

POLITECHNIKA ŚWIĘTOKRZYSKA
Wydział Elektrotechniki, Automatyki i Informatyki
Katedra Elektrotechniki Przemysłowej i Automatyki
Zakład Urządzeń i Systemów Automatyki

Komputerowe Wspomaganie Projektowania w Układach Sterowania

Wprowadzenie do pakietu Control System Toolbox

Instrukcja laboratoryjna

Paweł Strączyński

2018

1 Wstęp

Control System Toolbox to pakiet bibliotek i narzędzi dla systemu MATLAB służący do projektowania i modelowania układów dynamicznych. Pozwala m.in. na wykreślanie charakterystyk częstotliwościowych badanych systemów oraz ich odpowiedzi w dziedzinie czasu na wymuszenia. Control System Toolbox umożliwia m.in. projektowanie systemów SISO oraz MIMO jak również udostępnia narzędzia do projektowania układów sterowania.

2 Podstawowe polecenia pakietu CST

2.1 Definiowanie modelu systemu

tf - *transfer function*, funkcja służy do definiowania systemu w postaci transmitancji:

$$G(s) = \frac{NUM(s)}{DEN(s)} \quad (1)$$

gdzie NUM i DEN to odpowiednio macierze współczynników licznika oraz mianownika transmitancji.

$$SYS = tf(NUM, DEN)$$

Na listingu 2.1 przedstawiono sposób definiowania modelu obiektu opisanego transmitacją:

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

Listing 2.1: Definiowanie systemu w postaci transmitancji

```

1  clc; clear all;
2
3  %Pierwszy sposob zapisu transmitancji
4  s = tf('s');
5  SYS1 = s/(s^2 + 2*s + 10)
6
7  %Drugi sposob zapisu tej samej transmitancji
8  SYS2 = tf([1 0],[1 2 10])
9
10 fprintf('SYS:\nA: %d\nB: %d\nC: %d\nD: %d\n', SYS.A, SYS.B, SYS.C, SYS.D);

```

ss - *state space*, funkcja służy do definiowania systemu w postaci zmiennych stanu:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (3)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (4)$$

gdzie A,B,C oraz D to macierze równań stanu oraz wyjścia.

$$SYS = ss(A, B, C, D)$$

Na listingu 2.2 przedstawiono sposób definiowania modelu obiektu opisanego w przestrzeni stanu:

$$\dot{\mathbf{x}} = [-2]\mathbf{x}(t) + [-10]\mathbf{u}(t) \quad (5)$$

$$\mathbf{y} = [1]\mathbf{x}(t) + [0]\mathbf{u}(t) \quad (6)$$

Listing 2.2: Definiowanie systemu w postaci równań stanu

```

1 clc; clear all;
2 A=-2;
3 B=-10;
4 C=1;
5 D=0;
6 SYS = ss(A,B,C,D); %ss(-2,-10,1,0);
7 fprintf('SYS:\nA:\n%d\nB:\n%d\nC:\n%d\nD:\n%d\n',SYS.A,SYS.B,SYS.C,SYS.D);

```

zpk - funkcja służy do definiowania systemu w postaci transmitancji danej w postaci zer i biegunów oraz wzmocnień:

$$G(s) = K \cdot \frac{(s - z_1) \dots (s - z_m)}{(s - p_1) \dots (s - p_n)} \quad (7)$$

gdzie:

- pierwiastki licznika ($z_1 - z_m$) to zera transmitancji dane w postaci macierzy **Z**, natomiast pierwiastki mianownika ($p_1 - p_n$),
- bieguny transmitancji dane w postaci macierzy **P**.
- **K** macierz wzmocnień.

$$SYS = zpk(Z, P, K)$$

Na listingu 2.3 przedstawiono sposób definiowania modelu obiektu opisnego transmitancją:

$$G(s) = \frac{2s + 1}{(s + 2)(s + 5)} = \frac{2 \cdot (s + 0,5)}{(s + 2)(s + 5)} \quad (8)$$

Listing 2.3: Definiowanie systemu w postaci zer, biegunów oraz wzmocnień

```
1 clc; clear all;
2 Z = -0.5;
3 P = [-2 -5];
4 K = 2;
5 SYS= zpk(Z,P,K) %zpk(-0.5,[-2 -5],2)
```

Zarówno funkcja **tf**, **ss** jak i **zpk** mogą służyć również do definiowania modelu dyskretnego. W tym celu należy wywołać funkcję z trzecim argumentem określającym okres próbkowania.

$$SYS=tf(NUM,DEN,TS)$$

$$SYS=ss(A,B,C,D,TS)$$

$$SYS=zpk(Z,P,K,TS)$$

2.2 Zmiana postaci modelu

tf2ss - *transfer function to state space*, funkcja służy do zmiany postaci modelu z transmitancji na równania stanu.

$$G(s) = \frac{NUM(s)}{DEN(s)} = \mathbf{C} \cdot (s \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B} + \mathbf{D} \quad (9)$$

ss2tf - *state space to transfer function*, funkcja służy do zmiany postaci modelu z równań stanu na transmitancję względem wejścia *iu*.

$$[A,B,C,D] = tf2ss(NUM,DEN)$$

$$[NUM,DEN] = ss2tf(A,B,C,D,iu)$$

Na listingu 2.4 przedstawiono przykłady zmiany postaci modelu obiektu pomiędzy transmitancją a przestrzenią zmiennych stanu.

Listing 2.4: Zmiana postaci modelu: tf2ss, ss2tf

```

1 clc; clear all;
2 NUM = [1 0];
3 DEN = [1 2 10];
4 SYS1 = tf(NUM, DEN); % s/(s^2+2*s+10)
5 [A,B,C,D] = tf2ss(NUM,DEN);
6 fprintf('SYS:\nA:\nB:\nC:\nD:\n',SYS.A,SYS.B,SYS.C,SYS.D);
7 SYS2 = ss(A,B,C,D);
8 [NUM2, DEN2] = ss2tf(A,B,C,D);
9 SYS3 = tf2ss(NUM2,DEN2);
10 fprintf('SYS:\nNUM:\nDEN:\n',SYS.NUM,SYS.DEN);

```

zp2ss - *zero pole to state space*, funkcja służy do wyznaczenia macierzy zmiennych stanu na podstawie zer, biegunów oraz wzmocnień.

$$G(s) = \mathbf{C} \cdot (s \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B} + \mathbf{D} = K \cdot \frac{(s - z_1) \dots (s - z_m)}{(s - p_1) \dots (s - p_n)} \quad (10)$$

ss2zp - *state space to zero pole*, funkcja służy do wyznaczenia zer, biegunów i wzmocnień na podstawie równań stanu.

$$[A,B,C,D] = \text{zp2ss}(Z,P,K)$$

$$[Z,P,K] = \text{ss2zp}(A,B,C,D,iu)$$

Przykład użycia powyższych funkcji pokazano na listingu 2.5.

Listing 2.5: Zmiana postaci modelu: zp2ss, ss2zp

```

1 clc; clear all;
2 Z = -0.5;
3 P = [-2 -5];
4 K = 2;
5 SYS1 = zp2ss(Z,P,K);
6 [A, B, C, D] = zp2ss(Z,P,K);
7 SYS2 = ss(A,B,C,D);
8 [Z2, P2, K2] = ss2zp(A,B,C,D);
9 SYS3 = zp2ss(Z2,P2,K2);

```

tf2zp - *transfer function to zero pole*, funkcja służy wyznaczenia zer, biegunów i wzmocnień na podstawie transmitancji.

$$G(s) = \frac{NUM(s)}{DEN(s)} = K \cdot \frac{(s - z_1) \dots (s - z_m)}{(s - p_1) \dots (s - p_n)} \quad (11)$$

zp2tf - *zero pole to transfer function*, funkcja służy do wyznaczania transmitancji obiektu na podstawie zer, biegunów oraz wzmocnień.

$$[Z, P, K] = \text{tf2zp}(\text{NUM}, \text{DEN})$$

$$[\text{NUM}, \text{DEN}] = \text{zp2tf}(Z, P, K)$$

Przykład użycia funkcji **tf2zp** oraz **zp2tf** przedstawiono na listingu poniżej.

Listing 2.6: Zmiana postaci modelu: tf2zp, zp2tf

```

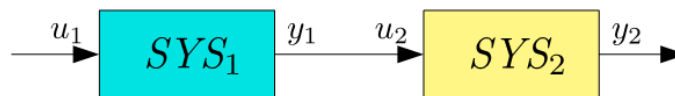
1 clc; clear all;
2 NUM = [1 0];
3 DEN = [1 2 10];
4 SYS1 = tf(NUM, DEN);
5 [Z, P, K] = tf2zp(NUM, DEN);
6 SYS2 = zp2tf(Z, P, K);
7 [NUM2, DEN2] = zp2tf(Z, P, K);
8 SYS3 = tf(NUM2, DEN2);

```

2.3 Łączenie modeli obiektów

series - funkcja służąca do tworzenia modelu składającego się z szeregowego połączenia modeli dwóch obiektów- jak na rysunku 2.3.

$$\text{SYS} = \text{series}(\text{SYS1}, \text{SYS2}, 'name')$$

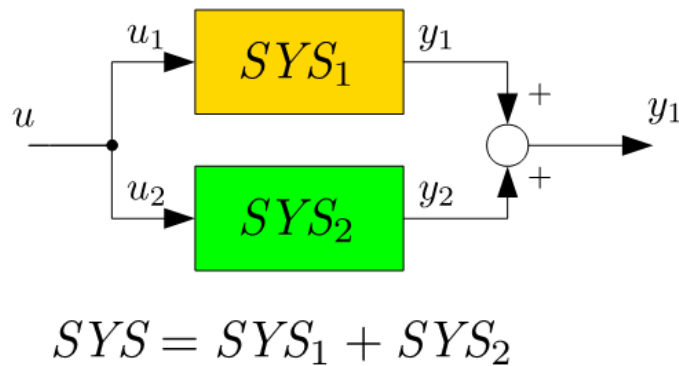


$$\text{SYS} = \text{SYS}_1 \cdot \text{SYS}_2$$

Rysunek 2.1: Szeregowe połączenie modeli dwóch obiektów

parallel - funkcja służąca do tworzenia modelu składającego się z równoległego połączenia modeli dwóch obiektów - jak na rysunku 2.3.

$$\text{SYS} = \text{parallel}(\text{SYS1}, \text{SYS2}, 'name')$$

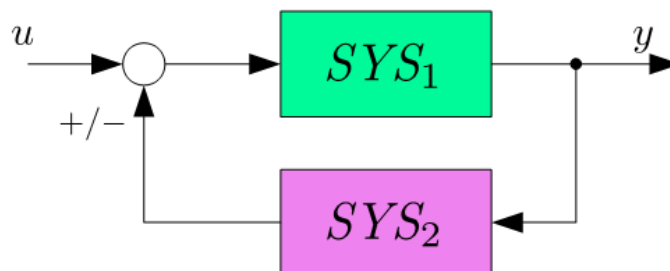


Rysunek 2.2: Równoległe połączenie modeli dwóch obiektów

feedback - funkcja służąca do tworzenia modelu ze sprzężeniem zwrotnym. - jak na rysunku 2.3.

$$SYS = feedback(SYS1, SYS2, SIGN)$$

gdzie *SIGN* określa rodzaj sprzężenia - domyślnie ujemne. Na listingu 2.7 przedstawiono przy-



Rysunek 2.3: Układ ze sprzężeniem zwrotnym

kład wykorzystania funkcji służących do łączenia modeli obiektów.

Listing 2.7: Łączenie modeli obiektów

```

1 clc; clear all;
2 s=tf('s');
3 SYS1 = 1/(0.2*s+1);
4 SYS2 = 1/(0.4*s+1);
5 SYS3 = series(SYS1, SYS2);
6 SYS4 = pararell(SYS1, SYS2);
7 SYS5 = feedback(SYS1, SYS2); %Układ z ujemnym sprzężeniem zwrotnym
8 SYS6 = feedback(SYS2, 1, +1); %Układ z dodatnim jednostkowym sprzężeniem zwrotnym

```

2.4 Badanie parametrów modelu obiektu

zero - funkcja służąca do wyznaczania zer i wzmocnień obiektu.

$$[ZERO, GAIN] = zero(SYS)$$

pzmap - funkcja służąca do wykreślania rozkładu zer i begunów modeli obiektów.

$$pzmap(SYS1, SYS2, \dots)$$

margin - funkcja służąca do wyznaczania zapasu modułu Gm i fazy Pm (w stopniach) oraz związane z nimi częstotliwości Wcg , Wcp .

$$[Gm, Pm, Wcg, Wcp] = margin(SYS)$$

allmargin - funkcja służąca do wyznaczania szczegółowych informacji dotyczących zapasu modułu Gm i fazy Pm (w stopniach).

$$S = allmargin(SYS)$$

gdzie S jest strukturą która zawiera pola takie jak: *GMFrequency*, *GainMargin*, *PMFrequency*, *PhaseMargin*, *DelayMargin* oraz *Stable*. Przykładowe użycie funkcji przedstawiono na listingu poniżej.

Listing 2.8: Badanie parametrów modelu obiektu

```

1 clc; clear all;
2 s = tf('s');
3 SYS = s/(s^2 + 2*s + 10)
4 [ZERO, GAIN] = zero(SYS);
5 [Gm, Pm, Wcg, Wcp] = margin(SYS);
6 S = allmargin(SYS);
7 disp(S.Stable);
8 pzmap(SYS);

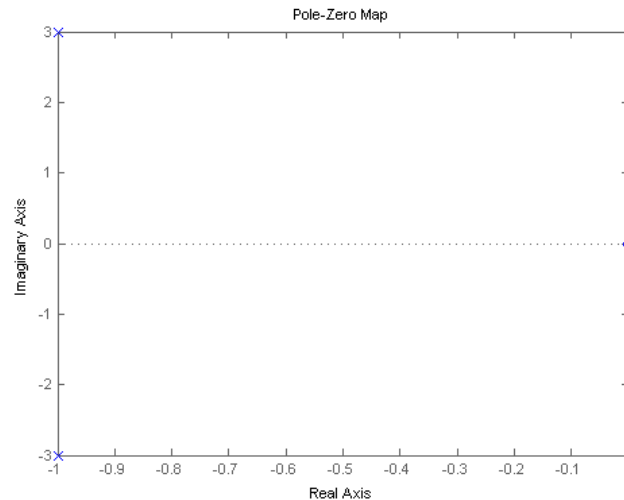
```

2.5 Wyznaczanie charakterystyk częstotliwościowych układu

bode - funkcja służąca do wykreślania charakterystyk logarytmicznych (charakterystyk Bode'go) dla jednego lub wielu obiektów.

$$bode(SYS1, SYS2, \dots, WMIN, WMAX)$$

gdzie parametry $WMIN$, oraz $WMAX$ są opcjonalne i określają zakres częstotliwości dla charakterystyki.



Rysunek 2.4: Rozkład zer i biegunów

nyquist – funkcja służąca do wykreślania charakterystyki amplitudowo – fazowej obiektu lub obiektów – charakterystyki Nyquist’a.

nyquist(SYS1, SYS2 . . . , WMIN, WMAX)

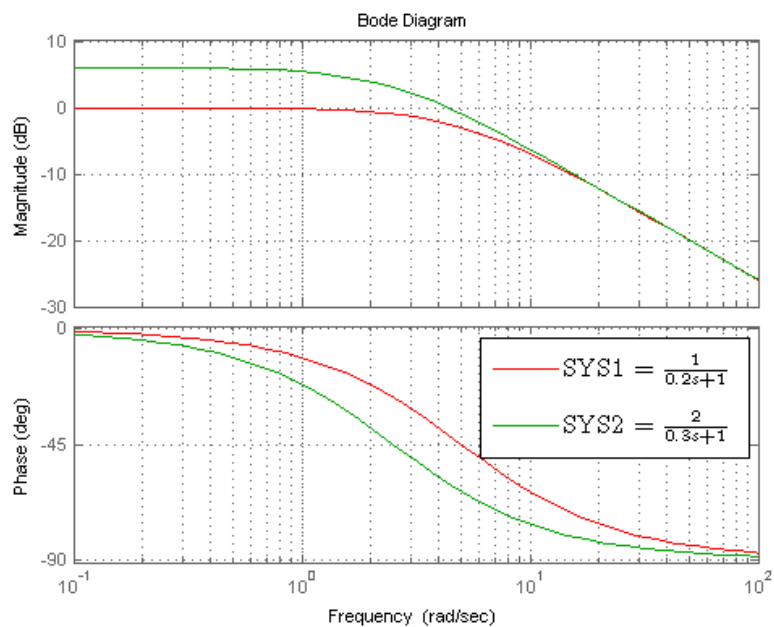
Przykładowy program wyznaczający charakterystyki częstotliwościowe dla obiektu inercyjnego I rzędu przedstawiono na listingu 2.9.

Listing 2.9: Wyznaczenie charakterystyk częstotliwościowych dla członu inercyjnego i rzędu

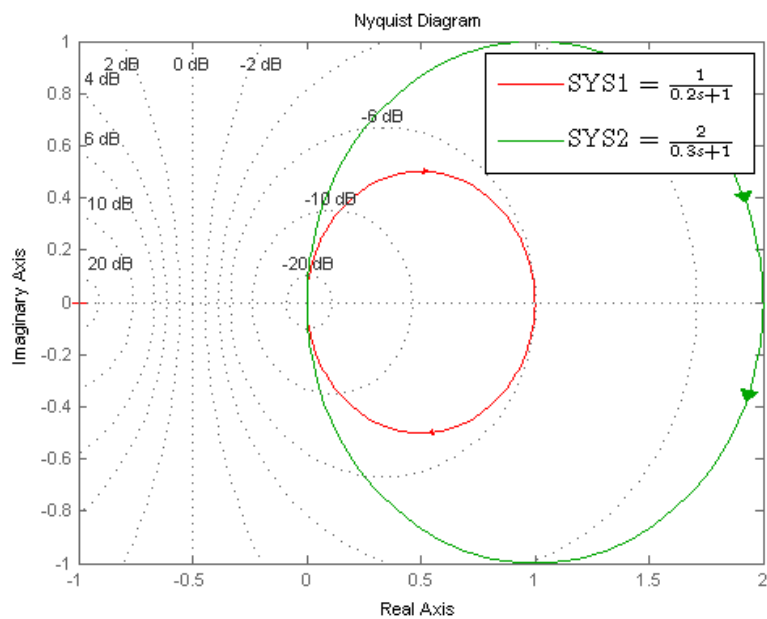
```

1 %Okreslenie transmitancji operatorowej czlonow inercyjnychI rzędu
2 s=tf('s');
3 SYS1=1/(0.2*s+1); SYS2=2/(0.4*s+1);
4
5 %Wykreslenie charakterystyk Bode'go
6 figure(1);
7 bode(SYS1,'r',SYS2,'g'); grid on; hold on;
8 leg=legend('SYS1= $\frac{1}{0.2s+1}$ ', 'SYS2= $\frac{2}{0.3s+1}$ ');
9 leg.Interpreter = 'latex'; leg.FontSize = 20;
10
11 %Wykreslenie charakterystyki Nyquist'a
12 figure(2);
13 nyquist(SYS1,'r',SYS2,'g'); grid on; hold on;
14 leg=legend('SYS1= $\frac{1}{0.2s+1}$ ', 'SYS2= $\frac{2}{0.3s+1}$ ');
15 leg.Interpreter = 'latex'; leg.FontSize = 20;

```



Rysunek 2.5: Charakterystyki Bode'go członów inercyjnych I rzędu



Rysunek 2.6: Charakterystyki Nyquist'a członów inercyjnych I rzędu

2.6 Wyznaczanie charakterystyk czasowych układu

`paraphstep` -funkcja służąca do wyznaczania charakterystyki czasowej odpowiedzi układu na wymuszenie w postaci skoku jednostkowego.

$$\text{step}(\text{SYS1}, \text{SYS2} \dots, \text{TFINAL})$$

$$\text{step}(\text{SYS1}, \text{SYS2} \dots, T)$$

`impulse` -funkcja służąca do wyznaczania charakterystyki czasowej odpowiedzi układu na wymuszenie w postaci impulsu Dirac'a.

$$\text{impulse}(\text{SYS1}, \text{SYS2} \dots, \text{TFINAL})$$

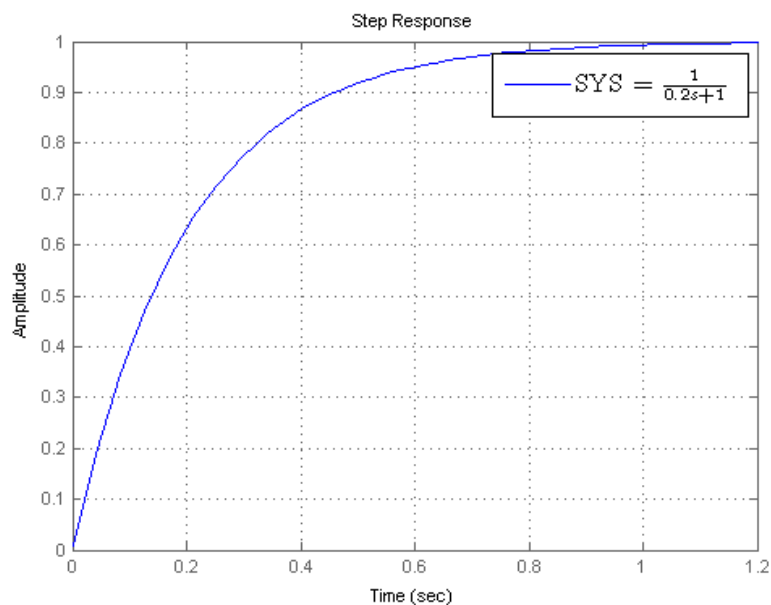
$$\text{impulse}(\text{SYS1}, \text{SYS2} \dots, T)$$

gdzie parametr TFINAL , oraz wektor współczynników T są opcjonalne i określają czas symulacji. W przypadku podania parametru TFINAL wykreslana jest odpowiedź w przedziale czasu od $t=0$ do $t=\text{TFINAL}$.

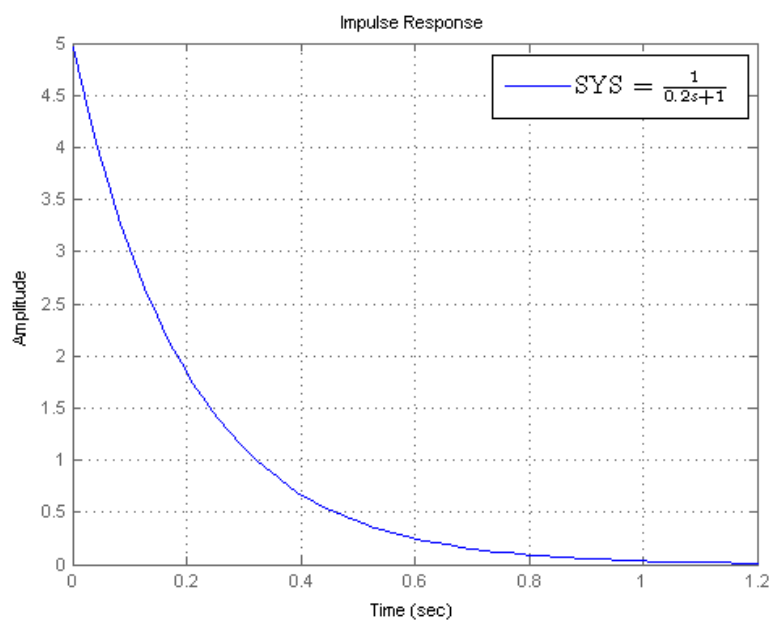
Listing 2.10: Wyznaczenie charakterystyk czasowych

```

1 %Okreslenie transmitancji operatorowej czlonow inercyjnychI rzędu
2 s=tf('s');
3 SYS=1/(0.2*s+1);
4
5 %Odpowiedz na wymuszenie skokowe
6 figure(1);
7 step(SYS);
8 grid on;
9 hold on;
10 leg=legend('SYS□=□$\frac{1}{0.2s+1}$');
11 leg.Interpreter = 'latex';
12 leg.FontSize = 20;
13
14 %Odpowiedz na wymuszenie impulsowe
15 figure(2);
16 impulse(SYS);
17 grid on;
18 hold on;
19 leg=legend('SYS□=□$\frac{1}{0.2s+1}$');
20 leg.Interpreter = 'latex';
21 leg.FontSize = 20;
```



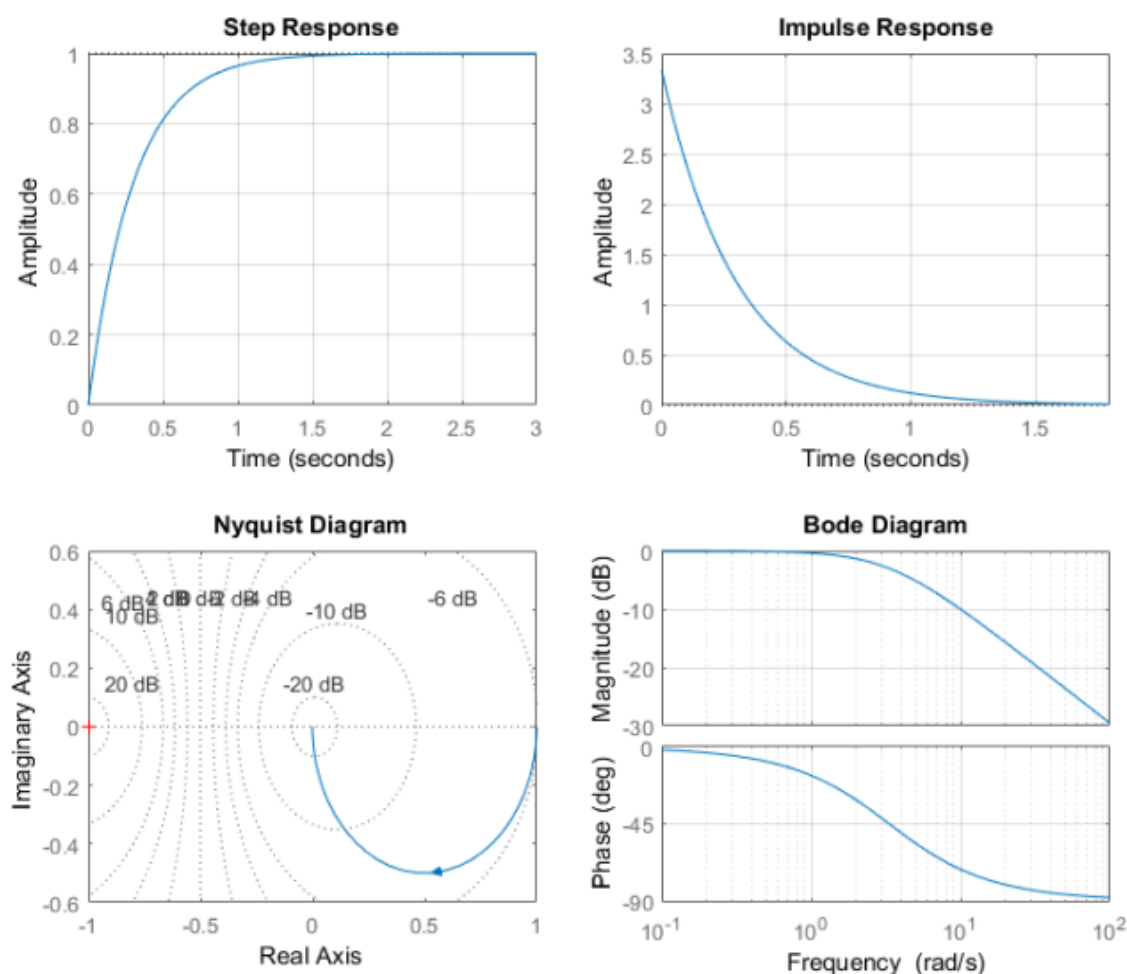
Rysunek 2.7: Odpowiedź układu na wymuszenie skokowe



Rysunek 2.8: Odpowiedź układu na wymuszenie impulsowe

2.7 Narzędzie Linear System Analyzer

Pakiet Control System Toolbox zawiera również ciekawe narzędzie graficzne do analizy systemów liniowych – Linear System Analyzer. Narzędzie to pozwala tworzyć wybrane charakterystyki badanego obiektu przy pomocy narzędzia graficznego. Charakterystyki podstawowych członów dynamicznych uruchomić można poleceniem `ltiview(SYS)` gdzie *SYS* to model obiektu który użytkownik chce analizować.



Rysunek 2.9: Przykład użycia narzędzia Linear System Analyzer

3 Zadania do wykonania

Zadanie 1 Utworzyć skrypt który dla obiektu inercyjnego I rzędu o transmitancji $G(s)$ wyznaczy:

- wartości macierzy $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ obiektu w przestrzeni stanu,
- wartości zer, biegunów oraz wzmocnień.

$$G(s) = \frac{1.2}{1 + 0.3s}$$

Zadanie 2 Utworzyć skrypt wyznaczający transmitancję obiektu opisanego równaniami stanu:

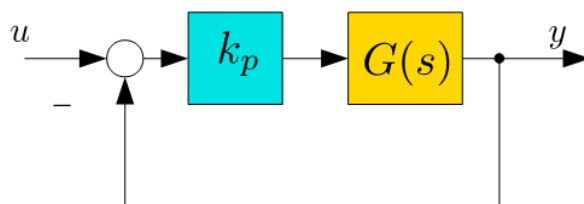
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 2 \\ -2 & -1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y} = \begin{bmatrix} -3 & 1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 2 \end{bmatrix} \mathbf{u}(t)$$

Zadanie 3 Obiekt o transmitancji $G(s)$ pracuje w układzie zamkniętym z regulatorem typu P - jak na rysunku poniżej.

$$G(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

Korzystając z funkcji pakietu Control System Toolbox zbadać stabilność układu dla wzmoc-



nień regulatora $k_p = 2, 3, 4$.

Zadanie 4 Dla układu z zadania poprzedniego wyznaczyć charakterystykę Nyquista oraz odpowiedź na wymuszenie skokowe. Omówić otrzymane wyniki.

Zadanie 5 Dla układu z zadania 3 wyznaczyć rozkład zer i biegunów. Omówić otrzymane wyniki.

Zadanie 6 Utworzyć skrypt który wyznaczy charakterystyki odpowiedzi skokowej i impulsowej członu opóźniającego o transmitancji $G(s)$

$$G(s) = k \cdot e^{-sL}$$

Zadanie 7 Wyznaczyć charakterystyki Bode'go i Nyquist'a dla obiektów o transmitancjach $G_1(s)$, $G_2(s)$, $G_3(s)$.

$$G_1(s) = \frac{k \cdot e^{-0.3s}}{1 + 3.2s}$$

$$G_2(s) = \frac{3}{s}$$

$$G_3(s) = \frac{1}{(0.2s + 1)(0.3s + 1)}$$

Zadanie 8 Wyznaczyć odpowiedź układu o transmitancji $G_3(s)$ z zadania poprzedniego na wymuszenie skokowe. Wykorzystać narzędzie Linear System Analyzer.

Zadanie 9 Wyznaczyć charakterystyki Nyquist'a i Bode'go układu opisanego równaniem:

$$0,3 \frac{d^2 y(t)}{dt^2} + \frac{dy(t)}{dt} = 2x(t)$$