Optimal Commnad of Aircrafts' Move Using a Reduced Order Observer

LUNGU MIHAI
Avionics Department
University of Craiova, Faculty of Electrotechnics
Blv. Decebal, No.107, Craiova, Dolj

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

ROMANIA
JULA NICOLAE
Military Technical Academy of Bucharest
Blv. George Coşbuc, no. 81 – 83, Sector 5
ROMANIA

CALBUREANU MADALINA
University of Craiova, Faculty of Mechanics

ROMANIA
CEPISCA COSTIN
Electrical Engineering Faculty
University POLITEHNICA of Bucharest
313 Splaiul Independentei, Sector 6
ROMANIA

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

Key-Words: algorithm, optimal, control law, observer.

1 Dynamic models of aircraft and reduced order observer

Aircraft dynamic may be described by equations

$$\dot{x} = Ax + Bu + Eu_p, \tag{1}$$

$$y = Cx, \tag{2}$$

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),

$A-$ 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 [1]

$$M\dot{z} = Fz + Gu + Hy, \tag{3}$$

$$\hat{x} = Pz + Qy, \tag{4}$$

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

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

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,

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

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

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

one obtains the equation of error e

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

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

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

if

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

$I-$ unity matrix.

2 Project algorithm of the reduced order observer (ALGLOOR)

First of all one chooses matrix N of form

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

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

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

With E of form

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

equation (8) is equivalent with system

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

Choosing

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

one results

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

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

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

With H and A of forms

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

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

condition (7) is equivalent with system

$$M_1A_3 = H_1C_1, \quad M_1A_4 - F_1 = H_1C_2, \quad M_3A_3 = H_2C_1, \quad M_3A_4 - F_3 = H_2C_2; \qquad (20)$$

one results

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

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

One yield

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

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

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

Condition (11) is equivalent with system

$$P_2 + Q_1C_1 = I_{(n-p)\times(n-p)}, \quad P_1 + Q_1C_2 = 0_{(n-p)\times(n-q)}, \quad P_4 + Q_2C_1 = 0_{(n-q)\times(n-p)}, \quad P_3 + Q_2C_2 = I_{(n-q)\times(n-q)}. \qquad (25)$$

For example, one chooses

$$P_1 = 0_{(n-p)\times(n-q)},\quad P_3 = 0_{(n-q)\times(n-q)} \tag{26}$$

and equations (26) become

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

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

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 $p = \mathrm{rang}\, C, q = \mathrm{rang}\, E$ 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).

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.

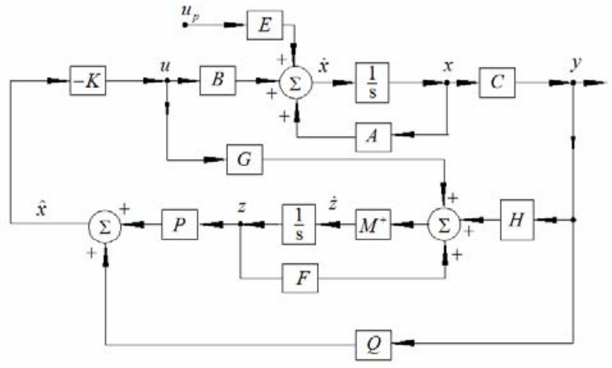

Fig.1

One remarks that the algorithm ALGLOOR doesn't use non singular transformations like other algorithms [2], [3], [4].

3 Numeric examples

A reduced order observer for state $\hat{x}$ estimation described by equation (31) from [5], 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 appendix; one obtained matrices

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

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

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

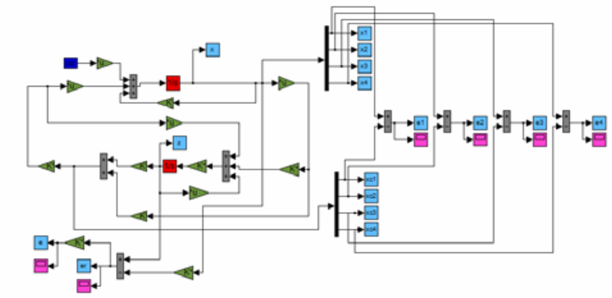

Fig.2

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

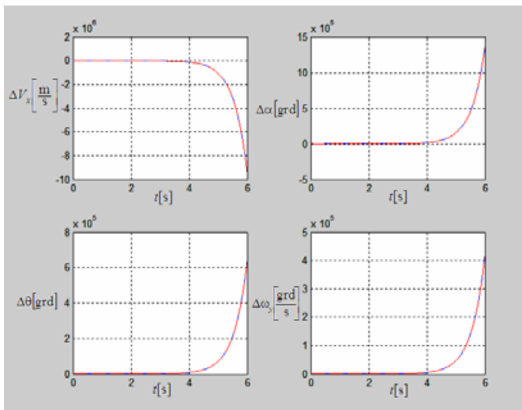

Fig.3

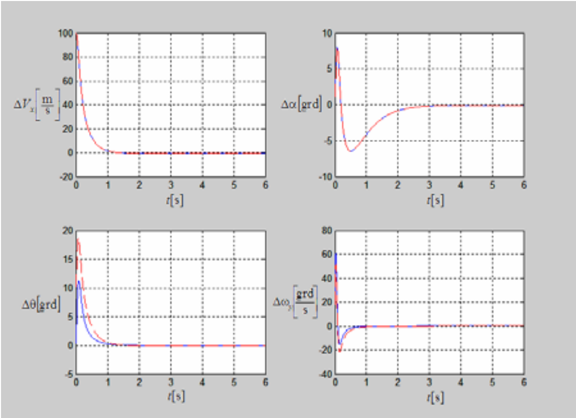

Fig.4

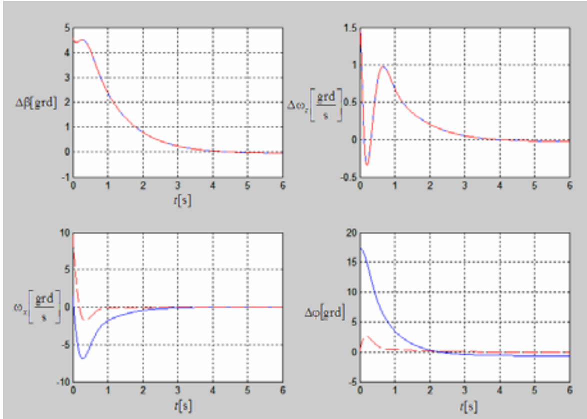

Fig.6

Similar, for lateral move described by equation [6]

$$\begin{bmatrix}\Delta\dot{\beta}\\ \Delta\dot{\omega}_z\\ \Delta\dot{\omega}_x\\ \Delta\dot{\varphi}\end{bmatrix}=\begin{bmatrix}-0.0558 & -0.9968 & 0.0802 & 0.0415\\ 0.598 & -0.115 & -0.0318 & 0\\ 0.305 & 0.388 & -0.465 & 0\\ 0 & 0.0805 & 1 & 0\end{bmatrix}\begin{bmatrix}\Delta\beta\\ \Delta\omega_z\\ \Delta\omega_x\\ \Delta\varphi\end{bmatrix}+\begin{bmatrix}0.0073 & 0\\ -0.475 & 0.123\\ 0.153 & 1.063\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_d\\ \delta_e\end{bmatrix}, \quad (32)$$

$C=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}$ and $E^T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, n=4, p=q=2,$

one obtains

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

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

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

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

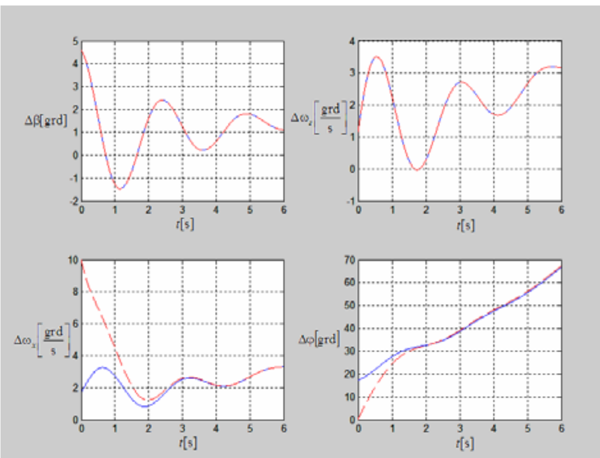

Fig.5

4 Appendix

```
clear all; run prog2sec; close all;
clear C; clear contor;
clear e; clear i;
clear r; clear m;
clear P; clear Q;
KK=K;
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];
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];      % up=randn(2,1)
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));
P3=eye((n-q),(n-q));
P4=-(pinv(c2))*c1;
P2=zeros((n-p),(n-p));
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;
F3=M3*A4-M3*A3*pinv(C2)*C1;
F2=zeros((n-p),(n-p));
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;
```

Program prog2sec

```
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(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
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(A
B*KKK);m=rank(T);
end
contor
Q=transpose(inv(T))*Qb*inv(T);R=Rb;
[KK,PP,EE] = LQR(A,B,Q,R);
```

5 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 available by numerical simulations using calculus program made by authors using dynamic models of aircrafts and optimal control laws.

References:

[1] 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, pag. 103 – 107.

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

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

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

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

[6] Lungu, R., Lungu, M. *Optimal Control of the Rocket's Lateral Deviation in Rapport with Equal Signal Line.* A 6 – a Conferinta Interna-ționala de Sisteme Electromecanice si Ener-getice, SIELMEN, 2 – 5 September 2007, Chi-sinau, Moldova Republic, 7 pag.