New Systems for Identification, Estimation and Adaptive Control of the Aircrafts Movement

Mihai Lungu[1], **Romulus Lungu**[1], **Constantin Rotaru**[2]

[1] Avionics Department, Faculty of Electrical Engineering,
University of Craiova,
107, Decebal Blvd., Craiova, Romania,
Lma1312@yahoo.com, mlungu@elth.ucv.ro

[2] Department of Aviation Integrated Systems, Military Technical Academy,
81-83, George Cosbuc Blvd., Bucharest, Romania,
crotaru@mta.ro

Abstract: This paper presents two new systems for identification and neuro-adaptive command with direct applicability to the control of the longitudinal and lateral aircrafts movement and to the rockets' vertical and horizontal movement. Also, a structure for the parametric estimation and discrete optimal command of the aircrafts' movement is presented. The design of these structures is based on the algorithms that belong to the authors of this paper. Theoretical results are validated by numerical simulations; the authors obtained Matlab/Simulink models and calculus programs for the identification, neuro-adaptive command, on-line estimation and discrete command of the aircrafts movement, respectively.

Keywords: adaptive, aircraft, neural network, identification, estimation.

1. Introduction

The flight to high attack angles (specific for the fight regimes) is affected by the effects of aerodynamic instability. The complexity and uncertainty in modeling of such non-linear phenomena are the main arguments for designing of evolved adaptive control structures, in these conditions the linear models being too far to describe correctly the dynamics of aircrafts. There is a high degree of uncertainty regarding the flight parameters and flying objects dynamics at high attack angles. Another argument is that the actuators are non-linear and present essential non-linear elements such as the saturation of displacement or the saturation of the mobile elements speed. The stability and handling qualities must also be maintained in conditions of sensors and actuators failures.

Control systems (denoted in this paper with A) must have the capability to identify defects, to isolate the damaged elements and to reconfigure the architecture in real time. The failures belong to sensors, actuators or force equipments. Thus, a real time redesign of the control systems is needed. The observers must be easily adaptable; that means that their design algorithms must allow aircrafts state estimation with or without the signals provided by the damaged sensors. In all these situations real-time adaptive control based on neural networks is adequate [1], [2], [3], [4], [5]. The training process of the neural networks is done by using signals provided by the state observers; these state observers have, as input, the tracking error vector.

The area of adaptive control has grown to be one of the richest in terms of algorithms, design techniques, analytical tools, and modifications. Despite the rich literature, the field of adaptive control may easily appear to an outsider as collection of unrelated tricks and modifications. The adaptive flight control systems, which are presented in this paper, were designed to provide invariant aircraft response characteristics throughout aircraft's atmospheric flight envelope.

The contributions of this paper are: an adaptive control system with neural network and adaptive controller, a complex system for the neuro-adaptive control of the aircraft movements and a system for the parametric estimation and discrete optimal command of aircraft longitudinal and lateral movements. So, the purpose of this paper is the design and software implementation in Matlab/Simulink environment of the three systems. By comparison with other adaptive algorithms from the specific literature, the authors' adaptive flight control systems are characterized by simplicity, proper function and dynamic stability. The study of the algorithms convergence and stability has been made by the paper authors by complex numerical simulations. The stability of the presented systems is obvious from the analysis of the system responses.

The paper is organized as follows: the design of the two new systems for aircrafts identification and neuro-adaptive command are given in section 2; the design of the system for the parametric estimation and discrete optimal command of aircraft longitudinal and lateral movements is presented in section 3; three numerical examples are included in section 4; finally, some conclusions are given in section 5.

2. Systems for the Aircrafts Identification and Neuro-Adaptive Command

The contribution command system in Figure 1 consists of a neural regulator, a conventional regulator and an identification neural network. It is used, with good results, to the stabilization of the aircraft longitudinal and lateral movements. This system works in two regimes: identification regime (I) and automatic control (R). In the identification stage (the switch has position I) the system is an open loop one (the signal is simultaneous applied to the control system and to the neural network and it is given by a signals generator block).

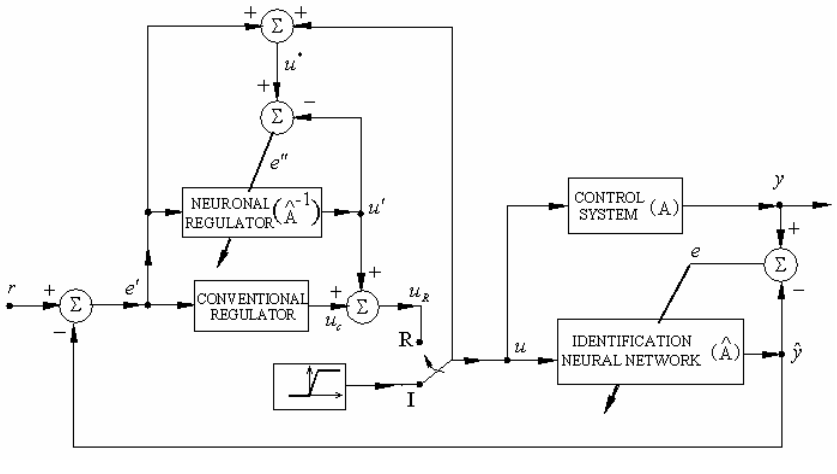

Figure 1. The adaptive control system with neural network and adaptive controller

After the identification process, the switch passes on the position R (automatic command); the input of the system is:

$$u = u_R = u_c + u', \tag{1}$$

where u_c is the command signal given by a conventional regulator and u' – the output of the neural regulator.

The neural regulator must model (form) the inverse of the neural network function; the neural regulator model must be $\hat{A}^{-1}$ is the neural network model for the estimation of the control system model (A). As input, for the neural regulator one chooses, for example, the error of the adaptive control system; $e' = r - \hat{y}$, r is the reference of the system, while $\hat{y}$ is the output of the neural network.

The neural regulator is modeled by a feed-forward network with the error $e'' = u^* - u'$; u^* is the imposed output of the neural regulator, while u' is the current output of the neural regulator. For the calculus of u^*, we start from the idea that, in stationary regime, the error e'' tends to zero in the same time with the adaptive system error e'.

Imposing $e' \cong e'' = u^* - u'$ and $u \to u'$ $(u_c \to 0)$, we get:

$$u^* \cong e' + u' \cong e' + u. \tag{2}$$

The training of the neural regulator stops when $e'' \to 0(e' \to 0)$; in that moment we remark that $u \to u^*$ $(u_c \to 0)$. Thus, on the direct way of the adaptive control system, we obtain the approximate inverse of the function from the feedback way, i.e. $\hat{A}^{-1}\hat{A} \cong 1$. When the switch passes on the position R (automatic control), the initial value of u is considered to be the input of the identification neural network (this signal is given by the signals generator).

Another contribution of this paper is related to the hierarchical neuro-adaptive command system in Figure 2; it may also be used to the stabilization of the aircrafts movement. Its design is presented below.

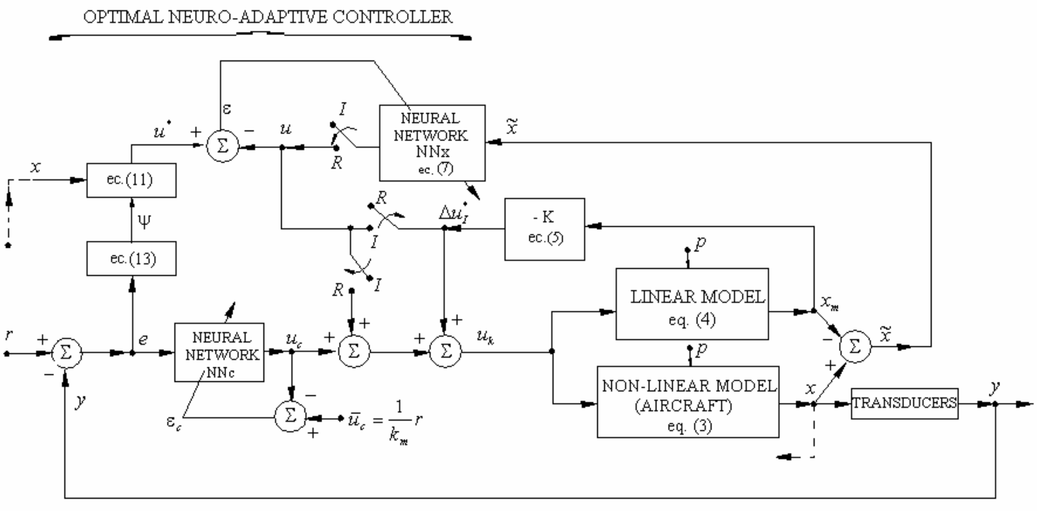

Figure 2. Complex system for the neuro-adaptive control of the aircrafts' movements

First of all, let us consider that the non-linear dynamic model of the aircraft is described by equation [6]:

$$\begin{aligned} \dot{x} &= f(x(t), u(t), p(t), t), \\ t &\in [t_0, t_1], x(t_0) = x_0, \end{aligned} \quad (3)$$

where x is the state vector $(n \times 1)$, u – the command vector $(m \times 1)$ and p – the vector of the disturbances with dimension $(n \times 1)$.

In the **initialization stage** of the neural network (NNx) one may use the linear control theory for the command of the non-linear control system; the linear system is obtained by the linearization of the non-linear system (3). Thus, assuming that all deviations are small, system (3) may be approximated with the linear one:

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + B\Delta u + D\Delta p, \\ \Delta x(t_0) &= \Delta x_0, \end{aligned} \quad (4)$$

where Δp (the vector of the disturbances) produces the deviation of the dynamic trajectory from the undisturbed reference one.

The matrices A, B and D are calculated with respect to the undisturbed reference trajectory.

We need to obtain an optimal command (Δu^*)which assures the convergence of the linear model trajectory to the reference trajectory. The expression of this optimal command is:

$$\Delta u^* = -K\Delta x = -R^{-1}\left[B^{\mathrm{T}}P + N\right]\Delta x, \quad (5)$$

where $R(m \times m)$ is a non-singular matrix, $N(n \times m)$– a symmetric matrix, K– the process gain matrix and P– the solution of the Riccati equation:

$$A^{\mathrm{T}}P + PA - PBR^{-1}B^{\mathrm{T}}P + Q = 0. \quad (6)$$

In the **training stage**, the neural network (with four input neurons, five neurons in the only one hidden layer and only one output neuron) computes the command law with respect to the state vector $\tilde{x} = x - x_m$ (Figure 2); the equation of the neural network is:

$$u = V^{\mathrm{T}}g[Wx + d] + b, \quad (7)$$

where W is the input weights vector, V – the vector of the output weights, d and b – the input and output biases and g – the vector of the sigmoid functions.

For the determination of the optimal command u^* we use the Hamiltonian equation [7], [8]:

$$\frac{\mathrm{d}H(x, u, \psi, t)}{\mathrm{d}u} = 0, \quad (8)$$

where ψ is the adjunct vector described by the equation $\dot{\psi} = -\frac{\partial H(x(t), u(t), \psi(t), t)}{\partial x}$ and H is the Hamilton's function:

$$H\left(x(t), u(t), \frac{\partial I(x,t)}{\partial x}, t\right) = -L(x(t), u(t), t) + \underbrace{\left[\frac{\partial I(x,t)}{\partial x}\right]^{\mathrm{T}}}_{\psi^{\mathrm{T}}} f(x(t), u(t), t);$$

the functional

$$J(x(t), u(t), t) = \int_{t}^{t_1} L(x(\tau), u(\tau), t)\,\mathrm{d}\tau$$

attaches a well determined number

$I(x(t), t) = -J(x(t), u(t), t)$

to the trajectory corresponding to an interval $[t, t_1)$. In the above equations the function $L(x(t), u(t), t)$ has the form:

$$L(x(t), u(t), t) = \left[\frac{\partial I(x,t)}{\partial x}\right]^{\mathrm{T}} \frac{\partial I(x(t),t)}{\partial t} + f(x(t), u(t), t).$$

Because $H(x, u, \psi, t) = H(x(u), u, \psi, t)$, the equation (8) becomes:

$$\frac{\partial H}{\partial u} + \frac{\partial H}{\partial x}\frac{\partial x}{\partial u} = 0 \qquad (9)$$

or, taking into account the equation of the adjunct vector ψ, it results:

$$\frac{\partial H}{\partial u} - \dot{\psi}\frac{\partial x}{\partial u} = 0. \qquad (10)$$

By the discretization of equation (10), we get:

$$\frac{\partial H[x(k), u(k)]}{\partial u(k)} - \psi(k+1)\frac{\partial x(k+1)}{\partial u(k)} = 0. \qquad (11)$$

The equation of the adjunct vector ψ becomes:

$$\dot{\psi}(t) = -G^{\mathrm{T}}\psi(t) + Q_y \Delta p(t), \qquad (12)$$

which assumes the knowledge of the external disturbance Δp; this fact is not possible in most of the cases and, that is why, in equation (12), the term $Q_y\Delta p(t)$ may be substituted by other term which expresses, more accurately, the effect of the disturbance. The disturbance affects the output vector of the nonlinear system (y) and the vector y_m associated to the linear model. Thus, the disturbance influences the system error $e' = r - y$; r is the reference (imposed) vector. So, the previous equation may be written:

$$\dot{\psi}(t) = -G^{\mathrm{T}}\psi(t) + k(r - y), \qquad (13)$$

where the matrix G has the form $G = A - BR^{-1}[B^{\mathrm{T}}P + N]$ and k is a variable gain coefficient; without losing the generality, it may be chosen 1. If k has other values $(k \neq 1)$, a supplementary disturbance appears and it is compensated by the feedback loop (feedback after the output vector y). Thus, the vector ψ may be calculated by the integration of equation (13), with $k = 1$. The neural network **NNc** models an equation of form (7) and **NNc** is trained through the minimization of the error:

$$\varepsilon_c = \left\|\bar{u}_c - u_c\right\|^2, \qquad (14)$$

where $\bar{u}_c$ is obtained by using the formula:

$$\bar{u}_c = \frac{1}{k_m} r; \qquad (15)$$

k_m is the gain coefficient of the subsystem on direct way. When $e \to 0\,(y \to r, u_c \to \bar{u}_c$, $x \to x_s = \text{ct.}), \psi \to 0, u^* \to u_s^* = \text{ct.} = h(x_s)$, $u \to u_s^* = \text{ct.}$ and $\varepsilon \to 0, u(k) \to u_s(k) = \text{ct.}$; this is equivalent with $x_m \to x_{ms} = \text{ct.} \to x_s$ and $\tilde{x} \to 0$. Thus, one chooses a linear model as close as possible from the non-linear model of the control system, whose state verifies, in an unperturbed regime, the relationship $x_m \to x$.

In the initialization stage, the model is brought in the state $x_m\,(\Delta x_m = 0)$ by means of the optimal control law $\Delta U_I^* = -K\Delta x_m$. In the training stage of the neural network **NNx**,

$$u_k = u_c + \Delta U_I^* + u, \qquad (16)$$

where u_c is the output of the network **NNc** (neural regulator); the input of the network is the deviation of the system output from the reference value r. The imposed value is $\bar{u}_c = r / k_m$, because, at equilibrium, $y = k_m u_c \to k_m \bar{u}_c \Leftrightarrow \bar{u}_c = y / k_m \to r / k_m$; the adaptive component u of the command law is the output of network **NNx**, whose input is the difference $\tilde{x} = x - x_m$ because of the disturbance p and of the deviation of the model A from the reference one. The imposed value u^* of the **NNx**'s output is calculated with respect to the adjunct vector ψ (the solution of equation (13)); it depends on the error $e = r - y$ generated by the disturbance p.

3. Parametric Estimation and Discrete Optimal Command of the Aircrafts Movement

The parametric estimation and discrete optimal command algorithm (ALGLDR) represents an important contribution of this paper authors.

Let us consider the discrete dynamic model of the aircraft (A) which is described by the equations:

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k), \\ y(k) &= C_d x(k) \end{aligned} \qquad (17)$$

and the estimated discrete dynamic model $(\hat{\mathrm{A}})$, described by the relationships:

$$\begin{aligned} \hat{x}(k+1) &= \hat{A}_d \hat{x}(k) + \hat{B}_d u(k), \\ \hat{y}(k) &= \hat{C}_d \hat{x}(k); \end{aligned} \qquad (18)$$

A_d, B_d, C_d are the discrete variants of the matrices A, B, C from general state equations [9] and $\hat{A}_d, \hat{B}_d, \hat{C}_d$ are the estimations of these matrices.

Another way to describe the dynamics of the estimated control system $(\hat{\mathrm{A}})$ is [10]:

$$\hat{y}(k+1) = z^T(k+1)\hat{b}(k) + e(k+1), \qquad (19)$$

where

$$\begin{aligned} e(k+1) &= y(k+1) - \hat{y}(k+1); \\ \hat{b}^T(k) &= \left[\hat{\alpha}^T(k)\ \hat{b}_1(k)\ \hat{\beta}^T(k)\right], \end{aligned} \qquad (20)$$

with

$$\begin{aligned} \hat{\alpha}^T(k) &= \left[-\hat{a}_1(k)\ -\hat{a}_2(k)\ \dots\ -\hat{a}_n(k)\right], \\ \hat{\beta}^T(k) &= \left[\hat{b}_2(k)\ \hat{b}_3(k)\ \dots\ \hat{b}_m(k)\right], \end{aligned} \qquad (21)$$

$$\hat{\alpha}(np \times p), \hat{b}_1(p \times m), \hat{\beta}[(m-1)p \times m];$$

$$z^T(k+1) = \left[\hat{Y}^T(k)\ u(k)\ U^T(k)\right], \qquad (22)$$

with

$$\begin{aligned} \hat{Y}^T(k) &= \left[\hat{y}(k)\ \hat{y}(k-1)\ \dots\ \hat{y}(k-n+1)\right], \\ U^T(k) &= \left[u(k-1)\ u(k-2)\ \dots\ u(k-m+1)\right]; \end{aligned} \qquad (23)$$

$\hat{Y}(np \times 1), U[(m-1)m \times 1]$. If $m = p$, the equation (19) becomes:

$$\begin{aligned} \hat{y}(k+1) = {} & \hat{\alpha}^T(k)\hat{Y}(k) + \hat{b}_1(k)u(k) + \\ & + \hat{\beta}^T(k)U(k); \end{aligned} \qquad (24)$$

if $m \neq p$, then $\hat{\beta}^T(k)$ cannot be multiplied with the vector $U(k)$ because of their dimensions. That is why, in equation (24), the last term is expressed for each concrete case (function of the values of m and p). Thus, in the case of longitudinal movement of the aircrafts $(n = 4, m = 1)$, the equation (24) gets the form:

$$\hat{y}(k+1) = \hat{\alpha}^T(k)\hat{Y}(k) + \hat{b}_1(k)u(k), \qquad (25)$$

where

$$\begin{aligned} \hat{Y}^T(k) &= \left[\hat{y}(k)\ \hat{y}(k-1)\ \hat{y}(k-2)\ \hat{y}(k-3)\right], \\ \hat{\alpha}^T(k) &= \left[-\hat{a}_1(k)\ -\hat{a}_2(k)\ -\hat{a}_3(k)\ -\hat{a}_4(k)\right], \end{aligned}$$

$\hat{b}_1$ is a $(p \times 1)$ vector, $\hat{y}$ is a $(p \times 1)$ vector and $u(k)$ is a (1×1) vector.

For the case of lateral movement of the aircrafts $(n = 4,\ m = 2), \forall p$, the equation (24) becomes:

$$\begin{aligned} \hat{y}(k+1) = {} & \hat{\alpha}^T(k) \cdot \hat{Y}(k) + \hat{b}_1(k) \cdot u(k) + \\ & + \hat{b}_2(k) \cdot u(k-1); \end{aligned} \qquad (26)$$

Y and $\hat{\alpha}$ have the above forms, $\hat{b}_1, \hat{b}_2$ are $(p \times 2)$ vectors, $y -$ a $(p \times 1)$ vector and $u -$ a (2×1) vector. For the obtaining of the command law $u(k)$ we choose the performance indicator:

$$\begin{aligned} J = {} & \left[\bar{y}(k+1) - \hat{y}(k+1)\right]^T \cdot \\ & \cdot Q\left[\bar{y}(k+1) - \hat{y}(k+1)\right] + u^T(k)Ru(k), \end{aligned} \qquad (27)$$

where $\bar{y}(k+1)$ is the imposed output vector, $Q(p \times p)$ and $R(m \times m)$ are symmetric and positive definite matrices. $u(k)$ is obtained from the optimum condition $(\partial J / \partial u(k) = 0)$; for the aircraft longitudinal movement, the authors obtained the equations [9]:

$$\begin{aligned} u(k) &= G\left[\bar{y}(k+1) - \hat{\alpha}^T(k)\hat{Y}(k)\right], \\ G &= \left[R + \hat{b}_1^T(k)Q\hat{b}_1(k)\right]^{-1}\hat{b}_1^T(k)Q, \end{aligned} \qquad (28)$$

while, for the aircraft lateral movement, $u(k)$ and matrix G are calculated as bellow:

$$\begin{aligned} u(k) &= G\left[\bar{y}(k+1) - \hat{\alpha}^T(k)\hat{Y}(k) - \hat{b}_2(k)u(k-1)\right], \\ G &= \left[R + \hat{b}_1^T(k)Q\hat{b}_1(k)\right]^{-1}\hat{b}_1^T(k)Q. \end{aligned} \qquad (29)$$

The matrices Q and R may be calculated, for example, by using ALGLX algorithm [9].

The block diagram of the system for the on-line parametric estimation and optimal command of the aircraft movements is represented in Figure 3. Below we present an original algorithm (**ALGLDR**) for the on-line parametric estimation and optimal command of the aircraft movements.

modification of the transfer function coefficients. This fact happens in reality: each second, the transfer functions associated to the aircraft movement change. Thus, the original algorithm (ALGLDR) may be considered an on-line algorithm for the parametric estimation and optimal command.

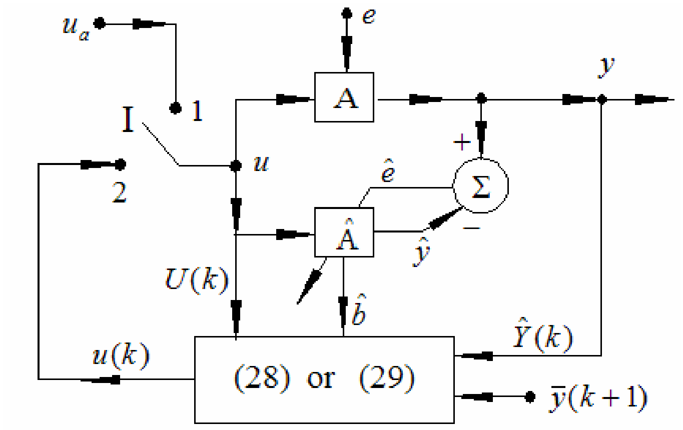

Figure 3. The system for the parametric estimation and optimal command

ALGLDR algorithm

Step 1: First of all, we make the off–line identification of the system by using, for example, the least square method; it results the parameters vector $\hat{b}_0 = \hat{b}(0)$; that refers to the coefficients $\hat{b}_j, j = \overline{1,r}, \hat{a}_i = \overline{1,n}$ of the discrete transfer functions of the aircraft estimated model Â (in Figure 3 switch "I" has position 1, e is the system disturbances and $u = u_a$ – the random input); $\hat{y}(t)$ is then calculated and the vectors $\hat{Y}_0 = \hat{Y}(0)$ and $U_0 = U(0)$ are memorized. The covariance matrix P_0 (obtained at the end of the identification $P_0 = P(0)$) is also memorized. Then, the matrices $A_d, B_d, \hat{A}_d, \hat{B}_d$ are calculated and, with these matrices, the state vectors x and $\hat{x}$ are computed by means of equations (17) or (18); these vectors (at the end of identification) are also memorized.

Step 2: For the simulation of aircraft parameters time variation, these parameters are modified (for example with 5%) and, with the new coefficients, the matrices A_d and B_d are calculated again. Because we have not, each second, the flight data from the aircraft, we simulate, without losing the generality, the on-line function of the system by permanent

Step 3: The switch "I" has now position 2 (on–line control); by means of ALGLX algorithm, the matrices Q' and R are obtained with respect to $\hat{A}_d, \hat{B}_d$ and, after that, the matrix $Q = (C^T)^+ Q' C^+$ is calculated; C^+ represents the Moore – Penrose pseudo-inverse of the matrix C.

Step 4: The matrix G from (28) or (29) is calculated with $\hat{b}_1$ extracted from $\hat{b}(k)$.

Step 5: The command $u(k)$ is obtained by using the equations (28) or (29) [9], [11].

Step 6: The vectors $x(k+1), \hat{x}(k+1)$ are obtained with (17) and (18), respectively [12]; $y(k+1)$ and $\hat{y}(k+1)$ are calculated with the formulas:

$$\begin{aligned} y(k+1) &= C_d \cdot x(k+1), \\ \hat{y}(k+1) &= \hat{C}_d \cdot \hat{x}(k+1). \end{aligned} \tag{30}$$

The vectors $\hat{Y}(k+1)$ and $U(k+1)$ are memorized and the error:

$$\hat{e}(k+1) = y(k+1) - \hat{y}(k+1) \tag{31}$$

is obtained.

Step 7: The actualization of the covariance matrix is made with the formula [13], [14]:

$$\begin{aligned} P(k+1) &= P(k) - \\ &- P(k)\frac{z(k+1)z^T(k+1)}{\lambda + z^T(k+1)P(k)z(k+1)}P(k); \end{aligned} \qquad (32)$$

λ is a constant with the value $\lambda \cong 1$. By using the equation (32), the actualization of the vector:

$$\hat{b}(k+1) = \hat{b}(k) + P(k+1)z(k+1)\hat{e}(k+1) \qquad (33)$$

is performed. In equation (33), $z(k+1)$ has the form (22).

Step 8: We increase the step value of $k(k \to k+1)$, we return to step 4 and the steps 5, 6, 7 and 8 are repeated until $k<k_{imposed}$. When $k=k_{imposed}$, the program stops and the obtained state variables $x_i(t)$ and $\hat{x}_i(t)$ are plotted.

4. Numerical Simulation Results

We use the system in Figure 1 to the stabilization of the longitudinal movement of the aircrafts. The dynamics of the aircraft longitudinal movement is:

$$\begin{bmatrix} \dot{\hat{V}} \\ \dot{\hat{\alpha}} \\ \dot{\hat{\theta}} \\ \dot{\hat{\omega}}_y \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 1 \\ 0 & 0 & 0 & 1 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} \hat{V} \\ \hat{\alpha} \\ \hat{\theta} \\ \hat{\omega}_y \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ b_p \end{bmatrix}}_{B} \delta_p, \qquad (34)$$

$$y = Cx + D\delta_p;$$

where x is the state vector, δ_p – the aircraft elevator deflection, $\hat{V}$ – the non-dimensional flight velocity, $\hat{\alpha}$ – the non-dimensional attack angle, $\hat{\theta}$ – the non-dimensional pitch angle, $\hat{\omega}_y$ – the non-dimensional pitch angular rate; the matrices A, B, C, D have the following forms [15]:

$$A = \begin{bmatrix} -0.026 & 0.025 & -0.1 & 0 \\ -0.36 & -3 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.4212 & -38.49 & 0 & -3.67 \end{bmatrix}, \qquad (35)$$

$$B^T = [0 \ 0 \ 0 \ 28], C = [0 \ 0 \ 1 \ 0], D = 0.$$

We consider that the elevator deflection (δ_p) is the only input of the system and that the non-dimensional pitch angle $(\hat{\theta})$ is the single system output. The input of the control system and of the neural network (NN) is ramp type within 2 seconds, with a slope of 0.05 rad/sec followed by a saturation zone with the amplitude $u = \delta_p = 0.1$ rad [9].

By using a simulation program (made by the paper authors in Matlab environment) the neural regulator and the neural network are trained. The training of the neural regulator and of the neural network stops when $e'' \to 0$ ($e' \to 0$ and $e \to 0$); that means that the stopping time is when $\hat{y} \to y \to r$; we have chosen $r = 0.209$ rad and $k_c = 0.7$ (the gain coefficient of the conventional regulator; $u_c = k_c e'$). The Matlab program plots the graphic of the mean square error of the identification neural network for its last training process depending on the training epochs number (Figure 4). In Figure 5 we represent the output of the control system $y(t)$ (with red circles) and the output of the neural network $\hat{y}(t)$ (with blue solid line) after the last training process. These two signals overlap; that means that the identification process is successfully done. In Figure 6, the signals $u^*(t)$ – the reference and $u'(t)$ – the neural regulator output are plotted. Because these 2 signals overlap too, the other neural network training process is successfully done $(e'' \to 0)$. The command signal of the adaptive control system $u(t)$ is represented in Figure 7.

In Figure 8 we represent the reference of the control system $r(t)$ (red dashed line) and the neural network response after the last training stage $\hat{y}(t)$ (blue solid line). The error of the system $(e'(t))$ is represented in Figure 9; this error tends to zero in a short transient regime – 3 seconds.

After the last training stage of the neural network training, the following weights have been obtained:

$$W_1 = \begin{bmatrix} -0.2257 & 1.3670 & -1.6547 & -20.5813 \\ 1.4443 & -1.1286 & 0.6718 & 5.6728 \\ 0.4096 & 0.1874 & 1.1743 & 32.6221 \\ -0.2822 & -1.3914 & 1.7408 & -20.816 \\ 0.0152 & 0.2540 & -1.1176 & 28.0938 \end{bmatrix},$$

$$B_1 = [3.5 \ -1.856 \ -2.535 \ 2.94 \ -3.68]^T,$$

$$W_2 = [-0.9948 \quad 0.2614 \quad 0.2337 \quad 0.3140 \quad -0.6736],$$

$$B_2 = 0.5451;$$

after the last training stage of the neural regulator we obtained the weights:

$$w_1 = \begin{bmatrix} -2.2990 & 3.5702 & 2.0010 & -1.4017 \\ -0.0532 & -0.8236 & -2.8633 & -2.6471 \\ 2.0321 & 1.3235 & -1.3111 & 2.5900 \\ -0.4527 & 1.5959 & -1.6810 & 0.0107 \\ -0.9301 & -0.3484 & 1.3832 & 0.7444 \end{bmatrix},$$

$$b_1 = [0.437 \ -1.699 \ 0.162 \ 0.378 \ -1.167]^T,$$

$$w_2 = [0.1880 \ -0.0120 \ 0.0367 \ -2.0067 \ -0.2235],$$

$$b_2 = 0.3585.$$

The stability of the control system is obvious from the graphical characteristics; thus, the identification processes are successfully done (the mean square error of the neural network training process tends to zero – Figure 4, the output of the control system and the output of the neural network are the same – Figure 5, the reference and the neural regulator output overlap – Figure 6). The system stability can be also remarked in Figure 8 (the output of the system (y) and its estimation (ŷ) tend to the desired input of the system (r)) and in Figure 9 (the difference between the system desired output and its nominal output tends to zero e' = r – y → 0).

For the validation of the ALGLDR algorithm (block diagram in Figure 3), the authors made other Matlab/Simulink programs and tested this algorithm for the longitudinal and lateral movement of aircrafts.

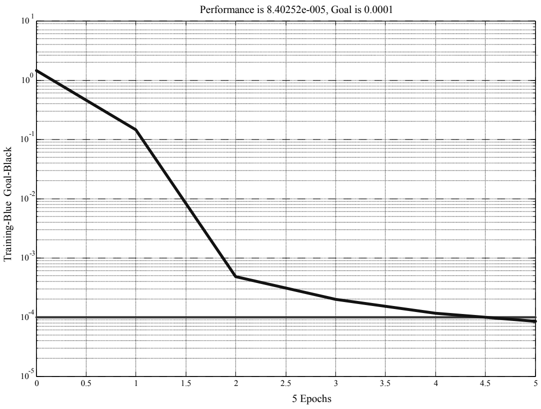

Figure 4. The mean square error of the neural network training process

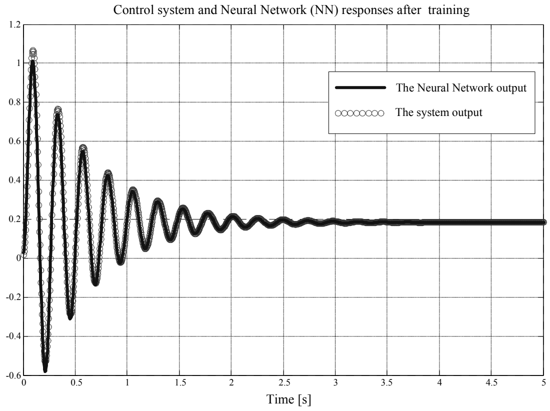

Figure 5. The output of the control system and of the neural network after last training process

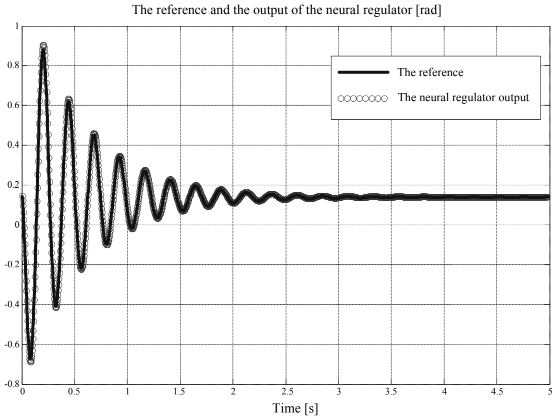

Figure 6. The signals $u^*(t)$ and $u'(t)$

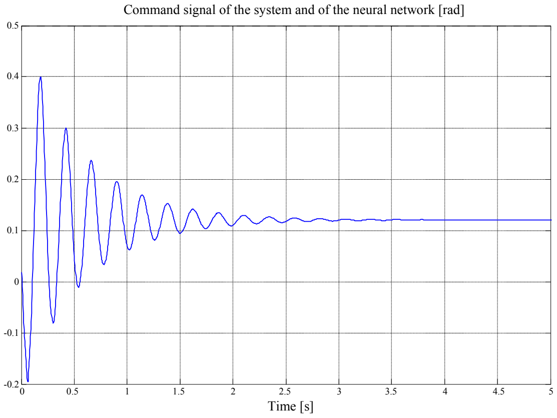

Figure 7. The command signal $u(t)$

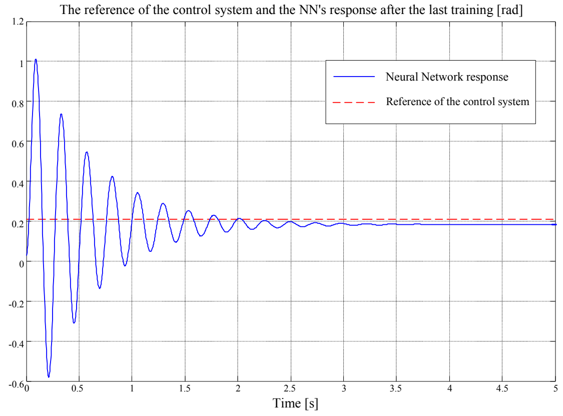

Figure 8. Reference of the control system and the NN's response after the last training stage

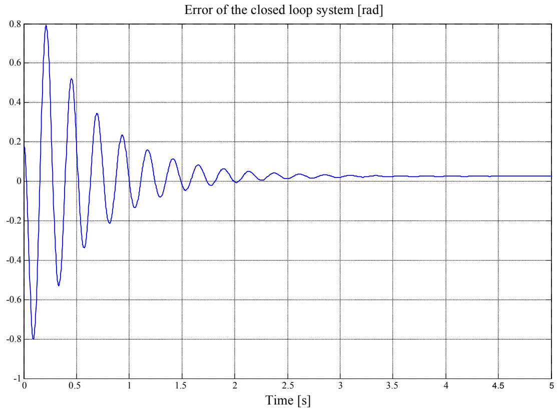

Figure 9. The error of the closed loop system

Let us consider another form for the longitudinal movement of the aircrafts [16]:

$$\begin{bmatrix} \Delta\dot{V}_x \\ \Delta\dot{\alpha} \\ \Delta\dot{\theta} \\ \Delta\dot{\omega}_y \end{bmatrix} = \begin{bmatrix} -0.007 & 0.012 & -9.81 & 0 \\ -0.128 & -0.54 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.065 & 0.96 & 0 & -0.99 \end{bmatrix} \cdot \begin{bmatrix} \Delta V_x \\ \Delta\alpha \\ \Delta\theta \\ \Delta\omega_y \end{bmatrix} + \begin{bmatrix} 0 \\ -0.04 \\ 0 \\ -12.5 \end{bmatrix} \delta_p .$$

where V_x is the longitudinal component of flight velocity, $\alpha -$ the aircraft attack angle, $\theta -$ the aircraft pitch angle, ω_y – the aircraft pitch angular rate and δ_p – the aircraft elevator deflection. The symbol Δ is used here to express the deviations of the nominal velocity, angles or angular rate from their desired values.

Using, step by step, the AGLDR algorithm, we obtained the time variations of the variables $x_1 = \Delta V_x, \hat{x}_1 = \Delta\hat{V}_x, x_2 = \Delta\alpha, \hat{x}_2 = \Delta\hat{\alpha},$ $x_3 = \Delta\theta, \hat{x}_3 = \Delta\hat{\theta},\ x_4 = \Delta\omega_y, \hat{x}_4 = \Delta\hat{\omega}_y, u = \delta_p$ in Figure 10 (the states x_i – blue solid line, their estimates $\hat{x}_i$ – red dashed line).

The ALGLDR algorithm is also tested for the lateral movement of the aircrafts. We consider the lateral movement of a Boeing 747 which flies with Mach number $M = 0.8$ at the altitude $H = 40000$ ft; the state equation corresponding to the aircraft movement in longitudinal plane is [9]:

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{\omega}_z \\ \Delta\dot{\omega}_x \\ \Delta\dot{\varphi} \end{bmatrix} = \begin{bmatrix} -0.0558 & -0.9968 & 0.0802 & 0.0415 \\ 0.598 & -0.115 & -0.0318 & 0 \\ 0.305 & 0.388 & -0.465 & 0 \\ 0 & 0.0805 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta\omega_z \\ \Delta\omega_x \\ \Delta\varphi \end{bmatrix} + \\ + \begin{bmatrix} 0.0073 & 0 \\ -0.475 & 0.123 \\ 0.153 & 1.063 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_d \\ \delta_e \end{bmatrix},$$

where β is the sideslip angle, ω_x – the roll angular rate, ω_z – the yaw angular rate, φ – the aircraft roll angle, δ_d – the deflection of the aircraft direction and δ_e – the deflection of aircraft ailerons. In this case, the symbol Δ also expresses the deviations of the variables $\beta, \omega_z, \omega_x$ and φ from their desired (imposed) values.

Using again, step by step, the AGLDR algorithm, we obtained the time variations of the variables $x_1 = \Delta\beta, \hat{x}_1 = \Delta\hat{\beta}, x_2 = \Delta\omega_z,$ $\hat{x}_2 = \Delta\hat{\omega}_z, x_3 = \Delta\omega_x, \hat{x}_3 = \Delta\hat{\omega}_x,\ x_4 = \Delta\varphi,$ $\hat{x}_4 = \Delta\hat{\varphi}, u_1 = \delta_d$ and $u_2 = \delta_e$ in Figure 11 (the states x_i – blue solid line, their estimates $\hat{x}_i$ – red dashed line).

The estimation processes are successfully achieved (the four state variables x_i and the four estimated state variables $\hat{x}_i$ overlap). Moreover, the stability of the system is assured by the stationary values of all the state variables; all the state variables are expressed as deviations (the symbol Δ is used for this) of their nominal values from their desired values. That is why, when $\Delta x_i \to 0$, the variables x_i tend to their desired value (these values are imposed, during the flight by the aircraft pilot).

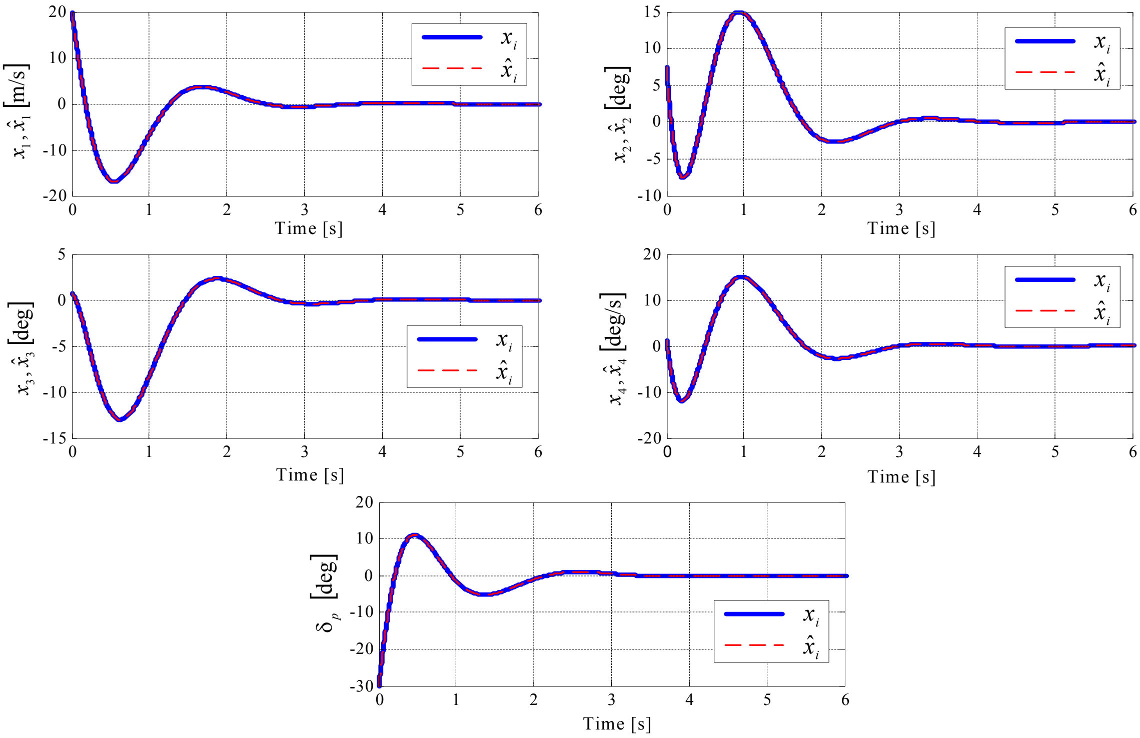

Figure 10. The dynamics of the state variables and their estimations for the aircraft longitudinal movement by using the ALGLDR algorithm

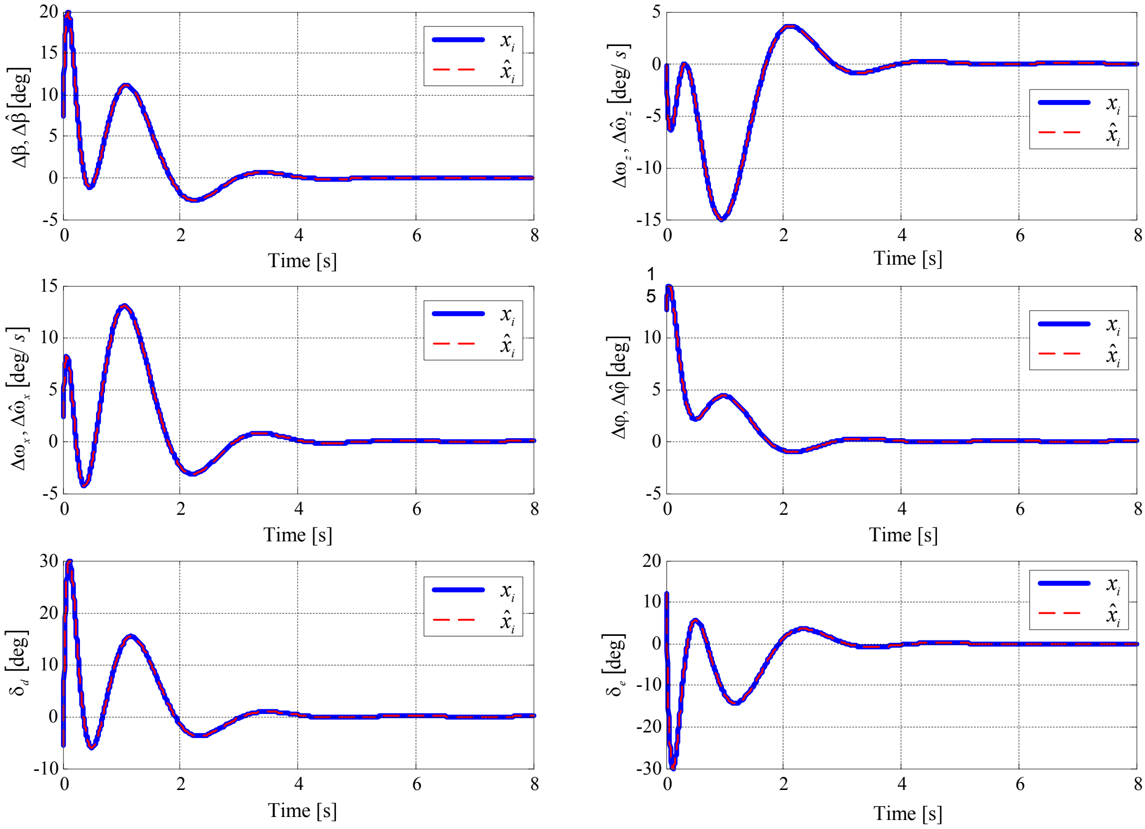

Figure 11. The dynamics of the state variables and their estimations for the aircraft lateral movement by using the ALGLDR algorithm

5. Conclusions

The paper presents new systems for the identification and neuro-adaptive command of the aircrafts and a new system for parametric estimation and discrete optimal command with direct applicability to the control of the flying objects movement. The identification/estimation and the automatic command algorithms represent authors' contributions. Also, the block diagrams of the three presented systems (Figure 1, 2 and 3) are original issues and may be used with good results to the command and control of aircrafts or rockets. The theoretical results are validated by three numerical simulations in Matlab/Simulink environment. The command system presented in Figure 1 consists of a neural regulator, a conventional regulator and an identification neural network. This system works in two regimes: identification regime (I) and automatic control (R). The neural regulator must model (form) the inverse of the neural network function. Another new neuro-adaptive command system, which may be used to the stabilization of the aircrafts movement, is the original hierarchical structure in Figure 2. It is characterized by an initialization stage and a training stage (the two neural networks – NNc and NNx are trained).

Acknowledgments

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. CALISE, A. J., **Flight Evaluation of an Adaptive Velocity Command System for Unmanned Helicopters,** AIAA Guidance, Navigation and Control Conference and Exhibit, vol. 2, 11–14 August, Austin, Texas, 2003.
2. CALISE, A. J., N. HOVAKYMYAN, M. IDAN, **Adaptive Output Control of Nonlinear Systems Using Neural Networks,** Automatica, vol. 37, (8), August, 2001, pp. 1201-1211.
3. CALISE, A. J., B. J. YANG, J. I. CRAIG, **Augmentation of an Existing Linear Controller with an Adaptive Element,** American Control Conference, Nr. ACC02, IEEE 1331, 2002. In Proceedings, p. 6.
4. HOSEINI, S. M., M. FARROKHI, A. J. KASHKONEI, **Robust Adaptive Control Systems Using Neural Networks,** The International Journal of Control, 2006, p. 7.
5. CALISE, A. J., E. N. JOHNSON, M. D. JOHNSON, J. E. CORBAN, **Applications of Adaptive Neural – Networks Control to Unmanned Aerial Vehicles,** Journal of Harbin Institute of Technology, 2006, vol. 38, No. 11, pp. 1865-1869.
6. LYSHEVSKI, S. E., **Identification Nonlinear Flight Dynamics: Theory and Practice,** IEEE Transactions on Aerospace and Electronic System, vol. 36, no. 2, April, 2000, pp. 383-392.
7. CHEN, X., W. GAOFENG, Z. WEI, C. SHENG, S. SHILEI, **Efficient Sigmoid Function for Neural Networks based FPGA Design,** Springer Publisher, 2006.
8. SHAO, L., J. WANG, S. SHAO, **Study on the Fitting Ways of Artificial Neural Networks,** Journal of Coal Science and Engineering, vol. 14, no.2, June, 2008.
9. LUNGU, M., **Sisteme de conducere a zborului,** Editura Sitech, Craiova, 2008, p. 329.
10. LUNGU, R., M. LUNGU, L. DINCA, E. STOENESCU, **On–line Parametric Identification and Discrete Optimal Command for the Aircrafts Longitudinal Movement,** Proceedings of the 7th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE '08), Venice, Italy, November 21-23, 2008.
11. LUNGU, R., M. LUNGU, **Optimal Control of the Rocket's Lateral Deviation in Rapport with Equal Signal Line.** 6th International Conference on Electro-mechanical and Electro-energetic Systems, September 2007, Chisinau, Moldova Republic.
12. LEE, T., Y. KIM, **Nonlinear Adaptive Flight Control Using Backsteping and Neural Networks Controller,**. Journal of

Guidance, Control and Dynamics, vol. 24, no. 4, July – Aug, 2007, pp. 675-782.

13. BALESTRASSI, P. P., E. POPOVA, A. P. PAIVA, J. W. MARANGON LIMA, **Design of Experiments on Neural Network's Training for Nonlinear Time Series Forecasting**, Elsevier Journal – Neuro-computing, 72, 2009, pp. 1160-1178.

14. LUNGU, M., R. LUNGU, N. JULA, C. CEPISCA, M. CALBUREANU, **Aspects Regarding a New Method for the Optimal Law's Synthesis of Aircrafts' Move**, WSEAS Transactions on Circuits and Systems, vol. 7, June 2008, pp. 492-501.

15. BODNER, V.A., **Teoria avtomaticeskogo upravlenia poletom,** Izd. Nauka, Moscova, 1964.

16. DONALD, M. **Automatic Flight Control Systems,** New York, London, Toronto, Sydney, Tokyo, Singapore, 1990.