A COMPUTER-AUTOMATED DESIGN TOOL FOR INTELLIGENT VIRTUAL PROTOTYPING OF OFFSHORE CRANES

Robin T. Bye*, Ottar L. Osen*, and Birger Skogeng Pedersen†
* Software and Intelligent Control Engineering Laboratory
Faculty of Engineering and Natural Sciences
Aalesund University College
Postboks 1517, NO-6025 Ålesund, Norway
Email: {roby,oo}@hials.no
† ICD Software AS
Hundsværgata 8, NO-6008 Ålesund, Norway
Email: bsp@icd.no

KEYWORDS

Computational Model; Simulator; Genetic Algorithm; Optimisation; Artificial intelligence.

ABSTRACT

In close collaboration with the maritime industry, virtual prototyping with maritime application has been an important research topic for Aalesund University College for some years. In this paper, we describe the development of a computer-automated design tool for intelligent virtual prototyping of offshore cranes. Our work is part of a research project funded by the Research Council of Norway and takes place in close cooperation with two partners from the maritime industry. A literature review of virtual prototyping, computer-automated design, and modelling and simulation of offshore cranes sets the stage for the description of a design tool whose main components consist of a computational model, a simulator, and a genetic algorithm. We show how domain-specific constraints can be accounted for in conjunction with an automated optimisation procedure of design parameters to yield crane specifications that closely match the desired design criteria. Limitations of slewing rings and hydraulic cylinders are of particular importance in offshore crane design and are used as an example of the multitude of design calculations that form the computational model. Being work in progress, we report on completed parts and the work that remains.

INTRODUCTION

20 years ago, Pratt (1995) defined virtual prototyping (VP) as the computer-aided construction of digital product models (usually virtual prototypes or digital mockups) and realistic graphical simulations for the purpose of design and functionality analyses in the early stages of the product development process. Later, in a review on VP, Wang (2002) defined VP as the construction and testing of virtual prototypes, where such prototypes are computer simulations of physical products that can be presented, analysed, and tested from concerned life-cycle aspects such as design and engineering, manufacturing, service and recycling as if on a real physical model.

The use of virtual prototypes and simulation techniques, for example in the shipbuilding (Kim et al., 2002) and automotive (Wöhlke and Schiller, 2005) industries has made a significant contribution to the process of evaluating and improving product design and to the validation of product planning and manufacturing processes (e.g., Mujber et al., 2004; De Sa and Zachmann, 1999; Weyrich and Drews, 1999). However, whereas VP is perhaps mostly associated with the design and development process of concrete commercial products, other uses exist, for example with respect to planning processes in the ship-building industry. As noted by Cha et al. (2010), process planning may be set up based on past experience but problems not expected in advance may, and will, still occur during production, since all ships and offshore structures to be constructed will differ in purpose, shape and size. To cope with these challenges, Cha et al. proposes an integrated simulation framework for the process planning of ships and offshore structures, separating a simulation kernel, and a basic simulation component from the application-specific simulation component.

For example, in the field of construction engineering, construction VP (CVP) can be utilised to facilitate integrated planning and visualisation of large construction projects, thereby assessing the executability of construction plans including site layout, temporary work design, and resource planning (Huang et al., 2007). Since the turn of the millennium, a number of CVP tools have been developed that allow construction teams to practice on major construction processes and examine various execution strategies in realistic virtual enviroments (VEs) before and during the actual construction (Waly and Thabet, 2003), thus enabling client briefing, simulation of properties such as lighting, acoustics, and energy consumption, and 4D (space and time) visualization of the building construction sequence (Sarshar et al., 2004; Yerrapathruni, 2003). Consequently, VP can cause flaws to be corrected early in the design phase, or in some circumstances, prevent flawed projects from even materialising.

Moreover, VP can be used as a tool for collaborative product design, broadly classified into component design and assembly design (Shyamsundar and Gadh, 2002). Even in the early days of public Internet, researchers described how to use the world wide web and VP for collaborative component design, where designers can view and modify component geometry and exchange ideas

* Equal first author contribution. Robin T. Bye is the corresponding author.

Proceedings 29th European Conference on Modelling and Simulation ©ECMS Valeri M. Mladenov, Petia Georgieva, Grisha Spasov, Galidiya Petrova (Editors)
ISBN: 978-0-9932440-0-1 / ISBN: 978-0-9932440-1-8 (CD)

online and in real-time (Chang et al., 1999; Chan et al., 1999). Likewise, for product assembly design, researchers were early to discuss online VEs in which collaborative assembly modeling can be performed using constraints and 3D models marked up and exchanged (Maxfield et al., 1995; Kuttner and Deitz, 1996). In multidisciplinary product development such as mechatronic engineering, cooperative VP is particulary useful, since inefficient communication between the designers and engineers from different domains can become a serious obstruction for accelerating the design of mechatronic products (Shen et al., 2005). Finally, Choi and Cheung (2008) note that employing virtual reality (VR) as a collaborative VP tool can greatly improve the product design, test and review loop before committing to physical fabrication, especially if fully-immersive VR systems such as a cave automatic virtual environment (CAVE) is used.

It is clear that VP encompasses a number of aspects for the design and development of virtual prototypes, including modelling, simulation, visualisation, analysis, testing, validation, optimisation, team cooperation, product presentation, and so on. In this paper, we will focus on computer-automated design (CautoD), combining artificial intelligence (AI) with VP in order to automate and optimise the design phase of offshore cranes.

Computer-Automated Design

The first scientific report of computer-automated product design is perhaps that of Kamentsky and Liu (1963), who created a computer programme able to determine suitable logic circuits satisfying certain hardware constraints while at the same time evaluating the ability of the logics to perform character recognition. Since then, many contributions of CautoD have been made, particularly in the field of structural engineering (see Hare et al., 2013, for a survey). The general paradigm appears to be that of optimisation, where the design problem is formulated as the minimisation of a cost function or maximisation of a fitness function. AI lends itself naturally to solving many complex optimisation problems, in particular by the use of nature-inspired heuristic computational algorithms. For example, based on condensed matter physics, the simulated annealing algorithm was one of the first nature-inspired algorithms used for design optimisation (Kirkpatrick, 1984; Čerǹy, 1985) and is still being used today for a variety of purposes, including CautoD for tensegrity systems (e.g., Xu and Luo, 2010).

Goldberg (1983) used a genetic algorithm (GA) for the design of computer-based control of gas pipeline systems, whereas Rajeev and Krishnamoorthy (1992) used a simple GA for optimizing structural systems, as did Pezeshk et al. (2000) for the design of 2D, geometrical, nonlinear steel-framed structures. More recently, Kaveh and Talatahari (2009) constructed a hybrid algorithm that employed both particle swarm optimisation (PSO) and ant colony optimization (ACO) to find an optimal design of different types of truss structures and frame structures, while Peng et al. (2010) used an adaptive GA for optimising the piping design process of offshore drilling platforms.

Also worth mentioning is a very recent PhD thesis, which analysed and investigated a homogeneous charge microwave ignition system through a simulation-based CautoD framework, in which intelligent bio-inspired optimisation algorithms can interrogate a simulator in search of novel design solutions (Schöning, 2014). Access to the full thesis is restricted by embargo and third-party copyright until September 2017, which underlines the potential financial impact of CautoD and VP in the industry.

Modelling and Simulation of Offshore Cranes

VP is an active focus of research of the newly formed Software and Intelligent Control Engineering (SoftICE) Laboratory at Aalesund University College (AAUC) of which both authors of this paper are members. Indeed, the work we present in this paper is part of one of two nationally funded research projects that the SoftICE lab currently participates in. In addition, for the last few years, our colleagues in the Mechatronics Laboratory at AAUC have published several papers relating to modelling, simulation and visualisation of offshore cranes and installations. (e.g., Sanfilippo, Hildre, Æsøy, Zhang and Pedersen, 2013; Sanfilippo, Hatledal, Schaathun, Pettersen and Zhang, 2013; Sanfilippo et al., 2014; Chu, Sanfilippo, Æsøy and Zhang, 2014; Chu, Æsøy, Zhang and Bunes, 2014; Halse et al., 2014; Hatledal et al., 2015).

As noted by Halse et al. (2014), advanced marine operations such as subsea installations consist of multiple subsystems that must be collectively controlled in a precise manner and typically involves controlling a crane and winch mounted on a vessel operating in a dynamic, uncertain environment affected by wind, waves, currents, shape and size of payload, and more. Modeling and simulation of such interactive multibody systems is a complex task that involves hydrodynamics, mechanics, hydraulics, electronics, and control systems (Halse et al., 2014).

Chu, Æsøy, Zhang and Bunes (2014) used the bond graph method as their approach for modelling offshore hydraulic cranes. This method is suitable for modelling systems of systems, where modules may be removed or added or connected to other systems, and thus the method lends itself naturally to VP. Using the same bond graph methodology, Sanfilippo, Hildre, Æsøy, Zhang and Pedersen (2013) built a modular prototyping system architecture in which a number of different maritime cranes or robotic arms with different kinematic structures and degrees of freedom were modelled and simulated in a VE. Focusing on the control of offshore cranes, Sanfilippo and colleagues have developed control algorithms for effective heave compensation and anti-sway control (Chu, Sanfilippo, Æsøy and Zhang, 2014), as well as using both GAs (Sanfilippo, Hatledal, Schaathun, Pettersen and Zhang, 2013) and artificial neural networks (ANNs) (Sanfilippo et al., 2014) for universal control of multiple cranes with different properties by means of a single unique input haptic device.

In a paper recently accepted for publication and to be presented at the 34th International Conference on Ocean, Offshore and Arctic Engineering in June this year, Hatledal et al. (2015) presents a voxel-based numerical method for computing and visualising the 3D workspace of offshore cranes. Despite the importance of a crane's lifting capacity in different positions in the workspace (often visualised as a 2D load chart), which depend on the properties of crane components such as cylinders, links, sheaves, and joints, workspace and load chart calculations are usually not taken into account in the design phase and are merely realised as an indirect consequence of a priori design choices (Hatledal et al., 2015). However, Hatledal et al. note that employing their algorithm as a trial-and-error VP tool during the preliminary design phase, factors such as the length of crane links and size of cylinders can be designed to yield better workspace characteristics.

Finally, we would like to point to recent work by Bak et al. (2011); Bak and Hansen (2013); Peng et al. (2010); Pawlus et al. (2014), who present detailed analyses on aspects of VP of offshore knuckleboom cranes and pipe handling equipment, including techniques for modelling, simulation, and parameter identification that can aid in the VP process.

Motivation and Aim

The contributions presented above are valuable in the pursuit of VP systems for offshore cranes but provide little insight into how the various models, calculations, simulations, and visualisations can be used for VP, and in particular, CautoD. Hatledal et al. (2015) mentions using trial-and-error to improve the design phase but although their method is a step forward from the traditional experience-based rule-of-thumb approaches employing pen-and-pencil or spreadsheet calculations commonly employed in the maritime industry, it is hardly satisfactory given the large number of design parameters. In short, the literature above provides various means to determine crane properties and behaviour based on pre-determined design parameters, somewhat analogous to calculating the forward kinematics of a robotic arm or offshore crane. However, the problem of determining the inverse kinematics is generally much harder, and continuing the analogy, we are thus faced with the following "inverse" challenge, for which analytical solutions are infeasible: How can we choose appropriate values for numerous, possibly conflicting, offshore crane design parameters such that the resulting cranes have the desired properties and behaviour that we want?

The research we present in this paper tries to answer this question and is part of the project *Artificial Intelligence for Crane Design (Kunstig intelligens for krandesign (KIK))* funded by the Research Council of Norway. The project is a collaboration between the SoftICE lab at AAUC and two industrial partners, ICD Software AS and Seaonics AS. We report on the current status of the project and the planned way forward. A goal of the project is to build an offshore crane simulator able to generate 2D-visualised crane load charts and calculate a number of crane properties on-the-fly based on a particular choice of design parameter values. Another goal is to use a method well-known from AI, namely a GA, to search through the vast number of design choices and combinations until the desired design criteria are satisfied. The aim of the project is to succeed in reaching these goals and develop a working software prototype of a simulator for intelligent CautoD and VP for offshore cranes that can be tested and further developed by offshore crane designers and manufacturers such as our aforementioned partner Seaonics AS.

METHOD

An offshore crane such as the one in Figure 1 is a complex machine. Often equipped with advanced control

Figure 1: The new boomerang-shaped crane of Seaonics AS, which has its wire/rope routed directly from winch to boom tip. This increases the work area compared to standard knuckleboom cranes and reduces wear and tear of the wire or rope. The crane is ideal for arctic operations and fiber rope use. Image courtesy of Seaonics AS.

systems for heave compensation, boom tip positioning, and anti-sway, it has many features in common with articulated robots. Even simple versions of such offshore cranes consist of a large number of components, such as hooks, winches, slewing rings, cylinders, booms, hinges, sheaves, and pedestals (see Figure 2). The placements, types, capacities, materials, and abilities of these components all affect the overall properties of the crane. Various parameters of interest can then be derived from the physical properties of the crane components themselves and their interrelationships.

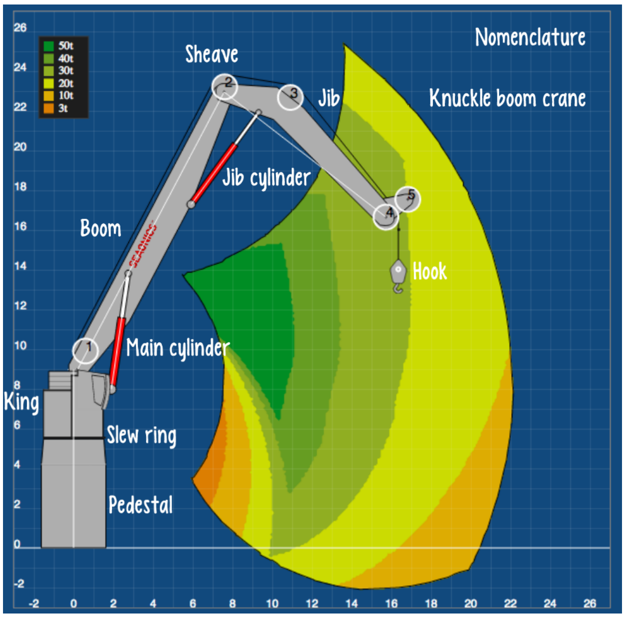

Figure 2: Illustration of the main components of an offshore knuckleboom crane and its 2D load chart. Image courtesy ICD Software AS.

Some of these derived parameters of the crane are of greater interest than others and constitute the main design parameters of the crane, for example key performance indicators (KPIs) such as the desired workspace and the working load limit (WLL) and safe working load (SWL) within that workspace, total weight, control system characteristics, durability, installation and operating costs, and safety concerns such as wind impact. Additionally, laws, regulations, and the use of design codes such as the standards provided by classification socities like DNV-GL (formerly DNV), Lloyd's Register Group Limited (formerly Lloyd's Bureau of Shipping), and the American Bureau of Shipping (ABS) all put constraints on the choice of design parameters.

Computational Model

When making a computational model (CM) of an offshore crane, we have to choose which parameters that need to be modelled since modelling everything is intractable. The CM we have developed has reduced the number of parameters from potentially several thousands to about 120 parameters. It is these 120 parameters that must be chosen by the designer to ensure that requirements by laws, regulations and standards, various KPIs, and other desired design criteria are met.

Example of Design Calculations

A complete overview of the myriad of various components, mathematical models, and design constraints contained in our CM cannot be provided in this paper but the interested reader may refer to aforementioned work on VP and modelling of offshore knuckleboom cranes (Bak et al., 2011; Bak and Hansen, 2013) or offshore pipe handling machines (Pawlus et al., 2014). Instead, we wish to illustrate the complexity of our CM by presenting an example of the design calculations required for a hydraulic cylinder as given by the DNV-GL's "Rules for Certification of Lifting Appliances" (Det Norske Veritas (DNV), 1999). Figure 3 shows a hydraulic cylinder and its key components with a nomenclature used for the design calculations that follows.

Hydraulic cylinders are the key component that provides "muscle power" to offshore cranes. To ensure compliance with DNV-GL standards, the cylinders must be designed to avoid buckling, a phenomenon of mathematical instability leading to failure mode, namely a sudden sideways deformation. Specifically, the buckling load P must be greater than the actual maximum cylinder force F by a factor of at least 2.3 (see Eq. 17).

The buckling calculations assume that stresses in the rod due to axial load, initial deflection, frictional moment in the bearings, and weight of the cylinder do not exceed the yield stress σ_y of the rod material (note that the weight of the cylinder can normally be ignored in the buckling calculations if intended for the design of an ordinary crane):

$$\frac{P}{A} + \left(\frac{P \cdot f_0}{W} + \frac{P \cdot \mu \cdot r}{W} + \frac{m_{\text{Cyl}} \cdot g \cdot L}{W}\right) \cdot \frac{P_E}{P_E - P} \leq \sigma_y \quad (1)$$

Simplified, the form of deflection for the cylinder is assumed to be:

$$y(x) = C_1 \cdot \sin\left(\frac{\pi \cdot x}{L}\right) + C_2 \cdot \sin\left(\frac{2 \cdot \pi \cdot x}{L}\right) \quad (2)$$

To find the acceptable load P, the following equations are used (note that some variables such as f_0, AA, BB, etc. are not given in the nomenclature as they are mainly intermediate auxiliary calculations):

$$\alpha = \frac{\pi \cdot L_2}{L} \quad (3)$$

$$AA = \frac{L_1}{2 \cdot I_1} + \frac{L_2}{2 \cdot I_2} + \frac{L}{4 \cdot \pi} \cdot \left(\frac{1}{I_1} - \frac{1}{I_2}\right) \cdot \sin(2 \cdot \alpha) \quad (4)$$

$$BB = \frac{4 \cdot L}{3 \cdot \pi} \cdot \left(\frac{1}{I_2} - \frac{1}{I_1}\right) \cdot \sin^3(\alpha) \quad (5)$$

$$CC = \frac{L_1}{2 \cdot I_1} + \frac{L_2}{2 \cdot I_2} + \frac{L}{8 \cdot \pi} \cdot \left(\frac{1}{I_1} - \frac{1}{I_2}\right) \cdot \sin(4 \cdot \alpha) \quad (6)$$

$$DD = \frac{\pi^2 \cdot E}{2 \cdot L} \quad (7)$$

$$a = 4 \cdot AA \cdot CC - BB^2 \quad (8)$$

$$b = -4 \cdot DD \cdot (4 \cdot AA + CC) \quad (9)$$

$$c = 16 \cdot DD^2 \quad (10)$$

$$P_E = \frac{-b - \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \quad (11)$$

$$f_0 = \frac{L_1 \cdot \Delta}{L \cdot L_3 \cdot 2} \cdot \sqrt{\frac{\pi^2 \cdot E \cdot I_2}{P_E}} \quad (12)$$

$$FF = \frac{d}{2 \cdot I_2} \cdot (\mu \cdot r + f_0) \quad (13)$$

$$GG = \frac{m_{\text{Cyl}} \cdot g \cdot L \cdot d}{16 \cdot I_2} \quad (14)$$

$$HH = \left[1 + 2 \cdot A \cdot FF - \frac{2 \cdot \sigma_y \cdot A}{P_E} + A^2 \cdot FF^2 + \frac{2 \cdot A^2 \cdot FF \cdot \sigma_y}{P_E} + \left(\frac{\sigma_y \cdot A}{P_E}\right)^2 + \frac{4 \cdot A \cdot GG}{P_E}\right]^{\frac{1}{2}} \quad (15)$$

We can then find P as

$$P = \frac{\sigma_y \cdot A}{2} + \frac{P_E}{2} \cdot (1 + A \cdot FF - HH) \quad (16)$$

Finally, the safety factor P/F against buckling must be at least 2.3 as given by the following formula:

$$\frac{P}{F} \geq 2.3 \quad (17)$$

Simulator

The CM is parameterised and implemented in software as a simulator. The simulator is then able to produce a number of outputs dependent on the parameter values of

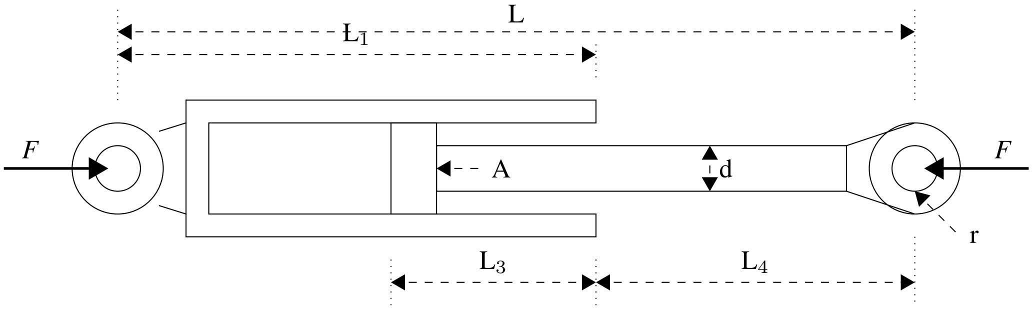

Figure 3: Diagram of hydraulic cylinder (above) and nomenclature for design calculations in Eqs.1-17 (below).

Parameter	Description
A	piston rod cross section area, mm^2
d	piston rod outer diameter, mm
E	modulus of piston rod material elasticity, 2.06 10^5 N/mm^2 for steel
F	actual maximum cylinder force, N
I_1	moment of inertia of cylinder tube cross section, mm^4
I_2	moment of inertia of piston rod cross section, mm^4
L	length of hydraulic cylinder from centre to centre of end eyes, mm
L_1	cylinder tube length from centre of end eye, mm
L_2	piston rod length from centre of end eye, mm
L_3	guiding length of piston in cylinder tube, mm
m_{Cyl}	weight of the hydraulic cylinder, kg
P	buckling load, N
r	piston rod end eye bearing radii, mm
W	section modulus of the rod, mm^3
Δ	clearance between piston guide and cylinder tube, mm
μ	coefficient of friction for end eyes (default value of $\mu = 0.19$)
σ_y	yield strength of piston rod, N/mm^2

the inputs and the CM. By comparing calculations made by the simulator with those of the crane manufacturers' own crane calculators we have been able to verify that our CM is accurate. Unfortunately, sufficiently generalised, flexible and detailed modelling software or crane calculators are not common. Moreover, because we want a tool for optimising the design of cranes, we are not able to utilize existing crane calculators ready-made for existing crane models.

A block diagram depicting the functionality of the simulator is shown in Figure 4.

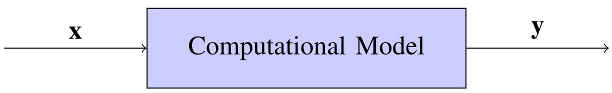

Figure 4: Offshore crane simulator. About 120 input design parameters in the input vector **x** are passed to a complex computational model, which in turn returns a number of KPIs and other properties of the resulting simulated crane in the vector **y**, e.g., load charts such as that depicted in Figure 2.

Note About Structure and Strength

Strength calculations and validation of the main structure is trivial and readily available from off-the-shelf finite element modelling (FEM) software, therefore these considerations have been excluded from the simulator. FEM software usually does not model the components though, as we do in our CM. We assume that structure and strength data are available from FEM software and focus on the constraints posed by the crane components, in particular the hydraulic cylinders and the slewing ring, for given crane configurations.

Choosing the Right Values

By adjusting the 120 parameters in the simulator, the effect of the parameters on a number of KPIs and other design criteria can be investigated. Of particular interest is the load chart, which can be calculated within a few seconds on an ordinary office computer, something that could take days when designers did their calculations by hand.

Since the formulas to calculate the KPIs are hard to invert or derive, finding closed, analytical solutions is generally not feasible. Consequently, a trial-and-error approach is still required by the designer, as depicted in Figure 5.

Genetic Algorithms

The GA is a bio-inspired stochastic search heuristic for solving search and optimisation problems. The algorithm is inspired by natural evolution, with elements such as inheritance, mutation, selection, and crossover. Most of the literature attributes the GA to Holland (1975), with subsequent popularisation by Goldberg (1989), and it is currently a very popular optimisation tool across many

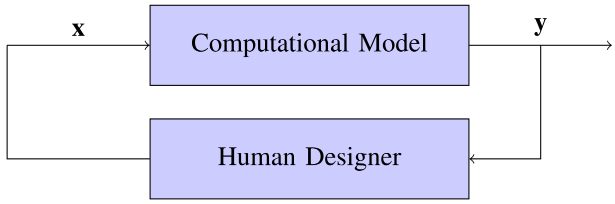

Figure 5: Human operator using a trial-and-error approach with the simulator to tune the input design parameters **x** until desired design criteria are met.

different disciplines (e.g., see Haupt and Haupt, 2004). The authors and colleagues have used GAs for a number of applications and projects, including dynamic resource allocation with maritime application (DRAMA), in which a fleet of tug vessels must be collectively positioned (Bye et al., 2010; Bye, 2012; Bye and Schaathun, 2014, 2015); adaptive locomotion of caterpillar-like robots (Li et al., 2014); universal control architecture for maritime cranes and robots (Sanfilippo, Hatledal, Schaathun, Pettersen and Zhang, 2013); the Java Intelligent Optimisation (JIOP) machine learning framework (Hatledal et al., 2014); and optimisation of swarms of boids (Alaliyat et al., 2014). Here, the intention is to use a GA in conjunction with our offshore crane simulator as a VP tool for optimising the design phase of offshore cranes. This requires selecting a suitable objective function that must incorporate the design criterions that we wish to optimise.

Objective Functions

Central to all GAs is the problem of determining values for a number of input parameters $\mathbf{x}$ such that some objective function $f(\mathbf{x})$ is optimised. Note that in GA terminology, the objective function is usually called a cost function if it is minimised, or a fitness function if it maximised. In this project, potential input parameters consist of any of the numerous components needed for building an offshore crane. The current version of our simulator incorporates more than 120 parameters that must be specified by the crane designer. Clearly, this large number of parameters makes the search space (the space of all possible combinations of parameter values) very large and design approaches such a trial-and-error will necessarily be both time-consuming and cost-inefficient and lead to suboptimal designs.

Nevertheless, choosing an appropriate objective function is no easy task. In addition to adhering to laws, regulations and standards, offshore cranes must be designed in accordance with the specific needs of the client. Optimising such a set of potentially conflicting design criteria (that is, there is a tradeoff between two or more objectives) is called multiobjective optimisation (MOO). Once an objective function, or in the case of MOO, several objective functions, has been selected, the GA can find an optimal solution by means of intelligently probing the search space and evolving better solutions. In the case of MOO, the GA will return a set of Pareto optimal solutions, which means that none of the objective functions can be improved without degrading others (e.g., see Haupt and Haupt, 2004; Arora, 2012, for details).

Description of a Basic GA

The basic steps that almost any GA consists of are outlined in high-level pseudocode in Algorithm 1 below, where we adopt a cost function as our objective function (loosely adapted from Haupt and Haupt, 2004):

```
/* INITIALISATION                                          */
define encoding scheme for chromosomes c;
define cost function f(c);
set criteria for selection, crossover, mutation, elitism;
generate initial population of chromosomes;
sort population in increasing order of cost;
bestChrom ← population[0];
set minCost, maxIterations;
i ← 1;
/* LOOP                                                    */
while i < maxIterations OR bestCost > minCost do
    evaluate cost for each chromosome;
    select chromosomes for mating;
    perform mating, crossover, mutation, elitism;
    update population;
    sort population in increasing order of cost;
    bestChrom ← population[0];
    bestCost ← f(bestChrom);
    i ← i + 1;
end
return i, bestChrom, bestCost;
decode bestChrom to original domain;
```

Algorithm 1: Basic GA.

A chromosome c is an encoded candidate solution to the problem of optimising an objective function $f(c)$. The objective function quantifies the quality of candidate solutions, that is, how well they fulfill the desired design criteria. The design parameters that we want to optimise must be translated (encoded) from their original domain to a format suitable for the GA, usually arrays of bits or real-valued numbers, in the latter case, often normalised to the interval $[0, 1]$. The bits or numbers are usually called genes.

The selection criterion determines which chromosomes in a population survives from one iteration to the next. For examle, using the roulette wheel method, the cost (fitness) associated with each chromosome is evaluated and the chromosomes are given a weighted selection probability according to their cost, where a smaller cost (greater fitness) results in a greater probability.

A pre-determined fraction, of chromosomes (typically half the population) is then randomly picked, with low cost (high fitness) chromosomes having a greater chance of being picked and kept for survival and reproduction.

For mating, several crossover methods exists, where genes from two parent chromosomes are combined into one or several offspring, which are then put back into the population, replacing those chromosomes that were not selected for mating.

After mating, a fraction of the chromosomes will have one or several of their genes mutated. This means flipping (inverting) bits for binary chromosomes, or changing the values of these genes to random numbers within some allowable range.

Next, each of the chromosomes in the updated population is evaluated by the objective function and the population is sorted in descending order of performance (ability to minimise cost or maximise fitness).

The process repeats until the maximum number of iterations has been reached, or the solution (the best chromosome) has reached a satisfactory performance. Then the algorithm ends and returns the best chromosome, which is decoded back to its original domain. In our case, the decoded solution specifies the optimal values for selected design parameters of an offshore crane.

Constraints

In terms of contraints of the crane design, the most interesting components are the hydraulic cylinders and the slewing ring. These components are commercially available “off-the-shelf” with given dimensions and performance ratings. In the case of the slewing ring, the maximum torque is of special interest, because it limits both the lifting distance and weight of the payload. The case of the cylinders are a bit more complex, however. When analysing the forces acting on the cylinders, we have to pay careful attention to the current angles of the joints in addition to the payload. Dependent of the forces acting on the cylinders it is possible to calculate two important constraints that originate from the properties of the cylinders: (i) maximum pressure; and (ii)buckling limit. Firstly, the cylinders will have a maximum hydraulic rating, independent of the cylinder position. This limit cannot be exceeded and constitues and important constraint on the properties of the crane. Secondly, depending on cylinder position, there will be a certain level of pressure where the cylinder will buckle (see mathematics presented above). Obviously the cylinder can handle higher pressure when close to the inner position than when close to the outer position. Both of these constraints can easily be incorporated by a GA when searching for optimal designs.

Automated Design Solution

With the framework we have presented above, we are able to automate the design process by replacing the human designer by a GA, as depicted in Figure 6. We

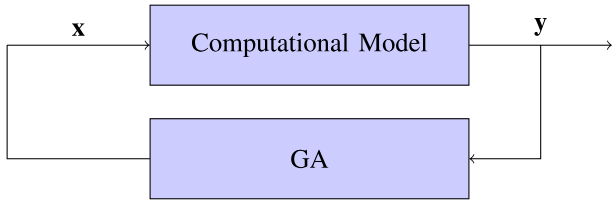

Figure 6: Automated design solution by means of a GA that automatically tunes the input design parameters **x** until desired design criteria are met.

propose the following solution using genetic algorithms (GA). A subset of the 120 design parameters are coded as variables (genes) in the GA. As an initial approach, we further simplify the problem by keeping a large number of the 120 parameters constant and let the GA optimise only those parameters that we consider most important for affecting our design criteria. As what we present here is work-in-progress, the point of this reduction in parameters is to reduce computation time and get a “feel” of our approach. We can easily extend the number of parameters later when we have gathered more insight into our automated VP process. The most important design criteria include KPI factors such as lifting capacities at different distances from the pedestal (derived from the load chart), weight and cost. The selected KPIs form the basis of the cost function of the GA and are weighted according to the preferences of the client.

RESULTS

Computational Model and Simulator

Currently, the CM is tailored for offshore knuckleboom cranes and has a very high level of detail. It incorporates mathematical models provided by our partner Seaonics AS, a design and manufacturing company of customised offshore lift and handling equipment as well as relevant laws, regulations, standards and design codes.

Our simulator is a parameterised Java implementation of the CM made by ICD Software AS, a software design and development company of offshore industrial control systems. In addition to an offline backend solution, two online server versions of the simulator have been developed.[1] One is a simple graphical interface for manually entering input values to the simulator. The other is using the WebSocket protocol, which is an advanced technology for interactive communication between client applications and a server. Using the WebSocket application program interface (API) enables computer programs, or clients, (instead of a human operator) to utilise the simulator and automatically simulate cranes for chosen sets of design parameters. Data is transferred by means of JavaScript Object Notation (JSON), which is a lightweight human-readable data-interchange format. The generation of a single full set of crane data from a particular choice of design parameters takes about 150–300 ms, whereas the overhead in transferring data between the simulator server and the client GA is in a similar range.

Graphical Web Interface

Figure 7 shows an example load chart generated by using the graphical web interface of the simulator. The load chart is divided into zones showing the SWL for different configurations of the crane and the chosen values of the 120 design parameters. The left column contains drawers where these design parameters can be set, whereas the right column shows some numerical results such as the load vector (position and SWL), slewing ring torque, boom angle, jib angle, and the main and jib cylinder data (compression force, buckling force and SWL, and pressure and its SWL).

Genetic Algorithm

Development of the genetic algorithm is still work-in-progress. We use the functional programming language Haskell for our implementation of the GA, as well as for the communication interface to the server version of the simulator. We expect a working prototype of the GA to be completed within the first half of 2015. Reasons for the choice of programming language is discussed in the next section.

[1]Access only via agreement with ICD Software AS.

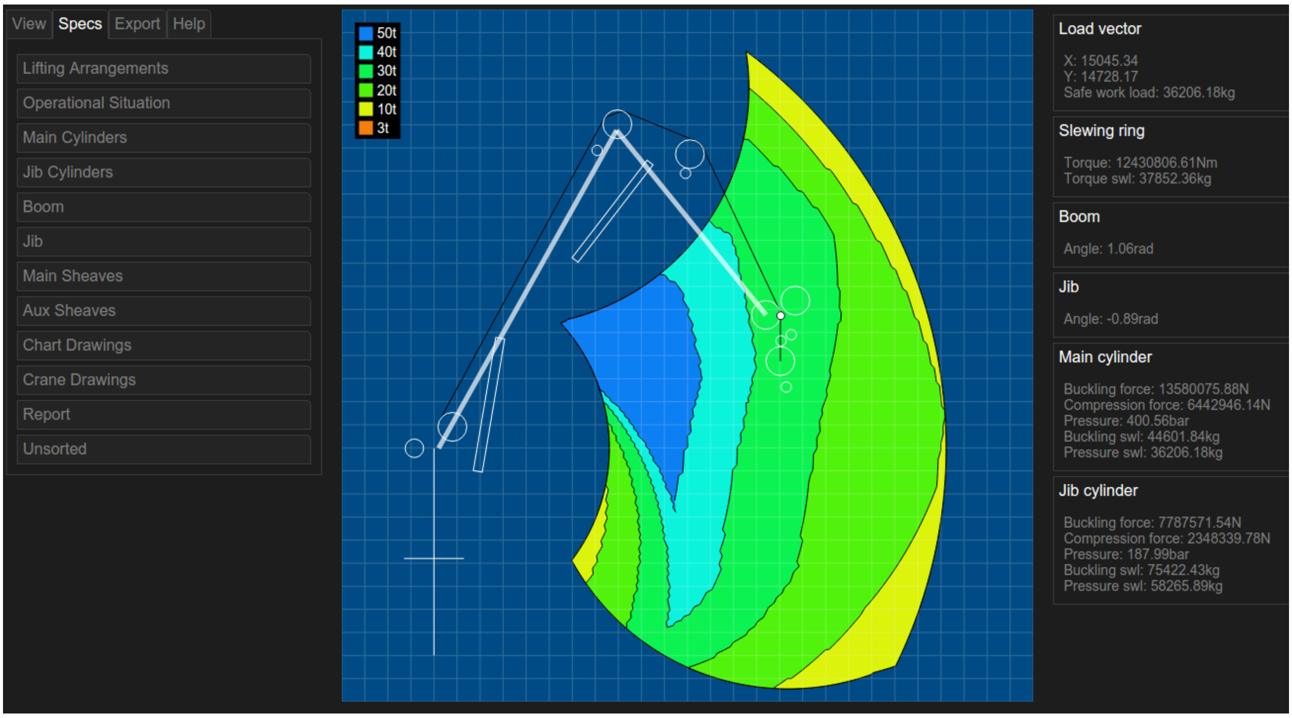

Figure 7: Example load chart generated by simulator using the graphical web interface.

DISCUSSION

This paper has described a CautoD tool for intelligent VP of offshore cranes. The three main parts of the tool are (i) a CM; (ii) a simulator, and (iii) a GA. Being work-in-progress, only (i) and (ii) have been completed at the current stage, whereas development of (iii) during the first half of 2015 is expected to lead to a working version of the complete system some time after this.

Functional Implementation of Genetic Algorithm

Haskell is a purely-functional programming language, which means that functions in Haskell are pure, there is no global state, and no side effects. In addition, the separation between pure and impure functionality makes code easier to debug. Code written in Haskell is therefore less error-prone and usually more concise, compact, and readable than imperative programming languages like C or Java.

Haskell is a good choice for parallel programming, which we believe is likely to be needed as the complexity of our simulator grows. Using pure parallelism guarantees deterministic processes and zero race conditions or deadlocks, however, non-pure concurrency related to pseudorandom number generators and other processes is also required.

Using Haskell for implementation makes our simulator very modular and extendable, something we believe is necessary in order to expand the simulator and design tool in the future.

Parallel Computing

The most computationally expensive part of the GA calculations is the evaluation of the cost function. Fortunately, calculating cost functions in a GA is known as "an embarrassingly parallel problem" because it involves solving many similar, but independent tasks simultaneously in parallel, with little or no need for intertask coordination and communication. Consequently, it is possible to speed up the GA by outsourcing cost function calculations to local computer clusters or computing clouds. An affordable and interesting option is to use general purpose computing on graphical processing units (GPGPUs), since GPUs in common modern desktop computer graphics cards are already optimized for parallelism.

Complexity and Sensitivity

We have not analysed algorithm complexity nor sensitivity with respect to the design parameters. A core of $N = 120$ parameters is implemented in the computational model but only $n \ll N$ parameters will be optimised by the GA for early phases of our work. When we have a successful solution with n parameters, we will begin to investigate both the effects of varying parameters (sensitivity) and expanding our solution and let the number n grow (complexity). Again, parallel computing may be needed in order to find optimal solutions within reasonable time.

Graphical User Interface and Added Functionality

It is essential to make our VP tool available to domain specialists without prior knowledge of AI or programming. Accordingly, we plan to develop a user interface where the weights in the fitness function of the GA are connected with sliders in the graphical user interface (GUI) that relate to the domain specialists' preferred KPIs or design criteria. Typical KPIs that can be weighted will include cost, lifting capacity, operating range and weight. The user shall then be able to move the sliders (e.g., in the range 0–100%) and adjust the relative importance of the KPIs and thus obtain the optimal crane for the given weights. In addition, the GA should be able to calculate and intelligently provide some realistic alternatives that the user might be interested in.

Concluding Remarks

Whilst our VP tool is currently not complete, we are confident that the framework we describe here will be of great value to crane designers, especially since our simulator is already in the process of being tested and used manually by human crane designers. We look forward to completing a working software of the tool as well as expanding it in the directions described above.

ACKNOWLEDGEMENTS

The SoftICE lab at AAUC wishes to thank ICD Software AS for their contribution towards the implementation of the simulator, and Seaonics AS for providing documentation and insight into the design and manufacturing process of offshore cranes. We are also grateful for the support provided by Regionalt Forskningsfond Midt-Norge and the Research Council of Norway through the VRI project *Artificial Intelligence for Crane Design (Kunstig intelligens for krandesign (KIK))*, grant no. 8321-14/04.

REFERENCES

Alaliyat, S., Yndestad, H. and Sanfilippo, F. (2014), Optimisation of Boids Swarm Model Based on Genetic Algorithm and Particle Swarm Optimisation Algorithm (Comparative Study), Proceedings of the 28th European Conference on Modelling and Simulation.

Arora, J. (2012), *Introduction to optimum design*, 3rd edn, Academic Press.

Bak, M., Hansen, M. and Nordhammer, P. (2011), Virtual prototyping-model of offshore knuckle boom crane, Proceedings of the 24th International Congress on Condition Monitoring and Diagnostics Engineering Management, pp. 1242–1252.

Bak, M. K. and Hansen, M. R. (2013), Analysis of Offshore Knuckle Boom Crane - Part One: Modeling and Parameter Identification, *Modeling, Identification and Control* 34(4), 157–174.

Bye, R. T. (2012), A receding horizon genetic algorithm for dynamic resource allocation: A case study on optimal positioning of tugs., *Series: Studies in Computational Intelligence* 399, 131–147. Springer-Verlag: Berlin Heidelberg.

Bye, R. T. and Schaathun, H. G. (2014), An improved receding horizon genetic algorithm for the tug fleet optimisation problem, Proceedings of the 28th European Conference on Modelling and Simulation, pp. 682–690.

Bye, R. T. and Schaathun, H. G. (2015), Evaluation heuristics for tug fleet optimisation algorithms: A computational simulation study of a receding horizon genetic algorithm, Proceedings of the 4th International Conference on Operations Research and Enterprise Systems (ICORES'15), pp. 270–282.

Bye, R. T., van Albada, S. B. and Yndestad, H. (2010), A receding horizon genetic algorithm for dynamic multi-target assignment and tracking: A case study on the optimal positioning of tug vessels along the northern Norwegian coast., Proceedings of the International Conference on Evolutionary Computation (ICEC 2010) — part of the International Joint Conference on Computational Intelligence (IJCCI 2010), pp. 114–125.

Cha, J.-H., Roh, M.-I. and Lee, K.-Y. (2010), Integrated simulation framework for the process planning of ships and offshore structures, *Robotics and Computer-Integrated Manufacturing* 26(5), 430–453.

Chan, S., Wong, M. and Ng, V. (1999), Collaborative solid modeling on the WWW, Proceedings of the 1999 ACM symposium on Applied computing, ACM, pp. 598–602.

Chang, H.-C., Lu, W. F. and Liu, X. F. (1999), WWW-based collaborative system for integrated design and manufacturing, *Concurrent Engineering* 7(4), 319–334.

Choi, S. and Cheung, H. (2008), A versatile virtual prototyping system for rapid product development, *Computers in Industry* 59(5), 477–488.

Chu, Y., Æsøy, V., Zhang, H. and Bunes, Ø. (2014), Modelling and simulation of an offshore hydraulic crane, 28th European Conference on Modelling and Simulation.

Chu, Y., Sanfilippo, F., Æsøy, V. and Zhang, H. (2014), An effective heave compensation and anti-sway control approach for offshore hydraulic crane operations, Mechatronics and Automation (ICMA), 2014 IEEE International Conference on, IEEE, pp. 1282–1287.

De Sa, A. G. and Zachmann, G. (1999), Virtual reality as a tool for verification of assembly and maintenance processes, *Computers & Graphics* 23(3), 389–403.

Det Norske Veritas (DNV) (1999), DNV's Rules for Certification of Lifting Appliances, Memo.

Goldberg, D. E. (1983), Computer-aided gas pipeline operation using genetic algorithms and rule learning, PhD thesis, University of Michigan Ann Arbor.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional.

Halse, K. H., Æsøy, V., Ponkratov, D., Chu, Y., Xu, J. and Pedersen, E. (2014), Lifting Operations for Subsea Installations Using Small Construction Vessels and Active Heave Compensation Systems: A Simulation Approach, ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers, pp. V01AT01A019–V01AT01A019.

Hare, W., Nutini, J. and Tesfamariam, S. (2013), A survey of non-gradient optimization methods in structural engineering, *Advances in Engineering Software* 59, 19–28.

Hatledal, L. I., Sanfilippo, F., Chu, Y. and Zhang, H. (2015), A Voxel Based Numerical Method for Computing and Visualizing the Workspace of Offshore Cranes, Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2015), St. John's, Newfoundland, Canada. Accepted for publication.

Hatledal, L. I., Sanfilippo, F. and Zhang, H. (2014), Jiop: a java intelligent optimisation and machine learning framework, Proceedings of the 28th European Conference on Modelling and Simulation (ECMS), Brescia, Italy, pp. 101–107.

Haupt, R. L. and Haupt, S. E. (2004), *Practical Genetic Algorithms*, 2nd edn, Wiley.

Holland, J. H. (1975), *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, Oxford, England.

Huang, T., Kong, C., Guo, H., Baldwin, A. and Li, H. (2007), A virtual prototyping system for simulating construction processes, *Automation in Construction* 16(5), 576–585.

Kamentsky, L. and Liu, C. (1963), Computer-automated design of multifont print recognition logic, *IBM Journal of Research and Development* 7(1), 2–13.

Kaveh, A. and Talatahari, S. (2009), Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization, Harmony search algorithms for structural design optimization, Springer, pp. 159–198.

Kim, H., Lee, J.-K., Park, J.-H., Park, B.-J. and Jang, D.-S. (2002), Applying digital manufacturing technology to ship production and the maritime environment, *Integrated*

Manufacturing Systems 13(5), 295–305.
Kirkpatrick, S. (1984), Optimization by simulated annealing: Quantitative studies, *Journal of statistical physics* 34(5-6), 975–986.
Kuttner, B. and Deitz, D. (1996), Reviewing designs in cyberspace, *Mechanical Engineering* 118(12).
Li, G., Zhang, H., Zhang, J. and Bye, R. T. (2014), Development of Adaptive Locomotion of a Caterpillar-like Robot Based on a Sensory Feedback CPG Model, *Advanced Robotics* 28(6), 389–401.
Maxfield, J., Fernando, T. and Dew, P. (1995), A distributed virtual environment for concurrent engineering, Virtual Reality Annual International Symposium, 1995. Proceedings., IEEE, pp. 162–170.
Mujber, T., Szecsi, T. and Hashmi, M. (2004), Virtual reality applications in manufacturing process simulation, *Journal of materials processing technology* 155, 1834–1838.
Pawlus, W., Choux, M., Hovland, G., Hansen, M. R. and Øydna, S. (2014), Modeling and simulation of an offshore pipe handling machine, Proceedings of the 55th Conference on Simulation and Modelling (SIMS 55), pp. 277–284.
Peng, Y., Yancong, L., Yongjun, S. and Guande, L. (2010), Optimum Piping Design on Offshore Platform Based on Improved Adaptive Genetic Algorithm, Proceedings of the 2010 WASE International Conference on Information Engineering-Volume 01, IEEE Computer Society, pp. 50–53.
Pezeshk, S., Camp, C. and Chen, D. (2000), Design of nonlinear framed structures using genetic optimization, *Journal of Structural Engineering* 126(3), 382–388.
Pratt, M. J. (1995), Virtual prototypes and product models in mechanical engineering, *Virtual Prototyping–Virtual environments and the product design process* 10, 113–128.
Rajeev, S. and Krishnamoorthy, C. (1992), Discrete optimization of structures using genetic algorithms, *Journal of Structural Engineering* 118(5), 1233–1250.
Sanfilippo, F., Hatledal, L. I., Schaathun, H. G., Pettersen, K. Y. and Zhang, H. (2013), A universal control architecture for maritime cranes and robots using genetic algorithms as a possible mapping approach, Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on, IEEE, pp. 322–327.
Sanfilippo, F., Hatledal, L. I., Zhang, H. and Pettersen, K. Y. (2014), A mapping approach for controlling different maritime cranes and robots using ANN, Mechatronics and Automation (ICMA), 2014 IEEE International Conference on, IEEE, pp. 594–599.
Sanfilippo, F., Hildre, H. P., Æsøy, V., Zhang, H. and Pedersen, E. (2013), Flexible Modeling And Simulation Architecture For Haptic Control Of Maritime Cranes And Robotic Arm, ECMS, pp. 235–242.
Sarshar, M., Christiansson, P. and Winter, J. (2004), Towards virtual prototyping in the construction industry: the case study of the DIVERCITY project, Proceedings of the World IT Conference for Design and Construction, pp. 18–24.
Schöning, C. (2014), Virtual prototyping and optimisation of microwave ignition devices for the internal combustion engine, PhD thesis, University of Glasgow.
Shen, Q., Gausemeier, J., Bauch, J. and Radkowski, R. (2005), A cooperative virtual prototyping system for mechatronic solution elements based assembly, *Advanced Engineering Informatics* 19(2), 169–177.
Shyamsundar, N. and Gadh, R. (2002), Collaborative virtual prototyping of product assemblies over the Internet, *Computer-Aided Design* 34(10), 755–768.
Černỳ, V. (1985), Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of optimization theory and applications* 45(1), 41–51.
Waly, A. F. and Thabet, W. Y. (2003), A virtual construction environment for preconstruction planning, *Automation in construction* 12(2), 139–154.
Wang, G. G. (2002), Definition and review of virtual prototyping, *Journal of Computing and Information Science in engineering* 2(3), 232–236.
Weyrich, M. and Drews, P. (1999), An interactive environment for virtual manufacturing: the virtual workbench, *Computers in industry* 38(1), 5–15.
Wöhlke, G. and Schiller, E. (2005), Digital planning validation in automotive industry, *Computers in industry* 56(4), 393–405.
Xu, X. and Luo, Y. (2010), Force finding of tensegrity systems using simulated annealing algorithm, *Journal of structural engineering* 136(8), 1027–1031.
Yerrapathruni, S. (2003), Using 4 D CAD and Immersive Virtual Environments to Improve Construction Planning, PhD thesis, Architectural Engineering.

AUTHOR BIOGRAPHIES

ROBIN T. BYE[2] graduated from the University of New South Wales, Sydney with a BE (Hons 1), MEngSc, and a PhD, all in electrical engineering. Dr. Bye began working at the AAUC as a researcher in 2008 and has since 2010 been an associate professor in automation engineering. His research interests belong to the fields of artificial intelligence, cybernetics, and neuroengineering.

OTTAR L. OSEN is MSc in Cybernetics from the Norwegian Institute of Technology in 1991. He is the head of R&D at ICD Software AS and an assistant professor at AAUC.

BIRGER SKOGENG PEDERSEN is a student of automation engineering at AAUC and a software engineer at ICD Software AS.

[2] www.robinbye.com