MODEL PREDICTIVE CONTROL ALGORITHM VERIFICATION WITH THE USE OF REAL TIME XPC TARGET PLATFORM

Anna Miller, Monika Rybczak
Gdynia Maritime University
Poland

ABSTRACT

There is a need to check the effects of automatic regulation system, before it is used in real conditions when sailing on Silm Lake. Simulink Real-Time platform, which is a component of Matlab software, is designed for simulation of automatic control systems working on the basis of real objects. In the simulations nonlinear mathematical model of real floating training ship is used as a control object. The model contains modeled filters used for longitudinal, transversal and rotational speeds on the basis of position change and transducers for thrusters and steering gear. Above mentioned simulations are conducted in real time, because it allows for taking into account time of the signals propagation and digital signal processing effects. This article presents the real time lab station system configuration. Simulation results for Model Predictve Control system are described and presented.

Keywords: Simulink Real-Time platform, real-time simulations, Model Predictive Control.

INTRODUCTION

In this article modifications to the test lab station, which was described in [1] and is located in Gdynia Maritime University are presented. The lab station was modernized and a software update was released, to expand simulation capabilities. The simulation system is very important, as it allows to verify method and control quality in predetermined conditions, before the automatic control system was used on board of the training ship model. The intention of this publication is to assemble information that enable proper software and hardware configuration while working with Matlab 2014a software. The lab station is used to verify the results of computer simulations taking into account a unique control system with controlled plant mounted on a real floating training ship. This ship is exploited in Ship Handling Research and Training Centre at Ilawa owned by the Foundation for Safety of Navigation and Environment Protection [2]. In practice this test bed is used for real time simulation of nonlinear ship's model behavior when a new control algorithm is being verified. Subsequently, after reaching the correct results, conduction on the real floating ship model on Silm Lake was tested. Use of this system allows to shorten, the duration of conducted tests.

Training ship model "Blue Lady" was built in 1:24 scale, based on a real VLCC (Very Large Crude Carrier) ship used for crude oil carriage. The main drive consists of DC motor, which propels a five-blade main propeller through a gear. This model is also equipped with bow and stern rotational thrusters, that simulate two 60-tonnes tugs' operation. "Blue Lady" also has a two-seat control stand, used for ship steering. This model has been described in detail in [3] and [4]. Training ship's silhouette is presented in Fig. l.

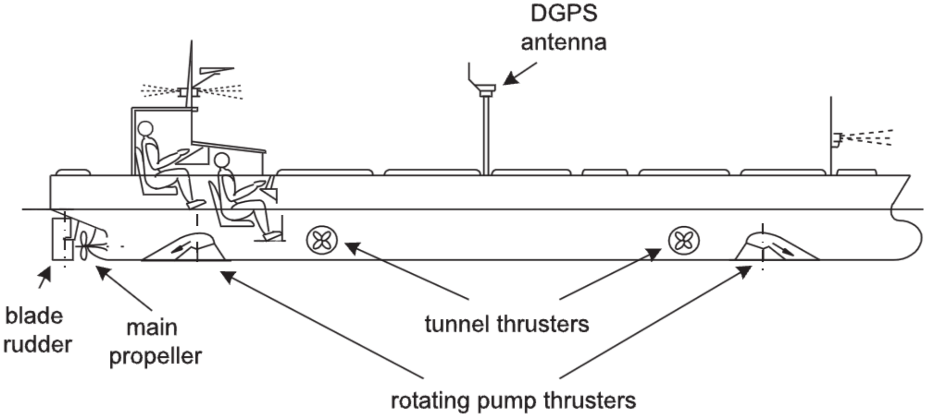


Fig. 1. "Blue Lady" silhouette, with thrusters, DGPS antenna and control stand location

In the test stand connections between control system and controlled object are identical to the ones used in real training ship model in Ilawa. The same controller schemes designed in Simulink environment, compilers and real-time control procedures are used. Simulink Real-Time (part of Matlab 2014a software) was used as a software platform and described later in the article. During the lab station preparation it was important to verify whether all elements of control scheme where compiled to executable code, that was used during realtime simulations.

1. REAL-TIME CONTROL WITH THE „SIMULINK REAL-TIME" SOFTWARE

The simulation test stand is an embedded system, which is a computer system with dedicated function of working in real time. The control algorithm was built on a personal computer (PC) working on Windows 7 operating system and Matlab

2014a software – called a host. Algorithm execution was held on a computer working in real-time – called a target. Real-time system is a computer system, where all computations are conducted in parallel with the course of an external process and their aim is to supervise, control and timely react to events taking place in the process. This system is described in a detailed way in [5]. In the described simulation system Hardware-in-the-Loop (HiL) model was used [6]. This model enables testing of the entire control systems functionality, like controllers, measurement systems and also communication between all parts of these systems. As a result testing of the control system in a virtual environment allows to verify for measurements and to correct errors before the final test in real conditions is being conducted. The described system relates to research carried out on Silm Lake. An example of a real-time system working with HiL model is presented in Fig. 2.

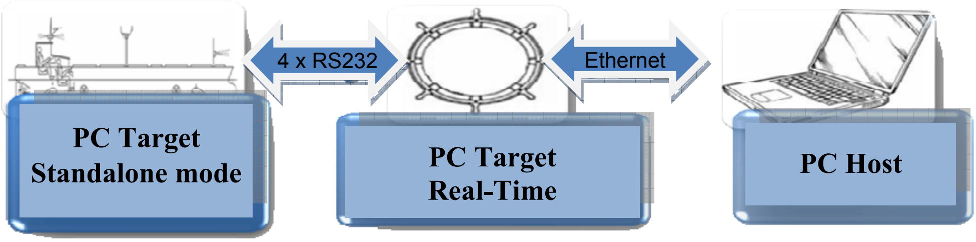


Fig. 2. An exemplary system setup using Simulink Real-Time to simulate a ship's control system in real-time

2. TEST BENCH CONFIGURATION DESCRIPTION

Launching of the simulation lab station in Gdynia Maritime University's laboratory was realized with the use of the appropriate hardware and software.

Modified real-time simulation test stand presented in Figure 3 consists of:

- training ship model „Blue Lady" simulator – „PC Target Standalone mode";
- ship automatic control system simulator – „PC Target Real-Time";
- real-time simulator, PC computer intended for results observation – „PC Host";
- network device - switch;
- RS-232 standard cables and converter.

On the training ship model „Blue Lady" and on the ship automatic control system simulators a real-time kernel package has been installed. The package xPC Target was used, which is a part of Real-Time Windows Target Matlab 2014a library. "PC Host" is a personal computer working on Windows 7 operating system and with Matlab 2014a installed. There is a need to install Software Development Kit

(SDK_ package for Microsoft Windows 7 operating system and Microsoft Visual Studio 2008 – C Language compiler installed on laptop – "Host PC". Simulation in the test stand is organized in a way that allows to start and stop the control process and to change selected parameters values during the simulation. Detailed information on the system is presented in [1]."Target PC" computers do not require any operating system installed a special bootable CD prepared in Matlab 2014a environment is used instead. As a result, the so-called real-time kernel, embedded in two simulators (PC Target Standalone mode" and „PC Target Real-Time") constituted the simulation system and after receiving the executable code could control the virtual object a mathematical model of VLCC Blue Lady.

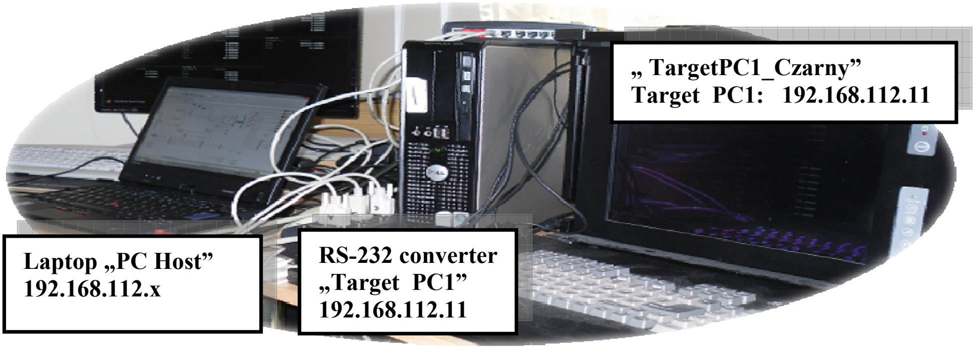


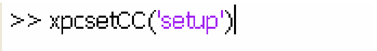
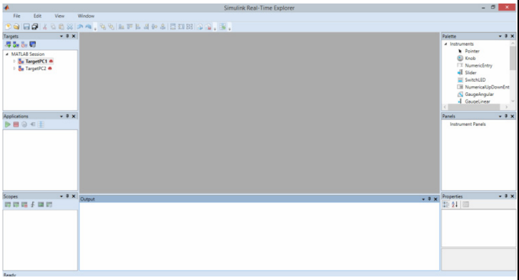
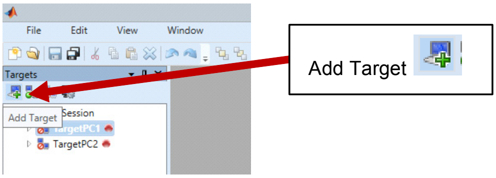
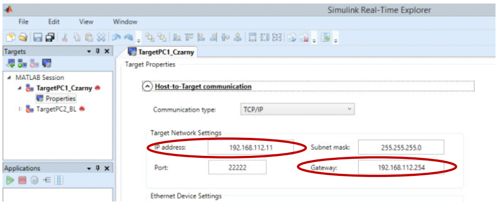
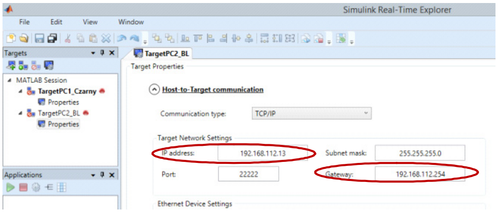
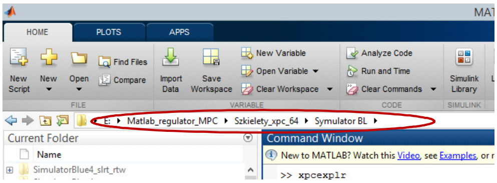
Fig. 3. Test stand physical configuration – training ship "Blue Lady" motion real-time simulation

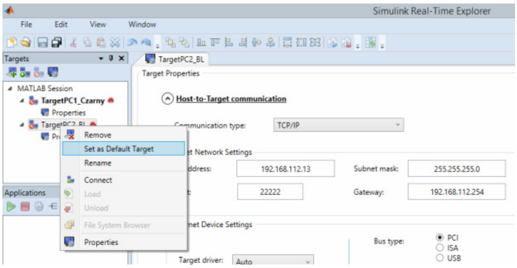
In order to run the simulation in real-time, it is necessary to correctly configure both network connections between elements of the system and 'Simulink Real-Time' Matlab 2014a software package. By following the steps described below, it is possible to conduct real-time simulations of an automatic control system working with a plant – the real floating training ship Blue Lady.

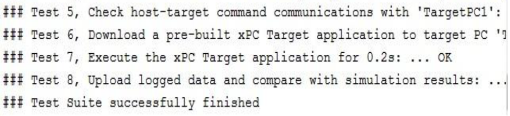
Before the simulations are conducted there is a need to configure network connection on the "Host PC". It communicates with the rest of system using local network, where IP adress is 192.168.112.x (x – arbitrary number) ang gateway 192.168.112.254.

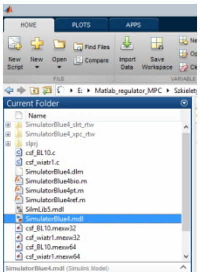
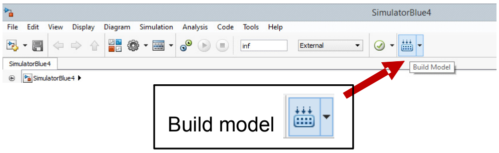
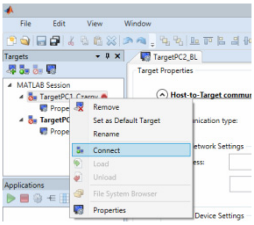
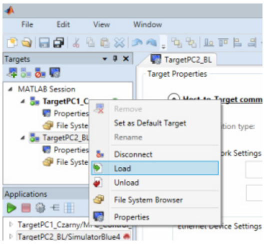
After Matlab 2014a version is run, Simulink Real-Time package hast to be configured. This procedure enables to set plant and controller simulator IP addresses as well as compilers configuration. Steps that should be followed, in order to ensure proper communication between the devices working in real-time, are presented below.

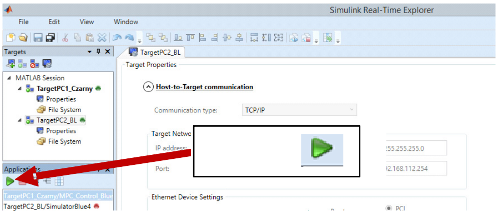
Table 1. Steps required for proper real-time simulation system configuration

Step number	Detailed step description	Screenshot
1	In Matlab Command Window type: *xpcsetCC('setup')* command, in this windo. A number corresponding to Microsoft Visual Studio 2008 should be selected.	 `>> xpcsetCC('setup')`
2	Run Simulink Real-Time Explorer that enables fast configuration of devices working in real-time, command: *xpcexplr.*	 Simulink Real-Time Explorer File Edit View Window Targets MATLAB Session TargetPC1 TargetPC2 Applications Scopes Output Palette Instruments Pointer Knob NumericEntry Slider SwitchLED NumericalUpDownEnt GaugeAngular GaugeLinear Panels Instrument Panels Properties Ready
3	In 'Targets' window add a second target using the 'Add target' option.	 File Edit View Window Targets Add Target Session TargetPC1 TargetPC2 Add Target
4	Configure **'TargetPC1_Czarny'**, which is the controller simulator working in real-time, for guidance use the screenshot on the right; IP address: **192.168.112.11**,	 Simulink Real-Time Explorer File Edit View Window Targets MATLAB Session TargetPC1_Czarny Properties TargetPC2_BL Applications Target Properties Host-to-Target communication Target Network Settings Ethernet Device Settings Communication type: TCP/IP IP address: 192.168.112.11 Subnet mask: 255.255.255.0 Port: 22222 Gateway: 192.168.112.254
5	Configure **'TargetPC2_BL'**, which is the Blue Lady ship simulator working in real-time, for guidance use the screenshot on the right; IP address: **192.168.112.13**	 Simulink Real-Time Explorer File Edit View Window Targets MATLAB Session TargetPC1_Czarny Properties TargetPC2_BL Properties Applications Target Properties Host-to-Target communication Target Network Settings Ethernet Device Settings Communication type: TCP/IP IP address: 192.168.112.13 Subnet mask: 255.255.255.0 Port: 22222 Gateway: 192.168.112.254
6	Change path to the one where Simulink schemes are saved (select as a Matlab working folder), in the upper Matlab window.	 MAT HOME PLOTS APPS New Script New Open Find Files Compare Import Data Save Workspace New Variable Open Variable Clear Workspace Analyze Code Run and Time Clear Commands Simulink Library FILE VARIABLE CODE SIMULINK Current Folder Name SimulatorBlue4_slrt_rtw Command Window E: ▸ Matlab_regulator_MPC ▸ Szkielety_xpc_64 ▸ Symulator BL ▸ New to MATLAB? Watch this Video, see Examples, or r `>> xpcexplr`

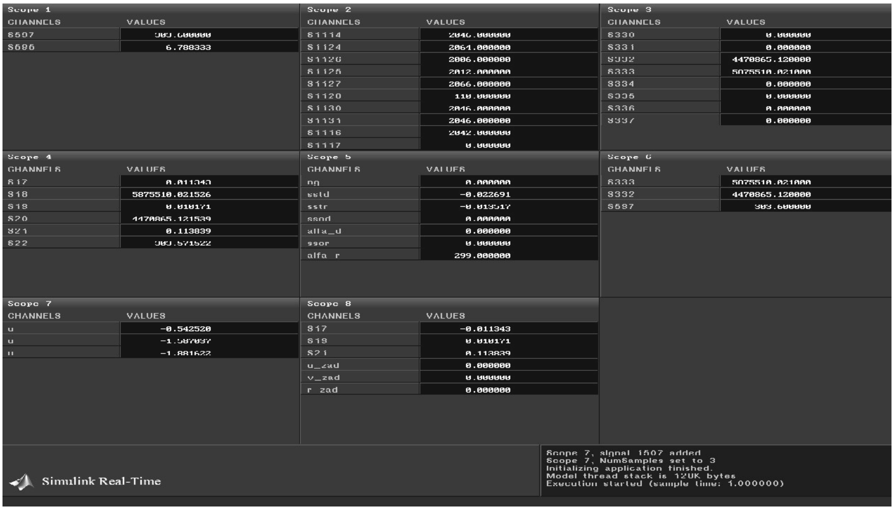
Step number	Detailed step description	Screenshot
7	In 'Simulink Real-Time Explorer' window set 'TargetPC2_BL' as the default.	 Simulink Real-Time Explorer Targets: MATLAB Session, TargetPC1_Czarny, TargetPC2_BL Remove / Set as Default Target / Rename / Connect / Load / Unload / File System Browser / Properties Target Properties – Host-to-Target communication Communication type: TCP/IP IP address: 192.168.112.13 Subnet mask: 255.255.255.0 Port: 22222 Gateway: 192.168.112.254 Bus type: ● PCI ○ ISA ○ USB Target driver: Auto

8	In Matlab Command Window type the: *xpctest* command, when the test is being conducted DO NOT switch off the target computer.	
		`### Test 5, Check host-target command communications with 'TargetPC1':`
		`### Test 6, Download a pre-built xPC Target application to target PC 'T`
		`### 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`

9	Open .mdl file, in the described case it is SimulatorBlue4.mdl from the Symulator BL folder and Simulink scheme containing synthesized controller block diagram.	 E: ▸ Matlab_regulator_MPC ▸ Szkielety Current Folder SimulatorBlue4_slrt_rtw SimulatorBlue4_xpc_rtw slprj csf_BL10.c csf_wiatr1.c SimulatorBlue4.dlm SimulatorBlue4bio.m SimulatorBlue4pt.m SimulatorBlue4ref.m SilmLib5.mdl SimulatorBlue4.mdl csf_BL10.mexw32 csf_wiatr1.mexw32 csf_BL10.mexw64 csf_wiatr1.mexw64 SimulatorBlue4.mdl (Simulink Model)
10	Compile both models by pushing 'Build model' application icon in Simulink package.	 SimulatorBlue4 File Edit View Display Diagram Simulation Analysis Code Tools Help inf External Build Model Build model
11	In 'Simulink Real-Time Explorer' window, connect Host with both Targets using 'Connect' command.	 Targets: MATLAB Session, TargetPC1_Czarny, TargetPC2_BL Remove / Set as Default Target / Rename / Connect / Load / Unload / File System Browser / Properties
12	In 'Simulink Real-Time Explorer' window load the the automatic control system scheme using 'Load' command. to 'TargetPC1_Czarny', and to 'TargetPC2_BL' - load the ship's simulator.	 Targets: MATLAB Session, TargetPC1_Czarny, TargetPC2_BL Remove / Set as Default Target / Rename / Disconnect / Load / Unload / File System Browser / Properties TargetPC1_Czarny/MPC_control TargetPC2_BL/SimulatorBlue4

Step number	Detailed step description	Screenshot
13	In 'Simulink Real-Time Explorer' 'Applications' window run both schemes. Results of the simulation may be observed on the scopes which are present on the 'TargetPC1_Czrany' and 'TargetPC2_BL' screens.	 Communication type: TCP/IP TargetPC1_Czarny/MPC_Control_Blue TargetPC2_BL/SimulatorBlue4

It is possible to run real-time simulations if compiler, all network connections, physical connections and devices of constituting the above mentioned system, are configured in a proper way. When simulation is running data is presented on the numerical scopes presented in Fig. 4. It gives an opportunity to maintain the way in which control algorithm is working during conducted simulations.



```
Scope 1                          Scope 2                          Scope 3
CHANNELS  VALUES                 CHANNELS  VALUES                 CHANNELS  VALUES
S597      303.600000             S1114     2046.000000            S330      0.000000
S686      6.788333               S1124     2061.000000            S331      0.000000
                                 S1126     2006.000000            S332      4470865.120000
                                 S1125     2012.000000            S333      5075510.021000
                                 S1127     2066.000000            S334      0.000000
                                 S1120     110.000000             S335      0.000000
                                 S1130     2046.000000            S336      0.000000
                                 S1131     2046.000000            S337      0.000000
                                 S1116     2042.000000
                                 S1117     0.000000

Scope 4                          Scope 5                          Scope 6
CHANNELS  VALUES                 CHANNELS  VALUES                 CHANNELS  VALUES
S17       0.011343               ng        0.000000               S333      5075510.021000
S18       5875510.021526         ssld      -0.022691              S332      4470865.120000
S19       0.010171               sstr      -0.013517              S597      303.600000
S20       4470865.121539         ssnd      0.000000
S21       0.113839               alfa_d    0.000000
S22       303.571522             ssor      0.000000
                                 alfa r    299.000000

Scope 7                          Scope 8
CHANNELS  VALUES                 CHANNELS  VALUES
u         -0.542520              S17       -0.011343
u         -1.587037              S19       0.010171
u         -1.881622              S21       0.113839
                                 u_zad     0.000000
                                 v_zad     0.000000
                                 r zad     0.000000

Simulink Real-Time               Scope 7, signal 1507 added
                                 Scope 7, NumSamples set to 3
                                 Initializing application finished.
                                 Model thread stack is 128K bytes
                                 Execution started (sample time: 1.000000)
```

Fig. 4. Target screenshot

3. CONTROLLER SYNTHESIS

Real-time simulation system allows for automatic control system behavior simulation, before it is applied to the real floating training ship model. In this paper, the control object is Blue Lady training ship's model described in a detailed way in [3] and [4]. During the simulations a multidimensional Model Predictive Controller (MPC) was used as an automatic control system for precise longitudinal, transversal and rotational speed adjustment. MPC controller structure is presented in Fig. 5.

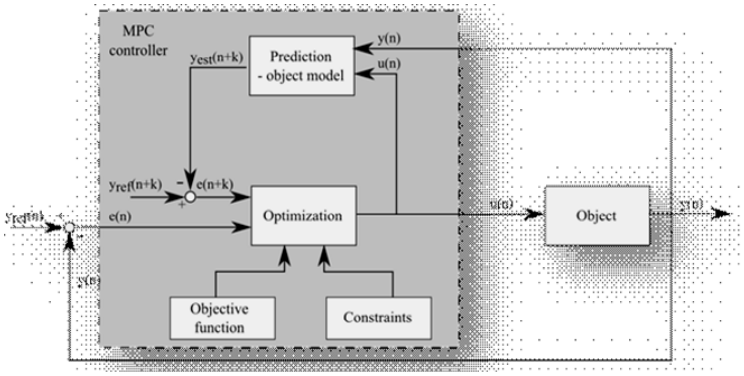


Fig. 5. MPC controller structure

The idea of Model Predictive Control is to determine future optimal control signal values, based on the object's mathematical model, known set points, past control signal values, and past output signal values. In case of the synthesized controller, based on training VLCC ship linear model, reference signal is $y_{ref} = [u_{ref}, v_{ref}, r_{ref}]$, where u_{ref}, v_{ref} and r_{ref} points of longitudinal, transversal and rotational speeds are set. MPC controller working on the basis of moving horizon idea was designed. Predicted values form a vector of control signals:

$$u(k) = u(k \mid k), u(k+1 \mid k), \ldots, u(k + N_u - 1 \mid k),$$

where N_u – is a control horizon.

Outside this range control signals satisfy the relationship: $u(k + p \mid k) = u(k + N_u - 1 \mid k)$ for $p \geq N_u$. For the control purpose only the first estimated control signal value (first element of the control signals vector) is used, which has been described in [7] and [8].

In MPC controller prediction is done on the basis of linear mathematical model presented by Gierusz in [3]. Control signals are values of thrust in ship's longitudinal plane (τ_x), thrust in ship's transversal plane (τ_y) and torque (τ_p). They are indicated as ***mv*** vector in Fig. 6. Input signals, constituting ***mo*** vector (Fig. 6) are respectively longitudinal (u), transversal (v) and rotational (r) speeds.

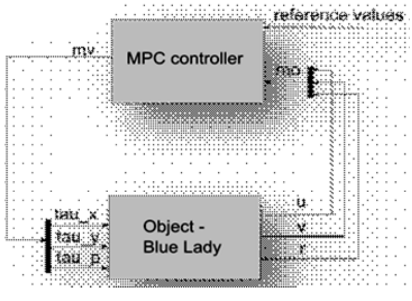


Fig. 6. MPC automatic control system block diagram

The use of linear model allows for significant simplification of controller structure and for decreasing computations duration. Moreover, use of a simple form of the objective function describe by the equations (1) is possible.

$$J = \sum_{p=N_1}^{N} \left\| Mx(k+p|k) - y(k+p) \right\|_{Q(p)}^{2} + \sum_{p=0}^{N_{u-1}} \left\| \Delta u(k+p) \right\|_{R(p)}^{2} \tag{1}$$

where: M – output matrix C (when objects model is described in state space), $y(k+p)$ – output signal in the $k+p$ moment of time, $\Delta u(k+p)$ – control signal change in the $k+p$ moment of time.

Norm $\|.\|_Q^2$ is defined on the basis of the cost function α as follows: $\|\alpha\|_Q^2 = \alpha^T Q\alpha$. Similarly Norm $\|.\|_R^2$ is defined on the basis of the cost function α as follows: $\|\alpha\|_R^2 = \alpha^T R\alpha$. Matrix Q – is a matrix of weights for the output signal errors and R – is a matrix of weights for the control signals increments.

Synthesis of MPC controller requires formation of an object's mathematical model, definition of cost function form and lengths of control and prediction horizons. In the next step control and output signal constraints are introduced. In the controller tuning process there is a possibility of weights coefficients introduction. They are penalty coefficients for error signal and control signal change. In case of automatic MPC control system for Blue Lady ship the values of longitudinal, transversal and rotational speeds shown in Table 2 were chosen. Lengthening of the prediction and control horizons elongates computation time, because it complicates optimization procedure. It also improves controller robustness. So there is a need to arbitrary and iteratively choose their lengths. The controller should be as robust to disturbances and as fast as possible.

Table 2. Model Predictive Controller parameter's values

Parameter	Value
Length of prediction horizon	40
Length of control horizon	2
Matrix of weights for the output signal errors Q	$\begin{bmatrix} 60 & 0 & 0 \\ 0 & 35 & 0 \\ 0 & 0 & 35 \end{bmatrix}$
Matrix of weights for the control signal increments R	$\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0.6 \end{bmatrix}$
Control signal τ_x constraints	$-20 < \tau_x < 50$
Control signal τ_y constraints	$-20 < \tau_x < 20$
Control signal τ_p constraints	$-20 < \tau_x < 20$

Model Predictive Control used for a real controlled object works using the configuration presented in Fig. 6. The same configuration is used for the controller used for real time simulations. For simulation purposes the correct schematic was built in Matlab Simulink and includes the synthesized predictive controller.

4. SIMULATION RESULTS

Real-time simulations were carried out and are presented in the left half of Fig. 7 called "simulations with xPC Target". The right half of Fig.7 presents simulations carried out in Matlab in non real-time environment and they are called "simulations without xPC Target".

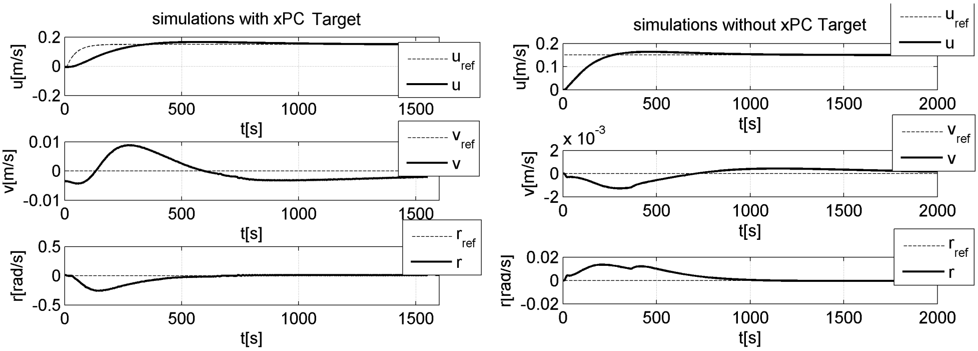


Fig. 7. Simulation results for longitudinal velocity u = 0.15 [m/s] and both transversal and rotational velocities equal zero v = 0 [m/s], r = 0 [rad/s]

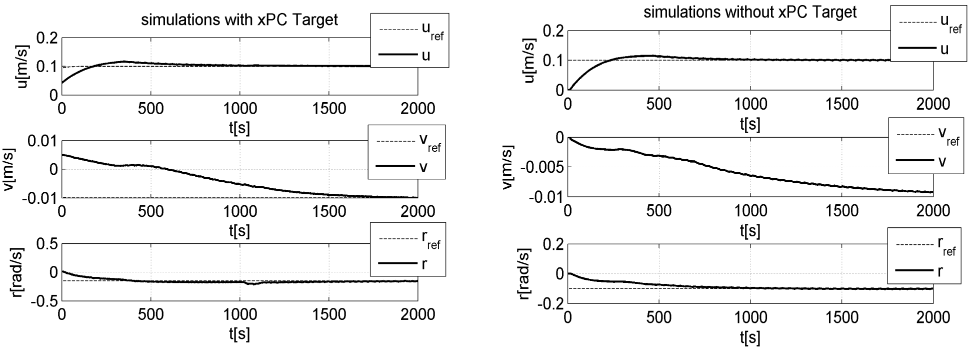


Fig. 8. Simulation results for longitudinal velocity u = 0.1 [m/s], transversal velocity v = -0.01 [m/s] and rotational velocity r = -0.1 [rad/s]

Simulation results from Fig. 7 show the system with longitudinal velocity u = 0.15 [m/s] and both transversal and rotational velocities equal to zero. The difference between real-time simulations (on the left) and non real-time simulations (on the right) are clearly visible. The long settling time of approx. 2000 [s] is caused by analogue to digital signal converters that calculate forces of ship's thrusters. In Fig. 8 results for three velocities, longitudinal u = 0.1[m/s], transversal v = -0.01 [m/s] and rotational r = -0.1 [rad/s] are presented.

When simulating designed control system in real-time, its operation is similar to controller performance in real lake conditions. Results obtained in real-time are worse than the ones from non real-time as they include dynamics of the ship model in real time. Designed controller is a discrete one. So when working on the lab station discretisation, signal transmition and algorithm computation times are taken into account. This lab station gives a possibility to analyze influence of the prediction and control horizons on the whole working control system. Too long horizon lengths selection caused system's unstability, which could not be seen during the simulations conducted directly in Matlab software.

5. CONCLUSIONS

A real-time simulator is essential for verifying calculations from computer simulations which are to be implemented on "Blue Lady" ship model. The test bed located in the laboratory of Maritime University in Gdynia is a half way point between computer simulations and real life experiments conducted on the ship model located on Silm lake. At this point calculations of computer simulations that do not show inaccurate results or do not include real conditions on Silm lake, can be changed. Upgrade from Matlab2011b to Matlab2014a proved to be beneficial because due to the new, more intuitive, interface of xPC Target it was easier and quicker to set communication between all the elements of the lab station. Verification done on the lab station allows to save many working hours of work that would have to be spent in real life conditions on the Silm lake on the "Blue Lady" ship model. Furthermore this method is extremely close to real ship control.

REFERENCES

[1] Rybczak M., *Lab station for verifying ship control algorithm with the use of real time platform xPC target.* Gdynia, Vol. 78, 2013, p. 97–109 (in Polish).

[2] Centrum Manewrowania Fundacji Bezpieczeństwa Żeglugi i Ochrony Środowiska w Iławie Kamionce http://www.ilawashiphandling.com.pl/training.html.

[3] Gierusz W., *Synthesis of multivariable systems of precise ship motion control with the aid of selected resistant system design methods,* Akademia Morska w Gdyni, Gdynia 2005 (in Polish).

[4] Rybczak M., *Wykorzystanie języka VRML do modelowania środowiska symulacyjnego ruchu modeli statków,* Gdynia 2006 (in Polish).

[5] Ułasiewicz J., *Systemy czasu rzeczywistego - QNX6 Neutrino,* Wyd. BTC, 2007.

[6] Pomirski J., Rak A., Gierusz W., *Control system for trials on material ship model,* Polish Maritime Research, Special Issue SI(74), 2012, Vol. 19, p. 25–30.

[7] Camacho E.F., Bordons C., *Model Predictive Control,* Springer, Great Britain, 1999.

[8] Oh S.R., Sun J., *Path following of underactuated marine surface vessels using line-of-sight based model predictive control,* Ocean Engineering 2010, 37, p. 289–295.