Kacper Szmajdel  
Joanna Nużka  
Szymon Gadzina  
Marcin Pilarski

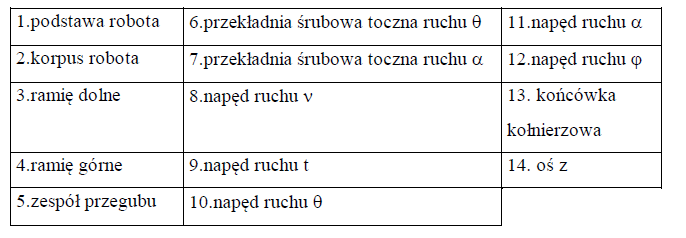
**Prototypowanie sterownika do robota IRp-6**

**1. Cel ćwiczenia**

Celem ćwiczenia było zaprojektowanie prototypu sterownika do robota IRp-6, realizującego poniższe funkcje:

* Sterowanie napędów: serwomechanizmy dla poszczególnych osi z uwzględnieniem ograniczeń dla wartości zadanych kątów
* Bazowanie robota
* Pozycjonowanie w układzie współrzędnych konfiguracyjnych - tryb pracy „ręcznej” (niezależne ruszanie osiami) i tryb „wyzwalany” (ustawianie wartości zadanych dla osi i uruchomienie ruchu)
* Zadawanie prędkości roboczej ruchu
* Koordynacja prędkości ruchu
* Pozycjonowanie w układzie kartezjańskim: implementacja prostego zadania kinematyki pozycji

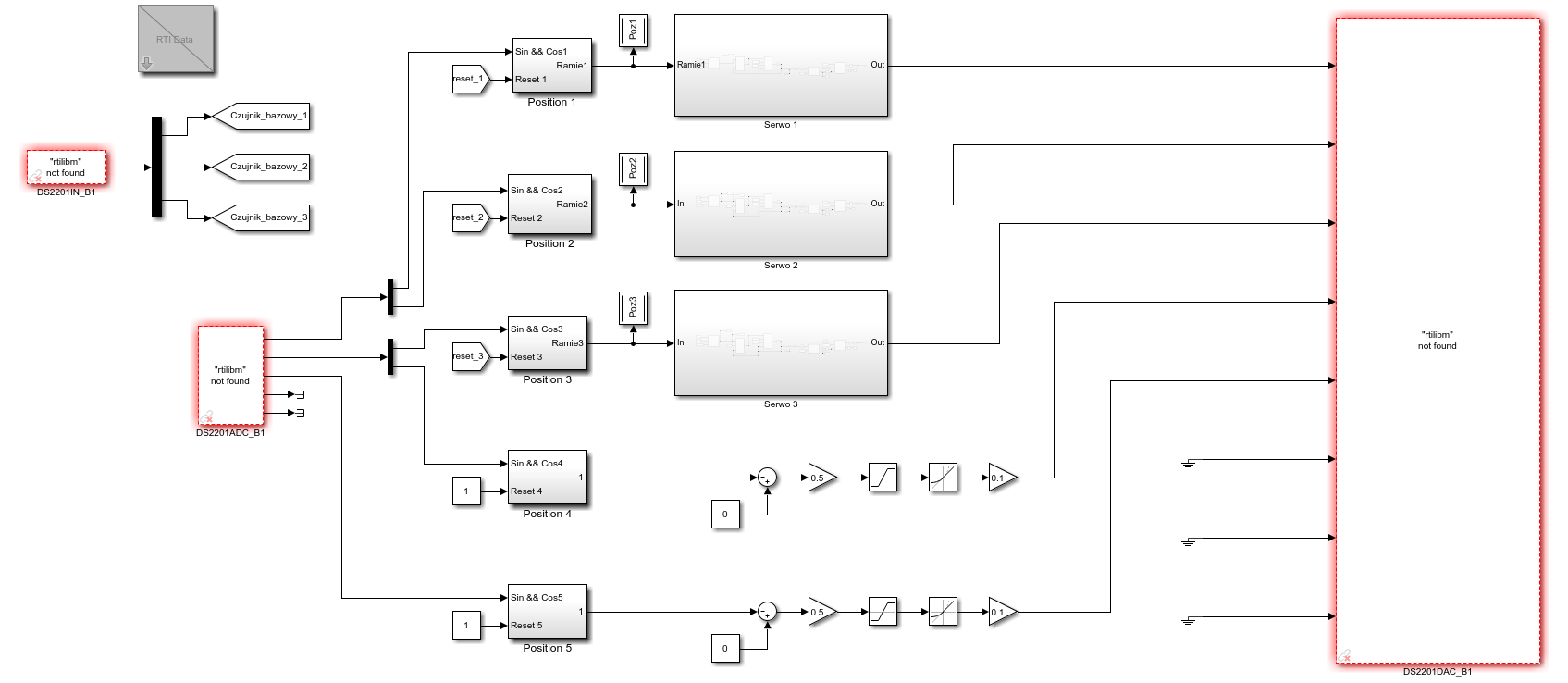
****

****

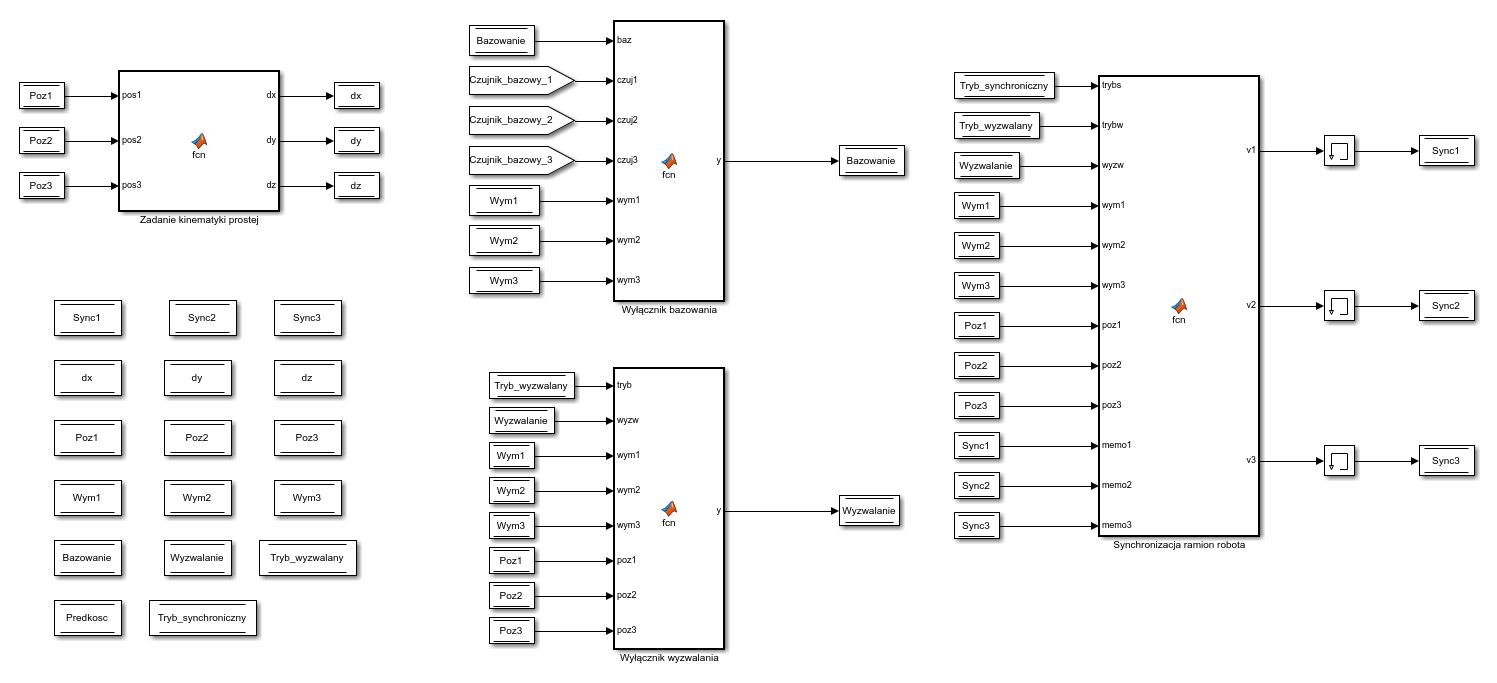
Rys. 1 Część manipulacyjna robota IRp-6

**2. Przebieg ćwiczenia**

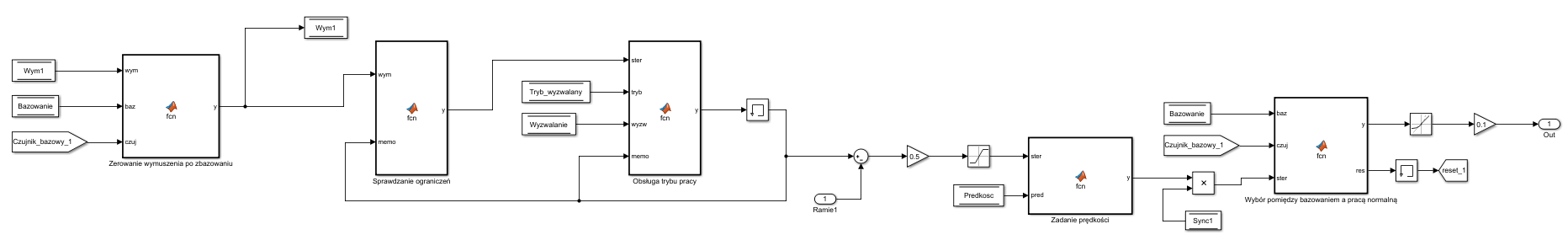
Aby zrealizować ćwiczenie, utworzyliśmy w Simulinku schemat blokowy serwomechanizmów, a następnie modyfikowaliśmy go, dodając kolejne funkcje. Stworzyliśmy także panel operatorski w programie Control Desk, dzięki któremu mogliśmy testować działanie programu na rzeczywistym robocie. Udało nam się zrealizować wszystkie punkty ćwiczenia.



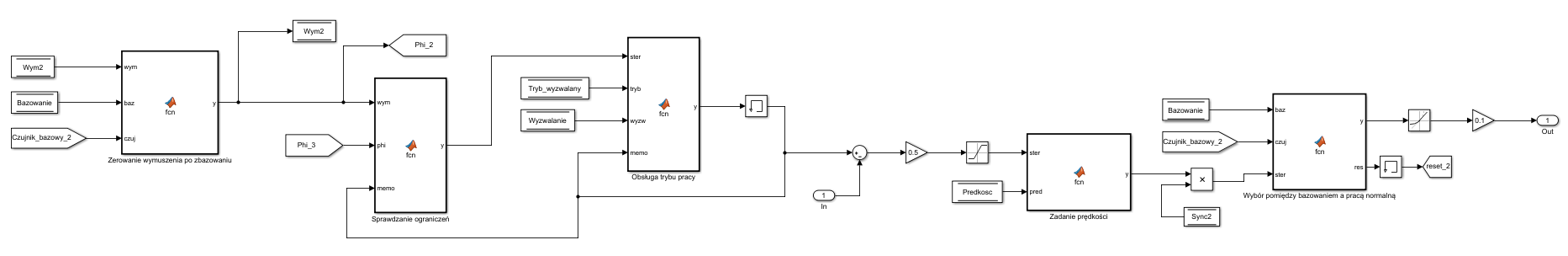
Rys. 2 Schemat głównego układu sterowania



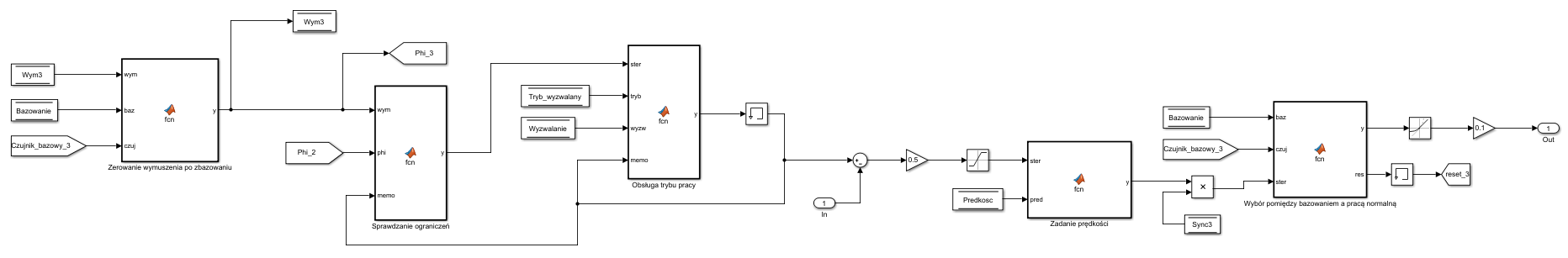
Rys. 3 Globalne zmienne układu oraz realizacje dodatkowych funkcji sterownika



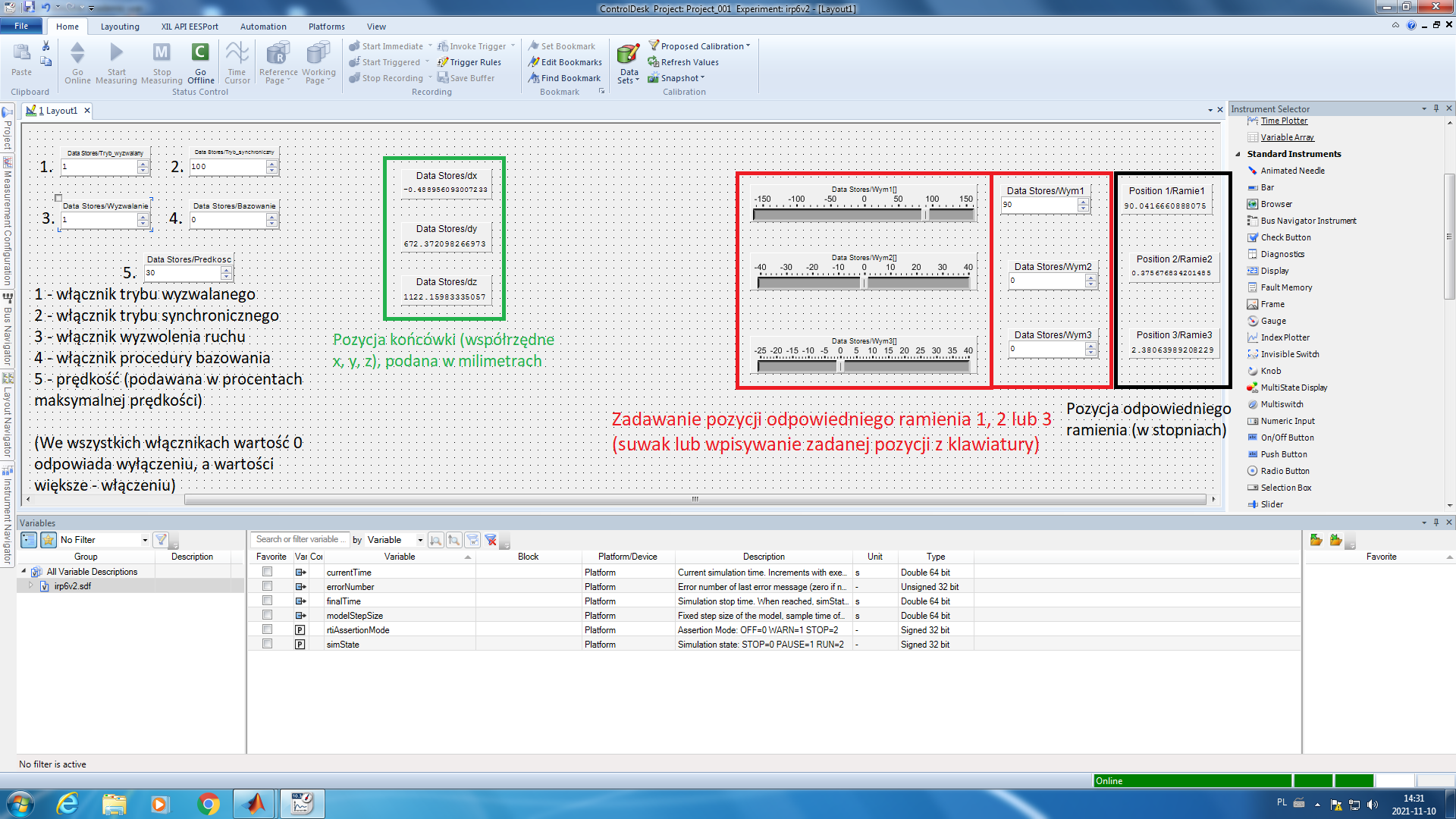
Rys. 4 Serwomechanizm sterujący pierwszym ramieniem



Rys. 5 Serwomechanizm sterujący drugim ramieniem



Rys. 6 Serwomechanizm sterujący trzecim ramieniem



Rys. 7 Panel operatorski

**3. Wnioski**

Podczas pracy nad sterownikiem zauważyliśmy, że bardzo pomocne przy bardziej złożonych schematach są bloczki Matlab Function. Pozwalają one na czytelniejsze zapisanie warunków   
i bardziej złożonych zależności. Należy jednak uważać, ponieważ działają one szybciej niż np. tradycyjne bloczki Switch, przez co w programie mogły powstawać pętle– sygnał wyjściowy z bloku był jednocześnie jednym z argumentów wejściowych, co powodowało błędy. Należało wprowadzać minimalne opóźnienie w programie za pomocą np. bloczków Memory.