7.4 Parametry i uruchomienie symulacji z okna edycji schematu

Menu w oknie edycji schematu zawiera pozycje i ikony pozwalające bezpośrednio zdefiniować parametry i uruchomić symulację, w tym:

Simulation – Symulacja – parametry i uruchamianie symulacji:

- Setup - Ustawienia – parametry symulacji
- Execution trace and Debug - Wykonaj śledź i analizuj
- Set Context - Ustaw kontekst
- Compile - Skompiluj
- Modelica initialize - Inicjalizacja Modelica
- Start – uruchomienie symulacji
- Stop – zatrzymanie symulacji

Simulation – parametry i uruchamianie symulacji:

- Start – uruchomienie symulacji
- Stop – zatrzymanie symulacji
- Configuration Parameters – parametry symulacji
  - Normal
  - Accelerator
  - Rapid Accelerator

Przed uruchomieniem symulacji należy zainicjować zmienne, które są używane na schemacie (np. uruchomić skrypt zawierający definicje/obliczenia zmiennych) oraz określić parametry symulacji (przynajmniej czas trwania).

7.4.1 Parametry symulacji

Parametry symulacji określają typ oraz parametry algorytmów obliczeniowych (ang. solver). Możliwość dobrania parametrów symulacji pozwala realizować obliczenia szybko i dokładnie. Zazwyczaj (w typowych zadaniach) wartości domyślnie spełniają swoje zadanie i poza czasem obliczeń nie ma potrzeby wprowadzania zmian. Parametry symulacji są zapamiętywane pliku wraz ze schematem.

Wybrane parametry symulacji, które można odczytać/ustawić w menu Simulation:

- **Final integration time – Ostateczny czas integracji = 1.0E05**
- Realtime scaling – Skalowanie czasu rzeczywistego = 0.0E00
- Integrator absolute tolerance - Integrator absolutnej toleracji = 1.0E-04
- Integrator relative tolerance - Integrator względnej toleracji = 1.0E-06
- Tolerance on time – Toleracja w czasie = 1.0E-10
- Max interation time interval – Maks.przedział czasu całkowania = 1.00001E05
- Solver 0 (CVODE) - 100 (IDA) = 0 (CVODE)
- Max step size (0 means no limit) – Maks. rozmiar kroku (0=brak limitu) = 0
- Set context - Ustaw kontekst – Definicja zmiennych kontekstowych

Uwagi:

- Domyślny czas symulacji (Final integration time) jest bardzo długi (10^5). Na początek lepiej zadać krótszy czas
- W parametrach symulacji nie można wpisywać zmiennych
- W symulacji wykorzystywany jest algorytm stałokrokowy (jedyny dostępny solver 0)

Simulation time:

- Start time = 0.0 **Stop time = 10.0**

Solver options:

- Type = Variable-step Solver: ode45
- Max step size = auto Relative tolerance = 1e-3
- Min step size = auto Absolute tolerance= auto
- Initial step size = auto

....

Uwagi:

- Domyślny czas symulacji (Stop time) jest dość krotki (10) i najprawdopodobniej trzeba go będzie wydłużyć.
- W parametrach symulacji można wpisać także zmienne
- Domyślnie stosowany jest zmiennokrokowy algorytm i to zapewnia zazwyczaj dobrą symulację. Ale jeśli czas symulacji jest długi, to należy określić maksymalny krok obliczeń (Max step size) – domyślnie jest on dobierany automatycznie jako 0.001 czasu symulacji (i zwiększa się przy wydłużaniu czasu)

7.4.2 Uruchomienie i zatrzymanie symulacji

Symulacja uruchamiana w oknie edytora nie wymaga zapamiętania schematu w pliku. Symulacja jest zwykle wykonywana w „pełnym biegu” ale może być też uruchamiana w trybie śledzenia.

Simulation\Start – Symulacja\Start
– uruchomienie symulacji w ustawionym trybie

Simulation\Execution trace and Debug - Symulacja\Wykonaj śledź i analizuj
– ustawienie trybu uruchomienia (poziom śledzenia)

Simulation\Start
– uruchomienie symulacji w pełnym biegu

Tools\Simulink Debugger
– uruchomienie symulacji w trybie debug

7.5 Parametry i uruchomienie symulacji w trybie wsadowym

Schemat zapisany w pliku można uruchomić w trybie wsadowym (batch mode) za pomocą funkcji wywołanej z głównej konsoli programu lub w skrypcie. Najprostsze wywołania funkcji są następujące:

scicos_simulate(scs_m);
gdzie: scs_m - struktura opisująca schemat

sim(*schemat*);
gdzie: *schemat* - nazwa pliku zawierającego schemat

Zmienna scs_m to zmienna systemowa, tworzona i aktualizowana po wczytaniu i uruchomieniu symulacji schematu w edytorze xcos.

Sekwencja poleceń przy uruchomienia symulacji bez otwierania edytora xcos jest następująca[1]:

loadXcosLibs();
- załadowanie bibliotek xcos (raz po uruchomieniu Scilaba)

importXcosDiagram('*schemat*.xcos');
- wczytanie schematu (tworzy/aktualizuje zmienną scs_m)
- opcjonalnie można otworzyć schemat xcos schemat1.xcos;

scicos_simulate(scs_m);
- uruchomienie symulacji (według zmiennej scs_m)

Plik ze schematem może znajdować się w kartotece ustawionej w File\SetPath (menu na głównej konsoli). Może to jednak powodować niejednoznaczności gdy w różnych kartotekach znajdują pliki o takiej samej nazwie (np. poprzednie wersje schematów). Bezpieczniej jest pracować na kartotece bieżącej.

Funkcje do uruchamiania symulacji pozwalają ustawić także parametry symulacji. Parametry, które zostaną w ten sposób podane przykrywają parametry zapamiętane wraz ze schematem. Podstawowe zastosowanie to możliwość ustalenia czasu symulacji (np. na 1000 jednostek):

scs_m.props.tf = 1000; - struktura scs_m wczytana wcześniej
scicos_simulate(scs_m);

sim(*schemat, 1000*);

W przypadku Matlaba możliwe jest również odczytywanie wartości wynikowych, na przykład rejestrowanie wektora czasu (zamiast rejestrowania wartości z bloku Clock na schemacie – p.7.3.1):

[t] = sim(*schemat, 1000*);

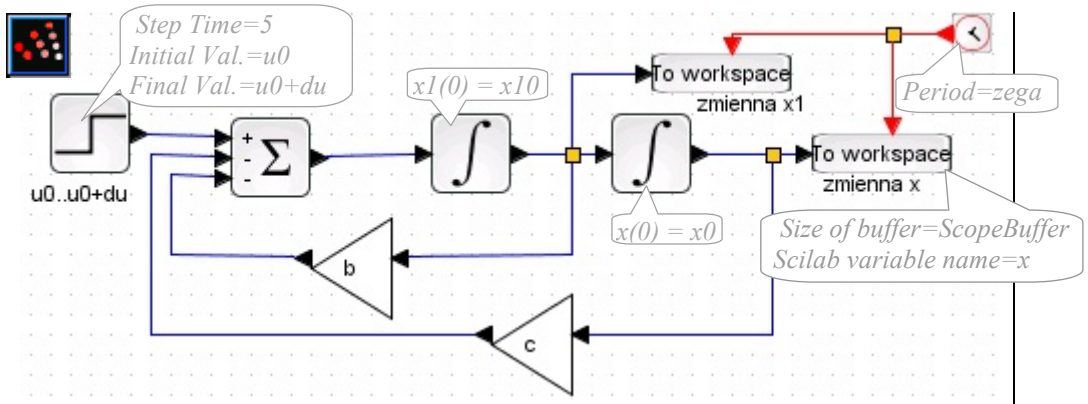
Możliwość uruchamiania symulacji w trybie wsadowym pozwala w prosty sposób zautomatyzować realizację badań.

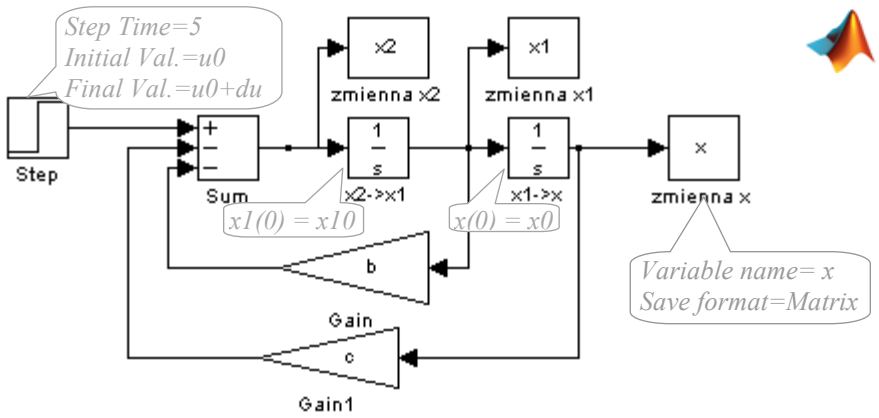
7.6 Przykład automatycznej realizacji programu badań

Zadanie polega na wygenerowaniu i porównaniu odpowiedzi układu $\ddot{x}(t)+b\dot{x}(t)+cx(t)=u(t)$ na wymuszenie skokowe *du* dla różnych wartości parametru *b*. Ponieważ należy wykonać kilka symulacji a ich wyniki dla porównania przedstawić na wspólnym wykresie, warto więc skorzystać z możliwości wsadowego uruchamiania symulacji modelu zapamiętanego w pliku graficznym.

Model układu jest zapisany w pliku graficznym o nazwie „wzór2”. Schemat jest sparametryzowany, to znaczy że parametry poszczególnych bloków są podawane w postaci zmiennych, które zostaną zainicjowane w skrypcie.

[1] Uwaga: Stan według wersji 5.3.3 (we wcześniejszych wersjach może być inaczej)





Cały program badań jest realizowany za pomocą skryptu, który obejmuje zainicjowanie zmiennych, wsadowe uruchomienie symulacji dla różnych wartości parametrów i wygenerowanie wykresów. Poniżej skrypty z zachowaniem oryginalnych kolorów i czcionek edytorów tekstowych.

```
tytul='Wplyw parametru b';
model = 'wzor2.xcos';          //pełna nazwa pliku ze schematem
loadXcosLibs();                //wczytaj biblioteki
importXcosDiagram(model);      //wczytaj model (tworzy zmienną scs_m)
czas=50; zegar=.2;             //czas symulacji i parametr: Clock/Period
ScopeBuffer = czas/zegar;      //parametr To workspace/Size of buffer
scs_m.props.tf=czas;           //czas trwania symulacji (zamiast czasu z pliku)

kolor = ['red','green','blue','cyan','magenta','yellow'];
c = 1;
tab_b = [2  4  8];             //tablica parametru b
imax = size(tab_b,2);          //ilość parametrów (ograniczenie pętli)
u0=2; du=1;                    //parametry skoku (w bloku Step)
x10=0;                         //war.początkowy x1(0)
x0=u0/c;                       //war.początkowy x(0)

figure(1); set(gca(),"auto_clear","off"), xgrid(2), ylabel(strcat([tytul,' - x']))
figure(2); set(gca(),"auto_clear","off"), xgrid(2), ylabel(strcat([tytul,' - x1']))

for i=1:imax
   form = kolor(i);
   b = tab_b(i);
   scicos_simulate(scs_m);                 //czas symulacji: scs_m.props.tf
   figure(1);plot(x.time, x.values,form)   //dane z bloków "To workspace"
   figure(2);plot(x1.time, x1.values,form) //jw
end
```

```
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
c = 1;
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
```