NEURO – ADAPTIVE COMMAND SYSTEMS FOR ROCKETS' MOVE

Romulus LUNGU, Mihai LUNGU

Avionics Department, University of Craiova, Faculty of Electrotechnics, Blv. Decebal, No.107, Craiova, Dolj, ROMANIA
romulus_lungu@yahoo.com, Lma1312@yahoo.com

Abstract **– Mathematical models of the move of the rockets with big attack angles, command laws and automat neuro – adaptive systems are presented. Using such structures, one studies longitudinal move of an agile rocket. Matlab/Simulink model is also obtained. A lot of graphical characteristics are presented in this paper.**

Keywords: *neural network, adaptive, rocket, move, longitudinal*

1. MATHEMATICAL MODELS

The model of the rocket's move is the one from [1]

$$\begin{aligned}
&\dot{V} = (a_x \cos\alpha + a_z \sin\alpha)\cos\beta + a_y \sin\beta,\\
&\dot{\alpha} = \omega_y - (\omega_x \cos\alpha + \omega_z \sin\alpha)\mathrm{tg}\beta + \frac{-a_x \sin\alpha + a_z \cos\alpha}{V\cos\beta},\\
&\dot{\beta} = \omega_x \sin\alpha - \omega_z \cos\alpha - \frac{(a_x \cos\alpha + a_z \sin\alpha)\sin\beta - a_y \cos\beta}{V}.
\end{aligned} \tag{1}$$

where V is the flight velocity, α – attack angle, β – sideslip angle.

To these one adds the moments' equilibrium equations

$$\begin{aligned}
&\dot{\omega}_x = \frac{M_x}{J_{xx}},\\
&\dot{\omega}_y = \frac{M_y}{J_{yy}} + \left(1 - \frac{J_{xx}}{J_{yy}}\right)\omega_x\omega_z,\\
&\dot{\omega}_z = \frac{M_z}{J_{zz}} - \left(1 - \frac{J_{xx}}{J_{yy}}\right)\omega_x\omega_y,
\end{aligned} \tag{2}$$

where M_x, M_y, M_z are the aerodynamic moments which operate round ox, oy, oz axes.

$$\begin{aligned}
&M_x = M_x^{\beta}\beta + M_x^{\omega_x}\omega_x + M_x^{\omega_z}\omega_z + M_x^{\delta_e}\delta_e + M_x^{\delta_d}\delta_d,\\
&M_y = M_y^{\alpha}\alpha + M_y^{\omega_y}\omega_y + M_y^{\delta_p}\delta_p,\\
&M_z = M_z^{\beta}\beta + M_z^{\omega_x}\omega_x + M_z^{\omega_z}\omega_z + M_z^{\delta_e}\delta_e + M_z^{\delta_d}\delta_d;
\end{aligned} \tag{3}$$

For a very good control of the agile air – air rockets' inclination, in [2] and [3] one introduces an aerodynamic roll angle is used; it verifies equation

$$\dot{\gamma} = \frac{\omega_x \cos\alpha + \omega_z \sin\alpha}{\cos\beta} + \frac{a_x \sin\alpha - a_z \cos\alpha}{V}\mathrm{tg}\beta. \tag{4}$$

The angular variables are grouped in vectors

$$\begin{aligned}
&x^T = [\alpha \quad \beta \quad \gamma],\\
&\omega^T = [\omega_x \quad \omega_y \quad \omega_z].
\end{aligned} \tag{5}$$

The second and the third equation (1) and equation (4) may be expressed under the vectorial form

$$\dot{x} = T(x)\omega + a_f, \tag{6}$$

where

$$T(x) = \begin{bmatrix} -\cos\alpha\mathrm{tg}\beta & 1 & -\sin\alpha\mathrm{tg}\beta \\ \sin\alpha & 0 & -\cos\alpha \\ \cos\alpha/\cos\beta & 0 & \sin\alpha/\cos\beta \end{bmatrix},$$
$$a_f = \begin{bmatrix} \dfrac{-a_x \sin\alpha + a_z \cos\alpha}{V\cos\beta} \\ -\dfrac{(a_x \cos\alpha + a_z \sin\alpha)\sin\beta - a_y \cos\beta}{V} \\ \dfrac{a_x \sin\alpha - a_z \cos\alpha}{V}\mathrm{tg}\beta \end{bmatrix}. \tag{7}$$

2. NEURO – ADAPTIVE COMMAND STRUCTURES

Equation (6) is equivalent with the following equations' system, in which a component u_x of the pseudo-command is distinguished [4]

$$\begin{aligned}
&\dot{x} = u_x,\\
&u_x = T(x)\omega + a_f.
\end{aligned} \tag{8}$$

Similarly, equation system (2) may be described by equations in which another component u_ω of the pseudo-command is distinguished

$$\begin{aligned}
&\dot{\omega} = u_\omega, u_\omega = f(z,\omega,\delta),\\
&\delta^T = [\delta_e \quad \delta_p \quad \delta_d].
\end{aligned} \tag{9}$$

By noting with $\tilde{x}$ the leading command (vector $\bar{x}$ is the imposed one), $\tilde{x} = \bar{x} - x$, pseudo – command u_c may have the form

$$u_x = K_x \tilde{x}, \tag{10}$$

where K_x is the feedback coefficients' matrix (positive defined).
Conform to equation (8), one may calculate angular velocity of the rocket (ω_c) using an dynamic inversion exterior loop

$$\omega_c = T^{-1}(x)(u_x - a_f), \tag{11}$$

Pseudo – command u_ω may be chosen of form

$$u_\omega = K_\omega \tilde{\omega} - u_a = u_c - u_a, \tag{12}$$

where K_ω is the feedback coefficients' matrix (positive defined), $\tilde{\omega} = \omega_c - \omega$ and u_a – adaptive command for inversion error's compensation. The command vector may be calculated with formula [1]

$$\delta_c = G_c^{-1}(u_\omega - F_c) = \hat{f}^{-1}(x, \omega, u_\omega), \tag{13}$$

or with the following equation

$$\delta_c = G_c^{-1}(K_\omega \tilde{\omega} - u_a - F_c); \tag{14}$$

F_c is calculated with the following equation

$$F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \dfrac{M_x^\beta \beta + M_x^{\omega x}\omega_x + M_x^{\omega z}\omega_z}{J_{xx}} \\ \dfrac{M_y^\alpha \alpha + M_x^{\omega y}\omega_y}{J_{yy}} + \left(1 - \dfrac{J_{xx}}{J_{yy}}\right)\omega_x\omega_z \\ \dfrac{M_z^\beta \beta + M_z^{\omega x}\omega_x + M_z^{\omega z}\omega_z}{J_{yy}} + \left(1 - \dfrac{J_{xx}}{J_{zz}}\right)\omega_x\omega_y \end{bmatrix}. \tag{15}$$

Another control structure may be obtained using stability theory with Liapunov functions if the controlled object (A) may be described by the non – linear equations system [5]

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + h_1(x_1)x_2, \\ \dot{x}_2 &= f_2(x_1, x_2, u), \end{aligned} \tag{16}$$

In figure 2 one presents the block diagram for above equations' modeling.

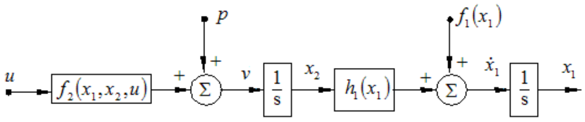


Figure 2: Block diagram for equations (16)

Let's consider now the following case: sub – system described by function f_1 is well known, while the one described by f_2 is aproximately known; that's why the command must be after variable x_2. The imposed vector $\bar{x}_2$ is

$$\bar{x}_2 = q_1(\tilde{x}_1, t). \tag{17}$$

This law must assure the stability of the variable $\tilde{x}_1$ in rapport with variable z (figure 4);

$$\tilde{x}_2 = \bar{x}_2 - x_2 = q_1(\tilde{x}_1, t) - x_2. \tag{18}$$

The second sub-system (described by the second equation (16)), due to the lack of the external disturbances $(p(t) = 0)$, may be described by equation

$$\begin{aligned} \dot{x}_2 &= v, \\ v &= f_2(x_1, x_2, u), \end{aligned} \tag{19}$$

where input v is a pseudo-command. If the function f_2 is invertible than the dynamic inversion of f_2 may be approximately done; $u = f_2^{-1}(x_1, x_2, v)$.
If f_2 is known than $f_2^{-1} f_2 = 1$ and if it is approximately known than the inversion of function f_2 is made with error $\varepsilon(x_1, x_2, u)$ and the first equation (19) becomes

$$\dot{x}_2 = v + \varepsilon(x_1, x_2, u) + p, \tag{20}$$

where ε has the form

$$\varepsilon(x_1, x_2, u) = f_2(x_1, x_2, u) - \hat{f}_2(x_1, x_2, u) = \varepsilon(\tilde{x}_1, \tilde{x}_2, \bar{x}_1, \dot{\bar{x}}_1, v), \tag{21}$$

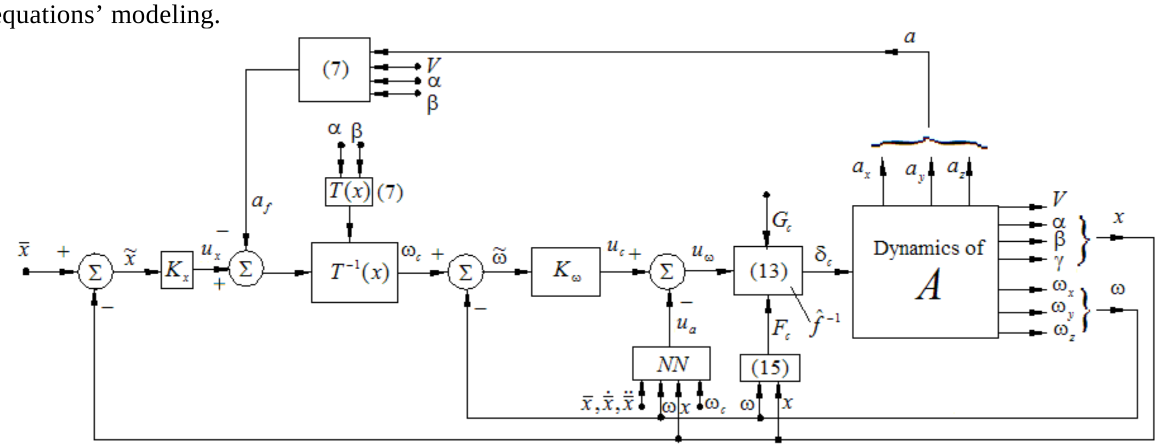


Figure 1: Adaptive command system based on dynamic inversion (variant 1)

with $\hat{f}_2 -$ calculated function.

The command law may be chosen [5]

$$v = u_c + \dot{\bar{x}}_2 + \bar{v} - u_a = K_2\tilde{x}_2 + \dot{\bar{x}}_2 + \bar{v} - u_a, \qquad (22)$$

where $u_c -$ the command in case $f_2^{-1} f_2 = 1, K_2 -$ positive defined matrix and $u_a -$ adaptive command for the inversion error compensation ε, obtained from the Sigma neural network;

$$u_a = W^T \sigma\left(V^T I\right), \qquad (23)$$

with $\sigma -$ the activation function of the hidden layer (2), $I -$ the input vector,

$$W^T = \begin{bmatrix} b_i & w_{ij} \end{bmatrix}, V^T = \begin{bmatrix} c_i & v_{ij} \end{bmatrix}, \qquad (24)$$

b_i and $c_i -$ bias, $w_{ij} -$ the weights of connections between level 1 and 2, $v_{ij} -$ the weights of connections between level 2 and 3. The structure of such a neural network is presented in figure 3.

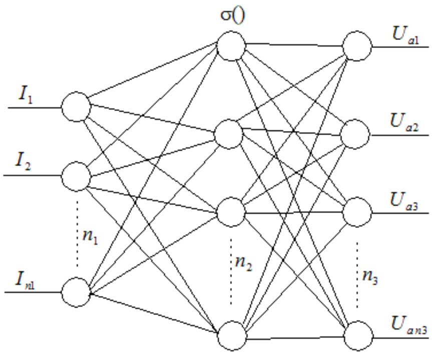


Figure 3: Neural network's structure

Learning rule is obtained using Liapunov stability theory as below [2]. Considering Frobenius norm of matrix A

$$\|A\|_F^2 = \mathrm{tr}\{A^T A\}, \qquad (25)$$

introducing the compact matrix

$$Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix}, \qquad (26)$$

with $\|Z\|_F \le \bar{Z}$, choosing the input vector of the neuronal network

$$I^T = \begin{bmatrix} 1 & \tilde{x}_1^T & \tilde{x}_2^T & \bar{x}_1^T & \dot{\bar{x}}_1^T & \ddot{\bar{x}}_1^T & u_a^T & \|Z\|_F \end{bmatrix} \qquad (27)$$

and standard Liapunov function

$$V_l = \frac{1}{2}\tilde{x}_2^T \tilde{x}_2 + \frac{1}{2}\mathrm{tr}\left(W^T \Gamma_W^{-1} W\right) + \frac{1}{2}\mathrm{tr}\left(V^T \Gamma_V^{-1} V\right), \qquad (28)$$

from stability analysis one obtains the term $\bar{v}$ from (22)

$$\bar{v} = K_z\left(\|Z\|_F + \bar{Z}\right)\left(\|\tilde{x}_1\| + \|\tilde{x}_2\|\right)e_2, \qquad (29)$$

where $K_z > 0$ and $e_2 = \tilde{x}_2 / \|\tilde{x}_2\|$.

The control system structure (PA-A) is presented in figure 4. In particular, system (16) represents non linear model of an aircraft (A), which may be, for example, an air – air rocket. Thus,

$$\begin{aligned} x_1 &= x = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T, \\ x_2 &= \omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T; \end{aligned} \qquad (30)$$

from the equivalence of equations (16) and (6) it results

$$h_1(x_1) = T(x),\ f_1(x_1) = a_f, \qquad (31)$$

with $T(x)$ and a_f of forms (7).

The second equation (16) is equivalent with the equations' system (8), where $x_1 = x, x_2 = \omega, u = \delta$ and

$$f_2(x_1, x_2, u) = f(x, \omega, \delta), \delta = \begin{bmatrix} \delta_e & \delta_p & \delta_d \end{bmatrix}^T, \qquad (32)$$

which may be separated in two components. Equation (19) is equivalent with equations' system (9), where $x_2 = \omega, f_2 = f$ and $v = u_\omega$.

Comparing equations (22) and (12) one results that matrix K_2 plays the role of matrix K_ω and $\tilde{x}_2 -$ the role of $\tilde{\omega} = \omega_c - \omega$. Thus, $\bar{x}_2$ plays the role of ω_c.

Comparing now equations (17) with (11) one obtains

$$q_1(\tilde{x}_1, t) = T^{-1}(x)\left(u_x - a_f\right), \qquad (33)$$

with pseudo – command u_x of form (10).

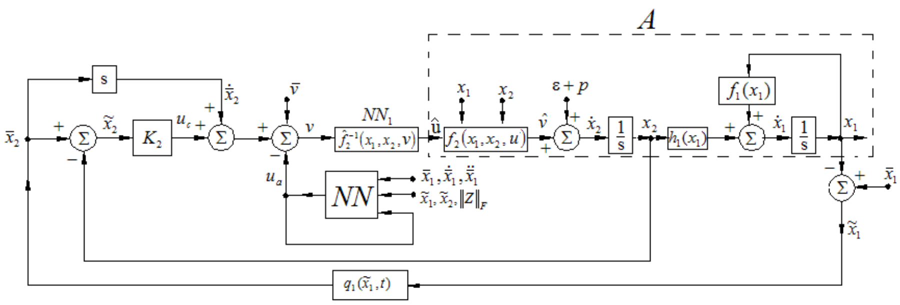


Figure 4: Adaptive command system based on dynamic inversion (variant 2)

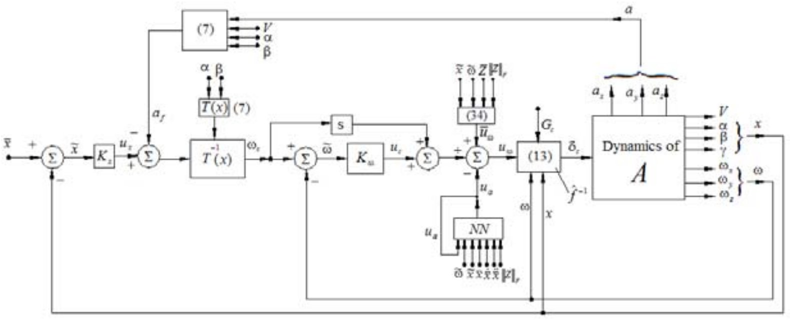


Figure 5: Adaptive command system based on dynamic inversion (variant 3)

Considering the inputs of the neural network from figure 4, one obtained block diagram from figure 5, where the role of $\bar{v}$ from (29) is played by $\bar{u}_\omega$; (29) becomes

$$\bar{u}_\omega = K_z\left(\|Z\|_F + \bar{Z}\right)\left(\|\tilde{x}\| + \|\tilde{\omega}\|\right)e_2\,, \qquad (34)$$

where $e_2 = \tilde{\omega}/\|\tilde{\omega}\|$.

3. NEURO – ADAPTIVE COMMAND FOR THE ROCKET'S MOVE

Let's consider now the case of a rocket's longitudinal movement described by equations [5]

$$\begin{aligned} &\dot{\alpha} = -\left(a_1 + a_2\alpha^2\right)\alpha + \omega, \\ &\dot{\omega} = \left(c_1 + c_2\alpha^2\right)\alpha + \left(c_3 + c_4\alpha^2\right)\delta, \\ &\delta = \frac{1}{\tau s + 1}u, \end{aligned} \qquad (35)$$

where $\begin{aligned} &a_1 = 1.02, a_2 = 1.3, c_1 = -57.2, c_2 = -322.2, \\ &c_3 = -70.15, c_4 = -360.25, \delta = 0.1s. \end{aligned}$

Block diagram of the closed loop system (PA - A) is presented in figure 6; it has been obtained using diagram blocks from figure 4 and figure 4.

By identification of system (35) with system (6) and of system (9) with (16) one obtains

$$\begin{aligned} &x_1 = x = \alpha, \bar{x} = \bar{\alpha}, x_2 = \omega_y = \omega, \bar{x}_2 = \omega_c = \bar{\omega}, \\ &T(x) = h_1(x_1) = 1, a_f = f_1(x_1) = -\left(a_1 + a_2\alpha^2\right)\alpha, \\ &f(x,\omega,\delta) = f_2(x_1, x_2, \delta) = \left(c_1 + c_2\alpha^2\right)\alpha + \left(c_3 + c_4\alpha^2\right)\delta, \\ &\bar{u}_\omega = \bar{v}, u_\omega = v = u. \end{aligned} \qquad (36)$$

The values of the other parameters from fig.5 are: $\bar{Z} = 50, K_z = 0.6$.

For the calculus of coefficients k_x and k_ω, one keeps only the linear part of the system from fig.5 ($a_2 = c_2 = c_4 = 0$). Closed loop transfer function $\left(H_0(s) = \dfrac{\alpha(s)}{\bar{\alpha}(s)}\right)$ is calculated and it's expressed as follows

$$\begin{aligned} &H_0(s) = \frac{c_3 k_x k_\omega}{s^2 + \left(a_1 + c_3 k_\omega\right)s + \left(c_3 k_\omega - c_1 - a_1 c_3 k_\omega + c_3 k_x k_\omega\right)} \\ &\Leftrightarrow H_0(s) = \frac{\alpha(s)}{\bar{\alpha}(s)} = \frac{c_3 k_x k_\omega}{s^2 + 2\xi\omega_0 + \omega_0^2}. \end{aligned} \qquad (37)$$

By setting $\zeta = 0{,}707$ and $\omega_0 = 5$, the two coefficients have expressions

$$\begin{aligned} &k_\omega = \frac{2\xi\omega_0 - a_1}{c_3}, \\ &k_x = \frac{\omega_0^2 - c_3 k_\omega + c_1 + a_1 c_3 k_\omega}{c_3 k_\omega}. \end{aligned} \qquad (38)$$

One chooses a feed-forward neural network with 8

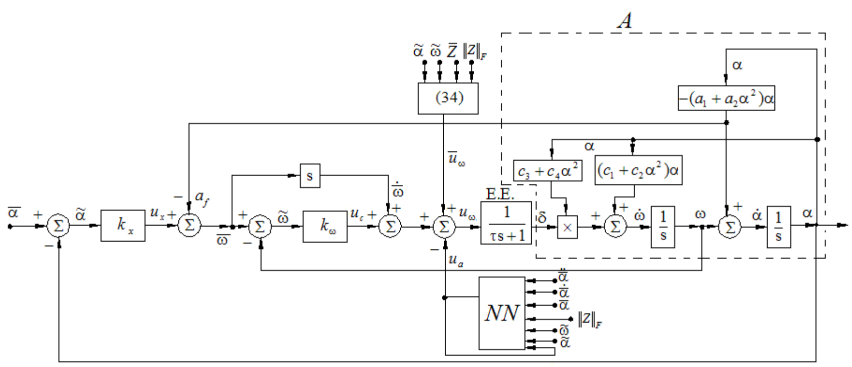


Figure 6: Neuro – adaptive command system for the rocket's move in vertical plane using model (35)

input neurons, a neuron on the hidden layer and an output neuron. Activation function for the hidden layer neuron is a linear one, while activation function of the neurons from the input layer has a **tansig** form (tangent hyperbolic)

$$\tan\mathrm{sig}(n) = \frac{2}{1+\exp(-2n)} - 1. \qquad (39)$$

The neural network's output (u_a) has the form

$$u_a = W^T \mathrm{tansig}\left(V^T I\right), \qquad (40)$$

where V is the weights' vector of the input neurons, W is the weights' vector of the hidden layer neurons, $I-$ inputs vector

$$I^T = \left[1 \quad \tilde{\alpha}^T \quad \tilde{\omega}^T \quad \overline{\alpha}^T \quad \dot{\tilde{\alpha}}^T \quad \ddot{\tilde{\alpha}}^T \quad U_a^T \quad \|Z\|_F\right],$$
$$V^T = [1 \quad 7 \quad -3 \quad 2 \quad -1 \quad -2 \quad 3 \quad 1], \mathrm{W} = [1]. \qquad (41)$$

Neglecting terms u_a and $\overline{u}_\omega$, the indicial response (fig.6) proves stabilization of angle α to its imposed value $(\overline{\alpha} = 1\mathrm{grd})$.

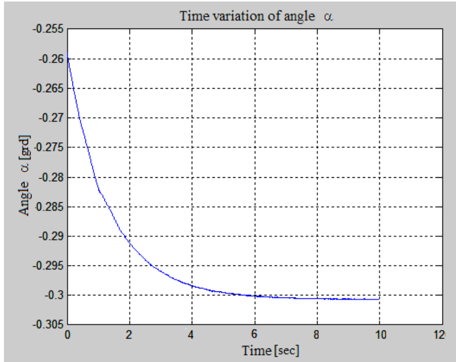


Figure 7: Time variation of the rocket's attack angle (without neural network)

Using Matlab/Simulink model from figure 8 one obtains $\alpha(t)$ (fig.9.a) and error $\tilde{\alpha}(t) = \overline{\alpha}(t) - \alpha(t)$ (fig.9.b).

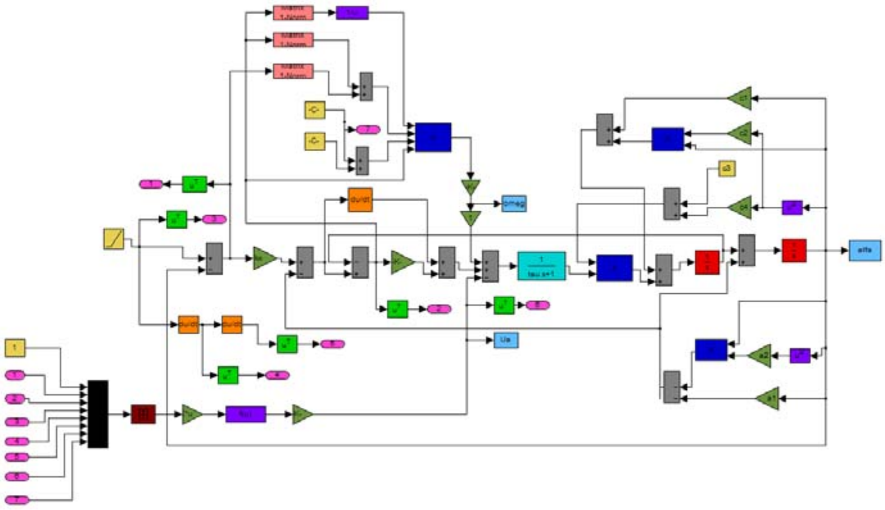
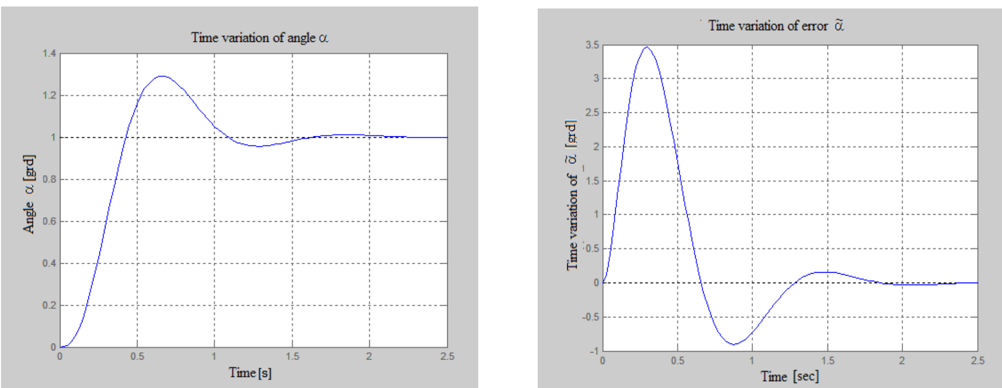


Figure 8: Matlab/Simulink model of the system from figure 6



a. b.

Figure 9: Dynamics of attack angle α and error $\tilde{\alpha}$ for the system from fig.5 (with neural network)

4. CONCLUSIONS

This paper presents nonlinear mathematical models of the rockets' move with big attack angles, laws and automat adaptive command structures using neural networks. One studies a neuro – adaptive command system for the agile rocket's move.

References

[1] R. Lungu, M. Lungu, N. Jula, C. Cepisca, *Stabilization Models and Structures for Move of very Maneuverable Flying Objects.* The 6th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE '07), Venice, Italy, November 21-23, 2007, pp. 135 – 138.

[2] A. Isidori, *Nonlinear Control Systems.* Springher, Berlin, 1995.

[3] M.B. McFarland, A.J. Calise, *Neural Networks for Stable Adaptive Control of Air – to – Air Missiles.* Proceedings of the AIAA Guidance, Navigation and Control Conference, Baltimore, Maryland, 1995, pp. 871 – 879.

[4] M.B. McFarland, A.J. Calise, *Neural – Adaptive Nonlinear Autopilot Design for Agile Anti – Air Missile.* Proceedings of the Guidance, Navigation and Control Conference, San Diego, California, July, 1996, pp. 21 – 28.

[5] M.B. McFarland, A.J. Calise, *Robust Adaptive Control of Uncertain Nonlinear Systems Using Neural Networks.* Proceedings of the American Control Conference, Albuquerque, New Mexico, June, 1997.