Warszawa, 6 kwietnia 2010

Laboratorium Podstaw Automatyki – (LPA)

Instrukcja do ćwiczenia nr 1:

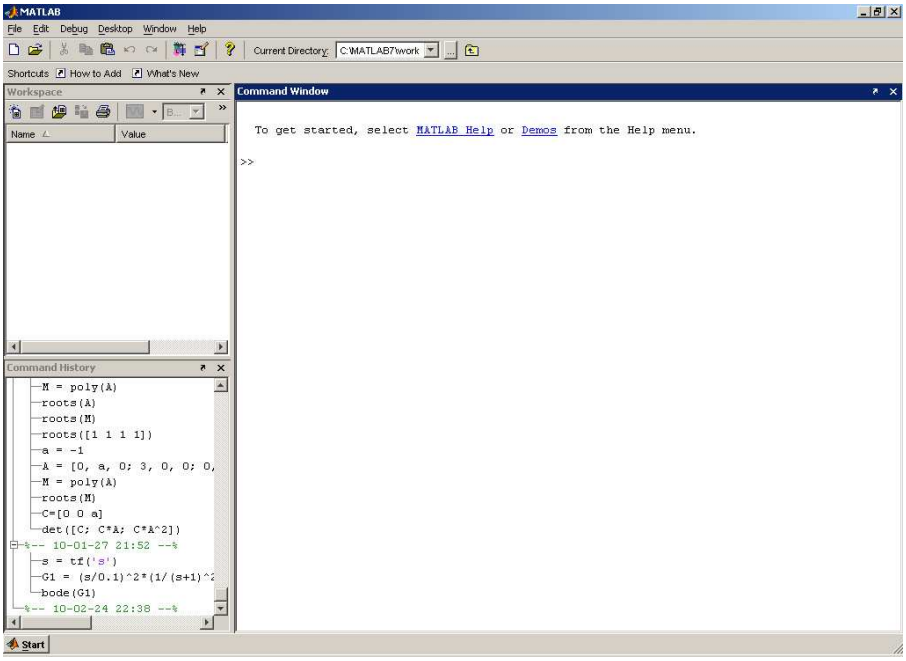
Zastosowanie środowiska Matlab/Simulink do analizy, projektowania i symulacji układów regulacji (na przykładzie regulatora PID)

Autorzy: Dominik Sierociuk i Wiktor Malesza
dominik.sierociuk@ee.pw.edu.pl
Wersja 1.0, (6 kwietnia 2010)
Źródło: http://www.ee.pw.edu.pl/~dsieroci/LTS/LTS_cw5_pid.pdf

1 Elementarne wprowadzenie do środowiska Matlab/Simulink i Control System Toolbox

Matlab (*Matrix Laboratory*) jest interakcyjnym środowiskiem do wykonywania naukowych i inżynierskich obliczeń, symulacji oraz wizualizacji danych.

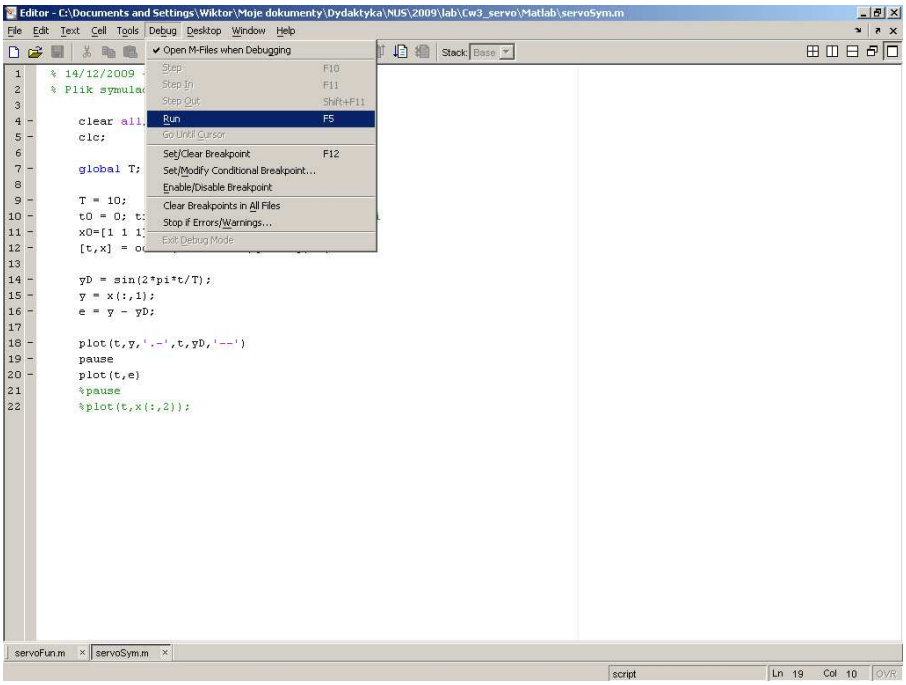
Na Rysunku 1 przedstawione jest typowe okno programu Matlab, zawierające w sobie okienka: **Command Window**, **Workspace** oraz **Command History**. Część **Command Window** służy do wprowadzania komend i odczytywania wyników ich działania. Część **Workspace** przedstawia wszystkie zmienne i obiekty aktualnie znajdujące się w przestrzeni roboczej (pamięci) programu. Obszar **Command History** zawiera historię wcześniej użytych komend (można z niej wywoływać poprzednie komendy, które są także dostępne poprzez przyciski strzałek góra-dół).



Rysunek 1. Widok okna środowiska Matlab

W edytorze możemy edytować, uruchamiać, debugować skrypty i funkcje programu Matlab. Środowisko to jest ściśle powiązane z programem i przestrzenią roboczą Matlaba. Wyróżnia kolorami słowa kluczowe,

komentarze, sprawdza poprawność programu podkreślając błędy składniowe, po wskazaniu zmiennych kursorem myszki pokazuje ich aktualną wartość, itd. W menu **Debug** znajdują się polecenia uruchamiania programu i pracy krokowej, przydatnej przy uruchamianiu programu. W menu **Text** znajdują się opcje przydatne przy pisaniu programów np.: umożliwiające zakomentowanie i odkomentowanie zaznaczonych bloków programu.



Rysunek 2. Widok okna edytora programu Matlab

1.1 Operacje na macierzach

Macierze w Matlabie definiujemy przy użyciu symboli `[]`. Macierz posiada elementy z ciała liczb zespolonych $\mathbb{C}$ (w szczególności z ciała liczb rzeczywistych $\mathbb{R}$). Poszczególne elementy w wierszu oddzielamy przecinkami lub spacjami, wiersze natomiast oddzielamy średnikami. Przykłady zapisu:

- macierzy $A \in \mathbb{R}^{2\times 3}$:

```
>> A = [1,2,3;4,5,6]

A =
     1     2     3
     4     5     6
```

- wektora wierszowego $b \in \mathbb{R}^{1\times 3}$:

```
>> b = [1 2 3]

b =
     1     2     3
```

- macierzy $D \in \mathbb{C}^{2\times 5}$:

```
>> D = [1+2*i i 1 2 3; 0 1-2*i 4 5 6]

D =
   1.0000 + 2.0000i        0 + 1.0000i   1.0000            2.0000            3.0000
        0             1.0000 - 2.0000i   4.0000            5.0000            6.0000
```

Do konkretnych elementów macierzy możemy odwoływać się w następujący sposób:

```
>> D(1,2)

ans =
     0 + 1.0000i
```

Do odwoływania się do większej ilości elementów możemy użyć operatora `:` (dwukropek). Użyty sam, oznacza cały dostępny zakres, użyty w formie `n:m` wskazuje na elementy od n-tego do m-tego, użyty w formie `n:p:m` oznacza elementy z przedziału od n-tego do m-tego co p-ty element, przy czym $n, m, p \in \mathbb{N}$. Przykładowo:

```
>> D(2,:)

ans =
        0             1.0000 - 2.0000i   4.0000             5.0000             6.0000


>> D(2,2:4)

ans =
   1.0000 - 2.0000i   4.0000             5.0000
```

Możemy oczywiście wykonywać operacje na macierzach, np.: dodawać `+`, odejmować `-`, mnożyć `*`, dzielić lewostronnie `\` lub prawostronnie `/`, wyznaczać odwrotność (polecenie `inv()`), itd. Możemy także scalać macierze o odpowiednich wymiarach, np.:

```
>> E = [A; b]

E =
     1     2     3
     4     5     6
     1     2     3
```

1.2 Operacje na wielomianach

Wielomiany w środowisku Matlab reprezentowane są poprzez wektory współczynników tych wielomianów. Przykładowo, wielomiany

$$p(s) = s^2 + 2s + 3 \quad \text{oraz} \quad q(s) = 3s + 4$$

definiowane są poprzez wektory

```
>> p = [1 2 3]

p =
     1     2     3
>> q = [3 4]

q =
     3     4
```

Aby dodać lub odjąć dwa wielomiany wystarczy dodać lub odjąć wektory ich współczynników (o ile są tej samej długości). Polecenie `conv` zwraca wynik mnożenia wielomianów

```
>> r = conv(p,q)

r =
     3    10    17    12
```

otrzymując wielomian

$$r(s) = p(s)q(s) = 3s^3 + 10s^2 + 17s + 12.$$

Polecenie `roots` zwraca zera danego wielomianu. Na przykład, aby wyznaczyć zbiór $\{s \in \mathbb{C}\colon\ p(s) = 0\}$, piszemy

```
>> zer_p = roots(p)

zer_p =
  -1.0000 + 1.4142i
  -1.0000 - 1.4142i
```

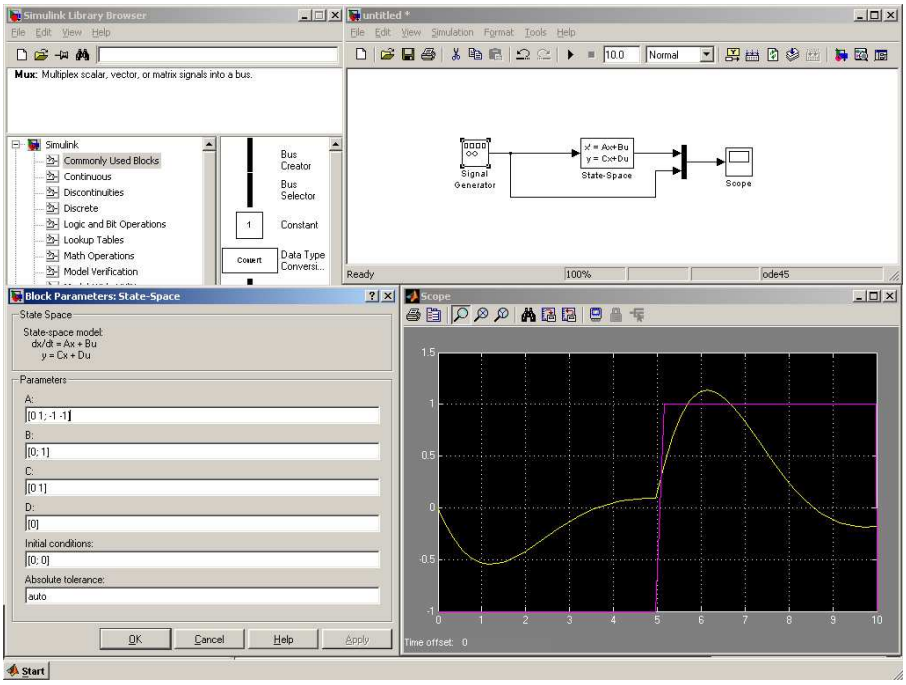
Polecenie `poly` tworzy wektor współczynników wielomianu z podanych jego zer, np.:

```
>> poly(zer_p)

ans =
    1.0000    2.0000    3.0000
```

1.3 Simulink

Simulink jest programem pakietu Matlab pozwalającym na symulację układów dynamicznych. Jego główną zaletą (w porównaniu z samym Matlabem) jest graficzny interfejs, umożliwiający w prosty sposób wprowadzanie i konfigurowanie symulowanego układu.

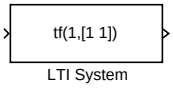


Rysunek 3. Widok okien programu Simulink

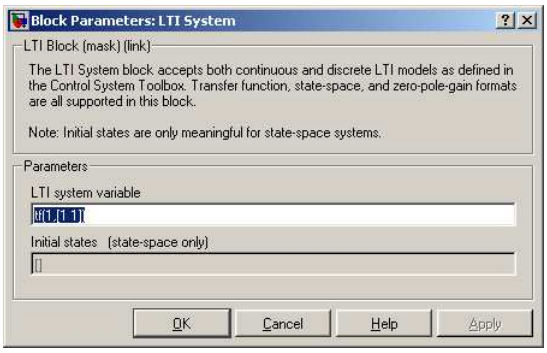
Uruchamiany jest on poleceniem `simulink` lub kliknięciem na ikonkę Simulink w menu programu Matlab (ósma od lewej). W oknie Simulinka **Library Browser** znajdują się bloki podzielone na kategorie tematyczne. Do najważniejszych z nich należą: **Simulink/Sources** (bloki sygnałów wejściowych: generator sygnałów, skok jednostkowy itd.), **Simulink/Sinks** (bloki akwizycji sygnałów: oscyloskop, zgrywanie danych do pliku lub przestrzeni roboczej), **Control System Toolbox** (CST) (blok umożliwiający użycie modeli (obiektów) utworzonych w Matlabie przy pomocy CST do symulacji w Simulinku).

Po utworzeniu nowego modelu możemy przeciągać i łączyć wybrane bloki, a następnie wybierając w menu **Simulation** opcję **Start** uruchomić symulację, której wyniki możemy obserwować na wirtualnych oscyloskopach (**Scope**). W menu **Simulation/Configuration Parameters** możemy dobrać parametry symulacji. Do najważniejszych z nich należą: czas końcowy symulacji (*Stop Time*), minimalny i maksymalny okres próbkowania symulacji (*Min Step Time*, *Max Step Time*), typ sposobu symulacji ze zmiennym lub stałym krokiem (*Variable-step*, *Fixed-Step*) oraz rodzaj użytego algorytmu numerycznego rozwiązywania równań różniczkowych (*Solver*).

- **LTI System** – blok umożliwiający użycie modeli (obiektów) utworzonych w Matlabie przy pomocy CST



(a). Ikona

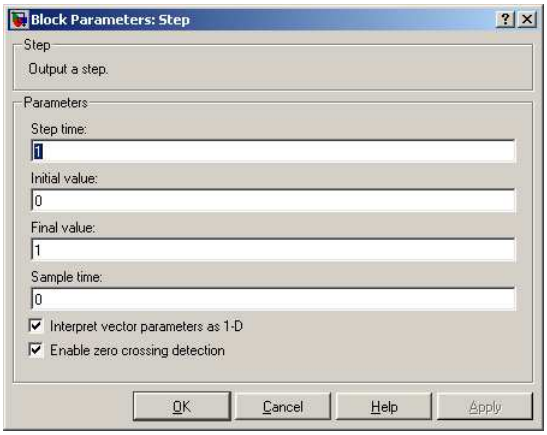


(b). Okno dialogowe

Rysunek 4. Blok **LTI System**

- **Step** – blok generujący skok jednostkowy (parametr *Step time* oznacza czas rozpoczęcia skoku (można przyjąć 0))

(a). Ikona

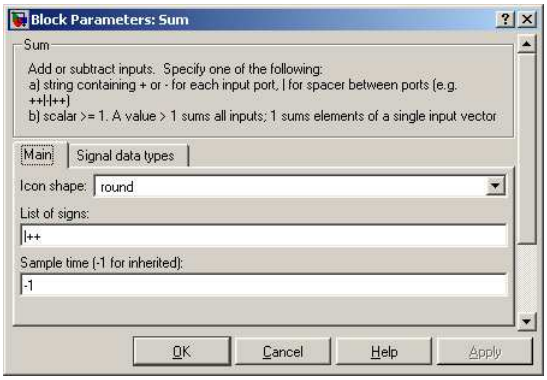


(b). Okno dialogowe

Rysunek 5. Blok **Step**

- **Sum** – blok sumowania sygnałów (parametr *List of signs* oznacza listę znaków $+, -$ odpowiadających im sygnałów, przy czym liczba tych znaków określa liczbę wejść sumatora)

(a). Ikona

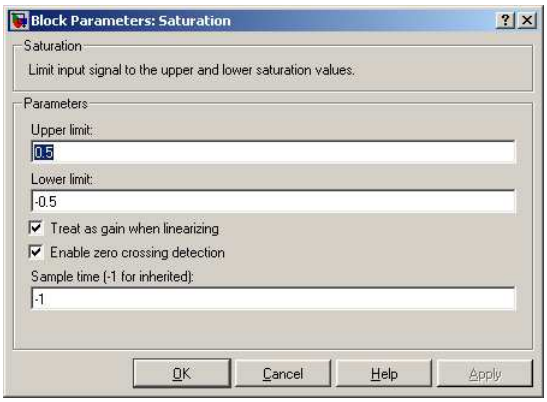


(b). Okno dialogowe

Rysunek 6. Blok **Sum**

- **Saturation** – blok nasycenia (*Upper limit* określa górną granicę sygnału, a *Lower limit* – dolną granicę sygnału)

(a). Ikona



(b). Okno dialogowe

Rysunek 7. Blok **Saturation**

2 Zapis transmitancyjny, kreślenie charakterystyk czasowych i częstotliwościowych

2.1 Model transmitancyjny (ciągły)

Transmitancję układu MIMO (w szczególności SISO) zapisujemy przy użyciu polecenia `tf`:

```
>> G = tf(L,M)
```

gdzie `L` oraz `M` są $p \times m$-wymiarowymi tablicami wektorów wierszowych współczynników wielomianów odpowiednio liczników oraz mianowników macierzy transmitancji, przy czym p jest liczbą wyjść, a m – liczbą wejść układu.

Przykładowo, układ MIMO o dwóch wejściach i jednym wyjściu opisany macierzą transmitancji postaci

$$G(s) = \begin{pmatrix} \frac{2s+4}{s^2+5s-3} & \frac{1}{s+1} \end{pmatrix} \tag{1}$$

definiujemy następująco:

```
>> G = tf({[2 4],[1]},{[1 5 -3],[[1 1]]})

Transfer function from input 1 to output:
   2 s + 4
-------------
s^2 + 5 s - 3

Transfer function from input 2 to output:
  1
-----
s + 1
```

Istnieje alternatywny sposób definiowania transmitacji, który zobrazowany jest na przykładzie dla transmitancji (1):

```
>> s = tf('s');
>> G = [(2*s + 4)/(s^2+5*s-4), 1/(s+1)]

Transfer function from input 1 to output:
   2 s + 4
-------------
s^2 + 5 s - 4

Transfer function from input 2 to output:
```

```
  1
-----
s + 1
```

W celu zdefiniowania transmitancji członu opóźniającego $G_o(s) = e^{-s\tau}$, gdzie τ jest czasem opóźnienia, możemy, przykładowo dla $\tau = 0.5$ sek, napisać:

```
>> Go = tf([1],[1])

Transfer function:
1

>> tau = 0.5;
>> Go.OutputDelay = tau

Transfer function:
exp(-0.5*s) * (1)
```

2.2 Model w przestrzeni stanu (ciągły)

Układ sterowania opisany w dziedzinie czasu:

$$\dot{x} = Ax + Bu,$$
$$y = Cx + Du,$$

przy czym $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ oraz $D \in \mathbb{R}^{p\times m}$, definiujemy w Matlabie następująco:

```
sys = ss(A,B,C,D)
```

Przykładowo, dla układu opisanego macierzami

$$A = \begin{pmatrix} 0 & 1 \\ -1 & -2 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 1 \end{pmatrix} \quad \text{oraz} \quad D = 0,$$

mamy:

```
>> A = [0 1; -1 -2]; B = [0; 1]; C = [1 1]; D = [0];
>> sys = ss(A,B,C,D)

a =
       x1  x2
   x1   0   1
   x2  -1  -2

b =
       u1
   x1   0
   x2   1

c =
       x1  x2
   y1   1   1

d =
       u1
   y1   0

Continuous-time model.
```

Uwaga1: Przekształcenie modelu w przestrzeni stanu do postaci transmitancyjnej można przeprowadzić przy użyciu funkcji `ss2tf` w następujący sposób: `[L,M] = ss2tf(A,B,C,D)`.
Uwaga2: W celu otrzymania układu zamkniętego pętlą sprzężenia zwrotnego można użyć funkcji

`sys = feedback(sys1,sys2)`, gdzie `sys1` jest układem w torze głównym, a `sys2` to układ w pętli sprzężenia zwrotnego. Domyślnie ustawione jest ujemne sprzężenie zwrotne.

2.3 Logarytmiczna charakterystyka częstotliwościowa amplitudy i fazy (Bode'go)

Do narysowania charakterystyki Bode'go służy funkcja `bode(sys)`, gdzie `sys` jest obiektem utworzonym na przykład przy pomocy funkcji `tf` lub `ss`. Przykład użycia funkcji `bode`:

```
>> G=tf([2],[1 2 5 2 0]);
>> bode(G)
>> grid
```

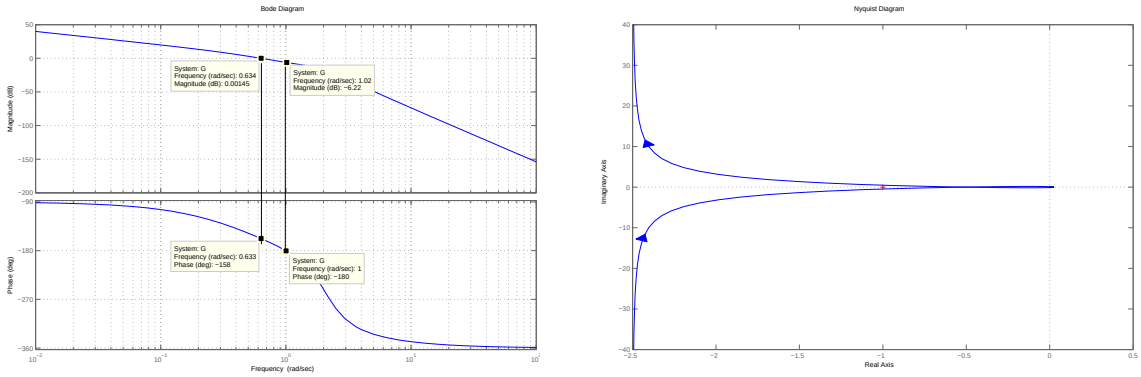
Instrukcja `grid` powoduje umieszczenie siatki logarytmicznej na aktywnym wykresie, patrz Rysunek 8a. Korzystając z narzędzi **Tools/Data Cursor** oraz polecenia **Insert/Line** można ustalić zapas fazy i wzmocnienia danego układu. Należy przy tym pamiętać, że zapas wzmocnienia dany na wykresie, wyrażony jest w skali logarytmicznej (dB), czyli $20\log(A)$, gdzie A jest wartością liczbową wzmocnienia.

2.4 Charakterystyka częstotliwościowa amplitudowo-fazowa (Nyquista)

Do narysowania charakterystyki Nyquista służy funkcja `nyquist(sys)`. Przykładowo

```
>> G=tf([2],[1 2 5 2 0]);
>> nyquist(G)
```

Charakterystyka na Rysunku 8b jest narysowana zarówno dla zakresu pulsacji od nieskończoności do zera jak i od minus nieskończoności do zera. Kierunek zmian pulsacji pokazują strzałki, natomiast czerwony krzyżyk oznacza punkt Nyquista $(-1, 0)$, pomocny przy badaniu stabilności układu zamkniętego.



(a). Charakterystyka Bode'ego transmitancji $G(s)$ **(b).** Charakterystyka Nyquista transmitancji $G(s)$

Rysunek 8. Charakterystyki częstotliwościowe transmitancji $G(s)$

2.5 Charakterystyki czasowe (odpowiedź skokowa i impulsowa)

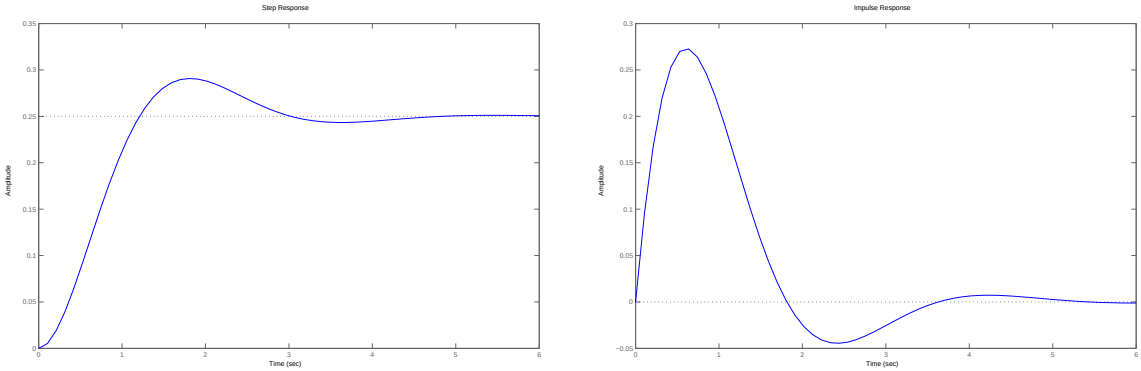
Do wyznaczenia odpowiedzi skokowej układu (odpowiedź na skok jednostkowy) służy funkcja `step(sys)` lub `step(sys,T)`, gdzie `T` jest czasem końcowym symulacji. Przykładowo

```
>> F=tf([1],[1 2 4]);
>> step(F)
```

Do wyznaczenia odpowiedzi impulsowej (na impuls Dirac'a) służy funkcja `impulse(sys)`. Na przykład

```
>> impulse(F)
```

Wyniki działania dwóch powyższych poleceń zobrazowane są na Rysunku 9.



(a). Odpowiedź skokowa transmitancji $F(s)$ **(b).** Odpowiedź impulsowa transmitancji $F(s)$

Rysunek 9. Charakterystyki czasowe transmitancji $F(s)$

3 Regulator PID

Regulator PID w ogólności składa się z trzech części: proporcjonalnej (P), całkującej (I) oraz różniczkującej (D). Nie wszystkie te składowe muszą występować jednocześnie, co powoduje, że mamy do dyspozycji wiele wariantów tego regulatora np.: P (człon proporcjonalny), PI (proporcjonalno-całkujący), PD (proporcjonalno-różniczkujący), PID (proporcjonalno-całkująco-różniczkujący).

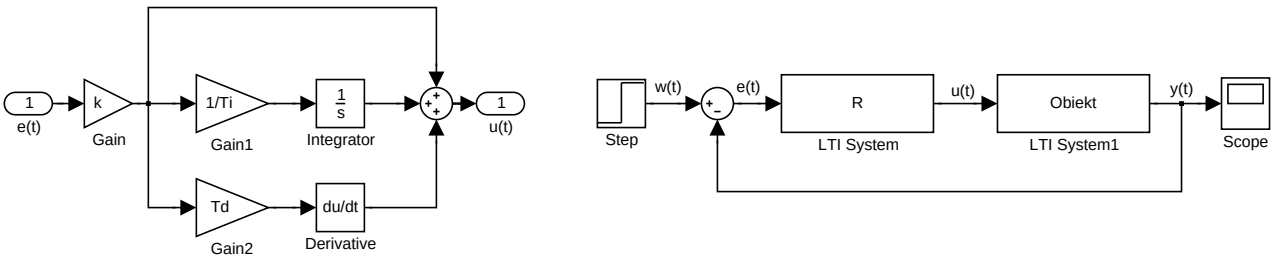
Podstawowe transmitancje tych regulatorów są następujące:

- regulator P
$$R(s) = K_R,$$
- regulator PI
$$R(s) = K_R \left(1 + \frac{1}{T_\mathrm{i} s}\right),$$
- regulator PD
$$R(s) = K_R(1 + T_\mathrm{d} s),$$
- regulator PID
$$R(s) = K_R \left(1 + \frac{1}{T_\mathrm{i} s} + T_\mathrm{d} s\right).$$

W dziedzinie czasu, prawo sterowania regulatora PID opisane jest następująco

$$u(t) = K_R \left(e(t) + T_\mathrm{d} \frac{\mathrm{d}e(t)}{\mathrm{d}t} + \frac{1}{T_\mathrm{i}} \int_0^t e(\tau)\,\mathrm{d}\,\tau\right),$$

przy czym $e(t) = w(t) - y(t)$ jest uchybem układu, patrz Rysunek 10b. Realizacja układu sterownia z wykorzystaniem regulatora PID przedstawiona jest na Rysunku 10.



(a). Struktura regulatora PID **(b).** Schemat układu regulacji z regulatorem PID

Rysunek 10. Układ sterowania z regulatorem PID w Simulinku

3.1 Dobór nastaw regulatora PID

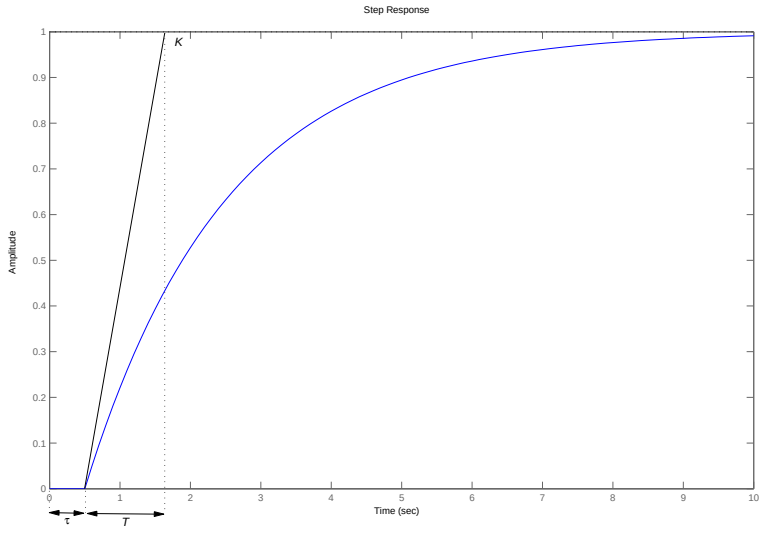
Istnieje wiele sposobów wyznaczania nastaw regulatora PID, [3]. Wyniki otrzymywane przez różne metody mogą się różnić od siebie i w większości przypadków są jedynie przybliżonymi wartościami, które warto następnie poddać ręcznemu dostrojeniu końcowemu, w zależności od wymagań i potrzeb sterowania. Jednym z najczęściej stosowanych algorytmów są nastawy Zieglera-Nicholsa [4], [5]. Do niewątpliwych zalet tego algorytmu należy prostota jego stosowania, do wad natomiast to, iż optymalność tych nastaw odnosi się tylko dla układu inercyjnego z opóźnieniem. Nastawy te zostały opracowane w sposób eksperymentalny dla układu

$$G(s) = \frac{Ke^{-\tau s}}{Ts+1}, \tag{2}$$

przy czym $e^{-\tau s}$ jest członem opóźniającym. Skutkuje to tym, że dla układów o dynamice (transmitancji) różniącej się od (2), uzyskane rezultaty będą tylko suboptymalne, jednakże w dużej mierze na tyle dobre, aby móc je zastosować w praktyce. Optymalność tych nastaw została określona jako zanikanie odpowiedzi przejściowej (tłumienie oscylacji) do 25% na okres. Istnieją dwa sposoby wyznaczania nastaw Zieglera-Nicholsa: z odpowiedzi skokowej i ze wzmocnienia krytycznego.

Nastawy Zieglera-Nicholsa wyznaczane z odpowiedzi skokowej układu

Metoda ta polega na odczytaniu z odpowiedzi skokowej układu parametrów K, τ oraz T. Dla modelu inercyjnego z opóźnieniem, danego równaniem (2), sposób ich wyznaczenia pokazuje Rysunek 11.



Rysunek 11. Wyznaczanie parametrów odpowiedzi skokowej układu inercyjnego z opóźnieniem (2)

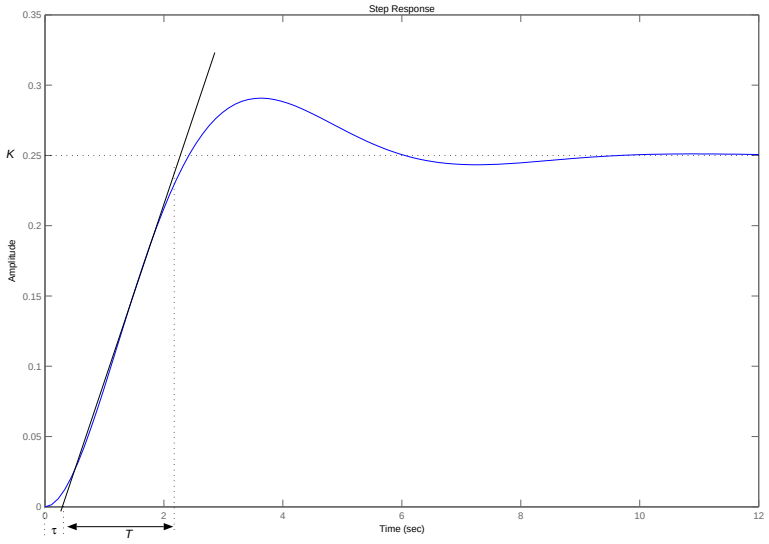
Parametr K oznacza wartość ustaloną odpowiedzi, τ jest czasem opóźnienia, a T – czasem narastania. Następnie na podstawie tych parametrów wyznaczane są nastawy regulatorów z zależności podanych w Tabela 1.

Tabela 1. Nastawy Zieglera-Nicholsa z odpowiedzi skokowej, gdzie $a = \frac{K\tau}{T}$

Parametry	P	PI	PID
K_R	$1/a$	$0.9/a$	$1.2/a$
T_i	-	3τ	2τ
T_d	-	-	0.5τ

W rzeczywistości jednak, większość układów posiada inną dynamikę niż daną równaniem (2). Aby w takim wypadku zastosować nastawy Zieglera-Nicholsa musimy założyć, że dany układ przybliżamy układem inercyjnym z opóźnieniem. Oczywiście im dynamika danego układu jest bliższa inercyjnego z opóźnieniem, tym bardziej

zbliżone do optymalnych nastawy otrzymujemy. Rysunek 12 przedstawia wyznaczenie parametrów odpowiedzi dla układu z przeregulowaniem (czyli z dynamiką różniącą się od inercyjnej z opóźnieniem).



Rysunek 12. Wyznaczanie parametrów odpowiedzi skokowej układu z przeregulowaniem

Nastawy Zieglera-Nicholsa wyznaczane ze wzmocnienia krytycznego

W metodzie tej konieczne jest wyznaczenie wzmocnienia krytycznego $K_R = K_{\mathrm{kr}}$ (członu proporcjonalnego P), czyli takiego, które doprowadzi układ do granicy stabilności. Można je odczytać z charakterystyki Bode'go lub wyznaczyć eksperymentalnie zwiększając wzmocnienie K_R (regulatora P bez członu I oraz D) zamkniętego układu sterowania, do momentu doprowadzenia układu do granicy stabilności (niegasnące oscylacje – układ zachowuje się jak generator). W momencie gdy układ będzie na granicy stabilności możemy odczytać okres oscylacji krytycznych T_{osc} (w sekundach). Na podstawie tych danych możemy obliczyć nastawy regulatorów zgodnie z Tabelą 2.

Tabela 2. Nastawy Zieglera-Nicholsa na podstawie wzmocnienia krytycznego

Parametry	P	PI	PID
K_R	$0.5K_{\mathrm{kr}}$	$0.45K_{\mathrm{kr}}$	$0.6K_{\mathrm{kr}}$
T_{i}	-	$T_{\mathrm{osc}}/1.2$	$T_{\mathrm{osc}}/2$
T_{d}	-	-	$T_{\mathrm{osc}}/8$

4 Przebieg ćwiczenia

Zespół otrzymuje od prowadzącego opis układu dynamicznego (w formie równań, transmitancji itd.). Dla danego modelu przeprowadza:

1. Badanie charakterystyk danego układu, zarówno częstotliwościowych jak i czasowych. Na podstawie charakterystyki częstotliwościowej wyznaczyć przybliżoną wartość wzmocnienia krytycznego w układzie zamkniętym.

2. Na podstawie odpowiedzi skokowej lub impulsowej doprowadzić układ zamknięty do granicy stabilności, poprzez odpowiednie dobieranie wzmocnienia regulatora. Ustalić na tej podstawie dokładną wartość wzmocnienia krytycznego K_{kr} i okresu oscylacji krytycznych T_{osc}.

3. Dobrać nastawy regulatorów P, PI oraz PID według nastaw Zieglera-Nicholsa (ze wzmocnienia krytycznego).

4. Przeprowadzić badanie charakterystyk częstotliwościowych i odpowiedzi skokowych otrzymanych regulatorów.

5. Przeprowadzić badanie charakterystyk częstotliwościowych i czasowych układu z otrzymanymi regulatorami. Dla regulatora PID przeprowadzić „ręczne" strojenie regulatora.

6. Dobrać nastawy regulatorów P, PI oraz PID według nastaw Zieglera-Nicholsa (z odpowiedzi skokowej).

7. Porównać otrzymane nastawy z nastawami otrzymanymi „ze wzmocnienia krytycznego" oraz porównać odpowiedzi skokowe układu z tymi regulatorami.

8. Przeprowadzić w Simulinku dokładną analizę działania wybranego regulatora PID w układzie sterowania, tzn. zaobserwować wartości sygnałów poszczególnych członów (proporcjonalnego, całkującego, różniczkującego). Następnie wprowadzić ograniczenie sterowania poprzez użycie bloku **Saturation** i zaobserwować działanie regulatora w wypadku gdy sterowanie jest ograniczane przez ten blok.

5 Przykładowe pytania kontrolne

1. Podaj transmitancję regulatora PID.

2. Narysuj schemat struktury regulatora PID.

3. Podaj pełną komendę Matlaba umożliwiającą zapis następującej transmitancji:

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

4. Dla jakiego układu nastawy Zieglera-Nicholsa zostały zaprojektowane?

5. Wyjaśnić, dlaczego strojenie regulatora metodą „wzmocnienia krytycznego" stosuje się tylko do układów, których rząd jest większy od 2, tzn. $\deg M(s) > 2$, gdzie $M(s)$ jest mianownikiem transmitancji (właściwej bądź ściśle właściwej) obiektu sterownia?

6. Jakie niebezpieczeństwo z punktu widzenia zastosowań w praktyce, niesie za sobą strojenie regulatora metodą „wzmocnienia krytycznego"?

7. Co to jest wzmocnienie krytyczne układu?

8. Podaj pełną komendę Matlaba do zapisania transmitancji regulatora PI.

9. Podaj wartość wzmocnienia regulatora P odpowiadającego wzmocnieniu logarytmicznemu $n \cdot 20\,\mathrm{dB}$, gdzie $n \in \mathbb{Z}$.

Bibliografia

[1] J.D. Powell G.F. Franklin and A. Emami-Naeini. *Feedback Control of Dynamic Systems, 3rd ed.* Addison-Wesley, 1994.

[2] H. Vogt J. Mazurek and W. Zydanowicz. *Podstawy automatyki.* Oficyna Wyd. Politechniki Warszawskiej, 2006.

[3] M. Tomera. *Instrukcja laboratorium Teorii Sterowania: Badanie układu sterowania z regulatorem PID.* Akademia Morska w Gdyni Katedra Automatyki Okretowej, 2009.

[4] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, (64):759–768, 1942.

[5] J. G. Ziegler and N. B. Nichols. Process lags in automatic control circuits. *Trans. ASME*, (65):433–444, 1943.