Podstawy Automatyki

Politechnika Poznańska
Katedra Sterowania i Inżynierii Systemów

ĆWICZENIE 3

Wprowadzenie do środowiska Simulink.

Celem ćwiczenia jest poznanie podstawowych cech środowiska symulacyjnego Matlab Simulink.

1 Wprowadzenie

Simulink jest specyficznym modułem rozszerzającym pakietu *Matlab*, służącym do modelowania, symulacji i analizy dynamicznych obiektów rzeczywistych. Wspiera modele liniowe i nieliniowe, dyskretne lub ciągłe w czasie, a także stanowiące hybrydę tych dwóch rodzajów. Można tworzyć także modele zawierające elementy o różnych okresach próbkowania.

Tworzenie modeli rzeczywistych obiektów i ich analiza w *Simulinku* dokonywane są za pośrednictwem interface'u graficznego, w ramach którego system modelowany jest w postaci schematu blokowego, tzn. przez wybór i łączenie bloków, udostępnianych w ramach pakietu *Simulink*. Do standardowego zbioru tych elementów należą biblioteki komponentów wizualizacyjnych, modelujących człony liniowe i nieliniowe oraz generujących sygnały źródłowe.

Bloki, z których tworzy się model systemu, reprezentują elementarne systemy dynamiczne, których działanie i sposób symulacji *Simulink* inherentnie zna. W ogólności blok posiadać może: zbiór wejść, zbiór stanów i zbiór wyjść. Wartości pojawiające się na wyjściach bloku są funkcją czasu i wartości wejściowych oraz zmiennych stanu (o ile dany blok jest w nie wyposażony). Konkretna funkcja zależy od typu użytego bloku.

Kluczowe właściwości wielu standardowych bloków zostały sparametryzowane. Dzięki temu można łatwo określać i dostosowywać cechy użytych komponentów na etapie tworzenia modelu (a niektóre także w czasie trwania symulacji).

W *Simulinku* istnieje możliwość modelowania złożonych systemów w postaci szeregu połączonych ze sobą podsystemów, z których każdy symbolizowany jest pojedynczym blokiem i reprezentuje pewien fragment schematu blokowego, stanowiący funkcjonalną całość. Takie podsystemy można używać jako składniki innych, bardziej złożonych (pod-) systemów, uzyskując w efekcie modele hierarchiczne.

Zadaniem niniejszego ćwiczenia jest zapoznanie się z podstawowymi cechami środowiska *Simulink* oraz sposobu wykorzystywania standardowych składników.

Simulink można uruchomić z poziomu okna poleceń *Matlaba*, wydając polecenie `simulink`, lub klikając na reprezentującą go ikonę w menu narzędziowym głównego okna *Matlaba* (Rys. 1). Po uruchomieniu ukaże się główne okno przeglądarki bibliotek *Simulinka* (Simulink Library Browser), z którego można uzyskać dostęp do wszystkich bloków wykorzystywanych do tworzenia modelu.

1.1 Uruchomić środowisko Simulink.

Rysunek 1: *Ikona uruchamiająca środowisko Simulink.*

2 Tworzenie nowego modelu

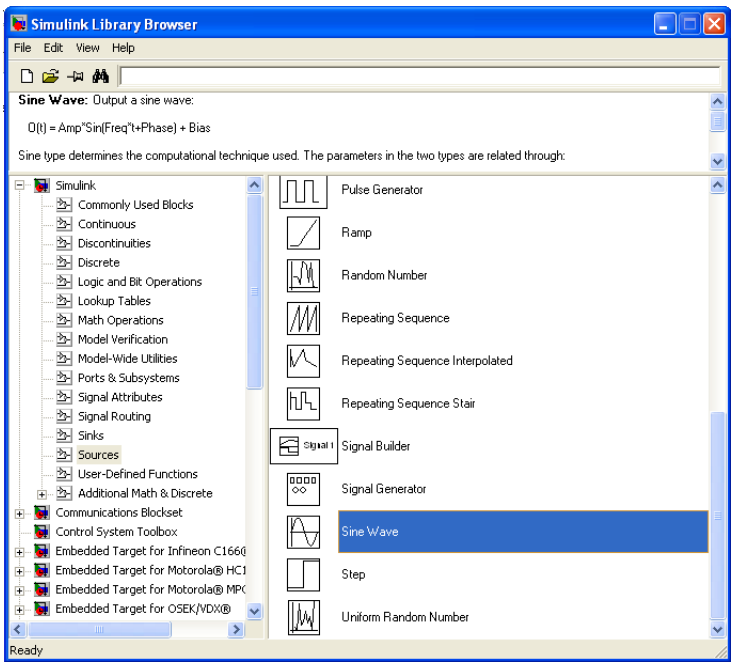
Aby rozpocząć tworzenie nowego modelu z menu File należy wybrać opcję New→Model lub kliknąć na ikonę nowego dokumentu na pasku narzędziowym. Spowoduje to wyświetlenie nowego okna, w obrębie którego tworzony będzie model.

2.1 Wskazówki do tworzenia modelu

Tworzenie prostego modelu składa się z następujących kroków:

- umieszczenie odpowiednich bloków w modelu;
- przestawianie bloków w modelu;
- łączenie bloków ze sobą;
- zapisywanie modelu.

Umieszczenie bloku w modelu wymaga umiejętności odnajdowania bloku w przeglądarce bibliotek. Dostęp do poszczególnych bloków można uzyskać rozwijając i wybierając pozycje w głównym oknie przeglądarki bibliotek Simulinka. Przykładowo blok generujący sygnał sinusoidalny (Sine Wave) znajduje się w bibliotece Sources (Rys. 2). Dokładny opis zawartości poszczególnych bibliotek przedstawiono w następnym podrozdziale.



Rysunek 2: *Widok okna przeglądarki bibliotek Simulink oraz wnętrze biblioteki źródeł.*

Wybrane bloki należy przenieść do tworzonego modelu przeciągając je myszką (metoda drag-and-drop) z okna Simulink Library Browser do okna modelu. Parametry bloków można zmieniać po dwukrotnym kliknięciu ikony bloku.

Zanim bloki zostaną połączone, należy rozmieścić je w taki sposób w oknie modelu, aby połączenie sygnałów było możliwie jak najczytelniejsze. Aby przemieścić blok należy przeciągnąć go myszką (metoda drag-and-drop) lub zaznaczyć blok lewym przyciskiem myszy i przestawiać go za pomocą strzałek na klawiaturze.

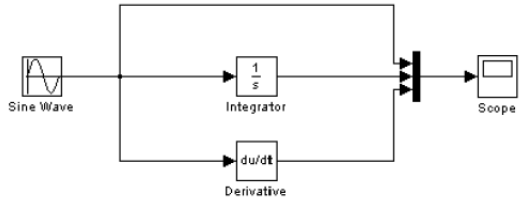
Po prawidłowym rozmieszczeniu bloków modelu, należy utworzyć połączenia pomiędzy nimi, reprezentujące połączenia sygnałów modelu. Należy zwrócić uwagę na symbole dołączone z jednej lub obu stron każdego z bloków. Oznaczają one porty wejściowe lub wyjściowe bloku:

- symbol > wskazujący na blok jest portem wejściowym;
- symbol > wskazujący na zewnątrz bloku jest portem wyjściowym.

Aby utworzyć połączenie pomiędzy dwoma blokami, należy umieścić kursor myszy nad portem wejściowym (lub wyjściowym) bloku. Kursor myszki zamieni się wówczas na znak plus (+). Przy wciśniętym lewym przycisku myszy należy przeciągnąć kursor do portu wyjściowego (lub wejściowego) drugiego bloku. Lewy przycisk myszy należy zwolnić dopiero w momencie, gdy kursor zamieni się na pogrubiony znak plus (+). Połączeń nigdy nie tworzy się pomiędzy portami tego samego rodzaju (wejściowego z wejściowym oraz wyjściowego z wyjściowym).

Jeżeli jeden port wyjściowy ma być połączony z dwoma portami wejściowymi, wówczas możliwe jest utworzenie rozgałęzienia. Należy utworzyć pierwsze połączenie, a następnie wybrać je prawym przyciskiem myszy (lub lewym z wciśniętym klawiszem Ctrl) w celu utworzenia drugiego połączenia. Możliwe jest również wytworzenie połączenia pomiędzy portem wejściowym a środkiem linii łączącej dwa inne bloki.

Przykładowy gotowy model przedstawiono na Rys. 3.



Rysunek 3: *Przykładowy model Simulinka.*

Podczas tworzenia modelu przydatne może okazać się stosowanie znanych skrótów klawiszowych (Ctrl+C, Ctrl+V, Ctrl+Z). Dodatkowo zastosowanie klawisza Del usuwa zaznaczony element modelu. Połączenie klawiszy Ctrl+R obraca orientację zaznaczonego bloku.

Przydatne może okazać się zaznaczanie wielu obiektów modelu naraz. Wystarczy objąć odpowiednie elementy bryłą brzegową. Należy najpierw określić początkowy róg prostokąta, przez odpowiednie ustawienie kursora myszy oraz naciśnięcie lewego przycisku. Przytrzymując lewy przycisk myszy, należy przeciągnąć kursor do przeciwległego rogu prostokąta oraz zwolnić przycisk myszy. W wyniku tych operacji wszystkie elementy znajdujące się wewnątrz prostokąta, zostaną zaznaczone.

Zapis utworzonego modelu odbywa się przez wybranie File→Save z menu okna modelu.

Aby uruchomić symulację modelu bez zmiany parametrów symulacyjnych (dokładny opis w podrozdziale 2.3), należy wybrać opcję menu Simulation→Start, lub nacisnąć przycisk Start Simulation z paska narzędzi.

2.2 Podstawowe składniki środowiska Simulink

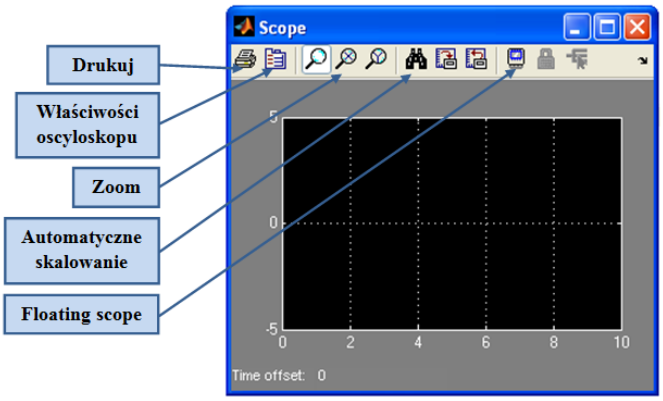
Simulink udostępnia 16 standardowych bibliotek bloków, Tab. 1 opisuje każdą z bibliotek. Podczas laboratoriów z przedmiotu Podstawy Automatyki wykorzystywane będą głównie bloki zawarte w bibliotekach Commonly Used Blocks, Continuous, Sources oraz Sinks. Ważną biblioteką, która może się okazać przydatna przy tworzeniu modeli, jest User-Defined Functions. Blok Matlab Fcn umożliwia wykorzystanie funkcji zapisanej przez użytkownika w postaci m-pliku. Wystarczy w oknie parametrów bloku podać nazwę funkcji oraz liczbę parametrów wyjściowych.

Biblioteka	Opis zawartości
Commonly Used Blocks	Zawiera najczęściej używane bloki, takie jak Constant, In1, Out1, Scope. Wszystkie zawarte w tej bibliotece bloki są częściami innych bibliotek.
Continuous	Zawiera bloki służące do modelowania funkcji liniowych (np.Derivative, Integrator).
Discontinuities	Zawiera bloki, których wyjścia są nieciągłymi funkcjami wejść (np. Saturation).
Discrete	Zawiera bloki, które reprezentują funkcje zdyskretyzowane (np. Unit Delay).
Logic and Bit Operations	Zawiera bloki, które dokonują operacji logicznych lub bitowych (np. Logical Operator Relational Operator).
LookUp Tables	Zawiera bloki, które wykorzystują lookup tables do określenia wyjść na bazie wejść (np.Cosine, Sine).
Math Operations	Zawiera bloki, które dokonują operacji matematycznych (np. Gain, Product, Sum)
Model Verification	Zawiera bloki, które umożliwiają stworzenie samo-sprawdzających się modeli (np. Check Input Resolution).
Model-Wide Utilities	Zawiera bloki, które dostarczają informacji o modelu (np. Model Info).
Ports & Subsystems	Zawiera bloki, które umożliwiają tworzenie podsystemów (np. In1, Out1, Subsystem).
Signal Attributes	Zawiera bloki, które modyfikują atrybuty sygnałów (np. Data Type Conversion).
Signal Routing	Zawiera bloki, które umożliwiają sterowanie przepływem sygnałów (np. Mux, Switch).
Sinks	Zawiera bloki, które wyświetlają lub eksportują wyjścia modelu (np. Out1, Scope).
Sources	Zawiera bloki, które wytwarzają albo importują wejścia modelu (np. Constant, In1, Sine Wave).
User-Defined Functions	Zawiera bloki, które umożliwiają wykorzystanie funkcji stworzonej przez użytkownika (np. Matlab Fcn).
Additional Math & Discrete	Zawiera dwie dodatkowe biblioteki bloków funkcji matematycznych i dyskretnych.

Tablica 1: *Opis zawartości bibliotek Simulinka.*

Często do wyświetlania sygnałów wykorzystywanych w symulacji, wykorzystuje się blok oscyloskopu (Scope). Okno Scope nie jest automatycznie otwierane przy uruchamianiu symulacji w *Simulinku*, mimo tego dane połączone z blokiem są zapisywane. Aby otworzyć okno oscyloskopu należy nacisnąć blok dwukrotnie lewym przyciskiem myszy. Jeżeli okno oscyloskopu zostanie otwarte już po zakończeniu symulacji, uzyskane dane zostaną wyświetlone w odniesieniu do czasu symulacji.

Rysunek 4: *Możliwości połączenia sygnałów z blokiem Scope: a. wyświetlenie wszystkich sygnałów na jednym wykresie; b. wyświetlenie sygnałów na 3 wykresach.*



Rysunek 5: *Okno bloku Scope wraz z opisem paska narzędzi.*

Za pomocą pojedynczego bloku Scope można wyświetlić więcej niż jeden sygnał (lub sygnał postaci macierzowej). Jeżeli wszystkie sygnały podłączymy do jednego portu wejściowego, wówczas dane będą przedstawione na wspólnym wykresie (Rys. 4 a). W takim przypadku, blok Scope przypisuje kolor do każdego z sygnałów w kolejności: niebieski, zielony, czerwony, turkusowy, purpurowy, żółty. Jeżeli do bloku Scope podłączy się więcej niż 6 sygnałów (lub sygnał posiada więcej niż 6 kolumn), blok Scope zaczyna przypisywanie kolorów z powrotem od niebieskiego.

Możliwe jest również zdefiniowanie wielu układów współrzędnych, na których będą prezentowane wykresy w ramach jednego okna oscyloskopu. W tym celu należy wybrać Właściwości oscyloskopu z menu bloku Scope (Rys. 5), a następnie ustawić odpowiednią wartość w polu Number of Samples.

Przydatną opcją jest również Automatyczne skalowanie, które dobiera skalę osi x i y tak, aby widoczny był cały przebieg sygnału.

2.1 Utworzyć model przedstawiony na Rys. 3 wykorzystując następujące bloki:

- Sine Wave - z biblioteki Sources;
- Integrator - z biblioteki Continuous;
- Derivative - z biblioteki Continuous;
- Mux - z biblioteki Commonly Used Blocks;
- Scope - z biblioteki Sinks.

2.2 Zapoznać się z parametrami poszczególnych bloków (okno pojawi się po dwukrotnym wybraniu danego bloku).

2.3 Zapisać model w folderze utworzonym na poprzednich zajęciach.

2.4 Uruchomić symulację i zaobserwować wyniki w bloku Scope.

2.5 Utworzyć funkcję w m-pliku zwracającą wynik sumowania dwóch parametrów wejściowych.

2.6 Stworzyć model w Simulinku wykorzystujący funkcję stworzoną w poprzednim punkcie. Wykorzystać blok Matlab Fcn (biblioteka User-Defined Functions). Jako parametry wejściowe podawać wartości stałe (dwa bloki Constant z biblioteki Sources), natomiast wyjście funkcji przedstawiać za pomocą bloku Display (biblioteka Sinks).

3 Eksportowanie i importowanie zmiennych

W przypadku niektórych modeli, konieczne może okazać się wykorzystywanie wcześniej zadeklarowanych zmiennych, znajdujących się w Workspace Matlaba. Niezwykle ważne jest, aby zmienna, którą chcemy importować była zadeklarowana w przestrzeni roboczej programu Matlab. Eksportowania i importowania zmiennych pomiędzy modelem Simulink a Workspace można dokonać na wiele różnych sposobów.

Pierwszy z nich polega na określeniu odpowiednich zmiennych w zakładce Data Import/Export (po wybraniu opcji Simulation→Configuration Parameters w oknie modelu). Należy pamiętać o tym, że sygnały, które chcemy eksportować należy połączyć z blokiem Out (biblioteka Sinks), a te które chcemy importować, z blokiem In (biblioteka Sources). W najprostszym przypadku, zmienne importowane z Workspace powinny być postaci wektorów kolumnowych.

Drugi sposób wymaga użycia bloków From Workspace oraz To Workspace, znajdujących się odpowiednio w bibliotekach Sources i Sinks. W przypadku importu danych w parametrach bloku należy podać nazwę zmiennych, które mają być importowane. Importowane zmienne również muszą być wcześniej zadeklarowane. Eksport danych wymaga podania nazwy struktury danych, która będzie przechowywać zmienną w Workspace (domyślnie: simout). Aby uzyskać dostęp do wartości zmiennej należy odwołać się do następującego elementu struktury:

```
>> simout.signals.values;
```

Przykładowo, nadając nazwy t oraz u w blokach To Workspace, można stworzyć wykres tych zmiennych za pomocą polecenia:

```
>> plot(t.signals.values, u.signals.values);
```

Dodatkowo, model w Simulinku może wykorzystywać stałe zadeklarowane w Workspace. Wystarczy ich deklaracje w Oknie Poleceń poprzedzić słowem *global*.

3.1 Przekształcić analitycznie równanie różniczkowe do postaci operatorowej (przy zerowych wartościach początkowych):
$\ddot{x} + 3\dot{x} + 2x = u$

3.2 Utworzyć model z wykorzystaniem bloku Transfer Fcn (biblioteka Continuous).

3.3 Przy założeniu, że u jest sygnałem prostokątnym o amplitudzie 1 i okresie równym 10s, przeprowadzić symulację modelu z parametrami domyślnymi.

3.4 Przekształcić schemat blokowy tak, by sygnał wyjściowy x był podłączony do bloku wyjścia (Out). W parametrach symulacji, na zakładce Data Import/Export ustawić eksportowanie czasu i zmiennej wyjściowej do przestrzeni roboczej Matlaba, ewentualnie modyfikując domyślne nazwy zmiennych.

3.5 Przeprowadzić symulację modelu. Wykreślić przebieg sygnału wyjściowego x w czasie z poziomu okna poleceń Matlaba i porównać z przebiegiem uzyskanym w modelu (blok Scope).

3.6 Zmodyfikować schemat blokowy tak, by dodatkowo eksportować sygnał u. Tym razem wykorzystać blok To Workspace (we właściwościach tego obiektu wybrać tablicę – array – jako format eksportowania danych).

3.7 Przeprowadzić symulację modelu. Wykreślić przebieg sygnału wejściowego u w czasie z poziomu okna poleceń Matlaba i porównać z przebiegiem uzyskanym w modelu (w tym celu można na wyjściu bloku generatora sygnału wejściowego, umieścić dodatkowy blok Scope).

3.8 W przestrzeni roboczej Matlaba wygenerować zmienną reprezentującą wektor czasu z całego horyzontu symulacji (np. $t = [0 : 0.1 : 10]^T$). Wygenerować zmienną reprezentującą wartości sygnału wejściowego dla wszystkich chwil symulacji (np. $u = sin(t)$).

3.9 Zmodyfikować schemat blokowy tak, aby importować sygnały wejściowe z przestrzeni roboczej Matlaba za pomocą bloku wejścia (In). W parametrach symulacji, na zakładce Data Import/Export ustawić importowanie wartości czasu i sygnału wejściowego u, ewentualnie modyfikując domyślne nazwy zmiennych na nazwy zadeklarowanych zmiennych.

3.10 Przeprowadzić symulację modelu.

4 Zmiana parametrów symulacji

Przed uruchomieniem symulacji modelu, możliwe jest ustawienie opcji symulacji, takich jak początkowa chwila symulacji, czas trwania, opcje importowania i eksportowania danych (zagadnienie dokładnej opisane w następnym rozdziale) oraz rodzaj całkowania, którego Simulink używa do rozwiązywania modelu w każdej chwili czasowej. Ustawienie czasu symulacji możliwe jest na pasku narzędzi (obok przycisku Stop Simulation). Aby dokonać zmian w opcjach symulacji należy wybrać Simulation→Configuration Parameters w oknie modelu lub nacisnąć Ctrl+E. Podczas laboratoriów z przedmiotu Podstawy Automatyki wykorzystywane będą jedynie zakładki Solver i Data Import/Export.

Solver określa metodę numeryczną do iteracyjnego rozwiązywania równań różniczkowych zwyczajnych. Jest to komponent oprogramowania Simulink, który określa następny krok czasowy symulacji, który jest wymagany, aby uzyskać zdefiniowaną przez użytkownika dokładność. Simulink udostępnia szereg różnych rodzajów solverów, spośród których można wyróżnić solvery o ustalonym kroku próbkowania i zmiennym kroku próbkowania.

Oba typy solverów obliczają następny krok symulacji na bazie sumy aktualnego czasu symulacji oraz wartości określanej jako czas próbkowania. W przypadku solverów o ustalonym czasie próbkowania, odstępy pomiędzy wszystkimi krokami symulacji są sobie równe. Inaczej zdefiniowany jest czas próbkowania dla solverów o zmiennym czasie próbkowania. W tym przypadku, czas próbkowania może się zmieniać w zależności od właściwości dynamicznych modelu. W ogólności, solver o zmiennym czasie próbkowania, zmniejsza czas pomiędzy kolejnymi krokami symulacji, gdy zmiany stanu modelu przebiegają szybciej (zapewniając jednocześnie większą dokładność symulacji) oraz zmniejsza go, gdy zmiany przebiegają wolniej (pomijając niepotrzebne kroki symulacji, jednocześnie ją przyspieszając).

Podczas wyboru typu solvera należy uwzględnić czy wygenerowany kod będzie wykorzystany na rzeczywistym systemie. Jeżeli planowane jest zaimplementowanie modelu na rzeczywistym obiekcie, którego taktowanie zegara jest niezmienne w czasie, należy użyć solvera o niezmiennym czasie próbkowania. Symulacja z użyciem solvera o zmiennym czasie próbkowania, może spowodować pominięcie błędów, które później pojawią się na rzeczywistym obiekcie.

4.1 Dla solverów o ustalonym kroku całkowania (fixed-step size) ustawić wartość kroku na 1 i przeprowadzić symulacje dla wszystkich dostępnych metod całkowania, porównując wyniki. Wykorzystać model z poprzedniego rozdziału.

4.2 Dla solverów o zmiennym kroku całkowania (variable-step size) ustawić maksymalną wartość kroku na 1.1, minimalną na 0.9 oraz początkową na 1. Przeprowadzić symulacje dla wszystkich dostępnych metod całkowania, porównując wyniki.

5 Tworzenie podsystemów i masek

Aby utworzyć podsystem w ramach gotowego modelu, wystarczy zaznaczyć te elementy, które mają wchodzić w skład nowego podsystemu oraz wybrać Edit→Create Subsystem lub nacisnąć Ctrl+G. Dwukrotne kliknięcie bloku podsystemu powoduje otwarcie nowego okna zawierającego schemat podsystemu.

Możliwe jest również utworzenie maski podsystemu. Ułatwia to zadawanie parametrów dla poszczególnych składników podsystemu gromadząc je wszystkie na standardowym oknie dialogowym. Aby możliwe było stworzenie maski podsystemu, bloki powinny zawierać parametry określone nazwami zmiennych (np. k1, Td) zamiast stałych wartości (np. 1, 20). Opcja tworzenia maski dostępna jest w menu kontekstowym podsystemu pod Mask Subsystem.

W otwartym edytorze maski dostępne są cztery zakładki:

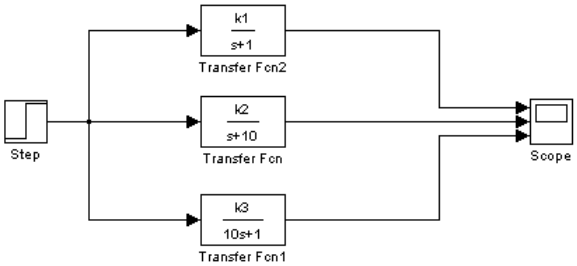
- Icon – pozwala określić wygląd bloku reprezentującego podsystem. Do ikony bloku można dodawać tekst, wstawiać rysunki czy rysować wykorzystując odpowiednie polecenia Matlaba;
- Parameters – pozwala określić parametry definiowalne przez użytkownika w oknie parametrów podsystemu, które będą powiązane z poszczególnymi składnikami; aby dodać parametr do maski podsystemu należy kliknąć przycisk Add, a następnie wypełnić następujące pola:
  - Prompt – tekst który pojawi się jako opis (nazwa) parametru;
  - Variable – nazwa zmiennej, która będzie reprezentowała parametr w podsystemie;
  - Type – typ pola edycyjnego do wprowadzania wartości parametrów; dostępne są: edit (linia edycyjna), checkbox (pole wyboru) i popup (lista rozwijalna);
  - Evaluate – określa czy wartość wprowadzona przez użytkownika ma być obliczana jako wyrażenie Matlaba czy dosłownie przypisywana zmiennej jako ciąg znaków;
  - Tunable – określa czy parametr może być zmieniany w trakcie trwania symulacji.
- Initialization – pozwala określić operacje inicjalizujące parametry podsystemu. Operacje te, które mogą być dowolnymi wyrażeniami Matlaba, są wykonywane przy ładowaniu modelu i uruchamianiu symulacji;
- Documentation – umożliwia wprowadzenie opisów podsystemu, które pojawią się w związanym z nim oknami parametrów i pomocy; na tej zakładce dostępne są trzy pola:
  - Mask type – tekst wyświetlany jako nazwa pola opisu podsystemu w oknie edycji parametrów;
  - Mask description – opis podsystemu wyświetlany w oknie edycji parametrów;
  - Mask help – opis pomocniczy podsystemu, wyświetlany po naciśnięciu przycisku Help na oknie edycji parametrów.

Dwukrotne kliknięcie bloku regulatora spowoduje wyświetlenie zdefiniowanego okna parametrów. Dostęp do edytora maski jest możliwy po wybraniu z menu kontekstowego bloku podsystemu pozycji Edit mask. Dostęp do schematu blokowego podsystemu jest możliwy po wybraniu z menu kontekstowego bloku podsystemu pozycji Look under mask.

5.1 Utworzyć model przedstawiony na Rys. 6. W celu zmienienia liczby sygnałów wejściowych bloku Scope, należy dwukrotnie nacisnąć na niego lewym przyciskiem myszy. W nowo otwartym oknie wybrać z paska narzędzi Parameters oraz określić Number of axes.

5.2 Zaznaczyć wszystkie bloki Transfer Fcn wraz z poprzedzającym je węzłem zaczepowym i utworzyć podsystem o nazwie *Obiekty*.

5.3 Utworzyć maskę podsystemu, umożliwiającą ustawianie wartości parametrów *k1*, *k2* oraz *k3*.



Rysunek 6: *Drugi przykładowy model Simulinka.*