

PROIECTAREA CONTROLLERELOR CONVENȚIONALE PENTRU CONTROLUL ABATERII LATERALE A AERONAVELOR

Lucrarea prezintă două sisteme de comandă automată a deplasării laterale întrucât deplasarea laterală a avionului poate fi comandată prin modificarea direcției de zbor ψ și/sau a unghiului de înclinare φ . Se pleacă de la modelele matriceale ale celor două mișcări și se proiectează legi de comandă pentru stabilizarea sistemelor. Se face, de asemenea, identificarea sistemelor utilizându-se o rețea neuronală de tip feedforward ce va fi antrenată până în momentul în care ieșirea acestei rețele neuronale va tinde către ieșirea sistemului condus (sistemul ce trebuie identificat).

1. CHESTIUNI TEORETICE

Deplasarea laterală a avionului poate fi comandată prin modificarea direcției de zbor ψ și/sau a unghiului de înclinare φ [21], [25], [26].

Comanda automată a deplasării laterale prin modificarea direcției de zbor

Considerându-se $\omega_x = \varphi = 0$ și $\delta_e = z_2 = 0$, mișcarea laterală a aeronavei este descrisă de ecuațiile

$$\begin{bmatrix} \dot{\beta} & \dot{\omega}_z & \dot{\psi} \end{bmatrix}^T = A \begin{bmatrix} \beta & \omega_z & \psi \end{bmatrix}^T + B \begin{bmatrix} z_1 & z_3 & \delta_d \end{bmatrix}^T, \quad (1.1)$$

cu

$$A = \begin{bmatrix} a_{31} & a_{13} & 0 \\ a_{31} & a_{33} & 0 \\ 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & b_{3d} \\ 0 & 0 & 0 \end{bmatrix}, \quad (1.2)$$

la care se anexează și ecuația deplasării laterale a centrului de greutate al A

$$\dot{y} = -\beta + \psi + v_y. \quad (1.3)$$

Coefficienții mișcării laterale a trei tipuri de aeronave, la diferite regimuri de zbor sunt prezentați în tabelul 1.1

Incluzând β în ψ , această ecuație devine

$$\dot{y} = \psi + v_y; \quad (1.4)$$

modelul matematic al mișcării laterale devine

$$\begin{bmatrix} \dot{\beta} \\ \dot{\omega}_z \\ \dot{\psi} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{13} & 0 & 0 \\ a_{31} & a_{33} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \omega_z \\ \psi \\ y \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & b_d & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_3 \\ \delta_d \\ v_y \end{bmatrix}, \quad (1.5)$$

echivalentă cu sistemul de ecuații

$$\begin{aligned} \dot{\beta} &= a_{11}\beta + a_{13}\dot{\psi} + z_1, \\ \ddot{\psi} &= a_{31}\beta + a_{33}\dot{\psi} + b_d\delta_d + z_3, \\ \dot{y} &= \psi + v_y. \end{aligned} \quad (1.6)$$

Tabelul 1.1

Tipul avionului/ Regimul de zbor	Coeficienți				
	n_{11}	n_{13}	n_{31}	n_{33}	n_d
Avion ușor H=10 km; M=0.8	0.097	-1	5.76	0.22	3.18
Avion mediu H=10 km; M=0.9	0.26	-1	69	0.89	29.6
Avion greu H=12 km; M=0.9	0.3	-1	50	0.9	19

Eliminând β între primele două ecuații (1.6), rezultă următoarele

$$\begin{aligned} \ddot{\psi} + a'_2\dot{\psi} + a'_1\psi &= b_d\dot{\delta}_d - a_{11}b_d\delta_d + a_{31}z_1 + \dot{z}_3 - a_{11}z_3, \\ \dot{y} &= \psi + v_y. \end{aligned} \quad (1.7)$$

în care:

$$\begin{aligned} a'_2 &= -(a_{11} + a_{33}), \\ a'_1 &= a_{11}a_{33} - a_{13}a_{31}. \end{aligned} \quad (1.8)$$

Legea de comandă se alege de forma

$$\delta_d = k_d^y(\bar{y} - y) - k_d^{\dot{y}}\dot{y} - k_d^{\psi}\psi - k_d^{\dot{\psi}}\dot{\psi}. \quad (1.9)$$

Prima dintre ecuațiile (1.7) poate fi exprimată sub forma matriceală – vectorială

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -a'_1 & -a'_2 \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} + b_d \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -a_{11} & 1 \end{bmatrix} \begin{bmatrix} \delta_d \\ \dot{\delta}_d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ a_{31} \end{bmatrix} z_1 + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -a_{11} & 1 \end{bmatrix} \begin{bmatrix} z_3 \\ \dot{z}_3 \end{bmatrix}. \quad (1.10)$$

Derivând ecuația (1.9), se obține

$$\dot{\delta}_d = k_d^y(\dot{\bar{y}} - \dot{y}) - k_d^{\dot{y}}\dot{y} - k_d^{\psi}\dot{\psi} - k_d^{\dot{\psi}}\ddot{\psi}. \quad (1.11)$$

Schema bloc a sistemului de comandă automată a deplasării laterale prin modifi-

careia direcției de zbor este dată în fig.1.1.

Ținând seama de relațiile (1.9) și (1.11), ecuația (1.10) devine

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a_{11}b_dk_d^y & -[a_1' + b_d(k_d^y - a_{11}k_d^y)] & -(a_2' + b_dk_d^y) \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \\ \ddot{y} \end{bmatrix} + b_d \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ a_{11}k_d^y & (a_{11}k_d^y - k_d^y) & -k_d^y \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} + b_dk_d^y \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -a_{11} & 1 \end{bmatrix} \begin{bmatrix} \bar{y} \\ \dot{\bar{y}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ a_{31} \end{bmatrix} z_1 + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -a_{11} & 1 \end{bmatrix} \begin{bmatrix} z_3 \\ \dot{z}_3 \end{bmatrix}. \quad (1.12)$$

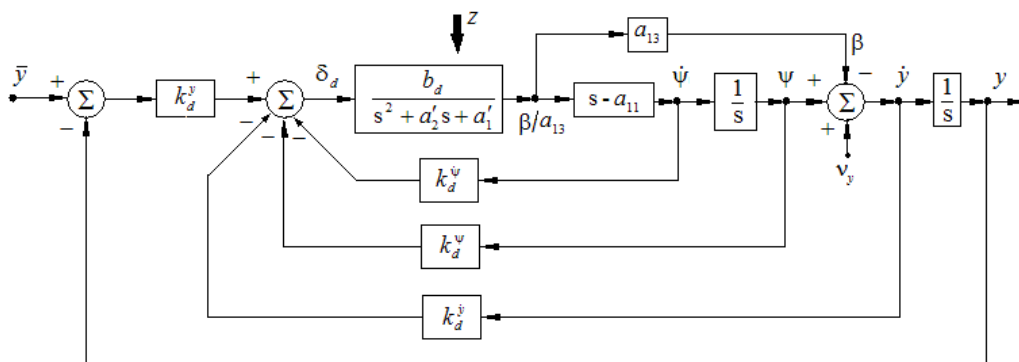


Fig.1.1 Sistem de comandă automată a deplasării laterale prin modificarea direcției de zbor

Din ecuația anterioară, prin aplicarea transformării Laplace în condiții inițiale nule, se obține ecuația

$$(s^4 + a_3s^3 + a_2s^2 + a_1s + a_0)y = b_dk_d^y(s - a_{11})\bar{y} + z(s), \quad (1.13)$$

în care

$$\begin{aligned} a_3 &= a_2' + b_dk_d^y, a_2 = a_1' + b_d(k_d^y - a_{11}k_d^y + k_d^y), \\ a_1 &= b_d(k_d^y - a_{11}k_d^y - a_{11}k_d^y), a_0 = -a_{11}b_dk_d^y, \\ z(s) &= a_{31}z_1 + (s - a_{11})z_3 + \{s^3 + (a_2' + b_dk_d^y)s^2 + [a_1' + b_d(k_d^y - a_{11}k_d^y)]s - a_{11}b_dk_d^y\}v_y. \end{aligned} \quad (1.14)$$

Funcția de transfer a sistemului în circuit închis este

$$H_0(s) = \frac{y(s)}{\bar{y}(s)} = \frac{b_dk_d^y(s - a_{11})}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} = \frac{b_dk_d^y(s - a_{11})}{s^4 + A_3\omega_0s^3 + A_2\omega_0^2s^2 + A_1\omega_0^3s + \omega_0^4}. \quad (1.15)$$

Impunând valori pentru A_3, A_2, A_1, ω_0 , din ecuațiile (1.14) și (1.15) se determină expresiile rapoartelor de transmisie [29].

Ecuațiile (1.14) sunt echivalente cu următoarele

$$\begin{aligned}
b_d k_d^{\psi} &= A_3 \omega_0 - a'_2, \\
b_d k_d^{\dot{y}} + b_d k_d^{\psi} - a_{11} b_d k_d^{\psi} &= A_2 \omega_0^2 - a'_1, \\
b_d k_d^y - a_{11} b_d k_d^{\dot{y}} - a_{11} b_d k_d^{\psi} &= A_1 \omega_0^3, \\
-a_{11} b_d k_d^y &= \omega_0^4.
\end{aligned} \tag{1.16}$$

Exprimat sub forma $A \cdot x = b$, sistemul anterior devine

$$\begin{bmatrix} 0 & 0 & 0 & b_d \\ 0 & b_d & b_d & -a_{11} b_d \\ b_d & -a_{11} b_d & -a_{11} b_d & 0 \\ -a_{11} b_d & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} k_d^y \\ k_d^{\dot{y}} \\ k_d^{\psi} \\ k_d^{\psi} \end{bmatrix} = \begin{bmatrix} A_3 \omega_0 - a'_2 \\ A_2 \omega_0^2 - a'_1 \\ A_1 \omega_0^3 \\ \omega_0^4 \end{bmatrix}. \tag{1.17}$$

Rezolvarea acestui sistem se face în Matlab utilizând secvența *lsqr* din cadrul programului din anexă.

Comanda automată a deplasării laterale prin modificarea unghiului de ruliu

Deplasarea laterală plană a centrului de greutate al A nu este eficientă prin controlul unghiului de direcție ψ . Mai indicată este comanda automată a deplasării laterale prin controlul unghiului de ruliu φ , deoarece astfel se crează forțe laterale mai mari.

Se studiază cazul în care $\beta = 0$ și mișcarea de ruliu este rapidă (momentul de înclinare în ruliu este considerat neglijabil). În acest caz particular ecuațiile de mișcare sunt

$$\dot{\psi} = -a_{14} \varphi, \dot{y} = \psi, \tag{1.18}$$

sistem echivalent cu ecuația

$$\ddot{y} = -a_{14} \varphi. \tag{1.19}$$

Deoarece la echilibru momentul de ruliu este nul, rezultă

$$\delta_e = 0 = k_e^y (\bar{y} - y) - k_e^{\dot{y}} \dot{y} + k_e^{\varphi} \varphi \tag{1.20}$$

și, din aceasta, se obține

$$-\varphi = k_y \bar{y} - (k_y + k_y s) y, \tag{1.21}$$

în care rapoartele de transmisie k_y și $k_{\dot{y}}$ au expresiile

$$k_y = \frac{k_e^y}{k_e^{\varphi}}, k_{\dot{y}} = \frac{k_e^{\dot{y}}}{k_e^{\varphi}}. \tag{1.22}$$

Eliminând φ între ecuațiile (1.19) și (1.20), se obține ecuația

$$\ddot{y} + a_1 \dot{y} + a_0 y = a_0 \bar{y}, \tag{1.23}$$

în care $a_1 = a_{14}k_y$, $a_0 = a_{14}k_y$. Funcția de transfer în circuit închis este

$$H_0(s) = \frac{y(s)}{\bar{y}(s)} = \frac{a_0}{s^2 + a_1s + a_0}. \quad (1.24)$$

Impunând valori pentru ξ și ω_0 , se obțin valorile rapoartelor de transmisie adimensionale

$$\hat{k}_y = \frac{\omega_0^2}{a_{14}}, \hat{k}_{\dot{y}} = \frac{2\xi\omega_0}{a_{14}} \quad (1.25)$$

și valorile rapoartelor de transmisie dimensionale

$$k_y = \frac{\hat{k}_y}{\tau_a V^*}, k_{\dot{y}} = \frac{\hat{k}_{\dot{y}}}{V^*}, \quad (1.26)$$

V^* fiind viteza de zbor a aeronavei, iar τ_a – constanta de timp aerodinamică.

2. MODUL DE LUCRU

Se realizează în Matlab programul din anexă. Acesta studiază ambele cazuri de sisteme de comandă a deplasării laterale descrise mai sus.

Comanda automată a deplasării laterale prin modificarea direcției de zbor

Pentru primul dintre sisteme se consideră cazul mișcării laterale pentru 3 tipuri de aeronave și regimuri de zbor (tabelul 1.1). Determinarea celor patru rapoarte de transmisie se face în Matlab prin rezolvarea sistemului de ecuații (1.17) – comanda *lsqr*. Sistemul descris anterior are ca mărime de intrare valoarea impusă a deplasării laterale \bar{y} , iar ca mărimi de ieșire deplasarea laterală y , derivata acesteia \dot{y} , unghiul de direcție ψ și viteza unghiulară de rotație ω_z . Vor rezulta așadar 4 funcții de transfer (în Matlab se face un ciclu *for* după $j; j=1:4$)

$$\begin{aligned} \frac{y(s)}{\bar{y}(s)} &= \frac{b_d k_d^y s - a_{11} b_d k_d^y}{(a'_2 + b_d k_d^\psi) s^3 + [a'_1 + b_d (k_d^\psi - a_{11} k_d^\psi + k_d^y)] s^2 + b_d (k_d^y - a_{11} k_d^y - a_{11} k_d^\psi) s - a_{11} b_d k_d^y}, \\ \frac{\dot{y}(s)}{\bar{y}(s)} &= \frac{-a_{13} b_d k_d^y s^3 + (b_d k_d^y + a_{11} a_{13} b_d k_d^y) s^2 - a_{11} b_d k_d^y s}{(a'_2 + b_d k_d^\psi) s^3 + [a'_1 + b_d (k_d^\psi - a_{11} k_d^\psi + k_d^y)] s^2 + b_d (k_d^y - a_{11} k_d^y - a_{11} k_d^\psi) s - a_{11} b_d k_d^y}, \\ \frac{\psi(s)}{\bar{y}(s)} &= \frac{b_d k_d^y s^2 - a_{11} b_d k_d^y s}{(a'_2 + b_d k_d^\psi) s^3 + [a'_1 + b_d (k_d^\psi - a_{11} k_d^\psi + k_d^y)] s^2 + b_d (k_d^y - a_{11} k_d^y - a_{11} k_d^\psi) s - a_{11} b_d k_d^y}, \\ \frac{\omega_z(s)}{\bar{y}(s)} &= \frac{b_d k_d^y s^3 - a_{11} b_d k_d^y s^2}{(a'_2 + b_d k_d^\psi) s^3 + [a'_1 + b_d (k_d^\psi - a_{11} k_d^\psi + k_d^y)] s^2 + b_d (k_d^y - a_{11} k_d^y - a_{11} k_d^\psi) s - a_{11} b_d k_d^y}. \end{aligned}$$

Pentru acest sistem se impun $\omega_0 = 2.5, A_1 = 4, A_2 = 6, A_3 = 4$.

Identificarea celor trei sisteme (corespunzătoare celor trei regimuri de zbor) se face aici utilizând metoda rețelelor neuronale. Se alege ca rețea neuronală o rețea de tip

feedforward cu 3 neuroni de intrare, 5 neuroni pe stratul ascuns și un neuron de ieșire. Se antrenează rețeaua neuronală 1000 de epoci sau până când eroarea (diferența dintre ieșirea sistemului condus și ieșirea NN) atinge valoarea dorită impusă 10^{-4} [27].

Prin intermediul programului Matlab (pentru cele trei tipuri de aeronave și regimuri de zbor) se obțin:

- rapoartele de transmisie adimensionale (kdy , $kdyp$, $kdpsi$, $kdpsip$) și dimensionale (kdc , $kdcp$, $kdyy$, $kdyypp$);
- funcțiile de transfer în circuit închis ($sysi$) și deschis ($sysd$);
- polii și zerourile sistemului în circuit închis ($poli$, $zerouri$);
- funcțiile de transfer în variabilă discretă (sys_z);
- matricele din descrierea matriceală a sistemului (A , B , C , D , respectiv A_z, B_z, C_z, D_z);
- caracteristicile de frecvență;
- funcția pondere;
- variația în timp a deplasării laterale y ;
- variația în timp a vitezei de variație a deplasării laterale \dot{y} ;
- variația în timp a unghiului de direcție (girație) ψ ;
- variația în timp a vitezei unghiulare de girație $\omega_z = \dot{\psi}$;
- ieșirea dorită a rețelei neuronale (T – target);
- ieșirile celor două sisteme (sistem condus și NN) înainte de antrenare;
- ieșirile celor două sisteme (sistem condus și NN) după antrenare.
- matricea W_1 a ponderilor pseudoneuronilor, W_2 – matricea (vectorul) ponderilor neuronilor din stratul ascuns, biasurile B_1 și B_2 , pentru neuronii din stratul ascuns și, respectiv, pentru neuronul de ieșire.

Comanda automată a deplasării laterale prin modificarea unghiului de ruliu

Se procedează asemănător și pentru sistemul de comandă automată a deplasării laterale prin modificarea unghiului de ruliu. Se utilizează ecuațiile (1.18) ÷ (1.26).

Valoarea coeficientului ce apare în ecuațiile mișcării pentru cele trei tipuri de aeronave (ciclul *for* dupa i ; $i=1:3$) se calculează cu $a_{14} = -n_{14}$, unde

- $n_{14} = -0.039$ pentru avion ușor $H = 10$ km;
- $n_{14} = -0.22$ pentru avion mediu $H = 10$ km;
- $n_{14} = -0.14$ pentru avion greu $H = 12$ km.

De această dată se calculează funcția de transfer (1.24) și se realizează modelul Matlab/Simulink din fig.1.2 ($L12_sch$). În simularea modelului Matlab/Simulink s-a ales pasul de integrare $Pas_integrare=0.1$, iar ca metodă de integrare metoda Runge – Kutta.

Identificarea celor trei sisteme (corespunzătoare celor trei regimuri de zbor) se face aici utilizând tot metoda cu rețele neuronale. Structura rețelei neuronale este aceeași cu cea din cazul comenzii deplasării laterale prin modificarea direcției. Se antrenează rețeaua neuronală timp de 5000 de epoci sau până când eroarea (diferența dintre ieșirea sistemului condus și ieșirea NN) atinge valoarea dorită impusă 10^{-8} .

Prin intermediul programului Matlab (pentru cele trei tipuri de aeronave și regi-muri de zbor) se obțin:

- rapoartele de transmisie adimensionale (k_y , k_{yp}) și dimensionale (k_{yy} , k_{yyp});
- funcțiile de transfer în circuit închis (sys_i) și deschis (sys_d);
- polii și zerourile sistemului în circuit închis ($poli$, $zerouri$);
- funcțiile de transfer în variabilă discretă (sys_z);
- matricele din descrierea matriceală a sistemului (A , B , C , D , respectiv A_z, B_z, C_z, D_z);
- caracteristicile de frecvență;
- funcția pondere;
- variația în timp a deplasării laterale y ;
- variația în timp a vitezei de variație a unghiului de ruliu φ ;
- variația în timp a unghiului de direcție (girație) ψ ;
- ieșirea dorită a rețelei neuronale (T – target);
- ieșirile celor două sisteme (sistem condus și NN) înainte de antrenare;
- ieșirile celor două sisteme (sistem condus și NN) după antrenare.
- matricea W_1 a ponderilor pseudoneuronilor, W_2 – matricea (vectorul) ponderi-lor neuronilor din stratul ascuns, biasurile B_1 și B_2 , pentru neuronii din stratul ascuns și, respectiv, pentru neuronul de ieșire.

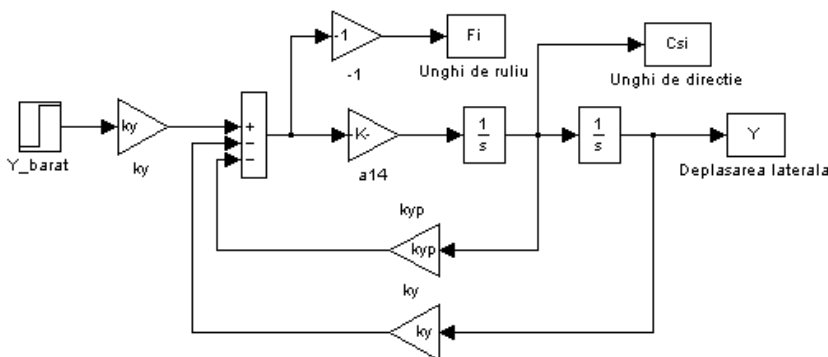


Fig.1.2 Modelul Matlab/Simulink al sistemului de comandă automată a deplasării laterale prin modificarea unghiului de ruliu

3. CHESTIUNI DE STUDIAT

Pe baza caracteristicilor grafice, a programului din anexă și a modelelor Matlab/Simulink (fig.1.2 – doar pentru sistemul de comandă a deplasării laterale prin modifi-carea unghiului de ruliu) se duc la îndeplinire următoarele sarcini de lucru:

- se calculează suprareglajul (σ) pentru cele 3 sisteme;
- se calculează durata regimului tranzitoriu (t_r) pentru cele 3 sisteme;
- se calculează coeficientul de amortizare (ξ) pentru cele 3 sisteme;

- se calculează rezervele de stabilitate RSA (Gm) [dB] și RSF (Pm) [grade] pentru cele 3 sisteme;
- se calculează frecvențele ω_π și ω_r din caracteristicile de frecvență pentru cele 3 sisteme;
- se analizează stabilitatea sistemelor pe baza funcției pondere, a criteriului de poziționare a polilor sau a frecvențelor ω_π și ω_r ;
- se obține caracteristica grafică ce reprezintă evoluția în timp a erorii de antrenare pentru avionul greu ce zboară cu $H=12$ km și $M=0.9$ pentru ambele sisteme;
- se concluzionează dacă metoda de identificare cu rețele neuronale este bună; acest lucru se face analizând cât de mari sunt diferențele între ieșirile celor două sisteme y și \hat{y} .

Se completează în tabelul de mai jos datele obținute.

Tabelul 1.2

Tip avion/ Regim de zbor	Rap.de transmisie		Poli sistem	Supra- reglajul (σ)	Durată regim tranzi- toriu (t_r)	Coeficient de amortizare (ξ)	Rezervele de stabilitate	
	Adimen- -sionale	Dimen- -sionale					RSA [dB]	RSF [grd]
Comanda automată a deplasării laterale prin modificarea direcției de zbor								
Avion ușor H=10 km; M=0.8								
Avion mediu H=10 km; M=0.9								
Avion greu H=12 km; M=0.9								
Comanda automată a deplasării laterale prin modificarea unghiului de ruliu								
Tip avion/ Regim de zbor	Rap.de transmisie		Poli sistem	Supra- reglajul (σ)	Durată regim tranzi- toriu (t_r)	Coeficient de amortizare (ξ)	Rezervele de stabilitate	
	Adimen- -sionale	Dimen- -sionale					RSA [dB]	RSF [grd]
Avion ușor H=10 km; M=0.8								
Avion mediu H=10 km; M=0.9								
Avion greu H=12 km; M=0.9								

Anexă

% Comanda automata a deplasarii laterale prin modificarea directiei de zbor

close all;clear all;

for i=1:3,

for j=1:4,

if i==1

H=10 % Avion usor, H=10 km

n11=0.097;n31=5.76;n13=-1;n33=0.22;nd=3.18;

elseif i==2

H=10 % Avion mediu, H=10 km

n11=0.26;n31=69;n13=-1;n33=0.89;nd=29.6;

else

H=12 % Avion greu, H=12 km

n11=0.3;n31=50;n13=-1;n33=0.9;nd=19;

end

a11=-n11;a31=-n31;a33=-n33;a13=-n13;bd=-nd;

a1p=a11*a33-a13*a31;a2p=-(a11+a33);A1=4;A3=4;A2=6;w0=2.5;

% Calculul rapoartelor de transmisie adimensionale

TL1=A3*w0-a2p;TL2=A2*(w0^2)-a1p;TL3=A1*(w0^3);TL4=w0^4;

AA=[0 0 0 bd;0 bd bd -a11*bd;bd -a11*bd -a11*bd 0;-a11*bd 0 0 0];

bb=[TL1;TL2;TL3;TL4];x=lsqr(AA,bb);

kdy=x(1)

kdyp=x(2)

kdpsi=x(3)

kdpsip=x(4)

a3=a2p+bd*kdpsip;a2=a1p+bd*(kdpsi-a11*kdpsip+kdyp);

a1=bd*(kdy-a11*kdyp-a11*kdpsi);a0=-a11*bd*kdy;

% Calculul rapoartelor de transmisie dimensionale

V=300;Ta=2.5;kdc=kdpsi

kdcp=Ta*kdpsip

kdyy=kdy/(Ta*V)

kdyy=kdyp/V

% Declararea celor 4 functii de transfer

if j==1 % Iesirea y

numi=[0 0 0 bd*kdy -bd*kdy*a11];

deni=[1 a2p+bd*kdpsip a1p+bd*(kdpsi-a11*kdpsip+kdyp) bd*(kdy-a11*kdyp-a11*kdpsi)
-a11*bd*kdy];

elseif j==2 % Iesirea psi

numi=[0 0 bd*kdy -bd*kdy*a11 0];

deni=[1 a2p+bd*kdpsip a1p+bd*(kdpsi-a11*kdpsip+kdyp) bd*(kdy-a11*kdyp-a11*kdpsi)
-a11*bd*kdy];

elseif j==3 % Iesirea Omegaz

numi=[0 bd*kdy -bd*kdy*a11 0 0];

deni=[1 a2p+bd*kdpsip a1p+bd*(kdpsi-a11*kdpsip+kdyp) bd*(kdy-a11*kdyp-a11*kdpsi)
-a11*bd*kdy];

else % Iesirea yp

numi=[0 0 bd*kdy -bd*kdy*a11 0]-[0 bd*kdy*a13 -bd*kdy*a11*a13 0 0];

deni=[1 a2p+bd*kdpsip a1p+bd*(kdpsi-a11*kdpsip+kdyp) bd*(kdy-a11*kdyp-a11*kdpsi)
-a11*bd*kdy];

end

```

numd=numi;dend=deni-numi;sysi=tf(numi,deni)
sysd=tf(numd,dend)
sysd=tf(numd,dend);poli=pole(sysi)
zerouri=zero(sysi);
[A,B,C,D]=tf2ss(numi,deni)
Ts=.025;sys_z=c2d(sysi,Ts);[num_z,den_z]=tfdata(sys_z,'v');
sys_z=tf(num_z,den_z)
[A_z,B_z,C_z,D_z]=tf2ss(num_z,den_z)
% Caracteristici grafice
h=figure(3*i-2);subplot(2,2,j);margin(sysd);[Gm,Pm,Wcg,Wcp]=margin(sysd)
Gm=20*log10(Gm)
h=figure(3*i-1);subplot(2,2,j);step(sysi);grid;
% Identificarea sistemului folosind retele neuronale
sim('S1');sim('S2');
M=length(y); % M va avea 2001 elemente
ny=3;nu=1;nh=5;d=0; % Numarul de straturi, numarul de neuroni si intarzierea
% Obtinerea matricei P a intrarilor
s1=max(0,nu+d-ny);s2=max(0,ny-nu-d);s3=max(nu+d,ny);
P=uy(s1+1:M-ny,1)'; % P=uy(1:M-3,1)';
for k=2:ny
    P=[P;uy(s1+k:M-ny+k-1,1)']; % P=[P;uy(2:M-2,1)'];P=[P;uy(3:M-1,1)'];
end
for k=1:nu
    P=[P;uy(s2+k:M-d-nu+k-1,2)']; %P=[P;uy(3:M-1,2)'];
end
% Se traseaza grafic iesirea dorita T a sistemului
T=y(s3+1:M,1)';timp=1:length(T);
% Initializarea retelei neuronale
Z=[min(P(1,:)) max(P(1,:));min(P(2,:)) max(P(2,:));
    min(P(3,:)) max(P(3,:));min(P(4,:)) max(P(4,:))];
net=newff(Z,[5 1],{'tansig' 'purelin'});
y1=sim(net,P); % y1 este iesirea retelei neuronale inainte de antrenare
% Antrenarea retelei neuronale
net.trainParam.epochs=1000;
net.trainParam.goal=1e-4;
net = train(net,P,T);grid;
y2 = sim(net,P); % y2 este iesirea retelei neuronale dupa antrenare
h=figure(3*i);subplot(2,2,j);
plot(timp,T,'bo',timp,y2,'r');grid;
title('Rasp.indiciale ale sist. si NN dupa antrenare');xlabel('Timp');
% Calculul si afisarea ponderilor
W1=net.iw{1,1}
W2=net.lw{2,1}
B1=net.b{1}
B2=net.b{2}
end % end for j
end % end for i

% Comanda automata a deplasarii laterale prin modificarea unghiului de ruliu
clear all;
for i=1:3,

```

```

if i==1
    H=10 % Avion usor, H=10 km
    n14=-0.039;
elseif i==2
    H=10 % Avion mediu, H=10 km
    n14=-0.22;
else
    H=12 % Avion greu, H=12 km
    n14=-0.14;
end
a14=-n14;V=300;Ta=2.5;w0=0.707;csi=1;
% Calculul rapoartelor de transmisie adimensionale
ky=(w0^2)/a14
kyp=(2*csi*w0)/a14
a1=a14*kyp;a0=a14*ky;
%Rapoarte de transmisie dimensionale
kyy=ky/(Ta*V)
kyypp=kyp/V
% Declararea functiei de transfer
numi=[0 0 a0];deni=[1 a1 a0];numd=numi;dend=deni-numi;
sysi=tf(numi,deni)
sysd=tf(numd,dend)
sysd=tf(numd,dend);poli=pole(sysi)
zerouri=zero(sysi);
[A,B,C,D]=tf2ss(numi,deni)
Ts=.025;sys_z=c2d(sysi,Ts);[num_z,den_z]=tfdata(sys_z,'v');
sys_z=tf(num_z,den_z)
[A_z,B_z,C_z,D_z]=tf2ss(num_z,den_z)
% Caracteristici grafice
h=figure;margin(sysd);[Gm,Pm,Wcg,Wcp]=margin(sysd)
Gm=20*log10(Gm)
sim('L12_sch');t=1:length(Y);Pas_integrare=0.1;timp=t*Pas_integrare;h=figure;
subplot(2,2,1);step(sysi);grid;
subplot(2,2,2);plot(timp,Fi);grid;xlabel('Timp');ylabel('Unghi de rului');
subplot(2,2,3);plot(timp,Y);grid;xlabel('Timp');ylabel('Deplasarea laterala');
subplot(2,2,4);plot(timp,Csi);grid;xlabel('Timp');ylabel('Unghi de directie');
% Identificarea sistemului folosind retele neuronale
sim('S1');sim('S2');
M=length(y); % M va avea 2001 elemente
ny=3;nu=1;nh=5;d=0; % Numarul de straturi, numarul de neuroni si intarzierea
% Obtinerea matricei P a intrarilor
s1=max(0,nu+d-ny);s2=max(0,ny-nu-d);s3=max(nu+d,ny);
P=uy(s1+1:M-ny,1)'; % P=uy(1:M-3,1)';
for k=2:ny
    P=[P;uy(s1+k:M-ny+k-1,1)']; % P=[P;uy(2:M-2,1)'];P=[P;uy(3:M-1,1)'];
end
for k=1:nu
    P=[P;uy(s2+k:M-d-nu+k-1,2)']; %P=[P;uy(3:M-1,2)'];
end
% Se traseaza grafic iesirea dorita T a sistemului
T=y(s3+1:M,1)';timp=1:length(T);

```

```

% Inicializarea rețelei neuronale
Z=[min(P(1,:)) max(P(1,:));min(P(2,:)) max(P(2,:));
  min(P(3,:)) max(P(3,:));min(P(4,:)) max(P(4,:))];
net=newff(Z,[5 1],{'tansig' 'purelin'});
y1=sim(net,P); % y1 este iesirea rețelei neuronale înainte de antrenare
h=figure;plot(timp,T,'b',timp,y1,'r');grid;
title('Raspunsul indicial dorit (albastru) si iesirea NN înainte de antrenare (rosu)');
xlabel('Timp');
% Antrenarea rețelei neuronale
net.trainParam.epochs=5000;
net.trainParam.goal=1e-8;
net = train(net,P,T);grid;
y2 = sim(net,P); % y2 este iesirea rețelei neuronale după antrenare
h=figure;plot(timp,T,'bo',timp,y2,'r');grid; xlabel('Timp');
title('Raspunsul indicial dorit (albastru) si raspunsul indicial al NN după antrenare (rosu)');
% Calculul si afisarea ponderilor
W1=net.iw{1,1}
W2=net.lw{2,1}
B1=net.b{1}
B2=net.b{2}
end

```