

Towards A Virtual Prototyping Framework for Ship Maneuvering in Offshore Operations

Guoyuan Li, Pedersen Birger Skogeng, Yuxiang Deng, Lars Ivar Hatledal and Houxiang Zhang *Faculty of Maritime Technology and Operations
Norwegian University of Science and Technology
Postboks 1517, N-6025, Aalesund, Norway
{guoyuan.li, birgersp, yude, laht, hozh}@ntnu.no*

Abstract

This paper presents a virtual prototyping framework that can achieve modeling, simulation and control for ship maneuvering in offshore operations. To fast construct scenario for certain ship maneuvering tasks, a scenario editor is designed to model the ship, the environment disturbances and the scene, respectively. The corresponding scenario containing physical properties is built up in a physics simulation engine AgX. A graphical user interface responsible for visualizing the simulation and plotting data in a real-time manner is developed, in which dynamic interaction including manual control, failure/recover of thrusters and weather change is implemented. Taking advantages of a distributed communication architecture, the plug-in control mechanism is realized to increase the flexibility for evaluating ship maneuvering performance. Through a case study of trajectory tracking, the framework is verified efficient in constructing scenario and testing control model for ship maneuvering applications.

Index Terms

Virtual prototyping, Ship maneuvering, Modeling and simulation, Plug-in control mechanism.

I. INTRODUCTION

There is an increasing trend of emerging demands in maritime applications, such as wind turbine installation [1] and heavy lifting operations [2], in which the ship is required not only for fine maneuvering in limited working spaces, but also for highly considering both safety and accuracy during offshore operations. However, the time frame with a new design of control model or an alternative procedure with respect to ship maneuvering transferred to industry may easily be up to 2-3 years. Introducing a virtual prototyping framework, by creating a fast, easy to use simulator simultaneously with flexible plug-in control models, will effectively shorten the time from design to practical use and allow more alternatives to be tested.

Nowadays, advanced computer technology has paved the way for virtual prototyping (VP). It is about presentation, testing and analysis of three-dimensional models prior to creating any physical prototypes, i.e., to create computer generated geometrical shapes and combine them into an “assembly” and test different mechanical motions, fit and function [3]. VP allows engineers to build and test virtual prototypes and realistic simulation visually and mathematically on computers, resulting in faster time-to-market through earlier and faster software development and improving communication throughout the supply chain.

Owing to the fast, efficient and full functional software models for complex systems, VP has been studied and widely applied in many fields. Schreck from Fraunhofer Institute has proposed a concept for the provision of a virtual training center by cost-effective simulation services via Internet [4]. A virtual

development and training platform was developed for interactive visualization and qualification. Li et al. proposed the construction VP for the construction industry to develop effective dynamic construction project planning and scheduling tools [5]. They succeeded to integrate building information models with VP for visualization, collision detection, assembly operation evaluation and simulation. Krupke et al. introduced a graphical user interface to create and control simulations for modular robots in an easy and flexible way [6]. The software intends to hide programming part for beginners whilst offering advanced optimization methods for researchers in modular robots. Choi and Cheung presented a versatile VP system for digital fabrication of multi-material prototypes to facilitate rapid product development [7]. The system provides visualization and analysis techniques to improve the design of a footwear product and its fabrication processes.

VP technologies are also applied for maritime applications. Metrikin et al. developed a numerical simulator of dynamic positioning (DP) in ice [8]. The simulator is able to model the ice basin, the DP vessel, the ice, the surrounding fluid and their mutual interactions. Ueng et al. presented efficient computational models for ship motions [9]. The system can assist users to learn the motions of a ship and tune ship behaviors by varying some predefined parameters. The Kongsberg Maritime AS has developed several kinds of simulators for maritime training, such as crane simulator, offshore vessel simulator and liquid cargo handling simulator [10]. The Offshore Simulator Centre (OSC) AS has similar commercial products for the simulation of crane, subsea and bridge operations and delivers offshore training concepts aimed at increasing safety for personnel involved in demanding offshore operations [11]. Currently, we collaborate with OSC AS, aiming to develop a VP framework for modeling, control and simulation of ships for fine maneuvering in offshore operations. The resultant framework could be either a software for education of ship maneuvering theory or a platform for research on advanced control algorithms.

The rest of the paper is organized as follows. Section II briefly introduces the overall structure of the VP framework for ship maneuvering. In Section III, all relevant components including modeling, simulation and control are described in detail. Section IV presents a case study of trajectory tracking and relevant evaluation results. Conclusion and future work are shown in section V.

II. FLEXIBLE FRAMEWORK STRUCTURE

In this paper, we propose a framework structure that can flexibly achieve modeling, simulation and control for ship maneuvering, as shown in Fig. 1. Modeling is the first step for virtual prototyping. In connection with ship maneuvering, a rapid testing scenario model can be divided into three sub-models: the ship itself, environment disturbances and the scene. All components related to ship control, such as the hull, the engine, the propellers, the rudders and the sensors, are abstracted and modeled into libraries. A flexible configurable Graphic User Interface (GUI) will guide the user to assemble the virtual ship according to actual demands. It is also essential to model offshore scenes and the weather — especially the environmental perturbations like current, waves and wind. Users can pick up different scenes and weathers together with the ship and set up corresponding attributes to complete the maneuvering scenario, as a consequence generating a JSON configuration file [12].

In a physics simulator called AgX, the JSON file is resolved to construct the scenario. AgX can run in real time using “real physics” in commercial simulators [13]. In particular, it is able to calculate hydrodynamics and simulate the interaction between water and the ship. Once the scenario is established in AgX, the framework will interact with users by providing web based visualization and related data for real-time plotting.

The highlight of the framework is the plug-in control models for the performance evaluation of ship maneuvering. Either existing control models such as station keeping, mooring and dynamic positioning, or user customized control models can be imported into the simulator. Users can even select manual

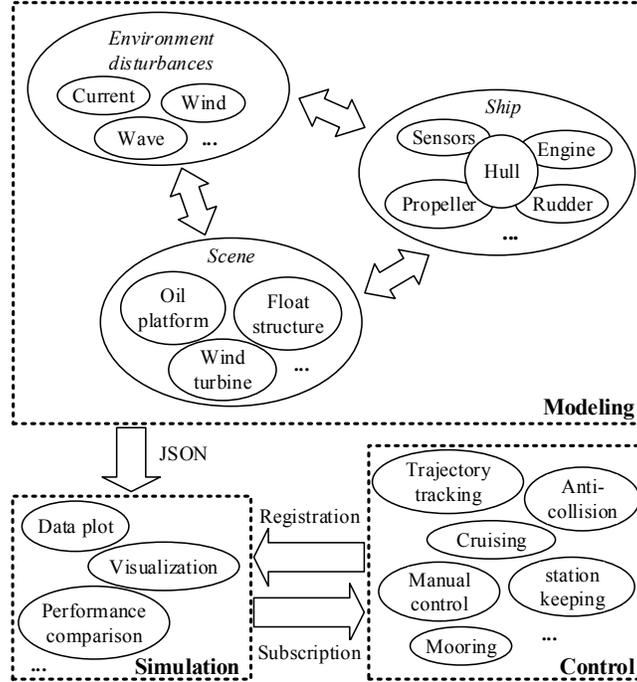


Fig. 1. A testing framework for virtual prototyping of ship maneuvering.

control of the ship, so as to make performance comparison after the simulation. The interaction between the control part and the simulation part is realized by means of registration and subscription. On the one hand, the simulator can access control parameters and resultant control force and torque after a control model is registered. On the other hand, the control model is allowed to retrieve ship status information as long as the information is subscribed through the control model.

The proposed flexible framework has several promising characteristics. First, from ship design point of view, it provides an efficient way to evaluate how propeller configuration affects ship-maneuvering performance. Second, ship-maneuvering training becomes possible due to the convenience of performance comparison between manual and autonomous control. Third, the framework can be used as a research platform for ship maneuvering, based on which studies and improvements of control algorithms can be done with reduced time effort.

III. IMPLEMENTATION OF SHIP MANEUVERING FRAMEWORK

A. Modeling

As mentioned in Section II, a complete scenario for ship maneuvering includes the ship, the environment and the scene. Ship modeling is of great concern among it, since the way to construct the ship will effect ship maneuvering performance eventually. More specifically, the hull property like center of mass, the thrusters properties such as relative position and orientation, as well as their physical constraints like the limitation of the shaft speed and the rudder angle, dominantly determine the ship's maneuverability. The thrusters are modeled in three types, i.e., the tunnel thruster, the azimuth thruster and the main propeller with rudders. Each of them has different characteristics and constraints. Besides that, the framework also offers all the ship status to users, including the velocity, the heading and the position. For environment perturbation modeling, as they are not the key part for VP, only some basic

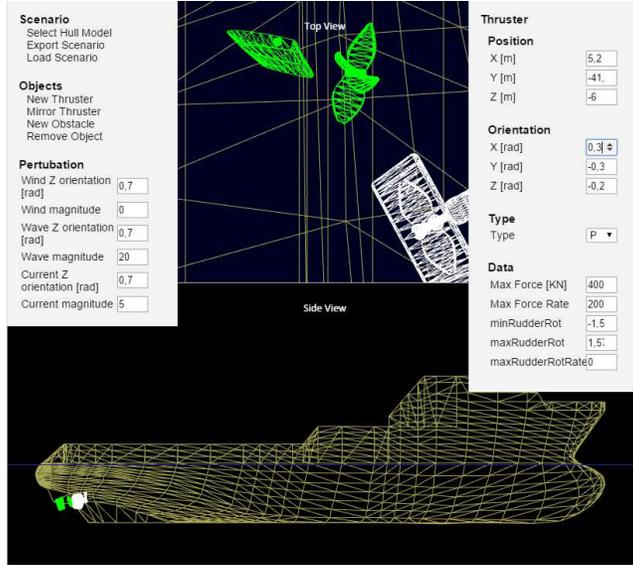


Fig. 2. Screenshot of ship maneuvering scenario editor.

information like orientation and force magnitude is required. Likewise, the scene can be set up with position, orientation and scaling ratio.

To configure a testing scenario, a simplistic “scenario editor” has been developed. To enable a wide range of compatibility, this editor is developed as a JavaScript-web application, with any modern web-browser as it’s only prerequisite. The graphics are developed in WebGL using ThreeJS 3D library [14]. The GUI of the editor consists of horizontal split view of the scenario, as shown in Fig. 2. The upper half of the view shows the scenario from a “top-down” perspective, while the lower half shows a side-to-side view. By using the computer mouse, the views can be shifted sideways or zoomed, giving the user a flexible overview of the the layout and elements in a scenario while configuring. From the left-hand menu, the user may change the ship model, add/remove thrusters or scenario “obstacles”. In addition, this menu contains the various environmental parameters for initialization. Once created, thrusters and obstacles may be selected in either of the views, enabling another menu to set various parameters for the selected object. Once configured, the scenario configurations can be exported to a JSON file.

B. Simulation

The testing scenario stored as a JSON file is resolved through JsonCpp [15]. In AgX, the corresponding scenario containing physical models is built up. Both the ship and the scene are constructed as rigid bodies. The wind and the wave are implemented as a force vector and a height field, whereas the current is realized as an extra speed exerting on the ship in the simulation. The gravity is set to $-9.81m/s^2$ to be the same as the real world. For the water with hydrodynamic effect, the pressure drag, the lift and the viscous drag coefficients are set as 1.0, 0.1 and 1.0, respectively.

A GUI for ship maneuvering simulation is developed, as shown in Fig. 3. It plays the role in 3D visualization of the scenario and real-time plotting function. Any data related to ship status can be drawn during the simulation. Both the visualization and the plot are synchronized with AgX. Data transmission between the GUI and AgX is realized in a plug-in manner, which will be introduced in the following section.

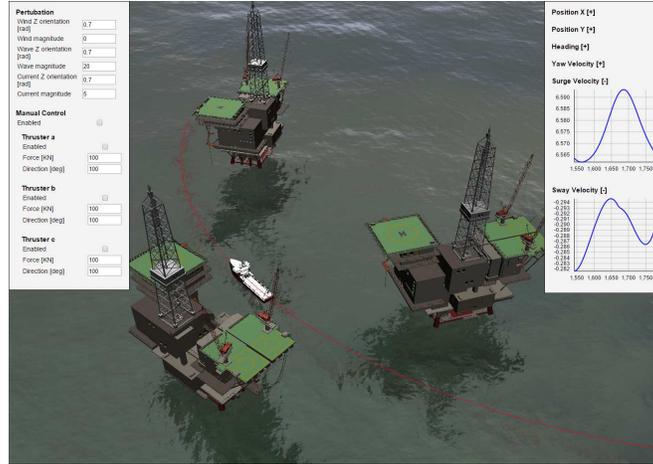


Fig. 3. Screenshot of the GUI for ship maneuvering simulation.

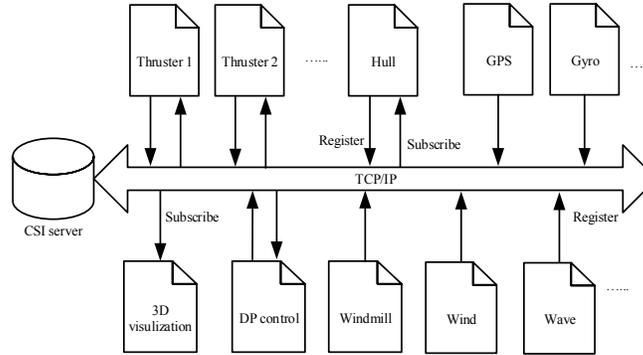


Fig. 4. Distributed communication architecture.

In addition, the GUI has the ability to interact with AgX dynamically. The feature is reflected from three aspects:

- Manual control capability: even if there is the imported control model, users can take over the control of the ship due to higher priority of manual control.
- Dynamic simulate failure/recover of thrusters: to emulate unexpected events on the ship, or to evaluate the robustness of certain control algorithms.
- Dynamic change of weather: to simulate a sudden environmental perturbation by altering its direction or magnitude.

C. Plug-in Control

To make the control algorithm be uncoupled with the simulator, a distributed control scheme is needed. Here we utilize a multi-threaded server based common simulation interface (CSI) to realize the plug-in control mechanism[16]. Fig. 4 illustrates the CSI communication architecture. The CSI server will play a role in data gathering and distribution. The client can be either any functionalities from the ship, the environment, the on-board sensors, the GUI for visualization and plotting, or any control algorithms. Through the TCP/IP protocol, a client can register to the CSI server to share its data, and subscribe interested data that other clients have registered.

```

<?xml version="1.0" encoding="UTF-8"?>
...
<register name="Trajectory tracking" type="Control">
  <parameter name="Track_type" type="int" value="1"/>
  <parameter name="Track_speed" type="double" value="1.0"/>
  ...
</register>
...
<subscribe name="Hull" type="Vessel">
  <attribute name="cg_north"/>
  <attribute name="cg_east"/>
  ...
</subscribe>
...
<subscribe name="Windmill" type="Obstacle">
  <attribute name="position"/>
  <attribute name="orientation"/>
</subscribe>
...

```

Fig. 5. An example of XML file for trajectory tracking control.

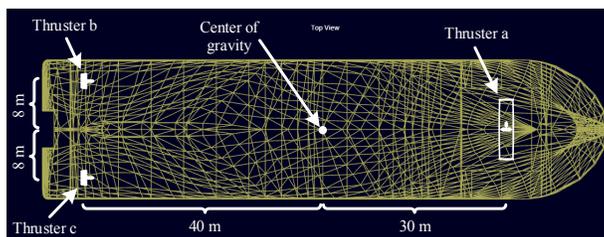


Fig. 6. Ship configuration for the trajectory tracking task.

For each client, an XML configuration file that contains the register/subscribe information must send to the CSI server to keep track of subscriptions and redirections of registered data. Fig. 5 shows an example of the XML file used for trajectory tracking, in which the control algorithm registers control parameters and subscribes related information of the ship and the windmill. The client application is platform irrelevant since the communication between the CSI server and the client are based on the XML streams via a socket connection. This means the control algorithm can be implemented in any programming languages, such as C++, java and Matlab. Through a CSI library, accessing the ship status from AgX and sending control commands to AgX can be easily achieved. As a result, users can write their own control methods using their preferable programming language freely.

IV. EXPERIMENT

To validate the effectiveness of the VP framework for ship maneuvering, a case study was carried out. A hull model with a mass of $m = 10^7 kg$ was selected. It has a length of $L = 94m$ and a width of $W = 23m$. The water line is $wl = 9m$ and the center of gravity related to the baseline, the centerline and the aft perpendicular is $G = [7m, 0m, 47m]$. The inertia matrix is set as $I = [6.5 \times 10^8, 0, 0; 0, 0, 4.3 \times 10^9; 0, 0, 0, 4.5 \times 10^9]$

We built up the ship with one tunnel thruster (thruster a) and two main propellers with rudders (thruster b and c), as shown in Fig. 6. Currently, as we do not implement the mapping from the thruster force to the shaft speed and the pitch angle, the control command onto the thruster is represented as a force magnitude. For simplicity, the three thrusters were set within the same force range $[-10^7 N, 10^7 N]$. Besides the ship,

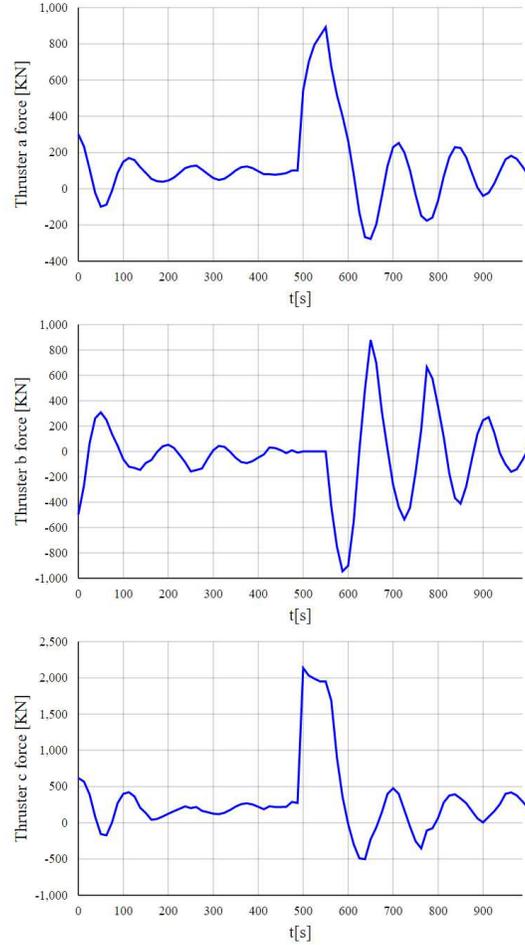


Fig. 7. Thruster force variation during the simulation.

an oil platform was constructed in the scenario. Initially, the oil platform was built at the origin. The ship was placed at $[0m, 160m]$ with the heading towards East. A constant wind perturbation from North to South with a magnitude of $10000N$ was applied on the ship.

The control task for the case study is to maneuver the ship in a very low speed around the oil platform by following an elliptic reference trajectory:

$$\begin{aligned}
 x_d &= 200 \cdot \sin(0.1t), \\
 y_d &= 150 \cdot \cos(0.1t), \\
 \psi_d &= \tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right).
 \end{aligned}$$

An XML file similar to Fig. 5 for the plug-in control algorithm is established, in which the ship's position and velocity were subscribed and the control of the three thrusters were registered. To achieve the goal, we followed the control method in [17] and implemented it in C++. To simplify the control, we fixed the rudder angle of the main propellers (thrusters b and c) to zero so that the force distributor has a unique solution for each thruster.

When the trajectory tracking algorithm started working, we tested dynamic interaction by deactivating thruster b in a short period and resuming it afterwards. The result is shown in Fig. 7, Fig. 8 and Fig. 9.

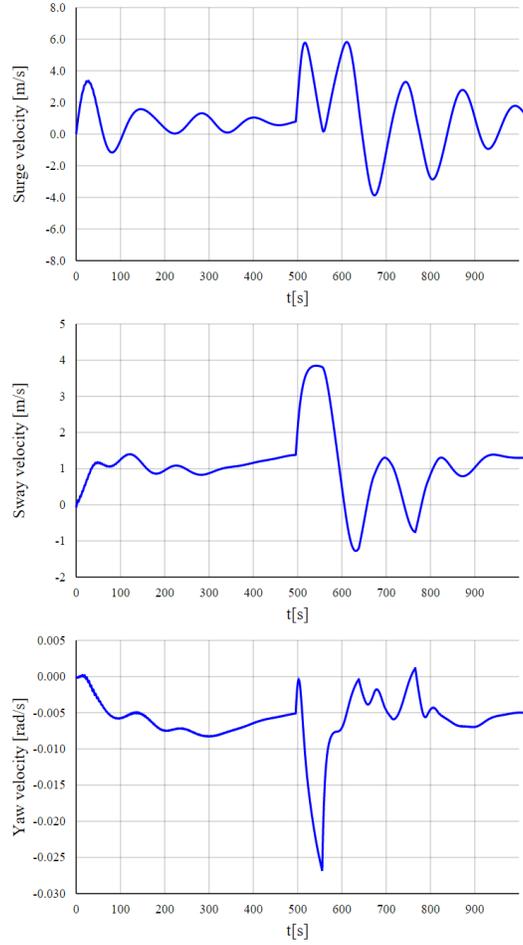


Fig. 8. Speed variation during the simulation.

It is noted that the ship followed the elliptic trajectory smoothly and accurately before the failure of thruster b. When the failure occurred, thruster a and c tried to increase their forces to compensate the force provided by thruster b, resulting in a velocity perturbation. However, due to underactuation, the tracking error increases. This phenomenon lasted even if thruster b recovered from the failure, since the control algorithm uses high gain on position error. The consequence is that the ship was driven back and forth to track back to the reference trajectory. With recovery of thruster b, the tracking performance gradually returned to normal state.

The case study shows the whole procedure to use the framework from building the ship to finally realizing the trajectory tracking task. From the result, we conclude the framework works well in modeling, simulation and control of ship maneuvering applications.

V. CONCLUSION

Motivated by the fact that the control model for new maritime applications from design to practice takes a relatively long period, it is necessary to simulate the application and test the control model to reduce time effort. In this paper, we emphasize the design of a virtual prototyping framework for ship maneuvering applications. The framework is comprised of three components, i.e., modeling, simulation and control.

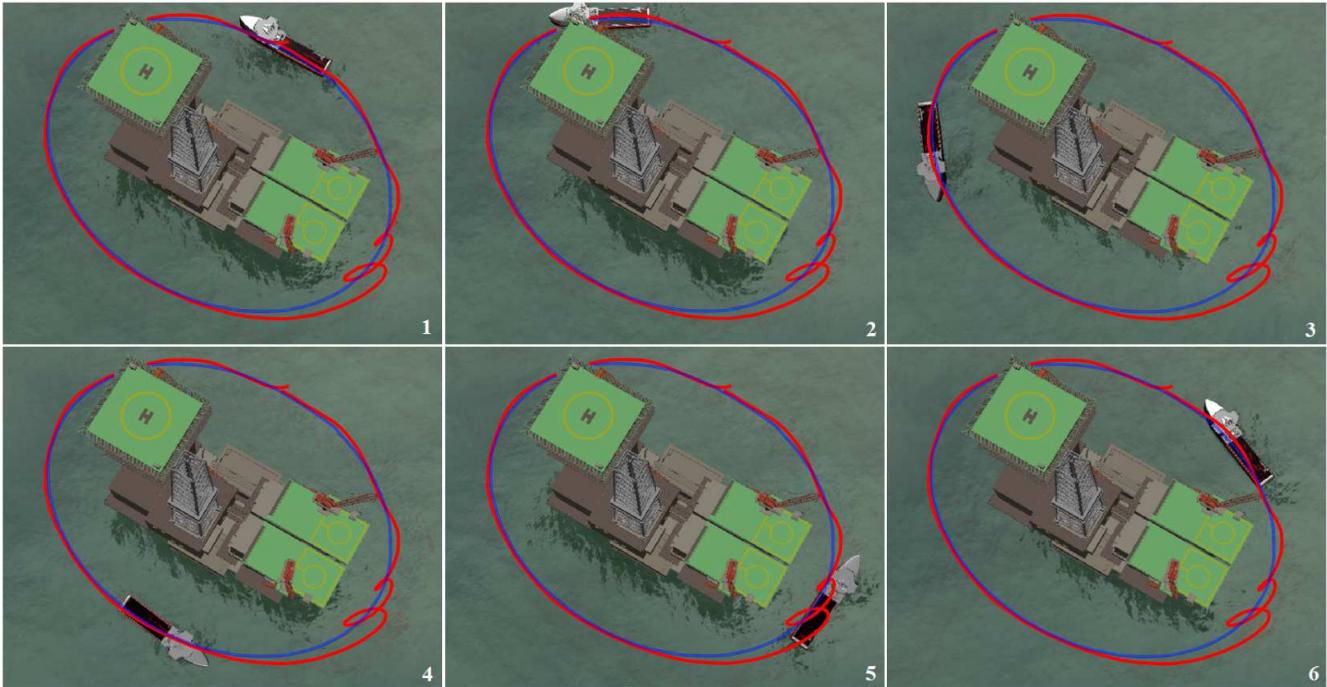


Fig. 9. Snapshots of the animation for trajectory tracking. The blue and the red lines represent the reference and the actual trajectories, respectively. Thruster b was deactivated when the ship positioned at in the fourth snapshot, and activated again after about 70s.

First, a scenario editor for modeling is developed, in which the ship, the scene and the environmental perturbations are all abstracted and modeled into libraries. Second, a GUI for ship maneuvering simulation is implemented to communicate to AgX for visualization, plotting and dynamic interaction. Third, a flexible plug-in control mechanism is realized based on the CSI architecture, enabling a feasible way for control model testing. A case study of trajectory tracking shows the proposed framework is efficient in realizing fast scenario construction and testing the control model for ship maneuvering in offshore operations.

For future work, we will concentrate on the refinement of the thrusters, especially modeling the mapping from the shaft speed, the pitch angle and the rudder angle to propulsion. Furthermore, the communication architecture will be improved by using Functional Mock-up Interface (FMI) standard.

ACKNOWLEDGEMENT

The research is partially supported by the project “An Approach toward Optimal Control of Ship manoeuvring in Offshore Operations” funded by RFF Midt-Norge, Norway (e-søknad ref nr: ES560471).

REFERENCES

- [1] D. Roddier, C. Cermelli and A. Weinstein, “WindFloat: a floating foundation for offshore wind turbines—part I: design basis and qualification process”, in *proc. of ASME Int. Conf. on Ocean, Offshore and Arctic Engineering (OMAE2009)*, Hawaii, USA, pp. 845-853, 2009.
- [2] K. P. Park, J. H. Cha and K. Y. Lee, “Dynamic factor analysis considering elastic boom effects in heavy lifting operations”, *Ocean Engineering*, vol. 38, no.10, pp. 1100-1113, 2011.
- [3] F. Zorriassatine, C. Wykes, R. Parkin and N. Gindy, “A survey of virtual prototyping techniques for mechanical product development”, *Proceedings of the institution of mechanical engineers, Part B: Journal of engineering manufacture*, vol.217, no.4, pp. 513-530, 2003.

- [4] G. Schreck, "Simulation services for training of plant operators", in *Knowledge and Technology Integration in Production and Services*, Springer US, pp. 79-86, 2002.
- [5] H. Li, T. Huang, C. W. Kong., H. L. Guo, A. Baldwin, N. Chan and J. Wong, "Integrating design and construction through virtual prototyping", *Automation in Construction*, vol. 17, no.8, pp. 915-922, 2008.
- [6] D. Krupke, G. Li, J. Zhang, H. Zhang and H. P. Hildre, "Flexible Modular Robotic Simulation Environment For Research And Education", in *proc. of 26th European Conference on Modeling and Simulation (ECMS)*, Koblenz, Germany, pp. 243-249, 2012.
- [7] S. H. Choi and H. H. Cheung, "A versatile virtual prototyping system for rapid product development", *Computers in Industry*, vol. 59, no. 5, pp. 477-488, 2008.
- [8] I. Metrikin, S. Løset, N. A. Jenssen and S. Kerkeni, "Numerical simulation of dynamic positioning in ice", *Marine Technology Society Journal*, vol. 47, no. 2, 14-30, 2013.
- [9] S. K. Ueng, D. Lin and C. H. Liu, "A ship motion simulation system", *Virtual reality*, vol. 12, no. 1, pp. 65-76, 2008.
- [10] Kongsberg Maritime AS, "Simulators" [Online]. Available: <http://www.km.kongsberg.com>. [Accessed: Feb. 20, 2016].
- [11] Offshore Simulator Centre AS, "Solutions" [Online]. Available: <http://www.osc.no>. [Accessed: Feb. 20, 2016].
- [12] Introducing JSON, "JSON" [Online]. Available: <http://www.json.org/>. [Accessed: Feb. 20, 2016].
- [13] A. Backman, "Algoryx—Interactive Physics", in *Proc. of the annual SIGRAD conference special theme: interaction*, Stockholm, Sweden, pp. 87-87, 2008.
- [14] Three.js [Online]. Available: <http://threejs.org/>. [Accessed: Feb. 22, 2016].
- [15] JsonCPP [Online]. Available: <https://github.com/open-source-parsers/jsoncpp/wiki>. [Accessed: Feb. 22, 2016].
- [16] W. Rekdalsbakken and Arne Styve, "Simulation of intelligent ship autopilots", in *proc. of 22th European Conference on Modeling and Simulation (ECMS)*, Nicosia, Cyprus, 2008.
- [17] G. Li, W. Li, K. H. Halse, H. P. Hildre and H. Zhang, "Hierarchical control of marine vehicles for autonomous maneuvering in offshore operations", *Ship Technology Research*, vol. 62, no. 2, pp. 72-80, 2015.