Programowanie w środowisku Matlab.
Projekt nr 2. Temat 220

Piotr Wachowicz, Grupa 37, numer indexu: 289746

27 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

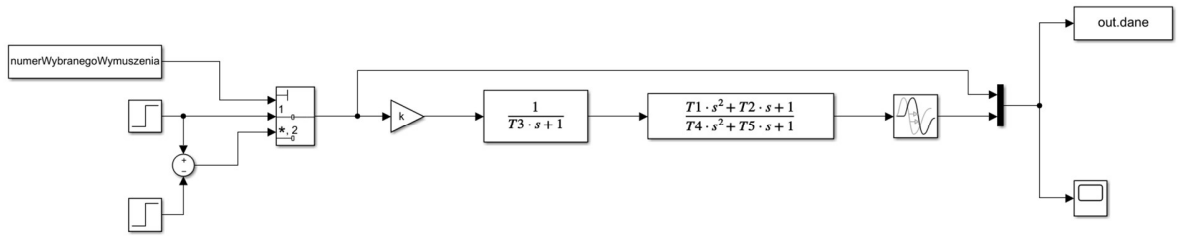
2 Sparametryzowanie transmitancji

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

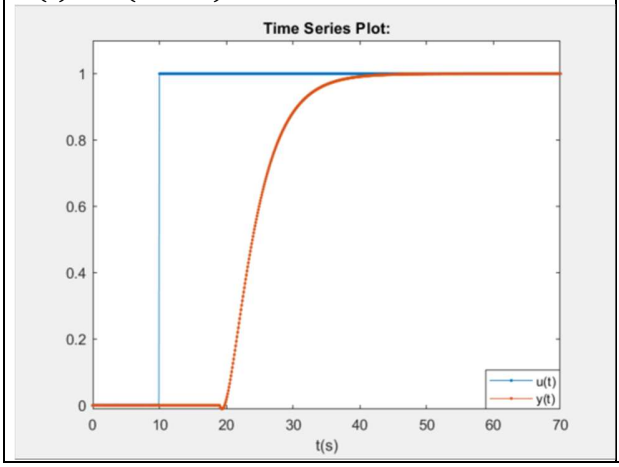
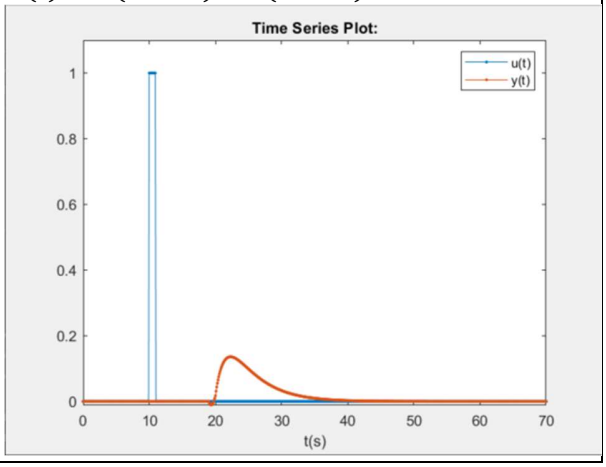
3 Układ bez regulatora



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. Ustawinono m. in.

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

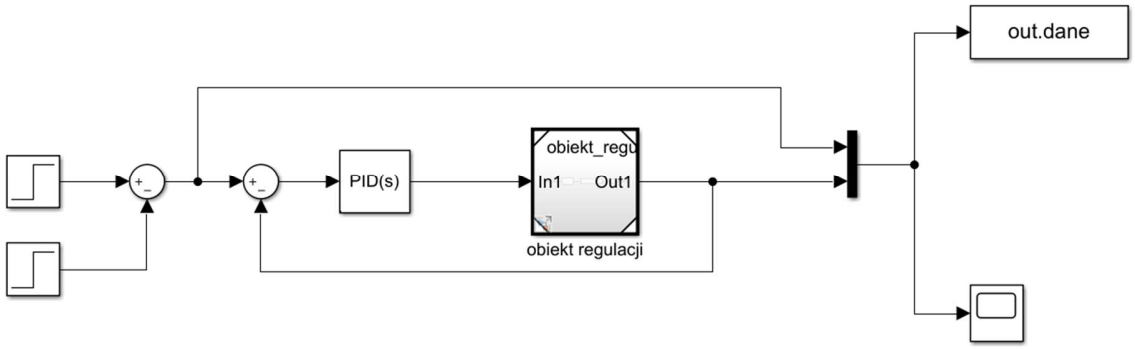
Odpowiedź na wymuszenie skokowe: $u(t) = 1(t-10)$	Odpowiedź na wymuszenie pulsowe: $u(t) = 1(t-10) - 1(t-11)$
	

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 – Nicholsa

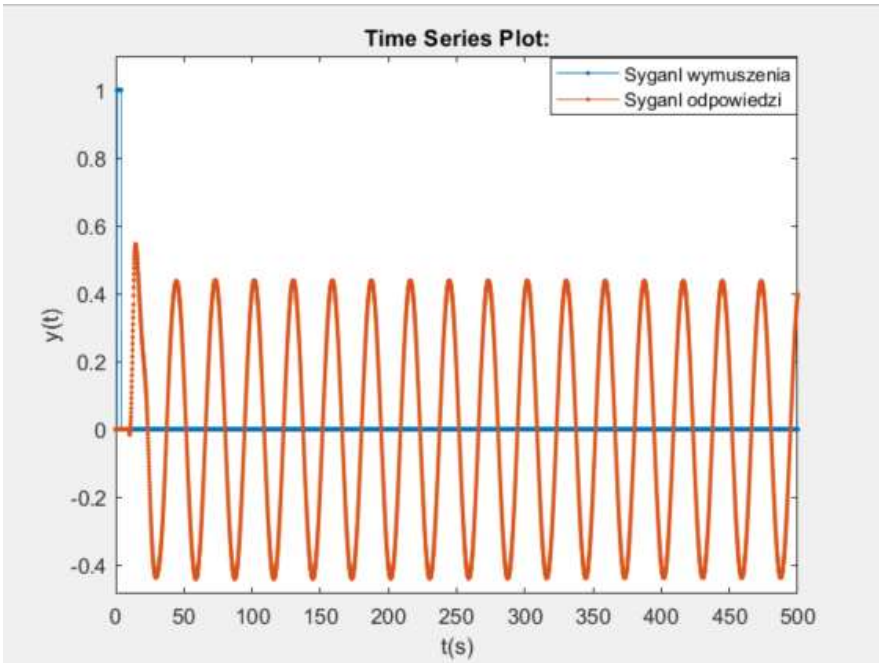
Doboru nastaw dokonano metodą wzbudzenia układu, wykorzystując układ ze schematu 2.



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

Czas opóźnienia obiektu regulacji wpływa na dobór nastaw, dlatego nie wolno go pominąć. Najpierw dobrano czas impulsu na 3s i znaleziono zgrubny przedział w którym mieści się k_{kryt} - (1,2). 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 najlepsza 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:

$$k_{kryt} = 1.406714439392090$$

$$T_{osc} = 28.6s$$

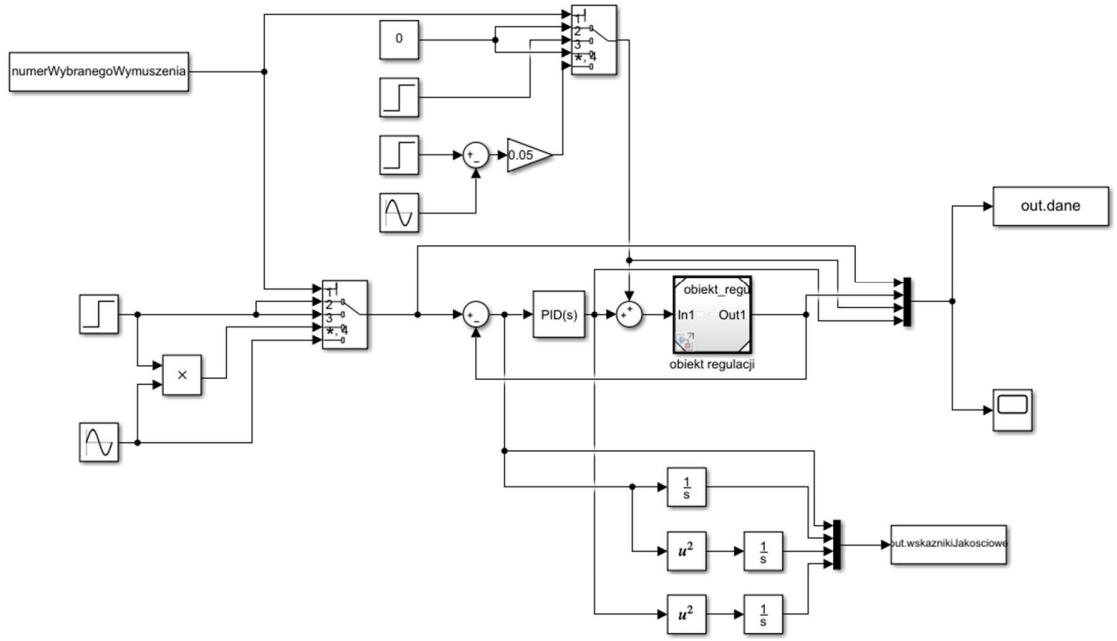
Dla regulatora PID określonego funkcją: $G_R(s) = P + I \cdot \frac{1}{s} + D\frac{N}{1+N\cdot\frac{1}{s}}$

$$P = 0{,}6 \cdot k_{kryt} = 0.844028663635254$$

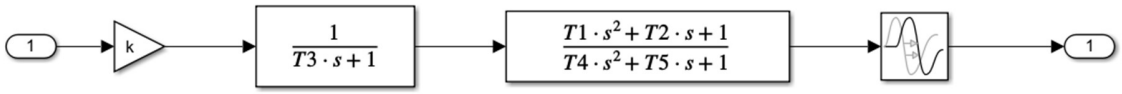
$$I = \frac{1}{0{,}5 \cdot T_{osc}} = 0.069930069930070$$

$$D = \frac{1}{0{,}12 \cdot T_{osc}} = 0.291375291375291$$

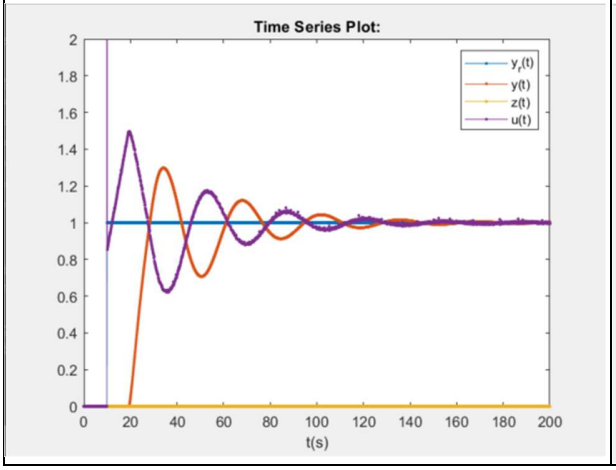
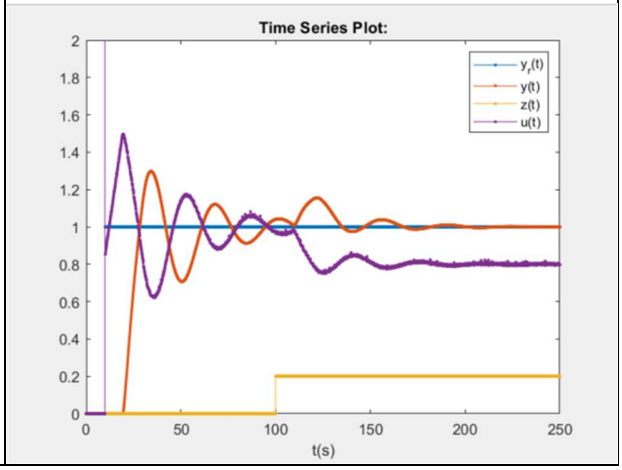
Wyznaczenie odpowiedzi układu z regulatorem na wymuszenia



Schemat 4 Model układu z regulatorem



Schemat 3 Model obiektu regulacji

$y_r(t) = 1(t-10), z(t) = 0$	$y_r(t) = 1(t-10), z(t) = 0.2 \cdot 1(t-100)$
	

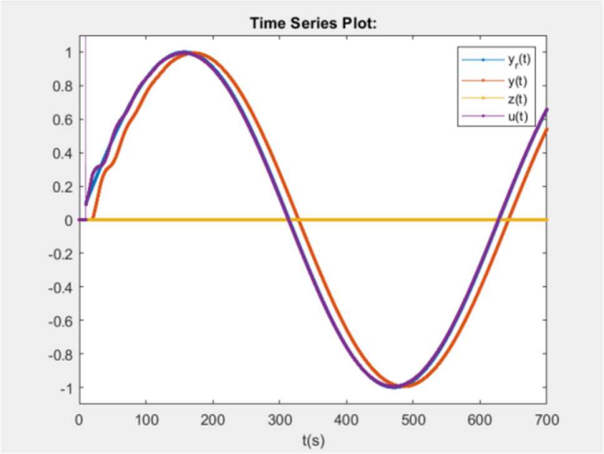
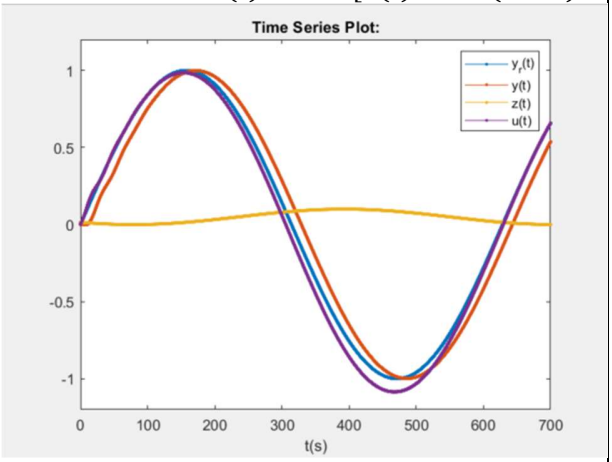
$y_r(t) = sin(0.01t) \cdot 1(t-10), \quad z(t) = 0$	$y_r(t) = sin(0.01t), \quad z(t) = 0.05[1(t) - cos(0.05t)$
	

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 Δ określa się jako $\Delta = 0{,}05e_m$

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

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 – $W = \int_0^T x^2(t)dt$

	$y_r(t) = 1(t-10), z(t) = 0$	$y_r(t) = 1(t-10), z(t) = 0.2 \cdot 1(t-100)$	$y_r(t) = sin(0.01t) \cdot 1(t-10), z(t) = 0$	$y_r(t) = sin(0.01t), z(t) = 0.05[1(t) - cos(0.05t)$
Czas regulacji	17.9	0.1	151.8	0.1
Przeregulowanie	29.0975%	29.0975%	72.8026%	100.0501%
Średni błąd regulacji	0.0028345	0.0045255	0.0011448	0.0011443
Całka kwadratu błędu regulacji	13.9596	14.2528	7.3682	7.7486
Energia sterowania w wyznaczonym czasie regulacji	489.5283 – 200s	195.7837 – 250s	322.6306 – 700s	349.731 – 700s

5 Kod programu

Program 1. Właściwy program

```
clear;
format short;

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 = 700;
numberOfDataSamples = Tsymulacji / Tp + 1;

regulowany = menu('Czy uklad ma zawierac regulator PID?', 'tak', 'nie');
switch (regulowany)
```

```
    case 1
        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]

        odpowiedziLubParametry = menu('Co chcesz zrobić?' ,'Obejrzeć odpowiedzi ukladu', 'Odczytac
wskaźniki jakości regulacji');
        switch (odpowiedziLubParametry)
            case 1
                plot(ans.dane, '.-')
                xlabel('t(s)');
                ylabel(' ');
                legend('y_r(t)','y(t)','z(t)', 'u(t)');
                [down up] = limits(ans.dane);
                ylim([down up]);
            case 2
                tswskazniki = getdatasamples(ans.wskaznikiJakosciowe,1:numberOfDataSamples);
                e = tswskazniki(:,1);
                Ie = tswskazniki(:,2);
                Ie2 = tswskazniki(:,3);
                ICV2 = tswskazniki(:,4);
                % [e Ie Ie2 ICV2]
```

```
                %czas regulacji
                eMaksymalne = max(e);
                eGraniczne = 0.05 * eMaksymalne;
                indeksWprowadzeniaWymuszenia = indeksSkoku(wymuszenie);
                indeksWprowadzeniaZaklocenia = indeksSkoku(zaklocenie);
                momentWprowadzeniegoSygnalu = max(indeksWprowadzeniaWymuszenia,
indeksWprowadzeniaZaklocenia);

                indeksySygnaluUregulowanego =
find(e(momentWprowadzeniegoSygnalu:numberOfDataSamples) < eGraniczne);
                if length(indeksySygnaluUregulowanego)>0
                    indeksSygnaluUregulowanego = indeksySygnaluUregulowanego(1);
                else
                    indeksSygnaluUregulowanego = inf;
                end
                tRegulacji = Tp * indeksSygnaluUregulowanego;

                %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);
```

```
                msgbox({['czas regulacji: ', num2str(tRegulacji)]...
                    ['przeregulowanie: ',num2str(przeregulowanie),' %']...
                    ['sredni blad regulacji: ',num2str(sredniBladRegulacji)]...
                    ['calka kwadratu bledu regulacji: ',num2str(calkaKwadratuBleduRegulacji)]...
                    ['energia sterowania: ',num2str(energiaSterowania)]...
                    },'parametry jakosciowe')
        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);
        % [wymuszenie odpowiedz]

        plot(ans.dane, '.-')
        xlabel('t(s)');
        ylabel(' ');
        legend('u(t)','y(t)');
        [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);
wartoscPoczatkowa = tablica(1);
indeks = 1;
for n=2:dlugosc
    if tablica(n) ~= wartoscPoczatkowa
        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','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});

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

kLower=1;
kUpper=2;

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

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

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

```
    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('Syganl wymuszenia','Syganl 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
```