

Sławomir JASZCZAK

ZACHODNIOPIOMORSKI UNIWERSYTET TECHNOLOGICZNY, WYDZIAŁ INFORMATYKI, ul. Żołnierska, 49, 71-210 Szczecin

Procedura hardware in the loop w syntezie algorytmów sterowania

Dr inż. Sławomir JASZCZAK



Od 1994 jako asystent na Wydziale Techniki Morskiej Politechniki Szczecińskiej prowadzi badania w zakresie sterowania pojazdami głębinowymi z wykorzystaniem metod sztucznej inteligencji. Od 2002 zatrudniony na stanowisku adiunkta w Zakładzie Sztucznej Inteligencji. Bieżące zainteresowania wiążą się z implementacją złożonych algorytmów sterowania dyskretnego i cyfrowego na platformie wykonawczej PLC.

e-mail: sjaszczyk@wi.zut.edu.pl**Streszczenie**

W artykule omówiono wykorzystanie technologii OPC do badania systemów sterowania zgodnie z procedurą Hardware In The Loop. Artykuł jest efektem eksperymentów nad zastosowaniem programu Simulink jako aplikacji klienckiej, za pomocą której zrealizowano model obiektu i KepwareOPC jako serwer OPC. Procedura HIL została przeprowadzona w układzie sterowania kursem pojazdu podwodnego z wykorzystaniem sterownika PLC jako platformy wykonawczej algorytmu sterowania i modelu pojazdu podwodnego w programie Simulink.

Słowa kluczowe: HIL (Hardware In The Loop) PLC (Programmable Logic Controller), OPC (OLE for Process Control), DDE (Dynamic Data Exchange).

Hardware in the loop procedure used for the control system synthesis**Abstract**

In this paper possibility of using OPC technology for the process of designing and testing the heading control algorithm according to the Hardware In The Loop (HIL) procedure is described. The main problem in the design of automated systems is the safe real time testing of the control algorithm, during a deployment. This algorithm synthesized at the computer simulation level according to the Model In The Loop procedure with use of specialized software does not take into account specification of the target platform i.e.: limited time for executing a single algorithm loop, limited program memory, limited precision of signal values etc. According to the HIL procedure the control algorithm is implemented using a target executing platform (i.e. PLC, microcontroller, PC + DAQ board, etc.), whilst a plant is the mathematical model in PC memory. The presented text is a result of the experiments on employing Simulink as a client application, which enables simulation of the specific and KepwareOPC as the OPC. The main advantage of the HIL procedure is the possibility of safe real time testing of the control algorithm. Besides the security reasons, also the cost of implementation and testing of algorithm on real object is essential. In the third paragraph the HIL experiment concerning the control of an underwater vehicle course angle is presented. The PLC controller with the control algorithm is a control system, whereas the underwater vehicle model is implemented in the Simulink.

Keywords: HIL (Hardware In The Loop) PLC (Programmable Logic Controller), OPC (Open Process Control), DDE (Dynamic Data Exchange).

1. Wstęp

Projektowanie układów sterowania w środowiskach akademickich opiera się na stosowaniu procedury MIL, polegającej na zasymulowaniu algorytmu sterowania, obiektu sterowania oraz układów wykonawczo-nastawczych i pomiarowych na podstawie modeli matematycznych opracowanych na drodze analizy teoretycznej lub identyfikacji doświadczalnej. Powszechny brak dostępu do rzeczywistych obiektów sterowania uzasadnia procedurę MIL. Realizacja i testowanie układu sterowania zgodnie z procedurą MIL, przy wykorzystaniu specjalizowanego oprogramowania np. Matlab nie uwzględnia jednak specyfiki docelowej platformy wykonawczej np. ograniczenie czasu wykonania pojedynczej pętli

algorytmu, ograniczonej pamięci programu, ograniczonej precyzji wartości sygnałów itp. Rozbieżność między działaniem algorytmu na poziomie symulacji komputerowej i w czasie rzeczywistym może prowadzić, w krańcowym przypadku, do zniszczenia obiektu skutkiem np. niestabilności układu sterującego. Często prowadzi one do sytuacji manualnego dostrajania algorytmu metodą prób i błędów, co oznacza w praktyce zarzucenie wyników, uzyskanych metodą MIL.

Bezpieczne testowanie algorytmu sterowania w czasie rzeczywistym jest szczególnie istotne w przypadku sterowania obiektami, których awaria mogłaby spowodować groźne oddziaływanie na człowieka. Innym problemem jest szkolenie operatorów do pracy z obiektami w warunkach normalnej eksploatacji. Eksperymentowanie w czasie rzeczywistym może być przyczyną uszkodzenia obiektu sterowania obsługiwane przez niedoświadczonego operatora lub zastosowania algorytmu sterowania, który nie spełnia wymogu stabilnej pracy obiektu.

W celu zbliżenia się do rzeczywistych warunków pracy układ sterowania symulacja i testowanie układu sterowania można zrealizować na poziomie Hardware-in-the-Loop, co oznacza, że algorytmy sterowania będą wykonywane na docelowej platformie wykonawczej. Oznacza to, że symulacja będzie realizowana za pomocą rzeczywistego regulatora (sterownik PLC VersaMax Micro UDD110 wraz zaimplementowanym algorytmem sterowania) i modelem dynamiki obiektu sterowanego. Powstaje pytanie w jaki sposób zapewnić synchronizację pracy sterownika PLC i modelu zaimplementowanego w pamięci komputera osobistego. Można w tym celu wykorzystać popularne standardy komunikacyjne, stosowane w automatyce przemysłowej tj. protokoły DDE i OPC. Zasada symulacji HIL opiera się o architekturę klient-serwer, w której OPC serwer (KepwareOPC) lub DDE serwer GE Fanuc Series 90 Protocol odpowiada za wymianę danych (konwersja danych maszynowych w systemowe, synchronizacja itp.), natomiast program Simulink jako aplikacji klienckiej, realizuje model obiektu. Zaletą procedury HIL jest możliwość bezpiecznego testowania algorytmu sterowania w czasie rzeczywistym. Poza względami bezpieczeństwa istotne są również koszty wdrożenia i testowania algorytmów na rzeczywistym obiekcie.

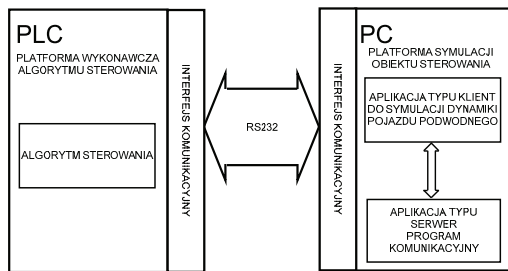
W rozdziale końcowym przedstawiono eksperyment HIL, polegający na sterowaniu kursem pojazdu podwodnego MAGIS.

2. Simulink, OPC i DDE w symulacji czasu rzeczywistego

Symulacja czasu rzeczywistego HIL wymaga zastosowania docelowej platformy wykonawczej i modelu matematycznego obiektu sterowania. Uogólnioną strukturę stanowiska badawczego przedstawiono na rys. 1.

Komunikację między rzeczywistą platformą wykonawczą algorytmu sterowania a modelem symulowanym w pamięci zapewniają urządzenia techniczne wraz z oprogramowaniem, umożliwiającym wymianę informacji między dwoma systemami. Na poziomie fizycznym komunikacja może być realizowana m.in. za pomocą interfejsów szeregowych (RS232C, RS422, RS485) lub interfejsu sieciowego. Natomiast konwersję danych maszynowych w dane systemowe i ich przesyłanie można zrealizować wykorzystując standardowe technologie, wykorzystywane w praktyce przemysłowej, dostępne dla typowego sterownika przemysłowego tj. DDE i OPC. Symulacja i testowanie na poziomie HIL uwzględnia uwarunkowania pracy układu regulacji w czasie rzeczywistym i jest odpowiednia przy założeniu dostępności docelowej platformy wykonawczej algorytmu sterowania. Projektant ma ponadto możliwość testowania wielu platform wykonawczych, biorąc pod uwagę ich rzeczywiste ograniczenia tj. prędkość wykonania programu sterującego, wielkość pamięci programu i danych, łatwość

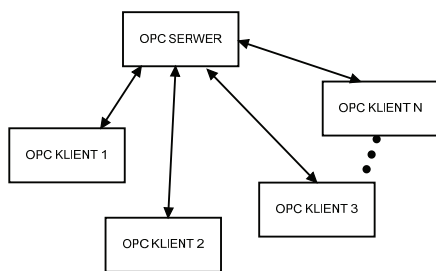
programowania, obsługę komunikacji itp. Nadal najistotniejsza okazuje się weryfikacja wyników syntezy algorytmu sterowania na poziomie Model-in-the-Loop.



Rys. 1. Elementy systemu do testowania i symulacji opartego na procedurze HIL
Fig. 1. Elements of the testing and simulation system based on the HIL procedure

Standard OPC został stworzony w celu zapewnienia współpracy wielu urządzeń i oprogramowania przemysłowych różnych producentów. Podstawowa idea polega na ujednoczeniu interfejsu na poziomie wymiany danych, tak aby niezależnie się od specyfiki np. sposobu adresowania zmiennych czy protokołu komunikacyjnego. OPC bazuje na architekturze klient – serwer, co przedstawiono na rys. 2. Klienci OPC np. Mathworks OPC Toolbox mają dostęp do zmiennych maszynowych tzw. itemów pobieranych przez OPC serwer z obiektów sterowanych.

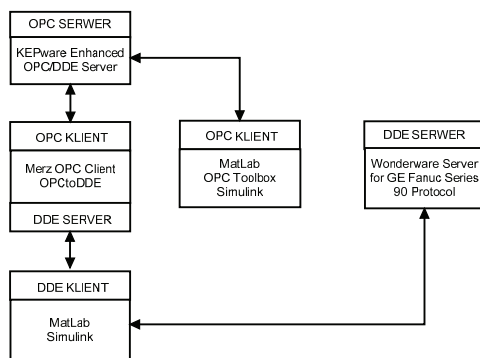
Więcej na temat standardu OPC można znaleźć na stronie [1].



Rys. 2. Elementy systemu do testowania i symulacji opartego na procedurze HIL
Fig. 2. Elements of the testing and simulation system based on the HIL procedure

W ramach niniejszego opracowania do realizacji symulacji i testowania HIL założono Kepware OPC serwer, działający jako w pełni funkcjonalna aplikacja przez dwie godziny oraz OPC Toolbox, działający na poziomie Matlab Command Window i Simulink. OPC Toolbox działa wyłącznie jako aplikacja kliencka i umożliwia swobodną integrację z dowolnym modelem układu sterowania, opracowanym w Simulinku.

Możliwe sposoby wymiany danych pomiędzy sterownikiem PLC a Simulinkiem przedstawiono na rys. 3.



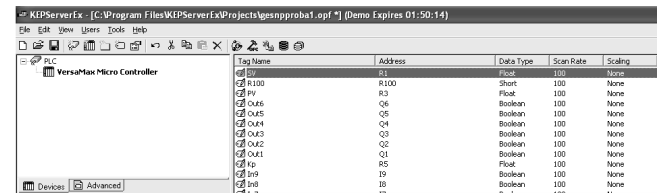
Rys. 3. Warianty komunikacji pomiędzy modelem w Simulinku a PLC przy wykorzystaniu OPC i DDE

Fig. 3. Variants of communication between a model in the Simulink and the control algorithm

Przygotowanie symulacji HIL obejmuje konfigurację serwera OPC oraz klienta OPC. Poprawna konfiguracja serwera OPC wymaga:

- określenia nazwy kanału i związania jej z portem komunikacyjnym i docelowym urządzeniem
- zadeklarowania listy zmiennych tzw. tagów wraz z adresami maszynowymi.

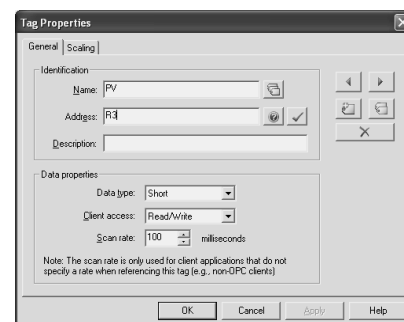
Okno konfiguracji serwera OPC przedstawiono na rys. 4.



Rys. 4. Okno konfiguracji serwera OPC

Fig. 4. A configuration window of the OPC server

Nazwa kanału wymiany danych oraz nazwa determinują strukturę nazw zmiennych do których będziemy się odwoływać w aplikacji klienckiej. Należy zwrócić uwagę, że nie odwołujemy się bezpośrednio do zmiennej maszynowej, a jedynie tzw. aliasu, reprezentującego drzewo danego kanału wymiany danych. Przykładowo do zmiennej maszynowej typu rejestrowego R3 w sterowniku PLC, określonej w serwerze OPC tagiem PV (rys. 5), odwołujemy się poprzez alias PLC_VersaMax Micro Controller.SV w kliencie OPC. <PLC> stanowi nazwę kanału, <VersaMax Micro Controller> nazwę urządzenia, natomiast <SV> etykietę zmiennej. Serwer OPC umożliwia ponadto indywidualne określenie czasu odświeżania każdej zadeklarowanej zmiennej.



Rys. 5. Deklaracja zmiennej w serwerze OPC

Fig. 5. A declaration of variables in the OPC server

Powiązanie aplikacji klienckiej tj. Simulinka z serwerem można zrealizować poprzez zastosowanie bloków OPC Read i Write po uprzednim skonfigurowaniu kanału wymiany danych za pomocą bloku OPC Config – rys. 5.

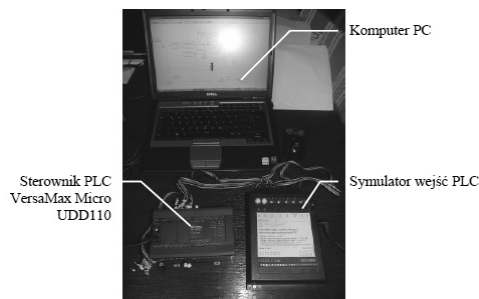
Blok OPC Read umożliwia czytanie wartości bezpośrednio z platformy sprzętowej, natomiast OPC Write zapis.

3. Wyniki badań eksperymentalnych

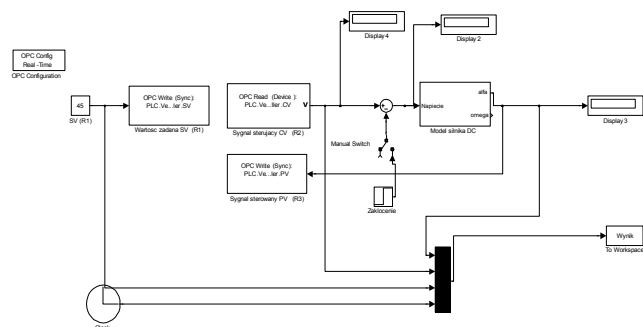
Zastosowanie środowiska MatLab/Simulink umożliwia swobodne stosowanie algorytmu sterowania, zaimplementowanego na docelowej platformie wykonawczej, do wielu odmiennych obiektów dynamicznych, z uwzględnieniem rzeczywistych warunków pracy takich jak: oddziaływanie zakłóceń, nieliniowości urządzeń wykonawczych i pomiarowych, niestacjonarność obiektu dynamicznego skutkiem zmiany punktu pracy. Wymienione uwarunkowania sprawiają, że procedura HIL stanowi cenne narzędzie do weryfikacji algorytmów sterowania bez narażania rzeczywistego obiektu na zniszczenie a obsługę na zagrożenie zdrowia skutkiem niestabilności układu sterowania.

W trakcie badań eksperymentalnych testowano układ stabilizacji głębokości pojazdu podwodnego, opierając się na modelu dynamiki pojazdu podwodnego MAGIS [2].

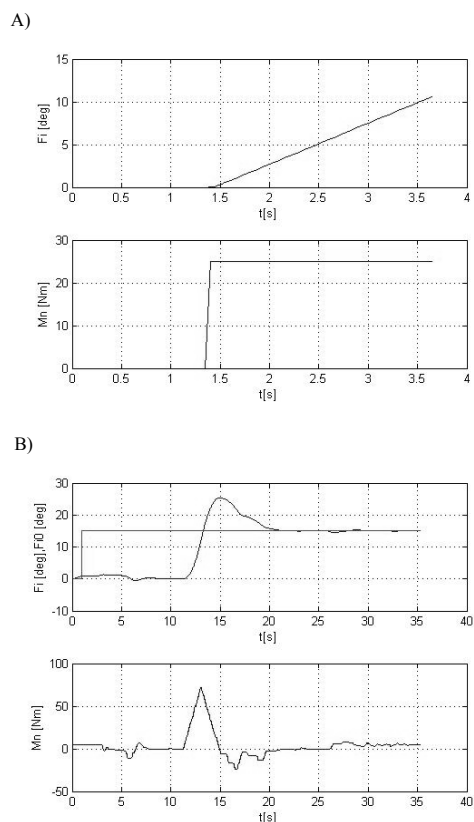
Wykorzystując stanowisko (rys. 6) oraz model układu regulacji w Simulinku (rys. 7), wykonano następujące próby: skokową, sterowanie ręczne oraz sterowanie automatyczne z wykorzystaniem klasycznego algorytmu PID IND. Dzięki zastosowaniu podprogramów w sterowniku PLC, wymienione eksperymenty nie wymagały żadnych zmian w układzie symulowanym, a jedynie wywołanie zmiennej logicznej, odpowiadającej podprogramowi.



Rys. 6. Stanowisko testowania układu sterowania metodą HIL
Fig. 6. Equipment employed to testing control system with HIL procedure



Rys. 7. Okno symulacji w programie Matlab/Simulink
Fig. 7. A simulation window prepared in Matlab/Simulink



Rys. 8. Wybrane wyniki symulacji HIL z modelem pojazdu podwodnego F_i – kąt kursowy, M_n – moment napędowy: A) próba skokowa, B) sterowanie manualne, C) sterowanie automatyczne z algorytmem PID IND
Fig. 8. Selected results of the HIL simulation with a model of the underwater vehicle F_i – course angle, M_n - driving moment: A) step experiment, B) manual control, C) automatic control with the PID IND algorithm

Wyniki eksperymentów potwierdzają prawidłowość wymiany danych między platformą wykonawczą algorytmów sterowania i platformą symulacyjną. Rozwinięciem przedstawionych eksperymentów może być przebadanie odporności algorytmów sterowania na zmieniające się sygnały zakłócające (Simulink umożliwia wprowadzenie również sygnałów, pochodzących z rzeczywistego obiektu) oraz zmieniającą się dynamikę obiektu.

4. Podsumowanie

Uzyskane wyniki potwierdzają możliwość zastosowania protokołu OPC w eksperymentach typu Hardware-in-the-Loop, co ma szczególne znaczenie w przypadku sterowania obiektami w przypadku których awaria, skutkiem nieumiejętnego sterowania przez niedoświadczzonego operatora lub niestabilnego oddziaływanie na człowieka i/lub całkowite zniszczenie obiektu.

Pozytywna weryfikacja działania symulacji HIL z typowym algorytmem przemysłowym tj. PID IND umożliwia prowadzenie badań z algorytmami bardziej złożonymi np. fuzzy z funkcjami samonastrajania oraz adaptacji.

5. Literatura

- [1] OPC Foundation. OPC specification. Dostępne na stronie <http://www.opcfoundation.org>
- [2] Graczyk T.: Selected Aspects of Underwater Remotely Operated Vehicle Design, Foundations of Control and Management Sciences, Politechnika Poznańska, ISSN 1731-2007.
- [3] Jaszczak S., Dramski M., Matejski M.: Symulacja i testowanie układu sterowania ruchem pojazdu podwodnego metodą HIL. Metody Informatyki Stosowanej, Gdańsk, 2008.
- [4] Skrucz P.: Wprowadzenie do testowania układów automatyki, XI International PhD Workshop OWD 2009.

otrzymano / received: 08.05.2010
przyjęto do druku / accepted: 04.06.2010

artykuł recenzowany