

# Programowanie w środowisku Matlab.

## Projekt nr 2. Temat 220

Piotr Wachowicz, Grupa 37, numer indexu: 289746

12 lutego 2020

### 1 Opis programu

Celem tego projektu jest zamodelowanie w środowisku Matlab wykorzystując program Simulink układu opisanego transmitancją:

$$G(s) = \frac{-s^2 + 2s + 1}{(3s + 1)(5s^2 + 5s + 1)} e^{-9s}$$

i wyznaczenie odpowiedzi na wymuszenie skokowe, pulsowe, zbudowanie układu regulacji z regulatorem PID, dobrać nastawy regulatora, zamodelować układ regulacji z sygnałem zadanym i zakłóceniem oraz wyznaczyć wskaźniki jakości regulacji. Dane do programu Simulink powinny być wprowadzone z programu Matlab

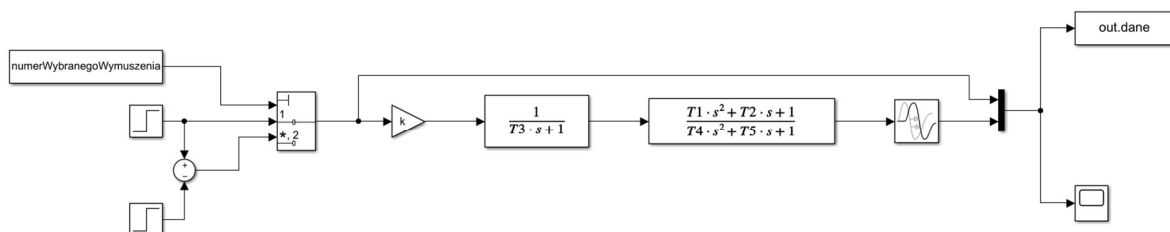
### 2 Sparametryzowanie transmitancji

W celu ułatwienia obliczeń parametry transmitancji zostały uzmiennione:

$$G(s) = k \frac{T_1 s^2 + T_2 s + 1}{(T_3 s + 1)(T_4 s^2 + T_5 s + 1)} e^{-T_0 s}$$

gdzie:  $k = 1, T_1 = -1, T_2 = 2, T_3 = 3, T_4 = 5, T_5 = 5, T_0 = 9$

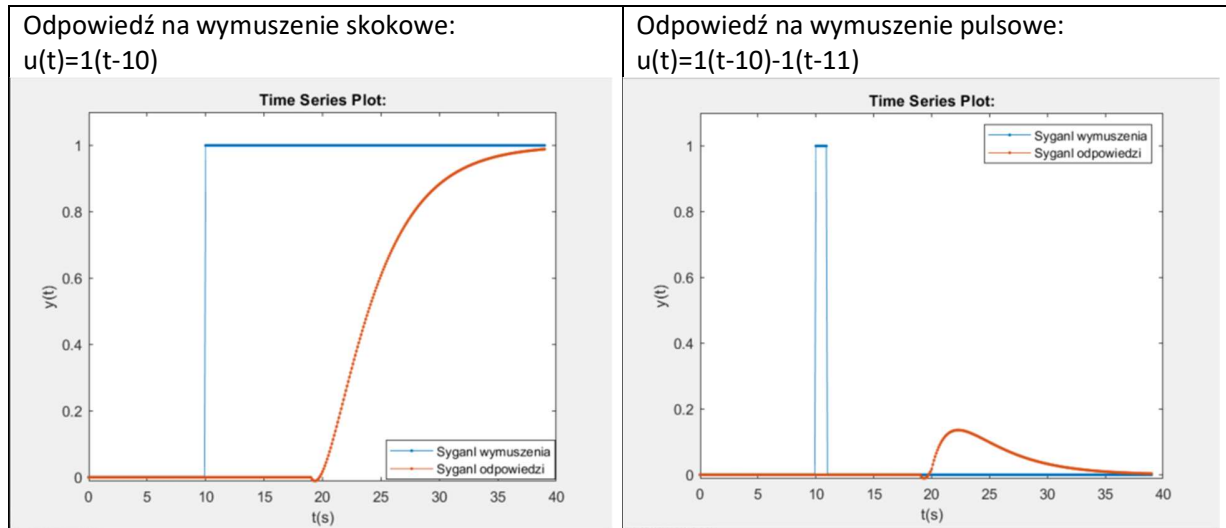
### 3 Układ bez regulatora



Schemat 1 Schemat modelu bez regulacji symulacji z programu Simulink

Przy pomocy programu Matlab ustawiono wszystkie parametry modelu projektu, aby można było wyznaczyć daną charakterystykę wykorzystując program Simulink. Ustawiono m. in.

- Parametry transmitancji, w tym wzmacnienie i opóźnienie transportowe układu
- Parametr numerWybranegoWymuszenia odpowiadający za wybór wymuszenia skokowego lub pulsowego

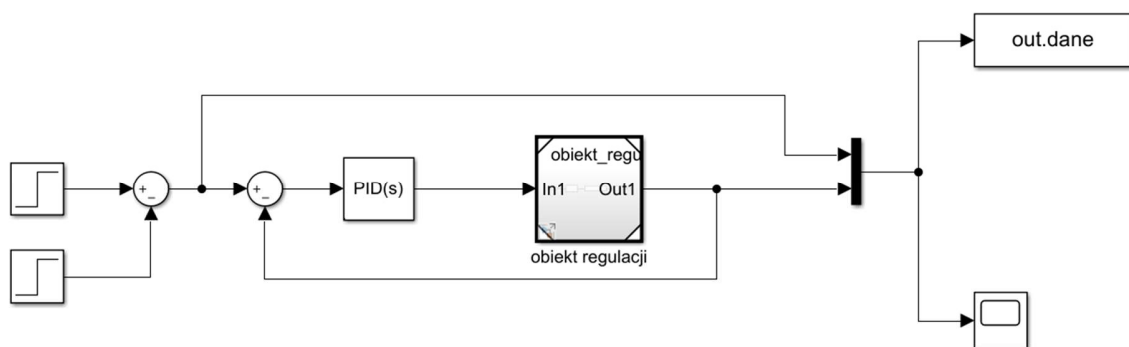


Powyższe charakterystyki obrazują jak obiekt regulacji reaguje na poszczególne wymuszenia. Przedstawiają charakter badanego obiektu.

## 4 Układ z regulatorem PID

### Dobór nastaw regulatora PID metodą Zieglera – Nicholasa

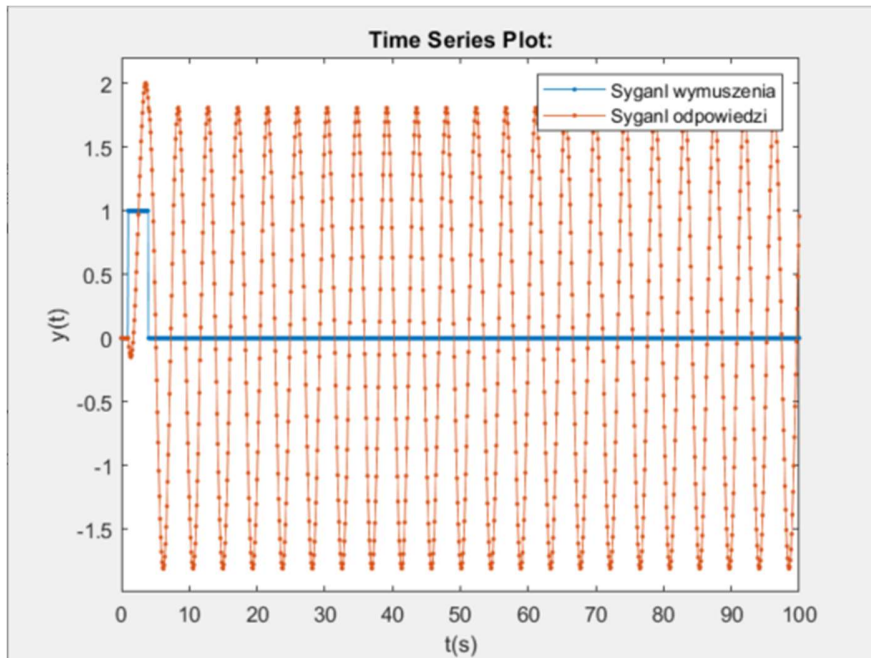
Doboru nastaw dokonano metodą wzbudzenia układu, wykorzystując układ ze schematu 2.



Schemat 2 Model do wyznaczenia nastaw regulatora PID metodą Z-N

Stabilność układu nie zależy od opóźnienia dlatego na czas wyznaczania nastaw regulatora pominięto je. Najpierw dobrano czas impulsu na 3s i znaleziono zgrubny przedział w którym mieści się  $k_{kryt}$  - (12,20). Następnie uruchomiono Program 2 w którym  $k_{kryt}$  zostało znaleziono metodą połowienia przedziałów z

dużą dokładnością już po 20 iteracjach. Sprawdzanie czy oscylacje było malejące czy rosnące było wykonywane przez porównywanie amplitudy drugiego i ostatniego zarejestrowanego maksimum lokalnego. To pozwalało uzyskać najlepszą dokładność.  $T_{osc}$  zostało znalezione jako odległość między kolejnymi ekstremami na osi x.



Wykres 1 Układ doprowadzony do niegasnących oscylacji.  $K_{kryt}=12.303901672363281$

Uzyskane wartości to:

$$k_{kryt} = 12.303901672363281$$

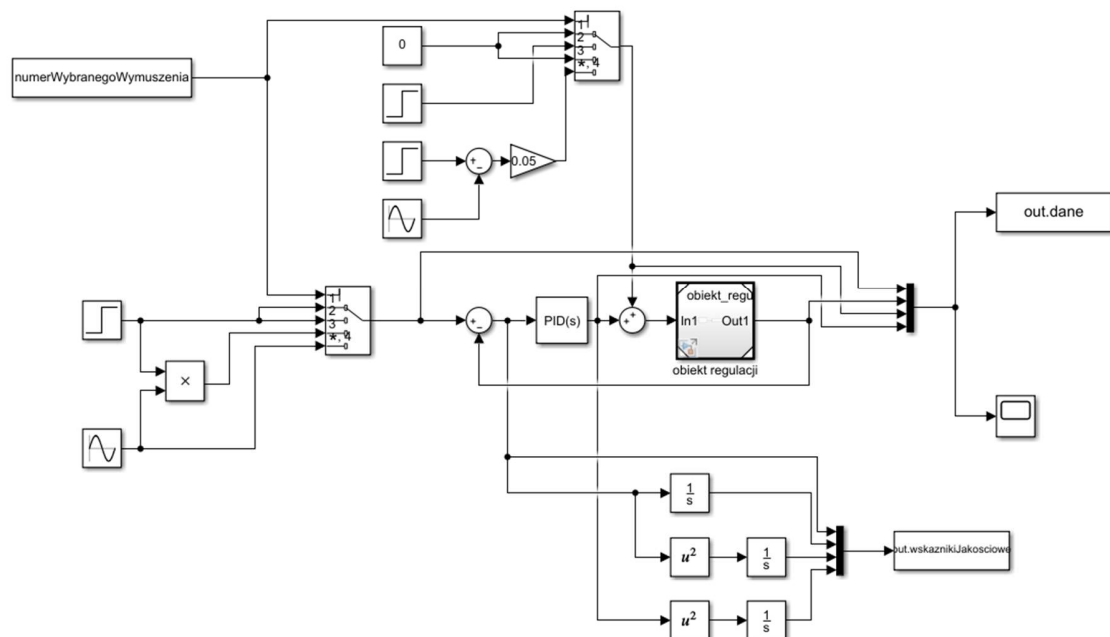
$$T_{osc} = 4,4s$$

$$k_{ZN} = 0,6 \cdot k_{kryt} = 7.382341003417968$$

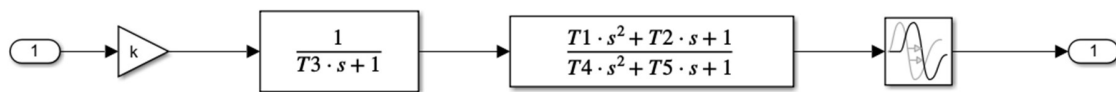
$$kiZN = \frac{1}{0,5 \cdot T_{osc}} = 0.454545454545455$$

$$kdZN = \frac{1}{0,12 \cdot T_{osc}} = 1.893939393939394$$

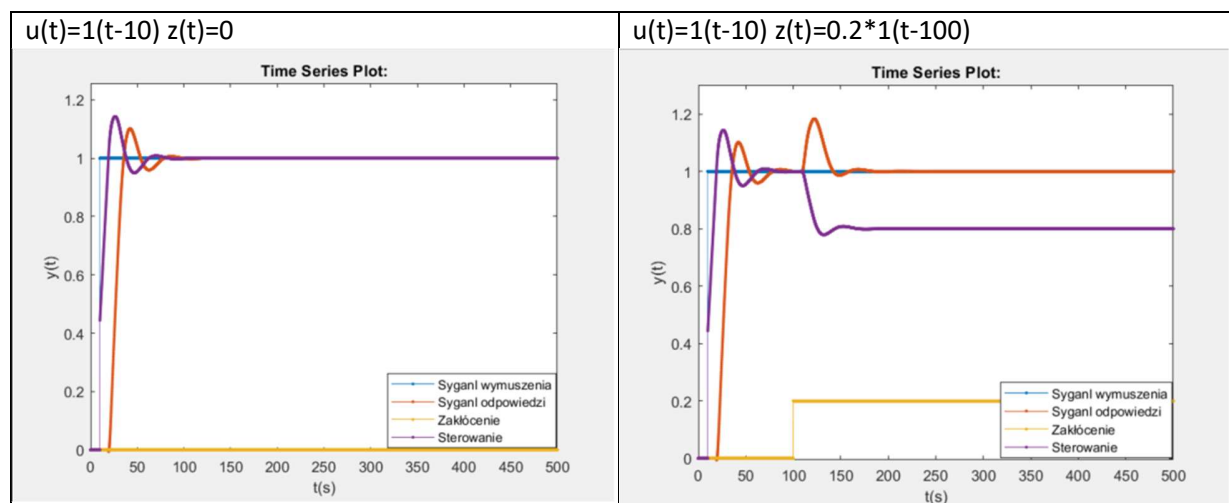
## Wyznaczenie odpowiedzi układu z regulatorem na wymuszenia



Schemat 4 Model układu z regulatorem



Schemat 3 Model obiektu regulacji



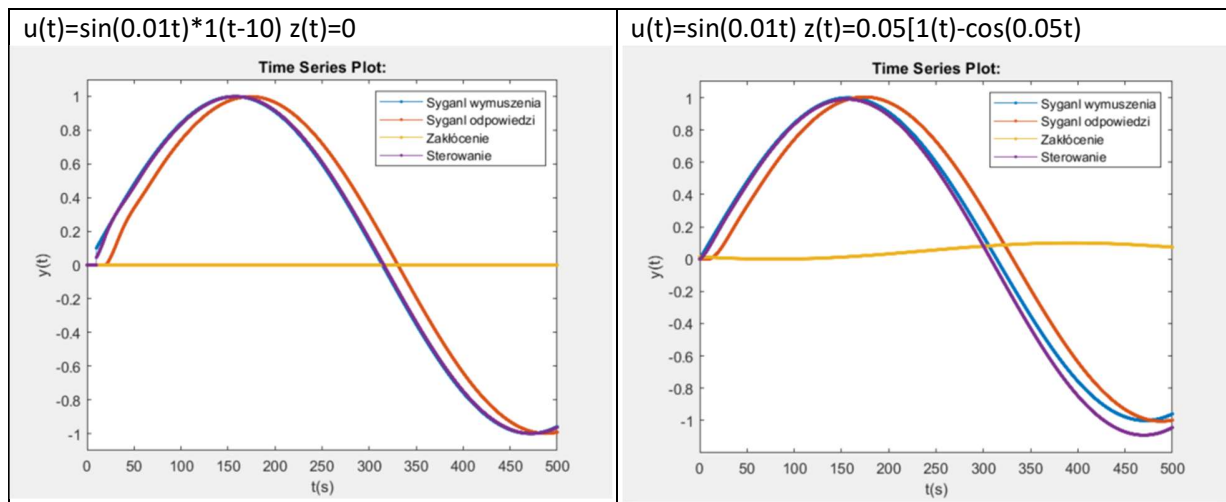


Tabela 1 Sterowanie i odpowiedzi układu z regulatorem dla różnych wymuszeń i zakłóceń

### Wskaźniki jakości regulacji

$e_m$  – maksymalna odchyłka dynamiczna

$t_r$  - **czas regulacji** – czas określony jako czas od chwili wprowadzenia pobudzenia do chwili, gdy odchyłka regulacji  $e(t)$  osiąga wartości mieszczące się w strefie tolerancji  $\pm\Delta$ . Wartość  $\Delta$  określa się jako  $\Delta = 0,05e_m$

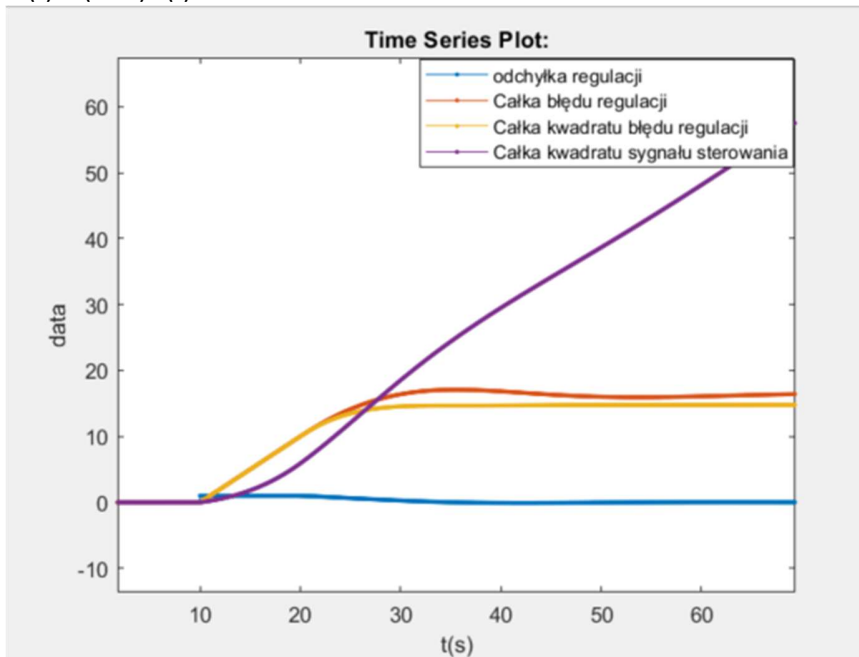
$K$  – **przeregulowanie** określa w procentach stosunek amplitudy drugiego odchylenia  $e_2$  do amplitudy pierwszego odchylenia  $e_1$  zgodnie ze wzorem  $K = \frac{e_2}{e_1} \cdot 100\%$

**Średni błąd regulacji** -  $e_{Avg} = \frac{1}{T} \int_0^T e(t) dt$

**Całka kwadratu błędu regulacji** w wyznaczonym czasie regulacji -  $e_{square} = \int_0^T e^2(t) dt$

**Energia sterowania** w wyznaczonym czasie regulacji –  $W = \int_0^T x^2(t) dt$

$$u(t)=1(t-10) \quad z(t)=0$$



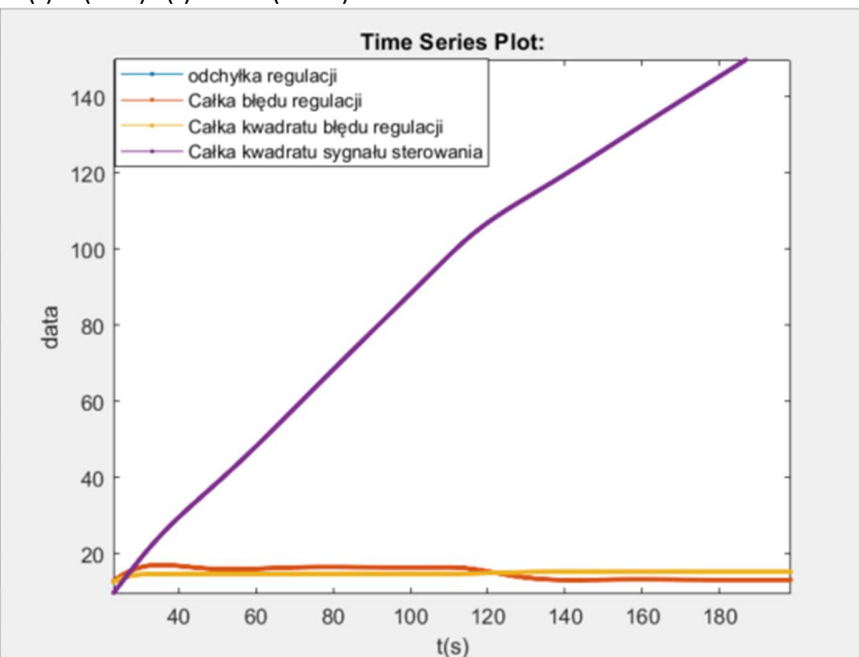
Przeregulowanie =  
10.072359516716723%

Średni błąd regulacji =  
0.003292191550915

Całka kwadratu błędu  
regulacji =  
14.774057619370755

Energia sterowania =  
4.882929616014422e+02

$$u(t)=1(t-10) \quad z(t)=0.2*1(t-100)$$



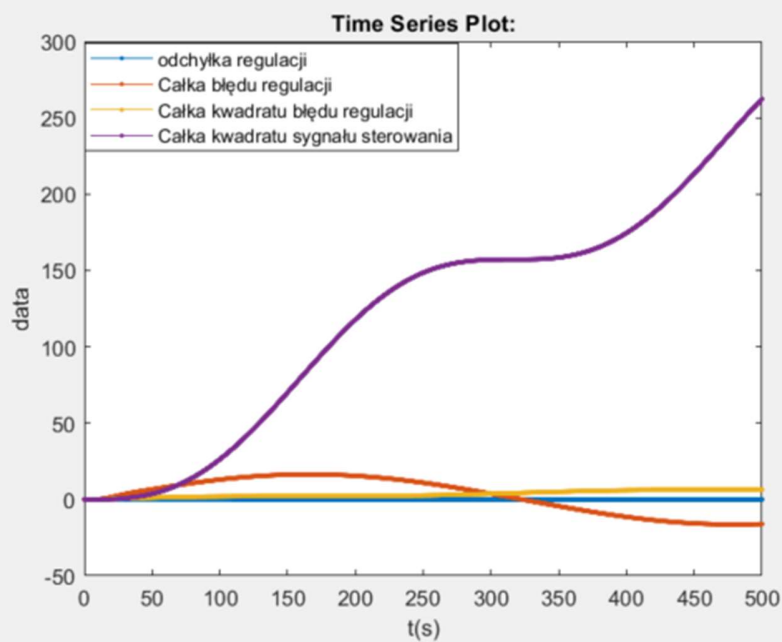
Przeregulowanie =  
18.193920165883672%

Średni błąd regulacji =  
0.002633753240730

Całka kwadratu błędu  
regulacji =  
15.240691478966239

Energia sterowania =  
3.501352198862078e+02

$$u(t)=\sin(0.01t)*1(t-10) \quad z(t)=0$$



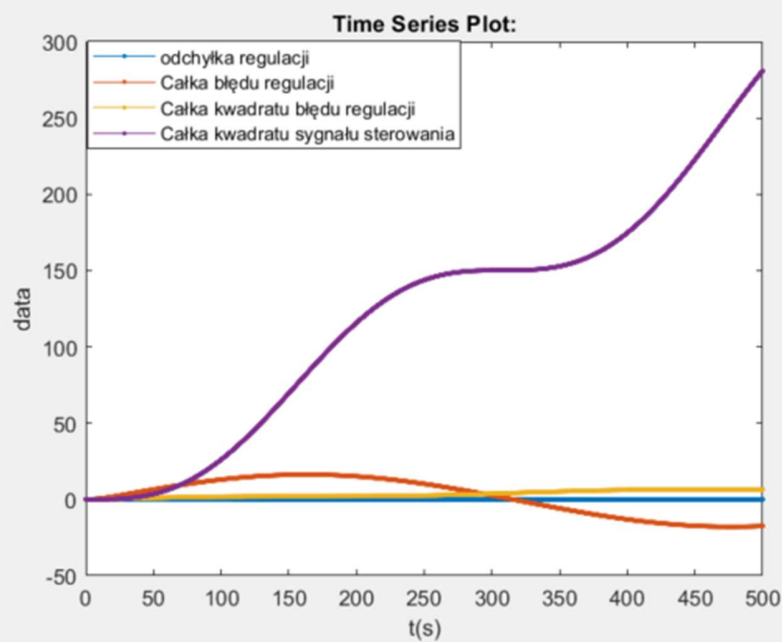
Przeregulowanie =  
80.381615369151078%

Średni błąd regulacji = -  
0.003217936520908

Całka kwadratu błędu  
regulacji =  
6.423863449025244

Energia sterowania =  
2.623607059759073e+02

$$u(t)=\sin(0.01t) \quad z(t)=0.05[1(t)-\cos(0.05t)]$$



Przeregulowanie =  
1.066712096788440e+02  
%

Średni błąd regulacji = -  
0.003492585658074

Całka kwadratu błędu  
regulacji =  
6.618885341920429

Energia sterowania =  
2.809662459711157e+02

## 5 Kod programu

### Program 1. Właściwy program

```
clear;

prompt={'k:', 'T_1:', 'T_2:', 'T_3:', 'T_4:', 'T_5:', 'T_0:'};
name='Input';
numlines=1;
defaultanswer={'1', '-1', '2', '3', '5', '5', '9'};
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';

answer=inputdlg(prompt,name,numlines,defaultanswer,options);

k=str2num(answer{1,1});
T1=str2num(answer{2,1});
T2=str2num(answer{3,1});
T3=str2num(answer{4,1});
T4=str2num(answer{5,1});
T5=str2num(answer{6,1});
T0=str2num(answer{7,1});

Tp = 0.1;
Tsymulacji = 500;
numberOfDataSamples = Tsymulacji / Tp + 1;

regulowany = menu('Czy układ ma zawierać regulator PID?', 'tak',
'nie');
switch (regulowany)
case 1
    odpowiedziLubParametry = menu('Co chcesz zrobić?', 'Obejrzeć
odpowiedzi układu', 'Odczytać wskaźniki jakości regulacji');
    numerWybranegoWymuszenia = menu('Wybierz wymuszenie i zakłócenie',
'u(t)=1(t-10) z(t)=0', 'u(t)=1(t-10) z(t)=0.2*1(t-100)',
'u(t)=sin(0.01t)*1(t-10) z(t)=0', 'u(t)=sin(0.01t) z(t)=0.05[1(t)-
cos(0.05t)]');
    sim('zRegulatorem',Tsymulacji);

    tsdane = getdatasamples(ans.dane,1:numberOfDataSamples);
    wymuszenie = tsdane(:,1);
    odpowiedz = tsdane(:,2);
    zaklocenie = tsdane(:,3);
```



```

sterowanie = tsdane(:,4);
% [wymuszenie odpowiedz zaklocenie sterowanie]

switch (odpowiedziLubParametry)
case 1
    plot(ans.dane, '-.')
    xlabel('t(s)');
    ylabel('y(t)');
    legend('Wymuszenie','Odpowiedź','Zakłócenie', 'Sterowanie');
    [down up] = limits(ans.dane);
    ylim([down up]);
case 2
    %wskazniki jakosciowe
    tswskazniki =
getdatasamples(ans.wskaznikiJakosciowe,1:numberOfDataSamples);
    e = tswskazniki(:,1);
    Ie = tswskazniki(:,2);
    Ie2 = tswskazniki(:,3);
    ICV2 = tswskazniki(:,4);
    % [e Ie Ie2 ICV2]

    plot(ans.wskaznikiJakosciowe, '-.')
    xlabel('t(s)');
    legend('odchyłka regulacji', 'Całka błędu regulacji', 'Całka
kwadratu błędu regulacji', 'Całka kwadratu sygnału sterowania');

    % %czas regulacji
    % eMaksymalne = max(e)
    % eGraniczne = 0.05 * eMaksymalne
    % indeksWprowadzeniaWymuszenia = indeksSkoku(wymuszenie)
    %
    % % indeksWprowadzeniaZaklocenia = indeksSkoku(zaklocenie);
    % % momentWprowadzeniegoSygnału =
max(indeksWprowadzeniaWymuszenia, indeksWprowadzeniaZaklocenia);
    % momentWprowadzeniegoSygnału = indeksWprowadzeniaWymuszenia
    %
    % indeksySygnałuUregulowanego =
find(e(momentWprowadzeniegoSygnału:numberOfDataSamples) < eGraniczne)
    % indeksSygnałuUregulowanego = indeksySygnałuUregulowanego(1)
    % tRegulacji = Tp * indeksSygnałuUregulowanego

    %przeregulowanie wyrazone w %

```

```

    przeregulowanie = - min(e) / max(e) * 100

    %sredni blad regulacji
    eCumulative = Ie(numberOfDataSamples);
    sredniBladRegulacji = eCumulative / numberOfDataSamples

    %calka kwadratu bledu regulacji
    calkaKwadratuBleduRegulacji = Ie2(numberOfDataSamples)

    %energia sterowania
    energiaSterowania = ICV2(numberOfDataSamples)
end

case 2
    numerWybranegoWymuszenia = menu('Wybierz wymuszenie', 'wymuszenie
skokowe u(t)=1(t-10)', 'wymuszenie pulsowe u(t)=1(t-10)-1(t-11)');
    sim('bezRegulatora',Tsymulacji);

    tsdane = getdatasamples(ans.dane,1:numberOfDataSamples);
    wymuszenie = tsdane(:,1);
    odpowiedz = tsdane(:,2);
    zaklocenie = tsdane(:,3);
    sterowanie = tsdane(:,4);
    % [wymuszenie odpowiedz zaklocenie sterowanie]

    plot(ans.dane, '.-')
    xlabel('t(s)');
    ylabel('y(t)');
    legend('Wymuszenie','Odpowiedź','Zakłócenie', 'Sterowanie');
    [down up] = limits(ans.dane);
    ylim([down up]);
end

function [lower, upper] = limits(dane)
    minimum=min(min(dane));
    maximum=max(max(dane));
    if minimum > 0 & maximum > 0
        lower = 0.9*minimum;
        upper = 1.1*maximum;
    else
        if minimum < 0 & maximum > 0

```

```

        lower = 1.1*minimum;
        upper = 1.1*maximum;
    else
        lower = 1.1*minimum;
        upper = 0.9*maximum;
    end
end
end

% function indeks = indeksSkoku(tablica)
%     dlugosc = length(tablica);
%     wartoscPoczątkowa = tablica(1);
%     for n=2:dlugosc
%         if tablica(n) ~= wartoscPoczątkowa
%             indeks = n;
%             return;
%         end
%     end
% end

```

## Program 2. Program pozwalający dobrać nastawy regulatora PID metodą Zieglera - Nicholsa

```
clear;
format long;

prompt={'k:', 'T_1:', 'T_2:', 'T_3:', 'T_4:', 'T_5:', 'T_0:'};
name='Input';
numlines=1;
defaultanswer={'1', '-1', '2', '3', '5', '5', '0'};
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';

answer=inputdlg(prompt,name,numlines,defaultanswer,options);

k=str2num(answer{1,1});
T1=str2num(answer{2,1});
T2=str2num(answer{3,1});
T3=str2num(answer{4,1});
T4=str2num(answer{5,1});
T5=str2num(answer{6,1});
T0=str2num(answer{7,1});

kLower=12;
kUpper=20;

for numerSymulacji=1:20
    kR = (kLower + kUpper)/2
    sim('zRegulatorem',100);

    x = (0:.1:100)';
    tsdata = getdatasamples(ans.dane,1:1001);
    y = tsdata(:,2);
    [PKS,LOCS] = findpeaks(y,x);
    last = length(PKS);

    if PKS(last) < PKS(2)
        kLower = kLower + (kUpper-kLower)/2;
    else
        if PKS(last) > PKS(2)
            kUpper = kUpper - (kUpper-kLower)/2;
        end
    end
end

kR
Tosc = LOCS(3) - LOCS(2)

kZN = 0.6*kR
kiZN = 1/(0.5*Tosc)
kdZN = 1/(0.12*Tosc)
```

```

plot(ans.dane, '-.')
xlabel('t(s)');
ylabel('y(t)');
legend('Sygnał wymuszenia','Sygnał odpowiedzi');

[down up] = limits(ans.dane);
ylim([down up]);

function [lower, upper] = limits(dane)
    minimum=min(min(dane));
    maximum=max(max(dane));
    if minimum > 0 & maximum > 0
        lower = 0.9*minimum;
        upper = 1.1*maximum;
    else
        if minimum < 0 & maximum > 0
            lower = 1.1*minimum;
            upper = 1.1*maximum;
        else
            lower = 1.1*minimum;
            upper = 0.9*maximum;
        end
    end
end
end

```