

STANOWISKO BADAWCZE WYKORZYSTUJĄCE PLATFORMĘ CZASU RZECZYWISTEGO xPC TARGET W CELU WERYFIKACJI ALGORYTMU STEROWANIA RUCHEM STATKU

Artykuł zawiera opis stanowiska badawczego, znajdującego się w laboratorium Akademii Morskiej w Gdyni, które wykorzystuje platformę czasu rzeczywistego xPC Target ze środowiska Matlab wersja 2011b. We wstępie scharakteryzowano model obiektu zastosowanego w stanowisku badawczym, jakim jest model statku „Blue Lady”. W drugim rozdziale przedstawiono krótki opis pakietu xPC Target. Kolejne kroki, związane z konfiguracją sprzętu i oprogramowania stanowiska badawczego, opisano w rozdziale 3. Natomiast ostatni rozdział przedstawia syntezę regulatora LMI (Linear Matrix Inequalities) oraz wyniki symulacji przeprowadzonych na komputerze w narzędziu symulacyjnym pakietu Matlab, tj. w Simulinku, porównane z wynikami przeprowadzonymi na stanowisku badawczym w czasie rzeczywistym.

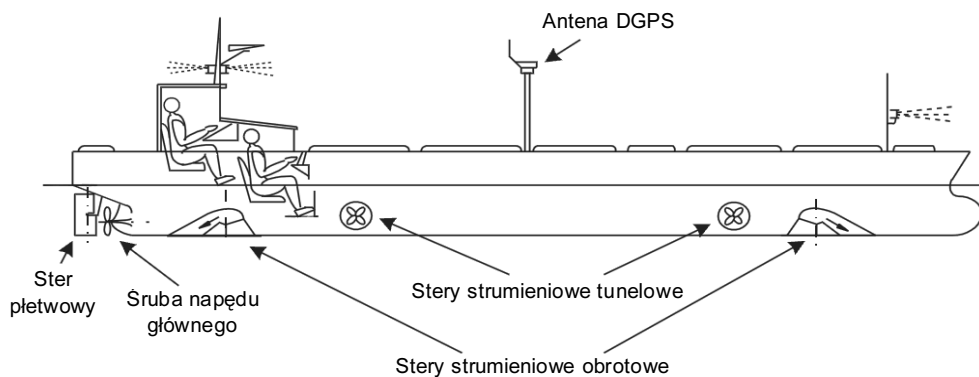
Słowa kluczowe: platforma czasu rzeczywistego xPC Target, jądro czasu rzeczywistego, Linear Matrix Inequalities, stanowisko badawcze, konfiguracja sprzętu, konfiguracja oprogramowania, Matlab 2011b 64-bitowy.

WSTĘP

Stanowisko badawcze znajdujące się w laboratorium Akademii Morskiej w Gdyni zostało zbudowane w celu weryfikacji regulatora w układzie sterowania z obiektem opartym na modelu statku, eksploatowanym w Badawczo-Szkoleniowym Ośrodku Manewrowania Statkami w Kamionce nad jeziorem Silm, niedaleko Iławy [1]. Sprawdzenie, czy jakość oraz sposób sterowania, osiągnane przez zaprojektowany układ, odpowiadają założonym lub przewidywanym warunkom, stanowi rozstrzygający test całego procesu syntezy regulatora. W tym przykładzie zastosowano symulację procesu z nieliniowym modelem statku w czasie rzeczywistym, co stanowi przygotowanie, po osiągnięciu poprawnych wyników, do testowania własnych obliczeń na statku treningowym w Iławie na jeziorze Silm. Dokładny opis obiektu znajduje się w publikacjach [3, 7], pokrótce statek treningowy o nazwie „Blue Lady” jest modelem zbudowanym w skali 1:24 na podstawie statku typu VLCC – (ang. *Very Large Crude Carrier*), czyli superzbiornikowiec, który miał służyć do przewozu surowej ropy. Napęd główny stanowi silnik prądu stałego napędzający poprzez przekładnię prawoskrętną, 5-łopatkową śrubę. Model wyposażony jest również w dwa pędniki tunelowe i dwa pędniki obrotowe, symulujące

pracę holowników (60-tonowych). „Blue Lady” posiada także dwuosobowe stanowisko kontrolne służące do sterowania statkiem.

Poniżej na rysunku 1 przedstawiono sylwetkę statku.



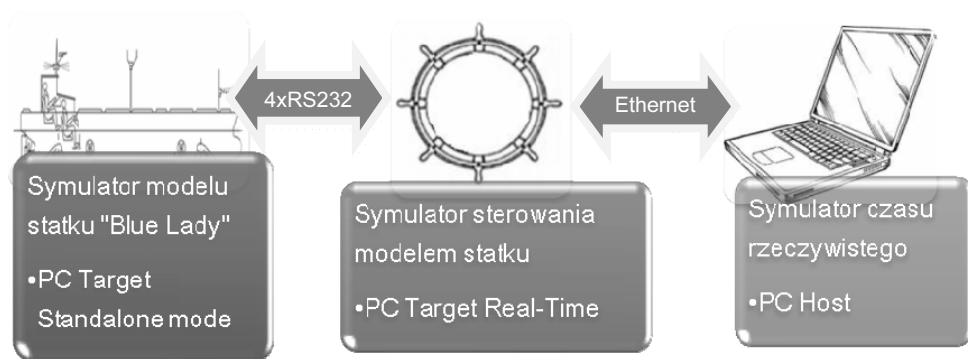
Rys. 1. Sylwetka statku „Blue Lady” z ukazaniem dwóch miejsc dla osób sterujących ruchem oraz z umieszczoną anteną DGPS

Fig. 1. „Blue Lady” ship model side view with 2 person operators cabin and GPS antenna

W stanowisku badawczym wprowadzono taki układ połączeń pomiędzy systemem sterującym a obiektem sterowania, aby był identyczny z tym, który jest eksploatowany w Iławie. Wykorzystano te same schematy blokowe środowiska Simulink oraz te same kompilatory i procedury uruchamiania, dotyczące sterowania w czasie rzeczywistym. Jako platformy użyto środowiska o nazwie „xPC Target” z pakietu Matlab, opisanego w dalszej części artykułu. Istotne było sprawdzenie na tym etapie symulacji, aby wszystkie elementy schematu sterowania kompilowały się do kodu wykonawczego, używanego podczas sterowania w czasie rzeczywistym.

1. „XPC TARGET” – STEROWANIE W CZASIE RZECZYWISTYM

Sterowanie w czasie rzeczywistym w tym przykładzie zrealizowano, stosując metodę *Hardware-in-the-Loop*, w skrócie HiL (rys. 2) [5]. Jest to metoda, która pozwala testować funkcjonalność całych układów sterowania, tj. regulatorów, układów pomiarowych czy wykonawczych oraz komunikację pomiędzy elementami systemu. Poprzez taką funkcjonalność testowanie regulatora w wirtualnym środowisku pozwala na weryfikowanie pomiarów i poprawienie błędów jeszcze przed ostateczną próbą na jeziorze Silm.



Rys. 2. Przykładowa realizacja wykorzystująca xPC Target w celu symulacji sterowania statkiem w czasie rzeczywistym

Fig. 2. Example of the use of xPC Target to simulate ship movement control in real time

Sterowanie w czasie rzeczywistym dla stanowiska badawczego zrealizowano, stosując dwa komputery stacjonarne PC oraz jeden komputer przenośny:

- symulator modelu statku „Blue Lady” – „PC Target Standalone mode”;
- symulator układu przeznaczonego do sterowania modelem statku – „PC Target Real-Time”;
- symulator czasu rzeczywistego (komputer przenośny) przeznaczony do obserwacji wyników – „PC Host”.

Na pierwszym i drugim symulatorze zainstalowane zostało jądro czasu rzeczywistego, wykorzystujące pakiet xPC Target z biblioteki *Real Time WorkShop* (RTW) ze środowiska Matlab (wersja 2011b), a na jednostce przenośnej („PC Host”) zainstalowano system operacyjny 64-bitowy Microsoft Windows 7 wraz z środowiskiem Matlab. Symulacja w tak zorganizowanym stanowisku laboratoryjnym pozwoliła na uruchamianie i zatrzymywanie procesu sterowania oraz na zmiany wartości liczbowych wybranych parametrów w trakcie trwania symulacji. Układy sterowania zostały zbudowane jako schematy blokowe w środowisku Simulink, z wykorzystaniem standardowych bloków z bibliotek i tzw. S-funkcji (bloków definiowanych przez użytkownika). Funkcje zostały zapisane jako pliki o rozszerzeniu .mex. Przy czym, jeżeli pakiet Matlab był zainstalowany na 64-bitowym systemie operacyjnym (tak jak w tym przypadku), należało przekompilować pliki .mex na .mex64, co udało się przy zastosowaniu odpowiedniego środowiska programistycznego (w tym przypadku Microsoft Visual Studio 2008). Komputery „Target PC” nie wymagały instalacji żadnego systemu operacyjnego i uruchamiane były specjalną płytą startową. Dzięki temu tzw. jądro czasu rzeczywistego, zawarte w dwóch symulatorach („PC Target Standalone mode” i „PC Target Real-Time”) po otrzymaniu z komputera przenośnego „Host PC” kodu wykonywalnego mogło sterować obiektem.

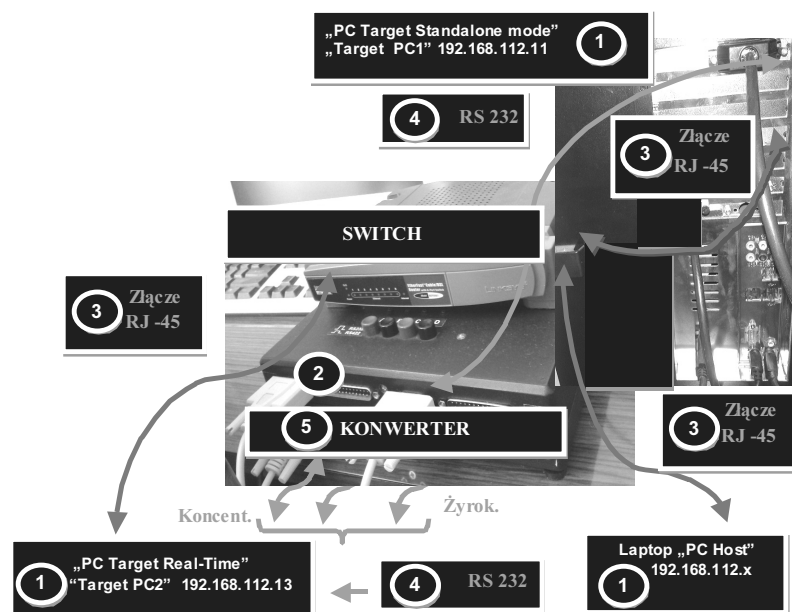
2. OPIS KONFIGURACJI STANOWISKA BADAWCZEGO

Uruchomienie stanowiska badawczego, znajdującego się w laboratorium Akademii Morskiej w Gdyni, wymagało zastosowania odpowiedniej konfiguracji sprzętu i oprogramowania.

2.1. Organizacja sprzętowa (rys. 3)

Do uruchomienia stanowiska badawczego wykorzystano:

- trzy jednostki komputerowe, w tym dwie stacjonarne nazwane jako „PC1 Target Czarny” i „PC2 Target BL” i trzecią – przenośną „PC Host” (oznaczoną na rysunku jako pozycja nr 1);
- urządzenie sieciowe – *switch* (oznaczone na rysunku 3 jako pozycja nr 2);
- trzy kable sieciowe, bez przeplotu (nieskrosowane), o złączu RJ-45 (oznaczone na rysunku 3 jako pozycja nr 3);
- 4 kable w standardzie RS-232 (oznaczone na rysunku nr 3 jako pozycja nr 4);
- konwerter RS-232 (oznaczony na rysunku 3 jako pozycja nr 5).



Rys. 3. Fizyczna konfiguracja stanowiska badawczego wykorzystanego do symulacji sterowania modelem statku „Blue Lady” w czasie rzeczywistym

Fig. 3. Configuration of lab station used to simulate „Blue Lady” ship model movement in real time



Rys. 4. Zdjęcie stanowiska badawczego wykorzystanego do symulacji sterowania modelem statku „Blue Lady” w czasie rzeczywistym

Fig. 4. Layout of lab station used to simulate "Blue Lady" ship model movement in real time

2.2. Organizacja oprogramowania

Zastosowano tu:

- system operacyjny 64-bitowy Microsoft Windows 7, zainstalowany na komputerze przenośnym;
- pakiet MATLAB 2011b, wersja 64-bitowa, zainstalowany na komputerze przenośnym;
- jądro czasu rzeczywistego uruchamiające się z płyt startowych na obu komputerach stacjonarnych, wykorzystujących pakiet xPC Target z biblioteki *Real Time WorkShop* (RTW) środowiska Matlab 2011b;
- pakiet SDK (*Software Development Kit*) dla systemu operacyjnego Microsoft Windows 7, zainstalowany na komputerze przenośnym;
- środowisko programistyczne *Microsoft Visual Studio* 2008, zainstalowane na komputerze przenośnym.

2.3. Konfiguracja oprogramowania

Poniżej zaprezentowano kroki, jakie należy kolejno wykonać w celu uruchomienia przez użytkownika stanowiska badawczego. Są to:

a) konfiguracja sieci (komputer przenośny – "Host PC" ze switchem)

Ustawienia karty sieciowej w komputerze przenośnym:

- Adres IP :192.168.112. x.
- Maska podsieci : bez zmian
- Brama domyślna :192.168.112. x.
(x – mogą pozostać numery istniejące);

b) konfiguracja pakietu xPC Target dla symulatora **Target PC2**

Po uruchomieniu pakietu Matlab wersji 2011b należało skonfigurować pakiet xPC Target, co dotyczyło załadowania adresów urządzeń symulujących pracę obiektu i regulatora.

Poniżej przedstawiono kroki, jakie ma wykonać użytkownik w celu poprawnej komunikacji urządzeń pracujących w czasie rzeczywistym.

W oknie komend pakietu Matlab 2011b pojawią się między innymi komendy:

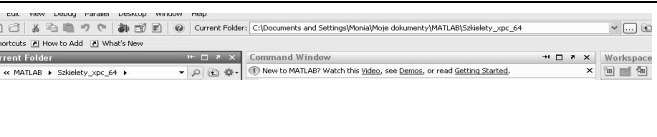
Target PC2 TcpIpTargetAddress 192.168.112.13

Target PC1 TcpIpTargetAddress 192.168.112.11

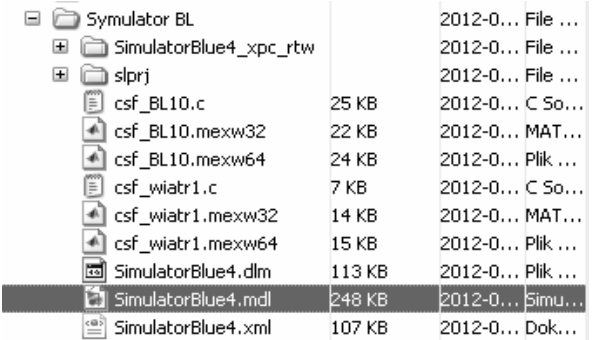

Co oznacza, że po zapytaniu „PC Host” o kolejne podłączone w sieci urządzenia, zostały wykryte dwa adresy IP. Po poprawnej komunikacji urządzeń zewnętrznych należy załadować odpowiednio dedykowane schematy blokowe dla obiektu, zaprojektowane w pakiecie Simulink, które w tym folderze znajdowały się pod nazwą symulator „BL”.

Tabela 1. Konfiguracja pakietu xPC Target dla symulatora Target PC2

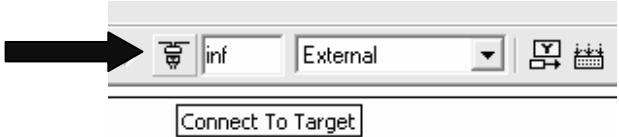
Table. 1. Configuration of xPC Target for Target PC2 simulator

Numer kroku	Opis czynności	Zrzut z ekranu monitora
1	Wpisać w okno komend, w pakiecie Matlab, polecenie: <i>xpcsetCC('setup')</i> , w oknie pojawi się komunikat o wyborze kompilatora, następnie wybrać numer odpowiadający Visual Studio 2008	<pre>>> xpcsetCC('setup')</pre>
2	załadować plik, nazwany <i>xpc_na_2_komp.m</i> , zawierający kolejno polecenia wypisane na zrzucie ekranu edytora obok	<pre>5 - tgs=xpctarget.targets 6 - tgs.makeDefault('TargetPC1') 7 - env=tgs.Item('TargetPC1') 8 - env.HostTargetComm='TcpIp' 9 - env.TcpIpTargetAddress='192.168.112.11' 10 - env.TcpIpGateway='192.168.112.254' 11 12 - env2=tgs.Item('TargetPC2') 13 - env2.HostTargetComm='TcpIp' 14 - env2.TcpIpTargetAddress='192.168.112.13' 15 - env2.TcpIpGateway='192.168.112.254'</pre>
3	Przejsć do właściwej ścieżki, w której chce się pracować, w górnym oknie Matlab	

cd. tabeli 1

Numer kroku	Opis czynności	Zrzut z ekranu monitora
4	Wpisać w oknie poleceń: <code>xptest</code> (Target PC2), podczas testu uruchomi się okno z testowanym układem, należy go NIE WYŁĄCZAĆ	<pre>>> xptest ### xPC Target v5.1 Test Suite ### Host-Target interface is: TCP/IP (Ethernet) ### Test 1, Ping target PC 'TargetPC1' using system ping: ... OK ### Test 2, Ping target PC 'TargetPC1' using xpctesttargetping: ... OK ### Test 3, Software reboot the target PC 'TargetPC1': .. OK ### Test 4, Build and download an xPC Target application using model x ### Test 5, Check host-target command communications with 'TargetPC1': ### Test 6, Download a pre-built xPC Target application to target PC ' ### Test 7, Execute the xPC Target application for 0.2s: ... OK ### Test 8, Upload logged data and compare with simulation results: .. ### Test Suite successfully finished</pre>
5	Wpisać w oknie poleceń: <code>tgs.makeDefault</code> (Target PC2)	<pre>>> tgs.makeDefault('TargetPC2') xPC Target Environment Object Name : TargetPC2 TargetRAMSizeMB : Auto MaxModelSize : 1MB SecondaryIDE : off NonPentiumSupport : off MulticoreSupport : off LegacyMultiCoreConfig : off</pre>
6	Otworzyć plik o rozszerzeniu <code>.mdl</code> , w tym wypadku <code>SimulatorBlue4.mdl</code> z katalogu Symulator BL	
7	Wcisnąć w Simulinku przycisk o nazwie <i>Incremental build</i>	

cd. tabeli 1

Numer kroku	Opis czynności	Zrzut z ekranu monitora
8	Jeżeli plik zostanie poprawnie załadowany, wcisnąć w Simulinku przycisk o nazwie <i>Connect to Target</i>	
9	Wcisnąć przycisk <i>PLAY</i> znajdujący się obok przycisku <i>Connect to Target</i> , w oknie Matlaba pojawi się następujący komunikat	<pre> ### Created DLM ..\SimulatorBlue4.dlm ### Successful completion of build procedure for model: Si ### Looking for target: TargetPC2 ### Download model onto target: TargetPC2 ### Create xPC Object tg xPC Object Connected = Yes Application = SimulatorBlue4 Mode = Real-Time Single-Tasking Status = stopped CPUOverload = none </pre>

c) konfiguracja pakietu xPC Target dla symulatora **Target PC1**

Przy ładowaniu regulatora na drugim symulatorze należy wykonać czynności analogiczne do punktu 2.3b):

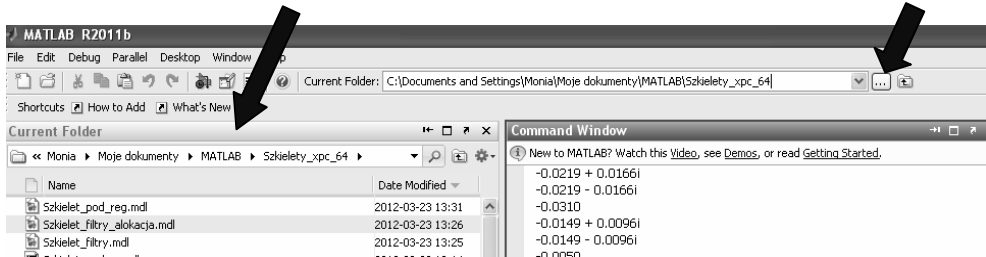
- przejść do właściwej ścieżki, w której chce się pracować, w górnym oknie Matlaba;
- wpisać w oknie poleceń: `xpctest (TargetPC1)`, podczas testu uruchomi się okno z testowanym układem, należy go NIE WYŁĄCZAĆ;
- wpisać w oknie poleceń: `tg.makeDefault (TargetPC1)`;
- otworzyć plik o rozszerzeniu `.mdl`, w tym wypadku np. `Szkielet_filtr_ _alokacja.mdl` z katalogu `Szkielety_xpc_64`;
- wcisnąć w Simulinku przycisk o nazwie *Incremental build*;
- jeżeli plik zostanie poprawnie załadowany, wcisnąć w Simulinku przycisk o nazwie *Connect to Target*;
- wcisnąć przycisk *PLAY* znajdujący się obok przycisku *Connect to Target*;

d) błędy występujące podczas komunikacji

W przypadku braku komunikacji między poszczególnymi urządzeniami należy sprawdzić:

- poprawność fizycznego połączenia (rys. 3);
- poprawność adresu IP w komputerze pełniącym rolę "Host PC" (pkt 2.3.);

- poprawność rozszerzeń plików, jeżeli symulacje wykonywane są w środowisku Matlab w wersji 64-bitowej, to pliki o rozszerzeniu powinny posiadać rozszerzenie .mex64 (zamiast rozszerzenia .c);
- poprawność zainstalowanego i używanego oprogramowania (pkt 2.2.);
- poprawność używanego folderu roboczego, czy jest załadowany w ścieżce dostępu Matlaba.



W razie dalszego braku komunikacji z którymś z symulatorów należy zresetować go, wciskając przycisk *Reset* na czarnej skrzynce (na rys. 3, nr 1), następnie znowu załadować w oknie Matlaba plik `tgsmakeDefault(TargetPC1)` lub `tgsmakeDefault(TargetPC2)` i ponownie dokonać konfiguracji zgodnie z powyższymi krokami.

3. SYNTEZA REGULATORA

Symulacja w czasie rzeczywistym ma pomóc w analizie otrzymanych wyników, gdzie obiektem w tym przypadku jest model statku treningowego „Blue Lady”, opisany w [3, 7]. Synteza regulatora została oparta na metodzie LMI (*Linear Matrix Inequality*), która dokładniej opisana jest w pozycjach [2, 4, 6, 8, 9]. Pierwszym krokiem w metodzie LMI jest sprawdzenie warunku stabilności (patrz równanie 1):

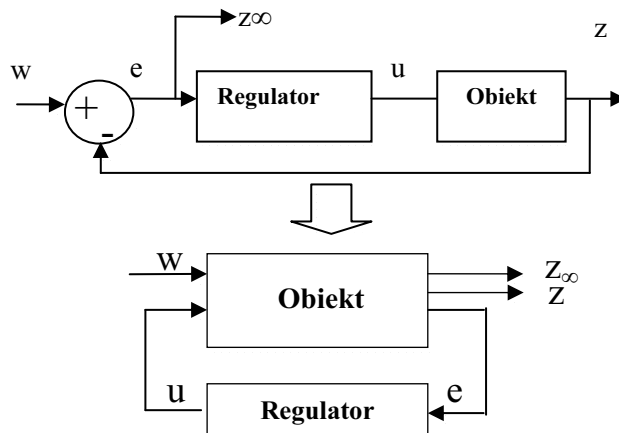
$$A^T P + P A < 0, \quad P = P^T \succ 0 \quad (1)$$

Spełnienie warunku $P = P^T \succ 0$ określa, czy układ regulacji jest stabilny i tym samym rozwiązuje problem istnienia, czyli problem znalezienia symetrycznej i dodatnio określonej macierzy P dla danej macierzy stanu układu A (gdzie $\succ 0$ oznacza, że wszystkie wartości własne macierzy P są dodatnie). Po określeniu macierzy „ P ” należy zdefiniować liczbę sygnałów wchodzących i wychodzących z obiektu i regulatora. W przypadku rozpatrywania układu zamkniętego równania stanu będą mieć następującą postać:

$$\begin{aligned} \dot{x} &= A x + B_u u + B_w w & z_\infty &= C_\infty x + D_{\infty u} u + D_{\infty w} w \\ z &= C_z x + D_{zu} u + D_{zw} w & e &= C_e x + D_{eu} u + D_{ew} w \end{aligned} \quad (2)$$

gdzie:

- x – zmienna stanu (gdzie wektor ma wymiary $n \times 1$),
- e – uchyb,
- z, z_{∞}, z_2 – sygnały wyjściowe układu, gdzie wektor z_{∞} definiuje normę dla obiektu opisanego jako $T_{w \rightarrow z_{\infty}}(s)$, a wektor z_2 definiuje normę dla obiektu opisanego jako $T_{u \rightarrow z_2}(s)$,
- u – wektor sygnału wyjściowego regulatora (wymiar $r \times 1$),
- w – wektor sygnału wejściowego układu (wymiar $m \times 1$),
- A – macierz stanu o wymiarach $n \times n$,
- B_u, B_w – macierz wejść o wymiarach $n \times r$,
- $C_z, C_{zu}, C_{\infty}, C_e$ – macierz wyjść o wymiarach $m \times n$,
- $D_{eu}, D_{ew}, D_{\infty u}, D_{zu}, D_{\infty u}$ – macierz transmisyjna o wymiarach $m \times r$.



Rys. 5. Struktura układu zamkniętego

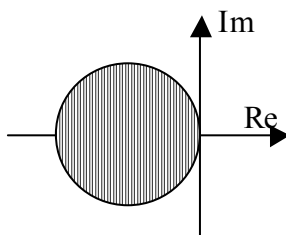
Fig. 5. Structure of a closed loop system

Po wykonaniu powyższych kroków w znalezieniu macierzy „ P ”, co oznacza, że bieguny układu zamkniętego leżą w lewej półpłaszczyźnie, należy zaprojektować właściwości dynamiczne zamkniętego układu regulacji poprzez lokowanie biegunów w określonym przez projektanta obszarze płaszczyzny zmiennej zespolonej s . W tym przykładzie zastosowano zdefiniowane następujące wzory i obszary (patrz równanie 3):

$$C_{stab} = \{s \in C \mid |s + q| < r\} \Leftrightarrow \begin{bmatrix} -r & s + q \\ - & -r \\ s + q & \end{bmatrix} < 0 \quad (3)$$

gdzie:

- r – promień okręgu,
- q – punkt środka okręgu.



Rys. 6. Lewa półpłaszczyzna z okręgiem

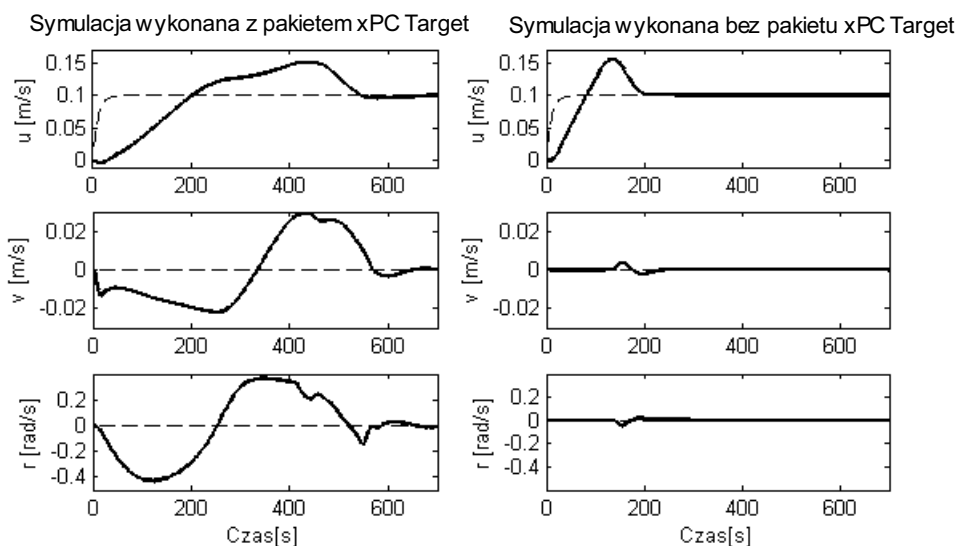
Fig. 6. Left half plane with a circle

Wybrane drogą eksperymentalną parametry określonego obszaru wynoszą:

- promień okręgu – $r = 0,9803$;
- punkt środkowy – $q = -1$

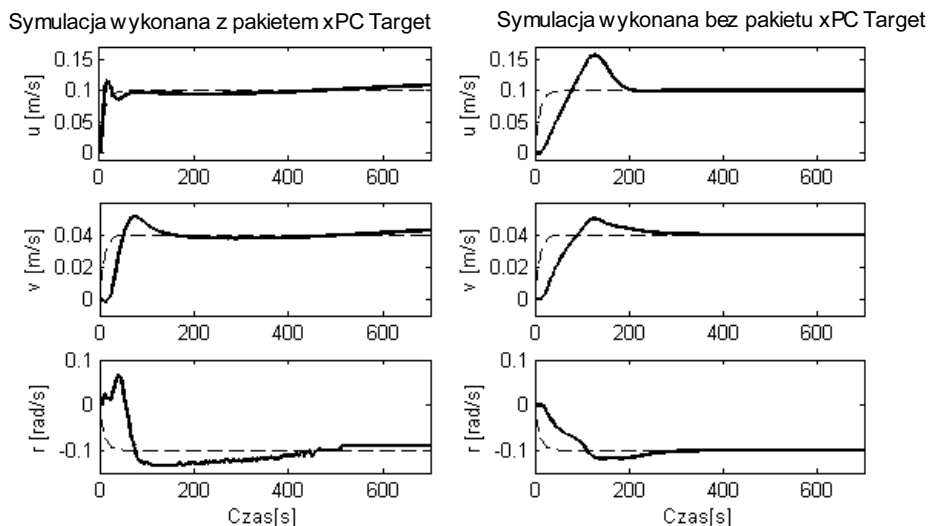
Po wykonaniu szeregu przekształceń i obliczeń opisanych w pozycjach [6, 8] w celu dokonania syntezy regulatora dla zamkniętego układu regulacji wykonano kilka pomiarów w symulatorze pakietu Simulink oraz na stanowisku badawczym wykorzystującym jądro czasu rzeczywistego.

Wyniki symulacji przeprowadzono na symulatorze rzeczywistym i zamieszczono je po lewej stronie rysunku 7 z tytułem „Symulacja wykonana z pakietem xPC Target”. Natomiast wyniki zamieszczone po prawej stronie rysunku 7 z tytułem „Symulacja wykonana bez pakietu xPC Target” wykonano na symulatorze w komputerze przenośnym.



Rys. 7. Wyniki symulacji przeprowadzone dla jednej prędkości wzdłużnej statku $u = 0,1$ [m/s] i pozostałych: prędkości poprzecznej $v = 0$ [m/s] i kątovej $r = 0$ [rad/s]

Fig. 7. Simulation results only with longitudinal velocity $u = 0,1$ [m/s], with transverse velocity $v = 0$ [m/s] and angular velocity $r = 0$ [rad/s]



Rys. 8. Wyniki symulacji przeprowadzone dla ustawień prędkości wzdłużnej statku $u = 0,1$ [m/s], prędkości poprzecznej $v = 0,04$ [m/s] i prędkości kątowej $r = -0,1$ [rad/s]

Fig. 8. Simulation results with longitudinal velocity $u = 0,1$ [m/s], transverse velocity $v = 0,04$ [m/s] and angular velocity $r = -0,1$ [rad/s]

Przeprowadzone wyniki zamieszczone na rysunku 7 dotyczą ustawień dla prędkości wzdłużnej $u = 0,1$ m/s i pozostałych prędkości równych zero. Widać różnicę między symulacjami wykonanymi na komputerze przenośnym bez jądra czasu rzeczywistego, umieszczonymi po prawej stronie rysunku, a symulacją w czasie rzeczywistym, zamieszczoną po lewej stronie rysunku. Długi czas ustalania równy około 600 s może wiązać się z uwzględnieniem przetworników analogowo-cyfrowych, zamieszczonych w celu przeliczenia sił dla pędników modelu statku.

Na drugim rysunku 8 zamieszczono wyniki dla ustawień trzech prędkości, tj. wzdłużnej $u = 0,1$ m/s, poprzecznej $v = 0,04$ m/s i kątowej $r = -0,1$ rad/s. Wyniki wykonane na symulatorze jądra czasu rzeczywistego są gorsze od tych wykonanych na komputerze przenośnym, ponieważ uwzględniają dynamikę modelu statku w czasie rzeczywistym.

PODSUMOWANIE

Stanowisko badawcze znajdujące się w laboratorium Akademii Morskiej w Gdyni jest potrzebne w celu weryfikacji syntezy regulatora przed wykonaniem badań na rzeczywistym modelu statku, zamieszczonym na jeziorze Silm. Weryfikacja przeprowadzona na stanowisku badawczym pozwala na zaoszczędzenie wielu godzin, jakie należałoby poświęcić na badania statku treningowego „Blue Lady” na jeziorze Silm, dodatkowo zaś jest maksymalnie zbliżona do realiów sterowania statkiem.

LITERATURA

1. Badawczo-Szkoleniowy Ośrodek Manewrowania Statkami w Kamionce, <http://www.ilawa.shiphandling.com.pl/training.html>.
2. Boyd S., EL Ghaoui L., Feron E., Balakrishnan V., *Linear Matrix in System and Control Theory*, SIAM, Philadelphia 1994.
3. Gierusz W., *Synteza wielowymiarowych układów sterowania precyzyjnego ruchem statku z wykorzystaniem wybranych metod projektowania układów odpornych*, Wydawnictwo Akademii Morskiej w Gdyni, Gdynia 2005.
4. Koziński W., *Projektowanie regulatorów. Wybrane metody klasyczne i optymalizacyjne*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2004.
5. Pomirski J., Rak A., Gierusz W., *Control system for trials on material ship model*, Polish Maritime Research, Vol. 19, Gdańsk 2012.,
6. Rybczak M., *Linear Matrix Inequalities in multivariable ship's steering*, Polish Maritime Research, Vol. 19, Gdańsk 2012.
7. Rybczak M., *Wykorzystanie języka VRML do modelowania środowiska symulacyjnego ruchu modeli statków*, Gdynia 2006.
8. Rybczak M., Gierusz W., *Linear Matrix Inequalities in multivariable ship's steering*, Polish Journal of Environmental Studies, Gdynia 2010.
9. Scherer C., Wieland S., *Linear Matrix Inequalities in Control*, Delft University of Technology, Delft 2004.

LAB STATION FOR VERIFYING SHIP CONTROL ALGORITHM WITH THE USE OF REAL TIME PLATFORM XPC TARGET

Summary

This article is a description of a lab station in Maritime Academy in Gdynia which uses a real time platform, the xPC Target from Matlab 2011b software in 64-bit version. Preface describes, in general, the lab station where the controlled system is a "Blue Lady" ship model. First chapter contains a short description of the xPC Target software. Second chapter shows steps required to configure hardware and software of the lab station. Third and final chapter is the LMI (Linear Matrix Inequalities) controller synthesis and simulation results, where results obtained from a computer simulation using Matlab's simulation tool Simulink are compared with results obtained from the real time lab station.

Keywords: *real time platform xPC Target, real time kernel, Linear Matrix Inequalities, lab station, hardware configuration, software configuration, Matlab 2011b 64-bit.*