Optimal Control of Flying Objects' Move After Estimated State Vector Using a Reduced Order Observer

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

CALBUREANU MADALINA
University of Craiova,
Faculty of Mechanics
ROMANIA
madalina.calbureanu@gmail.com

CEPISCA COSTIN
Electrical Engineering Faculty
University POLITEHNICA of Bucharest
313 Splaiul Independentei, Sector 6
ROMANIA
costin@wing.ro

Abstract: - In this paper one presents an algorithm for project of a reduced order observer (ALGLOOR) with applications to flying objects' move. The command law is an optimal one in rapport with state vector of the observer. Numeric calculus examples are also presented: for longitudinal and lateral move and vertical move of a rocket in vertical plane in rapport with equal signal line; using Matlab/Simulink model of closed loop system one has made numeric calculus programs and with them one obtained different time characteristics, which expresses state variables' dynamics of the flying objects dynamic models.

Key-Words: - algorithm, optimal, control law, rocket, state vector, reduced order observer, command

1 Introduction

Observers are used for estimation of flying object state vector using its input and output signals. Measuring some of the state vector's components (output vector's components) one reduces sensors' number especially the number of sensors for reachless or difficult measurable state variables (e.g. elastic deformations) [1], [2], [3]. First observer structures have been developed by Luenbergher in 1966, 1971, 1979. The observers are used both for systems with known inputs and unknown inputs systems; their project belongs to different researchers (Bhatta-Charyya, 1978; Chen and Patton, 1999; Darouach, 1994; Hostetter and Meditch, 1973; Hou and Müller, 1992; Hui and Zak, 1993 and 2005; Kudva, 1980; Kurek, 1983; Wang, 1975; Yang and Wilde, 1988; Krzminski and Kakzorek, 2005 and so on). Recent observers are obtained for systems with some unknown inputs or for systems with subsystems whose dynamic is unknown. For such systems one may use reduced order observers. Unknown inputs may be disturbances, errors and so on.

The estimator (filter) Kalman – Bucy [4], [5], [6] is the best observer for stochastic systems' state estimation. In case of nonlinear systems with partial known inputs or with unknown subsystems adaptive observers have been made; these observers use neural networks [7], [8], [9], [10], [11].

Recent researches deal with project of observers for the intern delay systems [12], [13], [14], based on Liapunov theory; observers' project means matrices inequalities solve. Linear state observers are used for x state estimation for linear system using measured variables vector y, the input vector or some components of this vector.

2 Dynamic models of flying objects and reduced order observer

Flying objects' dynamic may be described by equations [15]

$$\dot{x} = Ax + Bu + Eu_p, \quad (1)$$

$$y = Cx, \quad (2)$$

where x is the state vector $(n \times 1)$ of aircraft model, $y-$ output vector $(p \times 1)$, $u-$ the vector containing knowable inputs $(m_1 \times 1)$, u_p- vector $(m_2 \times 1)$ of unknown inputs (disturbances or nonlinear functions of non modeled systems) [16], $A-$ matrix $(n \times n)$, $B-$ matrix $(n \times m_1)$, $E-$ matrix $(n \times m_2)$, $C-$ matrix $(p \times n)$; matrices A, B, C are known.

Let's consider the reduced order observer described by equations [17], [18]

$$M\dot{z} = Fz + Gu + Hy, \quad (3)$$

$$\hat{x} = Pz + Qy, \quad (4)$$

with $z(r \times 1)$, $\hat{x}(n \times 1)$, $M(r \times r)$, $F(r \times r)$, $\det M = 0$ and $rang\, M \le n-q, G(r \times m), H(r \times p), P(n \times r)$, $Q(n \times p); m = m_1 + m_2; e$ is the observer's error

$$e = z - Nx, \quad (5)$$

By derivation of equation (5), replacing $\dot{x}$ with form (1), y with form (2), $\dot{z}$ with form (3) and assenting that all coefficients of x, u and u_p be null,

$$G = MNB, \quad (6)$$

$$HC = MNA - FN, \quad (7)$$

$$MNE = 0, \quad (8)$$

one obtains the equation of error e

$$M\dot{e} = Fe. \quad (9)$$

Error $(\hat{x} - x)$ may be expressed in rapport with e. Taking into account equations (4) and (5), one results

$$\hat{x} - x = Pe \quad (10)$$

if

$$PN + QC = I \Leftrightarrow [P \quad Q][N \quad C]^T = I. \quad (11)$$

$I-$ unity matrix.

3 Project algorithm of the reduced order observer (ALGLOOR)

First of all one chooses matrix N of form

$$N = \begin{bmatrix} N_1 & N_2 \\ N_3 & N_4 \end{bmatrix} = \begin{bmatrix} 0_{(n-q)\times(n-p)} & I_{(n-q)\times p} \\ 0_{(n-p)\times(n-p)} & 0_{(n-p)\times p} \end{bmatrix}, \quad (12)$$

where $I_{(n-p)\times(n-p)} = I_{n-p}$ is unity matrix and $I_{(n-q)\times p}$ is a matrix whose elements are equal with 1 at intersection of line i and column $j = i$ and all the other elements are null; $p = rang\, C, q = rang\, E = n - p$.

One calculates matrix M so that condition (8) be fulfilled. For this, matrix M is partitioned

$$M = \begin{bmatrix} M_{1(n-p)\times(n-q)} & M_{2(n-p)\times(n-p)} \\ M_{3(n-q)\times(n-q)} & M_{4(n-q)\times(n-p)} \end{bmatrix}. \quad (13)$$

With E of form

$$E^T = [E_1 \quad E_2], E_1[(n-p)\times q], \quad E_2[(n-q)\times q], \quad (14)$$

equation (8) is equivalent with system

$$M_2E_1 + M_1E_2 = 0, \quad M_4E_1 + M_3E_2 = 0. \quad (15)$$

Choosing

$$M_2 = 0_{(n-p)\times[n-p]}, M_4 = 0_{(n-q)\times(n-p)},$$

one results

$$M = \begin{bmatrix} M_1 & 0 \\ M_3 & 0 \end{bmatrix}, \quad M_1E_2 = 0, M_3E_2 = 0. \quad (16)$$

With M and N calculated, one determines matrix G using equation (6). Also, one calculates matrices F and H with (7). For this, matrix F is partitioned as follows

$$F = \begin{bmatrix} F_{1(n-p)\times(n-q)} & 0_{(n-p)\times(n-p)} \\ F_{3(n-q)\times(n-q)} & 0_{(n-q)\times(n-p)} \end{bmatrix}. \quad (17)$$

With H and A of forms

$$H^T = [H_1 \quad H_2], H_1[(n-p)\times p], H_2[(n-q)\times p], \quad (18)$$

$$A = \begin{bmatrix} A_{1(n-p)\times(n-p)} & A_{2(n-p)\times p} \\ A_{3p\times(n-p)} & A_{4p\times p} \end{bmatrix}, \quad (19)$$

condition (7) is equivalent with system

$$\begin{aligned} M_1A_3 &= H_1C_1, \\ M_1A_4 - F_1 &= H_1C_2, \\ M_3A_3 &= H_2C_1, \\ M_3A_4 - F_3 &= H_2C_2; \end{aligned} \quad (20)$$

one results

$$H_1 = M_1A_3C_1^+, \quad H_2 = M_3A_3C_1^+, \quad (21)$$

$$F_1 = M_1A_4 - M_1A_3C_1^+C_2, \quad F_3 = M_3A_4 - M_3A_3C_1^+C_2. \quad (22)$$

One yield

$$H = \begin{bmatrix} M_1A_3C_1^+ \\ M_3A_3C_1^+ \end{bmatrix}, \quad F = \begin{bmatrix} M_1A_4 - M_1A_3C_1^+C_2 & 0_{(n-p)\times(n-p)} \\ M_3A_4 - M_3A_3C_1^+C_2 & 0_{(n-q)\times(n-p)} \end{bmatrix}. \quad (23)$$

Matrices P and Q are obtained from equation (11). These matrices are partitioned as follows

$$P=\begin{bmatrix} P_{1(n-p)\times(n-q)} & P_{2(n-p)\times(n-p)} \\ P_{3(n-q)\times(n-q)} & P_{4(n-q)\times(n-p)} \end{bmatrix},\quad Q=\begin{bmatrix} Q_{1(n-p)\times p} \\ Q_{2(n-q)\times p} \end{bmatrix}. \tag{24}$$

Condition (11) is equivalent with system

$$\begin{aligned} &P_2+Q_1C_1=I_{(n-p)\times(n-p)}, \\ &P_1+Q_1C_2=0_{(n-p)\times(n-q)}, \\ &P_4+Q_2C_1=0_{(n-q)\times(n-p)}, \\ &P_3+Q_2C_2=I_{(n-q)\times(n-q)}. \end{aligned} \tag{25}$$

For example, one chooses

$$\begin{aligned} &P_1=0_{(n-p)\times(n-q)}, \\ &P_3=0_{(n-q)\times(n-q)} \end{aligned} \tag{26}$$

and equations (26) become

$$P_2+Q_1C_1=0, Q_1C_2=0; \tag{27}$$

$$P_4+Q_2C_1=0, Q_2C_2=I; \tag{28}$$

one results

$$Q_2=C_2^+, P_4=-C_2^+C_1, \tag{29}$$

and Q_1 and P_2 represent solution of system (28).

The above results lead to following algorithm (ALGLOOR).

Step 1: one calculates $p=\operatorname{rang} C, q=\operatorname{rang} E$ and verifies condition $p+q=n$; if this condition isn't fulfilled then the algorithm isn't useful in order to project the reduced order observer.

Step 2: one calculates matrix N with (12) and matrix M with (16);

Step 3: one calculates matrix G with (6);

Step 4: matrices A, C are partitioned; one calculates C_1^+ and, using this, matrices H and F are calculated with equation (23);

Step 5: matrices P, Q are partitioned using (24) and after that one calculates their components with (27), (28) and (30).

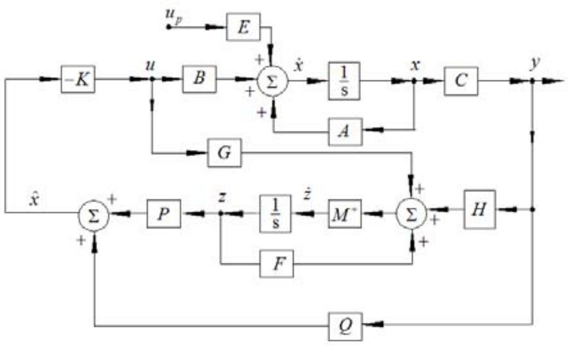

Fig.1 Modeling diagram for optimal command system

In fig.1 modeling diagram for optimal command system described by equations (1) and (2) and which has as component a reduced order observer (equations (3) and (4)) is presented.

One remarks that the algorithm ALGLOOR doesn't use non singular transformations like other algorithms [19], [20], [21].

4 Numeric examples

Example 4.1 (aircraft longitudinal move)

A reduced order observer for state $\hat{x}$ estimation described by equation (31) from [1] for longitudinal move with

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, E^T=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$n=4, p=2, q=2$$

may be projected using ALGLOOR algorithm

$$\begin{bmatrix} \Delta\dot{V}_x \\ \Delta\dot{\alpha} \\ \Delta\dot{\theta} \\ \Delta\dot{\omega}_y \end{bmatrix}=\begin{bmatrix} -0.007 & 0.012 & -9.81 & 0 \\ -0.128 & -0.54 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.065 & 0.96 & 0 & -0.99 \end{bmatrix}\begin{bmatrix} \Delta V_x \\ \Delta\alpha \\ \Delta\theta \\ \Delta\omega_y \end{bmatrix}+\begin{bmatrix} 0 \\ -0.04 \\ 0 \\ -12.5 \end{bmatrix}\delta_p. \tag{31}$$

Matlab program utilized is the first presented in Section 4; one obtained matrices

$$N=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, M=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, G=\begin{bmatrix} 0 \\ -12.5 \\ 0 \\ -12.5 \end{bmatrix},$$

$$H=\begin{bmatrix} 0 & 0 \\ 0.065 & 0.065 \\ 0 & 0 \\ 0.065 & 0.065 \end{bmatrix}, F=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.99 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -0.99 & 0 & 0 \end{bmatrix}, \tag{32}$$

$$P=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, Q=\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

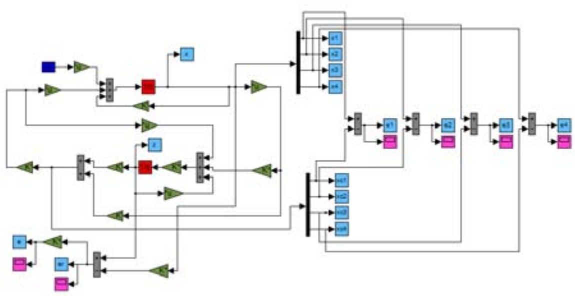

Fig.2 Matlab/Simulink model of the closed loop system

In fig.2 Matlab/Simulink model of the closed loop system described by equations $(1)\div(4)$ and $u=-K\hat{x}$ is presented. For $u=0$ (open loop) one obtains characteristics $x_i(t)$ (blue) and $\hat{x}_i(t)$ (red)- fig.3. These characteristics are also obtained for closed loop system – fig.4.

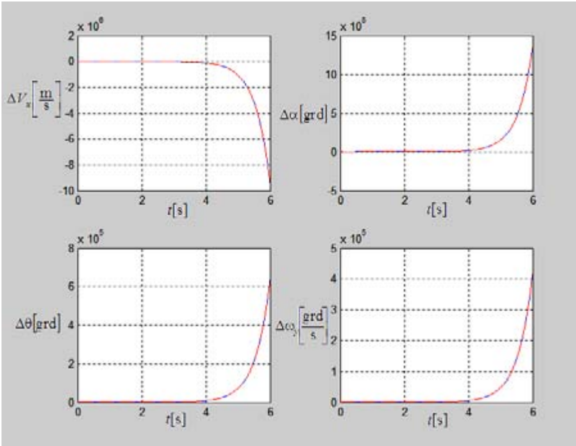

Fig.3 Characteristics $x_i(t)$ and $\hat{x}_i(t)$ (open loop)

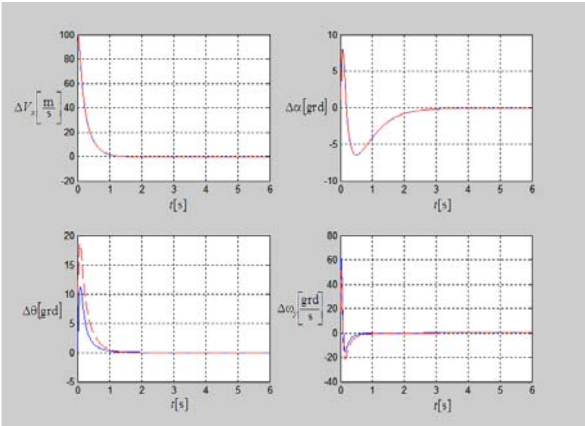

Fig.4 Characteristics $x_i(t)$ and $\hat{x}_i(t)$ (closed loop)

ALGLOOR may be used with good results using longitudinal move's model from [22]

Example 4.2 (aircraft lateral move)

Similar, for lateral move described by equation [1]

$$\begin{bmatrix}\Delta\dot{\beta}\\ \Delta\dot{\omega}_z\\ \Delta\dot{\omega}_x\\ \Delta\dot{\varphi}\end{bmatrix}=\begin{bmatrix}-0.0558 & -0.9968 & 0.0802 & 0.0415\\ 0.598 & -0.115 & -0.0318 & 0\\ 0.305 & 0.388 & -0.465 & 0\\ 0 & 0.0805 & 1 & 0\end{bmatrix}\begin{bmatrix}\Delta\beta\\ \Delta\omega_z\\ \Delta\omega_x\\ \Delta\varphi\end{bmatrix}+\begin{bmatrix}0.0073 & 0\\ -0.475 & 0.123\\ 0.153 & 1.063\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_d\\ \delta_e\end{bmatrix}, \quad (33)$$

are known

$$C=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, E^T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix},$$

$$n=4,\ p=q=2;$$

one obtains

$$N=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, M=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, G=\begin{bmatrix}0.15 & 1.06\\ 0 & 0\\ 0.15 & 1.06\\ 0 & 0\end{bmatrix},$$

$$H=\begin{bmatrix}0.30 & 0.38\\ 0 & 0.08\\ 0.30 & 0.38\\ 0 & 0.08\end{bmatrix}, F=\begin{bmatrix}-0.46 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ -0.46 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}, \quad (34)$$

$$P=\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, Q=\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\\ 0 & 0\end{bmatrix}.$$

Characteristics $x_i(t)$ (blue) and $\hat{x}_i(t)$ (red) are presented in fig.5 (without controller) and fig.6 (with controller $u=-K\hat{x}$).

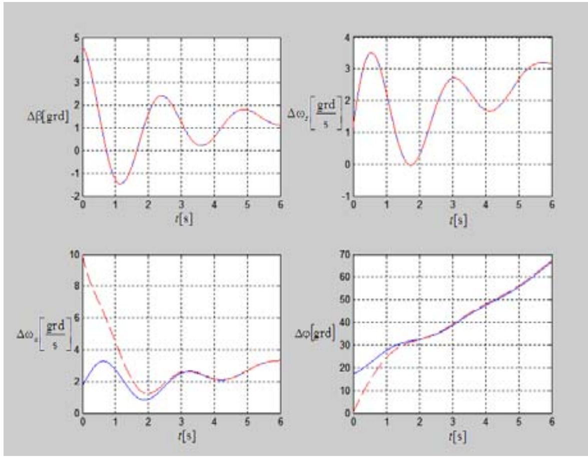

Fig.5 Characteristics $x_i(t)$ and $\hat{x}_i(t)$
(without controller)

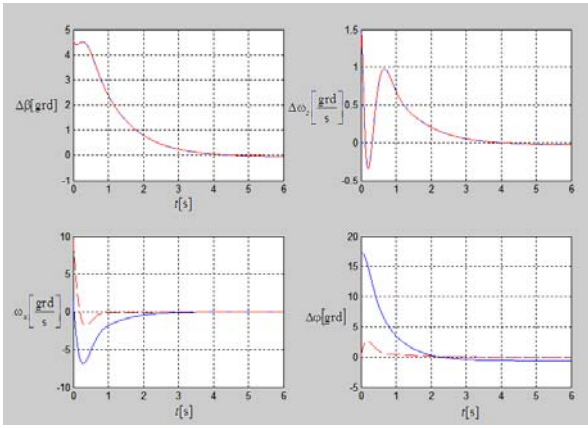

Fig.6 Characteristics $x_i(t)$ and $\hat{x}_i(t)$
(with controller)

Example 4.3 (rocket move)

One considers the case of a rocket move in vertical plane; it is directed by three points method (the coliniarity CP–R–T; CP – control point, R – rocket, T - target). Lateral deviation z in rapport with equal signal line is described by differential equation [23], [24]

$$\dot{z}=V\vartheta+Vf_T, \quad (35)$$

where V is the flight velocity, ϑ – slope of the trajectory ($\vartheta=\theta-\alpha, \theta$ – pitch angle), f_T – disturbance. ϑ is defined by [24]

$$T_V\dot{\vartheta}=\alpha-T_V\frac{g}{V}\cos\vartheta. \quad (36)$$

Expressing the above equation in variables α and θ, it becomes

$$T_V\dot{\theta}=T_V\dot{\alpha}+\alpha-T_V\frac{g}{V}\cos\vartheta \quad (37)$$

or has the liniarised form

$$T_V\dot{\theta} = T_V\Delta\dot{\alpha} + \Delta\alpha - \tilde{\alpha},$$
$$\tilde{\alpha} = -\frac{g}{V}T_V \cos\vartheta_0, \dot{\theta} = \Delta\dot{\theta}; \tag{38}$$

Flying object's move in vertical plane is defined by equation (36) and the following one

$$\dot{V} + \frac{c_x}{c_y^\alpha T_V}V = \left(1 + \frac{c_x}{c_y^\alpha}\right)\frac{F_T}{m} - g\sin\vartheta \tag{39}$$

Rocket's move around mass centre in vertical plane is described by equation [24]

$$J_z\ddot{\theta} + m_{\dot{\theta}}\dot{\theta} + m_\alpha\alpha = m_\delta\delta, \tag{40}$$

where J_z is the inertia moment in rapport with lateral (horizontal) axis, $m_{\dot{\theta}}$ – dynamic damp moment coefficient, m_α – static stabilisation moment coefficient, m_δ – command coefficient moment. With $\theta = \vartheta + \alpha$, (36) and (39) the above equation leads to linear form

$$\Delta\ddot{\alpha} + 2\xi\omega_0\Delta\dot{\alpha} + \omega_0^2\Delta\alpha = k_\delta\delta + m_{\dot{\theta}}\frac{g}{J_zV}\cos\theta_0, \tag{41}$$
$$\Delta\delta = \delta,$$

where ξ is the damp coefficient and ω_0 – proper frequency of the rotation move in vertical plane;

$$2\xi\omega_0 = \frac{m_{\dot{\theta}}}{J_z} + \frac{1}{T_V}, k_\delta = \frac{m_\delta}{J_z},$$
$$\omega_0^2 = \frac{m_\alpha}{J_z} + \frac{m_{\dot{\theta}}}{J_zT_V} + g\frac{\sin\theta_0}{VT_V} - \frac{\dot{T}_V}{T_V^2} \cong \frac{m_\alpha}{J_z} + \frac{m_{\dot{\theta}}}{J_zT_V}. \tag{42}$$

Equation (40) is equivalent with the following one

$$\ddot{\theta} = -\frac{m_{\dot{\theta}}}{J_z}\dot{\theta} - \frac{m_\alpha}{J_z}\alpha + k_\delta\delta, \tag{43}$$

which, taking into account

$$\frac{m_{\dot{\theta}}}{J_z} = 2\xi\omega_0 - \frac{1}{T_V} = a_1,$$
$$\frac{m_\alpha}{J_z} \cong \omega_0^2 - \frac{1}{T_V}\frac{m_{\dot{\theta}}}{J_z} = \omega_0^2 - \frac{a_1}{T_V} = a_0, \tag{44}$$

leads to

$$\ddot{\theta} = a_1\dot{\theta} - a_0\Delta\alpha + k_\delta\delta. \tag{45}$$

By derivation of equation (35), one obtains [25]

$$\ddot{z} = \frac{V}{T_V}\Delta\alpha + W_T, W_T = \dot{V}f_T + V\dot{f}_T - g\cos\vartheta_0; \tag{46}$$

W_T is an equivalent disturbance having the significance of normal acceleration to equal signal line [26]. For calculus of above equations' coefficients one uses calculus equations from [27]

$$T_V = \frac{1}{d_1}, \omega_0 = \sqrt{d_1d_4 - d_3},$$
$$\xi = \frac{d_1 + d_4}{2\sqrt{d_1d_4 - d_3}}, k_\delta = d_2, \tag{47}$$

where d_1, d_2, d_3, d_4 are read using diagrams or graphic characteristics for different rocket types at different flight seconds. For instance, for an Oerlikon rocket, in the 10th flight second

$$d_1 = 1.5\,\mathrm{s}^{-1}, d_2 = 40\,\mathrm{s}^{-2}, d_3 = -10\,\mathrm{s}^{-2}, d_4 = 1.2\,\mathrm{s}^{-1},$$
$$V = 400\mathrm{m/s}, T_V = 0.66\,\mathrm{s},\ \omega_0 = 4.669\,\mathrm{s}^{-1}, \xi = 0.062, \tag{48}$$
$$k_\delta = d_2 = 40\,\mathrm{s}^{-2}, a_1 = 0.92\,\mathrm{s}^{-1}, a_0 = 23.18\,\mathrm{s}^{-2}.$$

For the 40th flight second

$$d_1 = 2.5\,\mathrm{s}^{-1}, d_2 = 100\,\mathrm{s}^{-2}, d_3 = -80\,\mathrm{s}^{-2}, d_4 = 1\,\mathrm{s}^{-1},$$
$$V = 400\mathrm{m/s}, T_V = 0.4\,\mathrm{s}, \omega_0 = 9.083\,\mathrm{s}^{-1}, \xi = 0.0212, \tag{49}$$
$$k_\delta = d_2 = 100\,\mathrm{s}^{-2}, a_1 = 2.115\,\mathrm{s}^{-1}, a_0 = 87.7881\,\mathrm{s}^{-2}.$$

For the 50th flight second

$$d_1 = 1\,\mathrm{s}^{-1}, d_2 = 25\,\mathrm{s}^{-2}, d_3 = -60\,\mathrm{s}^{-2}, d_4 = 0.4\,\mathrm{s}^{-1},$$
$$V = 400\mathrm{m/s}, T_V = 1\,\mathrm{s}, \omega_0 = 7.77\,\mathrm{s}^{-1}, \xi = 0.011, \tag{50}$$
$$k_\delta = d_2 = 25\,\mathrm{s}^{-2}, a_1 = 0.8197\,\mathrm{s}^{-1}, a_0 = 61.2082\,\mathrm{s}^{-2}.$$

Choosing the state vector $x^T = [z \quad \dot{z} \quad \Delta\alpha \quad \dot{\theta}]$, system formed by equations (38), (45) and (46) becomes

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = \frac{V}{T_V}x_3 + W_T,$$
$$\dot{x}_3 = -\frac{1}{T_V}x_3 + x_4 + \frac{1}{T_V}\tilde{\alpha}, \tag{51}$$
$$\dot{x}_4 = -a_0x_3 - a_1x_4 + k_\delta\delta.$$

If the input is $u = \delta$ and the disturbances vector is $u_p^T = [\tilde{\alpha} \quad W_T]$, the above equations system has the form

$$\dot{x} = Ax + Bu + Eu_p, \tag{52}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{V}{T_V} & 0 \\ 0 & 0 & -\frac{1}{T_V} & 1 \\ 0 & 0 & -a_0 & -a_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ k_\delta \end{bmatrix}, E = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ \frac{1}{T_V} & 0 \\ 0 & 0 \end{bmatrix}. \tag{53}$$

Using ALGLOOR for the 10th flight second, one obtains

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, N = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$
$$M = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, H = \begin{bmatrix} 0 & 0 \\ 600 & 0 \\ 0 & 0 \\ 600 & 0 \end{bmatrix}, \tag{54}$$
$$F = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Eigen values of the open loop system are

$$s_1 = 0, s_2 = 0, \quad s_3 = 4.82, s_4 = -4.52. \tag{55}$$

In closed loop system one obtains the characteristics from fig.7.

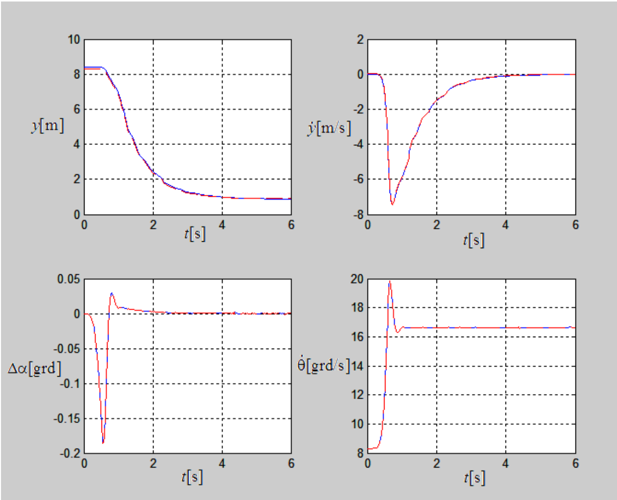

Fig.7 State variables $x_i(t)$ and $\hat{x}_i(t)$ of the optimal control system for rocket move in vertical plane using 10th flight second parameters

Similarly, for the 40th and 50th flight second, it results

$$C=\begin{bmatrix}0&0&1&0\\0&0&0&1\end{bmatrix}, N=\begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix},$$
$$M=\begin{bmatrix}0&0&1&0\\0&1&0&1\\0&0&1&0\\0&0&0&1\end{bmatrix}, G=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, H=\begin{bmatrix}0&0\\1000&0\\0&0\\1000&0\end{bmatrix}, \tag{56}$$
$$F=\begin{bmatrix}0&0&0&1\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}, P=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{bmatrix}, Q=\begin{bmatrix}0&0\\0&0\\1&0\\0&1\end{bmatrix},$$

respectively,

$$C=\begin{bmatrix}0&0&1&0\\0&0&0&1\end{bmatrix}, N=\begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix},$$
$$M=\begin{bmatrix}0&0&1&0\\0&1&0&1\\0&0&1&0\\0&0&0&1\end{bmatrix}, G=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, H=\begin{bmatrix}0&0\\400&0\\0&0\\400&0\end{bmatrix}, \tag{57}$$
$$F=\begin{bmatrix}0&0&0&1\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}, P=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{bmatrix}, Q=\begin{bmatrix}0&0\\0&0\\1&0\\0&1\end{bmatrix}$$

and characteristics from fig.8 anf fig.9.

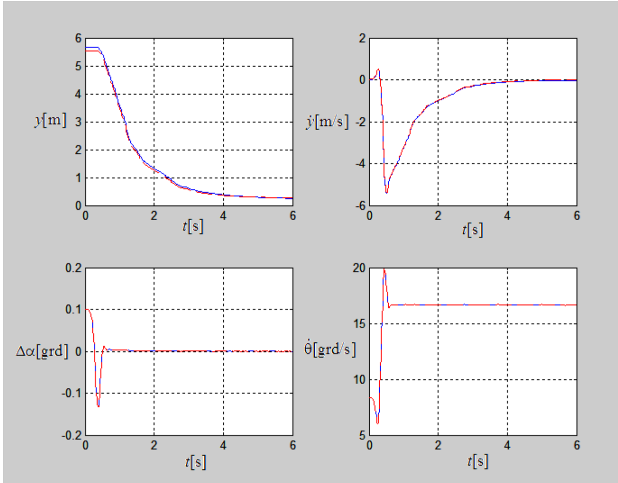

Fig.8 State variables $x_i(t)$ and $\hat{x}_i(t)$ of the optimal control system for rocket move in vertical plane using 40th flight second parameters

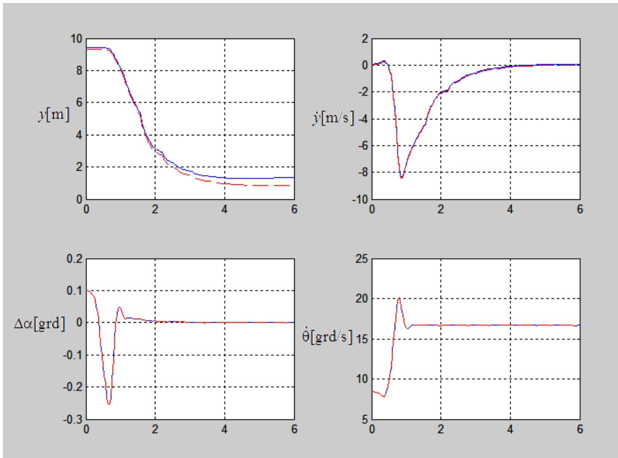

Fig.9 State variables $x_i(t)$ and $\hat{x}_i(t)$ of the optimal control system for rocket move in vertical plane using 50th flight second parameters

5 Matlab programs

One presents the program for the longitudinal move of the flying object and for rocket move in vertical plane at 10th flight second. The programs for lateral move or for rocket move in vertical plane at 40th and 50th flight second are similar. In programs presented in Subsections 5.1 and 5.2 one runs other subprograms: ALGLX & ALGLX10. These programs are shown in Subsections 5.3 and 5.4.

5.1 ALGLOOR program (longitudinal move)

```
clear all; run ALGLX; close all;
clear C; clear contor; clear e; clear i;
clear r; clear m;
clear P; clear Q; KK=K;
C=[1 0 0 0;0 1 0 0];
```

```
E=[1 0;0 1;0 0;0 0];
n=size(A,1); m=size(B,2);
q=size(E,2); p=size(C,1);
x0=[100;1;0;10];
xc0=[10;0;-10;20];
up=[-0.0375;-0.0963];
deltap=[5];
A1=A(1:(n-p),1:(n-p));
A2=A(1:(n-p),(n-p+1):size(A,2));
A3=A((n-p+1):size(A,1),1:(n-p));
A4=A((n-p+1):size(A,1),(n-p+1):size(A,2));
C1=C(:,1:(n-p));C2=C(:,(n-p+1):n);
% Step 1
N1=zeros((n-q),(n-p));
N2=eye((n-q),p);
N3=eye((n-p),(n-p));
N4=zeros((n-p),p);
N=[N1 N2;N3 N4];
% Step 2
M1=eye((n-p),(n-q));
M2=zeros((n-p),(n-p));
M3=eye((n-q),(n-q));
M4=zeros((n-q),(n-p));
M=[M1 M2;M3 M4];
% Step 3
I1=eye((n-p),(n-p));
I2=zeros((n-p),(n-q));
I3=zeros((n-q),(n-p));
I4=eye((n-q),(n-q));
I=[I1 I2;I3 I4];
P1=zeros((n-p),(n-q));
P2=zeros((n-p),(n-p));
P3=eye((n-q),(n-q));
P4=-(pinv(c2))*c1;
P=[P1 P2;P3 P4];
Q1=eye((n-p),p);
Q2=zeros((n-q),p);
Q=[Q1;Q2];
% Step 4
F1=M1*A4-M1*A3*pinv(C1)*C2;
F2=zeros((n-p),(n-p));
F3=M3*A4-M3*A3*pinv(C2)*C1;
F4=zeros((n-q),(n-p));
F=[F1 F2;F3 F4];
H1=M1*A3*pinv(C1);
H2=M3*A3*pinv(C1);
H=[H1;H2];
% Step 5
G=M*N*B; s=rand(1)+randn(1)*i;
MAT=[s*eye(n)-A E;C zeros(2,2)];
rank(MAT)
if rank(MAT)~=n+q
disp ('The method isn't useful');
end
if M*N*E~=[zeros((n-p),q); zeros((n-q),q)]
disp ('First convergence condition isn't fulfilled');
end
if M*N*B-G~=zeros(n,m)
disp ('Second convergence condition isn't fulfilled');
end
if M*N*A-F*N~=H*C
disp ('Third convergence condition isn't fulfilled');
end
if P*N+Q*C~=I
disp ('Fourth convergence condition isn't fulfilled');
end
% K=0
K=zeros(1,n);
sim ('schalgloor');
subplot(221); plot(t,x1,'b',t,xc1,'r--');grid;
subplot(222); plot(t,x2,'b',t,xc2,'r--');grid;
subplot(223); plot(t,x3,'b',t,xc3,'r--');grid;
subplot(224); plot(t,x4,'b',t,xc4,'r--');grid;
K=KK;
h=figure;sim ('schalgloor');
subplot(221);plot(t,x1,'b',t,xc1,'r--');grid;
subplot(222);plot(t,x2,'b',t,xc2,'r--');grid;
subplot(223);plot(t,x3,'b',t,xc3,'r--');grid;
subplot(224);plot(t,x4,'b',t,xc4,'r--');grid;
```

5.2 ALGLOOR program (rocket move)

```
% Rocket move s=10
run ALGX10;
close all;clear C;clear contor;clear e;
clear P1;clear P2;clear P3;clear P4;
clear P;clear P;clear k1;
KK=K;
% Calculus data
n=size(A,1);m=size(B,2);
q=size(E,2);p=size(C,1);
x0=[10;0;0;1]/100;
xc0=[1;10;-1;2]/100;
up=randn(2,1)
% Matrices A and C partition
A1=A(1:(n-p),1:(n-p));
A2=A(1:(n-p),(n-p+1):size(A,2));
A3=A((n-p+1):size(A,1),1:(n-p));
A4=A((n-p+1):size(A,1),(n-p+1):size(A,2));
C1=C(:,1:(n-p));
C2=C(:,(n-p+1):n);
% Step 1
N1=zeros((n-q),(n-p));
N2=eye((n-q),p);
N3=eye((n-p),(n-p));
N4=zeros((n-p),p);
N=[N1 N2;N3 N4];
% Step 2
M1=zeros((n-p),(n-q));
M2=eye((n-p),(n-p));
```

```
M3=zeros((n-q),(n-q));
M4=eye((n-q),(n-p));
M=[M1 M2;M3 M4];
% Step 3
I1=eye((n-p),(n-p));
I2=zeros((n-p),(n-q));
I3=zeros((n-q),(n-p));
I4=eye((n-q),(n-q));
I=[I1 I2;I3 I4];
P1=zeros((n-p),(n-q));
P2=eye((n-p),(n-p));
P3=zeros((n-q),(n-q));
P4=-(pinv(C2))*C1;
P=[P1 P2;P3 P4];
Q1=zeros((n-p),p);
Q2=pinv(C2);
Q=[Q1;Q2];
% Step 4
F1=zeros((n-p),(n-q));F2=A1;
F3=zeros((n-q),(n-q));F4=A1;
F=[F1 F2;F3 F4];
H1=A2;H2=A2;
H=[H1;H2];
% Step 5
G=M*N*B;
% Tests
s=rand(1)+randn(1)*i;
MAT=[s*eye(n)-A E;C zeros(2,2)];
if rank(MAT)~=n+q
disp ('The method isn't useful');
end
if M*N*E~=[zeros((n-p),q);zeros((n-q),q)]
disp ('First convergence condition isn't fulfilled');
end
if M*N*B-G~=zeros(n,m)
disp ('Second convergence condition isn't fulfilled');
end
if M*N*A-F*N~=H*C
disp ('Third convergence condition isn't fulfilled');
end
if P*N+Q*C~=I
disp ('Fourth convergence condition isn't fulfilled');
end
% K=0 (without controller)
K=zeros(1,n); sim('schrach');
subplot(221);plot(t,x1,'b',t,xc1,'r--');grid;
subplot(222);plot(t,x2,'b',t,xc2,'r--');grid;
subplot(223);plot(t,x3,'b',t,xc3,'r--');grid;
subplot(224);plot(t,x4,'b',t,xc4,'r--');grid;
h=figure;
subplot(221);plot(t,e1);grid;
subplot(222);plot(t,e2);grid;
subplot(223);plot(t,e3);grid;
subplot(224);plot(t,e4);grid;
% K=K (with controller)
K=KK;h=figure;sim('schrach');
subplot(221);plot(t,x1,'b',t,xc1,'r--');grid;
subplot(222);plot(t,x2,'b',t,xc2,'r--');grid;
subplot(223);plot(t,x3,'b',t,xc3,'r--');grid;
subplot(224);plot(t,x4,'b',t,xc4,'r--');grid;
h=figure;
subplot(221);plot(t,e1);grid;
subplot(222);plot(t,e2);grid;
subplot(223);plot(t,e3);grid;
subplot(224);plot(t,e4);grid;
```

5.3 ALGLX program

```
close all;
A=[-0.007 0.012 -9.81 0; -0.128 -0.54 0 1;
   0 0 0 1; 0.065 0.96 0 -0.99];
B=[0;-0.04;0;-12.5];
Q=[10 0 0 0;0 10 0 0;0 0 100 0;0 0 0 1]; R=[2];
[K,P,E] = LQR(A,B,Q,R);
I2=[1 0;0 1];
N3=randn(4,3);
contor=1;
T(:,1)=B(:,1);
for i=1:4
   for j=1:3
   T(i,j+1)=N3(i,j);
   end
end
Ab=(inv(T))*A*T;
Bb=(inv(T)*B);
Kb=place(Ab,Bb,E);
e=eig(Ab-Bb*Kb);
k1=Kb(1);k21=Kb(2);
k22=Kb(3);k23=Kb(4);
r1=5;Rb=[r1];R=Rb;
Pb=r1*[k1 k21 k22 k23; k21 10 0;
      k22 0 1 0; k23 0 0 1];
ee=eig(Rb);
Qb=-(Pb*Ab+(transpose(Ab))*Pb-Pb*Bb*Kb);
PPP=transpose(inv(T))*Pb*inv(T);
KKK=inv(R)*transpose(B)*PPP;
EEE=eig(AB*KKK);
m=rank(T);
while real(EEE(1))>0 | real(EEE(2))>0 |
     real(EEE(3))>0 | real(EEE(4))>0 | m<4
N3=randn(4,3);
contor=contor+1;
T(:,1)=B(:,1)
for i=1:4
  for j=1:3
  T(i,j+1)=N3(i,j);
  end
end
Ab=(inv(T))*A*T;
Bb=(inv(T)*B);
```

```
Kb=place(Ab,Bb,E);
e=eig(Ab-Bb*Kb);
k1=Kb(1);k21=Kb(2);
k22=Kb(3);k23=Kb(4);
r1=1;Rb=[r1];R=Rb;
Pb=r1*[k1 k21 k22 k23; k21 10 0;
        k22 0 1 0; k23 0 0 1];
ee=eig(Rb);
Qb=-(Pb*Ab+(transpose(Ab))*Pb-Pb*Bb*Kb);
PPP=transpose(inv(T))*Pb*inv(T);
KKK=inv(R)*transpose(B)*PPP;
EEE=eig(AB*KKK);
m=rank(T);
end
Q=transpose(inv(T))*Qb*inv(T); R=Rb;
[KK,PP,EE] = LQR(A,B,Q,R);
```

5.4 ALGLX10 program

```
close all;clear all;
d1=1.5;d2=40;d3=-20;d4=1.2;V=400;T1=1/d1;
Tv=T1; w0=sqrt(d1*d4-d3);csi=(d1+d4)/(2*w0);
kd=d2; a0=w0^2-(2*csi*w0)/Tv+1/(Tv^2);
a1=2*csi*w0-1/Tv;k0=70;
A=[0 1 0 0;0 0 V/Tv 0;0 0 -1/Tv 1;0 0 -a0 -a1];
B=[0;0;0;kd];
C=[0 0 1 0;0 0 0 1];
E=[0 0;0 0;1 0;0 1];
Q=[10 0 0 0;0 10 0 0;0 0 100 0;0 0 0 1];R=[2];
[K,P,E] = LQR(A,B,Q,R); I2=[1 0;0 1];
% Matrix T definition
N3=randn(4,3);
contor=1; T(:,1)=B(:,1);
for i=1:4
  for j=1:3
    T(i,j+1)=N3(i,j);
  end
end
Ab=(inv(T))*A*T;
Bb=(inv(T)*B);
Kb=place(Ab,Bb,E);
e=eig(Ab-Bb*Kb);
k1=Kb(1);k21=Kb(2);
k22=Kb(3);k23=Kb(4);
r1=1;Rb=[r1];R=Rb;
Pb=r1*[k1 k21 k22 k23;k21 1 0 0;
       k22 0 1 0;k23 0 0 1];
ee=eig(Rb);
Qb=-(Pb*Ab+(transpose(Ab))*Pb-Pb*Bb*Kb);
PPP=transpose(inv(T))*Pb*inv(T);
KKK=inv(R)*transpose(B)*PPP;
EEE=eig(A-B*KKK);m=rank(T);
while real(EEE(1))>0 | real(EEE(2))>0 |
     real(EEE(3))>0 | real(EEE(4))>0 | m<4
% Matrix T definition
N3=randn(4,3);
contor=contor+1;
T(:,1)=B(:,1);
for i=1:4
  for j=1:3
    T(i,j+1)=N3(i,j);
  end
end
Ab=(inv(T))*A*T;
Bb=(inv(T)*B);
Kb=place(Ab,Bb,E);
e=eig(Ab-Bb*Kb);
k1=Kb(1);k21=Kb(2);k22=Kb(3);k23=Kb(4);
r1=1;Rb=[r1];R=Rb;
Pb=r1*[k1 k21 k22 k23;k21 1 0 0;
       k22 0 1 0;k23 0 0 1];
ee=eig(Rb);
Qb=-(Pb*Ab+(transpose(Ab))*Pb-Pb*Bb*Kb);
PPP=transpose(inv(T))*Pb*inv(T);
KKK=inv(R)*transpose(B)*PPP;
EEE=eig(A-B*KKK);m=rank(T);
end
% Variant 1
Q=transpose(inv(T))*Qb*inv(T);R=Rb;
[KK,PP,EE] = LQR(A,B,Q,R);
```

6 Conclusion

A new algorithm for project of a reduced order observer is presented; it may be used in different applications and in case of aircrafts' automat command system. Theoretical results are validated by numerical simulations using calculus program made by authors using dynamic models of aircrafts, rockets' movement equations and optimal control laws.

References:

[1] Donald Mc.L. *Automatic Flight Control Systems.* New York, London, Toronto, Sydney, Tokyo, Singapore, 1990.

[2] Sîngeorzan D. *Regulatoare adaptive*. Editura Militară, Bucureşti, 1992.

[3] Yuan Y., Yu P., Librescu L., Marzocca P. Aero elasticity of Time–Delayed Feedback Control of Two–Dimensional Supersonic Lifting Surfaces. *Journal of Guidance, Control and Dynamics.* Vol. 27, nr. 5, 2004, pp. 795–804.

[4] Hanlon P.D., Maybeck P.S. Multiple–Model Adaptive Estimation Using a Residual Correlation Kalman Filter Bank. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, nr. 2, 2000, pp. 393–406.

[5] Kalman R., Falb P., Arbib M.A. *Teoria sistemelor dinamice*. Editura Tehnică, Bucureşti, 1975.

[6] Kulcsar B. LQR/LTR Controller Design for An Aircraft Model. *Periodica Politechnica Ser. Transp. Eng.*, Vol. 28, 2000, pp. 131–142.

[7] Hovakimyan N., Calise A.J., Madyastha V.K. An Adaptive Observer Design Methodology for Bounded Nonlinear Process. *In Proceedings of Conference on Design and Control, Las Vegas*, 2002, pp. 4700-4705.

[8] Jeong C.S., Yaz E.E., Bahakeem, A., Yaz, I. Nonlinear Observer Design with General Criteria. *International Journal of Innovative Computing, Information and Control*, Vol. 2, nr. 4, 2006.

[9] Juloski A.L., Mihajlovic W.P., Heemels M.H., Nijmeijer H. Observer Design for Experimental Rotor with Discontinuous Friction. *In Proceedings of the 2006 American Control Conference, Minneapolis, Minnesota, USA*, June 1–16, 2006.

[10] Lavretky E., Hovakimyan N., Calise A.J. Reconstruction of Continuous – Time Dynamics Using Delayed Output and Feed forward Neural Networks. *IEEE Transactions on Automatic Control*, September 2003.

[11] Lu J., Feng C., Xu S., Chu Y. Observer Design for a Class of Uncertain State – Delay Nonlinear Systems. *International Journal of Control, Automation and Systems*, Vol.4, nr. 4, 2006, pp. 448–455.

[12] Darouch, M. Linear Functional Observers for Systems with Delays in Stable Variables. *IEEE Transactions on Automatic Control*, Vol. 46, nr. 3, 2001, pp. 491–496.

[13] Hou M., Zitek P., Patton R.J. An Observer Design for Linear Time – Delay Systems. *IEEE Transactions on Automatic Control*, Vol. 47, nr. 1, 2002, pp. 121–125.

[14] Wang Z.D., James L., Burnham K.J. Stability Analysis and Observer Design for Neutral Delay Systems. *IEEE Transactions on Automatic Control*, Vol. 47, nr. 3, 2002, pp. 478 – 482.

[15] Naser Prljaca, Zoran Gajic, Optimal Control & Filtering of Weakly Coupled Linear Discrete Stochastic Systems by The Eigenvector Approach, *WSEAS Transactions on systems and control*, Volume 2, September 2007, pp. 435-441.

[16] Tain-Sou Tsay, Automatic Gain Control for Unity Feedback Control Systems with Large Parameters Variations, *WSEAS Transactions on systems and control*, Volume 2, December 2007, pp. 537–545.

[17] Krzeminsky S., Kaczorec T. Perfect Reduced – Order Unknown – Input Observer for Standard Systems. *Bulletin of the Polish Academy of Sciences Technical Sciences*, Vol. 52, nr.2, 2004, pp. 103–107.

[18] 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.

[19] Hui S., Zak S.H. Observer Design for Systems with Unknown Inputs. *Int. J. Appl. Math. Comput. Sci. 2005*, Vol. 15, nr. 4, pp. 431–446.

[20] Kim B.S., Calise A.J. Nonlinear Flight Control Using Neural Networks. *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Scottsdale, AZ, pp. 930–940, 1994.

[21] Krzeminski S., Kaczorec T. Perfect Reduced – Order Unknown–Input Observer for Descriptor Systems. *7th International Multi-conference on Informatics, Systemic and Cybernetics*, Orlando, 2003.

[22] Koray Kosar, Seher Durmaz, Elbrous Jafarov, Longitudinal Dynamics Analysis of Boeing 747-400, *Proceedings of the 9th WSEAS International Conference on Automatic Control, Modelling & Simulation*, Istanbul, Turkey, May 27-29, 2007, pp. 82–87.

[23] Lungu R., Lungu M. Optimal Control of the Rocket's Lateral Deviation in Rapport with Equal Signal Line. *A 6–a Conferinta Internaționala de Sisteme Electromecanice si Energetice*, SIELMEN, 2–5 September 2007, Chisinau, Moldova Republic.

[24] Lungu R. *Sisteme de dirijare aerospațială*. Editura Sitech, Craiova, 2002

[25] Lungu M. Calculul vitezelor de apropiere şi deviație de la linia țintei. *The International Symposium on Naval and Marine Education. NAV·MAR·EDU 2001*, Constanța, May 24–26, 2001, pp. 113–119.

[26] Lungu R., Dincă L., Lungu M. Dinamical Errors Calculus of Some Rockets Guidance Systems Using the Direct Method in Three Points. *The 3rd International Conference on Advanced Engineer Design, Praga*, June, 2003, CD–6 pp.

[27] Aron I., Lungu R. *Automate de stabilizare şi dirijare*. Editura Militară, Bucureşti, 1991.