

Modele układów dynamicznych - laboratorium

SIMULINK - wprowadzenie

SIMULINK

Simulink to przybornik (*toolbox*) pakietu Matlab przeznaczony do symulacji układów dynamicznych w trybie graficznym.

Simulink to interfejs graficzny do:

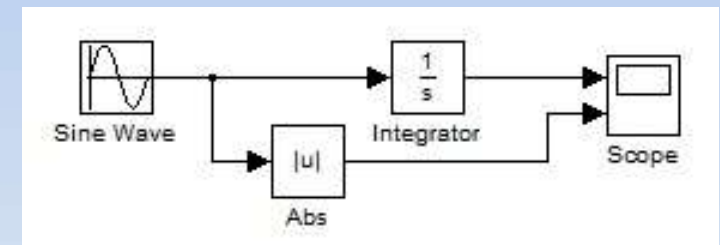
- konstrukcji schematu badanego modelu dynamiki,
- wyboru i konfiguracji algorytmu obliczeniowego (algorytmu całkowania, tzw. solvera),
- uruchamiania symulacji i rejestrowania wyników,
- uruchamiania innych narzędzi (*toolbox'ów*), na przykład do projektowania układów regulacji

Schemat badanego układu (modelu)

Budowanie modelu polega na przeciąganiu bloków z bibliotek Simulinka do obszaru roboczego i łączenie ich w jedną strukturę za pomocą połączeń liniowych.

Bloki w obszarze roboczym można:

- wybierać,
- kopiować,
- usuwać,
- obracać o kąt 90 (Ctr-R),
- edytować - wybór lub zamiana funkcji, parametrów warunków początkowych, zmiana nazwy, itd...



Badanie układów dynamiki w trybie graficznym

1. Schemat modelu

- konstruowany z bloków dostępnych w bibliotekach Simulinka,
- zapamiętywany w pliku *.mdl,

2. Wartości zmiennych i parametrów modelu

- wprowadzane w blokach Simulinka,
- podawane w postaci:
 - wartości,
 - zmiennych odczytywanych z przestrzeni roboczej Matlaba,
 - wyrażeń matematycznych na wartościach i zmiennych,

3. Sterowanie symulacją

- poprzez menu w oknie edycji schematu,
- w trybie wsadowym (za pomocą funkcji sim),

4. Wyniki symulacji

- wykresy graficzne: wykres czasowy, wykres fazowy)
- przestrzeń robocza Matlaba
- plik wyjściowy

Badanie układów dynamiki w trybie graficznym

Wariant 1 (najprostszy):

1. Przygotowanie schematu ze wszystkimi danymi
2. Uruchomienie poprzez menu
3. Obserwacja na wykresach (np. blok Scope)

Wariant 2 (zaawansowany):

1. Przygotowanie sparametryzowanego schematu
 - jako parametry bloków używane są zmienne a nie wartości
 - zastosowane są bloki do rejestrowania wyników symulacji
2. Przygotowanie skryptu zawierającego:
 - definicje wszystkich zmiennych
 - uruchomienie symulacji
 - generowanie wykresów na podstawie wyników z symulacji
3. Uruchomienie skryptu realizującego program bada

Podstawowe biblioteki Simulinka

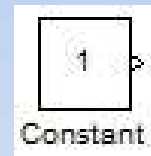
➤ Sources

- funkcje wymuszające, czyli bloki z których sygnały tylko wypływają

Wybrane parametry:

Wymuszenie stałe

liczba, zmienna, wyrażenie



- Constant value

Wymuszenie skokowe

w chwili t_s sygnał o wartości początkowej w_p zmienia się do wartości końcowej w_k



- Step time (t_s)
- Initial value (w_p)
- Final value (w_k)

Generator sinus



- Amplitude (=1),
- Bias (=0)
- Frequency (rad/s) (=1)
- phase (rad) (0=)

Zegar

Podaje bieżący czas symulacji
(momenty w których wykonywane są obliczenia)



Podstawowe biblioteki Simulinka

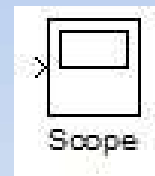
➤ *Sinks*

– *końcówki układów, do których informacja tylko doływa*

Wybrane parametry:

Oscyloskop

jedno wejście na które można podać pojedynczy sygnał lub wektor



- General – ilość osi, zakres czasu
- Data History – wielkość bufora danych

Zbieranie danych

sygnał wejściowy zostanie zapamiętany w zmiennej o podanej nazwie (xx)



- Variable name (xx)
- Save format – najprostszy **Array**

Wyświetlacz cyfrowy

pokazuje wartość pojedynczego sygnału lub wektora



- Format (np. short, long, ...)
- Decimation

Podstawowe biblioteki Simulinka

➤ *Continuous*

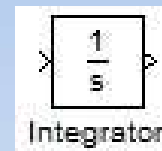
- *elementy liniowe ciągłe*

Wybrane parametry:

Blok całkujący

całkuje z sygnał wejściowy

Parametr *wp* określa wartość na wyjściu w chwili zero



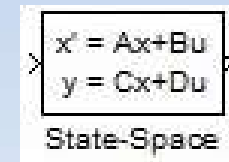
- Initial condition (*wp*)

Równania stanu

równania zdefiniowane za pomocą macierzy A, B, C, D

Na wyjściu bloku dostępny jest tylko wektor sygnałów wyjściowych *y*.

Można podać wektor wartości początkowych *wp* zmiennych stanu *x*



- A, B, C, D
- *wp*

Transmitancja

definiowana w postaci funkcji wymiernej

(stopień licznika < stopień mianownika)



- Numerator coefficients, np.: [1]
- Denominator coefficients, np.: [1 1]

Człon opóźniający

blok opóźniający sygnał wejściowy o stałą wartość (*T0*)



- Time delay (*T0*)
- Initial output (*wp*)
- Initial buffer size

Podstawowe biblioteki Simulinka

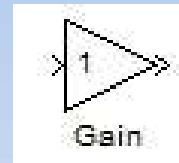
➤ *Math Operations*

– *operacje matematyczne*

Wybrane parametry:

Wzmocnienie

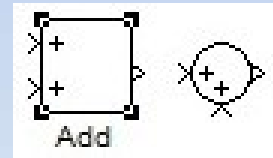
mnożenie sygnału przez wartość
(liczba, zmienna, wyrażenie)



- Gain

Suma

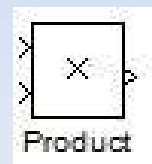
dodawanie zadanej ilości sygnałów
lub dodawanie/odejmowanie sygnałów
(zgodnie z podanym wektorem działań)



- Icon shape (rectangular, round)
- List of signs (np.: +/-)

Iloczyn

mnożenie zadanej ilości sygnałów
lub mnożenie/dzielenie sygnałów
(zgodnie z podanym wektorem działań)



- Number of inputs
(lub lista znaków, np.:*/)

Podstawowe biblioteki Simulinka

➤ *User-Defined Function*

– *wyrażenia matematyczne (liniowe/nieliniowe)*

Wybrane parametry:

Wyrażenie matematyczne

na sygnałach z wektora wejściowego u i zmiennych



- expression,
np. $\sin(u(1)*\exp(2.3*(-u(2))))$
(na wejście można podać sygnał lub wektor)

Podstawowe biblioteki Simulinka

➤ *Signal Routing*

– *przekazywanie sygnałów*

Wybrane parametry:

Multiplexer

łączenie sygnałów w wektor



- Number of input – ilość lub wektor, np. [2,1,3]
- Display option (none, bar, signals)

Demultiplexer

rozdzielanie wektora sygnałów



- Number of input – ilość lub wektor, np. [2,1,3]
- Display option (none, bar, signals)

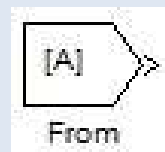
Etykieta „Go to”



- Go Tag – nazwa sygnału (domyślnie A)
- Tag Visibility – zasięg

Etykieta „From”

wirtualne połączenia



- Tag – nazwa sygnału (domyślnie A)

Podstawowe biblioteki Simulinka

➤ **Port & Subsystem**

– *złożone schematy*

Wybrane parametry:

Podsystem

możliwość zgrupowania części schematu w jeden blok



Ilość wejść i wyjść zależy od portów wejściowych i wyjściowych zawartych w bloku. Nazwy portów są widoczne na bloku. Schemat można sparametryzować (Mask Subsystem)

Port wejściowy



- Port number

Można zmieniać podpis pod blokiem (pojawi się automatycznie na bloku Subsystem)

Port wyjściowy



- Port number

Można zmieniać podpis pod blokiem (pojawi się automatycznie na bloku Subsystem)

Zasady konstrukcji schematów

Przykład – równanie liniowe n-tego rzędu

Liniowe równanie różniczkowe

ze skokowym sygnałem wymuszającym

i zadanymi warunkami początkowymi

$$a\ddot{x}(t) + b\dot{x}(t) + cx(t) = u(t)$$

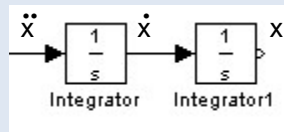
$$u(t) = \begin{cases} u_0 & \text{dla } t < t_0 \\ u_0 + d_u & \text{dla } t \geq t_0 \end{cases}$$

$$\dot{x}(0) = x_{10} \quad x(0) = x_0$$

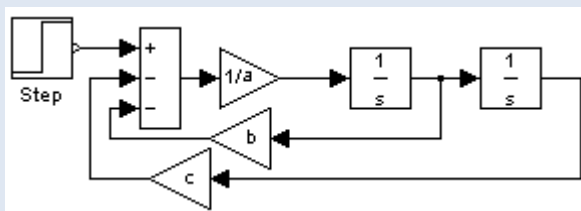
1) Przekształcić równanie – po lewej stronie zostaje najwyższa pochodna zmiennej x

$$\ddot{x}(t) = \frac{1}{a} (u(t) - b\dot{x}(t) - cx(t))$$

2) Wprowadzić „łańcuch” bloków całkujących



3) Narysować schemat blokowy na podstawie równania na najwyższą pochodną

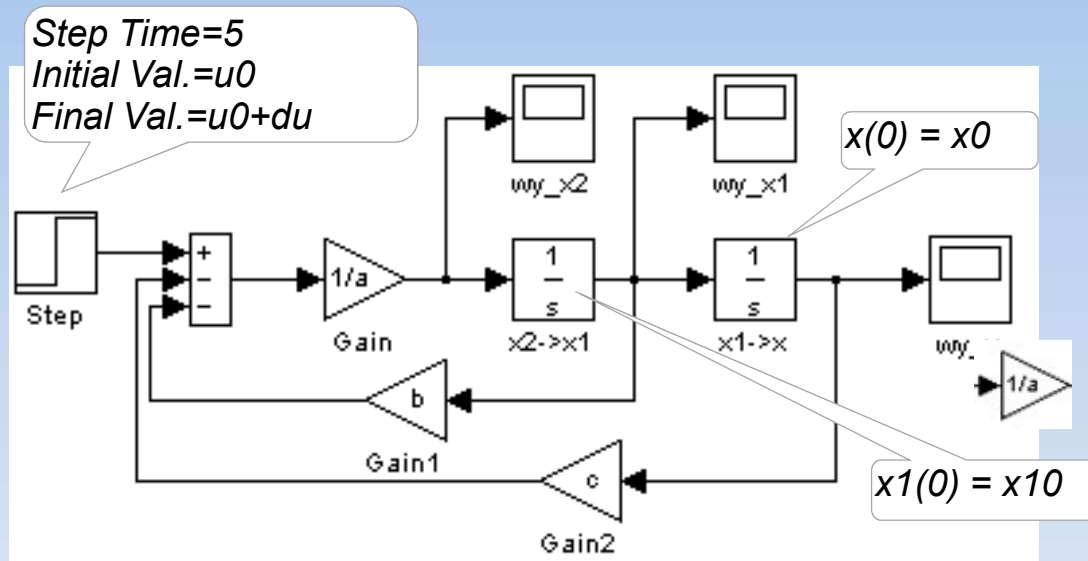


4) Wpisać parametry bloków

5) Dodać bloki wyjściowe (np. oscylosop)

Zasady konstrukcji schematów

Przykład – równanie liniowe n-tego rzędu



$$a\ddot{x}(t) + b\dot{x}(t) + cx(t) = u(t)$$

$$u(t) = \begin{cases} u_0 & \text{dla } t < t_0 \\ u_0 + d_u & \text{dla } t \geq t_0 \end{cases}$$

$$\dot{x}(0) = x_{10} \quad x(0) = x_0$$

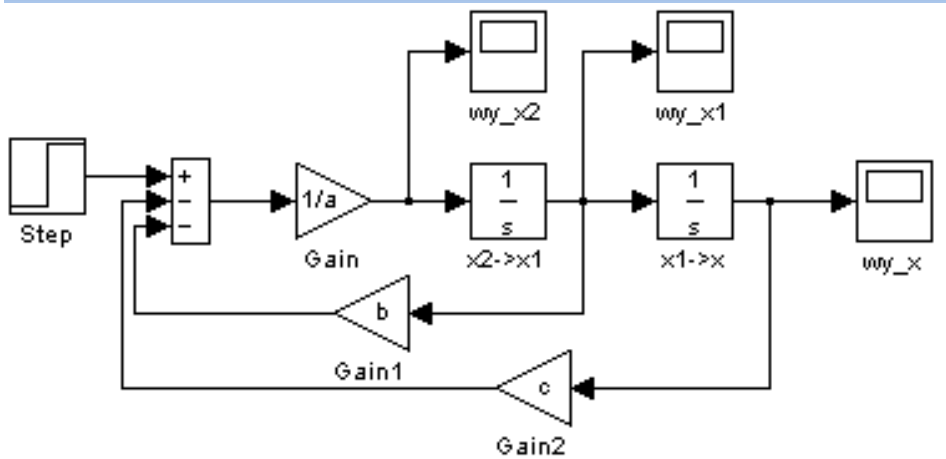
$a = 2; b = 8; c = 2;$

$u_0=2; du=0.5;$ %skok wartości na wejściu
 $t_0=5;$ %czas skoku

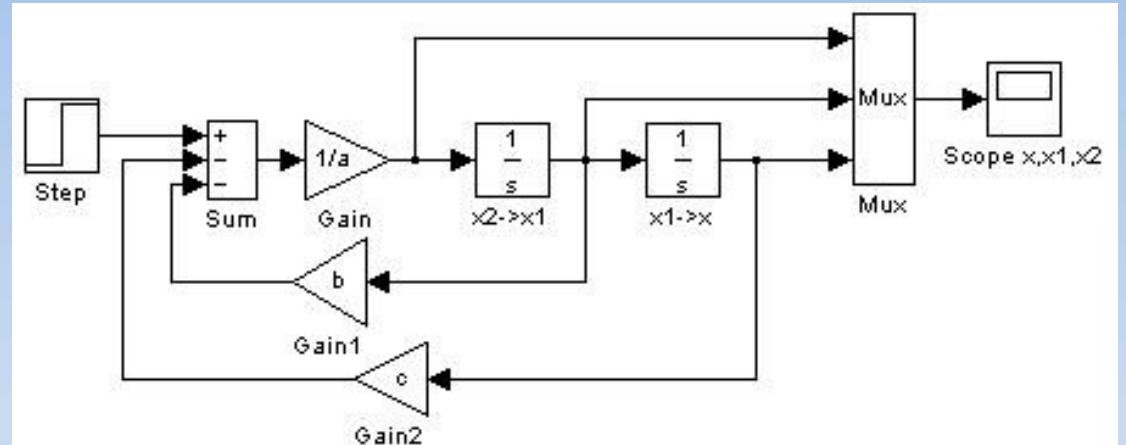
$x_0=u_0/c; x_{10}=0;$ %warunki początkowe
 %(stan równowagi)

Zasady konstrukcji schematów

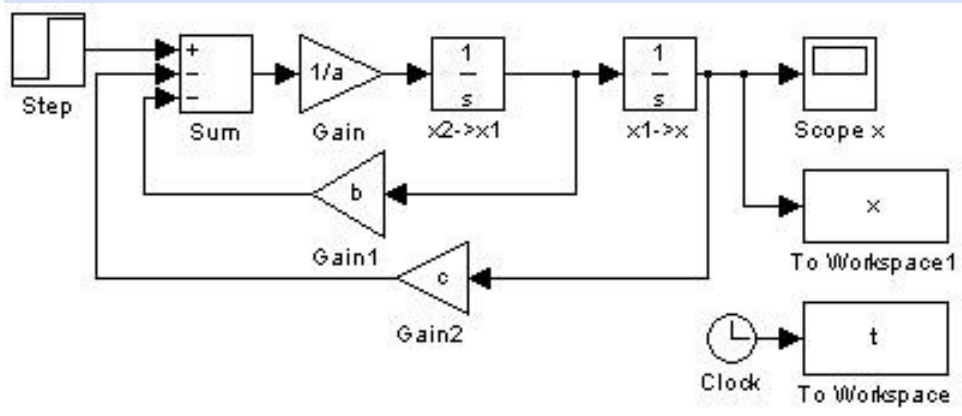
Prezentacja wykresów:



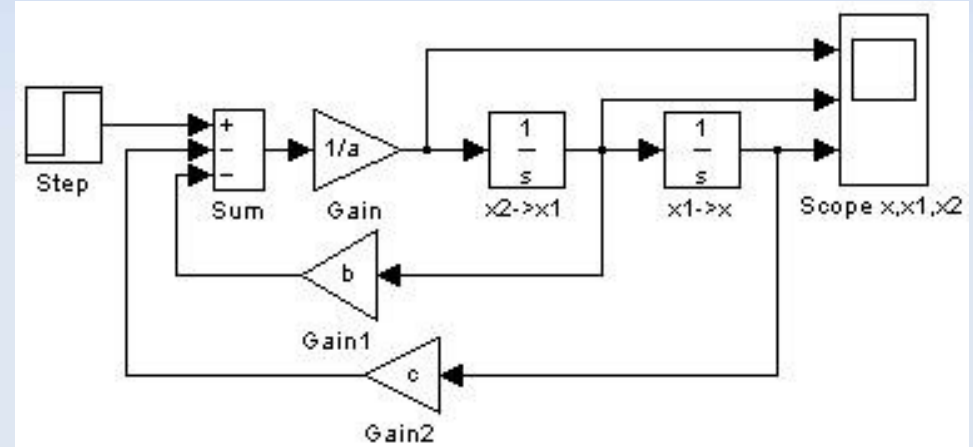
- w oddzielnych oknach



- w jednym układzie współrzędnych, różnymi kolorami.



- bieżąca prezentacja i rejestracja danych

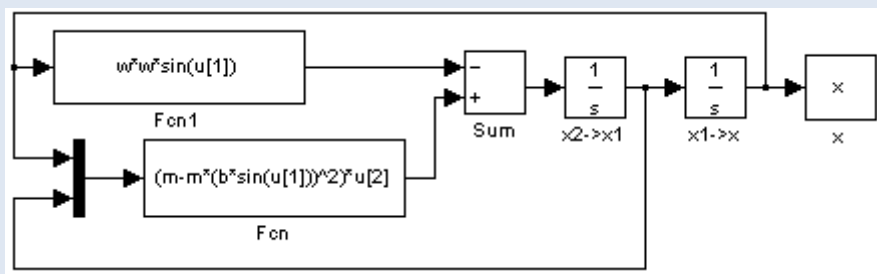
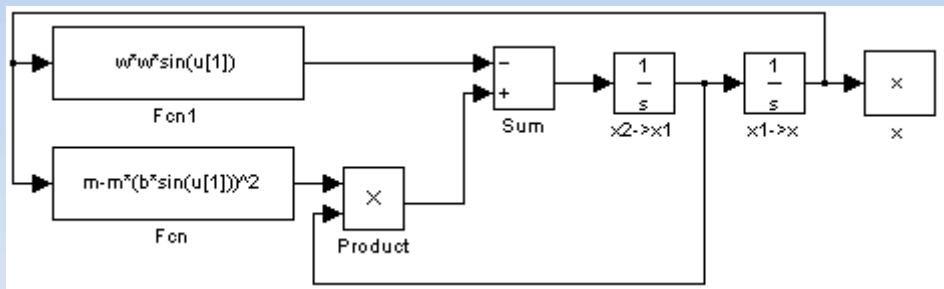


- w oddzielnych układach współrzędnych (analogicznie jak subplot)

Zasady konstrukcji schematów

Przykład - równanie nieliniowe n-tego rzędu

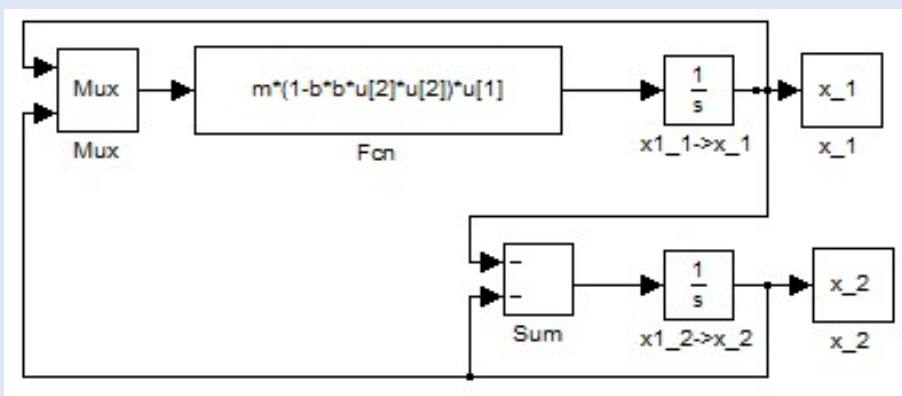
$$\ddot{x} - m(1 - b^2 \sin^2 x)\dot{x} + w^2 \sin x = 0 \quad \Longrightarrow \quad \ddot{x} = m(1 - b^2 \sin^2 x)\dot{x} - w^2 \sin x$$



Zasady konstrukcji schematów

Przykład – układ równań (liniowe/nielinowe)

$$\begin{cases} \dot{x}_1 - m(1 - b^2 x_2^2)x_1 = 0 \\ \dot{x}_2 + x_1 + x_2 = 0 \end{cases} \quad \Longrightarrow \quad \begin{cases} \dot{x}_1 = m(1 - b^2 x_2^2)x_1 \\ \dot{x}_2 = -x_1 - x_2 \end{cases}$$



Parametry algorytmu obliczeniowego

Rozwiązanie równania różniczkowego po zbudowaniu modelu i ustawieniu wartości początkowych (menu *Simulations*).

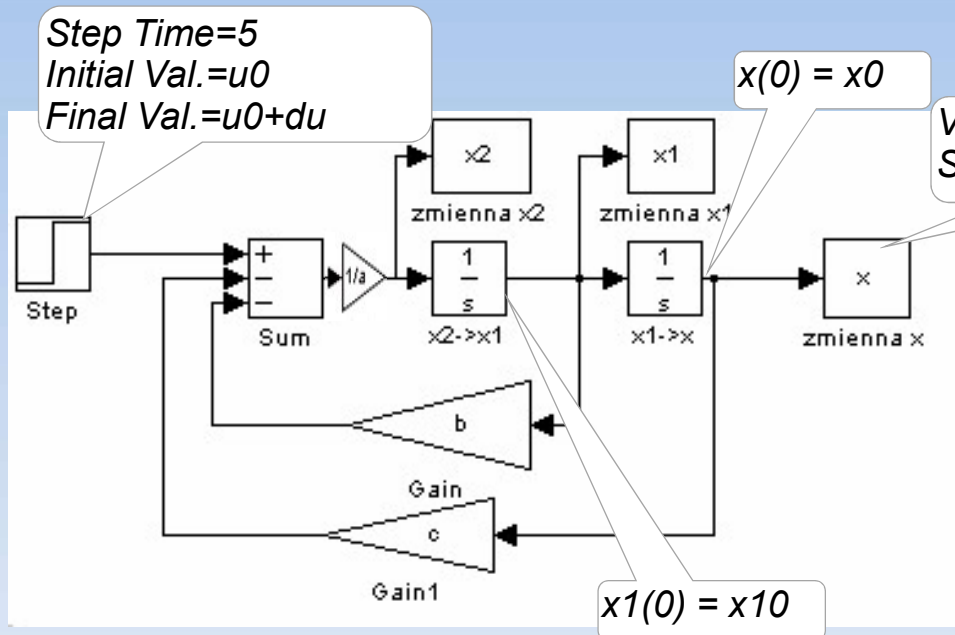
Wybór algorytmu całkowania i parametrów procesu w oknie *Configuration parameters*:

- czas początkowy (*Start time*) analizy,
- czas końcowy (***Stop time***) analizy,
- minimalny krok całkowania (*Min step size*),
- maksymalny krok całkowania (***Max step size***),
- początkowy krok całkowania (*Initial step size*),
- typ algorytmu: stało- lub zmiennokrokowy (*Type*),
- rodzaj algorytmu całkowania (*Solver*).

Uruchamianie w trybie wsadowym

Przykład

$$a\ddot{x}(t) + b\dot{x}(t) + cx(t) = u(t)$$



Odpowiedź na wymuszenie skokowe (skok u od u0 o wartość du)

```

tytul = 'Wpływ parametru b';
model = 'wzor2';           %nazwa pliku ze schematem
czas = 50;                 %czas trwania symulacji (patrz: sim)
kolor = 'rgbcmy';        %red,green,blue,cyan,magenta,yellows
a=3;c = 2;
tab_b = [1 2 3];         %tablica parametru b
imax = size(tab_b, 2);   %ilość parametrów (ograniczenie pętli)
u0=0; du=1;              %parametry skoku (w bloku Step)
x10=0;                   %war.początkowy x1(0)
x0 =u0/c;                %war.początkowy x(0)
fig1=figure, hold on, grid on, ylabel(strcat(tytul, ' - x'))
fig2=figure, hold on, grid on, ylabel(strcat(tytul, ' - x1'))
for i=1:imax
    form = kolor(i);      % format linii
    b = tab_b(i);         % kolejna wartość parametru b
    [t] = sim(model, czas); % „[t]=” zamiast bloku „Clock”
    figure(fig1), plot(t, x, form); % dane z bloków „To Workspace”
    figure(fig2), plot(t, x1, form); % jw
end
    
```