A Benchmarking Framework for Control Methods of Maritime Cranes Based on the Functional Mockup Interface

Filippo Sanfilippo, Lars Ivar Hatledal, Kristin Ytterstad Pettersen and Houxiang Zhang

Abstract

A benchmark framework for advanced control methods of maritime cranes is presented based on the use of the Functional Mockup Interface (FMI). The system integrates different manipulator models, all the corresponding hydraulic systems, various vessels, and the surrounding environment for visualisation. Different control methods can be transparently implemented and tested. A set of routine tests, different cost functions and metrics are provided – taking into account several factors, including position accuracy, energy consumption, quality, and safety for both the cranes and the surrounding environment. The concept of operation profiles (OP) is introduced, allowing for definition of different standard transporting and lifting operations. By considering task-oriented routines, this benchmark suite allows the comparison of different control methods independently from the specific crane model to be controlled.

Two alternative control methods for maritime cranes based on the use of artificial intelligence (AI) are extensively compared. The first method is based on the use of genetic algorithms (GA), while the second method involves the use of particle swarm optimisation (PSO). Simulation results are presented for both methods.

Index Terms

Maritime cranes, control methods, benchmark, functional mockup interface, artificial intelligence, machine learning.

I. Introduction

Some similarities can be identified between robotic arms and cranes. Both have a number of links serially attached to each other by means of joints that can be moved by some type of actuator. In both systems, the end-effector of the manipulator can be moved in space and placed in any desired location within the system's workspace, and it can carry a certain amount of load. However, traditional cranes are usually relatively big, stiff, and heavy because they normally need to move heavy loads at low speeds. While industrial robots are ordinarily smaller, they usually move small masses and operate at relatively higher velocities. Therefore, cranes are commonly actuated by hydraulic valves, while robotic arms are driven by servo motors or pneumatic or servo-pneumatic actuators.

Maritime cranes, compared with robotic arms, rely on a much more complex model of the environment with which they interact. Therefore, their control is always a challenging task, which involves many problems such as load sway, positioning accuracy, wave motion compensation, and collision avoidance.

This work is partly supported by the Research Council of Norway through the Centres of Excellence funding scheme, project number 223254, and the Innovation Programme for Maritime Activities and Offshore Operations, project number 217768.

Filippo Sanfilippo is the corresponding author and he is with the Department of Engineering Cybernetics, NTNU – Norwegian University of Science and Technology, 7491 Trondheim, Norway. filippo.sanfilippo@ntnu.no, http://filipposanfilippo.inspitivity.com.

Lars Ivar Hatledal and Houxiang Zhang are with the Department of Ocean Operations and Civil Engineering, NTNU – Norwegian University of Science and Technology, Postboks 1517, 6025 Aalesund, Norway. [laht, hozh]@ntnu.no.

Kristin Ytterstad Pettersen is with the Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, NTNU – Norwegian University of Science and Technology, 7491 Trondheim, Norway. kristin.y.pettersen@ntnu.no.

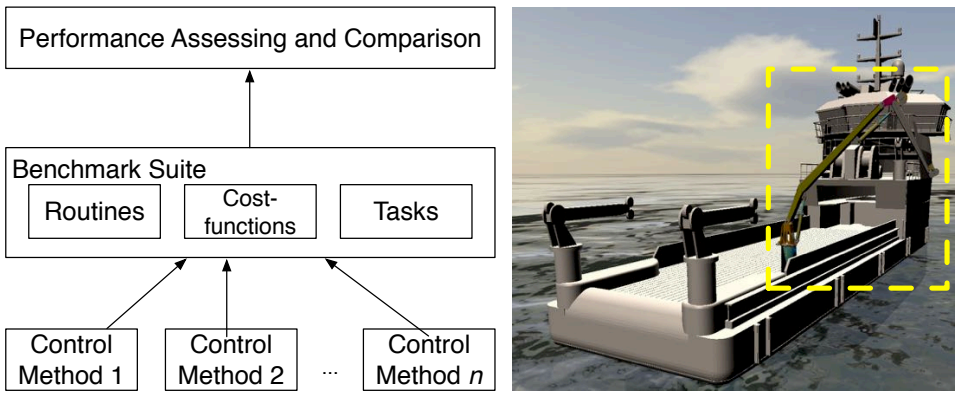


Fig. 1: The realisation of a benchmark suite for advanced control methods of maritime cranes.

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. When considering working efficiency and safety, this kind of control is extremely difficult to manage and relies on operators who have extensive experience with a high operating skill level. This is why more flexible and reliable control approaches for maritime cranes are needed.

The fact that maritime cranes normally utilise a flexible rope to make a hoisting motion that lifts or lowers the payload increases the complexity of crane operations. This leads to a unique characteristic of cranes in comparison with robotic manipulators. Since the suspending ropes are flexible, one of the major problems that crane systems face is a large sway angle in the payload motion during transfer. This is different from robotic manipulator systems, where the payload is often considered rigidly connected with the end-effector. As such, the control objectives of crane systems do not consist solely of joint and end-effector position trajectory tracking. Usually, rope winch solutions are also necessary to increase efficiency, operator safety, and comfort. Even though rope-related problems are relevant to the field of maritime cranes, the focus of this work is the development of a benchmarking methodology for assessing the system performance.

As opposed to the field of robotic arms, where at least a few benchmark suites and methods for estimating the efficiency of the considered control approach already exist, the field of maritime cranes lacks a universally recognised benchmarking methodology for assessing system performance. This is the main reason why it currently is extremely difficult not only to compare results of alternative control approaches, but also to assess the quality of the research presented by different authors [1].

This paper outlines a methodology for performing simulation-based verification and benchmarking in the area of maritime cranes control. By combining the rapid-prototyping approach with the concept of interchangeable interfaces, a simulation and benchmarking framework for advanced control methods of maritime cranes is presented. The system is based on the use of the Functional Mockup Interface (FMI) [2], which is a tool and independent standard for the exchange of dynamic models and for co-simulation. The framework makes it possible to efficiently integrate different manipulator models, all the corresponding hydraulic systems, various vessels, and the surrounding environment for visualisation. Different control methods can then be transparently implemented and tested. Based on the FMI, the underlying idea of the benchmark suite proposed in this paper is shown in Fig. 1. The suite includes a set of routine tests, different cost functions, and metrics that take into account several factors including position accuracy, energy consumption, quality, and safety for both the cranes and the surrounding environment. Each proposed routine test is task-oriented and consistently reproduces realistic on-board

operation scenarios. The concept of operation profiles (OP) is introduced, allowing us to define standard transporting and lifting operations. To show the potential of the proposed benchmark framework, two alternative control approaches are extensively compared as case studies. These methods are based on the adoption of artificial intelligence (AI). Specifically, the first method is based on the use of genetic algorithms (GA) [3], while the second method involves the use of particle swarm optimisation (PSO) [4]. It should be noted that the considered control methods take into account only the heave compensation problem, not the sway angle suppression for the rope. All problems related to rope pendulation or wave impact on the payload are not considered in this work but can be taken into account at a later stage.

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 the adopted framework architecture and methodology. In Section IV, the proposed benchmark suite is presented. The two considered control methods are presented as case studies in Section V. Related simulations and results that aim to compare the two control methods are shown in Section VI. In Section VII, conclusions and future works are outlined.

II. Related research work

Benchmarking as a means of objective comparison and competition among researchers is and has always been of great interest in robotics. The practice of comparative evaluation of different algorithms becomes increasingly useful when applied to a well-defined system in a specific domain, as opposed to field robotics in general. For this reason, efforts are being made to establish standard benchmarks for several robotic fields, including grasping manipulators, mobile robots, human-robot interaction, and robotic arms. In this context, networks and societies, such as the European robotics research network (EURON) or the institute of electrical and electronics engineers (IEEE), play an important role in working on defining standard benchmarking methodologies [5].

Usually, when trying to benchmark several robotic systems, a standard reference environment, reference tasks, and related performance metrics are to be defined. However, it is difficult to define a benchmark that is commonly accepted by the research community, mainly because of divergent viewpoints on various problems. In addition, the risk of fostering the development of specialized solutions for an abstracted, standardised setting exists. When considering robotic arms, a quite common approach to avoid these problems consists in organising scientific competitions and contests where benchmarks are usually discussed and then accepted by all the participants. Different famous competitions are known for complete integrated robotic systems, such as the defense advanced research projects agency's (DARPA) Grand Challenges [6], or the RoboCup@Home [7] (a competitive scenario for service robots). Unfortunately, participation in such big events is usually limited to a few selected groups, often simply because of limited resources and the lack of necessary hardware. In addition, this approach is difficult to apply to maritime cranes, mainly because of their size and complex operation scenarios.

To overcome these limitations, another possible way to compare different control methods is to use a virtual environment that closely simulates the desired systems and allows replication of a set of reference tasks and related performance metrics. For instance, concerning the field of grasping manipulation, a software environment for the comparative evaluation of algorithms for grasping and dexterous manipulation is presented in [8]. This tool allows the reproduction of well-defined experiments in real-life scenarios in every laboratory and, hence, provides benchmarks that pave the way for objective comparison and competition in the field of grasping. Considering the robotic planning domain, a benchmark tool for multi-robot simulation is presented in [9]. In the field of autonomous mobile robots, a unified benchmark framework for evaluating and comparing motion algorithms for autonomous mobile robots and vehicles is introduced in [10]. Focusing exclusively on robotic arms, a control engineering benchmark problem with industrial relevance is presented in [11]. The process is a simulation model of a nonlinear four-masses

system, which should be controlled by a discrete-time controller that optimises performance for given robustness requirements. Nonetheless, the control problem concerns only the so-called regulator problem. In [12], a benchmark problem for robust feedback control of a manipulator is introduced. The system to be controlled is an uncertain non-linear two-link manipulator. The control problem concerns only disturbance rejection. The proposed model is validated by experiments on a real industrial manipulator. However, most of these previous works mainly consider only a specific manipulator model, and it is not possible to dynamically exchange models or control methods.

The field of maritime cranes lacks a universally recognized benchmarking framework for assessing system performance. To the best of our knowledge, no standard benchmarking tools are currently available in this field – neither in the form of competitions, nor as routines to run in a simulation environment.

III. Framework architecture and crane models

Since maritime cranes usually operate in a complex and challenging environment, it is almost impossible to perform tests and benchmarks in a real experiment setup. This section presents a framework that uses a simulation setup to reproduce the challenging operation scenario of controlling offshore cranes. This framework can be used for testing different control approaches by running different benchmarking routines.

A. Framework architecture

Our previous work on developing a flexible control architecture for maritime cranes and robots is presented in [3], [4]. This architecture allows for modelling, simulation, and control of different models of maritime cranes. The manipulators to be controlled can be simulated in a 3D visualisation environment, which provides the user with intuitive visual feedback. However, in this preliminary work, the nature of the crane actuators – whether hydraulic, pneumatic, electric, or mechanical – was not considered. This is a quite relevant limitation because, when compared to robotic arms, offshore cranes are usually much heavier, stiffer, and mostly hydraulically actuated. Classic robotic control therefore does not assure the same effectiveness for offshore crane control. Hence, it is important to include the hydraulic systems to the crane models to be controlled, as well as the vessel models and the surrounding environment. In this respect, the system to be designed can be seen as a complex cyber-physical system (CPS) [13]. In order to realise such a CPS, it is necessary to combine different heterogeneous models. Unlike more traditional systems, a full-fledged CPS is typically designed as a network of interacting elements with physical inputs and outputs instead of as standalone devices. In this regard, it is critical to define transparent and efficient interfaces between the models.

For these reasons, the authors have chosen the FMI as the fundamental binding agent between the different models of the proposed framework. This section will refer to several specific functions, variables, and configurations related to the FMI. For a more detailed introduction to the FMI, the reader can refer to [14]. In practice, the FMI implementation by a software modelling tool enables the creation of simulation models that can be interconnected and are called FMUs (Functional Mockup Units). The FMI covers the aspects of model exchange and of co-simulation. In this work, since the intention is to couple different heterogeneous models together, the FMI for co-simulation is used. In addition to the selected standard, a client-server architecture with a master-slave modelling pattern is adopted. The proposed framework architecture is shown in Fig. 2. The communication between client and server is implemented by using the Hypertext Transfer Protocol (HTTP) [15], and it supports WebSockets [16]. Consequentially, multiple instances of the client can be run remotely on a standard web browser. The models are distributed in a hierarchical organisation. The following subsections describe the key elements of the framework.

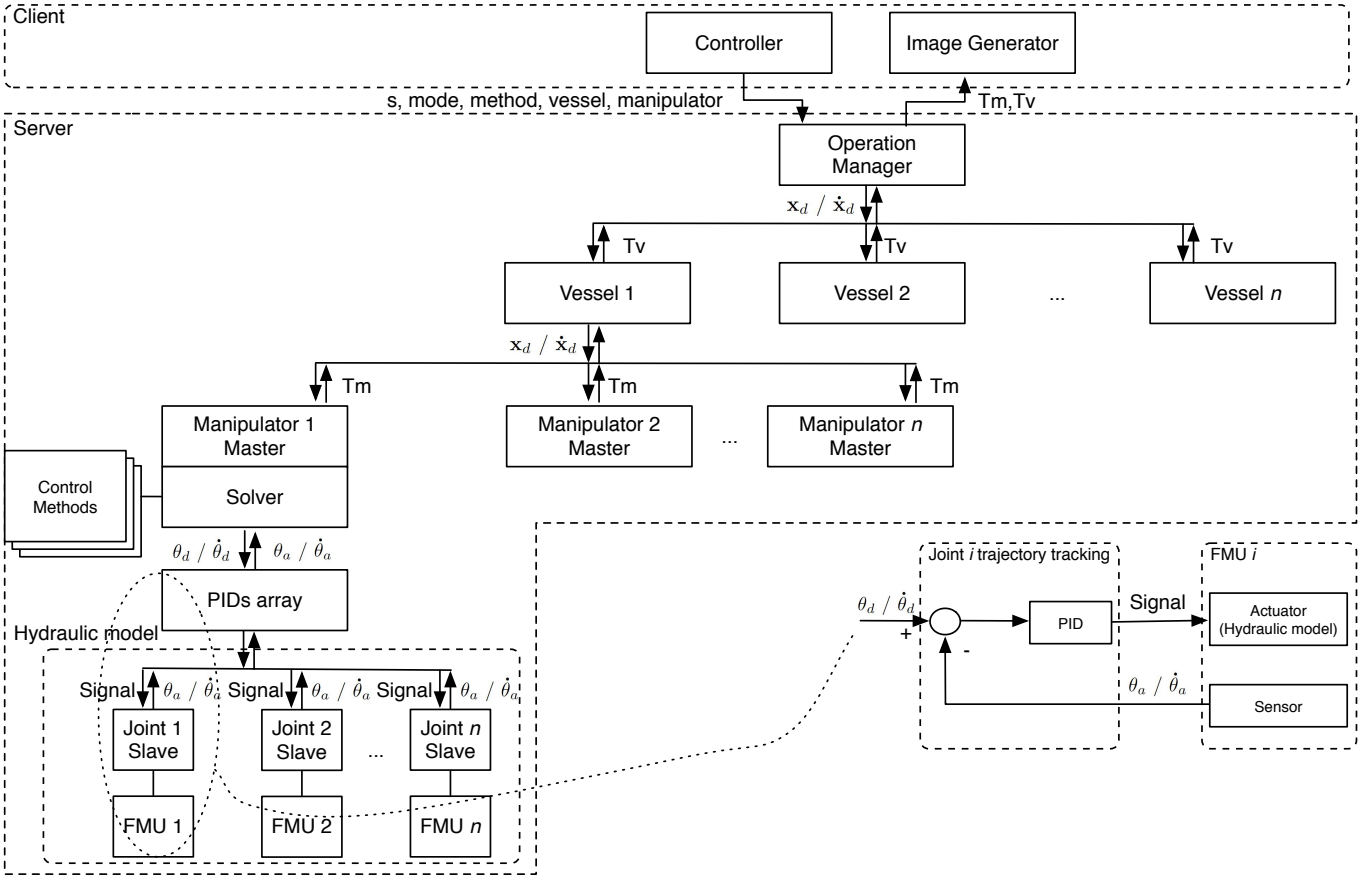


Fig. 2: The proposed framework architecture and a detail of the trajectory tracking approach (bottom right corner).

1) Client: the client essentially implements the visualisation layer, which consists of two components: the *Controller* and the *Image Generator*.

a) Controller: it includes different control inputs allowing the user to operate the models with different input devices, i.e. a joystick.

b) Image Generator: it generates the scene and all the visual components according to the simulated models. This component adopts the web graphics library (WebGL) [17], which is a JavaScript application programming interface (API) for rendering interactive 3D graphics and 2D graphics within any compatible web browser without the use of external plugins.

2) Server: the integration and models layers are implemented on the server side.

a) Operation Manager: it is possible to control the models in position mode or velocity mode. The user experience is substantially different in each case. When using the position control mode, the operator simply controls the crane tip position with constant velocity; when operating in velocity control mode, the operator also sets the end-effector velocity by using an input device, i.e. a joystick. To realise these two possible operation modes, when the operator manoeuvres the manipulator from the client side, a vector signal with no semantic, $\mathbf{s}$, is sent to the server together with the desired operation mode. According to the chosen scenario, the *Operation Manager* interprets the vector signal as the desired position, $\mathbf{x}_d$, or the desired velocity, $\dot{\mathbf{x}}_d$. In addition, the *Operation Manager* also receives a flag indicating the desired control method and two identification (ID) tags assigning the vessel model and the manipulator to be

controlled, respectively.

b) Manipulators: a master-slave pattern is adopted. Each manipulator behaves as a master, while each joint runs as a slave. Each manipulator can be controlled with a *Solver*, which implements a particular control method. According to the desired mode of operation, the control algorithm parses $\mathbf{x}_d$ or $\dot{\mathbf{x}}_d$ to the desired joint angles, θ_d, or desired joint velocities, $\dot{\theta}_d$, of the manipulator, respectively. Essentially, for all the different models to be controlled, the mapping methods have to implement the classic inverse kinematic equations that can be generalised as follows:

$$\theta_d = f_p^{-1}(\mathbf{x}_d), \tag{1}$$

concerning position control, and

$$\dot{\theta}_d = f_v^{-1}(\theta_a, \dot{\mathbf{x}}_d), \tag{2}$$

for velocity control, where θ_a is the the actual joint angles vector. The calculated joint angles, θ_d, or joint velocities, $\dot{\theta}_d$, are then forwarded to the hydraulic model of the crane in order to actuate the joints. In order to do this, these values need to be transformed into gain values as outlined in the next key step. As feedback from the hydraulic model, the actual joint angles, θ_a, and actual joint velocities, $\dot{\theta}_a$, are received and can be used by the selected control algorithm.

c) Trajectory Tracking: for all the models to be controlled, the different control methods calculate the corresponding sampling point configurations for the desired end-effector positions. In other words, each control algorithm works as a motion planner. 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 or 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 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. A possible solution for generating well-suited trajectories consists of using a Proportional-Integral-Derivative (PID) controller for each joint, as shown in Fig. 2 (bottom right corner). To tune the PID parameters, numerous methods exist, such as the one proposed in [18]. In this work, a trial-and-error tuning method has been adopted for the sake of simplicity.

d) Hydraulic Models: different cranes can be transparently modelled by using a modular approach. Various modelling tools can be adopted. For the sake of illustration, a 3 degree of freedom (DOF) knuckle boom crane model, including the hydraulic system, is developed to represent the mechanical properties. This particular crane is highlighted in Fig. 3. The actuators of the considered crane consist of one hydraulic motor at the foundation and base joint, and two hydraulic cylinders positioned between the base and the boom, and the boom and the jib, respectively.

Maritime cranes are complex systems. Different modelling approaches and software tools are available for modelling the dynamics of similarly demanding systems [19], [20]. However, most of these methods are usually suitable for only one specific disciplinary domain. For complex and multi-domain systems, like the considered hydraulic offshore crane, the challenge of interfacing these models tends to appear. For these reasons, the authors chose the bond graph method (BGM) [21] as the tool to model the system. In particular, the BGM is a highly modular energy-based approach for modeling and simulation of multi-domain dynamic systems [22]. One of the biggest advantages of using the BGM is that once the BG model of the system is ready, the system state equations can be algorithmically derived from it in a systematic manner. This process is usually automated using appropriate software, which can also derive equations in symbolic form. The so-called *20-sim* simulator [23] is used in this work. The BG modelling

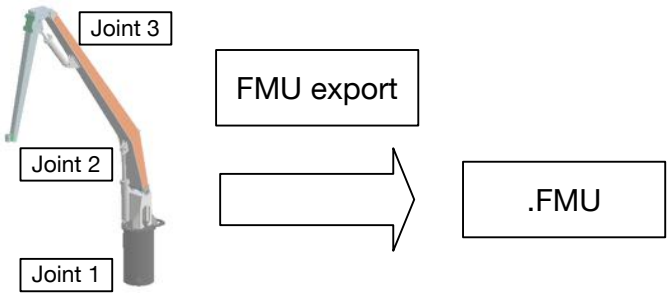


Fig. 3: The presented sub-models are automatically exported as FMUs for integration.

process is based on our previous work, which was initially presented in [22] and later developed in [24], [25]. For more details on modelling crane hydraulic systems using the BG method, the reader is referred to these works [22], [24], [25].

A modular prototyping system architecture is adopted in this work to build the model using the BG technique. This system is based on a library of crane beams, joints, and actuator models that can be used as components for simulating different cranes. Since the focus of the current work is the development of a benchmarking methodology for assessing system performance, the modelling of these components is beyond the scope of this paper.

It should be noted that our research group is planning to validated the crane model considered in this work by constructing a real smaller scale replica of the selected model.

The integration of the considered BG models with our framework architecture is possible thanks to the use of the FMI technology, which is supported by the *20-sim* simulation environment. The presented sub-models can be automatically exported as FMUs for integration. This process is illustrated in Fig. 3. It should be noted that the adopted FMI technology makes it possible to use any other modelling technique, thereby allowing the proposed architecture to be used as a research tool for investigating other modelling methodologies.

The proposed models can be controlled by different methods. Two novel control methods are presented as case studies in Section V.

IV. Benchmarks

This section proposes different quality measures and operational profiles. It should be noted that all the proposed methods are meant to be used for benchmarking different *Solvers*. Therefore, these metrics should be applied before the PID regulation process, in order to make a consistent comparison between different control methods. We classify the proposed quality measures into two groups, according to whether a direct or an indirect method is used to assess the considered properties.

A. Direct measures and metrics

A direct measure of a property of interest for a process is a measure done on that process whose value alone indicates the extent of the property of interest.

1) Position and joint error: from a static point of view, the accuracy of a crane control system can be evaluated by using a composed cost function that assesses two different contributions: the end-effector position error, a, and the joint configuration error, b. The first contribution can be assessed by measuring the Euclidean distance between the target position, $\mathbf{x}_t$, and the calculated position, $\mathbf{x}_c$, from the *Solver*:

$$a = d(\mathbf{x}_t, \mathbf{x}_c) = |\mathbf{x}_t - \mathbf{x}_c|, \tag{3}$$

where $\mathbf{x}_c$ is calculated by using forward kinematics, while the target position is given by the desired position:

$$\mathbf{x}_t = \mathbf{x}_d. \tag{4}$$

The second component of the proposed cost function considers the change in joint angles between two consecutive solutions, and it can be calculated as the Euclidean distance between the target joint configuration, θ_t, and the calculated joint configuration, θ_c:

$$b = d(\theta_t, \theta_c) = |\theta_t - \theta_c|, \tag{5}$$

where the calculated joint configuration is the output from the *Solver*. This contribution is considered in order to avoid multiple solutions when considering redundant manipulator models. The complete cost function is calculated as:

$$\text{cost} = \alpha \text{a} + \beta \text{b}, \tag{6}$$

where $\alpha, \beta \in \Re^+$ are weighting factors, such that $\alpha + \beta = 1$.

2) Joint torque: the joint torque effort, L, can be measured and used as a parameter of comparison between different crane control methods. For each particular control task, the joint effort can be assessed and calculated by using the following equation:

$$\begin{aligned} &\mathbf{T} = \int_0^{t_f} L(\theta_c, \dot{\theta}_c, \tau_c, t)\, dt \\ &\text{subject to} \qquad \theta_c(0) = \theta_{c_0}, \dot{\theta}_c(0) = 0, \\ &\qquad\qquad\qquad\quad \theta_c(t_f) = \theta_{c_f}, \dot{\theta}_c(t_f) = 0, \end{aligned} \tag{7}$$

where 0 is the initial time, t_f is the final time, θ_c is the calculated joint configuration, $\dot{\theta}_c$ is the calculated joint velocity, and τ_c is the calculated joint torque. In most cases, the effort, L, can be approximated by using the following equation:

$$L = \frac{1}{2} \|\tau_c\|^2. \tag{8}$$

A crane control method that requires less torque for the joints when considering a particular control task may be preferred to another control approach that requires a larger torque and consequently more energy to achieve the same task. In this regard, the joint torque can be also used as a cost function to be minimized in order to optimize planned motions when operating robotic arms [26].

3) Sway angle: unlike loads hoisted by cranes mounted on solid bases, load sway is affected by the ships motion when the load is hoisted by a maritime crane. The sway movements of the load have significant effects on the crane operations. Referring to Fig. 4, the sway angle of the load can be split into two components [27], longitudinal and lateral:

- ϕ defines the longitudinal sway angle of the load in the reference coordinate frame;
- χ defines the lateral sway angle of the load in the reference coordinate frame.

The variation over time of the sway angles, ϕ and χ, can be used as parameters of comparison between different crane control methods. This measure is a proposed method that can be used by researchers in the future.

It should be noted that for this current work, the payload sway angles were obtained in simulation. In fact, the payload sway angles can easily be retrieved in a simulated environment by adopting the proposed framework. However, when considering a real application scenario, the payload sway angles need to be measured or estimated. The challenge is that the payload sway angles can hardly be measured with sway angle sensors. In particular, even though sway angle sensors are adopted in ports, the measured angles

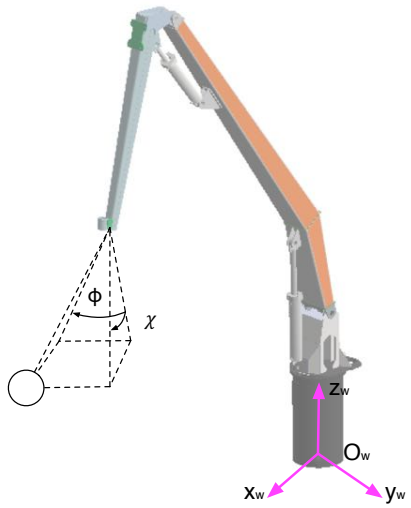


Fig. 4: The longitudinal and the lateral sway angles, ϕ and χ, respectively.

were not accurate until now [28]. Moreover, using sway angle sensors to measure the payload sway angles is relatively costly. Therefore, several researchers have developed different methods to estimate the payload sway angles. In [29], an angular velocity sensor is attached to the payload and a state observer is adopted to estimate the sway angles. This method does not require the use of a sway angle sensor. However, the estimated sway angles are still not highly accurate. In [30], the payload sway angles are estimated by using the crane's motion equations and a time compensation observer. However, some residual vibrations are still present in the estimated sway angles. In [28], a new method to estimate the payload sway angles without sensors is proposed. In this study, a reaction force observer is adopted to estimate first the power of the crane's winch motor and successively the payload sway angles. The efficiency of this method is verified by simulations and experiments. For a deeper understanding of the problem, the reader is referred to these works [28], [28]–[30]. A study of the methodology for measuring or estimating the payload sway angles is, however, beyond the scope of this paper.

To both reduce the effect of the payload pendulation and to minimise the wave impact during shipboard crane manipulation, our research group previously presented a novel approach to anti-sway control [31]. The proposed control mechanism is based on the concept of energy dissipation. An important advantage of the proposed methods for dissipating energy is that they are expected to be relatively easily combined with operator-induced actions. When a human operator is involved, it is unwanted and dangerous if the control algorithm suddenly performs actions opposite to those of the crane operator. Based on the same concept of energy dissipation, a combined heave compensation and anti-sway control approach for offshore crane operations was successively presented in [24]. The two control functions of heave compensation and anti-sway are transparently combined and simulated in an integrated modelling environment.

4) Total transfer time to target point: the total transfer time (TTT) to a target point can be defined as the total handling time that is necessary for the crane to reach a target point starting from a given initial configuration. The TTT may be considered as a parameter of comparison between different crane control methods. Intuitively, a crane control method that requires less TTT when considering a particular target point may be preferred to another control approach that requires more time to achieve the same task.

Maritime cranes are equipped with ropes. Assuming that during the crane operation the rope length is varied, the transfer process can be divided into three steps [32]: an accelerating motion step, a uniform

motion step, and a decelerating motion step. Therefore, using this perspective, the total transfer time, TTT, can be formally defined as follows:

$$TTT = \Delta t_a + \Delta t_u + \Delta t_d, \tag{9}$$

where Δt_a is the time interval spent during the accelerating motion step, Δt_u is the time interval used during the uniform motion step, and Δt_d is the time interval passed during the decelerating motion step.

It should be noted that the velocity of the crane boom will be constant during the uniform motion step, but at this time, the acceleration of the load is not zero, and some sway motion appears. For this reason, one of the main objectives of a crane control method is to keep the load sway angles, ϕ and χ, at zero during the uniform motion. Therefore, a general constrained minimisation problem may be written as follows:

$$\begin{aligned} &\min && TTT \\ &\text{subject to} && \phi = 0, \chi = 0. \end{aligned} \tag{10}$$

This formulation of the problem can be used as a guideline to design and compare different crane control methods.

It should also be noted that the presented direct measures represent different physical entities and therefore have different units of measurement (i.e., linear position, joint position, velocity, torque, sway angles, time). To combine and use the proposed direct measures as metrics components with different weighting coefficients, a normalisation procedure must be applied. This procedure is not addressed in this work. An alternative approach consists in formulating an optimisation problem involving more than one objective function. For instance, multi-objective optimisation [33] may be applied to realise a control method which requires high joint torque, but it results in a lower position and joint error or less payload oscillation.

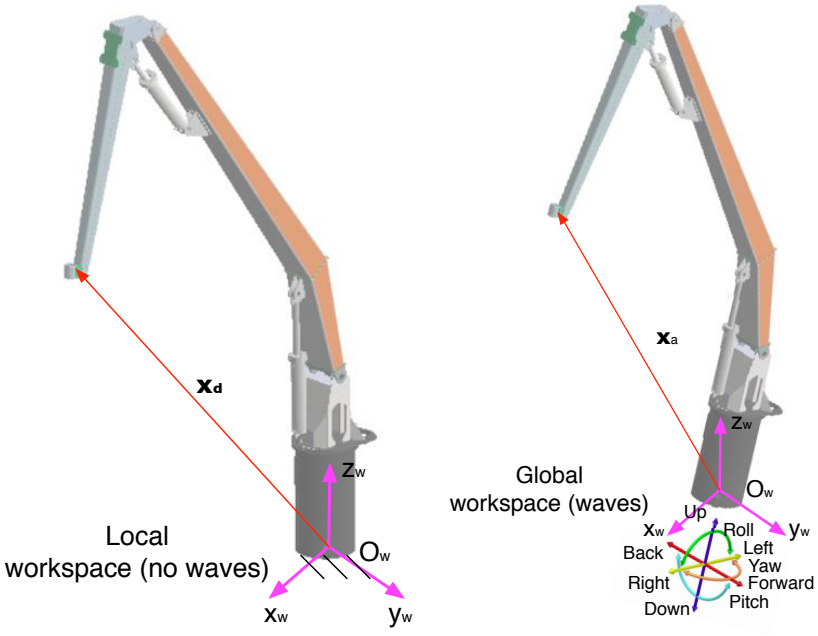
B. Indirect measures and metrics

An indirect measure of a property of interest for a process is a measure that is derived by measuring one or more different properties. By using those different measures, it is possible to determine the extent of the property of interest [34]. When considering maritime cranes, indirect measures may provide performance estimations of the adopted control method.

Unlike cranes mounted on fixed bases, offshore crane operations are significantly influenced by the ships motions caused by currents and waves. The dynamic forces generated from the heave motion of the vessel and the sway movements of the pendulate load have extensive effects on the crane structure and the lifting wire. Concerning the heave effect, much work and many investigations have been done to help reduce the risks in offshore crane operations [35], [24], [36].

One possible indirect measure may consist in running a heave compensation method for the considered crane model to see which control method gives the best performance. This approach can be seen as a safety test. Any heave compensation method can be used to run this test. For the sake of clarity, we will present a possible approach and then use it as an indirect method of comparison for the two case studies presented in Section V. The results of this comparison will be shown in Section VI.

Referring to Fig. 5, two different cases can be distinguished. In Fig. 5-a, the waves contribution is not considered. In order to consider the waves effect, a generalised model is depicted in Fig. 5-b by simply adding 6 DOF to the base of the crane. When manoeuvring the crane, the operator sets the desired end-effector position, $\mathbf{x}_d$, in the local workspace. However, because of the waves effect, it is necessary to



(a) (b)

Fig. 5: The idea behind the proposed heave compensation method: (a) the crane local workspace without waves contribution and (b) the global crane workspace with waves contribution.

consider the calculated end-effector position, $\mathbf{x}_c$, in the global workspace. By considering the difference between $\mathbf{x}_d$ and $\mathbf{x}_c$, the position error can be calculated:

$$\delta\mathbf{x} = \mathbf{x}_d - \mathbf{x}_c. \tag{11}$$

This error can be used as the input for any desired crane control method. For instance, the classical Jacobian method [37] can be applied as follows:

$$\delta\theta = J^{-1}(\theta)\delta\mathbf{x}, \tag{12}$$

where $\delta\theta$ is the required joint velocity vector and $J(\theta)$ is the Jacobian matrix.

In order to indirectly evaluate the performance of the considered control method to compensate for the waves impact, one of the previously presented direct measures can be used.

C. *Operational profiles and routine tests*

The reliability of a control system also depends on how the operator uses the crane. A good reliability estimation can be realised by testing the control methods to be compared as if they were used in a real operation scenario. By borrowing the idea from the software engineering domain [38], we introduce the concept of the *operational profile* (OP) as a quantitative characterisation of how the control methods are used by the operator. Different OP can be created, and each of them can be repetitively executed by using the control methods that are to be compared. These tests can be run in a batch mode within a simulation environment.

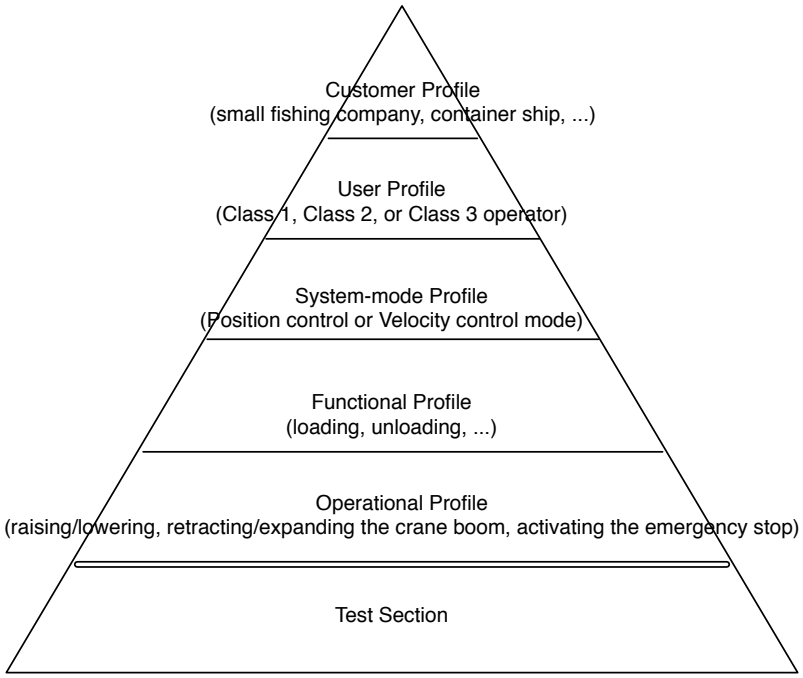


Fig. 6: The process for developing an OP for a crane control system.

A *profile* (P) is a set of independent possibilities, called *states*, and their associated probability of occurrence. Each state defines a particular configuration for the crane – either in the joint or in Cartesian space. For example, if state A occurs 60 percent of the time, B occurs 30 percent, and C occurs 10 percent, the profile is $P = [(A, 0.6), (B, 0.3), (C, 0.1)]$. An OP is the set of independent states that a control system performs and their associated probabilities.

The process for developing an OP for a crane control system is shown in Fig. 6. It involves one or more of the following steps.

1) Find the customer profile: as for any other product, a customer is the individual, group, or organisation that purchases the crane control system. A *customer profile* consists of an array of independent customer types. A *customer type* is one or more customers in a group who intend to use the control system in a relatively similar manner, and in a substantially different manner from other customer types. For instance, a *customer type* for a particular crane control system may vary from a small fishing company, which can use a small crane with little payloads, to a big container ship company, which may use a big and strong crane model to lift huge payloads. The *customer profile* is the set of customer types and their associated probabilities of using the control system.

2) Establish the user profile: the user of a crane control system may be different from the customer. The user is the crane operator. Several kinds of users may use the same control system for a specific model of crane. For instance, users can be categorised according to their level of experience. According to the crane regulations API-RP-2D [39] and latest addition, ABS, ANSI Standards, ASME B30 Rules, and OSHA rules, crane operators can be classified as follows:

- Class 1 Operator: no restrictions or limitations;
- Class 2 Operator: limited to making lifts under 50% of crane's lifting capacity at any radius, limited to loading and unloading boats in calm seas only, supervised with lift over 50% of crane's lifting capacity at any radius, supervised with personnel lifts;

- Class 3 Operator: limited to making lifts under 50% of crane's lifting capacity at any radius (with direct supervision only), limited to loading and unloading boats in calm seas only (with direct supervision only), cannot make personnel lifts under any conditions (with or without direct supervision).

The *user profile* is the set of user types and their associated probabilities of using the control system.

3) Define the system-mode profile: a system-mode is a way that a control system can operate. For instance, a crane control system may essentially operate in two different modes, which are position and velocity control. System-modes can be considered of as independent operational scenarios. Normally, a control system allows for switching among modes sequentially. The system-mode profile is represented by the list of system modes and their corresponding occurrence probabilities.

4) Determine the functional profile: after a good system-mode profile has been developed, the focus should turn to evaluation of each system mode for the functions performed during that mode, and then assigning probabilities to each of the functions. Functions are essentially tasks that the crane operator can perform with the control system. For example, loading or unloading boats can be seen as a function. In order to assign occurrence probabilities, the best data source consists of usage measurements taken in the field. These measurements may be obtained from system logs or data storage devices. Occurrence probabilities computed from the historical data should be updated to account for new control functions, users, or environments.

5) Determine the operational profile itself: a function may include several *operations*. Examples of such *operations* include raising or lowering the crane boom, retracting or expanding the crane boom, or activating the emergency stop. In turn, *operations* are made up of many *run categories*. In fact, the same *operation* may be performed with different specific requirements. For instance, the exact same *operation* of raising the crane boom may be required to be executed at low speed or high speed in terms of end-effector velocity. Finally, each *run category* can be partitioned into different *run types* according to the particular input state. For instance, the exact same *operation* of raising the crane boom may start with different initial conditions in terms of payload and initial joint configuration.

Once the operational profile is determined, it can be used to run a routine test section against different crane control methods, as shown in Section VI. It should be noted that the definition of *operational profiles* is very useful during the designing phase of a crane control system.

V. Case studies

A. Background and motivation

In order to design a control algorithm for a crane, the kinematic properties of the system need to be found. One approach is to derive the inverse kinematics (IK) model to be controlled. Typically, this approach enables researchers to either introduce analytical methods, which offer exact solutions for simple kinematic chains, or propose solutions based on numerical methods. However, when considering arms with redundant degrees of freedom, the inverse kinematics can have multiple solutions, and therefore singularity problems could arise. In addition, this method is not very flexible, especially when planning to control different arms using a universal input device, because then several IK models are needed, one for each arm or crane to be controlled.

An alternative solution to the problem may consist of using methods that derive the kinematic properties by applying an optimisation approach based on artificial intelligence (AI) [40]. AI is a branch of computer science dealing with the simulation of intelligent behaviour in computers. Regarding maritime cranes, using methods based on AI may make it possible to automatically obtain the kinematic properties of different models, and new models can also easily be obtained. In addition, several authors have highlighted

the fact that controlling the motion of maritime crane systems is a difficult non-linear problem, which is not easily solvable with conventional control methods. Therefore, the use of optimisation methods based on AI may be beneficial in this perspective.

For example, a popular and well-established AI strategy to optimise non-linear systems with a large number of variables is based on the use of genetic algorithms (GA) [41]. GA represents a class of stochastic search strategies modelled after evolutionary mechanisms. These population-based algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators (such as crossover and mutation) to these structures so as to preserve critical information [42]. GA are often considered as function optimisers and the main advantage in using them is that they have the ability to make a global search and explore the search space using different kinds of recombination operators. This characteristic may be beneficial in developing crane control systems. Exploiting this aspect, for instance, Kimiaghalam et al. addressed the optimal control problem of a gantry crane with no constraints, using GA [43]. In particular, the authors applied a genetic search in an effort to solve the most difficult part of the trajectory of the crane load. The results presented by the authors are promising. They achieved a noticeable time saving and very clean trajectories. Later on, Ajmal Deen Ali et al. used GA for automated path planning of cooperative crane manipulators [44]. The inverse kinematic problem, i.e. determining the joint angle configuration for the cooperative crane manipulator system in moving the object from pick location to place location, was defined as an optimisation problem and solved using GA. For generating the collision-free path, GA with an interference detection algorithm was employed and the search was made in the manipulator joint angle space (configuration space). The results from this approach are found to be slightly better than the conventional A* search algorithm [45], both in the distance travelled and in computation time. More recently, our research group presented a control algorithm based on GA to exploit the possibilities of controlling offshore cranes [3]. The proposed method allows for controlling different maritime cranes and, more generally, robotic arms regardless of their differences in size, kinematic structure, degrees of freedom, body morphology, constraints, and affordances. This method is described more in detail later in this section.

Another modern heuristic technique used in the field of AI for optimisation problems is known as particle swarm optimisation (PSO) [46]. PSO is a population-based stochastic optimisation technique. It shares many similarities with GA. The system is initialised with a population of random solutions and searches for optima by updating generations. The main difference of this method with respect to GA is that PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, explore the problem space by following the current optimum particles. The main advantages of PSO over GA are that PSO is relatively easier to implement and has few parameters to adjust. Also, the momentum effects on particle movement can allow faster convergence (i.e. when a particle is moving in the direction of a gradient) and more variety/diversity in search trajectories [47]. PSO has been successfully applied in many areas. Regarding maritime cranes, our research group recently presented an alternative control method based on PSO [4]. This method was implemented and simulated with the crane simulator developed by the Offshore Simulator Centre AS (OSC) [48]. This method is described more in detail later in this section.

Optimisation methods based on AI can be used to solve challenging non-linear optimal control problems. However, these methods (as well as other existing control methods) have not been objectively compared, to the best of our knowledge, because of the lack of a universally recognised benchmarking framework.

In this section, two alternative control methods that our research group previously developed are considered as case studies for benchmarking and comparison. The first method is based on the use of GA, while the second method involves the use of PSO. This comparison choice is motivated by the

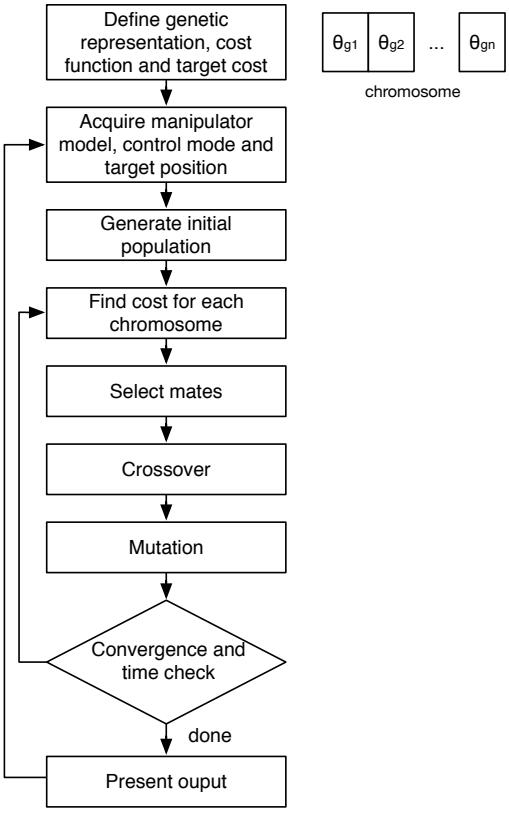


Fig. 7: Flow chart of the proposed mapping method based on GA.

fact that PSO is similar to GA in the sense that they are both population-based search approaches and that they both depend on information sharing among their population members to enhance their search processes using a combination of deterministic and probabilistic rules. The comparison results between the two selected methods are presented in Section VI. Note that even though the two proposed control methods are based on a kinematic approach, the response from the controlled crane model takes into account the system dynamics. In particular, the use of PID controllers at the joints level is adopted to control the system dynamics.

B. Case study 1: a control method based on GA

A crane control method based on the use of GA that was previously developed by our research group is briefly summarised in this section. For further details, the reader is referred to [3]. In order to automatically approximate the mapping equations for manipulator control, (1) and (2), a continuous GA was employed. The forward kinematic model is all that this approach requires. The setup of the proposed algorithm is the same, independently of both the manipulator model being controlled and the selected control mode (position or velocity). Furthermore, when selecting the control mode and controlling the manipulator, the same GA instance is used; the only difference lies in the semantics and the size of inputs and outputs.

The proposed algorithm's flow chart is shown in Fig. 7. The procedure is iterative, and at each iteration, a particular cost function is used to move a population of candidate solutions or chromosomes toward better solutions. The key steps of the algorithm are described below.

1) Define the genetic representation, cost function and target cost: initially, the genetic representation, the cost function, and the target cost are defined. In particular, each chromosome encodes its own properties or genes that consist of a set of joint angles, θ_g, constrained within their corresponding limits. The length of each chromosome is equal to the number of joints to be controlled.

A cost function similar to (3) assesses the Euclidean distance between the target position, $\mathbf{x}_t$, and the calculated position, $\mathbf{x}_c$. This distance is used to evaluate the fitness of every individual in the population. In addition, a target cost is also set to 0.01.

2) Acquire manipulator model, control mode, and target position: the proper manipulator model and control mode are acquired according to the operation scenario by the main iteration loop. Moreover, the target position, $\mathbf{x}_t$, is normalised according to the workspace of the manipulator to be controlled. This effectively maps the cost function to the corresponding workspace.

3) Generate the initial population: subsequently, an initial population is randomly generated, consisting of 50 individuals for each DOF of the model to be controlled. This choice is empirically determined.

4) Find cost for each chromosome: the main iteration loop nests the evolution process sub-loop. This process starts with the initial, randomly generated population, from step 2, and evaluates the fitness of each chromosome at each generation. The fitness is assessed according to the cost function (3). In addition, if there are any genes that violate the corresponding joint limits, their cost is considerably increased. The modification process for each individual's genome then starts in order to form a new generation.

5) Select mates: the *stochastic universal sampling method* [49], which is a fitness proportionate selection method, is used to select candidates that will be used as parents in the crossover process.

6) Crossover: the crossover function is defined as a single-point and uniform crossover method with a 50% crossover probability. This is used to create new offspring from the selected parent chromosomes.

7) Mutation: mutation is taken into account with the random stochastic addition of $\pm 5\%$ to the value of a chromosome's genes. In particular, the chance of mutation is 20% for each gene. A form of elitism is also used, such that 20% of the fittest chromosomes survive unaltered from one generation to the next. Note that since the operator executes continuous movements during manipulator operation, this form of elitism for survival between sequential iterations and consecutive target positions makes sense: consecutive input vectors do not differ much stochastically in terms of magnitude and direction.

8) Convergence and time check: the evolution process is repeated until a termination condition is reached. In particular, the fittest chromosome is returned when the cost drops below the predefined target cost of 0.01, or when the overall population evolution time exceeds 20 ms. At this point the GA stops evolving. Note that the normalisation of the target position according to the workspace of the manipulator relates the cost function to the specific manipulator. As such, a correlation will be present between the target cost and the considered model. The predefined target cost is weighted and proportionate to each specific workspace. For the considered case study, a 20 ms time limit allows for an acceptable level of evolution in the first few iterations, without affecting the operator's perception. For a small number of DOF, the time limit is seldom reached stochastically for target positions located inside the workspace boundaries after the first few iterations.

9) Present output: the fittest chromosome's genes are then presented as output. In particular, denoting these genes as θ_f and according to the operation scenario, the output is obtained as:

$$\theta_d = \theta_f, \tag{13}$$

when operating in position control mode, or as:

$$\dot{\theta}_d = \frac{\theta_f - \theta_a}{\Delta t}, \tag{14}$$

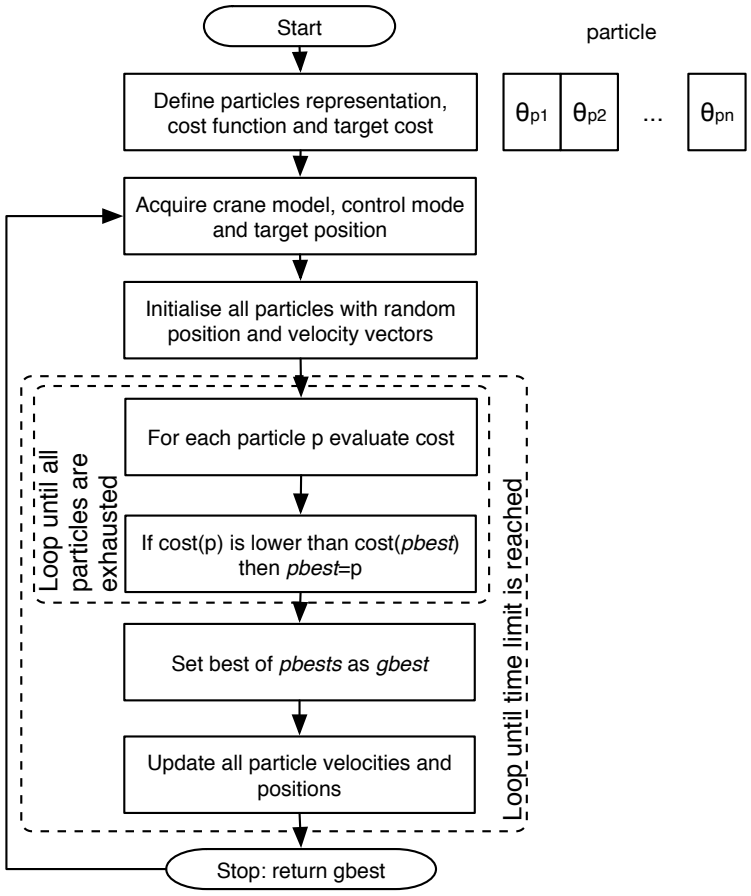


Fig. 8: The proposed PSO-based control method's flow chart.

when operating in velocity control mode.

C. Case study 2: a control method based on PSO

Another alternative crane control method based on the use of PSO was previously developed by our research group. This section briefly summarises the method. For further information, the reader is referred to [4]. In particular, the system automatically approximates (2), the mapping equation. The same instance of PSO is continuously used during control of the manipulator according to the desired control mode. The differences are the semantics and the size of the inputs and the outputs, which the system dynamically and automatically sets. Furthermore, an adaptive swarm size is used. PSO is a stochastic global optimisation method that iteratively attempts to improve a candidate solution in terms of a given measure of quality (fitness or cost) [50]. A population of candidate solutions, known as particles, is used to solve the problem. The particles are moved around the search space according to simple mathematical rules that define the particle's position and velocity. A particle's local best-known position influences its movement, but it is also guided toward the best-known positions in the search space. These positions are updated as other particles find better positions. The expected outcome is that the swarm will move toward the best solutions. In our particular case study, the flow chart for the algorithm is shown in Fig. 8. In the following, the algorithm's key steps are described.

1) Define particles representation, cost function, and target cost: each particle consists of a set of joint angles, θ_p, which are kept within their corresponding limits. The size of each particle is equal to the number of joints to be controlled. A composed cost function that assesses the Euclidean distance between the target position, $\mathbf{x}_t$, and the calculated position, $\mathbf{x}_c$, similarly to (6), is used to evaluate the quality of every particle in the population. For the considered case study, the parameters α and β are equally weighted so that $\alpha = \beta = \frac{1}{2}$. This heuristic choice allows for equally weighting the position error for both the Cartesian space and the joint space. In addition, the target cost is set to 0.01.

2) Acquire manipulator model, control mode, and target position: the main iteration loop acquires the correct manipulator model and control mode according to the operation scenario. According to the manipulator's workspace, the corresponding target position is normalised. This makes it possible to establish a correlation between the cost function and the corresponding workspace.

3) Initialise all particles: an initial population of particles is randomly generated. The population size is empirically defined as 10 times the number of DOF present in the model to be controlled. All particles are initialised with random position vectors.

4) Evaluate the cost for each particle: the optimisation process starts with the initial randomly initialised population. This is a sub-loop of the main iteration and at each step, the defined cost function is used to evaluate the cost of every particle. Each particle tracks its coordinates associated with the optimal solution achieved thus far in the solution space. This value is called personal best (*pbest*). The global best (*gbest*) is another value that is tracked by the PSO. It is the best value obtained thus far by any particle in its respective neighbourhood. Following this, the modification process of each particle's velocity and position begins.

5) Update particle velocities and positions: according to [51], each particle's velocity at iteration $k+1$ can be modelled according to:

$$\mathbf{v}(k+1) = w\mathbf{v}(k) + c_1\mathbf{R}_1(\mathbf{pbest} - \mathbf{s}(k)) + c_2\mathbf{R}_2(\mathbf{gbest} - \mathbf{s}(k)), \tag{15}$$

where w is an inertia weight that weighs the contribution of the previous velocity to control the particle's momentum, $\mathbf{v}(k)$ is the velocity of the particle at the iteration k, $\mathbf{s}(k)$ is the current searching point, $\mathbf{R}_1$ and $\mathbf{R}_2$ are vectors that are the same size of the swarm population and contain random numbers in the range of $[0,1]$, c_1 is a learning factor that determines the importance of *pbest*, and c_2 is a learning factor that determines the importance of *gbest*. c_1 and c_2 are chosen empirically but they may also be selected through a higher-level optimisation process.

Following this, modification of each particle's next position is obtained by adding the position at iteration k to the distance the particle will travel with the new velocity $\mathbf{v}(k+1)$:

$$\mathbf{s}(k+1) = \mathbf{s}(k) + \mathbf{v}(k+1). \tag{16}$$

6) Convergence and time check: the optimisation process is repeated until a termination condition is satisfied. In particular, when the cost drops below 0.01, or when the overall time spent updating the population exceeds 20 ms, the PSO stops updating and the global best particle is returned. Note that since the target position is normalised according to the workspace of the manipulator to be controlled, a correlation will exist between the target cost and the model considered, as the cost function is related to the manipulator. In this way, the predefined target cost is weighted and proportionate to each specific workspace. Given the 20 ms population updating time limit, the population is able to reach an acceptable level of fitness in the first few iterations without affecting the operator experience. After the first few iterations, the time limit is rarely reached for target positions located within the workspace boundaries.

7) Present Output: according to the operational scenario, the output is obtained by:

$$\theta_d = \theta_{pbest}, \tag{17}$$

when operating in position control mode, or with:

$$\dot{\theta}_d = \frac{\theta_{pbest} - \theta_a}{\Delta t}, \tag{18}$$

when operating in velocity control mode.

VI. Simulation results

In this section, the two presented control methods are considered for an extensive comparison. This comparison is performed in simulation by using the proposed benchmarking framework. Each considered control method is used to control the same crane model that was introduced in Section III. The lengths of the crane links are as follows: $L_1 = 2.560$ m, $L_2 = 7.010$ m, $L_3 = 3.500$ m. The maximum joint torques are as follows: $\tau_{max_1} = 100000$ Nm, $\tau_{max_2} = 220000$ Nm, $\tau_{max_3} = 70000$ Nm. The maximum joint velocities are as follows: $\dot{\theta}_{1max} = 0.88$ rad/s, $\dot{\theta}_{2max} = 1.23$ rad/s, $\dot{\theta}_{3max} = 1.29$ rad/s. The weight of the considered links are as follows: $w_{L_1} = 1500$ kg, $w_{L_2} = 2200$ kg, $w_{L_3} = 1300$ kg. As a simplification, the center of gravity is considered to be at the center of each link. The weight of the actuators is not considered. A 500 kg payload is considered. In order to consistently assess the performance of the proposed methods, a common OP is defined so that the same test section can be run against the two methods to be compared. In particular, the *customer type* is a small fishing company, which may use the crane to lift relatively small payloads. The *user type* is a Class 1 Operator with no restrictions or limitations. The adopted *system-mode* is position control. The *function* of lifting a payload is considered. The following sequence of *operations* is taken into account: raising the crane boom, rotating the crane base, and lowering the crane boom. This sequence resembles the most common operations that are executed with the crane in order to handle and transfer objects from large container ships to smaller lighters or to the quays of harbours. The input signal of this particular OP is generated by our framework by simply defining the corresponding path for the crane. During the generation of this input signal, the classical Jacobian method [37] is used to control the crane model and to generate the corresponding input samples. The input samples are stored as a temporal sequence. This sequence contains only input samples that produce reachable points in the crane's workspace.

A. Accuracy

The accuracy of the two proposed control methods is first analysed from a static point of view by considering the position error, as seen in (3). The joint error is not considered since this specific crane model is not a redundant manipulator. For each of the methods to be compared, trajectory tracking analyses of the Cartesian paths for X, Y, and Z coordinates are performed. The results are shown in Fig. 9 and in Fig. 10 for the control method based on GA and the control method based on PSO, respectively. Each time plot shows the actual, the desired, and the calculated coordinates. It should be noted that the actual coordinates are obtained after the PID regulation process. This means that the dynamics of the crane are considered. These plots are qualitatively very similar, showing the effectiveness of the two considered methods. Since the crane dynamics are considered, the torque limitation of the hydraulic actuators affects the trajectory tracking performance. This is noticeable when looking at the discrepancy between the calculated position and the actual position, as shown in Fig. 9 and in Fig. 10.

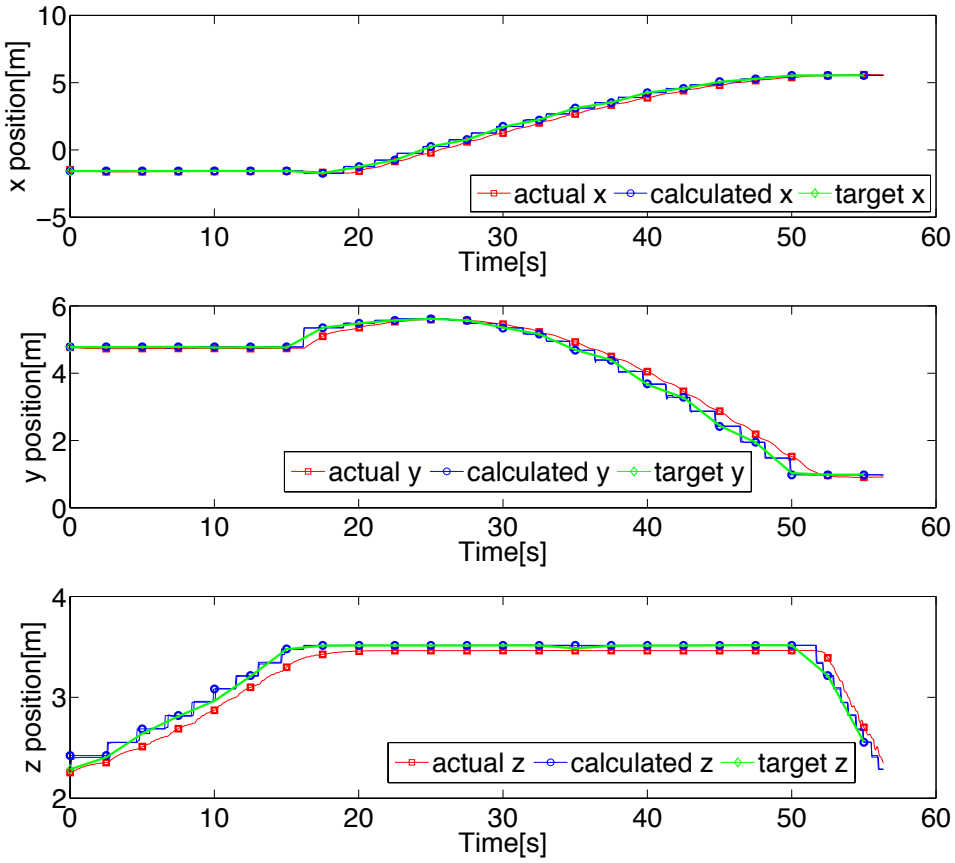


Fig. 9: Trajectory tracking analysis of the Cartesian paths for X, Y, and Z coordinates while using the control method based on GA.

The calculated position is obtained before the PID regulation process, while the actual position is obtained as a consequence of the dynamics generated by the crane hydraulic actuators.

In addition, a time plot of the position error is shown for the two considered methods in Fig. 11 and in Fig. 12 for the control method based on GA and the control method based on PSO, respectively. The control method based on PSO generally shows smaller position errors over time with smaller spikes.

Regarding the total transfer time to target point, the two compared methods are very similar.

B. Effectiveness

To assess the effectiveness of the two alternative control methods, the torque of each joint of the crane is monitored over time. The results are shown in Fig. 13 and in Fig. 14 for the control method based on GA and the control method based on PSO, respectively. It should be noted that the total execution time of the considered OP is similar for both the considered methods. Qualitatively, these time plots look very similar.

Moreover, the total joint effort for the considered OP is calculated by using equation (7). The results are shown in Table I for the two considered approaches. Practically, not very big differences between the two methods can be seen. However, the control method based on GA requires less torque for the joints over time for the considered OP. It is logical to suppose that bigger differences may be identified when comparing redundant manipulators.

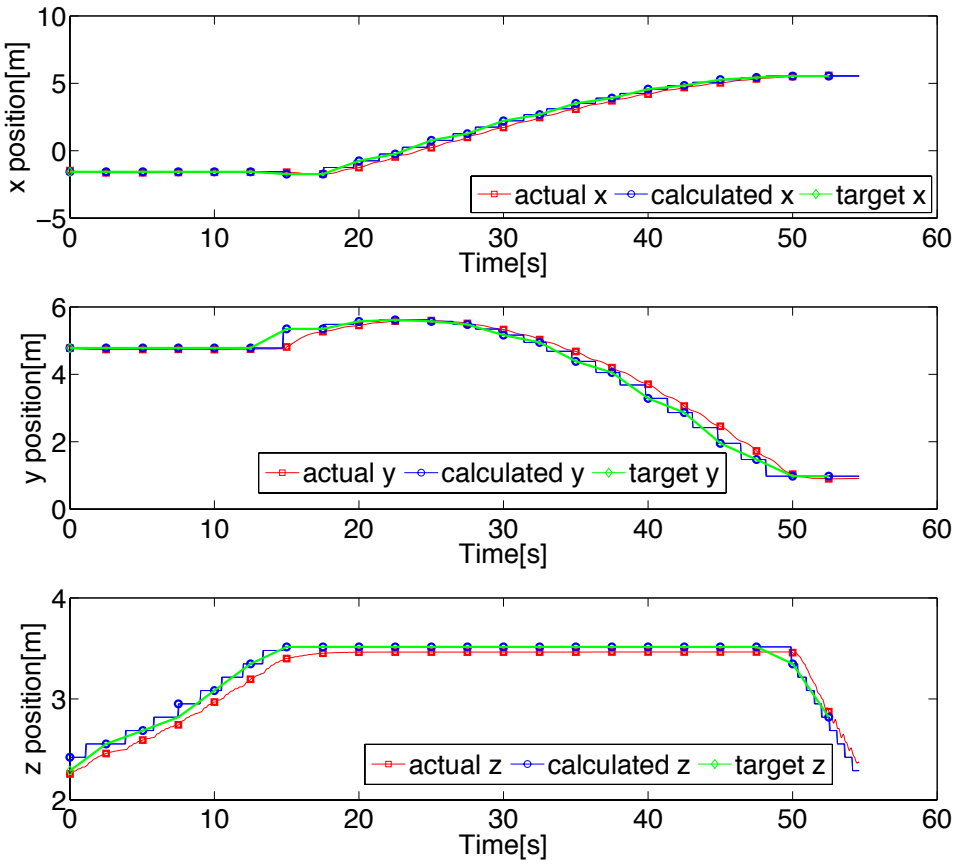


Fig. 10: Trajectory tracking analysis of the Cartesian paths for X, Y, and Z coordinates while using the control method based on PSO.

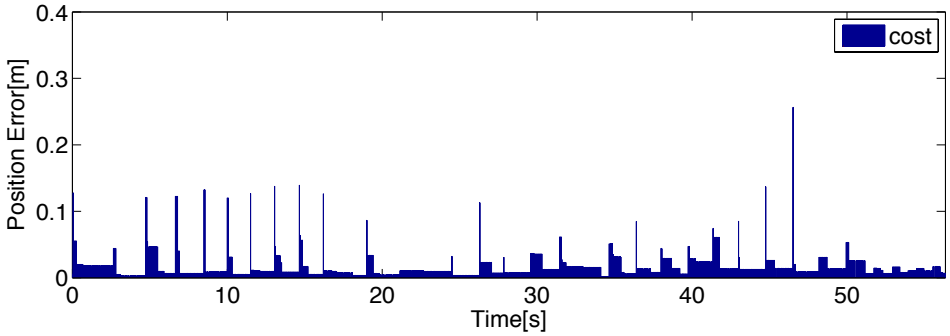


Fig. 11: The time plot of the position error when using the control method based on GA.

C. Performances

To estimate the performance of the two considered alternative control methods, a heave compensation test is run. In particular, the same initial conditions are considered concerning the wave parameters, and the previously adopted OP is used as input for the framework. A trajectory tracking analysis for the crane's end-effector position is performed for each considered control method showing the actual, the desired, and the calculated coordinates. Two cases are analysed: in the first case, no heave compensation

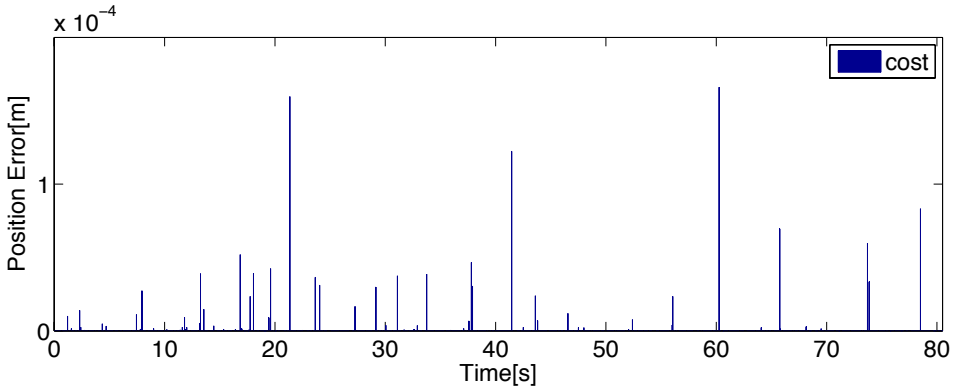


Fig. 12: The time plot of the position error when using the control method based on PSO.

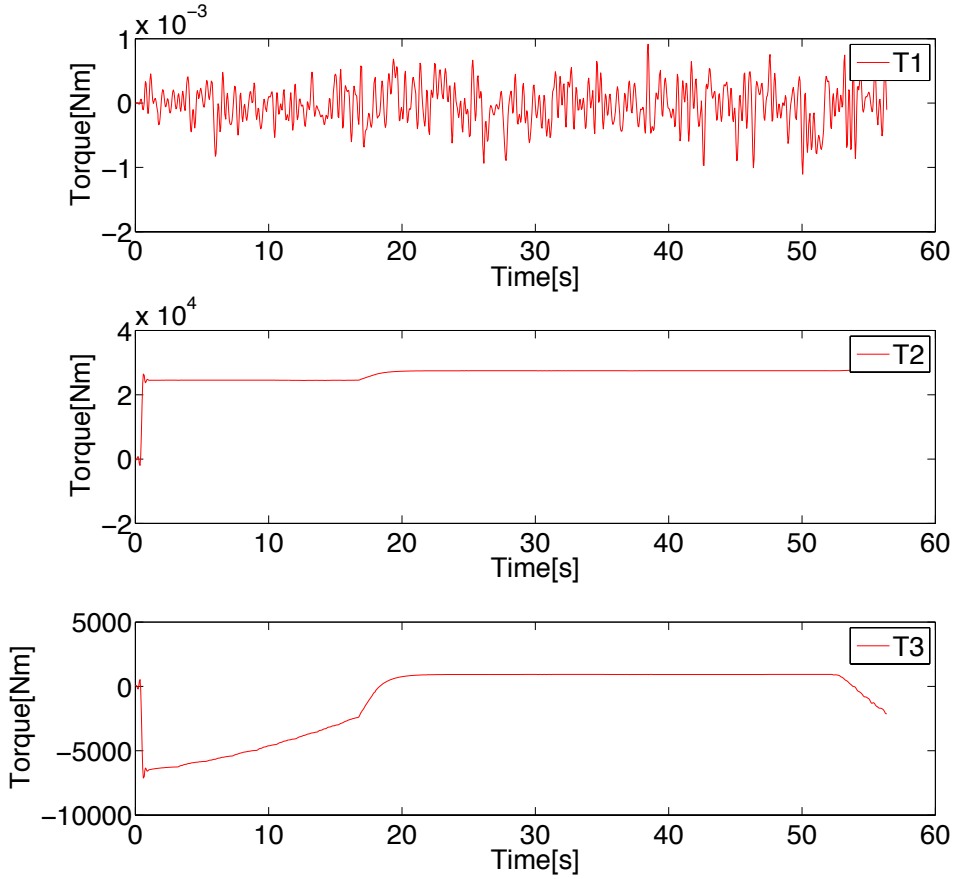


Fig. 13: Torque time plot while using the control method based on GA.

is applied; in the second case, the presented heave compensation method is adopted. The results are shown in Fig. 15 and in Fig. 16 for the control method based on GA and for the control method based on PSO, respectively. Basically, the two considered methods show similar performances in terms of heave compensation. It should be noted that the calculated position of the crane's end-effector is efficiently compensated, while the actual position is closer to the target position when applying the heave

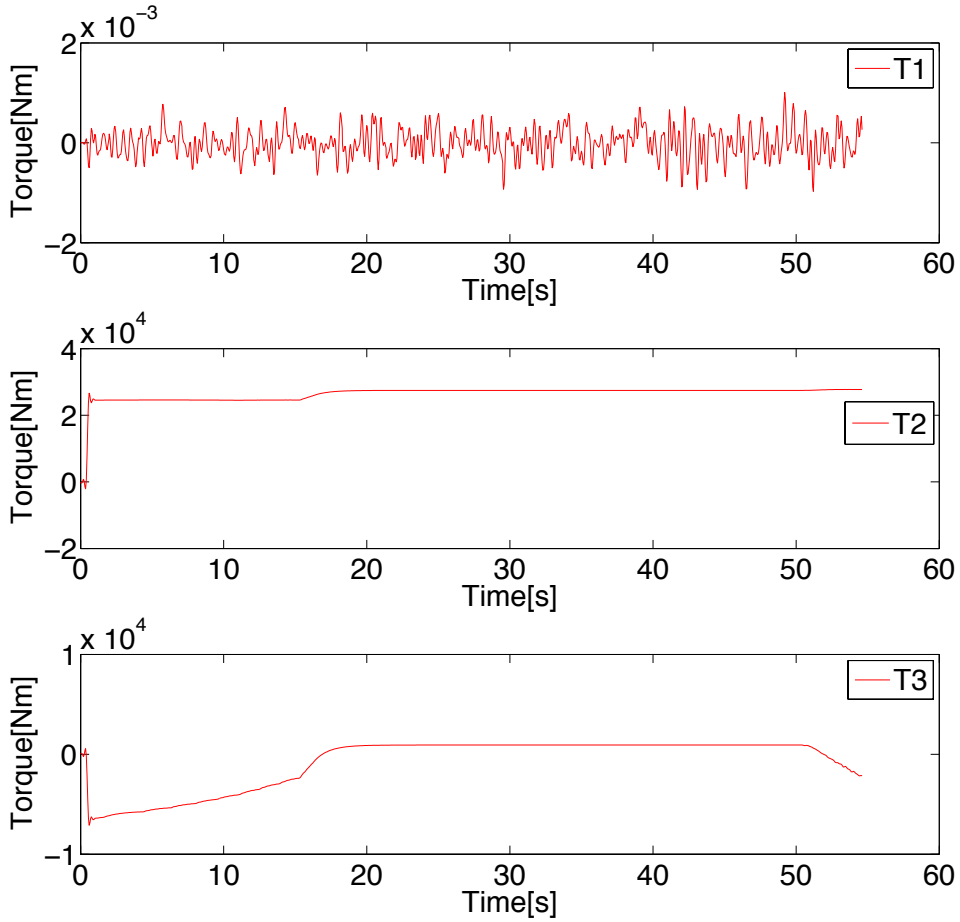


Fig. 14: Torque time plot while using the control method based on PSO.

TABLE I: Joint efforts calculated with equation (7)

Control method	T_1[Nm]	T_2[Nm]	T_3[Nm]
Based on GA	0	7.9078E+12	0.0878E+12
Based on PSO	0	7.6934E+12	0.0773E+12

compensation approach for both the considered methods.

D. Computational time

In general, computational time depends on the DOF of the simulated crane model. In this work, a specific 3 DOF knuckle boom crane model has been considered to perform a preliminary assessment of the two proposed control methods. Therefore, the possibility of scaling this method to crane models with a larger number of DOF need to be further investigated.

Concerning the presented case studies, the same time limit of 20 ms was selected for both the compared control methods. With the considered time limit, the control method based on PSO generally shows smaller position errors over time with smaller error spikes when compared to the method based on GA. The interpretation of these results is that the computational effort required by PSO to converge to a solution is less than that of GA using the same convergence criteria. This result is in line with the

findings presented in [52]. According to the same findings, it appears that PSO outperforms GA with a larger differential in computational efficiency when used to solve unconstrained non-linear problems with continuous design variables and less efficiency differential when applied to constrained non-linear problems with continuous or discrete design variables.

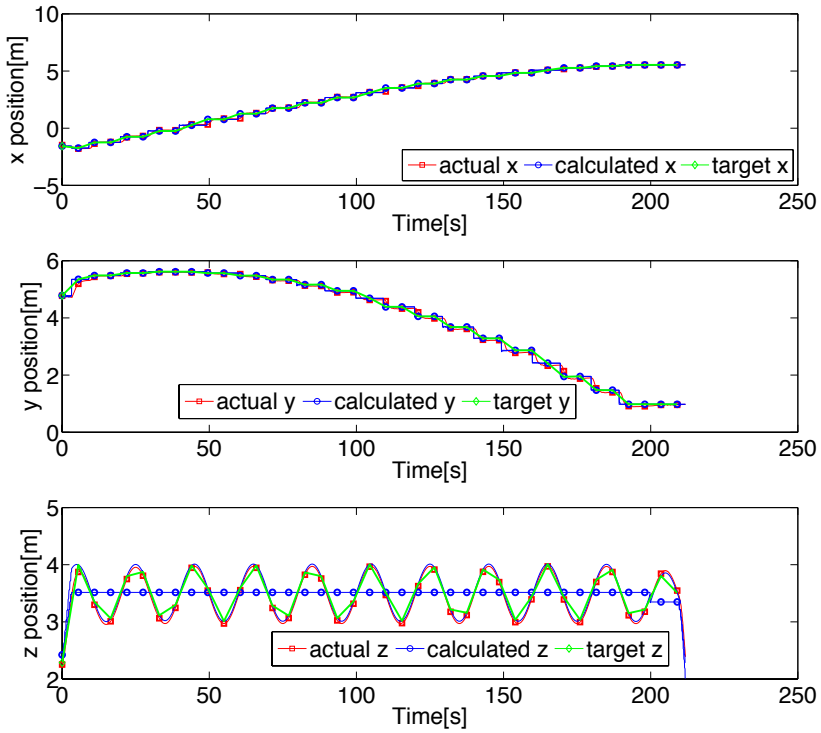
VII. Conclusions and future work

A benchmark framework for advanced control methods of maritime cranes was presented in this work. Based on the concept of interchangeable interfaces, this framework allows for modelling different demanding operation scenarios, including crane models with their corresponding hydraulic systems, vessels and surrounding environment. Different control methods can be transparently implemented and benchmarked by using a set of routine tests, different cost functions, and metrics. Several factors can be considered, including position accuracy, energy consumption, quality, and safety for both the cranes and the surrounding environment. Each proposed routine test is task-oriented, and systematically replicates realistic on-board operation scenarios. In this regard, the concept of operation profiles was proposed, allowing for defining different standard transporting and lifting operations. To show the potential of the proposed framework, two alternative control methods for maritime cranes and robots were considered for an extensive comparison in a simulated environment. The first method is based on the use of genetic algorithms (GA). The second method involves the use of particle swarm optimisation (PSO).

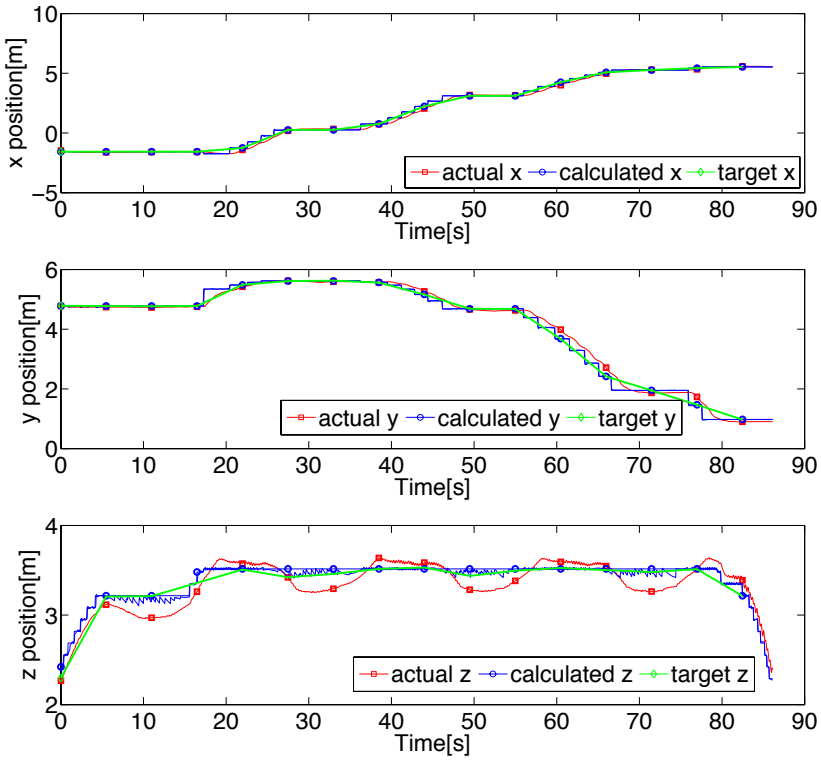
According to the obtained results, the two considered control methods show similar performances when controlling the adopted crane model. The method based on PSO showed slightly better results in terms of position error. Accordingly, from an energy point of view, the method based on PSO generally required slightly less torque for the joints over time. It is quite logical to suppose that bigger differences may be identified when comparing redundant manipulators.

It should be noted that the considered control methods take into account only the heave compensation problem while the sway angle suppression for the rope is not taken into account. All problems related to rope pendulation or wave impact on the payload are not considered in this work but can be taken into account at a later stage. In particular, the generality of the proposed control methods combined with the use of PID controllers may allow for considering the problem of sway angle suppression at a later stage. This is identified as future work.

In the future, additional control algorithms can be implemented and tested [53]. The possibility of comparing the proposed methods with other methods, such as PID and non-linear control approaches, is considered as future work. Additional crane models with a larger number of DOF and more accurate operational profiles can be also added. In this perspective, it is necessary to verify whether the computational time is acceptable even in the case of a crane with redundant DOF. Moreover, new routine tests, different cost functions, and metrics can also be implemented. Methods for computing and visualising the crane workspace may be integrated [54]. Techniques for developing operational procedures that can be used to improve situational awareness (SA) may be also be exploited [55]. Finally, some effort should be put into the standardisation of the integration layer and the communication protocol. An open-source version of the proposed framework could be publicly released to allow researchers to rapidly test new control methods and different models. These improvements will make the proposed framework even more reliable for both industrial and academic practice. It is the opinion of the authors that the key to maximising long-term, macroeconomic benefits for the maritime industry and for academic research relies on the closely integrated development of open content, open standards, and open sources.

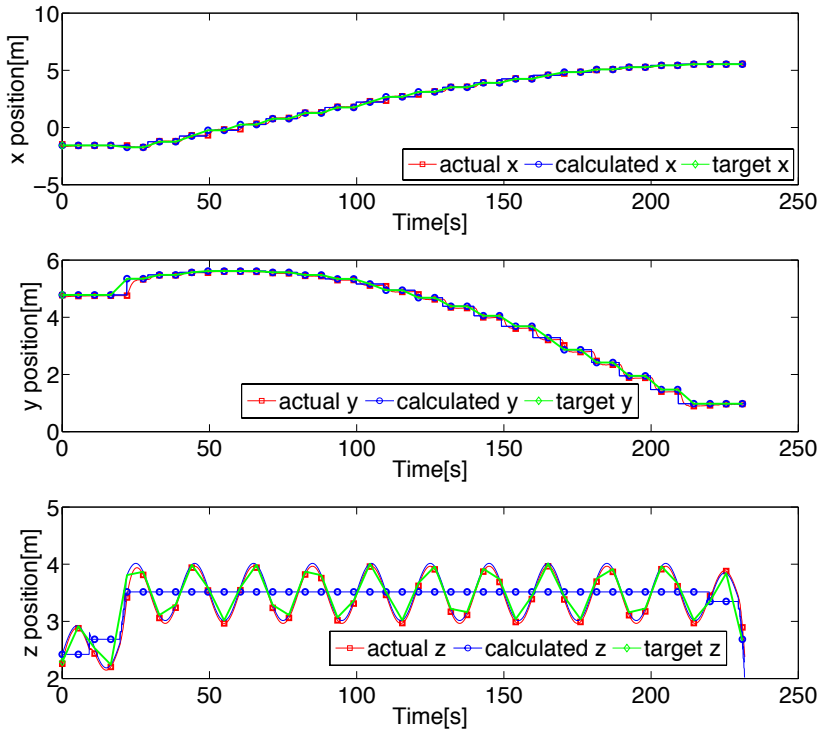


(a)

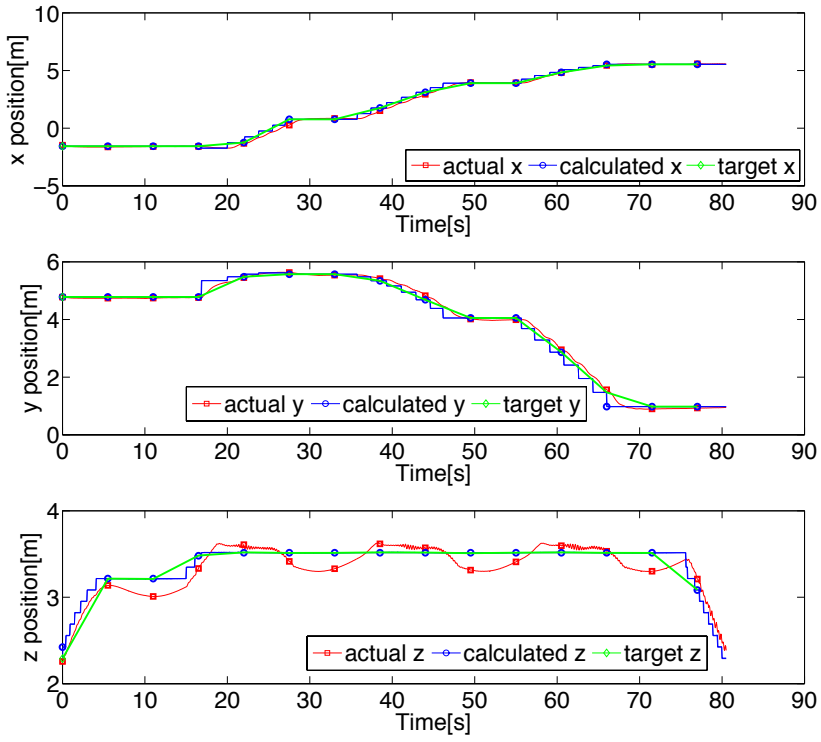


(b)

Fig. 15: (a) The wave effect on the crane's end-effector without heave compensation and (b) with heave compensation when using the control method based on GA.



(a)



(b)

Fig. 16: (a) The wave effect on the crane's end-effector without heave compensation and (b) with heave compensation when using the control method based on PSO.

REFERENCES

[1] A. P. del Pobil, *Benchmarks in Robotics Research*, ser. Lecture Notes for the Workshop of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006.

[2] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold *et al.*, "The Functional Mockup Interface for tool independent exchange of simulation models," in *Proc. of the 8th International Modelica Conference, Dresden*, 2011, pp. 20–22.

[3] 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.

[4] 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," *IEEE Journal of Oceanic Engineering*, vol. 41, no. 2, pp. 450–461, 2016.

[5] Euron - European Robotics Search Network. (2014, September) "Survey and Inventory of Current Efforts in Comparative Robotics Research". [Online]. Available: http://www.robot.uji.es/EURON/en/index.htm

[6] K. Mahelona, J. Glickman, A. Epstein, Z. Brock, M. Siripong, and K. Moyer, "Darpa grand challenge," 2007.

[7] T. Wisspeintner, T. Van Der Zant, L. Iocchi, and S. Schiffer, "Robocup@ home: Scientific competition and benchmarking for domestic service robots," *Interaction Studies*, vol. 10, no. 3, pp. 392–426, 2009.

[8] S. Ulbrich, D. Kappler, T. Asfour, N. Vahrenkamp, A. Bierbaum, M. Przybylski, and R. Dillmann, "The opengrasp benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1761–1767.

[9] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 189–208, 2008.

[10] D. Calisi, L. Iocchi, and D. Nardi, "A unified benchmark framework for autonomous mobile robots and vehicles motion algorithms (movema benchmarks)," in *Proc. of the Workshop on experimental methodology and benchmarking in robotics research (RSS 2008)*, 2008.

[11] S. Moberg and J. Öhr, "Robust control of a flexible manipulator arm: A benchmark problem," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 137–142, 2005.

[12] S. Moberg, J. Ohr, and S. Gunnarsson, "A benchmark problem for robust feedback control of a flexible manipulator," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 6, pp. 1398–1405, 2009.

[13] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363–369.

[14] Modelica Association Project. (2014, November) "Functional Mock-up Interface". [Online]. Available: http://www.fmi-standard.org/

[15] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol–http/1.1," Tech. Rep., 1999.

[16] I. Fette and A. Melnikov, "The websocket protocol (2011)," *Dostupné na: http://tools. ietf. org/html/rfc6455*, vol. 32, 2016.

[17] Mozilla Foundation. (2014, November) "WebGL". [Online]. Available: http://get.webgl.org/

[18] 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.

[19] P. M. La Hera and D. Ortiz Morales, "Modeling dynamics of an electro-hydraulic servo actuated manipulator: A case study of a forestry forwarder crane," in *Proc. of the IEEE World Automation Congress (WAC)*, 2012, pp. 1–6.

[20] L. Guang, X. Bin, Z. Tiesheng, and Y. Jidong, "Research on dynamic modeling and simulation of complex mechanical-electrical-hydraulic coupling system," in *Proc. of the IEEE Spring Congress on Engineering and Technology (S-CET)*, 2012, pp. 1–4.

[21] D. Karnopp, D. Margolis, and R. Rosenberg, *System dynamics: modeling and simulation of mechatronic systems*. John Wiley & Sons New Jersey, 2006, vol. 3.

[22] 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 European Conference on Modelling and Simulation (ECMS), Aalesund, Norway*, 2013, pp. 235–242.

[23] J. Amerongen, "Modeling, simulation and controller design for mechatronic systems with 20-sim 3.0," in *Proc. of the 1st IFAC conference on Mechatronic Systems, Darmstadt, Germany*, vol. 970, 2000.

[24] Y. Chu, F. Sanfilippo, V. Asoy, and H. Zhang, "An effective heave compensation and anti-sway control approach for offshore hydraulic crane operations," in *Proc. of the IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China*, 2014, pp. 1282–1287.

[25] Y. Chu, V. Æsøy, H. Zhang, and Ø. Bunes, "Modeling and simulation of an offshore hydraulic crane," in *Proc. of the 28th European Conference on Modelling and Simulation (ECMS), Brescia, Italy*, 2014, pp. 87–93.

[26] J. E. Bobrow, B. Martin, G. Sohl, E. Wang, F. C. Park, and J. Kim, "Optimal robot motions for physical criteria," *Journal of Robotic systems*, vol. 18, no. 12, pp. 785–795, 2001.

[27] Q. H. Ngo and K.-S. Hong, "Sliding-mode antisway control of an offshore container crane," *IEEE/ASME transactions on mechatronics*, vol. 17, no. 2, pp. 201–209, 2012.

[28] S. Ohtomo and T. Murakami, "Estimation method for sway angle of payload with reaction force observer," in *Proc. of the IEEE 13th International Workshop on Advanced Motion Control (AMC)*, 2014, pp. 581–585.

[29] H. Honda and T. Yoshida, "Estimating the sway angle of an overhead crane by an observer using angular velocity sensors," *Nippon Kikai Gakkai Ronbunshu C Hen(Transactions of the Japan Society of Mechanical Engineers Part C)(Japan)*, vol. 18, no. 10, pp. 3201–3206, 2006.

[30] H. Sano, K. Sato, K. Ohishi, and T. Miyazaki, "Robust design of vibration suppression control system for crane using sway angle observer considering friction disturbance," *Electrical Engineering in Japan*, vol. 184, no. 3, pp. 36–46, 2013.

[31] S. B. van Albada, G. D. van Albada, H. P. Hildre, and H. Zhang, "A novel approach to anti-sway control for marine shipboard cranes." in *Proc. of the 27th European Conference on Modelling and Simulation (ECMS), Aalesund, Norway*, 2013, pp. 249–256.

[32] Y. Shen, K. Terashima, and K. Yano, "Minimum time control using straight transfer for a rotary crane," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 43–48, 2005.

[33] K. Deb, K. Sindhya, and J. Hakanen, "Multi-objective optimization," in *Decision Sciences: Theory and Practice*. CRC Press, 2016, pp. 145–184.

[34] B. Kitchenham, S. L. Pfleeger, and N. Fenton, "Towards a framework for software measurement validation," *IEEE Transactions on software Engineering*, vol. 21, no. 12, pp. 929–944, 1995.

[35] G. Sarker, G. Myers, T. Williams, D. Goldberg *et al.*, "Comparison of heave-motion compensation systems on scientific ocean drilling ship and their effects on wireline logging data," in *Proc. of the Offshore technology conference*, 2006.

[36] 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*, 2008, pp. 538–543.

[37] B. Siciliano and O. Khatib, *Handbook of robotics*. Springer, 2008.

[38] M. R. Lyu *et al.*, *Handbook of software reliability engineering*. IEEE computer society press CA, 1996, vol. 222.

[39] American Petroleum Institute, *Operation and Maintenance of Offshore Cranes, API recommended practice 2D, fifth edition*, 2003.

[40] R. A. Brooks, "Intelligence without representation," *Artificial intelligence*, vol. 47, no. 1, pp. 139–159, 1991.

[41] L. Davis, "Handbook of genetic algorithms," 1991.

[42] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[43] B. Kimiaghalam, A. Homaifar, M. Bikdash, and G. Dozier, "Genetic algorithms solution for unconstrained optimal crane control," in *Proc. of the IEEE Congress on Evolutionary Computation (CEC'99)*, vol. 3, 1999.

[44] M. A. D. Ali, N. R. Babu, and K. Varghese, "Collision free path planning of cooperative crane manipulators using genetic algorithm," *Journal of computing in civil engineering*, vol. 19, no. 2, pp. 182–193, 2005.

[45] P. Sivakumar, K. Varghese, and N. R. Babu, "Automated path planning of cooperative crane lifts using heuristic search," *Journal of computing in civil engineering*, vol. 17, no. 3, pp. 197–207, 2003.

[46] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[47] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *International Conference on Evolutionary Programming*. Springer, 1998, pp. 611–616.

[48] Offshore Simulator Centre AS. (2016, December) "OSC". [Online]. Available: http://www.offsim.no/

[49] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 6, pp. 2193–2196, 2012.

[50] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 760–766.

[51] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proc. of the IEEE Congress on Evolutionary Computation*, vol. 1, 2001, pp. 94–100.

[52] R. Hassan, B. Cohanim, O. De Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, 2005, pp. 1–13.

[53] 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.

[54] L. I. Hatledal, F. Sanfilippo, Y. Chu, and H. Zhang, "A voxel-based numerical method for computing and visualising the workspace of offshore cranes," in *Proc. of the ASME 34th International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2015, pp. 1–7.

[55] F. Sanfilippo, "A multi-sensor fusion framework for improving situational awareness in demanding maritime training," *Reliability Engineering & System Safety*, vol. 161, pp. 12–24, 2017.

Filippo Sanfilippo [1] received a B.Sc. in computer engineering from the University of Catania, Catania, Italy, in 2009 and an M.Sc. in computer engineering from the University of Siena, Siena, Italy, in 2011. In 2008, he was a Visiting Scholar at the School of Computing and Intelligent Systems, University of Ulster, Londonderry, United Kingdom and in 2010 a Visiting Fellow at the Technical Aspects of Multimodal Systems (TAMS) research group, Department of Mathematics, Informatics and Natural Sciences, University of Hamburg, Hamburg, Germany. In 2015, he received a Ph.D. from the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. For his Ph.D. studies, he was awarded a research scholarship from the IEEE Oceanic Engineering Society (OES) scholarship program. He is currently working as a Postdoctoral Fellow at the Department of Engineering Cybernetics, NTNU in Trondheim, Norway. His research interests include control methods, robotics, artificial intelligence, and modular robotic grasping.

Lars Ivar Hatledal received a B.Sc. in Automation from Aalesund University College, Norway, in 2013. He is currently perusing an M.Sc. in Simulation and Visualisation and has been working at the same university as a Project Leader in the Department of Maritime Technology and Operations since 2013. His research interests include robotics, artificial intelligence, and 3D visualisation.

Kristin Ytterstad Pettersen received an M.Sc. and a Ph.D., both in Electrical Engineering, from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 1992 and 1996, respectively. In 1996, she became an Associate Professor, and in 2002 a Professor, in the Department of Engineering Cybernetics, NTNU. In 1999, she was a Visiting Fellow in the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ. In 2008 she was a Visiting Professor in the Section for Automation and Control, University of Aalborg, Denmark. She has published more than 190 scientific papers and three textbooks. Her research interests include non-linear control of mechanical systems with a special emphasis on marine robotics and snake robotics. She is an Associate Editor of the IEEE Transactions on Control Systems Technology and the IEEE Control Systems Magazine. She received the IEEE Transactions on Control Systems Technology outstanding paper award in 2006.

Houxiang Zhang received a Ph.D. in Mechanical and Electronic Engineering in 2003. From 2004, he worked as a Postdoctoral Fellow at the Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, University of Hamburg, Germany. Dr. Zhang joined the Department of Maritime Technology and Operations, Aalesund University College, Norway in April 2011, where he is a Professor in Robotics and Cybernetics.

[1] http://filipposanfilippo.inspitivity.com