# Non-linear adaptive system for the control of the aircrafts' roll angle

Mihai Lungu, Romulus Lungu, Lucian Sepcu

***Abstract*** — The paper presents a new complex adaptive non-linear system with one input and one output (SISO) which is based on dynamic inversion. Linear dynamic compensator makes the stabilization command of the linearised system using as input the difference between closed loop system's output and the reference model's output. The state vector of the linear dynamic compensator, the output and other state variables of the control system are used for adaptive control law's obtaining; this law is modeled by a neural network. The aim of the adaptive command is to compensate the dynamic inversion error. Thus, the command law has two components: the command given by the linear dynamic compensator and the adaptive command given by the neural network. As control system one chooses the non-linear model of the aircrafts' roll movements. The reference model is linear. One obtains the structure of the adaptive control system of the roll angle and the Matlab/Simulink models of the adaptive command system's subsystems. Thus, characteristics that describe the adaptive command system's dynamics are obtained.

***Keywords*****—** neural network, dynamic inversion, aircrafts, roll angle.

## I. Introduction

THE complexity and incertitude that appear in the non-linear and instable phenomena are the main reasons that require the projecting of non-linear adaptive structures for control and stabilization; in these cases the linear models are far from a good describe of the flying objects' dynamic. Another reason is the non-linear character of the actuators. The observers must be easily adaptable and their project algorithms must allow the state's estimation of the flying object even in the case of their failure or no use of the damaged sensors' signals. In these situations, it's good to use the real time adaptive control based on neural networks and dynamic inversion of the unknown or partial known nonlinearities from the dynamic model of the flying object [1]. The neural network's training is based on the signals from state observers; these observers get information about the control system's error [2], [3], [4].

Manuscript received March 27, 2010.

Mihai Lungu is lecturer at the University of Craiova, Faculty of Electrical Engineering, Department of Avionics. 107, Decebal Blv., Craiova, Romania, Email: Lma1312@yahoo.com, mlungu@elth.ucv.ro.

Romulus Lungu is professor at the University of Craiova, Faculty of Electrical Engineering, Department of Avionics. 107, Decebal Blv., Craiova, Romania, Email: rlungu@elth.ucv.ro.

Lucian Sepcu is assistant at the University of Craiova, Faculty of Electrical Engineering, Department of Avionics. 107, Decebal Blv., Craiova, Romania, Email: lsepcu@elth.ucv.ro.

## II. Dynamic SYSO Systems

Let's consider the dynamic system (A) with single input and single output described by equations

$$\dot{x} = f(x,u),\ y = h(x), \tag{1}$$

with $x(n \times 1), n -$ unknown $f$ and $h -$ unknown nonlinear functions, $u$ and $y -$ measurable.

One projects an adaptive control law $v$ after (in rapport with) the output; the neural network (NN) models a function that depends on the values of input and output of the system (A) at different time moments so that $y(t)$ follows the finite $y(t)$. The feedback linearization may be made through transformation [5], [6]

$$v = \hat{h}_r(y,u), \tag{2}$$

where $v$ is the pseudo-command signal and $\hat{h}_r(y,u) -$ the best approximation of $h_r(x,u) = h_r(x(y),\ u)$.

Equation (2) is equivalent with the following one

$$u = \hat{h}_r^{-1}(y,v)\ . \tag{3}$$

If $\hat{h}_r \equiv h_r$, one yields $y^{(r)} = v$; otherwise $\left(\hat{h}_r \neq h_r\right)$

$$y^{(r)} = v + \varepsilon, \tag{4}$$

where

$$\varepsilon = \varepsilon(x,u) = h_r(x,u) - \hat{h}_r(y,u) \tag{5}$$

is the approximation of function $h_r$ (inversion error). Assessing $y$ to follow $\bar{y}$, then $v$ has form [5], [6], [7], [8]

$$v = \bar{y}^{(r)} + v_{pd} - v_a + \bar{v}, \tag{6}$$

where $v_{pd}$ is the output of the dynamic linear compensator for stabilization, used for liniarised dynamic (4), with $\varepsilon = 0, v_a -$ the adaptive command that must compensate $\varepsilon$ and $\bar{v}$ has the form [5], [9]

$$\bar{v} = k_z\left( \left\|\hat{Z}\right\|_F + \bar{Z} \right)\frac{\bar{E}}{\left\|\bar{E}\right\|}\left\|\hat{E}\right\| + k_v\bar{E}, \tag{7}$$

with $k_z, k_v > 0$ gain constants, $\left\|\hat{Z}\right\|_F$ – the Frobenius norm of matrix $\hat{Z}, \bar{Z}$ – the ideal matrix of the neural network and $\bar{E} = \hat{E}P\bar{B},$ with $\hat{E}, P$ and $\bar{B}$ – matrices. The derivative $\bar{y}^{(r)}$ is introduced for the conditioning of the dynamic error

$\tilde{y} = \bar{y} - y$. This derivative is given by a reference model (command filter) [6]. $\bar{y}^{(r)}$ may be cumulated with other signals and it results the component $v_r$ of form (11).

Considering

$$Y^T = \left[y \ \dot{y} \ \ldots \ y^{(r-1)}\right],\ Z^T = \left[v \ \dot{v} \ \ldots \ v^{(p)}\right], \\ \lambda^T = \left[\lambda_0 \ \lambda_1 \ \cdots \ \lambda_{r-1}\right],\ b^T = \left[b_0 \ b_1 \ \cdots \ b_p\right], \tag{8}$$

with $b_i, i = \overline{0,p}, \lambda_j, j = \overline{0,r-1}$ – the coefficients of the numerator and denominator of the transfer function for the system with input $u_n$ and output $y$, the linear system with input $v$ and output $y$ is described by equation

$$y^{(r)} = -\lambda^T Y + b^T Z + \varepsilon. \tag{9}$$

If $p = 0$, then $Z = v, b = b_0$ and the previous equation becomes

$$y^{(r)} = -\lambda^T Y + b_0 v + \varepsilon. \tag{10}$$

In the particular case $y^{(r)} = \bar{y}^{(r)}$, one obtains

$$v_r = \frac{1}{b_0}\left(\bar{y}^{(r)} + \lambda^T \bar{Y}\right). \tag{11}$$

The compensator may be described by state equations

$$\dot{\varsigma} = A_c\varsigma + b_c e, v_{pd} = c_c\varsigma + d_c e, \tag{12}$$

where $\varsigma$ has at least dimension $(r-1)$,

$$e = \tilde{y} = ce, e^{\mathrm{T}} = \left[e \ \dot{e} \ \cdots \ e^{(r-1)}\right], c = \left[1 \ 0 \ 0 \ \cdots \ 0\right]_{1\times r}. \tag{13}$$

The state equation of the linear subsystem with input $(v + \varepsilon)$ and output $y$ is

$$\dot{x} = Ax + b(v + \varepsilon), v = v_{pd} - v_a + \bar{v}, \tag{14}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{r\times r},\ b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}_{r\times 1}. \tag{15}$$

The stable state $\bar{x}\left(\dot{\bar{x}} = v = \varepsilon = 0\right)$ verifies equation $A\bar{x} = 0$ and, taking into account (14), leads to the equation of the error vector $e \equiv \tilde{x} = \bar{x} - x$,

$$\dot{e} = Ae - bv_{pd} + b\left(v_a - \bar{v} - \varepsilon\right). \tag{16}$$

With notations

$$E = \begin{bmatrix} e \\ \varsigma \end{bmatrix}, \bar{A} = \begin{bmatrix} A - d_c bc & -bc_c \\ b_c c & A_c \end{bmatrix}, \bar{b} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \bar{C} = \begin{bmatrix} c & 0 \\ 0 & I \end{bmatrix}, \tag{17}$$

where $I$ is the identity matrix, one obtains

$$\dot{E} = \bar{A}E + \bar{b}\left(v_a - \bar{v} - \varepsilon\right), z = \bar{C}E; \tag{18}$$

$A_c, b_c, c_c, d_c$ from (12) are calculated so that $\bar{A}$ is a Hurwitz matrix.

For estimation of the vector $E$ one uses a linear state observer of order $(2r-1)$ described by equations

$$\dot{\hat{E}} = \bar{A}\hat{E} + L(z - \hat{z}), \hat{z} = \bar{C}\hat{E}, \tag{19}$$

with the gain matrix $L$ calculated so that matrix $\tilde{A} = \left(\bar{A} - L\bar{C}\right)$ is stable.

Considering $w$ – the sensor measure error, $y_m$ – the measured value of $y$, then $\tilde{y}_m = \bar{y} - y_m = \tilde{y} + w$ and the compensator's equations become

$$\dot{E} = \bar{A}E + \bar{b}\left(v_a - \bar{v} - \varepsilon\right) + Gw, z = \bar{C}E + Hw. \tag{20}$$

with $H^T = [1 \ 0], G^T = \left[-bd_c \ b_c\right]$.

If state $\varsigma$ of the compensator is known, one uses a reduced order observer for estimation of vector $e$

$$\dot{\hat{e}} = \bar{A}\hat{e} + L_r\left(z_1 - \hat{z}_1\right), z_1 = c\hat{e}. \tag{21}$$

The gain matrix $L_r$ is obtained so that matrix $\tilde{A} = \left(\bar{A} - L_r c\right)$ is stable. With vectors $\hat{e}$ and $\varsigma$ vector $\hat{E}^T = \left[\hat{e} \ \varsigma\right]$ is obtained. The signal $\bar{E} = \hat{E}^T P\bar{b}$ is used for neural network's adapting; the weights $\hat{W}$ and $\hat{V}$ are obtained with equations

$$\dot{\hat{W}} = -\Gamma_W\left[2\left(\hat{\sigma} - \hat{\sigma}'\hat{V}^T\eta\right)\hat{E}^T P\bar{B} + k\left(\hat{W} - \hat{W}_0\right)\right], \\ \dot{\hat{V}} = -\Gamma_V\left[2\eta\hat{E}^T P\bar{B}\hat{\sigma}' + k\left(\hat{V} - \hat{V}_0\right)\right], \tag{22}$$

where the role of $\bar{B}$ is played by $\bar{b}$. In (22) $\sigma$ is the sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-az}}, \tag{23}$$

$\hat{\sigma}' = \left.\frac{\mathrm{d}\hat{\sigma}(z)}{\mathrm{d}z}\right|_{z=z_0}$ is the Jacobian of vector $\hat{\sigma}$, $\hat{W}_0$ and $\hat{V}_0$ – the initial values of weights $\hat{W}, \hat{V}, \Gamma_W, \Gamma_V > 0$, $k > 2\left(k_1^2 + \gamma_1^2\left\|P\bar{B}\right\|^2\right)$, $k_1 = k_2\alpha_1 + \left\|P\bar{B}\right\|\gamma_1$, $k_2 = \left\|P\bar{B}\right\| + \left\|\tilde{P}\bar{B}\right\|$, $P$ and $\tilde{P}$ – the solutions of Liapunov equations

$$\bar{A}^T P + P\bar{A} = -Q, \\ \tilde{A}^T\tilde{P} + \tilde{P}\tilde{A} = -\tilde{Q}. \tag{24}$$

$P$ from the signal used for the neural network's adapting is the solution of first equation (24) with $\bar{A} = \left(A - d_c bc\right)$. Second output of the compensator $\left(\tilde{y}_a\right)$ is used for obtaining of an error signal that is useful for adapting of the neural network's weights (fig.1).

From (4) and (6) one yields

$$y^{(r)} = \bar{y}^{(r)} + v_{pd} - v_a + \bar{v} + \varepsilon, \tag{25}$$

equivalent with the dynamic error's equation

$$\tilde{y}^{(r)} = -v_{pd} + v_a - \bar{v} - \varepsilon. \tag{26}$$

Error $\varepsilon$ may be approximated with the output of a linear neural network NN [5]

$$\varepsilon = W^T\Phi(\eta) + \mu(\eta), \|\mu\| < \mu^*, \tag{27}$$

where $W$ is the weights' matrix for the connections between layer 2 and layer 3 (NN has 2 layers), $\mu(\eta)$ – the reconstruction error of the function and $\eta$ – the input vector of NN

$$\eta = \left[1 \ \bar{v}_d^T(t) \ \bar{y}_d^T(t)\right]^T, \tag{28}$$

where

$$\bar{v}_d^T(t) = \left[v(t) \ v(t-d) \ \cdots \ v\left(t - \left(n_1 - r - 1\right)d\right)\right]^T, \bar{y}_d^T(t) = \left[y(t) \ y(t-d) \ \cdots \ y\left(t - \left(n_1 - 1\right)d\right)\right]^T, \tag{29}$$

with $n_1 \geq n$ and $d > 0$; $v_a$ is projected so that

$$v_a = \hat{W}^T\Phi(\eta), \tag{30}$$

where $\hat{W}$ is the estimation of $W$.

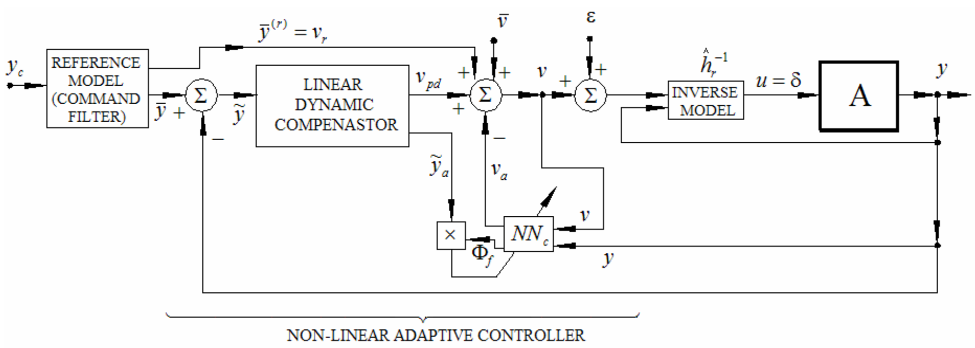


Fig.1. Automat control system with non-linear adaptive controller

Actuators' characteristics (time delays, nonlinearities with saturation zone) lead to neural network 's adapting difficulties. This is why a block "PCH" is introduced; it limits the adaptive pseudo-control $v_a$ and $v$ by the mean of one component which represents an estimation of the actuator's dynamic (PCH – Pseudo control Hedging). PCH "moves back the reference model" introducing a correction of the reference model's response; this correction depends on actuator's position [3], [9]. Because the dependence between $\delta$ and $\delta_c$ is expressed by a non-linear function $h_a$, one yields

$$\hat{h}_r(x,\delta_c) \neq \hat{h}_r(x,\hat{\delta});\qquad(31)$$

it results a difference between the two functions

$$v_h = \hat{h}_r(x,\delta_c) - \hat{h}_r(x,\hat{\delta}).\qquad(32)$$

Taking into account that

$$\hat{h}_r(x,\delta_c) = \hat{h}_r\left(x,\hat{h}_r^{-1}(x,v)\right) = v,\qquad(33)$$

function (32) becomes

$$v_h = v - \hat{h}_r(x,\hat{\delta}).\qquad(34)$$

This signal is introduced in the reference model as an additional input [3]; one compares it with $\bar{y}^{(r)}$ inside of the reference model and, after integration, it leads to the modify of the signals $\bar{y}$ and $\tilde{y}$.

## III. ADAPTIVE SYSTEM FOR THE COMMAND OF THE AIRCRAFTS' ROLL ANGLE

One considers the nonlinear model of the roll movement described by equations [11]

$$\begin{aligned}&\dot{x}_1 = x_2, \dot{x}_2 = f(x) + d_0 u, x_1 = \varphi, x_2 = \dot{\varphi}, u = \delta_e;\\ &f(x) = b_1 x_1 + b_2 x_2 + b_3|x_1|x_2 + b_4|x_2|x_2 + b_5 x_1^3;\end{aligned}\qquad(35)$$

where $b_1 = -0.0186, b_2 = 0.0152, b_3 = -0.0625,$ $b_4 = 0.01, b_5 = 0.021, d_0 = 1.$

The matriceal description of the movement is

$$\begin{aligned}&\dot{x} = Ax + h_r(x,u), x^T = [x_1\ \ x_2], A = \begin{bmatrix}0 & 1\\ b_1 & b_2\end{bmatrix},\\ &h_r(x,u) = b_3|x_1|x_2 + b_4|x_2|x_2 + b_5 x_1^3 + d_0 u.\end{aligned}\qquad(36)$$

The system (35) is equivalent with equation

$$\ddot{\varphi} = b_2\dot{\varphi} + b_1\varphi + b_3|\varphi|\dot{\varphi} + b_4|\dot{\varphi}|\dot{\varphi} + b_5\varphi^3 + d_0 u,\qquad(37)$$

where $\varphi$ is the aircraft roll angle and $u = \delta_e$ – the ailerons deflection. From (37) it results the relative grade of system $(r = 2)$ and the transfer function $H_d(s)$

$$H_d(s) = \frac{1}{s^2 + \lambda_1 s + \lambda_0}.\qquad(38)$$

One chooses the reference model described by equation

$$\bar{y} = \frac{\omega_{r0}^2}{s^2 + 2\xi_0\omega_{r0}s + \omega_{r0}^2}\,y_c,\qquad(39)$$

with $\xi_0 = 0.7$ and $\omega_{r_0} = 1\text{rad/s}; y = \varphi, \bar{y} = \bar{\varphi}.$ With (38), equation (9) becomes

$$\ddot{\varphi} = -\lambda_1\dot{\varphi} - \lambda_0\varphi + v + \varepsilon\qquad(40)$$

and by elimination of $\ddot{\varphi}$ between this and equation (37) and identification, one yields

$$\begin{aligned}&u = \delta_e = \frac{1}{d_0}v = \hat{h}_r^{-1}(x,v),\\ &\varepsilon = (\lambda_0 + b_1)\varphi + (\lambda_1 + b_2)\dot{\varphi} + b_3|\varphi|\dot{\varphi} + b_4|\dot{\varphi}|\dot{\varphi} + b_5\varphi^3.\end{aligned}\qquad(41)$$

Equation (26) becomes

$$\ddot{\tilde{\varphi}} = -v_{pd} + v_a - \bar{v} - \varepsilon,\qquad(42)$$

with $v_{pd}$ of form [4]

$$v_{pd} = k_p\tilde{y} + k_d\dot{\tilde{y}}.\qquad(43)$$

Implicit the dynamic equation of the error $\tilde{\varphi} = \bar{\varphi} - \varphi$ is

$$\begin{bmatrix}\dot{\tilde{y}}\\ \ddot{\tilde{y}}\end{bmatrix} = b_0\begin{bmatrix}0 & 1\\ -k_p & -k_d\end{bmatrix}\begin{bmatrix}\tilde{y}\\ \dot{\tilde{y}}\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}(b_0 v_a - b_0\bar{v} - \varepsilon),\qquad(44)$$

with $b_0 = 1$. One chooses $\lambda_0 = \lambda_1 = 1, k_p = \omega_0^2\ (\omega_0 = 1\text{rad/s})$ and $k_d = 2\xi\omega_0\ (\xi = 0.7)$.

One considers $\boldsymbol{E} = \boldsymbol{e} = \left[\tilde{\varphi}\ \ \dot{\tilde{\varphi}}\right]^T, \bar{C} = \bar{c} = [1\ \ 0], z = \tilde{y} = e$ and $\hat{z} = \hat{e}$; $\hat{\boldsymbol{E}}$ - the observer state (19) The gain matrix of the observer $L$ is obtained so that matrix $\tilde{A} = (\bar{A} - LC)$ is stable; $\bar{A}$ is the matrix of system from equation (44) with $b_0 = 1$.

The component $v_a$ is calculated using equation

$$\boldsymbol{v}_a = \hat{W}^T\sigma\left(\hat{V}^T\boldsymbol{\eta}\right),\qquad(45)$$

$P$ – the solution of Lyapunov equation (24) and $Q = I_2$; one chooses $\Gamma_V = 8, \Gamma_W = 10, k = 5$ and activation potentials between 0.1 and 1 [4]. The input vector $\eta$ has the form (28) with components (29); $n_1 = n = 2, d = 0.05$;

$$\eta^T = [1\ \ v(t)\ \ v(t-d)\ \ v(t-2d)\ \ v(t-3d)\ \ y(t)\ \ y(t-d)].\qquad(46)$$

For the calculus of component $\bar{v}$ one uses (7), where

$k_z = 0.6, k_v = 0.8, \bar{Z} = 30$.

The block diagram of the system for the control of aircrafts' roll angle is presented in fig.2, while the block diagram of the reference model is the one from fig.3 with $y = \varphi, y_c = \varphi_c, \bar{y} = \bar{\varphi}, \eta$ of form (45), $H_d(s)$ of form (38) and $\varepsilon$ obtained with (41). One has chosen the initial values $\varphi(0) = 20\text{grd}, \dot{\varphi}(0) = 100\text{grd/s}$.

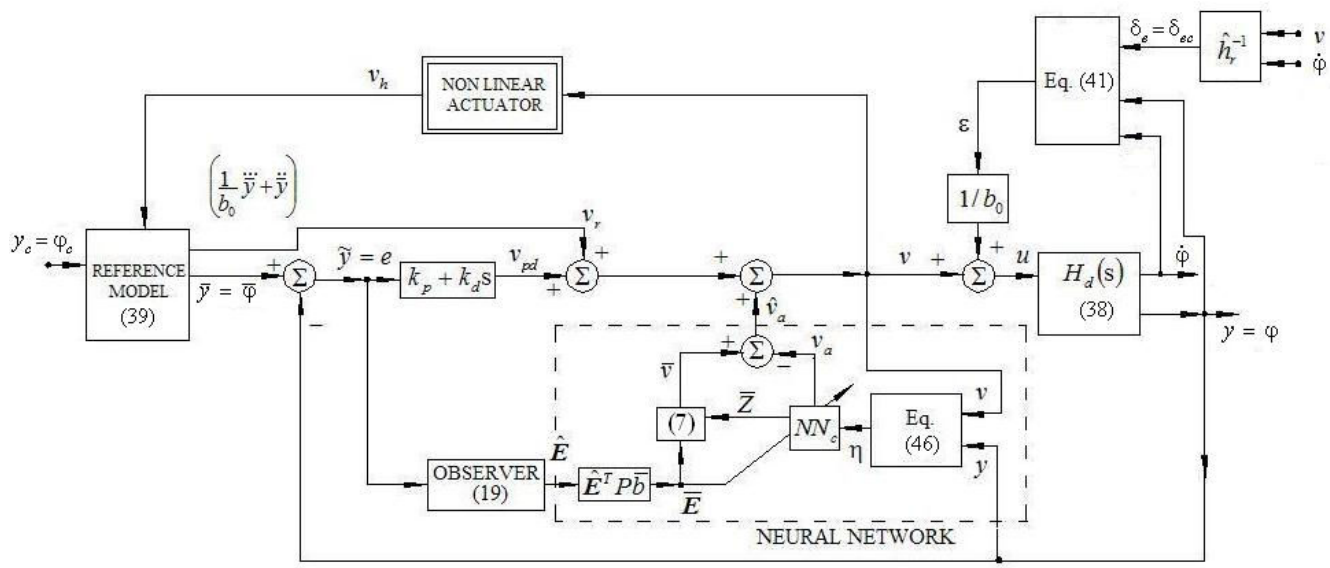


Fig.2. Block diagram of the system for the control of aircrafts' roll angle

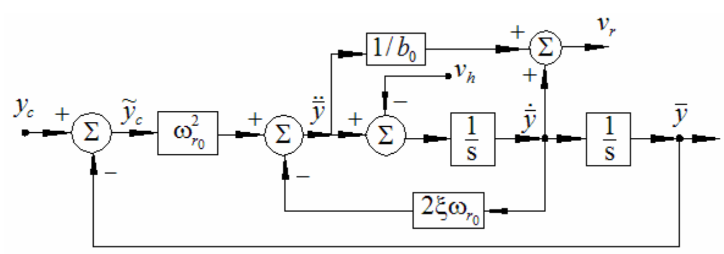

Fig.3. The block diagram of the reference model

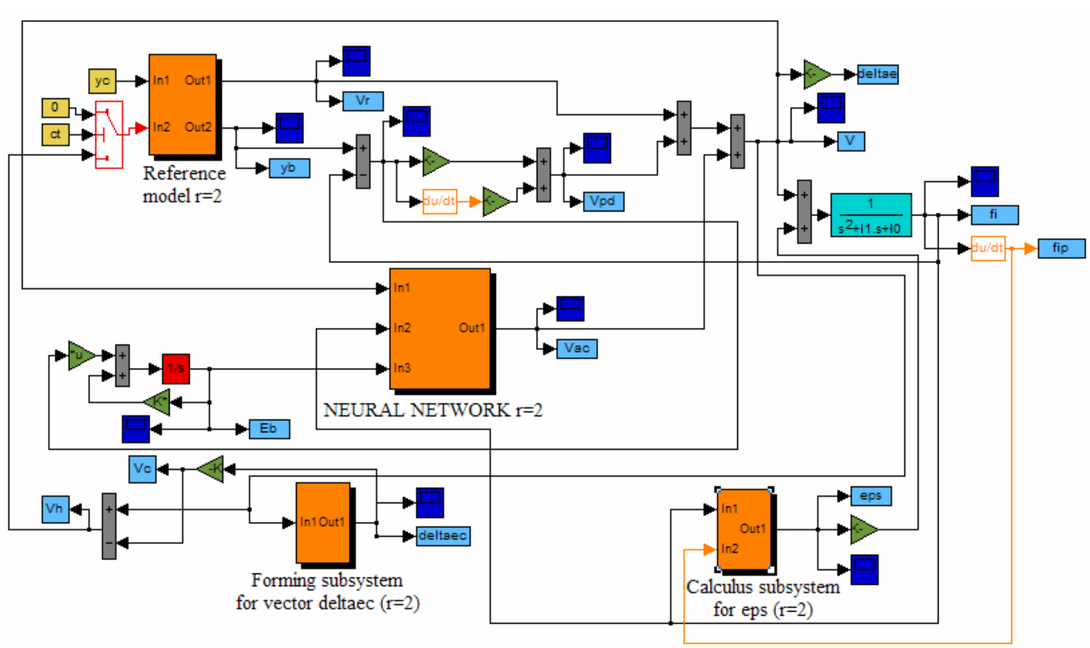


Fig.4 Matlab/Simulink model for the structure from fig.2

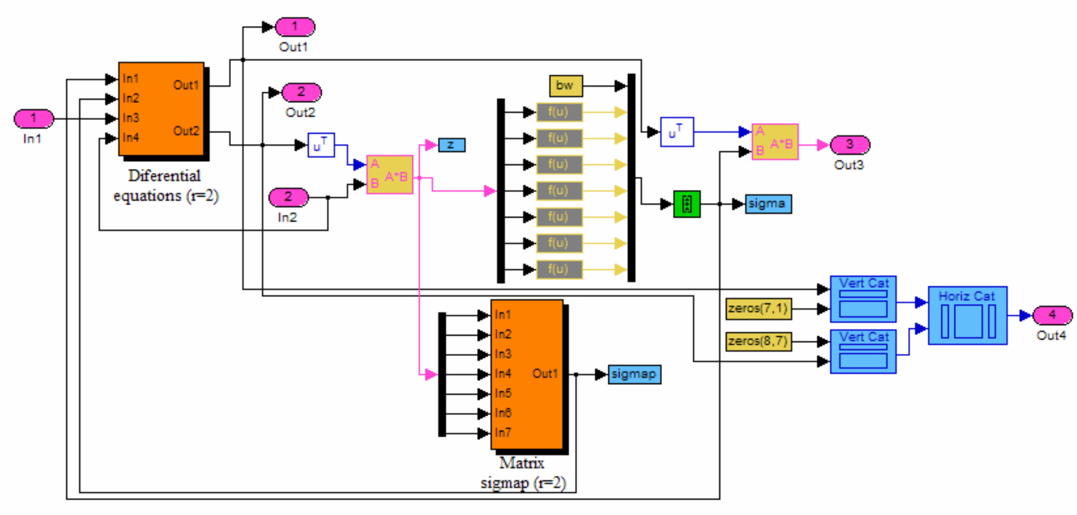


Fig.5 Matlab/Simulink model for “NEURAL NETWORK r=2”

In fig.4, the Matlab/Simulink model for the structure from fig.2 is presented. The four subsystems (“NEURAL NETWORK r=2”, “Reference model r=2”, “Forming subsystem for vector deltaec (r=2)” and “Calculus subsystem for eps (r=2))” are presented in figures 5-8.

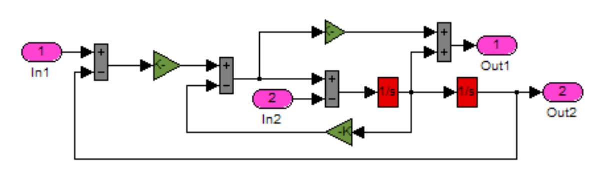


Fig.6 Matlab/Simulink model for “Refecence model r=2”

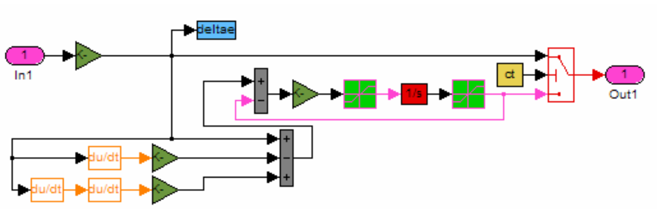


Fig.7 Matlab/Simulink model of “Forming subsystem for vector deltaec (r=2)”

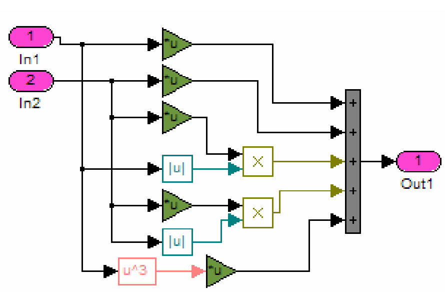


Fig.8 Matlab/Simulink model of “Calculus subsystem for eps (r=2)”

For $\varphi_c = 4°; \overline{c} = [1\ 0]; \overline{b}^T = [0\ 1]$, one obtains

$$L = \begin{bmatrix} -0.13 \\ 0.11 \end{bmatrix}, P = \begin{bmatrix} 1.41 & 0.50 \\ 0.50 & 0.71 \end{bmatrix} \qquad (47)$$

and matrices $W$ and $V$ after neural network training are

$$W^T = [-20.7035\ -3.7136\ -1.7557\ -8.8042\ -6.8664\ -5.9634\ -6.1923\ -6.4364],$$

$$V = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.02 & 0.02 & 0.04 & 0.08 \\ 0 & 0 & 0.01 & 0.01 & 0.01 & 0.01 & 0.02 \\ 0.01 & 0.01 & 0.02 & 0.02 & 0.02 & 0.03 & 0.03 \\ -0.27 & -0.29 & -0.30 & -0.31 & -0.32 & -0.24 & -0.08 \\ 0.02 & 0.03 & 0.04 & 0.06 & 0.09 & 0.19 & 0.40 \\ 0.06 & 0.07 & 0.09 & 0.11 & 0.14 & 0.25 & 0.47 \\ 0.12 & 0.14 & 0.16 & 0.18 & 0.23 & 0.34 & 0.58 \end{bmatrix}.$$

In fig.9 the functions $\overline{\varphi}(t), \varphi(t), \varepsilon(t), \hat{v}_a(t), \hat{\delta}_e(t), \delta_e(t)$ and $v(t)$ ($\overline{\varphi}, \varepsilon, \hat{\delta}_e$ – with blue color, continuous line and $\varphi, \hat{v}_a, \delta_e$ with red color, dashed line) are presented.

If the actuator is non-linear one obtains the characteristics from fig.10; additionally, characteristics $v_h(t)$ and $\dot{\varphi}(\varphi)$ appear. When $v_h = 0$ the actuator is in the saturation state and it works in the linear zone when $v_h \neq 0$. The characteristic $\dot{\varphi}(\varphi)$ (phase portrait of the system) shows that the non-linear system tends to a stable limit cycle.

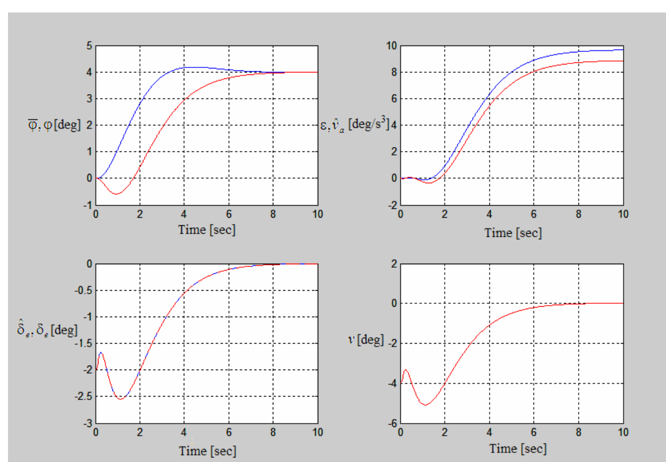


Fig.9. Time characteristics in the case of linear actuator’s use

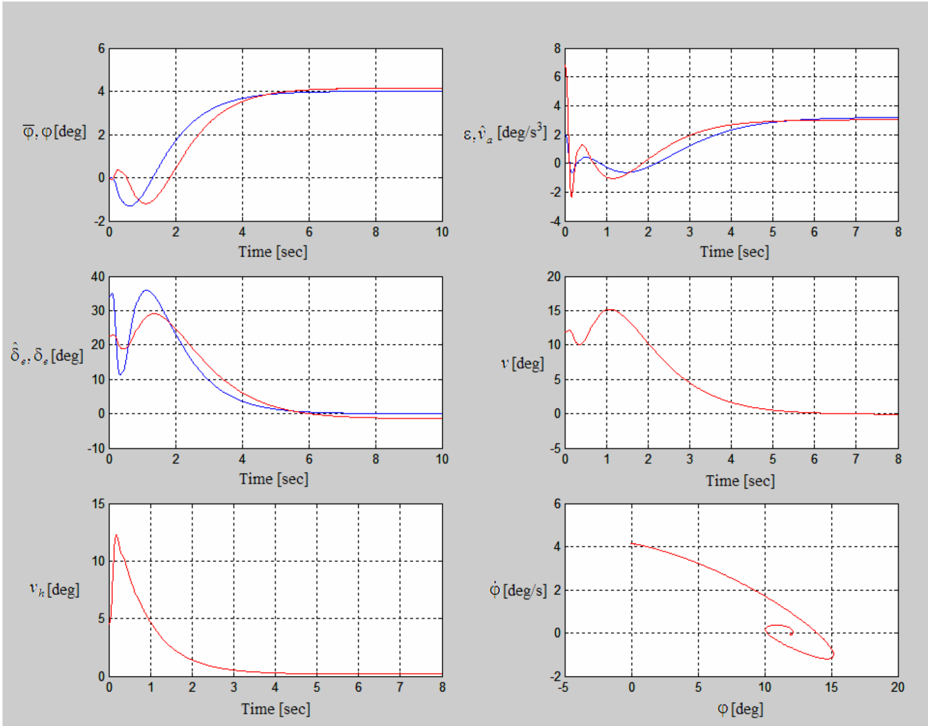


Fig.10. Time characteristics in the case of non-linear actuator's use

## IV. CONCLUSION

The aim of the adaptive command is to compensate the dynamic inversion error. Thus, the command law has two components: the command given by the linear dynamic compensator and the adaptive command given by the neural network. As control system one chooses the non-linear model of aircrafts' dynamics in longitudinal plain. The reference model is linear. One obtains the structure of the adaptive control system of the roll angle and Matlab/Simulink models of the adaptive command system's subsystems. Using these, some characteristics families are obtained; these describe the adaptive command system's dynamics with linear or non-linear actuator. The system is a stable one and has very good dynamic characteristics.

## REFERENCES


[1] A.J. Calise, "Flight Evaluation of an Adaptive Velocity Command System for Unmanned Helicopters". *AIAA Guidance, Navigation and Control Conference and Exhibit*, vol. 2, 11–14 August, 2003, Austin, Texas.

[2] S.M. Hosemi, M. Farrokhi, A.J. Kashkonei, "Robust Adaptive Control Systems Using Neural Networks". *The International Journal of Control*, vol. 3, 2006, 7 pp.

[3] E.N. Johnson, A.J. Calise, *Adaptive Guidance and Control for Autonomous Launch Vehicles.* IEEE Aerospace Conference, Biy Ykg, MT, April, 2001, 13 pp.

[4] M. Lungu, *Flyght control systems.* Sitech Publisher, Craiova, 2008, 329 pp.

[5] M. Lungu, R. Lungu, *Non-linear Adaptive Controllers with Linear Dynamic Compensator and Neural Network.* Proceedings of the 11th WSEAS International Conference on Automatic Control, Modelling And Simulation (ACMOS '09), Istanbul, Turkey, May 30 - June 1, 2009, pp. 334 – 339.

[6] A.J. Calise, N. Hovakymyan, M. Idan," Adaptive Output Control of Nonlinear Systems Using Neural Networks". *Automatica*, 37 (8), August, 2001, pp. 1201 – 1211.

[7] L.P. Gregory, *Adaptive Inverse Control of Plants with Disturbances*. Stanford University, 1998.

[8] N. Hovakimyan, F. Nardi, N. Kim, A.J. Calise," Adaptive Output Feedback Control of Uncertain Systems Using Single Hidden Layer Neural Networks". *IEEE Transactions on Neural Networks,* 13 (6), 2002, 22 pp.

[9] M. Sharma, A.J. Calise, *Adaptive Trajectory Control for Autonomous Helicopters.* AIAA Guidance, Navigation and Control Conference, 14 – 17 August, 2001, Montreal, Canada, 9 pp.

[10] N. Hovakimyan, N. Kim, A.J. Calise, J.V. Parasad, *Adaptive Output Feedback for High – Bandwidth Control of an Unmanned Helicopter.* AIAA Guidance, Navigation and Control Conference, 6 – 9 August, Montreal, Canada, 11 pp.

[11] A.J. Calise, Y. Shin, M.D. Johnson, *Comparison Study of Conventional and Neural Network Based Adaptive Control of Wing Rock.* AIAA Guidance, Navigation and Control Conference, Providence, RI, August, 2004. In Proceedings, 11 pp.