A Wave Simulator and Active Heave Compensation Framework for Demanding Offshore Crane Operations

F. Sanfilippo, L. I. Hatledal and H. Zhang
Department of Maritime Technology and Operations, Aalesund University College
Postboks 1517, 6025 Aalesund, Norway
{fisa, laht, hozh}@hials.no

W. Rekdalsbakken
Department of Engineering and Natural Sciences
Aalesund University College
Postboks 1517, 6025 Aalesund, Norway
wr@hials.no

K. Y. Pettersen
Centre for Autonomous Marine Operations and Systems
Department of Engineering Cybernetics, Norwegian University of Science and Technology
7491 Trondheim, Norway
kristin.y.pettersen@itk.ntnu.no

***Abstract*— In this work, a framework is presented that makes it possible to reproduce the challenging operational scenario of controlling offshore cranes via a laboratory setup. This framework can be used for testing different control methods and for training purposes. The system consists of an industrial robot, the *Kuka KR 6 R900 SIXX (KR AGILUS)* manipulator and a motion platform with three degrees of freedom. This work focuses on the system integration. The motion platform is used to simulate the wave effects, while the robotic arm is controlled by the user with a joystick. The wave contribution is monitored by means of an accelerometer mounted on the platform and it is used as a negative input to the manipulator's control algorithm so that active heave compensation methods can be achieved. Concerning the system architecture, the presented framework is built on open-source software and hardware. The control software is realised by applying strict multi-threading criteria to meet demanding real-time requirements.**

Related simulations and experimental results are carried out to validate the efficiency of the proposed framework. In particular, it can be certified that this approach allows for an effective risk reduction from both an individual as well as an overall evaluation of the potential harm.

***Index Terms*— Active Heave Compensation, Control Framework, System Integration.**

I. Introduction

During the last few decades, the maritime industry has shown a growing interest in developing new technologies for controlling modern vessels and related maritime equipment to perform increasingly demanding maritime operations. The operations of an offshore installation are associated with a high level of uncertainty because such an installation usually operates in a dynamic environment in which technical, human and organisational malfunctions may cause accidents. In the context of such a challenging operating environment, offshore cranes play a major role, given their use in transporting and lifting operations.

Under rough sea conditions, offshore activities involving crane operations lead to many problems such as load sway, positioning accuracy, collision avoidance and manipulation security. Even though the operating environment can be very challenging, it is still quite common to use relatively simple control interfaces to perform offshore crane operations. In most cases, the operator has to handle an array of levers and buttons to operate the crane joint by joint. When considering work efficiency and safety, this kind of control is extremely difficult to manage and relies on extensive experience and a high operating skill level of the operator. In particular, when a large wave impact occurs under extreme sea conditions, reliable control is almost impossible to be manually achieved. Currently, a huge amount of resources are spent on training operators and a great deal of cost can be wasted during the downtime waiting for a better weather condition. In this regard, more flexible and reliable control approaches are needed. Several research groups are investing resources in this direction. However, testing new control methods in a real setup environment is very difficult because of the challenging work-space in which maritime cranes are operated.

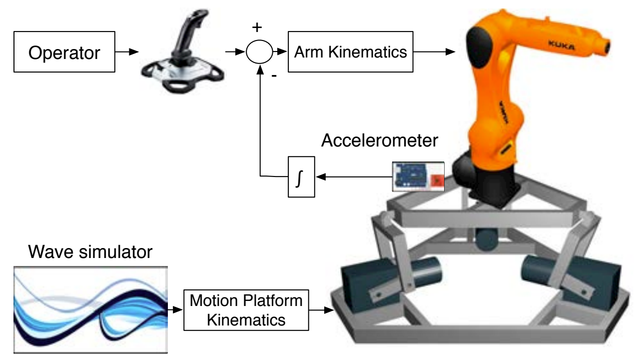


Fig. 1: The proposed wave simulator and active heave compensation framework for demanding offshore crane operations.

To give researchers the possibility of testing alternative control algorithms for maritime cranes in a realistic and safe laboratory setup, a waves simulator and active heave compensation framework for demanding offshore crane operations is proposed in this paper. The underlying idea is shown in Fig. 1. The system is composed of an industrial robot, the *Kuka KR 6 R900 SIXX (KR AGILUS)* manipulator, and of a motion platform with three degrees of freedom (DOFs). The motion platform allows the simulation of wave impacts, while the robotic arm can be manoeuvred by the user with a standard joystick or any other input device. An accelerometer is embedded on the platform to monitor the wave contribution. This same contribution is given as a negative input to the manipulator's control algorithm so that active heave compensation methods can be realised. It should be noted that only the heave compensation problem is addressed in this preliminary work, while all the issues related to rope pendulations are not considered (the robot on the platform is not equipped with any rope). A transparent user control interface can be implemented by using the proposed framework. In addition, the system can also be used for training purposes.

Regarding the system architecture, the presented framework is built on open-source software and hardware. Strict multi-threading criteria are applied to the control software to meet strict real-time requirements. The authors intend this work to be the first in a series of open-source designs to be released, and through the contributions of the open-source user community, result in a large number of design modifications and variations available to researchers. The official repository is available on-line at `https://github.com/aauc-mechlab/WaveSimulator`, along with several detailed class diagrams, all the mechanics, hardware schematics and demo videos.

The paper is organised as follows. A review of the related research work is given in Section II. In Section III, we focus on the description of both the proposed motion platform and the considered manipulator, analysing the system architecture and the communication protocol. Successively a possible control approach for the integrated framework is presented. Related simulations and results are shown in Section IV. In Section V, conclusions and future works are outlined.

II. Related Research Work

Unlike cranes mounted on fixed bases, offshore crane operations are significantly influenced by the ship motions resulting from currents and waves. The dynamic forces generated from the heave motion of the vessel and the sway movements of the load pendulation have significant effects on the crane operations. Operating in such a challenging scenario is very demanding. Advanced control methods are needed to compensate for the wave impact and to guarantee efficiency and safety.

Numerous research efforts and investigations have been done to help reduce the risk in offshore crane operations. Focusing exclusively on the heave compensation problem, two different approaches have been extensively investigated. The first technique, Passive Heave Compensation (PHC) [1], was the first to be proposed and is the simplest of these two approaches. A PHC system can simply be modeled as a spring damper system by means of hydraulic cylinders and compressors. The second method, Active Heave Compensation (AHC) [2], differs from PHC by having controlled actuators that actively try to compensate for the heave movements. To monitor the ship movements, commercial offshore cranes usually adopt some motion detection units, e.g. Inertial Measurement Unit (IMU) and Motion Reference Unit (MRU). Then, according to this data input, a control system calculates how the actuators have to react to the movements. The actuators can be electric or hydraulic winch systems or hydraulic cylinders.

Due to the challenging crane operational scenario in real applications, several studies have been performed by using a computer-simulated environment. For instance, a heave compensation system based on heave motion prediction and an inversion based control strategy was proposed in [3]. In particular, a combination of a trajectory tracking disturbance decoupling controller and a prediction algorithm was presented and evaluated with simulation and measurement results. In [4], our research group presented a modular prototyping system architecture that allows for modelling, simulation and control of different robotic arms by using the *Bond Graph Method*. The resulting models are simulated in a virtual environment and controlled using the same input haptic device, which also provides the user with a valuable force feedback.

However, a simulation approach is always limited when compared to a realistic experimental setup. For this reason, other researchers explored the possibility of replicating a laboratory experimental arrangement for performing these kinds of studies. For example, an inverse kinematic control strategy that uses the actuation capability of two cranes (hoist lengths and boom angles) to keep its load fixed in inertial space regardless of the motion of the ship on which the cranes are mounted was presented in [5]. Unique crane commands are computed using a minimum norm solution and a dynamic simulation can be achieved. A final verification of the system was performed using two cranes mounted on a motion controlled platform. In [6], a method for reducing the cargo pendulation was proposed based on the control of the slew and luff angles of the crane boom. The effectiveness of the method was demonstrated in a fully nonlinear three-dimensional computer simulation and in an experiment with a scale model of the considered crane mounted on a platform moving with three DOFs.

However, most of these previous works focus on the development and validation process of a specific control method for very distinct crane models. To the best of our knowledge, a general framework that allows for both reproducing in a laboratory setup the same challenging operation scenario as that of maneuvering offshore cranes and for testing different models and control approaches has not been released yet.

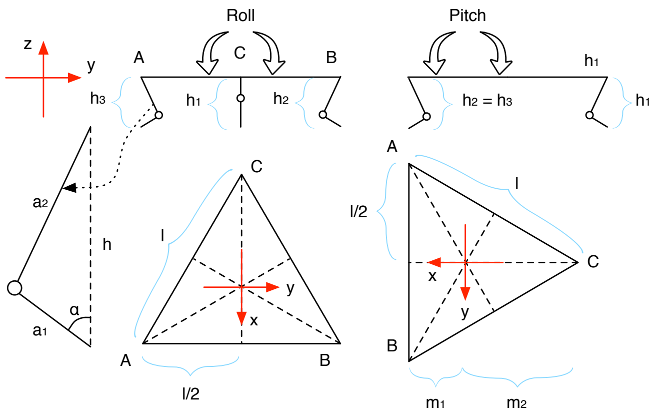


Fig. 2: Geometric characteristics of the considered motion platform: $a_1 = 150$mm, $a_2 = 330$mm, $l = 1075$mm, $m_1 = 310.3$mm, $m_2 = 620.7$mm.

III. System Architecture

In this section, the main components of the proposed architecture are presented. We first illustrate the considered motion platform from both a kinematic point of view as well as from a control point of view. Then the robotic arm is described focusing on the adopted control approach. Finally, the proposed integrated control system is depicted.

A. Motion Platform

A 3D model of the adopted motion platform is available on our public repository. This model is a type of parallel robot that incorporates three DOFs. It consists of three arms connected to universal joints at the top base. Each joint is actuated by a motor allowing for controlling the corresponding corner of the top base. The rotation range of each joint is limited to 125° which corresponds to the joint pointing straight up, and the corresponding platform corner to have its maximum height. Any higher value of the joint angle would make the corresponding corner of the platform to decline again.

Referring to Fig. 2, the design of the platform allows for movements along the Z axis (heave) and for rotations along the X and Y axes (roll and pitch, respectively). Given a desired heave position, h, each of the platform corners is raised or lowered to accommodate the position. For each corner of the equilateral triangle, h can be calculated as follows:

$$h = a_1 \cos(\alpha) + \sqrt{a_2^2 - a_1^2 \sin^2(\alpha)}, \qquad (1)$$

where a_1 is the lower arm, a_2 is the upper arm and α is the joint angle.

Concerning the roll movement, the height difference between h_2 and h_3 can be calculated as follows:

$$\Delta(h_2, h_3) = \sin(\phi) l, \qquad (2)$$

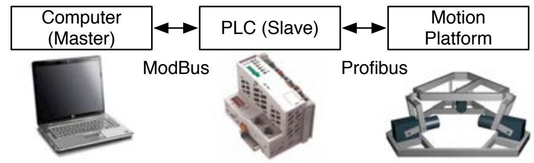


Fig. 3: To control the motion platform, a master-slave architecture is used with the controller acting as a master and the PLC as a slave.

where l is the length of the top base triangle and ϕ is the roll angle, which consequently can be found as:

$$\phi = \frac{\arcsin(\Delta(h_2, h_3))}{l}. \qquad (3)$$

Concerning the pitch movements, the height of h_2 and h_3 can be calculated as follows:

$$h_2 = h_3 = -\sin(\theta) m_1, \qquad (4)$$

where θ is the pitch angle and m_1 is shown in Fig. 2. h_1 can be calculated as follows:

$$h_1 = \sin(\theta) m_2, \qquad (5)$$

where m_2 is shown in Fig. 2. Consequently, the pitch angle, θ, can be obtained as follows:

$$\theta = \arcsin\left(\frac{\Delta(h_2, h_3) - h_1}{m_1 + m_2}\right). \qquad (6)$$

To simulate a realistic application scenario, the control system that actuates the motion platform is independent from the control system that operates the robotic arm. In particular, the motion platform is controlled by using a hardware platform based on a commercial *Programmable Logic Controller* (PLC) [7]. The control architecture, which is shown in Fig. 3, fully exploits the standard programming tools and the multi-tasking features offered by the PLC standard. By using the *Modbus* protocol [8], a master-slave pattern is set up with the controller acting as a master and the PLC as a slave. The three axes of the motion platform are driven by DC motors (203V). The motors are interfaced to a motor controller. In particular, a programmable power supply board is used to avoid buying costly *H bridge* circuits. This board can be remotely controlled from the PLC via Profibus [9]. Besides, the motor revolution is controlled by means of inverters.

B. Robotic Arm

The robotic arm that is placed on top of the presented motion platform is a *Kuka KR 6 R900 SIXX* manipulator. This manipulator is a 6 DOFs robotic arm with a slim design and a small footprint. The forward kinematics (FK) of this manipulator can be easily calculated by applying the standard Denavit-Hartenberg method [10]. In particular, the kinematics

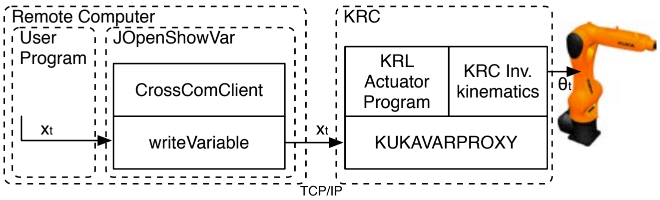


Fig. 4: *JOpenShowVar* allows for controlling the robotic arm by using the standard kinematics provided with the KRC. Alternatively, different control methods can be implemented according to current needs.

equations of a serial chain of 6 links like the considered robot, with joint parameters θ_i are given by:

$$T_A = {}^{0}_{6}T = \prod_{i=1}^{6} {}^{i-1}_{i}T(\theta_i), \tag{7}$$

where ${}^{i-1}_{i}T(\theta_i)$ is the general homogeneous transformation matrix from the frame of link i to link $i-1$.

The robot can be operated by the user by means of a standard joystick. To efficiently control the robot, the open-source cross-platform communication interface provided by *JOpenShowVar* [11] is used. This choice is motivated by the fact that *JOpenShowVar* allows researchers to implement alternative control algorithms according to current needs. For a more detailed introduction to *JOpenShowVar*, the reader can refer to [11] or to [12]. The control architecture is shown in Fig. 4. It is a client-server architecture with *JOpenShowVar* running as a client on a remote computer and *KUKAVARPROXY* acting as a server on the *Kuka Robot Controller* (KRC). *JOpenShowVar* locally interacts with the user program and remotely communicates with the *KUKAVARPROXY* server via *TCP/IP*.

In this preliminary study, the standard kinematics provided with the KRC is used to control the arm, as illustrated in Fig. 4. The user program simply works as a driver for the input device and uses the *writeVariable* method of *JOpenShowVar* to forward the end-effector's target position, $\mathbf{x}_t$, to a *Kuka Robot Language* (KRL) program, where the standard KRC inverse kinematics is used to calculate the desired joint angles θ_d.

C. Integrated Control System

The integrated control system architecture is shown in Fig. 5-a. It is a client-server architecture with the input device running as a client and communicating with a server where the logic of the control algorithm is implemented. The sever is implemented by following strict real-time criteria including multi-threading and synchronised methods. In the following, the key elements of the integrated control system will be presented referring to Fig. 5-a.

1) Wave generation: Random sinusoidal generators are used to reproduce the waves effect and to generate the input signal for the motion platform. The signal is generated as follows:

$$signal = \begin{bmatrix} A_h \sin(2\pi f t + \Omega) \\ A_\phi \sin(2\pi f t + \Omega) \\ A_\theta \sin(2\pi f t + \Omega) \end{bmatrix}, \tag{8}$$

where A_h is a random heave amplitude with uniform distribution in the range $[0,150]$mm, A_ϕ is a random roll amplitude with uniform distribution in the range $[0,100]$mm, A_θ is a random pitch amplitude with uniform distribution in the range $[0,100]$mm, f is a random frequency variable with uniform distribution in the range $[0,0.1]$Hz and Ω is a random phase variable with uniform distribution in the range $[-\pi,\pi]$. By using the kinematics of the platform, the corresponding joint angles, α, are calculated and used to actuate the motors.

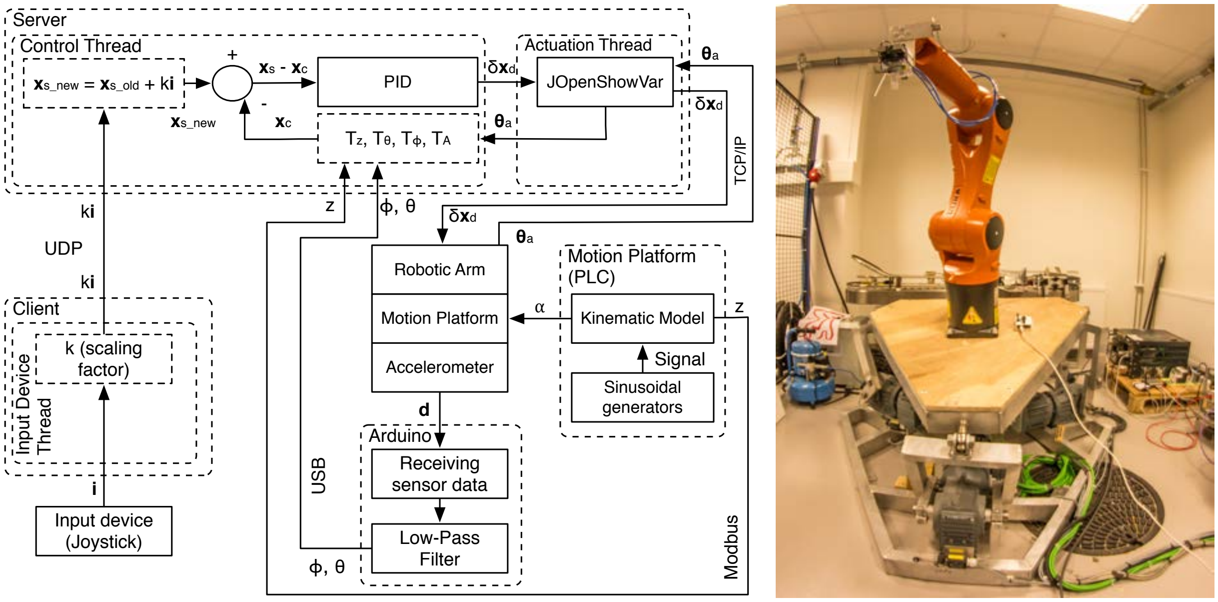
2) Heave, roll and pitch detection: To monitor the platform roll and pitch movements, an accelerometer sensor is used. The raw data of the movements, $\mathbf{d}$, is collected and received by a controller board. In particular, an *Arduino Uno* board [13] based on the *ATmega328* micro-controller is used. *Arduino* is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. On the software side, *Arduino* provides a number of libraries to make programming the micro-controller easier. The choice of using *Arduino* boards makes the presented framework easy to maintain and makes it possible to add new features in the future. The raw data is filtered from noise and the roll and pitch angles, ϕ and θ, are sent to the server by using the Universal Serial Bus (USB). Concerning the heave movements, the displacement along the z axis is obtained directly by reading the actual angles of the motion platform and by applying the forward kinematics. Then z is sent to the server by using the *Modbus* protocol.

3) Input device: In this study, a standard joystick is used as a universal input device on the client side. Each degree of freedom of the joystick corresponds to a translational axis in the workspace of the manipulator to be controlled. The joystick works as a position proportional replica whose motion maps exactly to the motion of the arm. A movement of the joystick in a particular direction will produce a translational motion of the robot's end-effector in the same direction, at a velocity proportional to the joystick displacement. When the operator's hand is removed from the joystick, the latter automatically returns to its starting point while the robot's end-effector keeps the last position. The joystick signal, $\mathbf{i}$, is scaled with a scaling factor, k, to fit the robot's workspace and then it is sent to the server by using the UDP protocol.

4) Server: In the following, the threads that run on the server side are described.

Control Thread: The *Control Thread* receives the following parameters:

- the scaled input signal from the joystick, $k\mathbf{i}$;
- the displacement, z, from the motion platform;



(a) (b)

Fig. 5: (a) the proposed integrated architecture: a client-server model is adopted. The server is implemented by following strict real-time criteria including multi-threading and synchronised methods. (b) the physical motion platform and the adopted robot

- the roll and pitch angles, ϕ and θ, from the accelerometer;
- the actual joint configuration from the manipulator, $\theta_{\mathbf{a}}$.

The current global robot's end-effector position, $\mathbf{x}_c$, can be obtained by using the following transformation matrix, T_c:

$$T_c = T_z T_\theta T_\phi T_A, \tag{9}$$

where T_z is the heave transformation matrix, T_θ is the pitch transformation matrix, T_ϕ is the roll transformation matrix and T_A is the arm transformation matrix.

At each control iteration, a set point, $\mathbf{x}_s$, is determined for the robot's end-effector as follows:

$$\mathbf{x}_{snew} = \mathbf{x}_{sold} + k\mathbf{i}, \tag{10}$$

where $\mathbf{x}_{snew}$ is the new set point and $\mathbf{x}_{sold}$ is the set point from the previous control iteration. The initial set point can be decided by the operator.

Successively, the difference between $\mathbf{x}_{snew}$ and $\mathbf{x}_c$ is calculated so that the corresponding sampling point configurations, $\delta\mathbf{x}_d$ (PID output), are obtained. To ensure smooth movements for the manipulators it is necessary to generate trajectories out of these given sampling points. A well-suited trajectory is the basic prerequisite for the design of a high-performance tracking controller and ensures that no kinematic nor dynamic limits are exceeded. Such a controller guarantees that the controlled robot will follow its specified path without drifting away. Therefore, feedback control has to be applied to be able to compensate for external disturbances as well as for disturbances from communication time delays. Note that time data is a free parameter because the sampling time of the mapping algorithm is generally not constant. As a possible solution for generating well-suited trajectories a Proportional Integral Derivative (PID) controller is used for each translational axis. To tune the PID parameters, different methods can be used, such as the one proposed in [14].

Actuation Thread: The *actuation thread* is used to communicate with the Kuka robot. This thread receives $\delta\mathbf{x}_d$ and uses the *writeVariable* method of *JOpenShowVar* to send the actuation values to the robot. In addition, the actual joint configuration, θ_a, is read by using the *readVariable* method of *JOpenShowVar* and sent back to the *Control thread*.

IV. Simulations and Experimental Results

The physical motion platform and the adopted robot are shown in Fig. 5-b. Related simulation are carried out to test the proposed framework. In detail, a time plot for the robot's end-effector position is performed. Fig. 6 shows a time plot for the robot's end-effector position. Active compensation is performed except for the highlighted time segment. It should be noted that the end-effector's movements are significantly affected by the motion of the platform when no active compensation is used. Contrary, wave effects are almost suppressed when adopting the proposed control method.

V. Conclusions and Future Work

This paper presents the features of a flexible framework that allows for reproducing in a laboratory setup the chal-

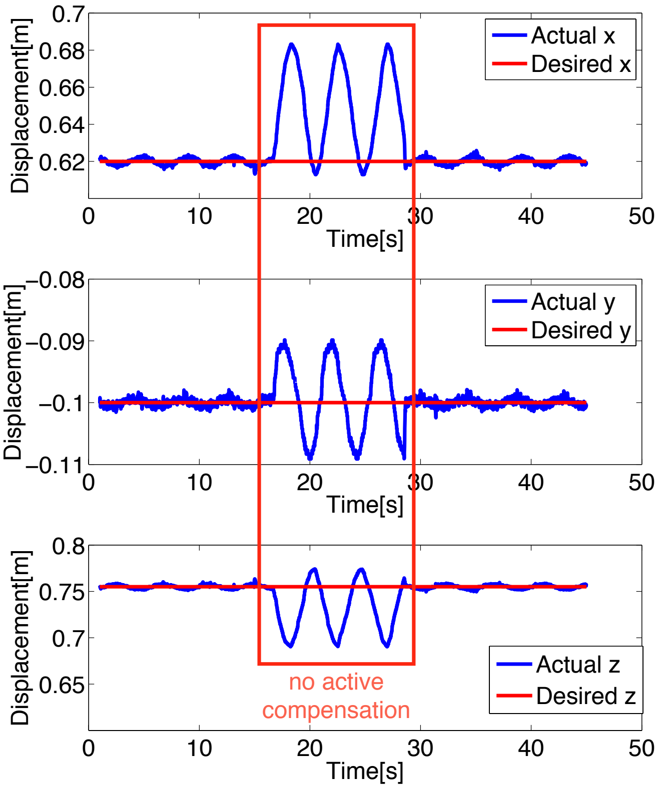


Fig. 6: A time plot for the robot's end-effector position is performed. Active compensation is performed except for the highlighted time segment. The end-effector's movements are significantly affected by the motion of the platform when no active compensation is used. Contrary, wave effects are almost suppressed when adopting the proposed control method.

lenging operation scenario of controlling offshore cranes. The system is built on open-source software and hardware and it can be used for testing different control algorithms as well as for training purposes. In the future, different control algorithms such as the ones implemented in [15] or in [16] may be tested as alternatives to the standard kinematic method. One more possibility that we are considering as future work is the integration of the proposed system with the integrated flexible maritime crane architecture that we recently developed [17]. Finally, some effort should be put in the standardisation process of the proposed framework to make it even more reliable for both the industrial and the academic practice. It is the opinion of the authors that the key to maximising the long-term, macroeconomic benefits for the robotics industry and for academic robotics research relies on the closely integrated development of open content, open standards, and open source.

VI. Acknowledgments

The contribution of the students Håkon Østrem, Håkon Eikrem and Bjarne Humlen in the implementation of this work is highly appreciated by the authors. This work was partly supported by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 - AMOS.

References

[1] J. Ni, S. Liu, M. Wang, X. Hu, and Y. Dai, "The simulation research on passive heave compensation system for deep sea mining," in *Proc. of the IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China*, 2009, pp. 5111–5116.

[2] S. Kuchler, T. Mahl, J. Neupert, K. Schneider, and O. Sawodny, "Active control for an offshore crane using prediction of the vessels motion," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 2, pp. 297–309, 2011.

[3] J. Neupert, T. Mahl, B. Haessig, O. Sawodny, and K. Schneider, "A heave compensation approach for offshore cranes," in *Proc. of the IEEE American Control Conference, Seattle, Washington, USA*, 2008, pp. 538–543.

[4] F. Sanfilippo, H. P. Hildre, V. Æsøy, H. Zhang, and E. Pedersen, "Flexible modeling and simulation architecture for haptic control of maritime cranes and robotic arm," in *Proc. of the 27th European Conference on Modelling and Simulation (ECMS), Aalesund, Norway*, 2013, pp. 235–242.

[5] F. A. Leban, J. Diaz-Gonzalez, G. G. Parker, and W. Zhao, "Inverse kinematic control of a dual crane system experiencing base motion," *Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–1, 2014.

[6] Z. Masoud, A. Nayfeh, and D. Mook, "Cargo pendulation reduction of ship-mounted cranes," *Nonlinear Dynamics*, vol. 35, no. 3, pp. 299–311, 2004.

[7] W. Bolton, *"Programmable logic controllers"*. Access Online via Elsevier, 2009.

[8] I. Modbus, "Modbus messaging on TCP/IP implementation guide v1. 0a," *North Grafton, Massachusetts (www.modbus.org/specs.php)*, 2004.

[9] PROFIBUS & PROFINET International (PI). (2014, November) "PROFIBUS". [Online]. Available: http://www.profibus.com/

[10] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2008.

[11] F. Sanfilippo, L. I. Hatledal, H. Zhang, M. Fago, and K. Y. Pettersen, "JOpenShowVar: an open-source cross-platform communication interface to kuka robots," in *Proc. of the IEEE International Conference on Information and Automation (ICIA), Hailar, China*, 2014, pp. 1154–1159.

[12] F. Sanfilippo, L. I. Hatledal, H. Zhang, M. Fago, and K. Y. Pettersen, "Controlling Kuka industrial robots: Flexible communication interface JOpenShowVar," *IEEE Robotics & Automation Magazine*, 2015, manuscript submitted for publication.

[13] Arduino. (2014, November) "Arduino, an open-source electronics prototyping platform". [Online]. Available: http://arduino.cc/

[14] I. Pan, S. Das, and A. Gupta, "Tuning of an optimal fuzzy pid controller with stochastic algorithms for networked control systems with random time delay," *ISA transactions*, vol. 50, no. 1, pp. 28–36, 2011.

[15] F. Sanfilippo, L. I. Hatledal, H. G. Schaathun, K. Y. Pettersen, and H. Zhang, "A universal control architecture for maritime cranes and robots using genetic algorithms as a possible mapping approach," in *Proc. of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China*, 2013, pp. 322–327.

[16] F. Sanfilippo, L. I. Hatledal, H. Zhang, and K. Y. Pettersen, "A mapping approach for controlling different maritime cranes and robots using ANN," in *Proc. of the IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China*, 2014, pp. 594–599.

[17] F. Sanfilippo, L. I. Hatledal, A. Styve, H. Zhang, and K. Y. Pettersen, "Integrated flexible maritime crane architecture for the Offshore Simulation Centre AS (OSC): A flexible framework for alternative maritime crane control algorithms," *submitted to the IEEE Journal of Oceanic Engineering*, 2015.