Programowanie w środowisku Matlab. Projekt nr 2. Temat 220

Piotr Wachowicz, Grupa 37, numer indexu: 289746

12 lutego 2020

# Opis programu

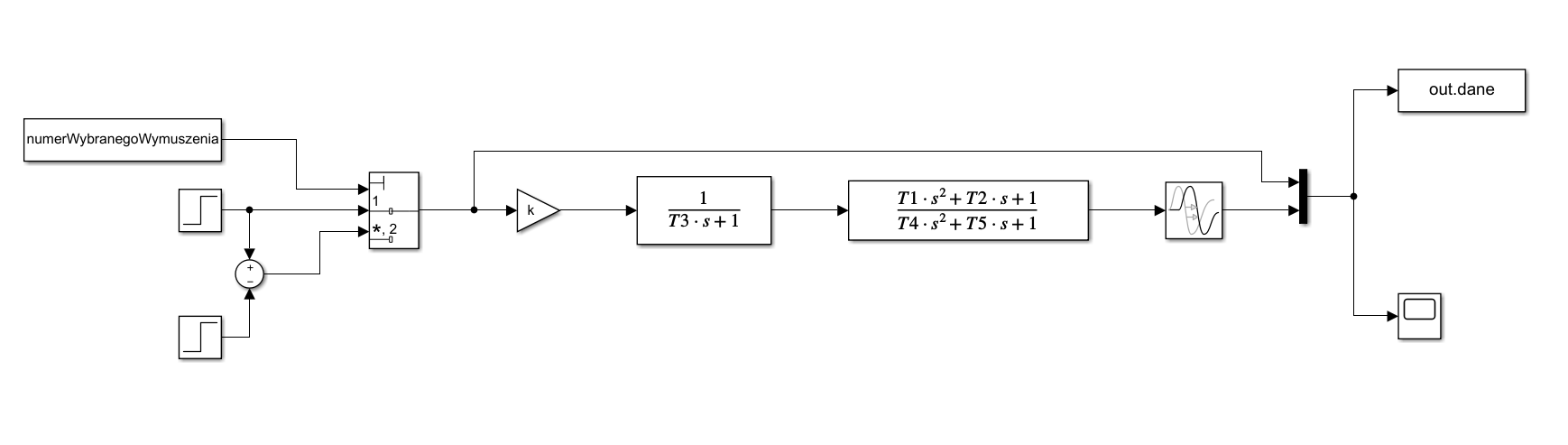
Celem tego projektu jest zamodelowanie w środowisku Matlab wykorzystując program Simulink układu opisanego transmitancją:

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

# Sparametryzowanie transmitancji

W celu ułatwienia obliczeń parametry transmitancji zostały uzmiennione:

gdzie:



# Układ bez regulatora

Schemat 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.

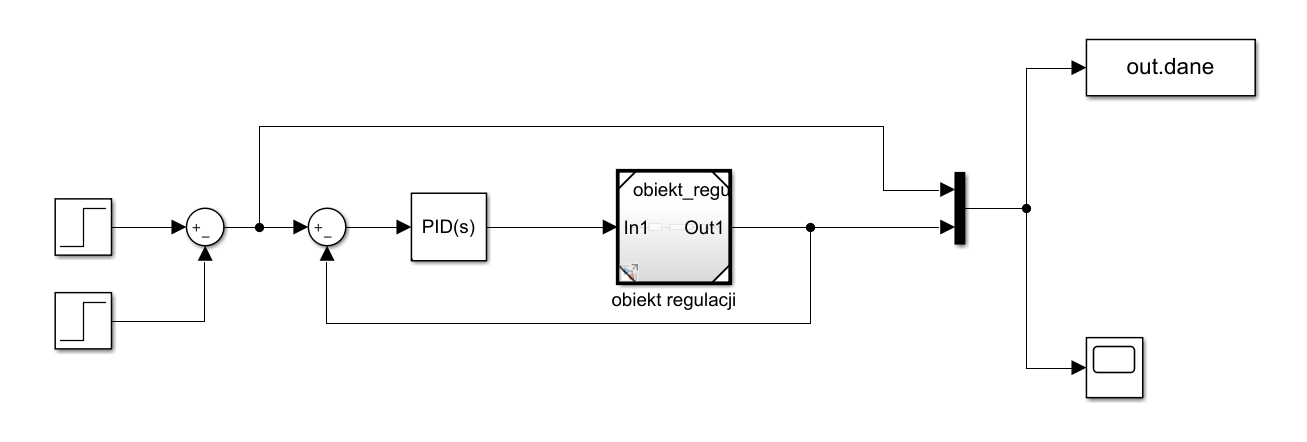
* Parametry transmitancji, w tym wzmocnienie i opóźnienie transportowe układu
* Parametr numerWybranegoWymuszenia odpowiadający za wybór wymuszenia skokowego lub pulsowego

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

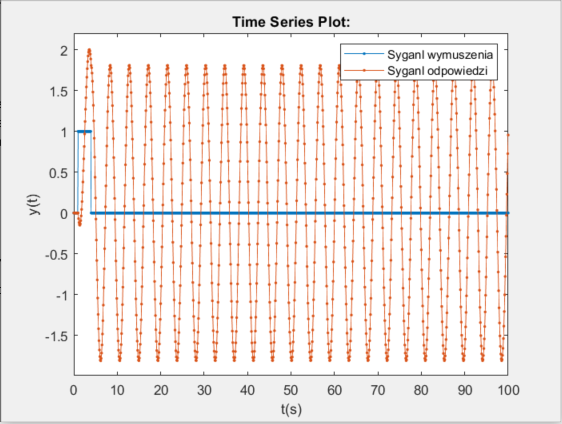
Powyższe charakterystyki obrazują jak obiekt regulacji reaguje na poszczególne wymuszenia. Przedstawiają charakter badanego obiektu.

# Układ z regulatorem PID

## Dobór nastaw regulatora PID metodą Zieglera – Nicholsa

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

Schemat 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 - (12,20). Następnie uruchomiono *Program 2* w którym 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. zostało znalezione jako odległość między kolejnymi ekstremami na osi x.

Wykres Układ doprowadzony do niegasnących oscylacji. Kkryt=12.303901672363281

Uzyskane wartości to:

## Wyznaczenie odpowiedzi układu z regulatorem na wymuszenia

Schemat Model obiektu regulacji

Schemat Model układu z regulatorem

|  |  |
| --- | --- |
| 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) |

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

## Wskaźniki jakości regulacji

– maksymalna odchyłka dynamiczna

- **czas regulacji** – czas określony jako czas od chwili wprowadzenia pobudzenia do chwili, gdy odchyłka regulacji e(t) osiąga wartości mieszczące się w strefie tolerancji . Wartość określa się jako

– **przeregulowanie** określa w procentach stosunek amplitudy drugiego odchylenia do amplitudy pierwszego odchylenia zgodnie ze wzorem

**Średni błąd regulacji -**

**Całka kwadratu błędu regulacji** w wyznaczonym czasie regulacji -

**Energia sterowania** w wyznaczonym czasie regulacji –

|  |  |
| --- | --- |
| u(t)=1(t-10) 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) 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) 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) 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 |

# 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 uklad ma zawierac regulator PID?', 'tak', 'nie');*

## *switch (regulowany)*

## *case 1*

## *odpowiedziLubParametry = menu('Co chcesz zrobić?' ,'Obejrzeć odpowiedzi ukladu', 'Odczytac 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);*

## *% % momentWprowadzeniegoSygnalu = max(indeksWprowadzeniaWymuszenia, indeksWprowadzeniaZaklocenia);*

## *% momentWprowadzeniegoSygnalu = indeksWprowadzeniaWymuszenia*

## *%*

## *% indeksySygnaluUregulowanego = find(e(momentWprowadzeniegoSygnalu:numberOfDataSamples) < eGraniczne)*

## *% indeksSygnaluUregulowanego = indeksySygnaluUregulowanego(1)*

## *% 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)*

## *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);*

## *% wartoscPoczatkowa = tablica(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','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('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*