.** (1996), *Neural adaptive control technology*, World Scientific.

This work was partially supported by the Polish Ministry of Science and Higher Education grant number 4125/B/T02/2009/36 and the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme (by Centre for Advanced Studies WUT).