Neuro - Adaptive Command Systems for very maneuverable flying objects

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

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

JULA NICOLAE
Military Technical Academy of Bucharest
Blv. George Coşbuc, no. 81 – 83, Sector 5
ROMANIA
nicolae.jula@gmail.com

CEPISCA COSTIN
Electrical Engineering Faculty
University POLITEHNICA of Bucharest
ROMANIA
costin@wing.ro

Abstract: - One presents a dynamic model of the very maneuverable flying objects (A), which expresses the dependence between the vector formed by angles of A regarding aerodynamic trihedron and the vector of angular velocities of A or the vector of linear acceleration components in rapport with the trihedron related to A. Control structures (stabilization of movement) are also presented. These consist of loops after angles, angular velocities and linear accelerations and an adaptive control loop with neuronal network for dynamic inversion errors compensation of the non-linear function which describes unknown system of the dynamic model of A. Adaptive command is projected upon stability theory using Liapunov functions [1], [2], [3]. As calculus examples one presented two systems and stabilization models of the very maneuverable rocket's longitudinal move [4]. Time evolution of the attack angle for the studied cases is also presented.

Key-Words: - dynamic model, adaptive command, neuronal network, rocket, maneuverable.

1 Introduction

The A's movement control takes into account the values of the A's angles in rapport with the aerodynamic trihedron and angular velocities and accelerations sensors utilization (placed on trihedron axis related to A). Dynamic model is made by two sub-systems: one of them is described by a well known non-linear function (f_1) and the other is described by a proximate known or unknown non-linear function (f_2). Control law synthesis is based on dynamic inversion (the f_2 inversion).

The control law has components expressed as functions of state variables and an adaptive component. This is obtained with a neuronal network with the role of f_2 inversion error compensation.

The control and stabilization of A's movement in non-linear description are closer to real flight conditions than the linear variants. The learning capacity of the neuronal networks in control of the non-linear systems is taken into account.

2 Spatial movement models of the flying objects

The following equations express dependences between linear accelerations a_x, a_y, a_z and angular velocities $\omega_x, \omega_y, \omega_z$ in rapport with trihedron related to flying machine A. These variables are available because of the accelerometers and gyro meters. Let $oxyz$ be the trihedron related to A with $ox-$ the longitudinal axis, $oy-$ the lateral axis and oz rectangular to ox and oy and $ox_a y_a z_a -$ aerodynamic trihedron; V is the flying velocity, $\alpha-$ attack angle, $\beta-$ side-slip angle (fig.1). For $ox_a y_a z_a$ and $oxyz$ superpose the following coordinates transformations are made

$$ox_a y_a z_a \xrightarrow[\dot{\beta}(oz_a)]{\beta} ox'_a y z_a \xrightarrow[\dot{\alpha}(oy)]{\alpha} oxyz. \quad (1)$$

Acceleration $\vec{a}$ is expressed with formula

$$\vec{a} = \dot{\vec{V}} + \vec{\omega} \times \vec{V}, \quad (2)$$

with

$$\vec{a} = \vec{a}_x + \vec{a}_y + \vec{a}_z, \quad (3)$$

$$\vec{\omega} = \vec{\omega}_x + \vec{\omega}_y + \vec{\omega}_z - \left(\dot{\vec{\alpha}} + \dot{\vec{\beta}}\right). \qquad (4)$$

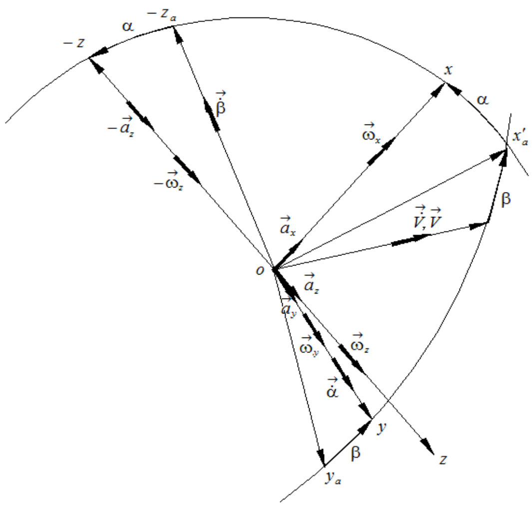


Fig.1 The trihedron related to A and the aerodynamic trihedron

From equation (2) one obtains

$$\vec{a}_x + \vec{a}_y + \vec{a}_y = \dot{\vec{V}} + \left(\vec{\omega}_x \times \vec{V}\right) + \left(\vec{\omega}_y - \dot{\vec{\alpha}}\right) \times \vec{V} + \left(\vec{\omega}_z - \dot{\vec{\beta}}\right) \times \vec{V}. \qquad (5)$$

Through projection on $ox_a y_a z_a$'s axes one obtains

$$\begin{aligned}
&\left[a_x \cos\alpha - a_z \sin\left(90^\circ + \alpha\right)\right]\cos\beta + a_y \sin\beta = \dot{V}, \\
&\left[a_x \cos\alpha - a_z \sin\left(90^\circ + \alpha\right)\right]\sin\left(90^\circ + \beta\right) + \\
&\quad + a_y \cos\beta + \left[\omega_x \sin\alpha - \omega_z \cos\alpha + \dot{\beta}\right]V = 0, \\
&-\left[\omega_x \cos\alpha - \omega_z \sin\left(90^\circ + \alpha\right)\right]V \sin\beta - \left(\omega_y - \dot{\alpha}\right)\cos\beta \cdot V - \\
&\quad - \left(a_x \sin\alpha - a_z \cos\alpha\right) = .
\end{aligned} \qquad (6)$$

$$\begin{aligned}
&\dot{V} = \left(a_x \cos\alpha + a_z \sin\alpha\right)\cos\beta + a_y \sin\beta, \\
&\dot{\alpha} = \omega_y - \left(\omega_x \cos\alpha + \omega_z \sin\alpha\right)\mathrm{tg}\beta + \frac{-a_x \sin\alpha + a_z \cos\alpha}{V\cos\beta}, \\
&\dot{\beta} = \omega_x \sin\alpha - \omega_z \cos\alpha - \frac{\left(a_x \cos\alpha + a_z \sin\alpha\right)\sin\beta - a_y \cos\beta}{V}.
\end{aligned} \qquad (7)$$

To these one adds the moments' equilibrium equations

$$\begin{aligned}
&\dot{\omega}_x = \frac{M_x}{J_{xx}}, \\
&\dot{\omega}_y = \frac{M_y}{J_{yy}} + \left(1 - \frac{J_{xx}}{J_{yy}}\right)\omega_x \omega_z, \\
&\dot{\omega}_z = \frac{M_z}{J_{zz}} - \left(1 - \frac{J_{xx}}{J_{yy}}\right)\omega_x \omega_y,
\end{aligned} \qquad (8)$$

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

$$\begin{aligned}
&M_x = M_x^\beta \beta + M_x^{\omega x}\omega_x + M_x^{\omega z}\omega_z + M_x^{\delta e}\delta_e + M_x^{\delta d}\delta_d, \\
&M_y = M_y^\alpha \alpha + M_y^{\omega y}\omega_y + M_y^{\delta p}\delta_p, \\
&M_z = M_z^\beta \beta + M_z^{\omega x}\omega_x + M_z^{\omega z}\omega_z + M_z^{\delta e}\delta_e + M_z^{\delta d}\delta_d;
\end{aligned} \qquad (9)$$

the coefficients of the angular variables represent variation speeds (slopes) of the aerodynamic moments regarding to respective angular variables (stability derivates).

Equations (7) and (8) are used especially in the case of very maneuverable aircrafts and in the case of agile rockets with big attack and side-slip angles. For a very good control of the agile air – air rockets' inclination, in [5] and [6] an aerodynamic roll angle is used; it verifies equation

$$\dot{\gamma} = \frac{\omega_x \cos\alpha + \omega_z \sin\alpha}{\cos\beta} + \frac{a_x \sin\alpha - a_z \cos\alpha}{V}\mathrm{tg}\beta \qquad (10)$$

and the angular variables are grouped in the vectors

$$x^T = \left[\alpha \quad \beta \quad \gamma\right], \ \omega^T = \left[\omega_x \quad \omega_y \quad \omega_z\right]. \qquad (11)$$

The second and the third equation (7) and equation (10) may be expressed under the vectorial form

$$\dot{x} = T(x)\omega + a_f, \qquad (12)$$

where

$$T(x) = \begin{bmatrix} -\cos\alpha\mathrm{tg}\beta & 1 & -\sin\alpha\mathrm{tg}\beta \\ \sin\alpha & 0 & -\cos\alpha \\ \cos\alpha/\cos\beta & 0 & \sin\alpha/\cos\beta \end{bmatrix},$$

$$a_f = \begin{bmatrix} \dfrac{-a_x \sin\alpha + a_z \cos\alpha}{V\cos\beta} \\ -\dfrac{\left(a_x \cos\alpha + a_z \sin\alpha\right)\sin\beta - a_y \cos\beta}{V} \\ \dfrac{a_x \sin\alpha - a_z \cos\alpha}{V}\mathrm{tg}\beta \end{bmatrix}. \qquad (13)$$

Equation (12) is equivalent with the following equations' system, in which a component u_x of the pseudo-command is distinguished [7]

$$\dot{x} = u_x, \ u_x = T(x)\omega + a_f. \qquad (14)$$

Similarly, equation system (8) may be described by equations in which another component u_ω of the pseudo-command is distinguished

$$\begin{aligned}
&\dot{\omega} = u_\omega, \ u_\omega = f(z, \omega, \delta), \\
&\delta^T = \left[\delta_e \quad \delta_p \quad \delta_d\right].
\end{aligned} \qquad (15)$$

Function f has two components, as we can see from (8) and (9)

$$u_\omega = f(x, \omega, \delta) = F(x, \omega) + G \cdot \delta; \qquad (16)$$

$$F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \dfrac{M_x^\beta \beta + M_x^{\omega x}\omega_x + M_x^{\omega z}\omega_z}{J_{xx}} \\ \dfrac{M_y^\alpha \alpha + M_x^{\omega y}\omega_y}{J_{yy}} + \left(1 - \dfrac{J_{xx}}{J_{yy}}\right)\omega_x \omega_z \\ \dfrac{M_z^\beta \beta + M_z^{\omega x}\omega_x + M_z^{\omega z}\omega_z}{J_{yy}} + \left(1 - \dfrac{J_{xx}}{J_{zz}}\right)\omega_x \omega_y \end{bmatrix}, \qquad (17)$$

$$G = \begin{bmatrix} \dfrac{M_x^{\delta e}}{J_{xx}} & 0 & \dfrac{M_x^{\delta d}\delta_d}{J_{xx}} \\ 0 & \dfrac{M_y^{\delta p}\delta_p}{J_{yy}} & 0 \\ \dfrac{M_z^{\delta e}}{J_y} & 0 & \dfrac{M_z^{\delta e}}{J_y} \end{bmatrix}. \qquad (18)$$

In the particular case of longitudinal move

$(\omega_x = \omega_z = \beta = 0)$ equations (7) and (8) becomes

$$\dot{\alpha} = \omega_y + \frac{a_x \sin\alpha - a_z \cos\alpha}{V\cos\beta}, \dot{\omega}_y = \frac{M_y}{J_{yy}}, \quad (19)$$

where

$$M_y = M_y^{\alpha}\alpha + M_y^{\omega_y}\omega_y + M_y^{\delta_p}\delta_p. \quad (20)$$

As a consequence $x = \alpha$, $\omega = \omega_y$, $T(x) = 1$ and $a_f = (a_x \sin\alpha - a_z \cos\alpha)/V\cos\beta$; equation (12) becomes

$$\dot{\alpha} = \omega_y + a_f, \quad (21)$$

and equations (14), (15) and (16) become

$$\dot{\alpha} = u_x, u_x = \omega_y + a_f; \quad (22)$$

$$\dot{\omega}_y = u_\omega, u_\omega = f(V, H, \alpha, \omega_y, \delta_p); \quad (23)$$

$$f(V, H, \alpha, \omega_y, \delta_p) = F(V, H, \alpha, \omega_y) + G \cdot \delta_p \quad (24)$$

with

$$F = F_y = \frac{M_y^{\alpha}\alpha + M_y^{\omega_y}\omega_y}{J_{yy}}, G = \frac{M_y^{\delta_p}}{J_{yy}}. \quad (25)$$

From (14) one results

$$\omega_c = T^{-1}(x)(u_x - a_f), \quad (26)$$

where u_x is the pseudo-command, which may be chosen

$$u_x = K_x\tilde{x}, \tilde{x} = \bar{x} - x, \quad (27)$$

with $\bar{x}-$ control command. From (16) one obtains

$$\delta_c = G_c^{-1}(u_\omega - F_c) = \hat{f}^{-1}(x, \omega, u_\omega), \quad (28)$$

with pseudo-command

$$u\omega = K_\omega\tilde{\omega} = u_c - u_a, \quad (29)$$

where $\tilde{\omega} = \omega_c - \omega$ and $u_a -$ the adaptive command for inversion error's compensation.

3 Stabilization structures for flying objects' movement

Control block diagram of the closed loop system is presented in fig.2.

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 [6], [8]

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + h_1(x_1)x_2, \\ \dot{x}_2 &= f_2(x_1, x_2, u), \end{aligned} \quad (30)$$

used by the system from fig.3.

The imposed vector $\bar{x}_2$ is

$$\bar{x}_2 = q_1(\tilde{x}_1, t). \quad (31)$$

This law must assure the stability of the variable $\tilde{x}_1$ in rapport with variable z (fig.3);

$$\tilde{x}_2 = \bar{x}_2 - x_2 = q_1(\tilde{x}_1, t) - x_2. \quad (32)$$

The second sub-system (described by the second equation (30)), due to the lack of the external disturbances, may be described by equation

$$\dot{x}_2 = v, v = f_2(x_1, x_2, u), \quad (33)$$

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 (33) becomes

$$\dot{x}_2 = v + \varepsilon(x_1, x_2, u) + p, \quad (34)$$

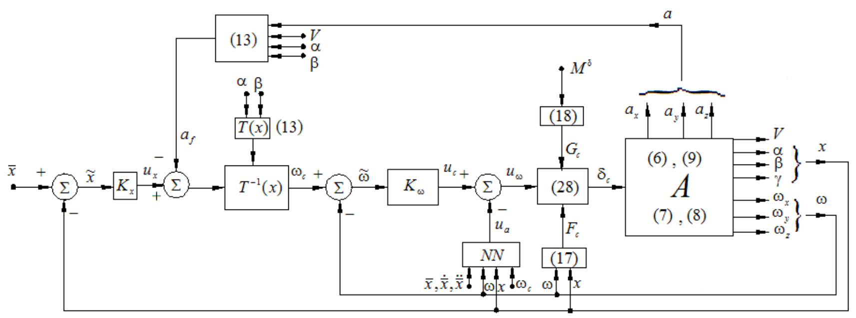


Fig.2 Block diagram of the stabilization system (variant 1)

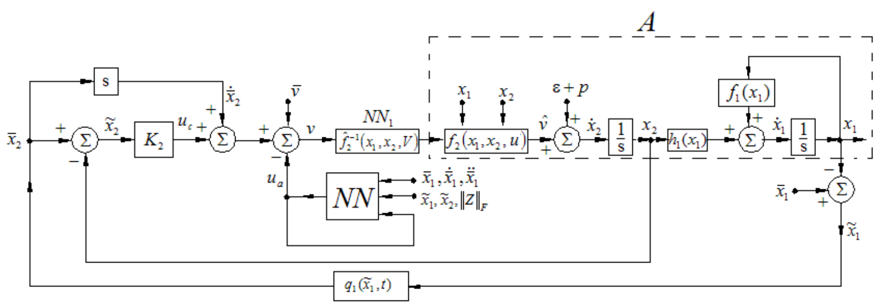


Fig.3 Block diagram of the stabilization system (variant 2)

where ε has the form

$$\varepsilon(x_1, x_2, u) = f_2(x_1, x_2, u) - \hat{f}_2(x_1, x_2, u) = \varepsilon(\tilde{x}_1, \tilde{x}_2, \bar{x}_1, \dot{\bar{x}}_1, v), \quad (35)$$

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

The command law may be chosen [6]

$$v = u_c + \dot{\bar{x}}_2 + \bar{v} - u_a = K_2\tilde{x}_2 + \dot{\bar{x}}_2 + \bar{v} - u_a, \quad (36)$$

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;

$$u_a = W^T \sigma(V^T I), \quad (37)$$

with σ – the activation function of the hidden layer (2), I – the input vector,

$$W^T = [b_i \quad w_{ij}], \qquad V^T = [c_i \quad v_{ij}], \quad (38)$$

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. Learning rule is obtained using stability theory of Liapunov [6]. Considering Frobenius norm of matrix A

$$\|A\|_F^2 = \mathrm{tr}\{A^T A\}, \quad (39)$$

introducing the compact matrix

$$Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix}, \quad (40)$$

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

$$I^T = [1 \quad \tilde{x}_1^T \quad \tilde{x}_2^T \quad \bar{x}_1^T \quad \dot{\bar{x}}_1^T \quad \ddot{\bar{x}}_1^T \quad u_a^T \quad \|Z\|_F] \quad (41)$$

and standard Liapunov function

$$V_l = \frac{1}{2}\tilde{x}_2^T\tilde{x}_2 + \frac{1}{2}\mathrm{tr}(W^T\Gamma_W^{-1}W) + \frac{1}{2}\mathrm{tr}(V^T\Gamma_V^{-1}V), \quad (42)$$

from stability analysis one obtains the term $\bar{v}$ from (36)

$$\bar{v} = K_z(\|Z\|_F + \bar{Z})(\|\tilde{x}_1\| + \|\tilde{x}_2\|)e_2, \quad (43)$$

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

The control system structure (PA-A) is presented in fig.3 (equivalent to the one from fig.4, where $\bar{v}$ is $\tilde{u}_\omega$).

4 Numeric examples

In particular, system (30) represents non linear model of an aircraft (A), which may be, for example, an air – air rocket. Thus,

$$x_1 = x = [\alpha \quad \beta \quad \gamma]^T, \qquad x_2 = \omega = [\omega_x \quad \omega_y \quad \omega_z]^T; \quad (44)$$

from the equivalence of equations (30) and (12) it results

$$h_1(x_1) = T(x),\ f_1(x_1) = a_f, \quad (45)$$

with $T(x)$ and a_f of forms (13).

The second equation (30) is equivalent with equations system (14), where $x_1 = x, x_2 = \omega, u = \delta$ and

$$f_2(x_1, x_2, u) = f(x, \omega, \delta), \qquad \delta = [\delta_e \quad \delta_p \quad \delta_d]^T, \quad (46)$$

which has two components of forms (16) with (17). From (16) command vector δ_c (δ_c – calculated with (28)) is expressed (v plays the role of u_ω). Indeed, equation (33) is equivalent with equations' system (15), where $x_2 = \omega, f_2 = f$ and $v = u_\omega$.

By comparing equations (37) and (26) 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. By comparing equations (31) and (26) one yields

$$q_1(\tilde{x}_1, t) = T^{-1}(x)(u_x - a_f), \quad (47)$$

with pseudo-command u_x of form (27).

Hence block diagram from fig.3 is equivalent with the one from fig.2. Equation (43) becomes

$$\bar{u}_\omega = K_z(\|Z\|_F + \bar{Z})(\|\tilde{x}\| + \|\tilde{\omega}\|)e_2, \quad (48)$$

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

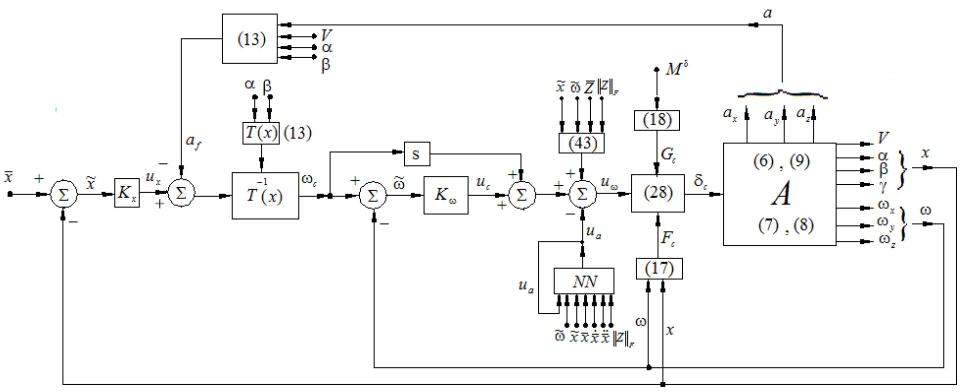


Fig.4 Block diagram of the stabilization system (variant 3)

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

$$\dot{\alpha} = -\left(a_1 + a_2\alpha^2\right)\alpha + \omega,\quad \dot{\omega} = \left(c_1 + c_2\alpha^2\right)\alpha + \left(c_3 + c_4\alpha^2\right)\delta,\quad \delta = \frac{1}{\tau s + 1}u, \tag{49}$$

where $a_1 = 1.02, a_2 = 1.3, c_1 = -57.2, c_2 = -322.2, c_3 = -70.15, c_4 = -360.25, \delta = 0.1\,s$.

Block diagram of the closed loop system (PA - A) is presented in fig.5; it has been obtained using diagram blocks from fig.3 and fig.4.

By identification of system (49) with system (12) and of system (15) with (30) one obtains

$$x_1 = x = \alpha, \bar{x} = \bar{\alpha}, x_2 = \omega_y = \omega, \bar{x}_2 = \omega_c = \bar{\omega},$$
$$T(x) = h_1(x_1) = 1, a_f = f_1(x_1) = -\left(a_1 + a_2\alpha^2\right)\alpha, \tag{50}$$
$$f(x,\omega,\delta) = f_2(x_1, x_2, \delta) = \left(c_1 + c_2\alpha^2\right)\alpha + \left(c_3 + c_4\alpha^2\right)\delta,$$
$$\bar{u}_\omega = \bar{v}, u_\omega = v = u.$$

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 $\left(H_0(s) = \frac{\alpha(s)}{\bar{\alpha}(s)}\right)$ is calculated and it's expressed as follows

$$H_0(s) = \frac{c_3 k_x k_\omega}{s^2 + (a_1 + c_3 k_\omega)s + (c_3 k_\omega - c_1 - a_1 c_3 k_\omega + c_3 k_x k_\omega)} \tag{51}$$
$$\Leftrightarrow H_0(s) = \frac{\alpha(s)}{\bar{\alpha}(s)} = \frac{c_3 k_x k_\omega}{s^2 + 2\xi\omega_0 + \omega_0^2}.$$

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

$$k_\omega = \frac{2\xi\omega_0 - a_1}{c_3},\quad k_x = \frac{\omega_0^2 - c_3 k_\omega + c_1 + a_1 c_3 k_\omega}{c_3 k_\omega}. \tag{52}$$

One chooses a feed-forward neural network with 8 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)

$$\tan\mathrm{sig}(n) = \frac{2}{1 + \exp(-2n)} - 1. \tag{53}$$

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

$$u_a = W^T \mathrm{tansig}\left(V^T I\right), \tag{54}$$

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

$$I^T = \begin{bmatrix} 1 & \tilde{\alpha}^T & \tilde{\omega}^T & \bar{\alpha}^T & \dot{\bar{\alpha}}^T & \ddot{\bar{\alpha}}^T & U_a^T & \|Z\|_F \end{bmatrix},\quad V^T = [1 \;\; 7 \;\; -3 \;\; 2 \;\; -1 \;\; -2 \;\; 3 \;\; 1], W = [1]. \tag{55}$$

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

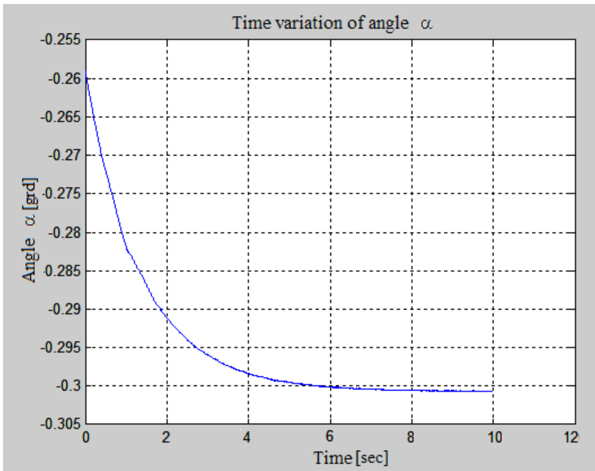


Fig.6 Time variation of the rocket's attack angle (without neural network)

Using Matlab/Simulink model from fig.7 one obtains $\alpha(t)$ (fig.8.a) and error $\tilde{\alpha}(t) = \bar{\alpha}(t) - \alpha(t)$ (fig.8.b).

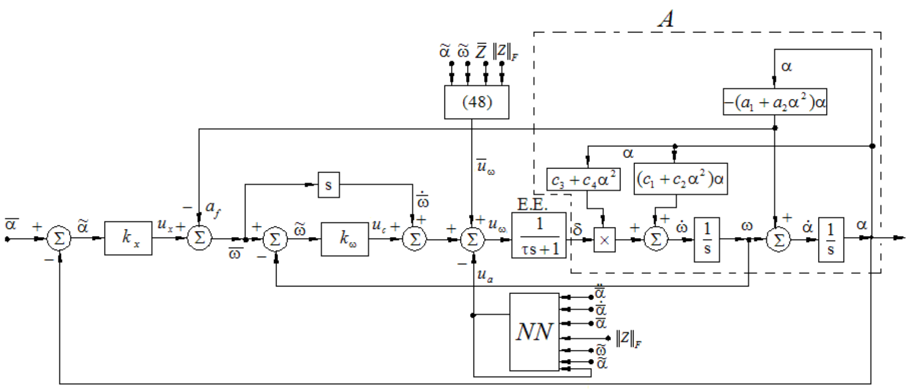


Fig.5 Neuro – adaptive command system for the rocket's move in vertical plane using model (49)

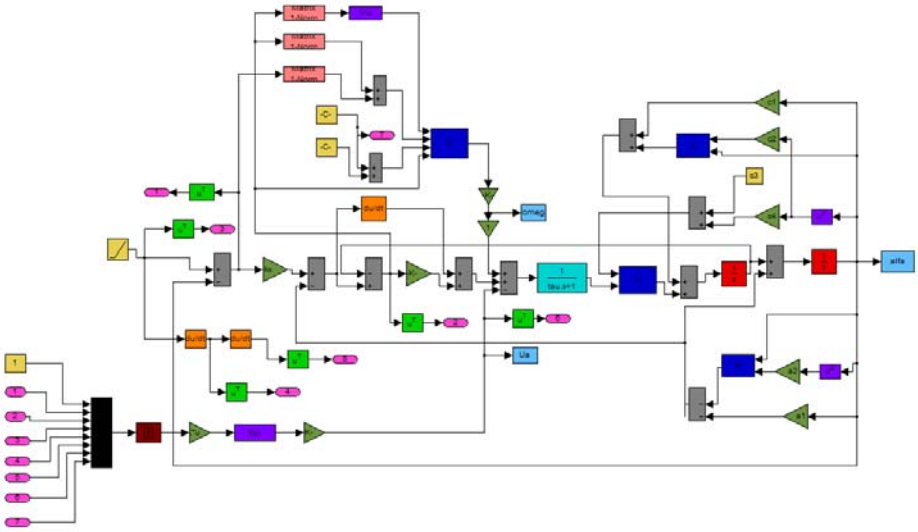
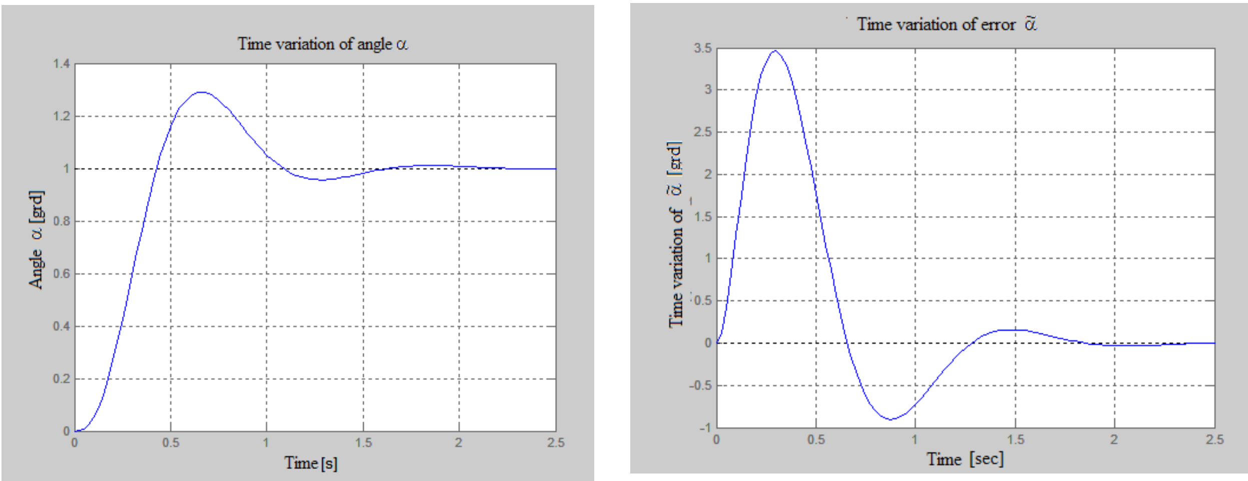


Fig.7 Matlab/Simulink model of the system from fig.5



a. b.

Fig.8 Dynamics of attack angle α and error $\tilde{\alpha}$ for the system from fig.5 (with neural network)

Other structures for the compensation of the unknown functions (from non-linear description of the flying objects' dynamic [9]) approximation errors are based on robust adaptive control using neural networks. The robustness deals with a parameter changing its value in the same time with non-linear functions of the control system (flying object [10]). Non-linear controller uses dynamic inversion and makes dynamic damp.

The controlled object (A) is described by equations system [5]

$$\begin{aligned} \dot{x}_1 &= f_1(x_1,t)+h_1(x_1,t)x_2, \\ \dot{x}_2 &= f_2(x,\bar{u},t), \\ \bar{u} &= [1+\mu\Delta(s)]u, \end{aligned} \quad (56)$$

where $x_1 \in R^{n-1}, x_2, u \in R; x = [x_1 \quad x_2]^T$ is the state, $u-$ command variable (input of A); functions f_1 and h_1 are known Lipschiz functions and f_2- partial known function, $\Delta(s)-$ transfer function of the non-modeled sub-system (considered stable), $\mu \geq 0$; one considers that sub-system x_1 has stable state $x_1 = 0$.

Function f_2 may be approximated as follows

$$f_2(x,u,t) = \hat{f}_2(x,t)+\hat{g}_2(x,t)u+\tilde{f}_2 \quad (57)$$

and system (56) becomes

$$\begin{aligned} &\dot{x}_1 = f_1(x_1,t)+h_1(x_1,t)x_2, \\ &\dot{x}_2 = \hat{f}_2(x,t)+\hat{g}_2(x,t)u+\tilde{f}_2+[f_2(x,\bar{u},t)-f_2(x,u,t)], \\ &\bar{u} = [1+\mu\Delta(s)]u; \end{aligned} \quad (58)$$

the second equation (58) has been obtained by adding and deducting in the right term $f_2(x,u,t)$ and taking into account equation (57); if f_2 is a global Lipschiz in u, then there is a constant $\bar{g}_2$ so that [5]

$$|f_2(x,\bar{u},t)-f_2(x,u,t)| \leq \bar{g}_2|\bar{u}-u| = \bar{g}_2\mu\Delta(s)u, \quad (59)$$

and $\tilde{f}_2-$ error of function f_2's aproximation; it may be compensated using a feed-forward linear neural network, whose output is the adaptive commnad u_a; the neural netwok is a Sigma – Pi one whose output is given by the following equation

$$u_a = \tilde{f}_2 = W^T g(x,u,t) + \varepsilon', \qquad (60)$$

where vectorul W is the weights' vector of the network, $g-$ base functions' vector and $\varepsilon'-$ neural network's reconstruction error.

Command law u may be chosen so that in second equation (58) one compensates components $\hat{f}_2$ and $\tilde{f}_2$. Hence, $\hat{g}_2 u$ must contain terms $\left(-\hat{f}_2\right)$ and $(-u_a)$. Also, command law must contain term $\dot{\bar{x}}_2; \tilde{x}_2 = \bar{x}_2 - x_2$, with $\bar{x}_2 = q_1(\tilde{x}_1, t)$. Thus, u may have the following form (this form is the same with the one from [5])

$$\hat{g}_2 u = \hat{g}_2 v + k\left[1 + |m| + |v|\right]\tilde{x}_2, \qquad (61)$$

with

$$\hat{g}_2 v = k_2 \tilde{x}_2 + \dot{\bar{x}}_2 - \hat{f}_2 - u_a, \qquad (62)$$

where

$$\tilde{x}_2 = \bar{x}_2 - x_2 = q_1(\tilde{x}_1, t) - x_2, \qquad (63)$$

q_1 beeing the feedback function of the system and $\tilde{x}_1 = \bar{x}_1 - x_1$; u_a has form (60), where one neglected ε', with W solution of differential equation

$$\dot{W} = -\tilde{x}_2 \mathrm{R}\, g(x,u,t), \qquad (64)$$

where $\mathrm{R} = \mathrm{R}^{\mathrm{T}}$ is a positive defined matrix; m is solution of equation [5]

$$\dot{m} + \gamma m = |u|. \qquad (65)$$

The term from (61), which contains the bracket, expresses the robustness.

Let's consider, for example, the case of rocket's longitudinal move. For this one projects an automat pilot using law (61). Longitudinal movement's model is described by equations (49) namely

$$\begin{aligned} \dot{\alpha} &= -\left(a_1 + a_2\alpha^2\right)\alpha + \omega, \\ \dot{\omega} &= \left(c_1 + c_2\alpha^2\right)\alpha + \left(c_3 + c_4\alpha^2\right)\bar{u}, \\ \dot{\bar{u}} &= \frac{1}{\delta}(\bar{u} - u), \end{aligned} \qquad (66)$$

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

Equations (66) have form (56) where: $x_1 = \alpha$, $x_2 = \omega_y = \omega -$ pitch angular velocity

$$\begin{aligned} &f_1(x_1,t) = -\left(a_1 + a_2 x_1^2\right)x_1, \\ &h_1(x_1,t) = 1, \\ &f_2(x,\bar{u},t) = \left(c_1 + c_2 x_1^2\right)x_1 + \left(c_3 + c_4 x_1^2\right)\bar{u}, \\ &\bar{u} = \frac{1}{\tau s + 1}u, \end{aligned} \qquad (67)$$

where τ is the performing element's time constant.

With notation $\tilde{x}_1 = \bar{x}_1 - x_1$, first equation (66) becomes

$$\dot{\tilde{x}}_1 = \left(a_1 + a_2 x_1^2\right)x_1 + \dot{\bar{x}}_1 - x_2. \qquad (68)$$

For $x_2 = \omega$ from this, one obtains equation

$$x_2 = -\dot{\tilde{x}}_1 + \left(a_1 + a_2 x_1^2\right)x_1 + \dot{\bar{x}}_1 \qquad (69)$$

and, with this, one expresses $\bar{x}_2$ so that term $\tilde{x}_2 = \bar{x}_2 - x_2$ contains terms between brackets from previous equation;

$$\bar{x}_2 = k_1\tilde{x}_1 + \left(a_1 + a_2 x_1^2\right)x_1 + \dot{\bar{x}}_1 = q_1(\tilde{x}_1, t). \qquad (70)$$

Hence,

$$\tilde{x}_2 = \dot{\tilde{x}}_1 + k_1\tilde{x}_1, \qquad (71)$$

which expresses the fact that this component of command is a proportional derivative type; $\tilde{x}_2 \to 0$ $(x_2 \to \bar{x}_2)$ in the same time with $\tilde{x}_1 \to 0\,(x_1 \to \bar{x}_1)$.

Thus, closed loop system is described by equations (66) or (67) – model of the rocket's longitudinal move, equations (60), (61), (62), (64), (65) and equation (71); functions $\hat{f}_2$ and $\hat{g}_2$ from (61) and (62) are obtained from equation (67) using (57);

$$\begin{aligned} \hat{f}_2 &= \hat{c}_1 x_1, \\ \hat{g}_2 &= \hat{c}_3. \end{aligned} \qquad (72)$$

These expressions and equation for $\dot{\bar{x}}_2$ allows calculus of (62); $\dot{\bar{x}}_2$ is obtained from derivation of

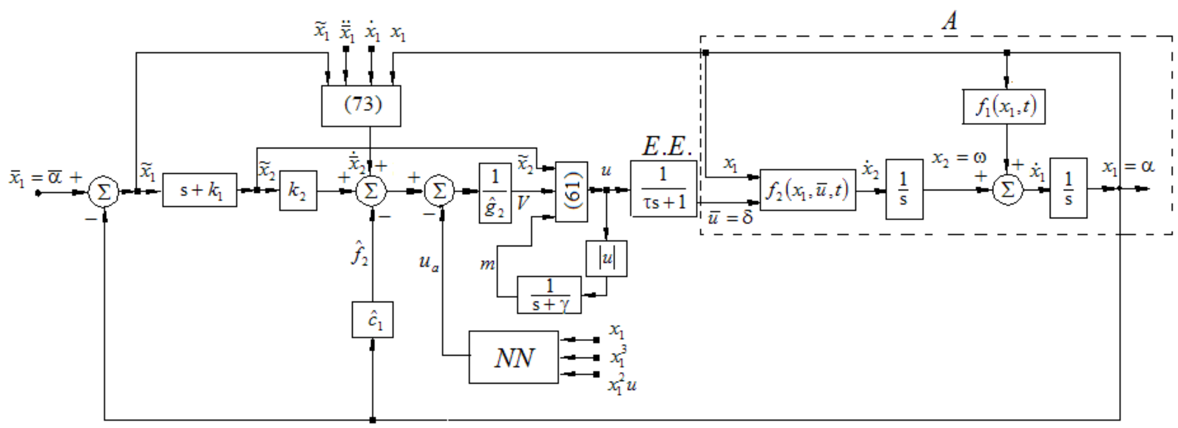


Fig.9 Neuro – adaptive command system for the rocket's move in vertical plane with non linear model (66)

(70); one yields

$$\dot{\bar{x}}_2 = k_1\dot{\tilde{x}}_1 + \ddot{\bar{x}}_1 + a_1 + a_2 x_1^2 + 2a_2 x_1^2 \dot{x}_1. \qquad (73)$$

The control system, obtained using the previous equations, is presented in fig.9.

For the study of the system one chooses the following parameters' values

$$\bar{x}_1 = \bar{\alpha} = 15\text{grd}, \hat{c}_1 = 0.1, \hat{g}_2 = \hat{c}_3 = -100,$$
$$\text{R} = 120I, \tau = 0.1\text{s}, \gamma = 1\text{s}^{-1},$$
$$k_1 = 5s^{-1}, k_2 = 20, k = 0.005.$$

Neural network is a feed-forward one, with 3 input neurons, one hidden layer's neuron and an output neuron. The three inputs of the network are $\alpha, \alpha^3, \alpha^2 u$. Activation functions are the linear one for the hidden layer neuron and tangent – hyperbolic one for the output's neurons.

Neural network's output is

$$u_a = W^T \text{tansig}\left(V^T I\right), \qquad (74)$$

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

$$I^T = [\alpha \quad \alpha^3 \quad \alpha^2 u],$$
$$V^T = [1 \quad -1 \quad 2], \qquad (75)$$
$$\text{W} = [1].$$

System's analysis may be made without components $\dot{\bar{x}}_2$ and u_a (without neural network) or using these components. Without them the stabilization system is oscillatory, non damped and α doesn't tend to $\bar{\alpha}$ (fig.10). Insert of the two variable leads to a non linear stable system ($\alpha \to \bar{\alpha}$).

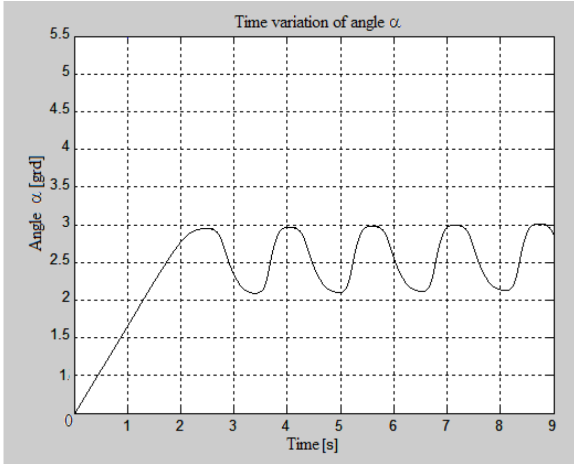


Fig.10 Time variation of the rocket's attack angle (without neural network)

Using Matlab/Simulink model from fig.11 one obtains time variations of the attack angle (fig.12.a) and of the error $\tilde{\alpha} = \bar{\alpha} - \alpha$ (fig. 12.b) ($(\bar{\alpha} = 1\,\text{grd})$).

In the structure of the observers one introduced neural networks because of their capacities of nonlinearities' approximation and because of their learning's ability. Thus, the use of neural network is extended to adaptive observers for non – linear systems and to control architectures observer – controller.

Neural network for adaptive command of the systems with dynamic inversion processes state variables given by observers; it is used for compensation of the non linear inversion error or for state errors' compensation [11], [12], [13], [14], [15], [16].

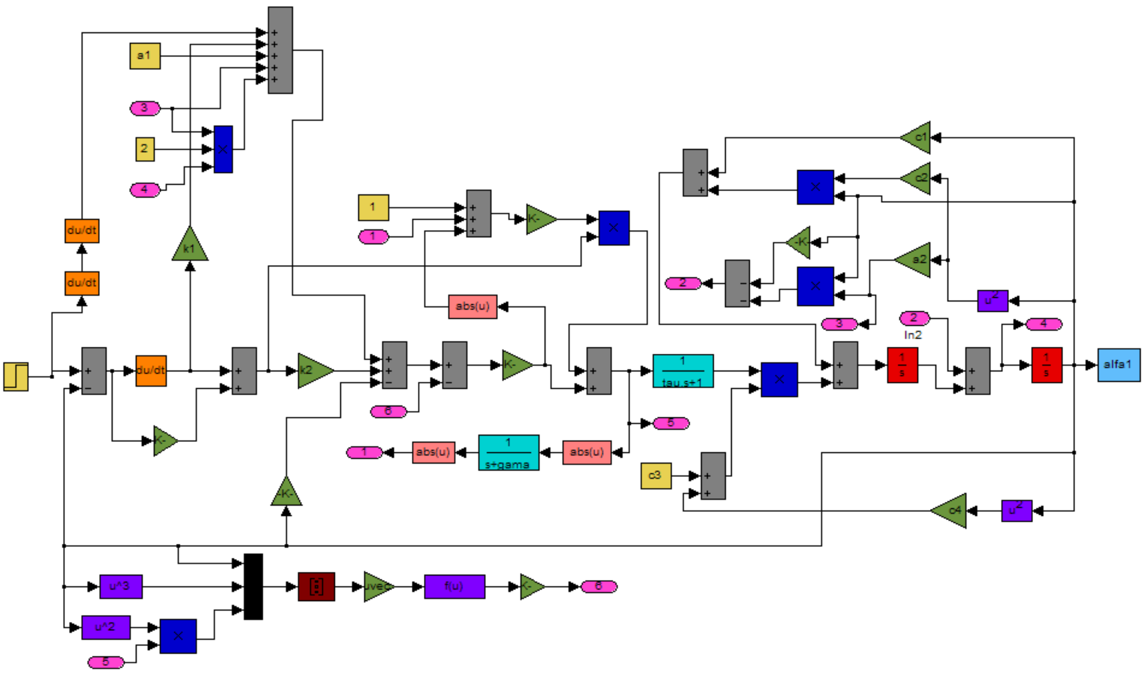
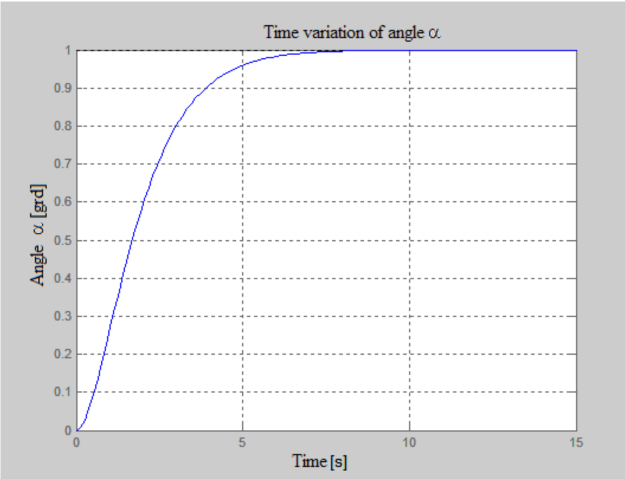
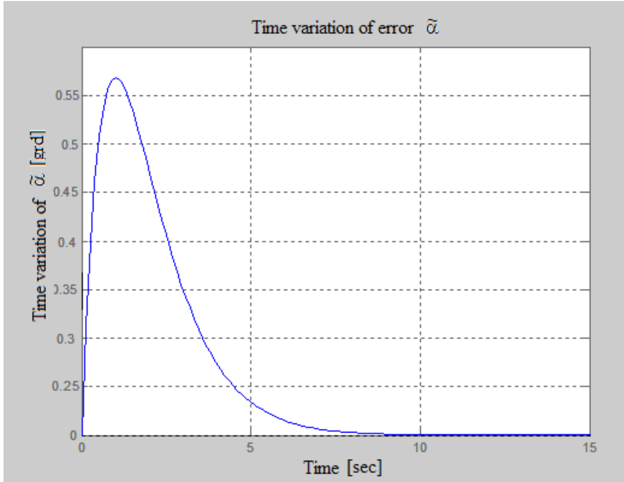


Fig.11 Matlab/Simulink of the system from fig.9



a.



b.

Fig.12 Dynamics of attack angle α and error $\tilde{\alpha}$ for the system from fig.9 (with neural network)

5 Conclusion

One presents some equivalent forms of models for A's movement as functions of A's angles related to aerodynamic trihedron, of angular velocities and linear accelerations.

Stabilization structures have some control loops after angles, angular velocities and linear accelerations and a control adaptive loop using a neuronal network for dynamic inversion error compensation of non-linear unknown function from model A.

The adaptive command synthesis is based upon Liapunov function. Also, one presents the study's results of very maneuverable rocket's longitudinal move; the rocket's move is described by non-linear models; for this theoretical study neuro-adaptive command laws have been used.

References:

[1] Darouch M. Linear Functional Observers for Systems with Delays in Stable Variables. *IEEE Transactions on Automatic Control*, vol. 46, no. 3, 2001, pp. 491 – 496.

[2] Fu M.Y., Duan G.R., Song S.M. *Design of Unknown Input Observer for Linear Time – Delay Systems*.

[3] Gejic Z. *Introduction to Linear and Nonlinear Observers*. Rutgers University, USA.

[4] Carlos M., Vélez S., Andrés A. Multirate control of an unmanned aerial vehicle, *WSEAS Transactions on Circuits and Systems* Issue 11, Volume 4, November 2005, ISSN 1109 - 2734.

[5] McFarland M.B., Calise A.J. Robust adaptive control of uncertain nonlinear systems using neural networks. *Proceedings of the American Control Conference, Albuquerque*, New Mexico, June, 1997.

[6] McFarland M.B., Calise A.J. Multilayer Neural Networks and Adaptive Nonlinear Control of Agile Anti – Air Missiles. *AIAA Guidance, Navigation and Control Conference*, Baltimore, Maryland, August, 1997.

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

[8] Isidori A. *Nonlinear control systems*. Springher, Berlin, 1995.

[9] Ching-Hung Lee, Bo-Ren Chung, FNN-based Disturbance Observer Controller Design for Nonlinear Uncertain Systems. *WSEAS Transactions on systems and control*, Vol. 2, March 2007.

[10] Calistru C.N., Robust Control via Genetic Algorithms and Integral Criteria Minimization. *WSEAS Transactions on Systems*, Issue 4, Vol. 3, pp. 1669 - 1674, June 2004, ISSN1109-2777.

[11] Calise A.J., Johnson E.N., Johnson M.D., Corban J.E. Applications of Adaptive Neural – Networks Control to Unmanned Aerial Vehicles. *Journal of Harbin Institute of Technology*, 2006, vol. 38, no. 11, pp.1865–1869.

[12] Calise A.J., Lee H., Kim N. High Bandwidth Adaptive Flight Control. *AIAA Guidance, Navigation and Control Conference*. 14–17 August, 2000, In Proceedings, 11 pp.

[13] Hovakimyan N., Kim N., Calise A.J., Parasad J.V.R. Adaptive Output Feedback for High –

Bandwidth Control of an Unmanned Helicopter. *AIAA Guidance, Navigation and Control Conference*, 6 – 9 August, Montreal, Canada, In Proceedings, 11 pp.

[14] Johnson E.N., Calise A.J. Pseudo – Control Hedging: A New Method for Adaptive Control. *Navigation Guidance and Control Technology Workshop*, November, 1 – 2, 2000,23 pp.

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

[16] Johnson E.N., Calise A.J. Neural Network Adaptive Control of Systems with Input Saturation. *American Control Conference*, 2001, In Proceedings, 2001, 6 pp.