



Politechnika Wroclawska

Prototypowanie systemów sterowania

Prowadzący:

dr hab. inż. Mateusz Dybkowski, prof. uczelni

Opracował:

mgr inż. Szymon Bednarz, dr hab. inż. Mateusz Dybkowski, prof. uczelni

Laboratorium nr 6

Zasady testowania w czasie rzeczywistym. Zapoznanie się ze sprzętem firmy dSPACE i oprogramowaniem do realizacji testów w czasie rzeczywistym

1. Wprowadzenie

Celem zajęć jest zapoznanie się techniką Hardware-in-the-Loop (HiL) stosowaną do testów układów sterowania w czasie rzeczywistym, a także z urządzeniami firmy dSPACE umożliwiającymi ich realizację.

2. Testowanie systemów sterowania w czasie rzeczywistym

Prototypowanie systemów sterowania metodą Model-in-the-Loop (MiL) najczęściej nie nakłada wymagań czasu rzeczywistego podczas testów. Oznacza to, że czas trwania symulacji (czas wykonania poszczególnych operacji) nie jest możliwy do obliczenia. Przykładowo dla zdefiniowanego czasu wykonywania symulacji T_s rzeczywisty czas jej wykonywania może być znacznie dłuższy lub krótszy, co może wynikać między innymi z wymaganej dokładności obliczeń i zastosowanej metody numerycznej.

Jednakże każdy system sterowania powinien być testowany pod rygorem czasu rzeczywistego, ponieważ docelowo będzie pracował w takich warunkach. System czasu rzeczywistego jest takim systemem, którego poprawność działania zależy nie tylko od samego wyniku, ale również od czasu, w którym został on osiągnięty.

Oznacza to, że dla zdefiniowanego czasu trwania symulacji T_s rzeczywisty czas jej wykonywania jest dokładnie równy T_s . Każda kolejna wartość wyjściowa musi być obliczona w ściśle określonym czasie (systemy deterministyczne). Takie rozwiązanie pozwala dokładnie odwzorować działanie rzeczywistego sterownika i obiektu. Można wyróżnić dwie podstawowe metody prototypowania/testowania układów sterowania w czasie rzeczywistym:

- Wykorzystanie zasobów sprzętowych komputerów osobistych poprzez implementację symulatora systemu czasu rzeczywistego, który synchronizuje obliczenia wykonywane przez komputer zgodnie z wymaganiami czasu rzeczywistego. Przykładem może być oprogramowanie MATLAB/Simulink i narzędzie Simulink Desktop Real-Time (w starszych wersjach znane jako Real-Time Windows Target).
- Wykorzystanie specjalizowanych urządzeń wyposażonych w wydajne układy mikroprocesorowe, układy FPGA, karty I/O itd. Obliczenia wykonywane w układzie sterowania oraz w obiekcie sterowania są wykonywane pod reżimem czasu rzeczywistego. Rola komputera osobistego ogranicza się wtedy najczęściej do

programowania sprzętu i prezentacji graficznej danych a także zadawania wartości referencyjnych wybranych zmiennych poprzez wirtualne panele sterownicze.

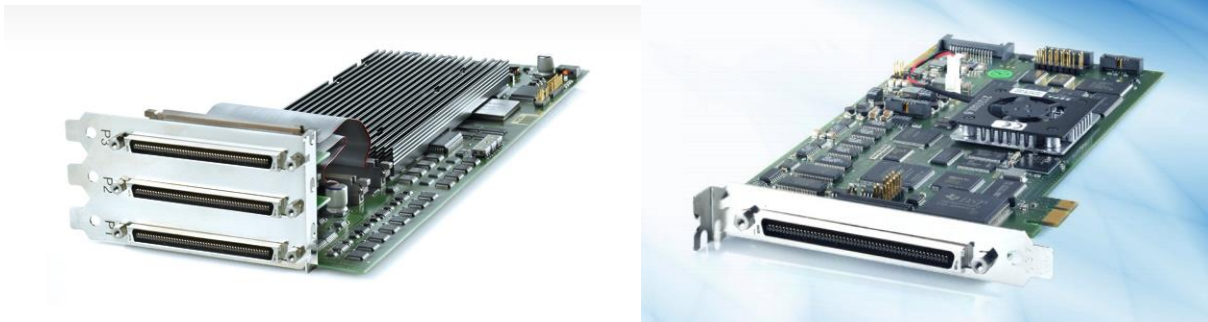
3. Testowanie układów sterowania w czasie rzeczywistym z zastosowaniem oprogramowania MATLAB/Simulink i sprzętem dSPACE

Oprogramowanie MATLAB/Simulink pozwala na projektowanie układów sterowania metodą MiL. Ważną cechą tego środowiska jest to, że umożliwia automatyczne generowanie kodu z utworzonego w postaci graficznej schematu blokowego sterowania w formacie przeznaczonym dla różnorodnych platform sprzętowych. Dzięki temu możliwe jest testowanie rozwiązań w czasie rzeczywistym na docelowych urządzeniach. Obecnie podstawowe narzędzia do automatycznej generacji kodu to:

- **MATLAB Coder** - generuje niezależny kod C/C++ z kodu języka MATLAB.
- **Simulink Coder** (dawniej Real-Time Workshop) - generuje i uruchamia samodzielny kod C/C++ do rozwijania i testowania algorytmów modelowanych w Simulinku. Kod wynikowy może być użyty w wielu typach aplikacji (w tym aplikacji czasu rzeczywistego) takich jak przyspieszanie symulacji, szybkie prototypowanie i testowanie metodą HiL.
- **Embedded Coder** - tworzy kod C/C++ ze skryptów w języku MATLAB, z modeli Simulinka oraz diagramów Stateflow, które mają czytelność i efektywność profesjonalnych ręcznie pisanych kodów. Wygenerowany kod jest wyjątkowo zwięzły i szybki, co jest zasadniczym wymaganiem w systemach wbudowanych, docelowych systemach szybkiego prototypowania, mikroprocesorach używanych w produkcji masowej i symulatorach czasu rzeczywistego. Rozszerza MATLAB Coder i Simulink Coder o optymalizację i właściwości konfiguracyjne nieodzowne do generacji kodu produkcyjnego (np. profilowanie kodu).

Na zajęciach studenci zapoznają się z zastosowaniem oprogramowania MATLAB/Simulink do programowania urządzeń firmy dSPACE. W laboratorium znajduje się kilka specjalistycznych kart do szybkiego prototypowania. Są one wyposażone we własną pamięć i wydajny procesor oraz analogowe i cyfrowe układy wejścia/wyjścia sprzęgające kartę z czujnikami oraz układami wykonawczymi. Karty te mogą być instalowane w komputerach PC (ISA i PCI), w komputerach przemysłowych z szyną VME oraz w obudowach zwanych AutoBox, dołączanych do szyny ISA, PCI, VME lub poprzez sieci: CAN, Ethernet i inne. Przy korzystaniu z kart prototypowych komputer PC używany głównie jako interfejs użytkownika. Program symulacyjny czasu rzeczywistego i obsługa interfejsów są realizowane przez program wgrany do karty prototypowej.

W laboratorium dydaktycznym wykorzystywane są trzy modele kart: DS1102, DS1103 i DS1004, które w zależności od wersji wyposażone są w układy peryferyjne takie jak przetworniki analogowo-cyfrowe (A/D) i cyfrowo-analogowe (D/A), wejścia i wyjścia cyfrowe (DIO), wyjścia PWM, interfejsy komunikacyjne (Ethernet, CAN, UART), interfejsy do obsługi encoderów i/lub resolverów. Poniżej przedstawiono podstawowe parametry poszczególnych kart.

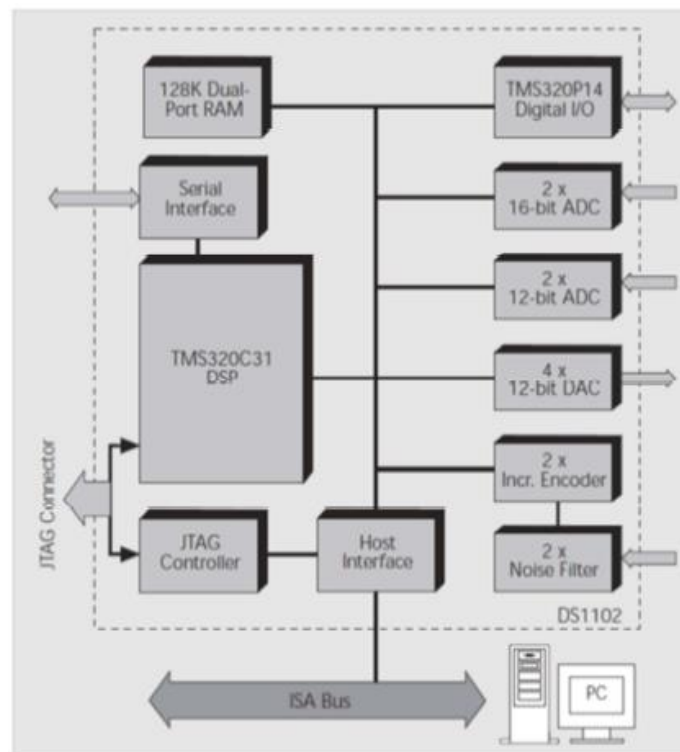


Rys. 1. Karty dSPACE 1103 i 1104



Rys. 2. Panele połączeniowe CP 1103 oraz CP 1104

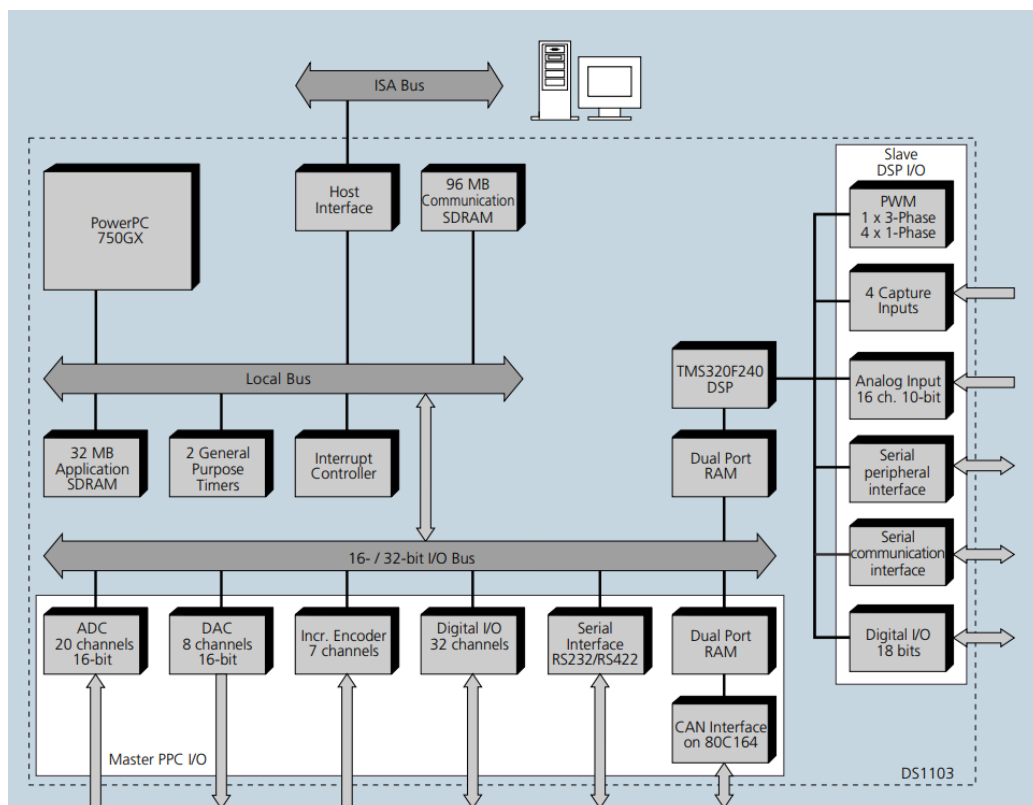
I. Karta do szybkiego prototypowania DS1102



Rys. 3. Schemat ideowy karty DS1102

Procesor główny (Master)	TMS320C31 32-bitowy zmiennoprzecinkowy Zegar: 60MHz
Wejścia analogowe	2 kanały A/D (16-bitów) 2 kanały A/D (12-bitów) Zakres napięć $\pm 10V$
Wyjścia analogowe	4 kanały D/A (12-bitów) Zakres napięć $\pm 10V$
Wejścia/wyjścia cyfrowe	26 równoległych kanałów I/O
Procesor pomocniczy (Slave)	TMS320P14 Do obsługi wejść/wyjść cyfrowych

II. Karta do szybkiego prototypowania DS1103

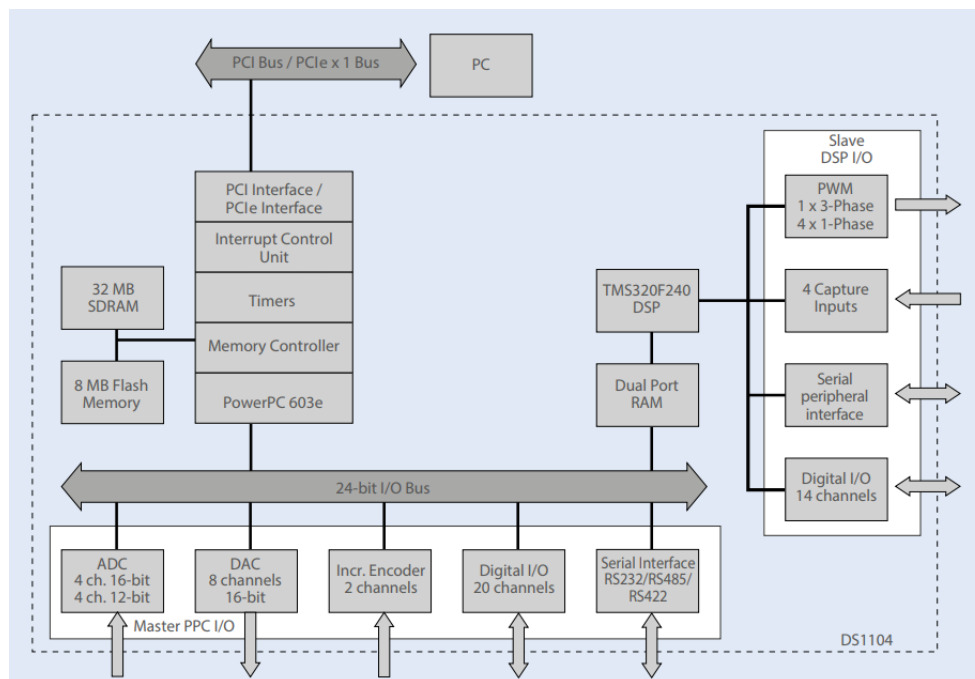


Rys. 4. Schemat ideowy karty DS1103

Procesor główny (Master)	PPC 750GX Zegar: 1GHz 32 kB wewnętrznej pamięci cache
Pamięć	32 MB SDRAM – pamięć lokalna (aplikacje i dane) 96 MB SDRAM – pamięć globalna
Wejścia analogowe	16 multipleksowane kanały A/D (16-bitów) Zakres napięć $\pm 10V$
Wyjścia analogowe	8 kanałów D/A (16-bitów) Zakres napięć $\pm 10V$
Wejścia/wyjścia cyfrowe	32 równoległe kanały I/O (4 kanały po 8 bitów konfigurowalnych jako wejście lub wyjście)
Procesor pomocniczy (Slave)	TMS320F240 Wykorzystywany do obsługi: <ul style="list-style-type: none"> • 18 bitowego portu cyfrowego

	<ul style="list-style-type: none"> • 16 kanałowego wejścia analogowe (10-bitów) • 3- fazowego wyjścia PWM • 4 jednofazowych wyjść PWM • 2 portów szeregowych
--	--

III. Karta do szybkiego prototypowania DS1104

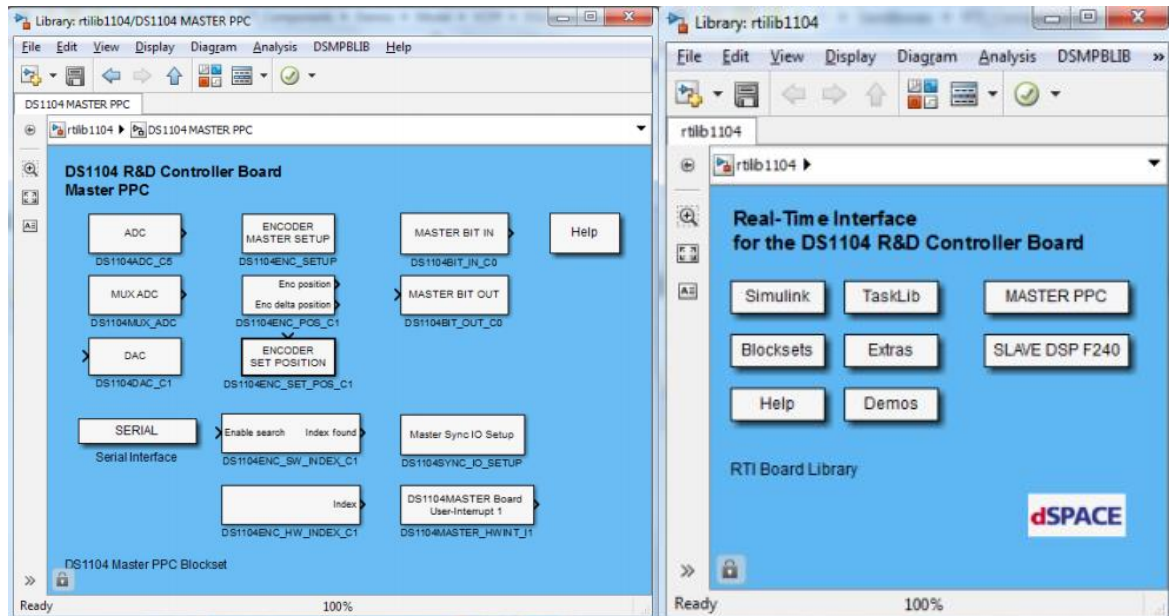


Rys. 5. Schemat ideowy karty DS1104

Procesor główny (Master)	MPC8240 64-bitowy zmiennoprzecinkowy Zegar: 250 MHz 32 kB cache
Pamięć	32 MB SDRAM – pamięć globalna 8 MB pamięci Flash
Wejścia analogowe	4 multiplexowane kanały A/D (1 x 16 bitów) 4 równoległe kanały A/D (4 x 16 bitów) Zakres napięć ±10V
Wyjścia analogowe	8 kanałów D/A (16-bitów) Zakres napięć ±10V
Wejścia/wyjścia cyfrowe	20 równoległych kanałów I/O konfigurowalnych jako wejście lub wyjście
Procesor pomocniczy (Slave)	TMS320F240 Wykorzystywany do obsługi: <ul style="list-style-type: none"> • 14 bitowego portu cyfrowego • 3- fazowego wyjścia PWM • 4 jednofazowe wyjść PWM • 1 portu szeregowego

Jak wspomniano wcześniej karty te programowane są za pomocą oprogramowania MATLAB/Simulink oraz dodatkowych narzędzi:

- **Simulink Coder** (dawniej wcześniej Real-Time Workshop) generuje i uruchamia kod C/C++ na bazie schematu blokowego w Simulinku.
- **RTI (Real-Time Interface)** biblioteka dostarczana przez producenta dSPACE zawiera bloki obsługujące podzespoły danej karty, które mogą być zastosowane bezpośrednio w schemacie blokowym układu sterowania tworzonym w Simulinku. Jest rozszerzeniem narzędzia Simulink Coder do implementacji modeli Simulink i Stateflow na symulatorze sprzętowym czasu rzeczywistego. Model czasu rzeczywistego jest kompilowany, pobierany i automatycznie uruchamiany na symulatorze sprzętowym.



Rys. 6. Biblioteka RTI dla karty DS1104

Do zarządzania procesem przeprowadzania testów w czasie rzeczywistym stosowane jest dedykowane oprogramowanie **ControlDesk** wspomagające proces rozwoju dowolnego sterownika. W ramach projektu w oprogramowaniu definiujemy zbiór eksperymentów, z których każdy składa się z dwóch zasadniczych elementów: wykorzystywanego podczas eksperymentu sprzętu oraz panelów sterowania. Panele sterowania to zestaw instrumentów graficznych (kontrolki) układanych na ekranie przez użytkownika, a służący do wizualizacji wartości fizycznych odczytanych bezpośrednio ze sprzętu. Proces przeprowadzania eksperymentu przypomina obsługę typowego oprogramowania SCADA.

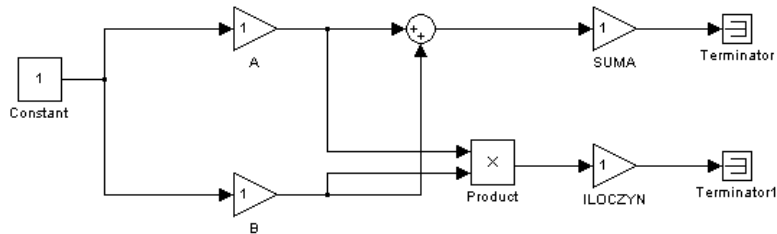
Wirtualne panele sterowania pozwalają na wizualizację przebiegu eksperymentu z wykorzystaniem szerokiej gamy instrumentów takich jak: listy parametrów, wykresy, pola edycyjne, suwaki, guziki i wszystkie inne elementy typowe dla aplikacji komputerowych oraz kontrolki spotykanych w systemach sterowania.

4. Przykład

Proces tworzenia projektu można podzielić na kilka etapów:

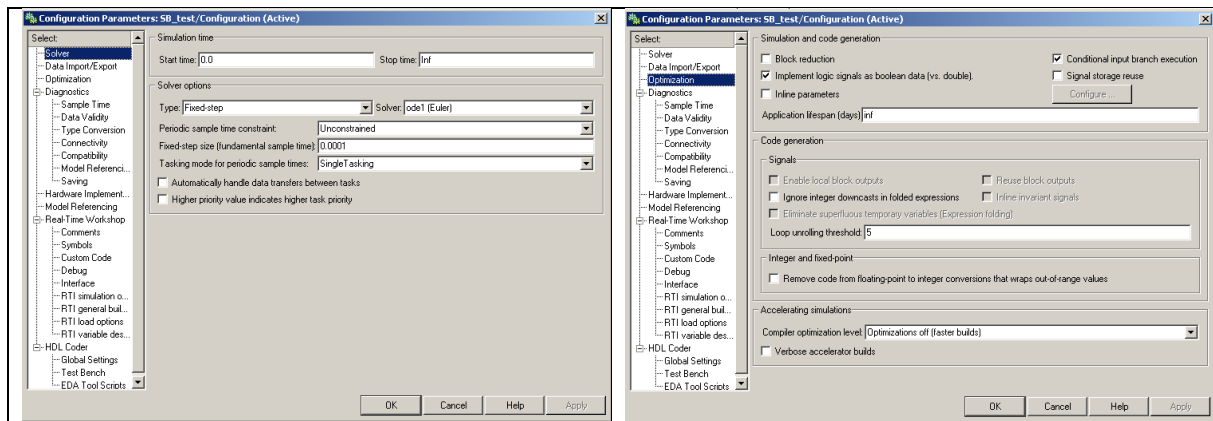
I. Utworzenie modelu systemu w Simulinku

UWAGA: Przed przystąpieniem do prac należy utworzyć Nowy folder na Pulpicie a następnie ustawić go jako folder roboczy MATLABA.



Rys. 7. Przykładowy model relizujący funkcję dodawania i mnożenia dwóch sygnałów

II. Konfiguracja modelu (Simulation -> Configuration Parameters)




III. Generacja kodu do wybranej platformy sprzętowej (Ctrl+B)

IV. Konfiguracja oprogramowania ControlDesk

- Utworzenie nowego eksperymentu: File -> New Experiment

UWAGA: należy ustawić ścieżkę docelową eksperymentu (Working root) taką jak w jak w Matlabie

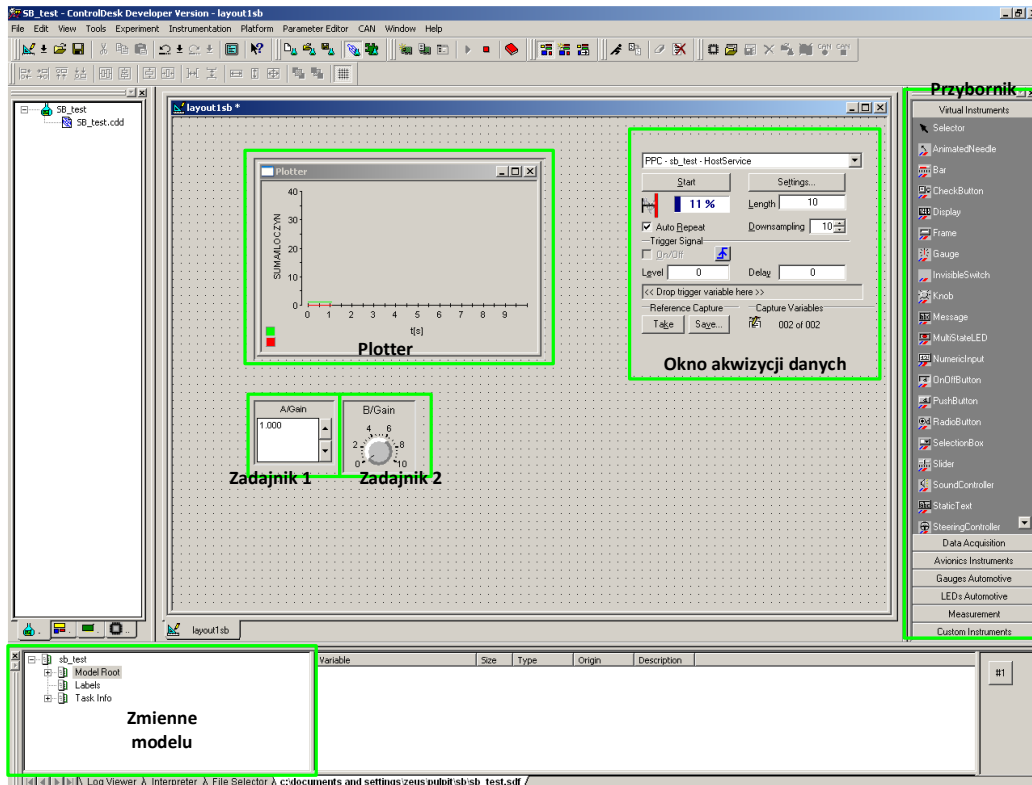
- Import pliku *.sdf z definicjami i ścieżkami do danych (ikona ).

V. Utworzenie wirtualnego panelu sterowania

- Dodanie panelu (Layout): File -> New -> Layout
- Włączenie **przybornika elementów**: View -> Controlbars -> Instrument Selector
- Dodanie elementów:
 - **Okno akwizycji danych**: Data Acquisition -> Capture Settings
 - **Plotter**: Data Acquisition -> PlotterArray
 - **Zadajnik 1**: Virtual instruments -> Numeric input
 - **Zadajnik 2**: Virtual instruments -> Knob


Odpowiednie zmienne wybiera się z menu kontekstowego (Zmienne modelu), które znajdują się w zakładce Model Root. Nazwy zmiennych pokrywają się z nazwami bloków modelu w Simulink.

- W celu wyświetlenia zmiennej należy przenieść ją do plotera (kolumna Variable - > Out),
- aby zmieniać wartości zmiennej należy przenieść ją do odpowiedniego zadajnika (kolumna Variable -> Gain).



Rys. 8. Docelowy wygląd panelu sterowania

VI. Testowanie

- Uruchomienie aplikacji: Animation mode (ikona .
- W bloku Capture Settings z menu rozwijanego należy wybrać: PPC – nazwa_modelu_Simulink_HostService
- Zmiana długości bufora danych: (pole Length).

5. Zadania do wykonania:

- I. Konfiguracja środowiska MATLAB/Simulink.
- II. Opracowanie generatorów wybranych sygnałów (np. sygnał stały, sinusoidalny, prostokątny) oraz układu resetującego przebiegi czasowe w trakcie trwania symulacji.

- III. Budowa wirtualnego panelu sterowania w programie ControlDesk umożliwiającemu zmianę parametrów generowanych sygnałów oraz obserwację ich przebiegów.

Literatura:

[1] www.mathworks.com

[2] www.dspace.com