8. Badania układów dynamiki w trybie tekstowym

Definicja układów dynamiki w środowisku graficznym jest bardzo uniwersalną metodą prowadzenia badań. Jednak większość z nich można zrealizować znacznie szybciej – definiując w skrypcie, za pomocą funkcji, zarówno modele jak i zaplanowane badania, szczególnie gdy przedmiotem badań są układy liniowe. W Matlabie wymaga to jednak dodatkowego narzędzia Control System Toolbox.

8.1 Liniowe modele dynamiki – funkcje

8.1.1 Definicja modeli

Podstawowe funkcje do definiowania modeli opierają się na równaniach stanu i transmitancjach, które stają się parametrami funkcji definiujących:

syslin - *linear system definition* | ss, tf, zpk – *LTI model*

Wynikiem działania tych funkcji są obiekty typu „model", czyli zmienne do przechowywania modeli w przestrzeni roboczej. W zależności od podanych parametrów te same funkcje pozwalają zdefiniować modele ciągłe i dyskretne.

1. Definicja modelu w postaci równań stanu $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ zawsze wymaga uzupełnienia o równania wyjściowe $\mathbf{y} = \mathbf{Cx} + \mathbf{Du}$:

mSS = syslin('c', A, B, C [,D [,x0]]) - *model ciągły*	mSS = ss(A, B, C, D) - *model ciągły*
mSS = syslin('d', A, B, C [,D [,x0]]) - *model dyskretny*	mSS = ss(A, B, C, D, TS) - *model dyskretny [TS- czas próbkowania]*

gdzie: A, B, C, D - macierze równań stanu i równań wyjściowych, a x0 - wektor warunków początkowych.

Przykład układu drugiego rzędu : $\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-1 & 1\\ 2 & -3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}1 & 0\\ 2 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}, \begin{bmatrix}y_1\\ y_2\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}$

A=[-1,1; 2,-3];B=[1,0; 2, 1];C=[1,0; 0,1]; D=[0,0; 0,0];	A=[-1,1; 2,-3];B=[1,0; 2, 1];C=[1,0; 0,1]; D=[0,0; 0,0];
SS1=syslin('c', A, B, C); - *utwórz i zapamiętaj model o nazwie SS1*	SS1=ss(A, B, C, 0); - *utwórz i zapamiętaj model o nazwie SS1*
SS1=syslin('c', A, B, C, D); - *jw*	SS1=ss(A, B, C, D); - *jw*
[a,b,c,d] = abcd(SS1) - *podstaw macierze modelu* SS1 *do* a,b,c,d	[a,b,c,d] = ssdata(SS1) - *podstaw macierze modelu* SS1 *do* a,b,c,d
SS1('A'), SS1('X0') - *podaj macierz A i war.początk. modelu* SS1	

2. Definicja modelu w postaci transmitancji $FW(s) = \dfrac{WN(s)}{WD(s)}$ jest dedykowana przede wszystkim do funkcji wymiernych:[1]

s=poly(0,'s'); - *definicja zmiennej wielomianu*	mTF = tf(N, D [,TS]); - *model ciągły lub dysktretny (*TS*)*
mTF = syslin('c', WN, WD); - *model ciągły (lub* 'd' - *dysktretny)*	
s=poly(0,'s');	s = tf('s'); - *definicja zmiennej Laplace'a*
mTF = syslin('c', FW); - *jw*	mTF = FW; - *model ciągły (tylko SISO)*
	mZPK = zpk(Z, P, K [,TS]); - *model ciągły lub dysktretny (*TS*)*
gdzie: WN, WD – wielomian licznika i mianownika (funkcje s) FW – funkcja wymierna (funkcja s) Tylko modele SISO. Modele MIMO definiowane tak jak macierze, np. [mTF1, mTF2]	gdzie: N, D – wektory współczynników licznika i mianownika Z, P – wektory zer i biegunów; K – wzmocnienie FW – funkcja wymierna (funkcja s) Modele SISO i MIMO

[1] Uwaga: W programach symulacyjnych (także w Matlabie i Scilabie) występuje ograniczenie by stopień wielomianu licznika był < (lub <=) od stopnia wielomianu mianownika transmitancji, która ujawnia się albo już na etapie definicji, albo na etapie przetwarzaniu modeli.

Różne sposoby definiowania typowych transmitancji zostaną przedstawione na przykładzie: $G(s) = \frac{1}{4s^2+4s+1} = \frac{1}{(2s+1)^2} = \frac{1}{4(s+\frac{1}{2})^2}$

```
s = poly(0,'s');
G1 = syslin('c',1, 4*s^2+4*s+1)    - transmitancja G1

G2 = syslin('c',1, (1+2*s)^2)      - transmitancja G2
G3 = syslin('c', 1/((1+2*s)^2))    - transmitancja G3

H4 = 1/((1+2*s)^2)                 - funkcja wymierna (wyr.algebraiczne)
G4 = syslin('c',H4)                - transmitancja G4
```

```
G1 = tf (1, [4 4 1]);              - transmitancja G1

s = tf('s');
G2 = 1/(4*s^2+4*s+1)               - transmitancja G2
G3 = 1/((1+2*s)^2)                 - transmitancja G3

G4 = zpk([], [-0.5 -0.5], 1/4)     - transmitancja G4
```

Przykłady szczególnych przypadków transmitancji: $G_p(s) = 3$, $G_r(s) = s$, $G_o(s) = e^{-s5}$

```
Gp = 3                             - wzmocnienie jako prosta zmienna
~~Gp = syslin('c',3/1)~~           - niepoprawne
Gp = syslin('c',3/(0*s+1))         - wzmocnienie (istnieje zmienna s)
~~Gr = syslin('c',s)~~             - niepoprawne
Gr = syslin('c',s/(0*s+1))         - cz.różniczkujący (istnieje zmienna s)
```

```
Gp = 3                             - wzmocnienie jako prosta zmienna
Gp = tf(3)                         - wzmocnienie
Gp = 3/(0*s+1);                    - wzmocnienie (istnieje zmienna s)
Gp = s                             - cz.różniczkujący (istnieje zmienna s)
Gp = tf(s)                         - jw
Go = exp(-5*s);                    - opóźnienie (istnieje zmienna s)
```

Typowe transmitancje mają postać funkcji wymiernych, w których stopień licznika jest mniejszy (lub ≤) niż stopień mianownika. Transmitancje, które nie spełniają tego założenia mają ograniczony zakres przetwarzania, więc w razie konieczności dla tych szczególnych przypadków stosuje się przybliżenia:

$G_p(s) = k_p \rightarrow \frac{k_p}{T_r s + 1}$

człon proporcjonalny rzeczywisty

$G_r(s) = s \rightarrow \frac{s}{T_r s + 1}$

człon różniczkujący rzeczywisty

$G_o(s) = e^{-sT_o} \rightarrow \frac{1 - sT_0/2}{1 + sT_0/2}$

aproksymacja Padé

8.1.2 Operacje na modelach

Elementarne modele zdefiniowane na podstawie równań stanu i transmitancji są obiektami, które mogą być przedmiotem dalszego przetwarzania, a w szczególności konwersji typu modelu oraz konstrukcji schematów blokowych.

1. Konwersja modelu umożliwia przekształcenie równań stanu na transmitancje i odwrotnie.

```
ss2tf, tf2ss                       - funkcje konwersji
```

```
ss, tf, zpk                        - funkcje definicji zastosowane do konwersji
ss2tf, ss2zpk, tf2ss, tf2zpk, zpk2ss, zpk2tf   - funkcje konwersji
```

Załóżmy że model mSS to równania stanu, model mTF to transmitancja oparta na wielomianach, wówczas:

```
mTF = ss2tf(mSS)
mSS = tf2ss(mTF)
[off, WN, WD] = ss2tf(mSS)
```

gdzie: WN, WD – wielomian licznika i mianownika (funkcje s)

```
mTF = tf(mSS)
mSS = ss(mTF)
[N,D] = ss2tf(A, B, C, D, nr_wejscia)
[A,B,C,D] = TF2SS(N,D)
```

gdzie: N, D – wektory współczynników licznika i mianownika

Możliwości konwersji w Matlabie są szersze ze względu transmitancje zdefiniowane przez zera i bieguny (model mZPK). Funkcje zpk oraz ss2zpk i tf2zpk można zastosować nie tylko do zmiany typu modelu ale pośrednio do wyznaczenia zer i biegunów liniowych modeli

2. Konstrukcja schematów blokowych na bazie obiektów zdefiniowanych wcześniej opiera się na kilku podstawowych operacjach i funkcjach, które realizują połączenie równoległe (+), szeregowe (*), ze sprzężeniem zwrotnym (/.).

S1 + S2	S1 + S2;	parallel(S1,S2 [,wej,wyj])
S1 * S2	S1 * S2;	series(S1,S2 [,wej,wyj])
Sl1 /. Sl2	--------	feedback(S1,S2 [,wej,wyj])
--------	--------	connect(S1,S2,wej,wyj)
--------	S1 \ S2 oraz S1 / S2	inv(S1) * S2 oraz S1 * inv(S2)

Można łączyć modele różnych typów, ważny może być jedynie rozmiar modelu, czyli ilość wejść i wyjść. Rozmiar każdego modelu prostego czy złożonego można sprawdzić za pomocą funkcji size(nazwa_modelu).

8.2 Podstawowe badania obiektów liniowych

Badania dynamicznych własności obiektów wchodzą obszar narzędzi CACSD[1] i w najprostszym przypadku polegają na zarejestrowaniu odpowiedzi skokowej lub impulsowej oraz wyznaczeniu charakterystyk częstotliwościowych obiektu [1].

8.2.1 Charakterystyki czasowe

Odpowiedź skokowa jest reakcją obiektu na skok jednostkowy (od 0 do 1) podawany w stanie równowagi w zerowych warunkach początkowych. Analogicznie odpowiedź impulsowa jest reakcją na impuls Diraca w takich samych warunkach. Takie eksperymenty realizują następujące funkcje:

csim - *odpowiedzi czasowe układów liniowych*	step - *odpowiedz skokowa układu liniowego* impulse - *odpowiedz impulsowa układu liniowego*
Tylko dla modeli SISO. Funkcja zwraca wektor wartości.	Modele SISO i MIMO. Funkcje tworzą wykres lub zwracają wartości.

W przykładzie poniżej zdefiniowano prosty człon inercyjny i wyznaczono jego odpowiedzi czasowe - skokową i impulsową.

```
s=poly(0,'s'); S1=syslin('c', 1/(2*s+1) ); t=0:0.05:5;
subplot(2, 1, 1)
plot(t,csim('step', t, S1));          //wyznacz i rysuj odp.skokową
subplot(2, 1, 2)
odpi = csim('impulse', t, S1);        // wyznacz odp.impulsową
plot(t, odpi);                        // rysuj odp.impulsową
```

```
s = tf('s'); S1 = 1/(2*s+1);
subplot(2, 1, 1)
step(S1,5)                    % wyznacz i rysuj odp.skokową
subplot(2, 1, 2)
[odpi, t] = impulse(S1,5);    % wyznacz odp.impulsową
plot(t, odpi);                % rysuj odp.impulsową
```

W kolejnym przykładzie wyznaczono odpowiedzi dwuwymiarowego modelu opartego na równaniach stanu

```
A=[-1,1; 2,-3];B=[1,0; 2, 1];C=[1,0; 0,1]; D=[0,0; 0,0];
S2=syslin('c', A, B, C);  S2t=ss2tf(S2);
t=0:0.05:5;
```

- Ze względu na ograniczenie do modeli SISO należałoby robić badania poszczególnych transmitancji z modelu S2t

```
A=[-1,1; 2,-3];B=[1,0; 2, 1];C=[1,0; 0,1]; D=[0,0; 0,0];
S2=ss(A, B, C, D);
subplot(2, 1, 1)
step(S2,5)                          % wyznacz i rysuj odp.skokową
subplot(2, 1, 2); hold on; ylabel('x1'); xlabel('u1 i u2');
[odpi, t] = impulse(S2,5);          % wyznacz odp.impulsową
plot(t, odpi(:,1,1)); plot(t, odpi(:,1,2));     % rysuj x1(u1) i x1(u2)
```

[1] Computer Aided Control System Design

8.2.2 Charakterystyki częstotliwościowe

W badaniach układów dynamiki duże znaczenie mają charakterystyki częstotliwościowe [1][7]. Punktem wyjścia dla nich jest opis modelu za pomocą transmitancji, w której zastosowano podstawienie $s=j\omega$, a następnie wykorzystano własności liczb zespolonych:

$$G(j\omega)=\frac{L(j\omega)}{M(j\omega)}=P(\omega)+jQ(\omega)=A(\omega)e^{j\varphi(\omega)}$$

Fizyczna interpretacja zmiennej ω[1] i różne postacie transmitancji są podstawą do konstrukcji kilku typów charakterystyk:

- charakterystyki części rzeczywistej $P(\omega)$ i części urojonej $Q(\omega)$ transmitancji,
- charakterystyka amplitudowo-fazowa $P(Q)$ – wykres Nyquista,
- charakterystyka amplitudowa $A(\omega)$ i charakterystyka fazowa $\varphi(\omega)$,
- logarytmiczna charakterystyka modułu $M(\omega)=20\lg A(\omega)$ i fazy $\varphi(\omega)$ - wykresy Bodego,
- logarytmiczna charakterystyka amplitudowo-fazowa $M(\varphi)$.

Najczęściej stosowane charakterystyki mają wsparcie w następujących funkcjach:

bode, gainplot	- *ch. Bodego (oś częstotliwości w Hz)*	bode, bodemag	- *ch. Bodego (oś częstotliwości w rad/sek)*
nyquist	- *ch.Nyquista*	nyquist	- *ch.Nyquista*
black	- *ch.Nicholsa (Black-Nichols)*	nichols	- *ch.Nicholsa*
nicholschart	- *siatka do ch.Nicholsa*		
g_margin, p_margin	- *charakterystyczne parametry ch.częstotliw.*	allmargin, margin	- *charakterystyczne parametry ch.częstotliw.*
repfreq	- *odpowiedź częstotliwościowa*	freqresp	- *odpowiedź częstotliwościowa*
Tylko dla modeli SISO i SIMO. Funkcja zwraca wektor wartości.		Modele SISO i MIMO. Funkcje tworzą wykres lub zwracają wartości.	

8.2.3 Projektowanie układów sterowania

Narzędzia typu CACSD obejmują wiele specjalistycznych funkcji przeznaczonych do projektowania układów sterowania. Wśród nich znajdują się narzędzia związane z operacjami na macierzach i sterowaniem liniowo-kwadratowy-Gaussa (LQG – ang. *Linear-quadratic-Gaussian*), które jest fundamentalnym zagadnieniem sterowania optymalnego. Regulator LQG jest kombinacją filtru Kalmana z regulatorem liniowo-kwadratowym (LQ). Problem LQ dotyczy sterowania układem dynamicznym opisanych przez układ liniowych równań różniczkowych, które minimalizuje koszt opisany funkcjonałem kwadratowym. Rozwiązanie tego problemu można z łatwością obliczyć za pomocą komputera dysponując na przykład funkcjami:

ricc	- *równanie Riccatiego*	care	- *równanie Riccatiego*
linmeq	- *równania Sylvester and Lyapunov*	lyap	- *równanie Lyapunova*
lqe	- *linear quadratic estimator (Kalman Filter)*	kalman	- *estimator Kalmana*
lqr	- *kompensator LQ*	lqr	- *regulator LQ*
lqg	- *kompensator LQG*	lqg	- *synteza sterowania LQG*

8.2.4 Definicja i własności podstawowych członów dynamiki

Dla ilustracji badań układów liniowych w poniższych skryptach zdefiniowano kilka członów dynamiki i wyznaczono ich charakterystyki czasowe i częstotliwościowe. Przy okazji w komentarzach zaznaczono ograniczenia w definicji lub w badaniach szczególnych przypadków transmitancji, gdy stopień licznika jest większy niż stopień mianownika.

[1] ω to pulsacja [rad/sek]; $\omega=2\pi f$, gdzie f to częstotliwość [Hz]