

**Wydział Elektryczny**  
**Zespół Automatyki (ZTMAiPC)**  
**ZERiA**

## LABORATORIUM MODELOWANIA I SYMULACJI

### Ćwiczenie 4

#### PODSTAWOWE UKŁADY DYNAMICZNE

**Program ćwiczenia:**

##### 1. Podstawowe wymuszenia w dziedzinie czasu

Utworzyć i uruchomić skrypt **podstawowe\_wymuszenia.m**

```
%podstawowe wymuszenia w dziedzinie czasu
%ten skrypt uruchamiamy tylko raz
t=0:0.1:10; %wektor czasu
wymuszenie_skokowe=0*t+1; %wymuszenie skokowe
wymuszenie_predkosciowe=0.1*t; %wymuszenie liniowo narastające
plot(t,wymuszenie_skokowe,'r-',t,wymuszenie_predkosciowe,'g:');
axis([0,10,0,1.3])
xlabel('czas [s]');
ylabel('1(t), 0.1*t*1(t)')
legend('wymuszenie skokowe 1(t)', 'wymuszenie prędkościowe 0.1*t*1(t)')
```

##### 2. Układ proporcjonalny

Utworzyć i uruchomić skrypt **uklad\_proporcjonalny.m**

```
%układ proporcjonalny
% G(s)=b0/a0
opis=['b--'; 'r:x'; 'm.-'];
clc
a0=5 %przeprowadzić symulacje dla a) a0=5; b) a0=3; c) a0=4
b0=[1,5,10] %przeprowadzić symulacje dla a) b0=1,5,10; b) b0=2,4,8; c) b0=1,6,12
for i=1:3
    %definicja systemu
    licznik=[b0(i)]
    mianownik=[a0]
    system=tf(licznik,mianownik)
    % odpowiedź impulsowa
    figure(10)
    axis([0,10,-2,2]) %dopasować zakresy osi
    hold on
    grid on
    [wy,czas]=impulse(system,max(t));
    plot(czas,wy,opis(i,:))
    xlabel('czas [s]');
    ylabel(['b0=', num2str(b0)])
    title('odpowiedź impulsowa')
```

```

    %odpowiedź skokowa
    figure(11)
    axis([0,10,0,5])      % dopasować zakresy osi
    hold on
    grid on
    [wy,czas]=step(system,max(t));
    plot(czas,wy,opis(i,:))
    xlabel('czas [s]');
    ylabel(['b0=',num2str(b0)])
    title('odpowiedź skokowa')

    %odpowiedź prędkościowa
    figure(12)
    axis([0,10,0,2.5])    % dopasować zakresy osi
    hold on
    grid on
    [wy,czas]=lsim(system,wymuszenie_predkosciowe,t);
    plot(czas,wy,opis(i,:))
    xlabel('czas [s]');
    ylabel(['b0=',num2str(b0)])
    title('odpowiedź prędkościowa')

    %charakterystyki częstotliwościowe Bodego
    figure(13)
    axis([0,10,0,2.5])    % dopasować zakresy osi
    hold on
    grid on
    bode(system,opis(i,:));
    grid on
    title('charakterystyki częstotliwościowe Bodego')

end

```

### 3. Układ inercyjny pierwszego rzędu

Utworzyć i uruchomić skrypt `uklad_inercyjny_p_rz.m`

```

%układ inercyjny pierwszego rzędu
%G(s)=k/Ts+1
opis=['b--';'r:x';'m.-'];
clc
k=1      %przeprowadzić symulacje dla a) k=1; b) k=2; c) k=0.5
T=[0.5,2,5] %przeprowadzić symulacje dla a) T=0.5,2,5; b) T=1,3,6 c) T=0.25,1,4

for i=1:3
    %definicja systemu
    licznik=[k]
    mianownik=[T(i),1]

    system=tf(licznik,mianownik)
    % odpowiedź impulsowa
    figure(10)
    axis([0,10,-2,2])    % dopasować zakresy osi
    hold on
    grid on
    [wy,czas]=impulse(system,max(t));
    plot(czas,wy,opis(i,:))
    xlabel('czas [s]');
    ylabel(['T=',num2str(T)])
    title('odpowiedź impulsowa')

```

```

%odpowiedź skokowa
figure(11)
axis([0,10,0,2])      % dopasować zakresy osi
hold on
grid on
[wy,czas]=step(system,max(t));
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź skokowa')

%odpowiedź prędkościowa
figure(12)
axis([0,10,0,2])      % dopasować zakresy osi
hold on
grid on
[wy,czas]=lsim(system,wymuszenie_predkosciowe,t);
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź prędkościowa')

%charakterystyki częstotliwościowe Bodego
figure(13)
axis([0.01,100,-90,30]) % dopasować zakresy osi
hold on
grid on
bode(system,opis(i,:));
title('charakterystyki częstotliwościowe Bodego')

end

```

#### 4. Układ całkujący idealny

Utworzyć i uruchomić skrypt **uklad\_calukujacy\_id.m**

```

%układ całkujący idealny
%G(s)=k/Ts
opis=['b--';'r:x';'m.-'];
clc
k=1      %przeprowadzić symulacje dla a) k=1; b) k=2; c) k=0.5
T=[0.5,2,5] %przeprowadzić symulacje dla a) T=0.5,2,5; b) T=1,3,6 c) T=0.25,1,4

for i=1:3
    %definicja systemu
    licznik=[k]
    mianownik=[T(i),0]

    system=tf(licznik,mianownik)
    % odpowiedź impulsowa
    figure(10)
    axis([0,10,-2,2.5]) % dopasować zakresy osi
    hold on
    grid on
    [wy,czas]=impulse(system,max(t));
    plot(czas,wy,opis(i,:))
    xlabel('czas [s]');
    ylabel(['T=',num2str(T)])
    title('odpowiedź impulsowa')

```

```

%odpowiedź skokowa
figure(11)
axis([0,10,0,2.5]) % dopasować zakresy osi
hold on
grid on
[wy,czas]=step(system,max(t));
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź skokowa')

%odpowiedź predkosciowa
figure(12)
axis([0,10,0,2.5]) % dopasować zakresy osi
hold on
grid on
[wy,czas]=lsim(system,wymuszenie_predkosciowe,t);
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź prędkościowa')

%charakterystyki częstotliwościowe Bodego
figure(13)
axis([0.01,100,-90,30]) % dopasować zakresy osi
hold on
grid on
bode(system,opis(i,:));
title('charakterystyki częstotliwościowe Bodego')
end

```

## 5. Układ całkujący rzeczywisty

Utworzyć i uruchomić skrypt **uklad\_calukujacy\_rz.m**

```

%układ całkujący rzeczywisty
%G(s)=k/s(Ts+1)
opis=['b--';'r:x';'m.-'];
clc
k=1 %przeprowadzić symulacje dla a) k=1; b) k=2; c) k=0.5
T=[0.5,2,5] %przeprowadzić symulacje dla a) T=0.5,2,5; b) T=1,3,6 c) T=0.25,1,4

for i=1:3
%definicja systemu
licznik=[k]
mianownik=[T(i),1 0]

system=tf(licznik,mianownik)
% odpowiedź impulsowa
figure(10)
axis([0,10,-2,2.5]) % dopasować zakresy osi
hold on
grid on
[wy,czas]=impulse(system,max(t));
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź impulsowa')
%odpowiedź skokowa

```

```

figure(11)
axis([0,10,0,2.5])      % dopasować zakresy osi
hold on
grid on
[wy,czas]=step(system,max(t));
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź skokowa')

%odpowiedź prędkościowa
figure(12)
axis([0,10,0,2.5])      % dopasować zakresy osi
hold on
grid on
[wy,czas]=lsim(system,wymuszenie_predkosciowe,t);
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź prędkościowa')

%charakterystyki częstotliwościowe Bodego
figure(13)
axis([0.01,100,-200,100]) % dopasować zakresy osi
hold on
grid on
bode(system,opis(i,:));
title('charakterystyki częstotliwościowe Bodego')

end

```

## 6. Układ różniczkujący rzeczywisty

Utworzyć i uruchomić skrypt **uklad\_rozniczkujacy\_rz.m**

```

%układ różniczkujący rzeczywisty
%G(s)=ks/Ts+1
opis=['b--';'r:x';'m.-'];
clc
k=1          %przeprowadzić symulacje dla a) k=1; b) k=2; c) k=3
T=[0.5,2,5] %przeprowadzić symulacje dla a) T=0.5,2,5; b) T=1,3,6 c) T=0.25,1,4

for i=1:3
    %definicja systemu
    licznik=[k 0]
    mianownik=[T(i),1]

    system=tf(licznik,mianownik)
    % odpowiedź impulsowa
    figure(10)
    axis([0,10,-5,2]) % dopasować zakresy osi
    hold on
    grid on
    [wy,czas]=impulse(system,max(t));
    plot(czas,wy,opis(i,:))
    xlabel('czas [s]');
    ylabel(['T=',num2str(T)])
    title('odpowiedź impulsowa')

```

```

%odpowiedź skokowa
figure(11)
axis([0,10,0,2])    % dopasować zakresy osi
hold on
grid on
[wy,czas]=step(system,max(t));
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź skokowa')

%odpowiedź prędkościowa
figure(12)
axis([0,10,0,0.2])    % dopasować zakresy osi
hold on
grid on
[wy,czas]=lsim(system,wymuszenie_predkosciowe,t);
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['T=',num2str(T)])
title('odpowiedź prędkościowa')

%charakterystyki częstotliwościowe Bodego
figure(13)
axis([0.01,100,-60,100])    % dopasować zakresy osi
hold on
grid on
bode(system,opis(i,:));
title('charakterystyki częstotliwościowe Bodego')

end

```

## 7. Układ oscylacyjny

Utworzyć i uruchomić skrypt **uklad\_oscylacyjny.m**

```

%układ oscylacyjny
%G(s)=omega^2/s^2+2*zeta*omega+omega^2;
opis=['b--';'r:x';'m.-'];
clc
omega=2*pi/1;
k=omega.*omega
zeta=[0.2,1,5]    %przeprowadzić symulacje dla a)zeta=0.2,1,4; b)zeta=0.4,2,6;
                    %                                c)zeta=0.1,1.5,3

for i=1:3
    %definicja systemu
    licznik=[k]
    mianownik=[1,2.*omega.*zeta(i),k]

    system=tf(licznik,mianownik)
    % odpowiedź impulsowa
    figure(10)
    axis([0,10,-3,5])    % dopasować zakresy osi
    hold on
    grid on
    [wy,czas]=impulse(system,max(t));
    plot(czas,wy,opis(i,:))
    xlabel('czas [s]');
    ylabel(['zeta=',num2str(zeta)])

```

```
title('odpowiedź impulsowa')

%odpowiedź skokowa
figure(11)
axis([0,10,0,2]) % dopasować zakresy osi
hold on
grid on
[wy,czas]=step(system,max(t));
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['zeta=',num2str(zeta)])
title('odpowiedź skokowa')

%odpowiedź prędkościowa
figure(12)
axis([0,10,0,1]) % dopasować zakresy osi
hold on
grid on
[wy,czas]=lsim(system,wymuszenie_predkosciowe,t);
plot(czas,wy,opis(i,:))
xlabel('czas [s]');
ylabel(['zeta=',num2str(zeta)])
title('odpowiedź prędkościowa')

%charakterystyki częstotliwościowe Bodego
figure(13)
axis([0.01,10000,-180,30]) % dopasować zakresy osi
hold on
grid on
bode(system,opis(i,:));
title('charakterystyki częstotliwościowe Bodego')

end
```

## Sprawozdanie

1. Dla układów dynamicznych napisać ogólne wzory na transmitancje
2. Przedstawić transmitancje badanych układów oraz otrzymane na ich podstawie charakterystyki czasowe i częstotliwościowe.
3. Dla otrzymanych wykresów określić wpływ parametrów transmitancji (np. stałej czasowej) na przebiegi charakterystyk czasowych i częstotliwościowych.

## Literatura

1. B. Mrozek, Z. Mrozek: *MATLAB i Simulink: poradnik użytkownika*. Helion, Gliwice, 2004.
2. A. Zalewski, R. Cegieła: *Matlab - obliczenia numeryczne i ich zastosowania*. Wydawnictwo Nakom, Poznań, 2000.
3. J. Brzózka, L. Dorobczyński: *Programowanie w Matlab*. Wydawnictwo Mikom, Warszawa, 1998.