

Laboratorium Techniki Obliczeniowej i Symulacyjnej

Ćwiczenie 7. Analiza liniowych obiektów dynamicznych.

Opracował: dr inż. Sebastian Dudzik

1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z metodami analizy obiektów dynamicznych w programie Matlab z wykorzystaniem przybornika *Control System Toolbox*.

2. Wprowadzenie

2.1. Postacie modeli liniowych obiektów dynamicznych w *Control System Toolbox*

2.1.1. Postać transmitancyjna

Układy opisane liniowymi równaniami różniczkowymi o stałych współczynnikach nazywają się układami liniowymi stacjonarnymi. Jeżeli współczynniki te zmieniają się w czasie, lecz nie są zależne od wielkości wejściowych ani wyjściowych układ nazywany jest niestacjonarnym. Układy opisane równaniami nieliniowymi noszą nazwę układów nieliniowych. Dla układu liniowego, stacjonarnego i jednowymiarowego, tj. o jednym wejściu i jednym wyjściu zależność pomiędzy sygnałem wyjściowym $y(t)$ (odpowiedzią układu) i wejściowym $x(t)$ (wymuszeniem) określona jest ogólnym równaniem:

$$a_n \frac{d^n y(t)}{dt^n} + \dots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_m \frac{d^m x(t)}{dt^m} + \dots + b_1 \frac{dx(t)}{dt} + b_0 x(t), \quad (1)$$

gdzie: $a_i, i = 0 \dots n$ — stałe współczynniki, zależne od struktury i od wartości parametrów układu; $b_i, i = 0 \dots m$ — stałe współczynniki, zależne od źródła sygnału wejściowego oraz od wartości parametrów układu i jego struktury.

Rząd n najwyższej pochodnej sygnału wyjściowego występującej w równaniu nazywamy rzędem układu. Poddając obie strony równania różniczkowego (1) przekształceniu Laplace'a dla zerowych warunków początkowych dostaniemy:

$$\begin{aligned} M(s)Y(s) &= N(s)X(s) \\ M(s) &= a_n s^n + \dots + a_1 s + a_0 \\ N(s) &= b_m s^m + \dots + b_1 s + b_0. \end{aligned} \quad (2)$$

Stosunek transformaty Laplace'a sygnału wyjściowego $Y(s)$ układu do transformaty Laplace'a sygnału wejściowego $X(s)$, przy zerowych warunkach początkowych nazywamy transmitancją operatorową układu:

$$G(s) = \frac{Y(s)}{X(s)}. \quad (3)$$

Znając transmitancję układu można wyznaczyć odpowiedź układu na dowolny sygnał wejściowy $x(t)$:

$$y(t) = L^{-1}[Y(s)] = L^{-1}[G(s)X(s)]. \quad (4)$$

Postać modelu SISO (ang. *Single Input Single Output*), czyli o jednym wejściu i jednym wyjściu, dla transmitancji w postaci ilorazu wielomianów zmiennej s , może być zapisana w języku MATLAB na dwa sposoby. Za pomocą funkcji `tf`:

`h=tf(num,den)`,

gdzie `num`, `den` — wektory zawierające współczynniki wielomianów odpowiednio: licznika i mianownika. Dla przykładu, transmitancję:

$$h(s) = \frac{s}{s^2 + 2s + 10} \quad (5)$$

, można wprowadzić do programu MATLAB w następujący sposób: `h = tf([1 0],[1 2 10])`. Za pomocą wyrażenia zmiennej s :

1. Zdefiniowanie zmiennej s , jako symbolu specjalnego: `s = tf('s')`;
2. Wprowadzenie transmitancji jako wyrażenia zmiennej s : `H = s/(s^2 + 2*s + 10)`;

2.1.2. Postać „wzmocnienie-zera-bieguny”

W poprzednim punkcie podana została zasada zapisu transmitancji układu w postaci ilorazu dwóch wielomianów. Inną formą zapisu transmitancji jest podanie pierwiastków wielomianów licznika i mianownika transmitancji. Pierwiastki zerujące wielomian licznika transmitancji to zera (ang. zeros). Pierwiastki zerujące wielomian mianownika transmitancji to bieguny (ang. poles). Bieguny to także pierwiastki równania charakterystycznego równania różniczkowego, zatem widząc zapis zer i biegunów transmitancji można często bardzo szybko określić własności systemu dynamicznego. Jest to jeden z powodów dlaczego zapis ten jest ważny. Postać wzmocnienie-zera-bieguny może być zapisana jako:

$$h(s) = k \frac{(s - z_1) \dots (s - z_m)}{(s - p_1) \dots (s - p_m)}, \quad (6)$$

gdzie: k — wzmocnienie; $z_1 \dots z_m$ — zera układu; p_1, \dots, p_m — bieguny układu.

Aby wprowadzić model w postaci wzmocnienie-zera-bieguny, można zastosować funkcję `zpk`. Składnia wywołania funkcji `zpk` jest następująca:

`h=zpk(z,p,k)`

gdzie: `z` oraz `p` — wektory odpowiednio: zer i biegunów układu; `k` — wzmocnienie.

2.1.3. Postać zmiennych stanu

Sposoby opisu układów dynamicznych przedstawione poprzednio mają niestety wadę. Aby je zastosować musimy przyjąć warunki początkowe równania (1) równe zero. Stwarza to ograniczenie, np. przy modelowaniu działania silnika, gdzie musimy założyć, że w chwili początkowej prędkość obrotowa wału jest równa zero. Tego ograniczenia nie ma zapis modelu systemu dynamicznego w postaci układu równań różniczkowych:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \quad (7)$$

Pogrubienie symboli oznacza, że mamy do czynienia z wektorami. Zmienna $\mathbf{x}(t)$ jest tzw. wektorem zmiennych stanu. Należy zauważyć, że w równaniu (7) występuje tylko pochodna pierwszego rzędu, zatem rozwiązanie takiego równania sprowadza się do rozwiązania równania pierwszego rzędu przy zadanych warunkach początkowych. Dla systemów wyższych rzędów trzeba rozwiązać układ kilku równań pierwszego rzędu z warunkami początkowymi odpowiednio dla każdej zmiennej stanu $x(t)$. Na przykład: aby uzyskać zapis równania szóstego rzędu w postaci zmiennych stanu, należy tak je przekształcić, aby uzyskać sześć równań różniczkowych pierwszego rzędu. Przy rozwiązaniu każdego z nich uwzględnia się odpowiednie warunki początkowe.

Do utworzenia modelu w postaci równań stanu służy funkcja `ss(A,B,C,D)`, gdzie: A, B, C, D — macierze opisujące równanie (7).

2.2. Charakterystyki liniowych obiektów dynamicznych w środowisku MATLAB

Najważniejsze charakterystyki czasowe liniowych obiektów dynamicznych to: charakterystyka skokowa, impulsowa i liniowo-czasowa.

Charakterystyka skokowa jest to odpowiedź $y(t) = h(t)$ układu, na którego wejście doprowadzony został sygnał skokowy $x(t)$ opisany równaniem $x(t) = a\mathbf{1}(t)$, gdzie:

$$\mathbf{1}(t) = \begin{cases} 0 & \text{dla } t \leq 0 \\ 1 & \text{dla } t > 0 \end{cases} \quad (8)$$

W programie MATLAB, do wykreślenia charakterystyki skokowej obiektu, wykorzystuje się funkcję `step`. Można ją wywołać w następujący sposób: `[y,t,x] = step(sys)`, gdzie: \mathbf{y} — wektor odpowiedzi skokowej (odpowiedź skokowa), \mathbf{t} — wektor czasu, \mathbf{x} — wektor trajektorii stanu (tylko dla modeli w postaci zmiennych stanu), `sys` — liniowy model dynamiczny (reprezentowany w postaci transmitancji lub innej).

Charakterystyka impulsowa układu jest to odpowiedź $y(t) = k(t)$ układu, na którego wejście doprowadzony został sygnał w postaci impulsu Diraca $x(t) = \delta(t)$ (impuls o jednostkowej energii, nieskończonej amplitudzie i nieskończone krótkim czasie trwania):

$$\delta(t) = \begin{cases} 0 & \text{dla } t \neq 0 \\ \infty & \text{dla } t = 0 \end{cases} \quad (9)$$

W programie MATLAB, do wykreślenia charakterystyki impulsowej obiektu, wykorzystuje się funkcję `impulse`. Można ją wywołać w następujący sposób: `[y,t,x] = impulse(sys)`, gdzie: y — wektor odpowiedzi impulsowej (odpowiedź impulsowa), t — wektor czasu, x — wektor trajektorii stanu (tylko dla modeli w postaci zmiennych stanu), `sys` — liniowy model dynamiczny (reprezentowany w postaci transmitancji lub innej).

Charakterystyka liniowo-czasowa jest to odpowiedź $y(t) = v(t)$ układu, na którego wejście doprowadzony został sygnał $x(t)$ liniowo zależny od czasu:

$$x(t) = \begin{cases} 0 & \text{dla } t < 0 \\ bt & \text{dla } t \leq 0 \end{cases} \quad (10)$$

W programie MATLAB, do wykreślenia charakterystyki liniowo-czasowej obiektu, można wykorzystać funkcję `lsim`. Umożliwia ona symulację odpowiedzi obiektu na dowolne wymuszenie (również wymuszenie liniowe). Można ją wywołać w następujący sposób: `[y,t,x] = lsim(sys,u,t,x0)`, gdzie: y — wektor odpowiedzi (odpowiedź obiektu na zadane wymuszenie), t — wektor czasu, x — wektor trajektorii stanu (tylko dla modeli w postaci zmiennych stanu), `sys` — liniowy model dynamiczny (reprezentowany w postaci transmitancji lub innej), u — macierz (wektor dla układów SISO) wymuszeń, x_0 — warunki początkowe (tylko dla modeli w postaci zmiennych stanu).

Oprócz opisu w dziedzinie czasu liniowy układ dynamiczny może być opisywany w dziedzinie częstotliwości. W praktyce duże znaczenie mają charakterystyki częstotliwościowe wyznaczone w skali logarytmicznej, nazywane charakterystykami Bodego (H.W. Bode - opracował metody projektowania wzmacniaczy ze sprzężeniem zwrotnym, 1945):

- Logarytmiczna charakterystyka amplitudowa $Lm(\omega)$ (logarytmiczny moduł wzmocnienia) jest określona zależnością:

$$Lm(\omega) = 20 \log A(\omega) = 20 \log |G(j\omega)|, \quad (11)$$

podawana w decybelach [dB].

- Logarytmiczna charakterystyka fazowa $\phi(\omega)$ jest zależnością przesunięcia fazowego od częstotliwości przedstawionej w skali logarytmicznej.

Para charakterystyk Bodego przedstawia zależność logarytmu wzmocnienia i przesunięcia fazowego od częstotliwości w sposób jawny.

W programie MATLAB, do wykreślenia charakterystyk Bodego obiektu, można wykorzystać funkcję `bode`. Wykreśla ona charakterystyki: amplitudową i fazową obiektu w skali logarytmicznej. Funkcję można wywołać w następujący sposób: `[mag,phase,w] = bode(sys)`, gdzie: `mag` — w ogólnym przypadku trójwymiarowa macierz wzmocnień obiektu dla zadanych pulsacji (charakterystyka amplitudowa), `phase` — w ogólnym przypadku trójwymiarowa macierz przesunięć fazowych pomiędzy sygnałami wejściowym i wyjściowym (charakterystyka fazowa), `w` — wektor pulsacji, `sys` — liniowy model dynamiczny (reprezentowany w postaci transmitancji lub innej).

3. Program ćwiczenia

1. Uruchomienie programu MATLAB.

W ćwiczeniu wykorzystano program MATLAB w wersji 5.3 (R11.1). Uruchomienie programu następuje poprzez skrót na pulpicie (Matlab5.3) lub bezpośrednio z katalogu $C:\backslash\text{MatlabR11}\backslash\text{bin}\backslash$.

2. Uruchomienie programu Wordpad.exe.

Program można uruchomić poprzez wywołanie: $\text{Start}\backslash\text{Programy}\backslash\text{Akcesoria}\backslash\text{Wordpad}$ lub poprzez skrót na pulpicie.

3. Przejście do katalogu roboczego dla grupy laboratoryjnej.

Domyślnym katalogiem startowym (roboczym) programu MATLAB jest $C:\backslash\text{MatlabR11}\backslash\text{work}\backslash$. Zadanie polega na przejściu do podkatalogu katalogu *work*. Podkatalog (utworzony na pierwszych zajęciach laboratoryjnych) nazwany jest wybranymi 2 nazwiskami studentów, wchodzących w skład grupy laboratoryjnej.

- (a) Wprowadzić:

```
>>pwd
```

W programie MATLAB każde wprowadzone polecenie zatwierdza się klawiszem <ENTER>. Zwrócić uwagę na ścieżkę dostępu do katalogu bieżącego.

- (b) Wprowadzić:

```
>>cd nazwa_podkatalogu
```

Parametr *nazwa_pod-katalogu* powinien składać się z nazwisk 2 wybranych studentów grupy laboratoryjnej (np. >>cd KowalskiNowak).

4. Wprowadzanie modeli liniowych obiektów dynamicznych w postaci transmitancji

- (a) Wyprowadzić transmitancję $G(s) = \frac{Y(s)}{U(s)}$ na podstawie równania:

$$a_1 \frac{dy(t)}{dt} + a_0 y(t) = k_0 u(t) \quad (12)$$

ROZWIĄZANIE: $G(s) = \frac{Y(s)}{U(s)} = \frac{k_0}{a_1 s + a_0}$

- (b) Wprowadzić model (12) w postaci transmitancyjnej do środowiska MATLAB dla: $k_0 = 5, a_0 = 1, a_1 = 4$. Wprowadzić i uruchomić następujący skrypt:

```
%metody definiowania systemów dynamicznych
%skrypt model_1_rz.m - postać transmitancyjna
clc
k0=5; a0=1; a1=4;
licznik_system_nr_1=[k0];
mianownik_system_nr_1=[a1 a0];
sys_1rz_1_trans=tf(licznik_system_nr_1,mianownik_system_nr_1);
disp('to jest system dynamiczny o transmitancji:')
sys_1rz_1_trans
```

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (c) Podobnie jak w poprzednim podpunkcie, wprowadzić dwa modele w postaci transmitancyjnej o następujących parametrach:

i. `sys_1rz_2_trans` — $k_0 = 10, a_0 = 1, a_1 = 4$

ii. `sys_1rz_3_trans` — $k_0 = 5, a_0 = 3, a_1 = 10$

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (d) Wyprowadzić transmitancję $G(s) = \frac{Y(s)}{U(s)}$ na podstawie równania:

$$a_2 \frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_0 y(t) = k_1 \frac{du(t)}{dt} + k_0 u(t) \quad (13)$$

- (e) Wprowadzić model (13) w postaci transmitancyjnej do środowiska MATLAB dla:

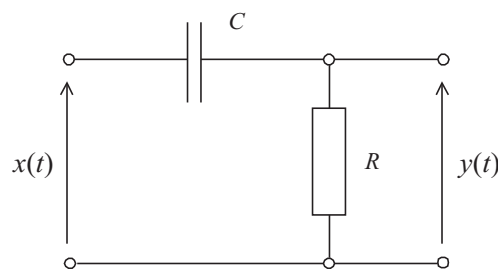
i. `sys_2rz_1_trans` — $k_0 = 5, k_1 = 6, a_0 = 1, a_1 = 4, a_2 = 8$

ii. `sys_2rz_2_trans` — $k_0 = 5, k_1 = 3, a_0 = 2, a_1 = 2, a_2 = 1$

iii. `sys_2rz_3_trans` — $k_0 = 5, k_1 = 10, a_0 = 7, a_1 = 3, a_2 = 3$

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (f) Wyprowadzić transmitancję $G(s) = \frac{Y(s)}{X(s)}$ przedstawionego na rys. 1 modelu obwodu RC. Wprowadzić model w postaci transmitancyjnej do środowiska MATLAB, przyjmując: $R = 1k\Omega, C = 2\mu F$. Skopiować zawartość okna poleceń



Rys. 1. Obwód RC

programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

5. Wprowadzanie modeli liniowych obiektów dynamicznych w postaci wzmacnienie-zera-bieguny

- (a) Dla transmitancji modelu (12) wyliczyć pierwiastki mianownika oraz współczynnik wzmacnienia.
- (b) Przekształcić model (12) z postaci transmitancyjnej do postaci wzmacnienie-zera-bieguny dla: $k_0 = 5, a_0 = 1, a_1 = 4$. Wprowadzić i uruchomić następujący skrypt:

```
%metody definiowania systemów dynamicznych
%skrypt model_1_rz.m - postać wzmacnienie-zera-bieguny
sys_1rz_1_zpk =zpk(sys_1rz_1_trans)
get(sys_1rz_1_zpk)
sys_1rz_1_zpk.z{:}
sys_1rz_1_zpk.p{:}
sys_1rz_1_zpk.k
[zera,bieguny,wzmacnienie]=zpkdata(sys_1rz_1_zpk,'v')
```

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (c) Podobnie jak w poprzednim podpunkcie, przekształcić dwa modele z postaci transmitancyjnej do postaci wzmacnienie-zera-bieguny dla następujących parametrów:
- i. `sys_1rz_2_zpk` — $k_0 = 10, a_0 = 1, a_1 = 4$
 - ii. `sys_1rz_3_zpk` — $k_0 = 5, a_0 = 3, a_1 = 10$

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (d) Dla transmitancji modelu (13) wyliczyć pierwiastki mianownika i licznika oraz współczynnik wzmacnienia.
- (e) Przekształcić model (13) z postaci transmitancyjnej do postaci wzmacnienie-zera-bieguny dla następujących parametrów:
- i. `sys_2rz_1_zpk` — $k_0 = 5, k_1 = 6, a_0 = 1, a_1 = 4, a_2 = 8$
 - ii. `sys_2rz_2_zpk` — $k_0 = 5, k_1 = 3, a_0 = 2, a_1 = 2, a_2 = 1$
 - iii. `sys_2rz_3_zpk` — $k_0 = 5, k_1 = 10, a_0 = 7, a_1 = 3, a_2 = 3$

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (f) Dla transmitancji modelu obwodu RC z p. 4f wyliczyć pierwiastki mianownika oraz współczynnik wzmacnienia.

- (g) Przekształcić model obwodu RC z p. 4f z postaci transmitancyjnej do postaci wzmocnie-nie-zera-bieguny.
Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (h) Narysować mapę zer i biegunów dla modelu z p. 4b
Wprowadzić i uruchomić następujący skrypt:

```
pzmap(sys_1rz_1_trans)
title ('kolka to zera a krzyzyki to bieguny - kliknij na nie')
```

- (i) Narysować mapę zer i biegunów dla modeli wprowadzonych w punktach: 4c i 4e oraz modelu obwodu RC z p. 4f.

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (j) Utworzyć własny model dynamiczny z dwoma zerami rzeczywistymi z zakresu $z \in (-5; -1)$, jednym biegunem równym zero i dwoma biegunami zespolonymi sprzężonymi z zakresu liczb dla części rzeczywistej $Re\{p\} \in (-5; -1)$ i dla urojonej $Im\{p\} \in (-5; 5)$, współczynnik wzmocnienia $k = 10$. Wykorzystać funkcję `zpk` programu MATLAB.

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

6. Wprowadzanie modeli liniowych obiektów dynamicznych w postaci zmiennych stanu

- (a) Przekształcić model wprowadzony w p. 4b do postaci zmiennych stanu.
Wprowadzić:

```
sys_1rz_1_ss=ss(sys_1rz_1_trans)
```

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (b) Przekształcić modele wprowadzone w punktach: 4c i 4e oraz model obwodu RC z p. 4f do postaci zmiennych stanu.

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

7. Wykreślanie charakterystyk czasowych liniowych modeli dynamicznych

- (a) Wykreślić odpowiedź skokową, impulsową oraz odpowiedź na wymuszenie harmoniczne modelu (12) dla: $k_0 = 5, a_0 = 1, a_1 = 4$. Wprowadzić i uruchomić następujący skrypt:


```
%Wykreślanie charakterystyk czasowych modeli
step(sys_1rz_1_trans)
impulse(sys_1rz_1_trans)
t=0:0.01:40;
okres=10;
u=sin(2*pi*t/okres);
[y,t]=lsim(sys_1rz_1_trans,u,t)
plot(t,y,t,u)
```

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (b) Wykreślić odpowiedź skokową, impulsową oraz odpowiedź na wymuszenie harmoniczne modeli wprowadzonych w punktach: 4c i 4e oraz modelu obwodu RC z p. 4f. Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

8. Wykreślanie charakterystyk częstotliwościowych liniowych modeli dynamicznych

- (a) Wykreślić charakterystyki Bodego modelu (12) dla: $k_0 = 5, a_0 = 1, a_1 = 4$. Wprowadzić i uruchomić następujący skrypt:

```
grid on
bode(sys_1rz_1_trans)
```

Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

- (b) Wykreślić charakterystyki Bodego modeli wprowadzonych w punktach: 4c i 4e oraz modelu obwodu RC z p. 4f. Skopiować zawartość okna poleceń programu MATLAB oraz zawartość okna edytora do programu Wordpad. Wyczyścić zawartość okna poleceń programu MATLAB.

4. Opracowanie sprawozdania

W sprawozdaniu należy umieścić polecenia oraz wyniki ich działania skopiowane w trakcie ćwiczenia z okna środowiska MATLAB. Do każdej linii kodu oraz do każdego wyniku, należy dodać komentarz objaśniający.

Przykład.

...2+round(6/9+3*2)/2-3 — obliczenie wartości wyrażenia. Funkcja round(6/9+3*2) zaokrągla wynik działania 6/9+3*2 do najbliższej liczby całkowitej...