

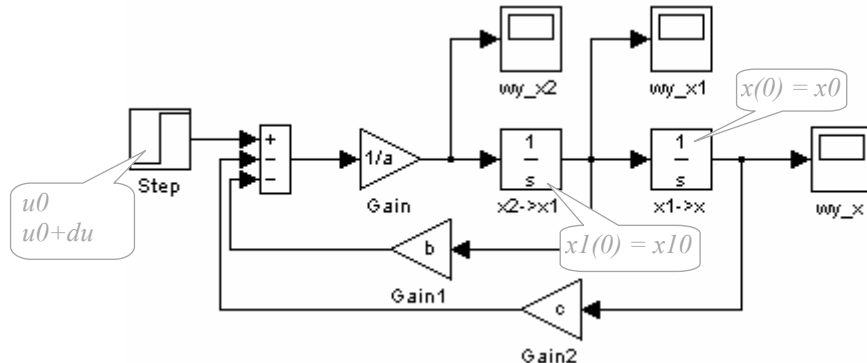
1. Metody definicji modeli i symulacji

1.1 Rozwiązywanie równania różniczkowego – model graficzny¹

Modele graficzne to aplikacje równania $a\ddot{x} + b\dot{x} + cx = u \Rightarrow \ddot{x} = \frac{1}{a}(u - b\dot{x} - cx)$

1° Schemat z blokiem całkującym i bieżącą prezentacją wykresów w oddzielnych oknach

Na schemacie zastosowano bloki Integrator oraz kilka bloków Scope.



Rys. 1-1. Wykresy w oddzielnych oknach (plik „wzor1.mdl”)

Na schemacie wykorzystano następujące zmienne:

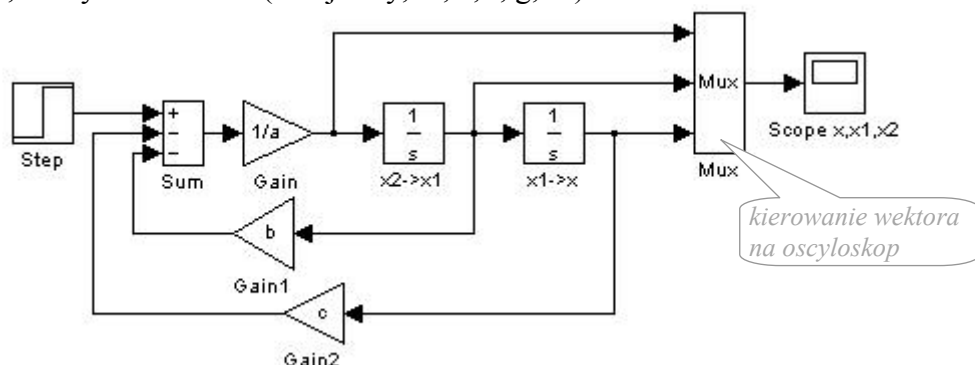
a = 2; b = 8; c = 1;	%obliczenie dowolnego stanu ustalonego
u0=2; du=1; %skok na wejściu	u0=2;
x0=0; x10=0; %war.początkowe	x0=u0/c; x10=0;

Na pasku tytułowym okna związanego z blokiem Scope pojawia się etykieta bloku (podpis przy bloku). Domyślną etykietę można zmienić (np. na nazwę/opis zmiennej) i wykorzystać do identyfikacji wyświetlanych wykresów. Pewne właściwości okna wykresów można ustawić wybierając ikonę „Properties” (która otwiera okno właściwości bloku).

Prezentacja danych w bloku Scope odbywa się tylko wtedy, gdy okno wykresów zostało otwarte przez operatora. Jeśli ilość wyświetlanych wartości przekroczy pojemność bufora bloku, to zawartość bufora zostanie przesunięta (widoczne będą ostatnie wartości). Rozmiar bufora można ustawić w oknie właściwości bloku (na zakładce „Data history” należy wyłączyć ograniczenie lub ustawić większą wartość).

2° Schemat z blokiem całkującym i bieżącą prezentacją wykresów we wspólnym oknie

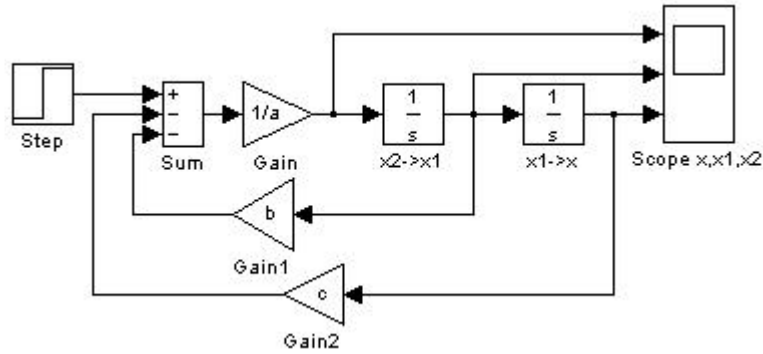
Blok Scope ma domyślnie jedno wejście ale można na nie skierować wektor sygnałów, tworzony za pomocą bloku Mux (Rys. 1-2). Wykresy są rysowane w jednym układzie współrzędnych, różnymi kolorami (kolejno: y, m, c, r, g, b¹)



Rys. 1-2. Wykresy w jednym oknie (plik „wzor1b.mdl”)

Aby zmienić ilość wejść bloku Scope (Rys. 1-3) należy w oknie właściwości bloku na zakładce „General” wpisać ilość osi (np. 3) – każde wejście będzie wyświetlane na oddzielnym wykresie (analogicznie jak funkcja subplot).

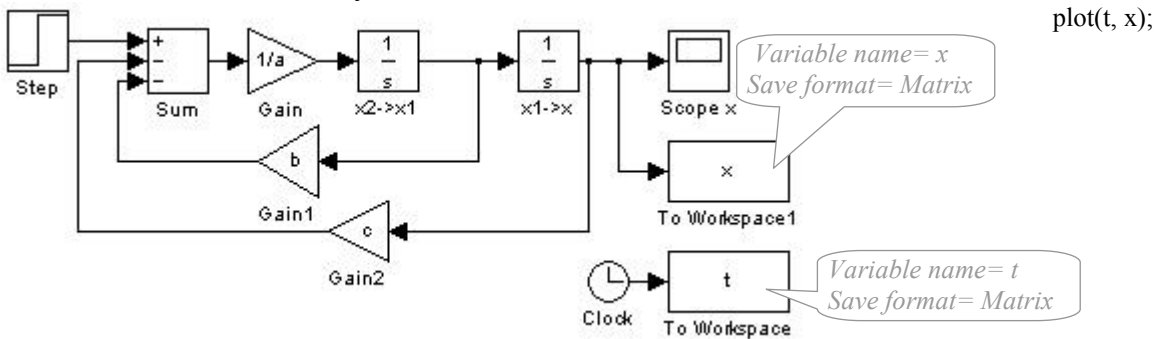
¹ yellow, magenta, cyan, red, green, blue



Rys. 1-3. Oddzielne wykresy w jednym oknie (plik „wzor1bb.mdl”)

3° Schemat z blokiem całującym, bieżącą prezentacją wykresów i rejestracją danych

Przebieg symulacji można zarejestrować a następnie odtworzyć go za pomocą funkcji plot. Do rejestracji można wykorzystać blok To workspace (Rys. 1-4), w którym należy zdefiniować nazwę zmiennej i jej format, np. struktura lub wektor. Stosowanie najprostszego formatu wektora danych (Matrix) wymaga aby zarejestrować również wektor czasu – na schemacie zastosowano w tym celu źródło Clock.



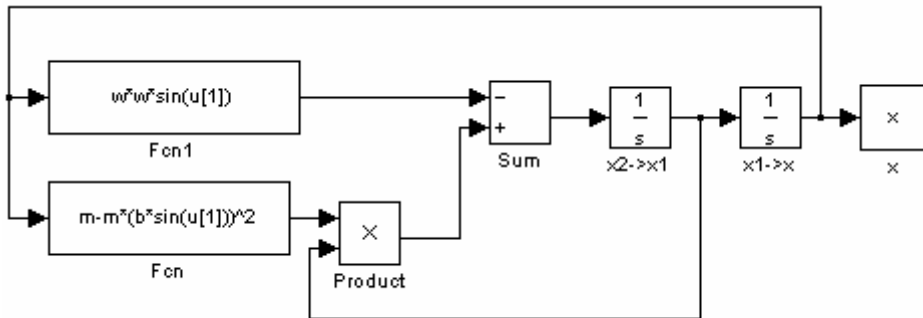
Rys. 1-4. Bieżąca prezentacja wykresu i rejestracja danych (plik „wzor1c.mdl”)

Na schemacie zastosowano zarówno bieżącą prezentację wyników jak i ich rejestrację. Rozwiązanie zawiera pewien nadmiar, ponieważ blok Scope także umożliwia rejestrację danych jeśli w oknie właściwości bloku na zakładce „Data history” zostanie włączona opcja „Save data to workspace”), podana nazwa i format zmiennej.

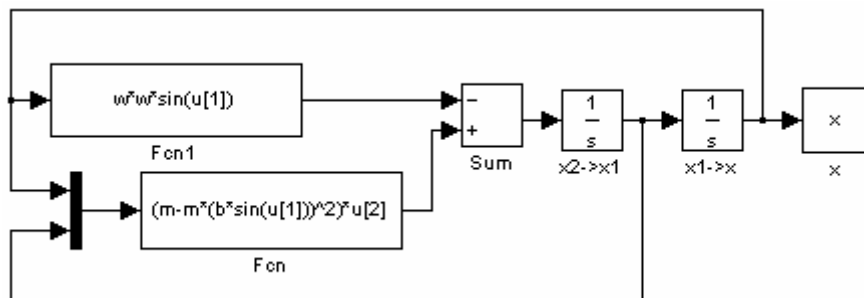
1.2 Przykład rozwiązywania równania różniczkowego nieliniowego

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

Wariant 1

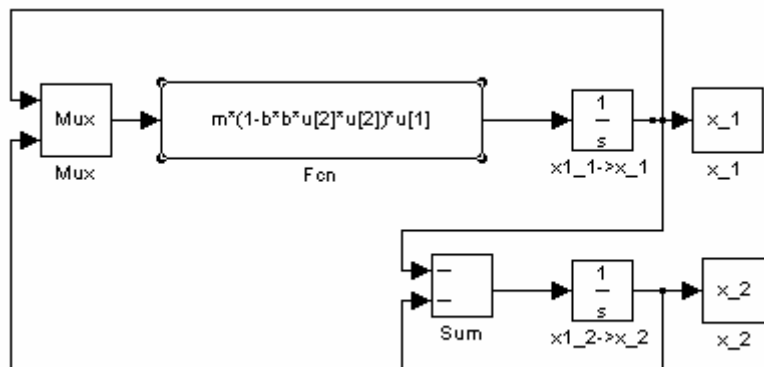


Wariant 2



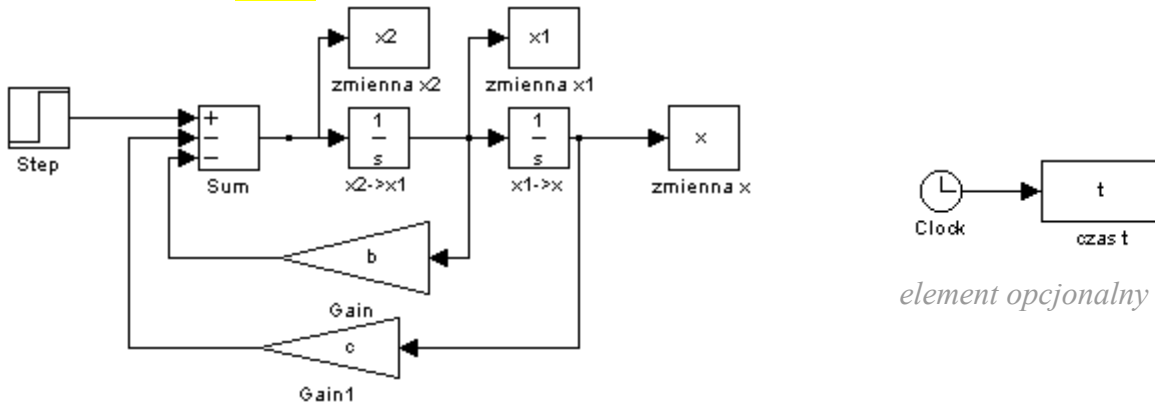
1.3 Przykład rozwiązywania układów równań różniczkowych

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



1.4 Rozwiązywanie równania różniczkowego – model graficzny i skrypt

Model graficzny „wzor2.mdl” + skrypt dla rozwiązania $\ddot{x} + b\dot{x} + cx = u$



```

tytul = 'Wpływ parametru b';
model = 'wzor2';
opcje = simget(model); %opcjonalne (do zmiany parametrów)
opcje = simset('MaxStep', 1, 'RelTol', 1e-6); %opcjonalne (do zmiany parametrów)
czas = 1000; %czas trwania symulacji (zamiast czasu z modelu)

kolor = 'rgbcmy'; %red,green,blue,cyan,magenta,yellows
c = 1;
tab_b = [1 2 3]; %tablica parametru b
imax = size(tab_b, 2); %ilość parametrów (do pętli)
u0 = 0; du = 1; %parametru skoku (w bloku Step)
x0 = u0/c; %war.początkowy x(0) w końcowym bloku całkującym
    
```

```

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, opcje); % „[t]” zamiast bloku „czas t”
    figure(fig1), plot(t, x, form); % dane z bloków „To Workspace”
    figure(fig2), plot(t, x1, form); % jw
end
    
```

Zmiana skali, np. połowa skali pionowej:

```

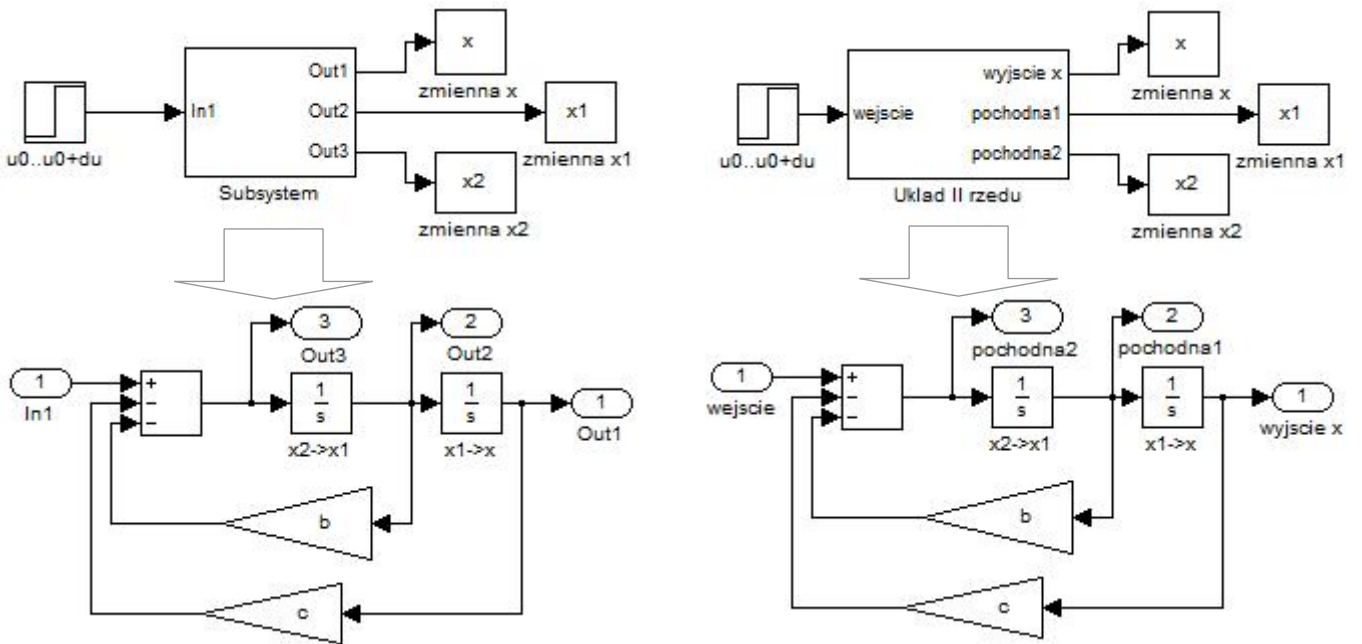
figure(fig2);
osie = axis;
osie(4)=osie(4)/2
axis(osie);
    
```

Wykres:

- ❑ całego przebiegu zmiennej x: `plot(t, x, 'r')`
- ❑ ½ przebiegu: `ile = round(size(t)/2), plot(t(1:ile), x(1:ile), 'm')`
- ❑ wartości stałej: `plot(t, 11, 'm')`
- ❑ wyrażenia: `plot(t, x2+b.*x1+c.*x)`

1.5 Grupowanie i parametryzowanie (blokowanie, maskowanie)

Model graficzny „wzor2.mdl” – po wykonaniu operacji „Edit/Create Subsystem”



Podpisane porty In/Out i Subsystem

Parametryzowanie („Edit/Mask Subsystem”²) bloku Subsystem „Układ II rzędu”

#	Prompt	Variable	Type	Evaluate	Tunable	Tab name
1	Parametr c	c	edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Parametr b	b	edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	war.początkowy x(0)	x0	edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	war.początkowy x1(0)	x10	edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

zmienna wewnętrzna bloku Subsystem (parametr)

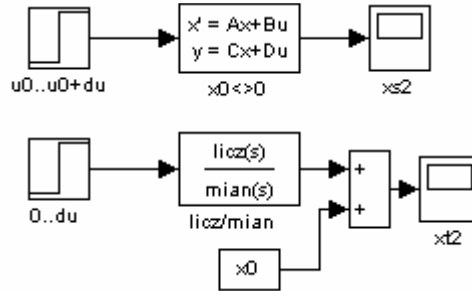
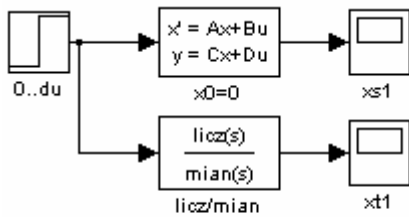
wartość parametru (liczba, zmienna Matlaba, wyrażenie)

²W wer. 5.3: operacja „Edit/Edit Mask”; zakładki w oknie edytora: Icon, Initialization (zawiera parametry), Documentation

1.6 Równania stanu i transmitancje

Modele graficzne:

- symulacja od zerowego stanu ustalonego
- symulacja od dowolnego stanu ustalonego



Realizacja zadania w skrypcie

```
%model: x'=a*x+b*u
```

```
%Badanie: odpowiedź na skok du
```

```
a=-2;
```

```
b=1;
```

```
du=1;
```

```
%Od stanu u0=0 (->x0=0) skok o 1
```

```
%Równania stanu: A=a, B=b, C=1, D=0
```

```
A=[a];
```

```
B=[b];
```

```
C=[1];
```

```
D=0;
```

```
ModelSS=ss(A,B,C,D);
```

```
%transmitancja: b / (s-a)
```

```
licz=[b];
```

```
mian=[1 -a];
```

```
ModelTF=tf(licz, mian);
```

```
step(ModelSS, 'r', ModelTF, 'g', 10)
```

```
%skok od dowolnego stanu ustalonego: 0=a*x0+b*u0
```

```
u0=3;
```

```
%-----
```

```
x0=-b*u0/a;
```