Neuro-Observer with Application to Longitudinal Motion of an Aircraft with Big Attack Angle

Mihai Lungu
University of Craiova, Faculty of Electrical Engineering, Craiova, Romania
Lma1312@yahoo.com, mlungu@elth.ucv.ro

Abstract— **In this paper a neural network observer for nonlinear systems is presented. The proposed neuro-observer is a three-layer feedforward neural network (NN), trained by means of the error backpropagation learning algorithm; according to this algorithm, the neural network training process becomes a nonlinear function optimization problem. The weights and the biases are permanently modified in order to minimize the mean squared error between the actual outputs and the NN desired outputs in a gradient descent manner. The good results of the neural networks are due to their capacity of nonlinear functions' approximation. The observer also includes a correction term which guarantees the good tracking as well as bounded neural network weights. The neural network is used to parameterize the nonlinearities of the system. The validation of the proposed observer scheme is made through Matlab/Simulink numerical simulation to reconstruct the unavailable state variables of a big attack aircraft longitudinal motion. In fact, the motion of the aircrafts with big attack angle is a nonlinear and complex system, which makes difficult the design and the implementation of efficient control and observation laws. It will be shown that all the components of the error vector tend to zero, this fact proving both the proper functioning of the NN and the very good estimation of the state variables.**

I. Introduction

The state estimation problem is widely presented in many scientific papers. In most cases only the input (inputs) and the output (outputs) of a system are measurable and, therefore, the estimation of the variables plays an important role in the control process [1], [2]. In the recent years many nonlinear observers have been designed; we can mention here the high-gain observers, sliding mode observers [3], [4], [5], and so on. These are complex observers and they can be used when the system structure is known. That is why the neural networks may be successfully used in the control of the dynamic systems [6], [7], [8]. The good results of the neural networks are due to their capacity of nonlinear functions' approximation [9], [10]. Most systems are nonlinear and it's difficult to design a controller or observer; so far many linearization techniques have been used to overcome these drawbacks. Linearization generally limits the controller performances and the observer design. In such cases, one uses the neural networks to approximate the nonlinear functions [11].

Several schemes based on Lyapunov theory have been proposed in the literature. For example, in [12] a neural network controller is derived using a filtered error passivity approach leading to new NN passivity properties. By using the Lyapunov theory and a tuning algorithm, including a correction term to backpropagation, the stability of the neural network controller is guaranteed [11]. Some specific bounds are determined in [13] and the tracking error bound can be made arbitrarily small by increasing a certain feedback gain [11]. The correction terms involve a second-order forward-propagated wave in the back-propagation network. In order to tackle more general nonlinear systems, [14] extend the previous result and two linearly parameterized neural networks are used to capture the unknown dynamic of the system, the weights of the NN being adjusted by means of adaptation laws in order to obtain the stability of the overall scheme [11]. In the next sections of the paper the neural network observer for nonlinear systems, designed in [11], is used to estimate the state of an aircrafts with big attack angle. The neural network observer (neuro-observer) consists, as its main part, of a feedforward neural network with three layers of neurons; the NN is trained with the error backpropagation learning algorithm by using a correction term which guarantees the good training of the neural network. The stability of the neural network observer may be studied by means of the direct Lyapunov method.

The paper is organized as follows: the architecture of the artificial neural network, necessary to the observer design, is given in section II; the design of the neural network observer is presented in section III; in section IV we validate this observer by means of complex numerical simulations for the case of an aircraft with big attack angle. Finally, some conclusions are shared in section V.

II. Arhitecture of the Artificial Neural Network

The neural network of the observer is chosen as feedforward NN with three layers of neurons (a single hidden layer NN). Each input neuron is connected, via a weight, with all the neurons from the hidden layer, while each hidden layer neuron is connected with the output neurons by means by a weight. Thus, the output of the neuron j from the hidden layer is [11]:

$$a_j = \sigma(z_j) \tag{1}$$

with

$$z_j = \sum_i v_{ji} x_i + \mu_j ; \tag{2}$$

v_{ji} is the weight between the input neuron i and the neuron j in the hidden layer, μ_j – the bias for the neuron j from the hidden layer, while $\sigma(\cdot)$ is the activation function of the hidden layer neurons (sigmoid function)

$$\sigma(z_j) = \frac{1}{1 + e^{-z_j}} ; \tag{3}$$

The output of the neuron k from the output layer is:

$$y_k = \sum_j w_{kj} a_j ; \quad (4)$$

w_{kj} is the weight between the neuron j in the hidden layer and the output neuron k. If the neural network has n input neurons (pseudo-neurons), s neurons in the hidden layer, and m output neurons, the equation associated to the neural network may be written under a matriceal form considering the vector of the inputs $x^T = [x_1 \ x_2 \ \cdots \ x_n]$, the vector of the outputs $y^T = [y_1 \ y_2 \ \cdots \ y_m]$, and the matrices of the weights and biases $W^T = [w_{kj}], V = [v_{ji}], \mu = [\mu_1 \ \mu_2 \ \cdots \ \mu_s]$. The vector of biases μ may be included, as the first column, into the matrix V. Thus, $W \in R^{s \times m}, V \in R^{s \times n}$ and the output y of the NN has the form [11]:

$$y = W^T \sigma(V^T x). \quad (5)$$

For the training of the neural network there are some algorithms in the specialty literature; the most used algorithm is the error backpropagation learning algorithm [15]. According to this algorithm, the neural network training process becomes a nonlinear function optimization problem. The weights and the biases are permanently modified in order to minimize the mean squared error between the actual outputs and the NN desired outputs in a gradient descent manner [7].

III. The Design of the Observer

In this section we present the design of the neuro-observer for the state estimation of the nonlinear systems; the observer has a neural network which is used for the estimation of the unknown nonlinear function which interferes in the system dynamics. We consider the system [11], [16]:

$$\begin{cases} \dot{x}(t) = Ax(t) + g(x,u), \\ y(t) = Cx(t); \end{cases} \quad (6)$$

$x(t) \in R^n$ is the state vector, $u(t) \in R^{m_u}$ – the vector of inputs, $y(t) \in R^{m_y}$ – the vector of outputs, $g(x,u)$ – unknown nonlinear function. The matrix A is Hurwitz, while the pair (A, C) must be observable in order to design the observer.

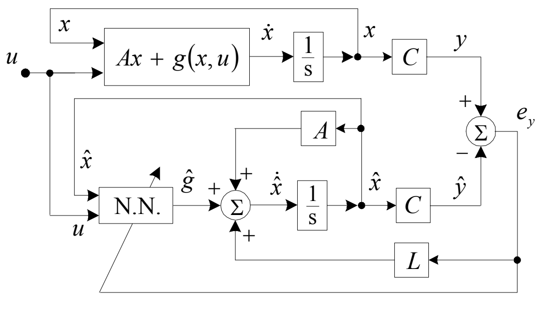

Fig. 1. Block diagram of the ensemble system – neuro-observer

We design the observer described by equations [11]:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + \hat{g}(\hat{x},u) + L(y - \hat{y}), \\ \hat{y}(t) = C\hat{x}(t), \end{cases} \quad (7)$$

where $\hat{x}$ is the observer state and $\hat{y}$ – the output of the observer for the nonlinear system. The gain matrix L of the state observer is chosen so that $(A - LC)$ is a Hurwitz matrix [11]. The block diagram of the ensemble system – neural network observer is presented in fig.1.

As one can see in fig.1 the neural network is used for the estimation of the unknown nonlinear function g. The main use of the NNs is the approximation of the functions [17]. Thus, considering the function $\gamma(x) : R^n \to R^m$, in [18] one has shown that, for a sufficient number of neurons in the hidden layer (s), there exist the weights and the biases so that any function, continuous on a compact set, may be expressed as following [1]:

$$\gamma(x) = W\sigma(Vx) + \varepsilon(x), \quad (8)$$

where $\varepsilon(x)$ is the approximation error of the nonlinear function; $\|\varepsilon(x)\| \le \varepsilon_N$. Moreover, for $\varepsilon_N > 0$, one can find a neural network so that $\|\varepsilon(x)\| \le \varepsilon_N \ (\forall) x$ [12], [18]. We suppose that the weights W and V are bounded by known values so that $\|W\| \le W_M$ and $\|V\| \le V_M$ [12], [13]. Thus, the function $g(x,u)$ from the first equation (6) may be approximated by a neural network having the weights W and V [11]:

$$g(x,u) = W\sigma(Vz) + \varepsilon(x), \quad (9)$$

where $z = [x \ u]$ and $\varepsilon(x) = g(x,u) - \hat{g}(\hat{x},u)$. The estimation $\hat{g}(\hat{x},u)$ of the function $g(x,u)$ is written:

$$\hat{g}(\hat{x},u) = \hat{W}\sigma(\hat{V}\hat{z}), \quad (10)$$

with $\hat{z} = [\hat{x} \ u]$. Thus, the equations (7) of the neural network observer become:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + \hat{W}\sigma(\hat{V}\hat{z}) + L(y - \hat{y}), \\ \hat{y}(t) = C\hat{x}(t). \end{cases} \quad (11)$$

We define the errors:

$$e_x = x - \hat{x}, e_y = y - \hat{y}, e_W = W - \hat{W} \quad (12)$$

and, by using equations (6), (9), and (11), it results [11]:

$$\dot{e}_x = Ge_x + e_W \sigma(\hat{V}\hat{z}) + \xi(t), \quad (13)$$

with

$$\begin{aligned} G &= A - LC, \\ \xi(t) &= W[\sigma(Vz) - \sigma(\hat{V}\hat{z})] + \varepsilon(x); \end{aligned} \quad (14)$$

$\xi(t)$ is a bounded disturbance term; $\|\xi(t)\| \le \bar{\xi}$ = constant because the sigmoid function and the weights W and V are bounded [14]. Accordingly, we obtain:

$$\begin{cases} \dot{e}_x(t) = Ge_x(t) + e_W \sigma(\hat{V}\hat{z}) + \xi(t), \\ \dot{e}_y(t) = Ce_x(t). \end{cases} \quad (15)$$

For the training of the NN we need a learning rule which guarantees the stability of the observer. Moreover, the weights update methodology, by using the standard Lyapunov technique, is based on the error back-propagation learning algorithm where some new terms have been added for the stability of the observer and of the errors associated to the NN weights [11]. The weights update is presented in [4], some modifications regarding the NN weights' differential equations being made in [11]. The correction terms have been borrowed from [18]. As a consequence, in [11], the following differential equations for the calculation of $\hat{W}$ and $\hat{V}$ have been obtained [11]:

$$\begin{cases} \dot{\hat{W}} = Se_x\sigma^T\left(\hat{V}\hat{z}\right) - k\|e_x\|S\hat{W}, \\ \dot{\hat{V}} = \left[\hat{W}\sigma\left(\hat{V}\hat{z}\right)\left(1-\sigma\left(\hat{V}\hat{z}\right)\right)\right]^T Fe_x\hat{z}^T - k\|e_x\|F\hat{V}, \end{cases} \quad (16)$$

where the symmetric and positive definite matrices S and F $\left(S = S^T > 0, F = F^T > 0\right)$ have the expressions:

$$\begin{aligned} S &= -\eta_1 G^{-T}C^TC, \\ F &= -\eta_2 G^{-T}C^TC; \end{aligned} \quad (17)$$

$\eta_1, \eta_2 > 0$ are the learning rates of the neural network, while k is a small positive constant.

The use of the matrices S and F in equations (16) allows the simplification of the observer stability study. Lyapunov's direct method is used in order to ensuring the stability of the proposed non-conventional observer and of the NN weight errors. This study is widely presented in [11] and we will not insist on it.

IV. The Use of the Observer to the Longitudinal Motion of the Aircraft with Big Attack Angle

The validation of the above presented neuro-observer, for the state estimation of a nonlinear system, is made in this section for the case of longitudinal motion of an aircraft with big attack angle. The longitudinal motion of the aircrafts with big attack angle is a nonlinear and complex system, which makes difficult the design and implementation of efficient control and observation laws. Thus, the dynamics of the longitudinal motion, generally unstable, of such an aircraft is described in [19] and [20] by the state equation:

$$\dot{x} = Af\left(x\right) + Bu, \quad (18)$$

where

$$\begin{aligned} & f^T\left(x\right) = \left[\tilde{x}_1^T \;\; \tilde{x}_2^T\right]^T, \\ & \tilde{x}_1 = \left[x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5\right]^T, \\ & \tilde{x}_2 = \left[x_2x_5 \;\; x_1^{1/3} \;\; x_1^3 \;\; x_2^{1/3} \;\; x_2^2 \;\; x_2^3 \;\; x_3^{1/3} \;\; x_3^3 \;\; x_4^3 \;\; x_5^3\right]^T, \\ & x_1 = \Delta\text{V}, x_2 = \Delta\alpha, x_3 = \Delta q, x_4 = \Delta c_p, x_5 = \Delta C_m, \end{aligned} \quad (19)$$

V is the aircraft velocity, $\alpha -$ the aircraft attack angle, $q -$ the aircraft pitch angular rate, $c_p -$ the lift coefficient, $C_m -$ pitch moment coefficient, the symbol Δ is associated with the perturbation of the variables from their nominal values, and $u = \delta_e$ is the elevator deflection, while the matrices A and B have the forms:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 0 & a_{3,5} & a_{3,6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & 0 & 0 & a_{4,7} & a_{4,8} & a_{4,9} & a_{4,10} & a_{4,11} & a_{4,12} & a_{4,13} & a_{4,14} & 0 \\ a_{5,1} & a_{5,2} & a_{5,3} & 0 & a_{5,5} & 0 & a_{5,7} & a_{5,8} & a_{5,9} & a_{5,10} & a_{5,11} & a_{5,12} & a_{5,13} & 0 & a_{5,15} \end{bmatrix},$$

$$B^T = \left[b_{11} \;\; b_{12} \;\; b_{13} \;\; b_{14} \;\; b_{15}\right].$$

The above dynamic model is equivalent with the following one:

$$\begin{aligned} \dot{x} &= Ax + g(x,u), \\ u &= \delta_e, x = \left[\Delta\text{V} \;\; \Delta\alpha \;\; \Delta q \;\; \Delta c_p \;\; \Delta C_m\right]^T, \end{aligned} \quad (20)$$

with

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 \\ a_{31} & a_{32} & a_{33} & 0 & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 \\ a_{51} & a_{52} & a_{53} & 0 & a_{55} \end{bmatrix}, g(x,u) = \begin{bmatrix} b_{11}u \\ b_{12}u \\ b_{13}u + g_3(x) \\ b_{14}u + g_4(x) \\ b_{15}u + g_5(x) \end{bmatrix}; \quad (21)$$

$$\begin{aligned} g_3(x) &= a_{36}x_2x_5, \\ g_4(x) &= a_{47}x_1^{1/3} + a_{48}x_1^3 + a_{49}x_2^{1/3} + a_{4,10}x_2^2 + a_{4,11}x_2^3 + \\ &\quad + a_{4,12}x_3^{1/3} + a_{4,13}x_3^3 + a_{4,14}x_4^3, \\ g_5(x) &= a_{57}x_1^{1/3} + a_{58}x_1^3 + a_{59}x_2^{1/3} + a_{5,10}x_2^2 + a_{5,11}x_2^3 + \\ &\quad + a_{5,12}x_3^{1/3} + a_{5,13}x_3^3 + a_{5,15}x_5^3\,. \end{aligned} \quad (22)$$

The validation of the algorithm is performed in Matlab/Simulink environment. Thus, we elaborated the Simulink model in fig.2. Here two subsystems may be observed: *Block forming g* (the subsystem for the obtaining of the vector g by using the state vector x and the vector of the inputs u - equation (21)) – fig.3 and *Neural network* (the subsystem used for the approximation of the nonlinear function g; this subsystem has three inputs: $\hat{x}$ (xc), u (u) and e_x (e_x) and only one output $\hat{g}$ (gc)) – fig.4. The subsystem *Neural network* has a subsystem, too – *NN Weights Update* (fig.5). It calculates the weights $\hat{W}$ and $\hat{V}$ (Wc and Vc) by using, according to equations (16), the signals $\hat{z}$ (zc), e_x (e_x), and the matrices S, F. The neural network has the following structure: feedforward NN with 3 layers (the input layer has 6 neurons, the hidden layer has 5 neurons, while the output layer has only one neuron). The activation function, used for the neuron in the output layer, is the sigmoid function (equation (3)). The weights $\hat{W}$ and $\hat{V}$ are iteratively updated; at the end of the training process, we obtain:

$$\hat{W}^T = \left[1.162 \;\; 1.173 \;\; -0.035 \;\; 0.325 \;\; 0.173\right],$$

$$\hat{V} = \begin{bmatrix} -0.179 & 0.11 & 0.284 & 0.836 & -0.387 & 0.651 \\ 0.716 & 1.052 & -1.318 & 1.237 & 0.68 & 1.175 \\ -0.581 & 0.058 & 0.707 & -1.579 & 0.809 & -1.191 \\ 2.168 & -0.095 & 1.612 & -1.431 & 0.707 & -0.019 \\ -0.135 & -0.827 & -0.688 & 0.568 & 1.283 & -0.155 \end{bmatrix}; \quad (23)$$

the first column of the matrix $\hat{V}$ is the vector of the biases for the neurons in the hidden layer. The neural network of the observer estimates the nonlinear function $g(x,u)$, and, after that, the neuro-observer estimates the state vector x.

In fig.7 we present the time evolution of the 5 state variables $\left(x_i, i = \overline{1,5}\right)$ and of the 5 estimated state variables $\left(\hat{x}_i, i = \overline{1,5}\right)$. Time evolutions of the 5 components of the error $\left(e_x = x - \hat{x}\right)$ are presented in fig.6. All the 5 components $e_{x_i}, i = \overline{1,5}$, tend to zero very quickly, this fact proving both the proper functioning of the neural network (estimation of the nonlinear function) and the very good estimation of the state variables (figs.6 and 7).

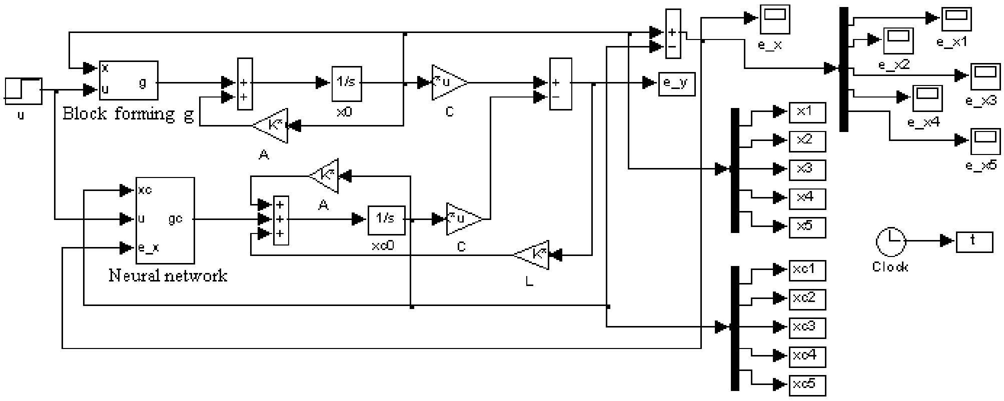

Fig. 2. Matlab/Simulink model of the ensemble system-neural network observer

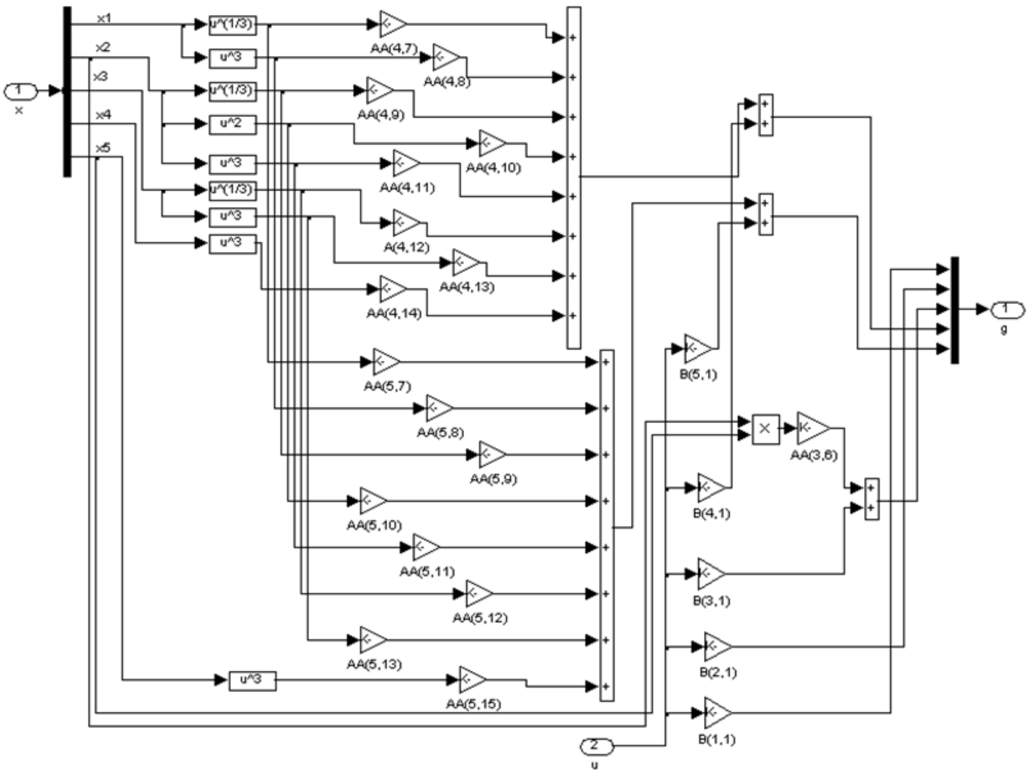

Fig. 3. Matlab/Simulink model of the subsystem "Block forming g"

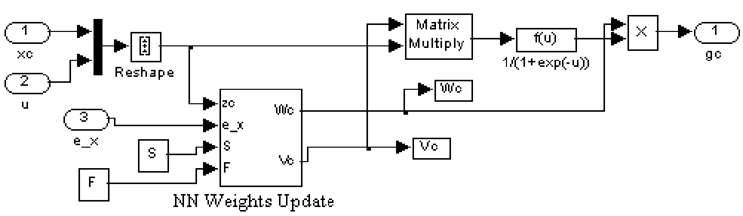

Fig. 4. Matlab/Simulink model of the subsystem "Neural network"

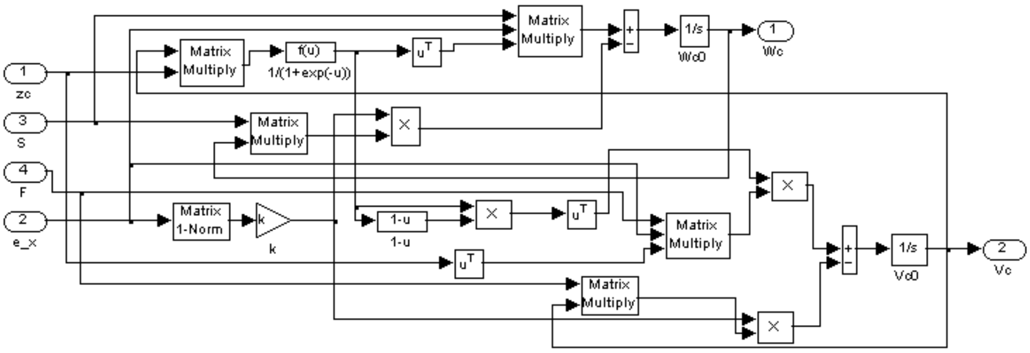

Fig 5. Matlab/Simulink model of the subsystem “NN Weights Update”

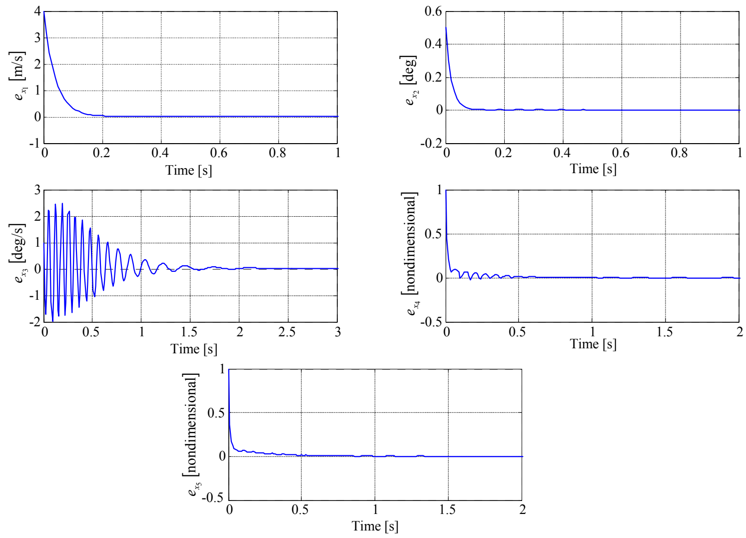

Fig. 6. The five components of the estimation error

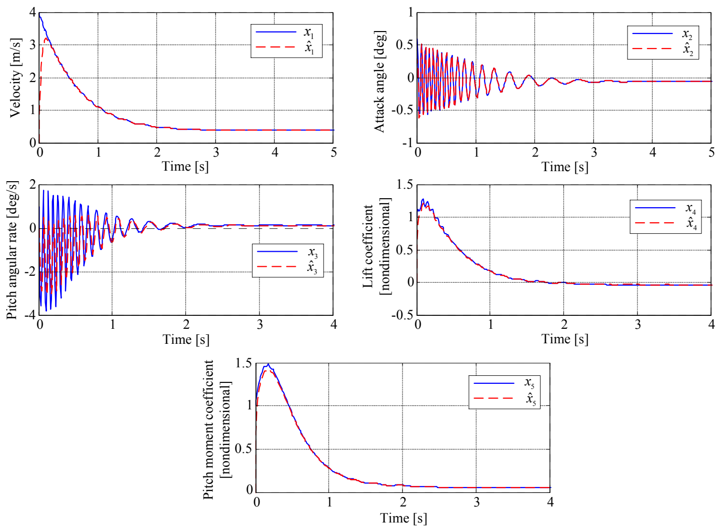

Fig. 7. The state variables (x_i) and the estimated state variables $(\hat{x}_i)$

V. CONCLUSION

This paper deals with the study of a neural network observer for nonlinear systems. The proposed neuro-observer is a three-layer feedforward neural network, trained by means of the error backpropagation learning algorithm. The neural network, adopting a sigmoid activation function, is used to parameterize the nonlinearities of the system. The neural networks may be successfully used in the control of the dynamic systems. The good results of the neural networks are due to their capacity of nonlinear functions’ approximation. Most systems are nonlinear and it is difficult to design a controller or an observer; so far the linearization techniques have been used to overcome these drawbacks. The linearization technique limits the performances of the controller and the observer design. In such cases, one uses the neural networks to approximate the nonlinear functions.

The validation of the proposed state observer scheme is made through Matlab/Simulink numerical simulation to reconstruct the unavailable state variables of a big attack aircraft longitudinal motion. All the components of the error vector tend to zero very quickly, this fact proving both the proper functioning of the neural network and the very good estimation of the state variables.

ACKNOWLEDGMENT

This work was supported by the strategic grant POSDRU/89/1.5/S/61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007 – 2013.

REFERENCES

[1] J. Theocharis and V. Petridis, "Neural network observer for induction motor control," *IEEE Control Systems*, 1994.

[2] A. Germani, C. Manes, and P. Pepe, "A new approach to state observation of nonlinear systems with delayed output," *IEEE Transactions on Automatic Control*, Vol. 47, 2002.

[3] F. Abdollahi and H. A. Talebi, "A stable neural network observer with application to flexible joint manipulators," 9th International Confe-rence on Neural Information Processing, pp. 1910-1914, 2002.

[4] F. Abdollahi and H. A. Talebi, "A stable neural network observe-based observer with application to flexible joint manipulators," *IEEE Transaction on Neural Network*, Vol 1, pp. 118-129, 2006.

[5] K. Jain, "Artificial neural networks, a tutorial," *Computer - Special issue: neural computing*, pp. 31-44, 1996.

[6] K. Narendra and K. Parthasarathy, "Identification and control of dynamic system using neural networks," *IEEE Transaction on Neural Network*, Vol. 1, pp. 4-27, 1990.

[7] T. Michael and R. Harley, "Identification and control of induction machines using artificial neural networks," *IEEE Transaction on Industry Applications*, Vol. 31, pp. 612-619, 1995.

[8] K. Seong-Hwan, P. Tae-Sik, Y. Ji Yoon, and P. Gwi-Tae, "Speed sensorless vector control of an introduction motor using neural network speed estimation," *IEEE Transaction on Industrial Electronics*, Vol. 48, pp. 609-614, 2001.

[9] J. Olivera, J., X. Guan, "Theory of monomial networks," *Proceedings of Implicit and Nonlinear Systems*, pp. 96-101, 1992.

[10] J. Park and I. Sandberg, "Universal approximation using radial-basis function networks," *Neural Computers*, vol.3, pp. 246-257, 1991.

[11] A. N. Lakhal, A. S. Tlili, and B. Benhadj, "Neural Network Observer for Nonlinear Systems Application to Induction Motors," *International Journal of control and Automation*, vol. 3, no. 1, 2010.

[12] F. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Transaction on Neural Network*, Vol. 6, pp. 703-715, 1995.

[13] Y. H. Kim, F. L. Lewis, and C. T. Abdallah, "Nonlinear observer design using dynamic recurrent neural networks," 35th Conference on Decision and Control, Kobe, Japan, pp. 949 - 954, 1996.

[14] J. Vargas and E. Hemerly, "Neural adaptive observer for general nonlinear systems," Proceedings of the American Control Conference, Chicago, 2000.

[15] M. Godoy and K. Bose, "Neural network based estimation of feedback signals for a vector controlled induction motor drive," *IEEE Transaction on Industry applications*, Vol 31, pp. 620-629, 1995.

[16] T.L. Grigorie, L. Dinca, and J. Corcau, "The Density Altitude. Influence Factors and Evaluation," 6th WSEAS International Conference on Dynamical Systems and Control, Kantaoui, Sousse, Tunisia, May 3-6, 2010.

[17] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[18] R. Selmic and R. Lewis, "Multimodel neural networks identification and failure detection of nonlinear systems," Proceeding of the 40th IEEE Conference on Decision and Control, Florida, USA, pp. 3128-3133, 2001.

[19] B. Etkin and L. D. Reid, "Vehicle Stability and Pitch," 5th Conference on Mathematics and Computers, Australia, June 14 – 16, 2000.

[20] S. E. Lyshevski, "Nonlinear Identification of Control Systems," Proceedings of the American Control Conference, Philadelphia, PA, pp. 2366 – 2370, 1998.