

Wprowadzenie do SIMULINKA

Mirosław Tomera

1. WSTĘP

SIMULINK jest pakietem oprogramowania służącym do modelowania, symulacji i analizowania układów dynamicznych. Można implementować w nim zarówno układy liniowe jak i nieliniowe modelowane w czasie ciągłym, dyskretnym lub w hybrydowym w którym część układu pracuje w czasie dyskretnym natomiast inna część w czasie ciągłym.

SIMULINK wyposażony jest w interfejs graficzny pozwalający w łatwy sposób budować modele w postaci schematów blokowych przy użyciu myszki komputerowej. Przy użyciu tego interfejsu można wykresować modele w podobny sposób jak to odbywa się przy użyciu kartki i ołówka. SIMULINK zawiera biblioteki źródeł i rejestratorów sygnału, elementów liniowych i nieliniowych oraz połączeń między nimi. Można również tworzyć swoje własne bloki przy użyciu S-Funkcji. Modele mają strukturę hierarchiczną, można przeglądać układ z najwyższego poziomu, klikając dwukrotnie myszką na bloki przechodząc się na niższe poziomy zawierające więcej szczegółów. Taka organizacja struktury modelu pozwala na łatwe zorientowanie się w pracy systemu i w jaki sposób poszczególne części oddziałują na siebie.

Po zdefiniowaniu modelu można przeprowadzać symulacje z użyciem wybranej metody całkowania z poziomu menu SIMULINKA lub po wpisaniu odpowiednich poleceń w oknie komend MATLABA. Menu SIMULINKA jest szczególnie dogodne do pracy interaktywnej, podczas gdy linia komend MATLABA przy uruchamianiu wielokrotnych symulacji po zmianie pewnych parametrów. Wykorzystując oscyloskop lub inne bloki wyświetlaczowe można obserwować wyniki symulacji podczas uruchamiania modelu Simulinka. Wyniki symulacji mogą być umieszczane w przestrzeni roboczej MATLABA (workspace) pozwalając w ten sposób na dalszą ich obróbkę [1].

2. BUDOWANIE PROSTEGO MODELU

Przykład ten ilustruje w sposób skrótowy w jaki sposób buduje się model w SIMULINKU. W modelu tym odbywa się wyznaczanie odpowiedzi skokowej na podstawie transmitancji i wyświetlanie zarówno wymuszenia jak i wyników symulacji.

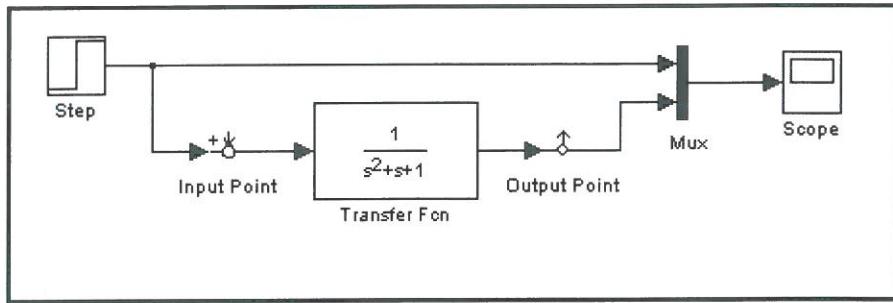
Przykład 1

Wyznaczyć odpowiedź skokową układu o transmitancji

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (1.1)$$

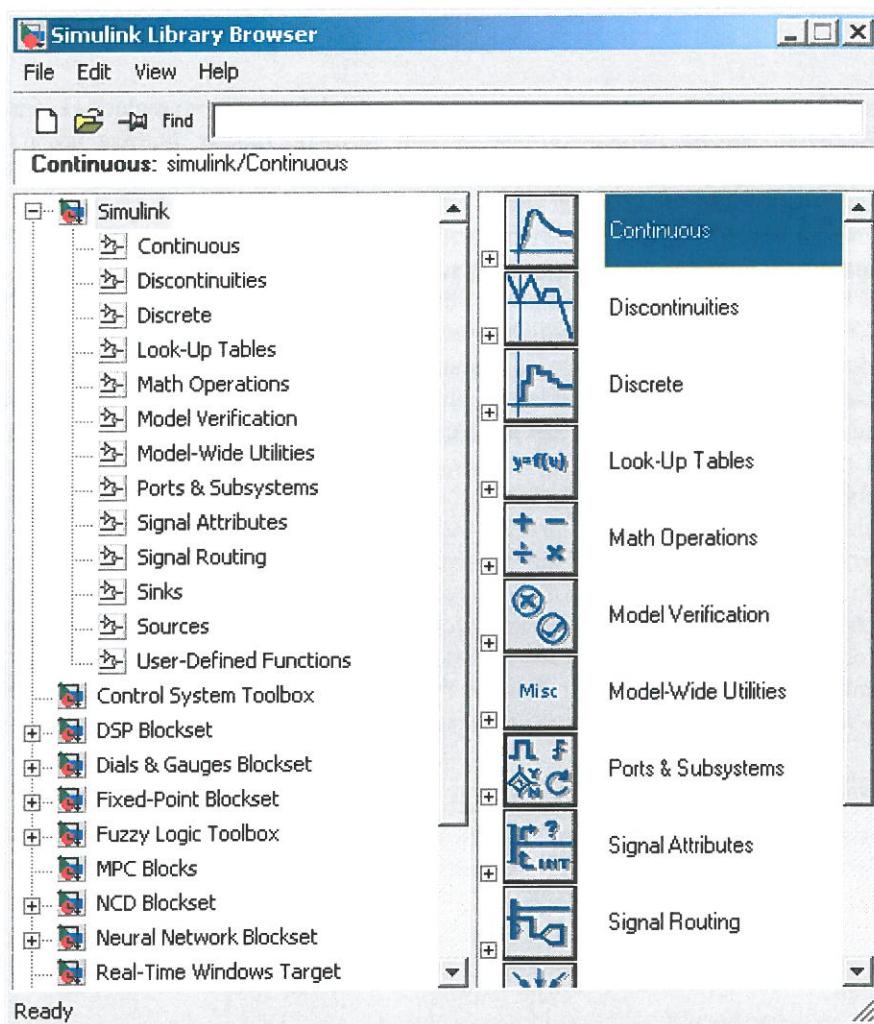
dla $\omega_n = 1$ oraz $\zeta = 0.5$. Wyniki symulacji wyświetlić na oscyloskopie

Rozwiązanie. Zadanie to zrealizowane zostanie w następującym układzie.



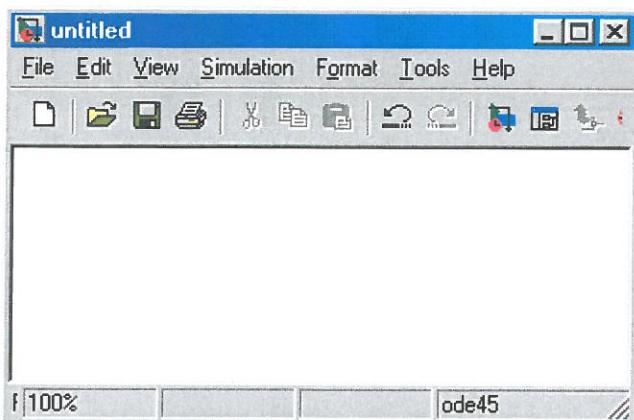
Rys. 1.1. Kompletny model badanego układu

Aby utworzyć model, najpierw należy wpisać polecenie `simulink` w oknie komend MATLABA, pojawi się wówczas okno biblioteki SIMULINKA.



Rys. 1.2. Okno bibliotki SIMULINKA

Aby utworzyć nowy model, należy wybrać opcję `File/New/Model` lub odpowiednią ikonę w oknie biblioteki SIMULINKA i wówczas zostanie nowe okienko do tworzenia modelu

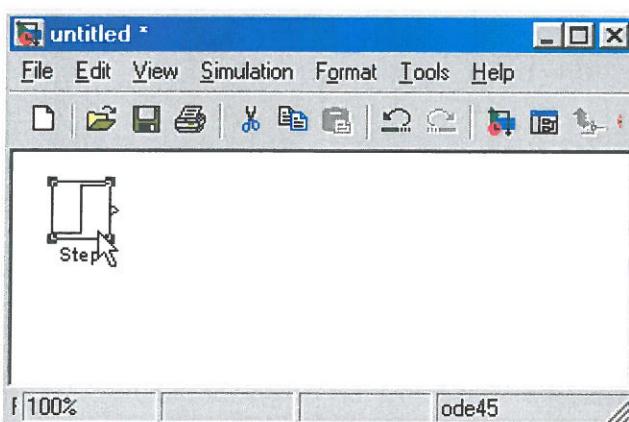


Rys. 1.3. Okno do tworzenia nowego modelu

Aby utworzyć pożądany w tym przykładzie model należy skopiować odpowiednie bloki z następujących bibliotek SIMULINKA:

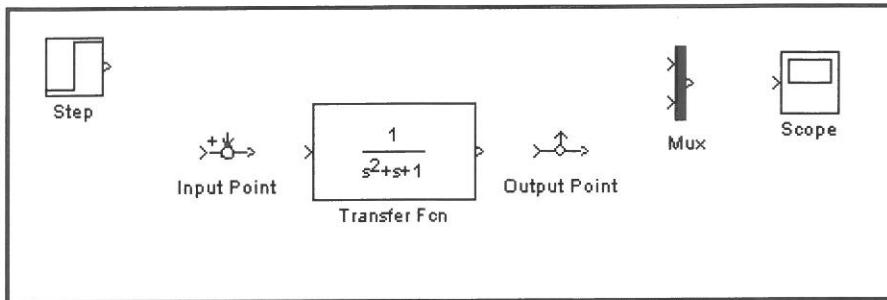
- Sources (blok Step) – wymuszenie skokowe
- Sinks (blok Scope) – oscyloskop
- Continuous (blok Transfer Fcn) – transmitancja
- Control System Toolbox (ikony: Input Point, Output Point) – złącza wejścia i wyjścia dla programu LtiView
- Signals Routing (blok Mux) – multiplekser

Aby skopiować z biblioteki blok Step, najpierw należy rozwinąć drzewo biblioteki Sources w celu wyświetlenia jej zawartości przez kliknięcie na węzeł Sources, następnie należy kliknąć na węzeł Step w celu wybrania go. Teraz przeciągając blok Step z biblioteki do okna modelu SIMULINK tworzy kopię tego bloku w punkcie do którego został przesunięty.



Rys. 1.4. Skopiowanie bloku Step do okna modelu

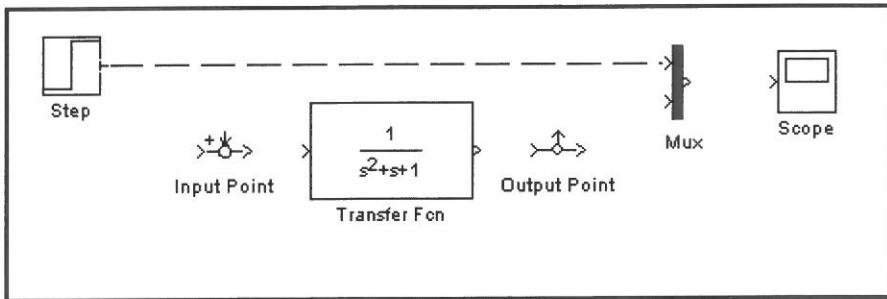
W podobny sposób dokonuje się skopiowania do okna modelu pozostałych potrzebnych elementów schematu z odpowiednich bibliotek SIMULINKA. Zmiany położenia bloku z jednego miejsca w inne dokonuje się poprzez naciśnięcie myszki na blok i jego przesunięcie. Po skopiowaniu wszystkich bloków model układu wygląda następująco:



Rys. 1.5. Skopiowane wszystkie bloki do realizacji schematu

Przeglądając ikony bloków widać, że blok Step ma grot strzałki z prawej strony, natomiast blok Mux ma dwa z lewej i jeden z prawej. Symbol $>$ umieszczony na wejściu bloku oznacza port wejścia (wejście) do bloku, natomiast jeśli symbol ten znajduje się na wyjściu wówczas oznacza port wyjścia (wyjście) z bloku. Jeśli sygnał przekazywany jest z wyjścia jednego bloku na wejście innego bloku wówczas odbywa się to poprzez linię łączącą. Kiedy bloki zostaną ze sobą połączone, wówczas symbole portów znikną.

Teraz jest czas aby połączyć bloki ze sobą. Połącz blok Step z górnym portem bloku Mux. Jeśli umieszczony zostanie wskaźnik myszki na porcie wyjściowym z prawej strony bloku Step, wówczas znak kurSORA zmieni swój kształt na krzyż. Naciśnij wówczas przycisk myszy i przesuń kurSORA do górnego portu wejściowego bloku Mux. Zauważ, że gdy wcisnięty jest przycisk myszy, wówczas linia łącząca bloki jest kreskowana, natomiast gdy wskaźnik osiąga blok Mux wówczas zmienia się kształt kurSORA na podwójny krzyżyk (rys. 1.6).



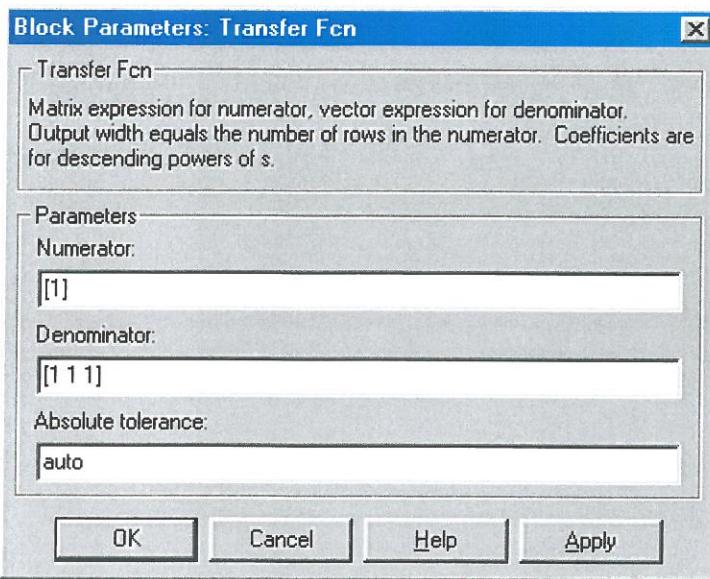
Rys. 1.6. Łączenie dwóch bloków linią

Teraz po zwolnieniu przycisku myszy bloki zostaną połączone. Z rysunku 1 widać, że są linie które łączą ten sam port wyjściowy z dwoma różnymi portami wejściowymi. Takie linie łączą port wyjściowy bloku Step z portem wejściowym bloku Mux i blokiem Transfer Fcn.

Chcąc założyć węzeł na linii łączącej blok Step i blok Mux i poprowadzić linię do portu wejściowego bloku Transfer Fcn należy:

- umieścić wskaźnik kurSORA na linii pomiędzy blokami Step i Mux;
- wcisnąć klawisz **Ctrl** (lub kliknąć na prawy przycisk myszy);
- trzymając wcisnięty przycisk myszy przesunąć kurSORA do portu wejściowego bloku Transfer Fcn;
- zwolnij przycisk myszy i SIMULINK narysuje linię pomiędzy punktem początkowym i portem wejściowym bloku Transfer Fcn.

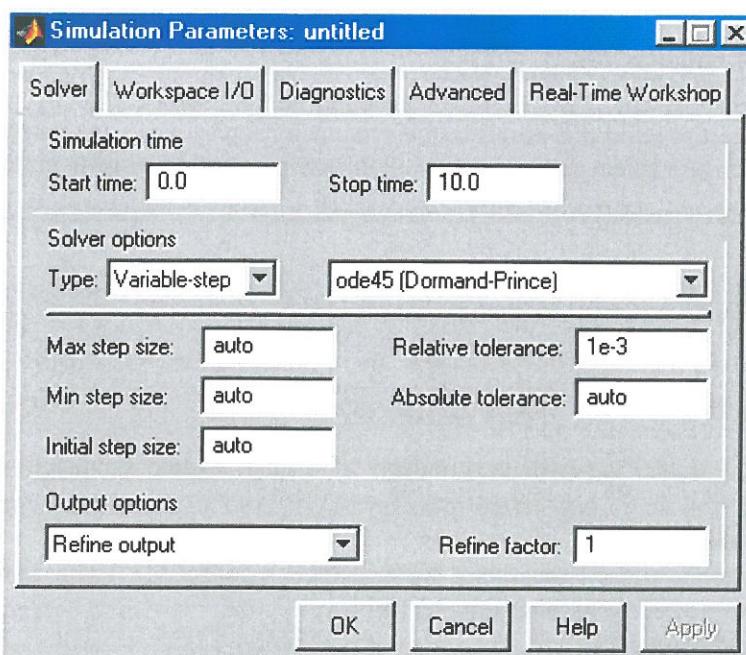
Wykonaj w ten sposób wszystkie konieczne połączenia. Pozostaje jeszcze wprowadzenie parametrów badanej transmitancji w tym celu kliknij dwukrotnie na blok Transfer Fcn



Rys. 1.7. Wprowadzenie parametrów badanej transmitancji

i wprowadź współczynniki licznika i mianownika transmitancji. Uzyskany został model pokazany na rysunku 1.

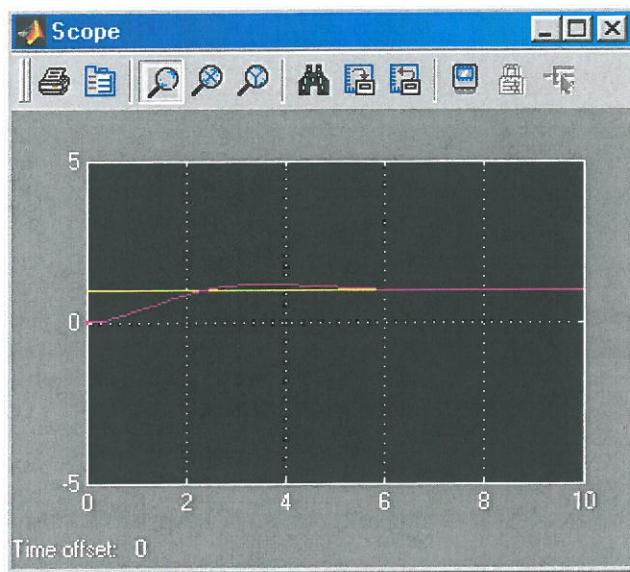
Teraz otwórz blok Scope (oscyloskop) do przeglądania wyników symulacji. Mając otwarte okno oscyloskopu uruchom symulację na 10 sekund. Ustawianie parametrów symulacji odbywa się w oknie **Simulation Parameters** z menu **Simulation**.



Rys. 1.8. Wprowadzenie parametrów symulacji

Zauważ, że w oknie dialogowym domyślnie ustawiona jest wartość czasu końcowego (**Stop time**) na 10.0 sekund. Po ustawieniu parametrów symulacji kliknij na przycisk OK. SIMULINK przyjmuje te wartości i zamyka okno dialogowe.

W menu **Simulation** wybierz polecenie **Start** i obserwuj przebiegi w bloku Scope.



Rys. 1.9. Wprowadzenie parametrów symulacji

Symulacja kończy się osiąga czas końcowy (**Step time**) określony w oknie dialogowym **Simulation Parameters** lub kiedy zostanie wybrane polecenie **Stop** z menu **Simulation**. Aby zachować ten model, wybierz polecenie **Save** z menu **File** i wprowadź nazwę i wybierz katalog w którym ma być umieszczony. Plik ten zawiera opis modelu.

3. ZAPISYWANIE WYNIKÓW SYMULACJI

W SIMULINKU możliwe jest również przeprowadzanie symulacji i zapisywanie wybranych zmiennych do pamięci roboczej MATLABA celem dalszej obróbki. Poniższy przykład ilustruje te możliwości.

Przykład 2

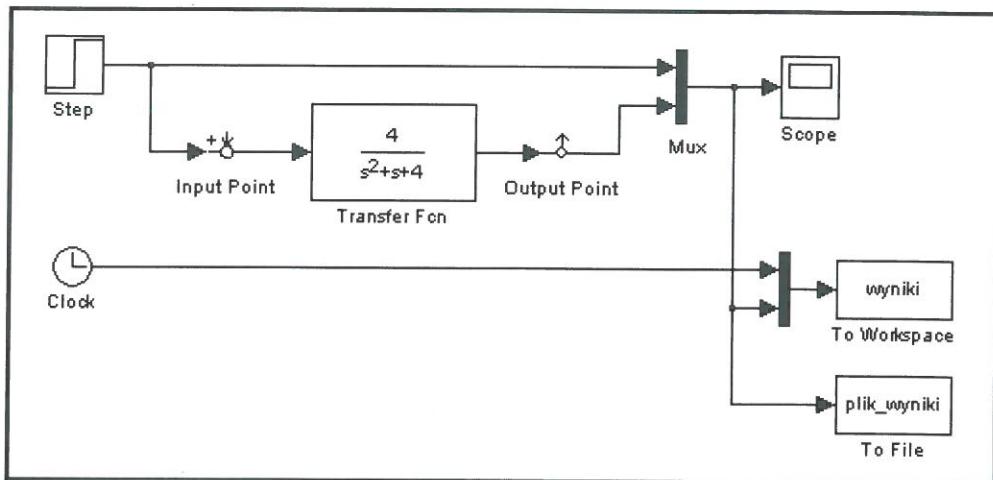
Wyznaczyć odpowiedź skokową układu o transmitancji

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.1)$$

dla $\omega_n = 2$ oraz $\zeta = 0.25$. Czas trwania symulacji 10 sekund. Wyniki symulacji wyświetlić na oscyloskopie oraz zapisać do przestrzeni roboczej MATLABA i w pliku dyskowym pod nazwą `plik_wyniki`, następnie:

- uzyskane wyniki symulacji znajdujące się w pamięci roboczej MATLABA zapisać do pliku dyskowego tekstowego o nazwie `odp_skokowa`,
- zamknąć okno SIMULINKA,
- wyczyścić przestrzeń roboczą MATLABA poleceniem `clear`,
- odczytać umieszczone w pliku dyskowym dane i przedstawić je na wykresie,
- uzyskany wykres zamieścić w dokumencie Worda,

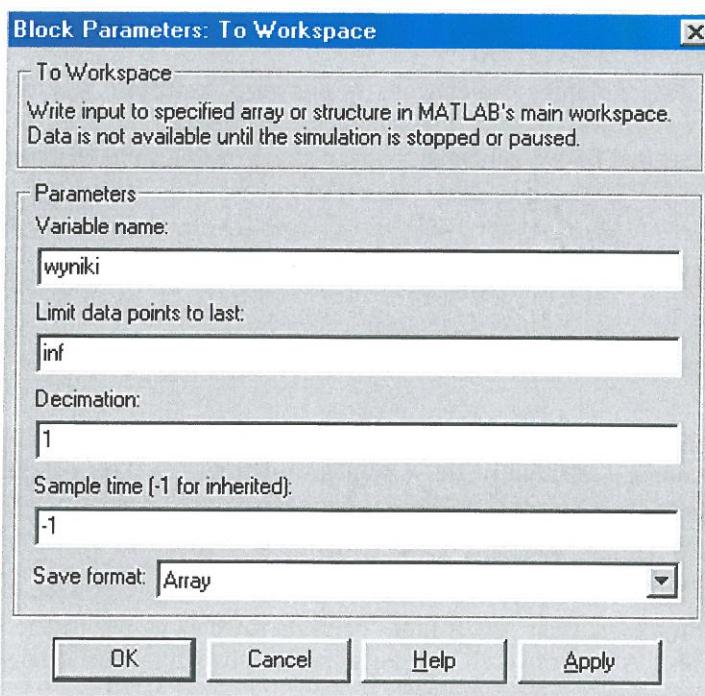
Rozwiązanie. Zadanie to zrealizowane zostanie w następującym układzie.



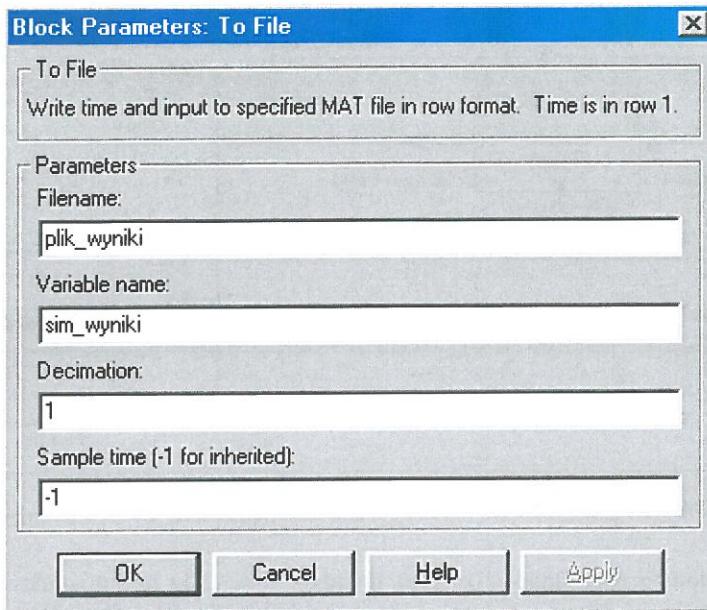
Rys. 2.1. Model układu z przykładu 2

W porównaniu z modelem z przykładu 1 w tym układzie (rys. 2.1) zmianie uległa badana transmitancja oraz umieszczone zostały dodatkowe bloki takie jak: To Workspace (zapis danych do przestrzeni roboczej MATLABA) To File (zapis danych do pliku binarnego). Po dwukrotnym kliknięciu myszką na blok To Workspace pojawi się okno dialogowe pokazane na rysunku 2.2, w oknie tym należy zmienić nazwę zbioru zapisywanyego do przestrzeni roboczej (Variable name) na wyniki oraz ustawić typ zapisywanych danych (Save format) na Array (tablicowe). Wcisnąć przycisk OK. i zamknąć okno dialogowe bloku To Workspace. Dokonać również zmian w okienku dialogowym dla bloku o nazwie To File pokazanym na rysunku 2.3, w okienku tym zmienić nazwę zbioru dyskowego (Filename) na plik_wyniki oraz nazwę zapisywanych zmiennych (Variable name) na sim_wyniki.

Uzyskany model układu z rysunku 2.1. zachować pod nazwą uklad_IIRz.mdl.



Rys. 2.2. Okno dialogowe bloku To Workspace



Rys. 2.3. Okno dialogowe bloku To File

Otworzyć okno oscyloskopu (Scope) i przeprowadzić symulację. Wyniki symulacji zapisane zostały w binarnym pliku dyskowym o nazwie `plik_wyniki.mat` oraz w przestrzeni roboczej MATLABA umieszczona została tablica o nazwie `wyniki`. Wpisując w oknie komend MATLABA polecenie

```
>> dir *.mat
```

można sprawdzić czy utworzony został binarny plik dyskowy o nazwie `plik_wyniki.mat` oraz wpisując nazwę utworzonej w przestrzeni roboczej tablicy

```
>> wyniki
```

na ekranie pojawią się dwie kolumny zmiennych, w pierwszej kolumnie jest wymuszenie a w drugiej odpowiedź wyjściowa badanej transmitancji. Dodatkowo w pamięci roboczej MATLABA znajduje się czas pod nazwą zmiennej `tout` o czym można się przekonać wpisując polecenie

```
>> whos
```

Name	Size	Bytes	Class
<code>tout</code>	<code>57x1</code>	456	double array
<code>wyniki</code>	<code>57x2</code>	912	double array

`Grand total is 171 elements using 1368 bytes`

Chcąc zapisać wyniki symulacji znajdujące się w pamięci roboczej MATLABA do tekstowego pliku dyskowego o nazwie `odp_skokowa` należy zapisać następujące komendy

```
>> wyniki_txt = [tout wyniki]
>> save odp_skokowa wyniki_txt -ascii
```

Utworzony zostanie plik dyskowy typu ASCII który pozwala również na późniejsze wczytanie tych danych, np. do Excela. Aby sprawdzić czy rzeczywiście utworzony został tekstowy plik dyskowy o nazwie `odp_skokowa` należy zapisać

```
>> dir *
```

Teraz należy zamknąć okno SIMULINKA i wyczyścić przestrzeń roboczą MATLABA komendą

```
>> clear
```

Wpisując teraz do okna komend MATLABA nazwę zbioru wyniki

```
>> wyniki
```

uzyska się następującą odpowiedź

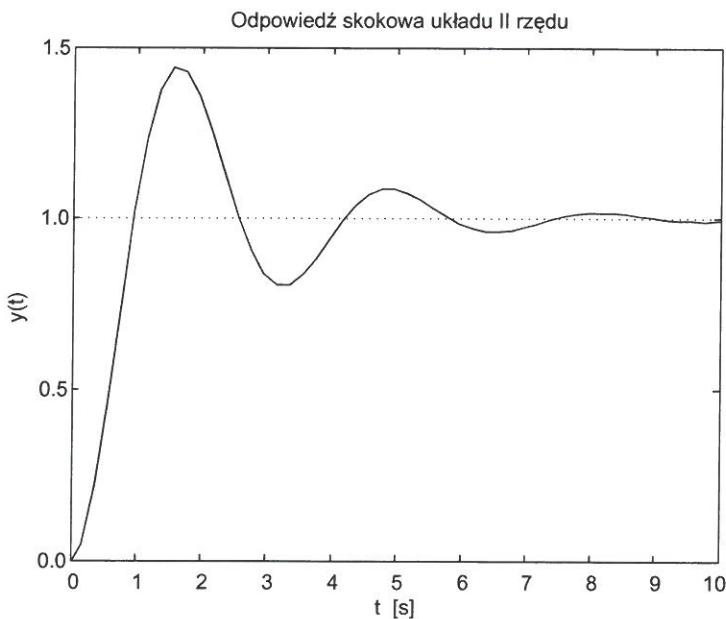
```
??? Undefined function or variable 'wyniki'.
```

Jednak wyniki uzyskane podczas symulacji zabezpieczone zostały w plikach dyskowych pod nazwami: `plik_wyniki` i `odp_skokowa`. Odczyt wyników symulacji zapisanych w plikach dyskowych i następnie przedstawienie tych danych na wykresie zostanie wykonane przy użyciu następującego kodu programu

```
clear
close all
load odp_skokowa -ascii      % Odczyt danych
                                % z tekstopowego pliku dyskowego
wyniki = odp_skokowa;          % Wstawienie tych danych
                                % do tablicy o nazwie: wyniki
                                % Wektor czasu
ta = wyniki(:,1)              % Wektor wymuszenia
ua = wyniki(:,2)              % Wektor odpowiedzi
ya = wyniki(:,3)              % Wektor czasu

load plik_wyniki               % Odczyt danych
                                % z binarnego pliku dyskowego
tb = sim_wyniki(1,:);          % Wektor czasu
ub = sim_wyniki(2,:);          % Wektor wymuszenia
yb = sim_wyniki(3,:);          % Wektor odpowiedzi

figure(1)
plot( ta, ua, 'b:', ta, ya, 'b-', tb, ub, 'k:', tb, yb, 'k-' )
xlabel('t [s]')
ylabel('y(t)')
title('Odpowiedź skokowa układu II rzędu')
```



Rys. 2.4. Uzyskany w MATLABIE rysunek z wynikami symulacji

Przy pisaniu skryptu w MATLABIE na uwagę zasługuje sposób podstawiania danych pod zmienne (np. ta i tb). Wyniki symulacji, które zapisywane były bezpośrednio do binarnego pliku dyskowego w bloku To File (rys. 2.1) umieszczane tam były wierszami, przy czym do pliku dodatkowo bez udziału użytkownika, w pierwszym wierszu wpisany został czas. Wyniki symulacji zapisane w tekstowym pliku dyskowym umieszczone zostały tam kolumnami, wynika to z tego, że MATLAB zapisuje te same dane w bloku To Workspace (rys. 2.1) do pamięci roboczej – kolumnami.

Uzyskany wykres przedstawiający wyniki symulacji pokazany został na rysunku 2.4. Wykres ten może być zachowany w pliku dyskowym do późniejszego odczytania go w oknie graficznym MATLABA lub wyeksportowany np. z rozszerzeniem *.emf w celu późniejszego wstawienia go np. do dokumentu tworzonego w Wordzie.

4. STEROWANIE MODELEM SIMULINKA Z POZIOMU KOMEND MATLABA

Bardzo dogodnym rodzajem pracy ze schematami utworzonymi w Simulinku jest możliwość sterowania pracą Simulinka z poziomu Matlaba. Zadanie to wykonuje się poprzez napisanie odpowiedniego skryptu i sterowanie wykonywaniem symulacji w SIMULINKU przy pomocy odpowiednich komend MATLABA. Problem ten zilustrowany został w przykładzie 3.

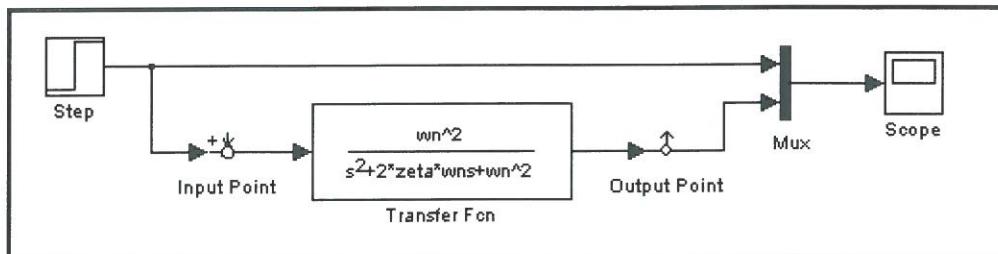
Przykład 3

Wyznaczyć odpowiedź skokową układu o transmitancji

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.1)$$

dla $\omega_n = 2$ oraz $\zeta = 0.25$. Czas trwania symulacji 15 sekund. Transmitancję zaimplementować w Simulinku, natomiast sterowanie symulacją ma się odbywać z poziomu Matlaba.

Rozwiązanie. Do realizacji tak postawionego zadania zbudowany został schemat pokazany na rysunku 3.1 i zapisany pod nazwą uklad_IIRz.mdl. W bloku Transfer Fcn wartości liczbowe parametrów muszą zostać zastąpione zmiennymi (rys. 3.2).



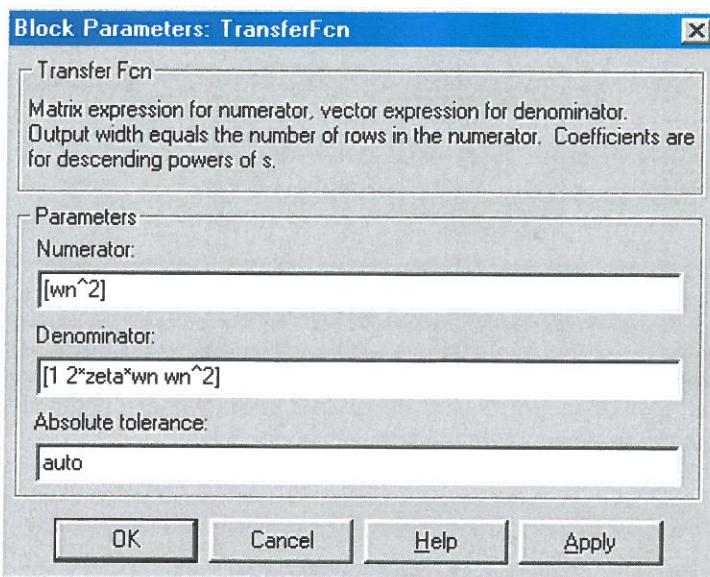
Rys. 3.1. Schemat modelu z zastąpionymi współczynnikami transmitancji na zmienne

Dodatkowo napisany został poniższy program Matlaba i zapisany pod nazwą model_IIRz.m

```

clear % Wyczyszczenie pamięci roboczej Matlaba
wn = 2; % Wartość częstotliwości drgań własnych
zeta = 0.6; % Wartość współczynników tłumienia
tmax = 6; % Zakres czasowy wykonywanej symulacji
open_system('uklad_IIRz') % Otwarcie modelu Simulinka
sim('uklad_IIRz', tmax) % Wykonanie symulacji w zadanym odcinku
% czasu
close_system % Zamknięcie modelu Simulinka

```



Rys. 3.2. Okno dialogowe bloku transmitancji z wprowadzonymi zmiennymi

5. MASKOWANIE PODUKŁADÓW MODELU SIMULINKA

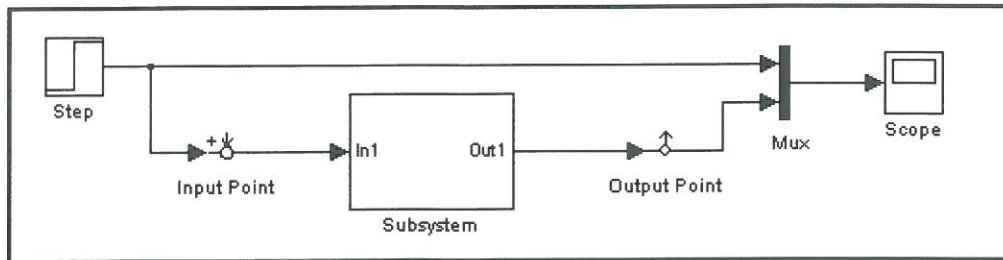
Maskowanie jest właściwością SIMULINKA pozwalającą na wprowadzenie okna dialogowego oraz ikony dla podukładu. Przy użyciu maskowania można:

- uprościć model poprzez zastąpienie pojedynczym oknem wielu okien dialogowych występujących w podukładzie. Zamiast wprowadzania parametrów dla każdego bloku w oddzielnym oknie dialogowym, te parametry są wprowadzane w oknie dialogowym maski i wprowadzane do bloków w zamaskowanym podsystemie;
- zastosować bardziej opisowy i użyteczny interfejs przez zdefiniowanie okna dialogowego ze swoimi własnymi zmiennymi i ich opisem oraz tekstem pomocy;
- zdefiniować komendy, które obliczają zmienne których wartości zależą od parametrów bloku;
- utworzyć ikonę bloku, która oznacza własności podsystemu
- zabezpieczyć przed niezamierzonymi modyfikacjami podukładu poprzez ukrycie ich zawartości pod wprowadzony interfejs.

Przykład 4

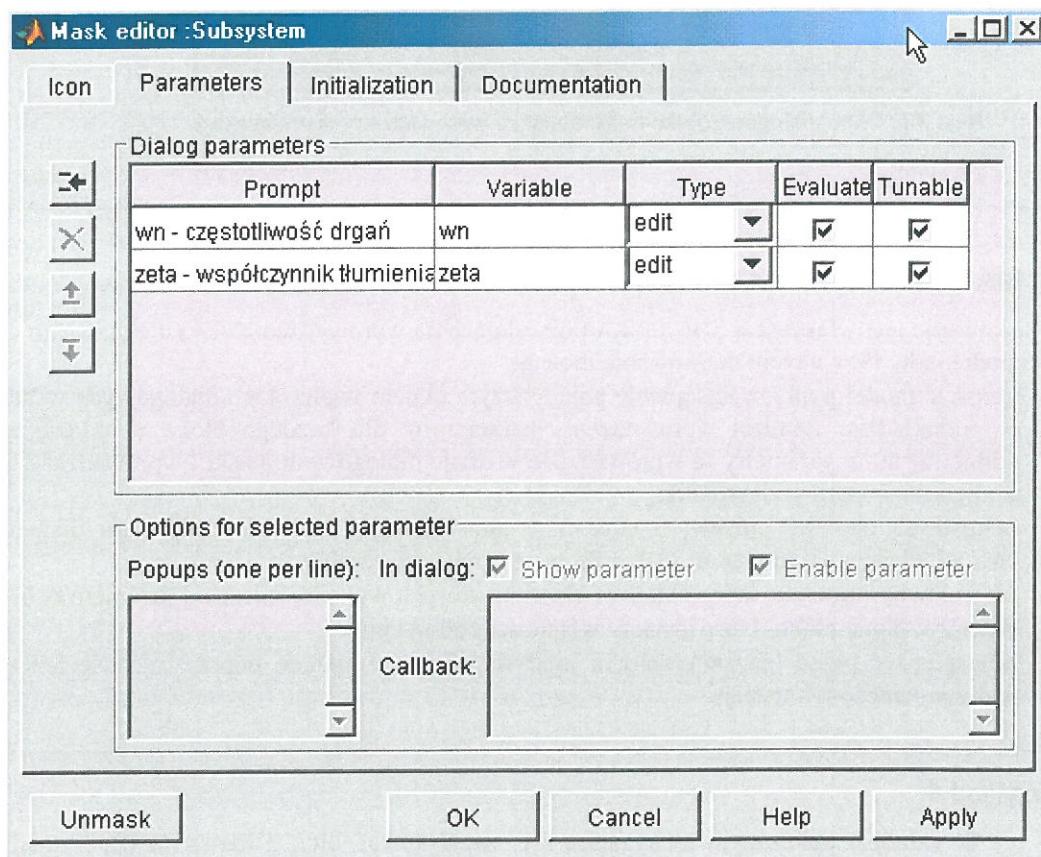
W układzie pokazanym na rysunku 3.1. zamaskować blok z transmitancją II rzędu opisana równaniem (3.1) i ustawić wartości parametrów na $\omega_n = 1$ oraz $\zeta = 0.5$. Dodatkowo zaprezentować w jaki sposób te parametry na masce mogą być zmieniane przy użyciu kodu Matlaba.

Rozwiązanie. W zbudowanym w Simulinku modelu pokazanym na rysunku 3.1 z bloku Transfer Fcn utworzony zostanie podukład, co realizuje się przez zaznaczenie tego bloku i wybranie polecenia **Edit/Create Subsystem**. Spowoduje to zastąpienie bloku transmitancji nową ikoną o nazwie Subsystem (rys. 4.1), którą to nazwę można zmienić na inną. Klikając teraz dwukrotnie na ikonę bloku Subsystem można otworzyć nowe okno z blokiem transmitancji z której uczyniony został podukład.



Rys. 4.1. Model układu z utworzonym podukładem (Subsystem)

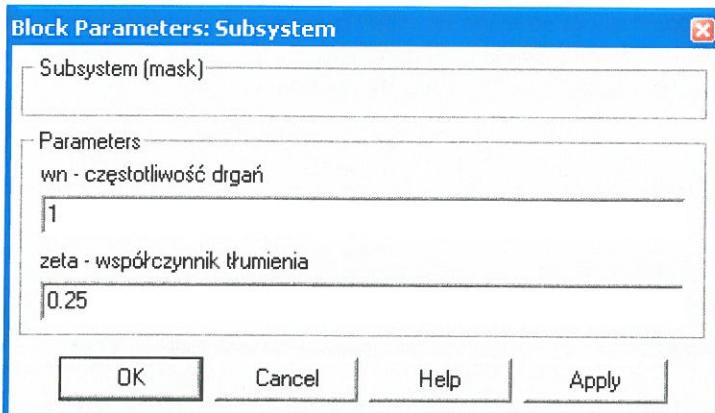
Maskowanie bloku Subsystem odbywa się poprzez zaznaczenie jego ikony i wybranie polecenia **Edit/Mask subsystem** spowoduje pojawienie się edytora maskowanego podukładu w edytorze tym wprowadzone zostaną zmienne, które występują w podukładzie.



Rys. 4.2. Okno edytora maski podsystemu

Po wcisnięciu przycisku **OK** w oknie dialogowym edytora maski, podukład Subsystem zostanie zamaskowany i po dwukrotnym kliknięciu na jego ikonę pojawi się okno dialogowe zamaskowanego podukładu (rys. 4.5), do którego wpisuje się wartości parametrów transmitancji w tym przypadku układu II rzędu czyli $\omega_n = 1$ oraz $\zeta = 0.25$.

Teraz do zamaskowanego podukładu dostęp jest możliwy po wybraniu opcji **Edit/Look under mask**.



Rys. 4.5. Okno dialogowe zamaskowanego podukładu

6. MODELOWANIE RÓWNAŃ DYNAMICZNYCH W SIMULINKU

SIMULINK pozwala na modelowanie dowolnego zestawu równań różniczkowych, szczególnie równań nieliniowych, które nie mogą być rozwiązywane w inny sposób. W podrozdziale tym przedstawione zostaną przykłady ilustrujący sposób modelowania w Simulinku równań dynamicznych liniowych.

Przykład 5

W przykładzie 4 zamaskowany został model matematyczny układu przedstawiony w postaci następującej transmitancji II rzędu

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (5.1)$$

Rozbudować w Simulinku układ z przykładu 4 o równoważny opis przedstawiony w postaci równań dynamicznych. Uzyskane wyniki porównać na wspólnym oscyloskopie.

Rozwiążanie. W pierwszej kolejności transmitancja opisana wzorem (5.1) przekształcona zostanie do następującego, równoważnego równania różniczkowego

$$\frac{d^2 y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) = \omega_n^2 u(t) \quad (5.2)$$

Pojedyncze równanie różniczkowe drugiego rzędu może zostać zdekomponowane do dwóch równań różniczkowych pierwszego rzędu przez wprowadzenie zmiennych stanu. W tym przypadku zostaną wprowadzone zostały dwie zmienne stanu o postaci

$$x_1(t) = y(t) \quad (5.3.a)$$

$$x_2(t) = \frac{dy(t)}{dt} \quad (5.3.b)$$

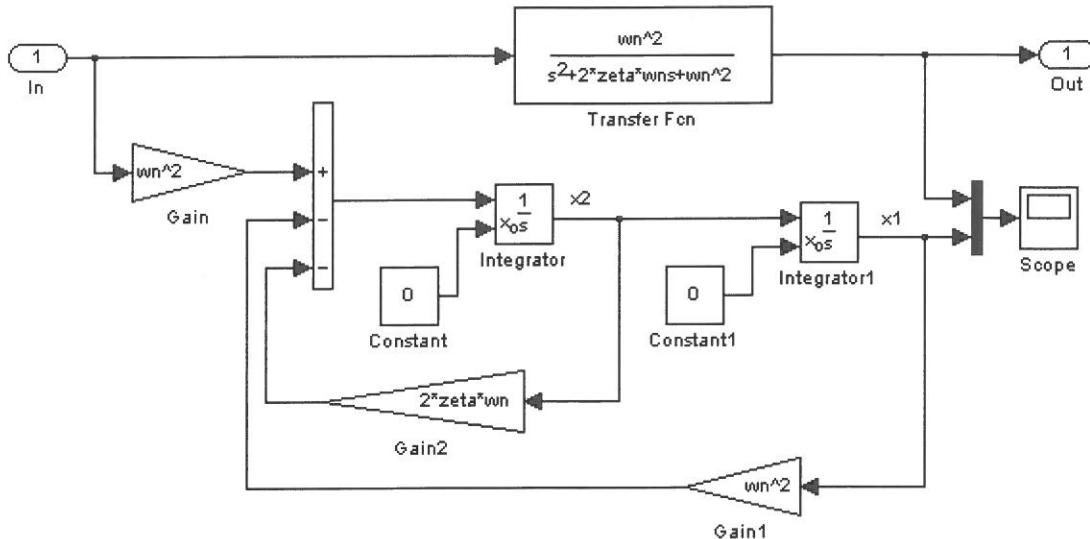
W ten sposób uzyskuje się następujące równania stanu

$$\dot{x}_1(t) = x_2(t) \quad (5.4.a)$$

$$\dot{x}_2(t) = -\omega_n^2 x_1(t) - 2\zeta\omega_n x_2(t) + \omega_n^2 u(t) \quad (5.4.b)$$

Schemat Simulinka z zamodelowanymi równania różniczkowymi (5.4.a), (5.4.b) pokazany został na rysunku 5.1. Opis przy użyciu równań stanu pozwala na wprowadzanie warunków

początkowych, które wyprowadzone zostały na wejścia integratorów. Jeżeli poprawnie został zamodelowany opis przy użyciu równań dynamicznych, to przy zerowych warunkach początkowych przebiegi czasowe na ‘Scopie’ będą się pokrywać.



Rys. 5.1. Schemat rozbudowanego wnętra bloku ‘Subsystem’ z rysunku 4.3

Przykład 6

Obiekt regulacji opisany jest poniższym zestawem równań stanu

$$\dot{x}_1(t) = -x_1(t) + 2 \cdot x_3(t) + u(t) \quad (6.1.a)$$

$$\dot{x}_2(t) = x_1(t) + x_2(t) - x_3(t) \quad (6.1.b)$$

$$\dot{x}_3(t) = x_2(t) - x_3(t) \quad (6.1.c)$$

i następującym równaniem wyjścia

$$y(t) = x_2(t) + x_3(t) \quad (6.2)$$

- a) Wyznacz transmitancję operatorową obiektu $G(s)$.
- b) Zamodeluj powyższe równania dynamiczne w Simulinku.

Rozwiążanie. W pierwszej kolejności równania (6.1) – (6.2) zapisane zostaną w postaci macierzowej, równań stanu

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 2 \\ 1 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cdot u \quad (6.3)$$

i równania wyjścia

$$y = [0 \ 1 \ 1] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (6.4)$$

Transmitancja operatorowa wyznaczona zostanie przy użyciu komendy Matlaba o nazwie `ss2tf`. Najpierw muszą zostać zakodowane macierze w postaci następującego kodu programu Matlaba

```

clear                                     % wyczyszczenie pamięci roboczej Matlaba
clc                                       % wyczyszczenie okienka 'Command Window'
A = [-1 0 2; 1 1 -1; 0 1 -1]           % macierz stanu
B = [ 1; 0; 0]                           % macierz wejścia
C = [0 1 1]                             % macierz wyjścia
D = 0                                     % macierz transmisji
[num, den] = ss2tf( A, B, C, D)          % Wyznaczenie transmitancji

```

Po uruchomieniu powyższego kodu programu Matlab odpowiada następująco

```

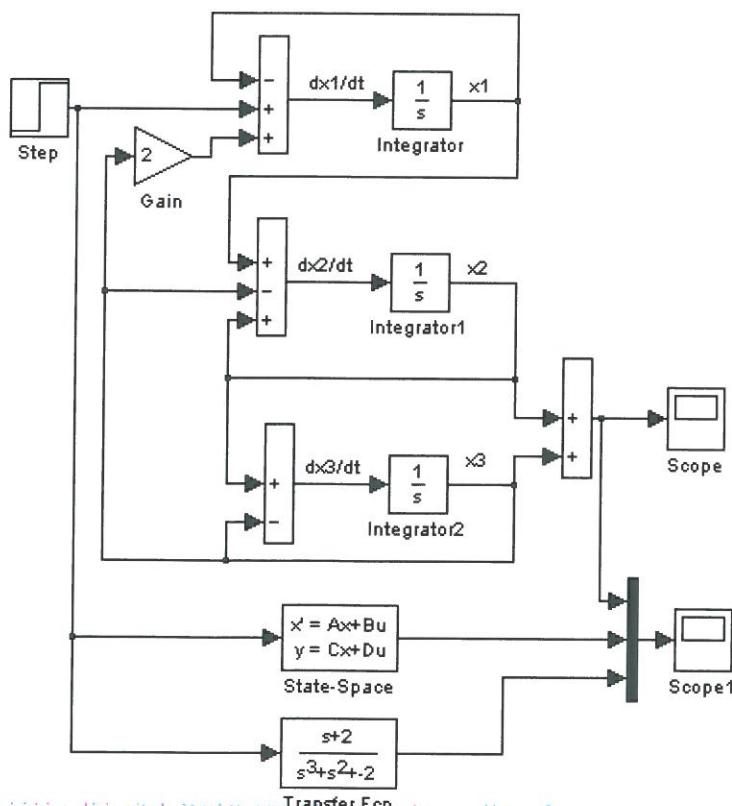
num =
0      0.0000    1.0000    2.0000
den =
1.0000    1.0000    0.0000   -2.0000

```

Wektor num zawiera współczynniki licznika, natomiast wektor den – współczynniki mianownika transmitancji, czyli w tym przypadku uzyskuje się następującą transmitancję.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{s+2}{s^3 + s^2 - 2} \quad (6.5)$$

Wyznaczona transmitancja opisana równaniem (6.5) będzie stanowić wzorzec przy modelowaniu w Simulinku równań opisanych równaniami (6.1) – (6.2). Na rysunku 6.1. znajduje się uzyskany model w Simulinku dla rozważanego zestawu równań dynamicznych.



Rys. 6.1. Schemat układu zamodelowanego w Simulinku.

Przykład 7

Dla poniższego równania różniczkowego

$$\frac{d^2 y(t)}{dt^2} + 2 \frac{dy(t)}{dt} + 5 y(t) = 10 \cdot 1(t) \quad (7.1)$$

z warunkami początkowymi

$$y(0) = -4 \quad (7.2.a)$$

$$y^{(1)}(0) = -10 \quad (7.2.b)$$

- a) Wyznacz równania dynamiczne
- b) Zamodeluj uzyskane równania dynamiczne w Simulinku.
- c) Rozwiąż powyższe równanie różniczkowe przy użyciu rachunku operatorowego Laplace'a i porównaj ze sobą uzyskane rozwiązania czasowe.

Rozwiązań.

a) W pierwszej kolejności pojedyncze równanie różniczkowe (7.1) zostanie zdekomponowane do dwóch równań różniczkowych pierwszego rzędu przez wprowadzenie zmiennych stanu. W tym przypadku zostaną wprowadzone dwie zmienne stanu o postaci

$$x_1(t) = y(t) \quad (7.3.a)$$

$$x_2(t) = \frac{dy(t)}{dt} \quad (7.3.b)$$

Różniczkując obustronnie zdefiniowane zmienne stanu (7.3) uzyskuje się następujące równania różniczkowe pierwszego rzędu

$$\dot{x}_1(t) = \frac{dx_1(t)}{dt} = \frac{dy(t)}{dt} \quad (7.4.a)$$

$$\dot{x}_2(t) = \frac{dx_2(t)}{dt} = \frac{d^2 y(t)}{dt^2} \quad (7.4.b)$$

W równaniach (7.4) wprowadzone zostaną zmiany po prawej stronie znaku równości. Do równania (7.4.a) zostanie wprowadzona zdefiniowana w równaniu (7.3.b) zmienna stanu x_2 , natomiast do równania (7.4.b) zostanie podstawiona druga pochodna $\ddot{y}(t)$ wyznaczona z równania (7.1) i następnie zmienne: $y(t)$ oraz pierwsza pochodna $\dot{y}(t)$ zostaną zastąpione zmiennymi stanu zdefiniowanymi równaniami (7.3).

$$\dot{x}_1(t) = x_2(t) \quad (7.5.a)$$

$$\dot{x}_2(t) = -5x_1(t) - 2x_2(t) + 10 \cdot 1(t) \quad (7.5.b)$$

Na podstawie zadanych w równaniu różniczkowym warunków początkowych (7.2) wyznaczone zostaną warunki początkowe dla wektora stanu

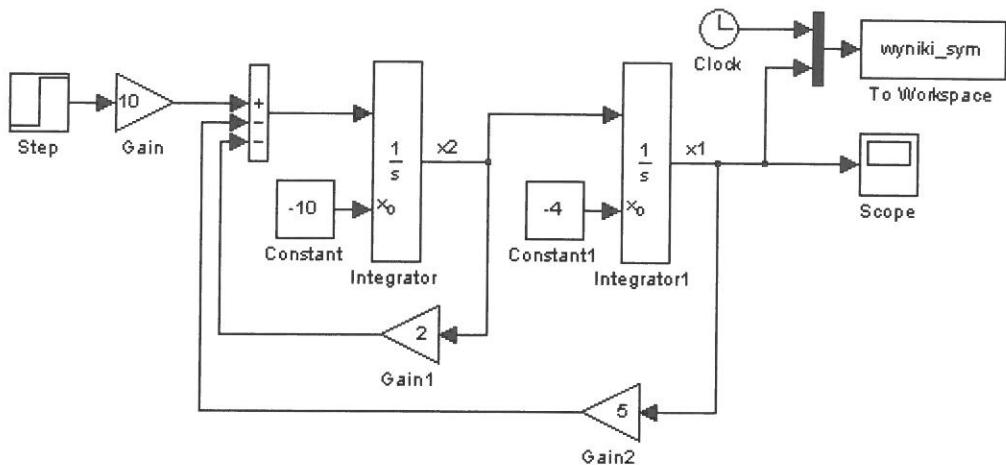
$$x_1(0) = y(0) = -4 \quad (7.6.a)$$

$$x_2(0) = y^{(1)}(0) = \frac{dy(0)}{dt} = -10 \quad (7.6.b)$$

Z równania (7.3.a) wyznaczone zostanie równanie wyjścia

$$y(t) = x_1(t) \quad (7.7)$$

b) Uzyskane równania (7.5), (7.6) i (7.7) zostały zamodelowane w Simulinku na integratorach i uzyskany schemat blokowy pokazany został na rysunku 7.1.



Rys. 7.1. Zamodelowane w Simulinku uzyskane równania dynamiczne (ex_7_sim.mdl).

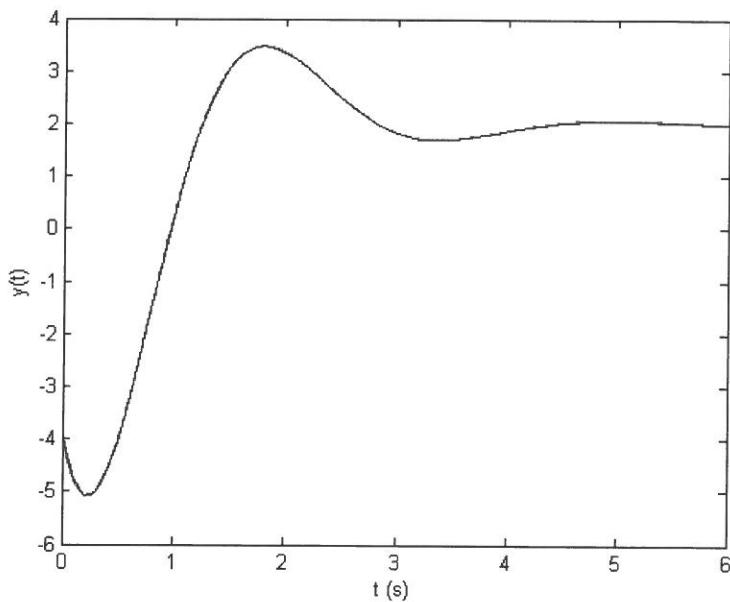
c) Korzystając z rachunku operatorowego Laplace'a, dla równania różniczkowego (7.1) z warunkami (7.2) wyznaczone zostało następujące rozwiązanie operatorowe [2]

$$Y(s) = \frac{-4s^2 - 18s + 10}{s^3 + 2s^2 + 5s} \quad (7.8)$$

którego postać czasowa jest następująca

$$y(t) = 10 \cdot e^{-t} \cos(2t + 126.9^\circ) + 2 \cdot 1(t) \quad (7.9)$$

Na rysunku 7.2 przedstawione zostały wykresy czasowe rozwiązań równania różniczkowego (7.1) uzyskane w zamodelowanym układzie w Simulinku (Rys. 7.1) i przy użyciu rachunku operatorowego Laplace'a (7.9)



Rys. 7.2. Porównanie rozwiązań równania różniczkowego (7.1) uzyskanych po zastosowaniu rachunku operatorowego Laplace'a i po zamodelowaniu w Simulinku uzyskanych równań dynamicznych.

Wyniki przedstawione na rysunku 7.2. uzyskane zostały przy użyciu następującego kodu programu zapisanego w Matlabie.

```

close all % Zamknięcie wszystkich okien graficznych

close all % Zamknięcie wszystkich okien graficznych
clear % Wyczyszczenie pamięci roboczej Matlaba
tmax = 6; % Odcinek czasu
open_system('ex_7_sim') % Otwarcie przygotowanego modelu Simulinka
sim('ex_7_sim',tmax) % Wykonanie symulacji w zadanym odcinku
% czasu

tS = wyniki_sym(:,1); % Podstawienie wektora czasu
yS = wyniki_sym(:,2); % Pobranie z przestrzeni roboczej Matlaba
% uzyskanego rozwiązania

% Rachunek operatorowy Laplace'a
numC = [-4 -18 10]
denC = [1 2 5 0]

[rC, pC, kC] = residue( numC, denC)

M = abs( rC(1))
fi = angle( rC(1))*180/pi
A = 2*M
sigma = real( pC(1))
w = imag( pC(1))

tC = [0:0.1:tmax]';
yC = A*exp(sigma*tC).*cos(w*tC + fi*pi/180) + rC(3);

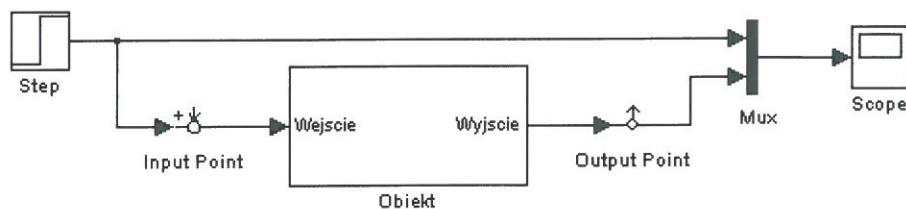
id = figure(1)
plot(tS, yS, tC, yC, 'k')
xlabel('t (s)')
ylabel('y(t)')

set( id, 'Color', [1 1 1])

```

ĆWICZENIA W SIMULINKU

S1. Zbudować model pracujący w układzie z rys. S1.1



Rys. S1.1. Schemat blokowy

W bloku obiekt należy zamaskować układ o poniższej transmitancji

$$G(s) = k \frac{1 + bs}{(1 + Ts)^2 (a_2 s^2 + a_1 s + a_0)}$$

w której ma być dostęp do każdego parametru.

Po zamaskowaniu podstawić w okienku dialogowym zamaskowanego bloku wartości liczbowe poszczególnych współczynników transmitancji, wyznaczonych z liter imienia i nazwiska jednej z osób ćwiczących w danej podgrupie i tak:

$$\begin{aligned}k &= \text{liczba liter w imieniu} \\b &= 0.1 * (\text{liczba liter w nazwisku}) \\T &= \text{liczba spółgłosek w imieniu} \\a_2 &= \text{liczba spółgłosek w nazwisku} \\a_0 &= \text{liczba samogłosek w nazwisku} \\a_1 &= \sqrt{2a_2 a_0}\end{aligned}$$

Dla tak utworzonego schematu przeprowadzić symulację.

M2. Wyznacz transmitancje operatorowe $G(s) = Y(s)/U(s)$ dla układów opisanych następującymi zestawami równań dynamicznych i zamodeluj je w Simulinku.

- a) $\begin{aligned}\dot{x}_1(t) &= -2x_1(t) - 3x_3(t) + u(t) \\ \dot{x}_2(t) &= x_1(t) - x_2(t) + x_3(t) \\ \dot{x}_3(t) &= x_2(t) + x_3(t) - u(t) \\ y(t) &= 3x_1(t) + x_2(t) + 2x_3(t)\end{aligned}$
- b) $\begin{aligned}\dot{x}_1(t) &= -2x_1(t) - 5x_2(t) - 6x_3(t) + u(t) \\ \dot{x}_2(t) &= x_1(t) - x_2(t) + x_3(t) - u(t) \\ \dot{x}_3(t) &= x_2(t) + u(t) \\ y(t) &= x_1(t) + 3x_2(t) + 6x_3(t)\end{aligned}$
- c) $\begin{aligned}\dot{x}_1(t) &= -x_2(t) - x_3(t) \\ \dot{x}_2(t) &= x_1(t) + u(t) \\ \dot{x}_3(t) &= 5x_1(t) + 2x_2(t) + x_3(t) - u(t) \\ y(t) &= 2x_1(t) + 3x_2(t) + 5x_3(t)\end{aligned}$
- d) $\begin{aligned}\dot{x}_1(t) &= -7x_1(t) - 2x_2(t) + u(t) \\ \dot{x}_2(t) &= x_1(t) - x_2(t) - x_3(t) \\ \dot{x}_3(t) &= x_2(t) \\ y(t) &= 3x_1(t) + x_2(t) + 3x_3(t)\end{aligned}$
- e) $\begin{aligned}\dot{x}_1(t) &= -4x_1(t) - 4x_2(t) - 2x_3(t) + u(t) \\ \dot{x}_2(t) &= x_1(t) + x_2(t) - x_3(t) + 2u(t) \\ \dot{x}_3(t) &= x_2(t) + x_3(t) - u(t) \\ y(t) &= -2x_1(t) + x_2(t) + 2x_3(t)\end{aligned}$

f) $\dot{x}_1(t) = -4x_1(t) - x_2(t) - 2x_3(t) + u(t)$

$$\dot{x}_2(t) = x_1(t) + x_2(t)$$

$$\dot{x}_3(t) = x_2(t) + x_3(t) - u(t)$$

$$y(t) = 3x_1(t) + 6x_2(t) + 8x_3(t)$$

g) $\dot{x}_1(t) = -2x_1(t) - x_2(t) - 5x_3(t) + u(t)$

$$\dot{x}_2(t) = x_1(t) + x_2(t) + x_3(t)$$

$$\dot{x}_3(t) = -x_1(t) + x_2(t) - x_3(t)$$

$$y(t) = 2x_1(t) + 3x_2(t) + 5x_3(t)$$

h) $\dot{x}_1(t) = -x_1(t) - 2x_2(t) + x_3(t) + u(t)$

$$\dot{x}_2(t) = x_1(t) + x_2(t) + x_3(t) - 2u(t)$$

$$\dot{x}_3(t) = -x_1(t) + x_2(t) - x_3(t) - u(t)$$

$$y(t) = 2x_1(t) + x_2(t) + 3x_3(t)$$

M3. Zapisz poniższe równania różniczkowe w postaci równań dynamicznych i zamodeluj je na integratorach w Simulinku. Uzyskane rozwiązania porównaj z wynikami uzyskanymi przy użyciu rachunku operatorowego Laplace'a.

a) $\frac{d^3 y(t)}{dt^3} + 5 \frac{d^2 y(t)}{dt^2} + 4 \frac{dy(t)}{dt} = 2\delta(t)$

$$y(0) = 1$$

$$y^{(1)}(0) = -1$$

$$y^{(2)}(0) = 1$$

b) $\frac{d^2 y(t)}{dt^2} + 5 \frac{dy(t)}{dt} + 6 y(t) = 2e^{-t}$

$$y(0) = 0$$

$$y^{(1)}(0) = -2$$

c) $\frac{dy(t)}{dt} + 2 y(t) = 6 \cos 2t$

$$y(0) = 2$$

d) $\frac{d^3 y(t)}{dt^3} + 4 \frac{d^2 y(t)}{dt^2} + 3 \frac{dy(t)}{dt} = 3\delta(t)$

$$y(0) = 1$$

$$y^{(1)}(0) = -1$$

$$y^{(2)}(0) = 1$$

e) $\frac{dy(t)}{dt} + 2 y(t) = \frac{1}{2}t^2$

$$y(0) = -1$$

f) $\frac{dy(t)}{dt} + 3 y(t) = 4 \sin 4t$

$$y(0) = 1$$

g) $\frac{d^2 y(t)}{dt^2} + 4 \frac{dy(t)}{dt} + 5 y(t) = \cdot 1(t)$
 $y(0) = -3$
 $y^{(1)}(0) = 2$

h) $\frac{d^2 y(t)}{dt^2} + 3 \frac{dy(t)}{dt} + 2 y(t) = 3 e^{-3t}$
 $y(0) = 0$
 $y^{(1)}(0) = 1$

i) $\frac{d^2 y(t)}{dt^2} + 2 \frac{dy(t)}{dt} + 2 y(t) = 3 \sin 2t$
 $y(0) = 2$
 $y^{(1)}(0) = -3$

j) $\frac{d^2 y(t)}{dt^2} + 2 \frac{dy(t)}{dt} + y(t) = 5 e^{-3t} + t$
 $y(0) = 2$
 $y^{(1)}(0) = 1$

k) $\frac{d^2 y(t)}{dt^2} + 4 y(t) = t^2$
 $y(0) = 1$
 $y^{(1)}(0) = 2$

l) $\frac{d^2 y(t)}{dt^2} + 3 \frac{dy(t)}{dt} + 6y(t) = e^{-3t} \cos 2t$
 $y(0) = 1$
 $y^{(1)}(0) = 2$

LITERATURA

1. Mrozek B., Mrozek Z. (2018). *MATLAB i Simulink. Poradnik użytkownika.* Wydanie IV. Wydawnictwo Helion, Gliwice.
2. Tomera M. (2002). *Rachunek operatorowy Laplace'a - Matlab.* Materiały pomocnicze do nauki teorii sterowania.

