A SOFTWARE FRAMEWORK FOR INTELLIGENT COMPUTER-AUTOMATED PRODUCT DESIGN

Robin T. Bye[*], Ottar L. Osen[*,†] Birger Skogeng Pedersen[*],
Ibrahim A. Hameed[*] and Hans Georg Schaathun[*]
[*] Software and Intelligent Control Engineering Laboratory
Faculty of Engineering and Natural Sciences
Norwegian University of Science and Technology
NTNU in Ålesund, Postboks 1517, NO-6025 Ålesund, Norway
[†] ICD Software AS
Hundsværgata 8, NO-6008 Ålesund, Norway

KEYWORDS

Virtual Prototyping; Product Optimisation; Artificial Intelligence; Genetic Algorithm

ABSTRACT

For many years, NTNU in Ålesund (formerly Aalesund University College) has maintained a close relationship with the maritime industrial cluster, centred in the surrounding geographical region, thus acting as a hub for both education, research, and innovation. Of many common relevant research topics, virtual prototyping is currently one of the most important. In this paper, we describe our first complete version of a generic and modular software framework for intelligent computer-automated product design. We present our framework in the context of design of offshore cranes, with easy extensions to other products, be it maritime or not. Funded by the Research Council of Norway and its Programme for Regional R&D and Innovation (VRI), the work we present has been part of two separate but related research projects (grant nos. 241238 and 249171) in close cooperation with two local maritime industrial partners.

We have implemented several software modules that together constitute the framework, of which the most important are a server-side crane prototyping tool (CPT), a client-side web graphical user interface (GUI), and a client-side artificial intelligence for product optimisation (AIPO) module that uses a genetic algorithm (GA) library for optimising design parameters to achieve a crane design with desired performance. Communication between clients and server is achieved by means of the HTTP and WebSocket protocols and JSON as the data format.

To demonstrate the feasibility of the fully functioning complete system, we present a case study where our computer-automated design was able to improve the existing design of a real and delivered 50-tonnes, 2.9 million EUR knuckleboom crane with respect to some chosen desired design criteria.

Our framework being generic and modular, both client-side and server-side modules can easily be extended or replaced. We demonstrate the feasibility of this concept in an accompanying paper submitted concurrently, in which we create a simple product optimisation client in Matlab that uses readily available toolboxes to connect to the CPT and optimise various crane designs by means of a GA. In addition, our research team is currently developing a winch prototyping tool to which our existing AIPO module can connect and optimise winch designs with only small configuration changes. This work will be published in the near future.

Corresponding author: Robin T. Bye, `robin.t.bye@ntnu.no`.

INTRODUCTION

With the geographical heart of the Norwegian maritime cluster located on campus, NTNU in Ålesund (formerly Aalesund University College) has played the role of a hub for collaborations in education, research, and innovation. According to the Global Centre of Expertise (GCE) Blue Maritime, one of only three industrial clusters in Norway that have been awarded this prestigious title and funding from the Norwegian Innovation Clusters Programme, the Norwegian maritime cluster consists of 13 design companies, 14 ship yards, 20 ship-owner companies, 169 equipment suppliers, 22,500 employees, and a total annual revenue of about 5.7 billion EUR.[1] Together with two of these companies, ICD Software AS (provider of industrial control systems software)[2] and Seaonics AS (designer and manufacturer of offshore equipment),[3] we have received funding from the Research Council of Norway and its Programme for Regional R&D and Innovation (VRI) for two independent but related research projects (grant nos. 241238 and 249171) for using artificial intelligence (AI) for intelligent computer-automated design (CautoD) of offshore cranes and winches, respectively.

Our main focus is on the development of a generic and modular software framework for intelligent CautoD of products such as crane and winches. We use the recently completed crane design project for exemplification but emphasise that our work has strong synergies with our

[1] `www.bluemaritimecluster.no`, accessed 8 February 2016.
[2] `www.icdsoftware.no`
[3] `www.seaonics.com`

ongoing winch design project and generically can be extended to other products and CautoD methodologies.

In the following, we begin with a background overview of virtual prototyping (VP) in general and CautoD in particular, offshore cranes, and the motivation and aim of our work. Next, we outline the method we have used, including details about the software architecture of our framework and each of its main components. Then we demonstrate how our first, complete version of our system is able to improve the design of a real knuckleboom crane that has already been sold and delivered to a company in Baku, Azerbaijan. Finally, we discuss our results and impact on future work.

BACKGROUND

Virtual Prototyping (VP)

We may define 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 (Pratt, 1995). Both the shipbuilding (Kim et al., 2002) and automotive (Wöhlke and Schiller, 2005) industries have made significant use of VP aspects such as modelling, simulation, and visualisation techniques, in 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).

Common VP methodologies include computer-aided design (CAD), realistic virtual environments (VEs), virtual reality (VR), and CautoD. CAD is related to the use of computer systems to aid in the creation, modification, analysis, or optimization of a design (Narayan et al., 2008), and is generally associated with 2D and 3D modelling software. VEs can be used to improve collaboration and teamwork in product design, for example in construction engineering (e.g., Waly and Thabet, 2003; Sarshar et al., 2004; Yerrapathruni, 2003) or component and assembly design (e.g., Shyamsundar and Gadh, 2002; Chang et al., 1999; Chan et al., 1999), and can be particularly useful in multidisciplinary product development (e.g., mechatronics engineering), to avoid inefficient communication between designers and engineers with different backgrounds that can slow the design process (Shen et al., 2005). Likewise, employing VR as a collaborative VP tool can greatly improve the product design, test and review loop before committing to physical fabrication (Choi and Cheung, 2008). Here, our main focus is on applying AI for CautoD, and a GA in particular, to automate and optimise the design phase of product development.

Computer-Automated Design (CautoD)

Likely the first scientific report of CautoD was reported by Kamentsky and Liu (1963), who created a computer programme for determining suitable logic circuits satisfying certain hardware constraints while at the same time evaluating the ability of the logics to perform character recognition. Many contributions of CautoD have been made since then, particularly in the field of structural engineering (see Hare et al., 2013, for a survey). The general paradigm is that of optimisation, where one formulates the design problem as the optimisation of an objective function; that is, minimisation of a cost function or maximisation of a fitness function. For complex optimisation problems for which analytical and exact solutions are difficult or impossible to obtain, AI methods such as machine learning and evolutionary computation can often be used with great success (see Zhang et al., 2011, for a survey).

A seminal example of design optimisation using AI is the simulated annealing (SA) algorithm (Kirkpatrick, 1984; Černý, 1985), which is still in use today, e.g., for CautoD of tensegrity systems (Xu and Luo, 2010). Another highly influential algorithm is the GA, which has been used for a variety of design optimisation problems, including computer-based control of gas pipeline systems (Goldberg, 1983), structural systems (Rajeev and Krishnamoorthy, 1992), 2D geometrical nonlinear steel-framed structures (Pezeshk et al., 2000), the piping process of offshore drilling platforms (Peng et al., 2010), and the influential work on design planar and space structures by Erbatur et al. (2000). Finally, we would like to guide the reader to the work of Kaveh and Talatahari (2009), who used particle swarm optimisation (PSO) and ant colony optimization (ACO) to find an optimal design of different types of truss structures, and the work of Schoning and Li (2013); Schöning (2014), who used a set of different AI algorithms that could connect and interrogate a simulator for a homogeneous charge microwave ignition system in search of novel design solutions, much in the same vain as the software framework we present here.

Virtual Prototyping of Offshore Cranes

An offshore crane such as the one in Figure 1 is a complex system of components interacting to achieve safe movement of heavy goods, often under harsh and difficult conditions.

Even simple versions of such offshore cranes consist of a large number of components, including hooks, winches, slewing rings, cylinders, booms, hinges, sheaves, and pedestals (see Figure 2 for an illustration of the main components). The choice of crane components and their physical properties and interrelationships determines various measures of performance of interest to the crane designer. For example, key performance indicators (KPIs) such as the desired workspace, the working load limit (WLL) and the safe working load (SWL) within that workspace; the total weight of the crane; its control system characteristics, durability, installation and operating costs; and safety concerns are all factors that the crane designer must take into account when designing a new crane. In addition, laws, regulations, and the use of design codes such as the standards provided by classification socities like DNV-GL, Lloyd's Register Group Limited, and the

Figure 1: Boomerang-shaped knuckleboom crane by Seaonics AS with an ingeniously designed wire or fibre rope running directly from winch to boom tip. The design increases the workspace compared to standard knuckle-boom cranes while reducing wear and tear of the wire or fibre rope. This crane is ideal for arctic operations and fibre rope use. Image courtesy of Seaonics AS.

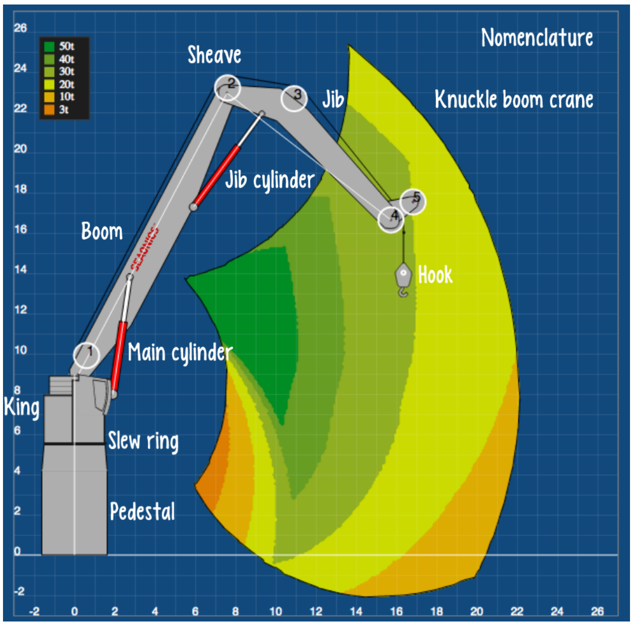

Figure 2: Illustration of the main components of an offshore knuckleboom crane and its 2D load chart. Image courtesy of ICD Software AS.

American Bureau of Shipping all put constraints on the choice of design parameters.

VP is currently the main focus of research of the Software and Intelligent Control Engineering (SoftICE) Laboratory[4] at NTNU in Ålesund. For the last few years, the SoftICE lab and our colleagues in the Mechatronics Laboratory[5] have published several papers relating to VP and aspects of modelling, simulation and visualisation of offshore cranes, marine operations, installations, and industrial robotics (e.g., Bye et al., 2015; Sanfilippo et al., 2013, 2015; Chu et al., 2014, 2015; Chaves et al., 2015; Hatledal et al., 2015).

Of particular interest is the recent work by colleagues Hatledal et al. (2015), who present 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 depends on the properties of crane components such as the length of the boom and jib and the maximum pressure of their cylinders, 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 the boom and jib and the size of their cylinders can be designed to yield better workspace characteristics.

Several other researchers have recently published relevant work relating to VP of offshore cranes, operations, and control. Bak et al. (2011); Bak and Hansen (2013); Peng et al. (2010); Pawlus et al. (2014) present VP of offshore knuckleboom cranes and pipe handling equipment, including techniques for modelling, simulation, and parameter identification that can aid in the VP process. Park and Le (2012) employ VP technology for modelling and control design of a mobile harbour crane system, whereas He et al. (2014) focus on VP-based multibody systems dynamics analysis of offshore cranes.

The work presented above shows that VP of offshore cranes and is an active field of research, however, as we will detail in the following section, most of this work can be improved by intelligent automation of the design process, which is the motivation for our own work.

Motivation and Aim

While the contributions above are valuable, the various models, calculations, simulations, and visualisations are mainly used as VP tools requiring a human to manually try and test design solutions until satisfied. For example, the work of Hatledal et al. (2015) describe how their VP environment can be used for a trial-and-error procedure to improve the traditional experience-based rule-of-thumb approaches employing pen-and-pencil or spreadsheet calculations commonly employed in the maritime industry but their method is hardly satisfactory given the large number of design parameters involved.

The findings reported in the literature typically describe various means to determine offshore crane properties and behaviour based on pre-determined design parameters, analogous to calculating the forward kinematics of a robotic arm. However, the inverse problem, namely that of choosing appropriate, possibly conflicting, values for numerous offshore crane design parameters such that some desired design criteria are satisfied, is much harder.

The aim of our two research projects is to solve this problem for respectively offshore cranes and winches

[4] `blog.hials.no/softice`
[5] `www.mechatronics.hials.org`

by means of an intelligent CautoD software framework employing a GA for searching through the vast number of design choices and combinations until an optimised design is found. The design will inevitably involve tradeoffs and thus be Pareto optimal, which means that improving some aspect of the design will necessarily detract from another.

METHOD

Software Architecture

The diagram in Figure 3 shows the software architecture for our complete system.

We have adopted a server-client software architecture, in which the CPT act as as server to which clients such as the AIPO module and the web GUI can connect via two different communication interfaces. The first interface uses the Hypertext Transfer Protocol (HTTP), with data transferred as JavaScript Object Notation (JSON), a lightweight human-readable data-interchange format. The second interface uses WebSocket (WS), which is a protocol providing full-duplex communication channels over a single TCP connection, also with JSON as the data format. Because WS enables streams of messages on top of TCP, uing WS for communication is advantageous for bidirectional conversations involving many small messages being sent to and from a server. If this is not a concern, clients may prefer to use HTTP, since this protocol is very mature and well supported in many different programming languages. Two server modules that implement the communication interfaces both connect to a third module, the crane calculator. All three modules are implemented in Java and together constitute the CPT.

On the client side, we have implemented an AIPO module that uses a GA library module for optimisation. Both modules are programmed in Haskell and are used for CautoD of offshore cranes. In addition, we have implemented a web GUI in JavaScript. Finally, to test the modularity of the software framework, we have also implemented a crane optimisation client in Matlab. The Matlab client and a test suite for optimal crane design is presented in an accompanying paper submitted together with this paper (Hameed et al., 2016).

The CPT server accepts messages that conform to a set of rules that we have defined. Specifically, the AIPO module sends a JSON message object consisting of three parts (subobjects): (i) a "base" object with a complete set of default design parameter values; (ii) a "mods" object with a subset of design parameter values that modifies the corresponding default values; and (iii) a "kpis" object with the desired KPIs to be calculated and returned by the CPT.

The design is highly modular. On the server side, we can swap prototyping tools as long as they conform to the HTTP/JSON or WS/JSON communication interfaces and message formats that we have defined. Thus, we can easily create a winch prototyping tool (currently under development in a parallel project) or another product prototyping tool, as long as the tool is able to receive such messages as described above and calculate and return desired KPIs. Likewise, on the client side, different optimisation clients can connect to the CPT (or another product optimisation tool), again as long as they conform to the communication interface.

Crane Calculator

The components of an offshore crane may total several thousand parameters, making it infeasible to manually pick good values for each parameter. However, through the years, crane designers have been able to reduce this number to a set of about 120 design parameters that are considered the most important. Based on the values of these parameters, which can be set manually or by a CautoD tool such as AIPO, our crane calculator is able to calculate a fully specified crane design and its associated KPIs. The goal of the designer is thus to determine appropriate design parameter values that achieve desired design criteria (KPIs), while simultaneously meeting requirements by laws, regulations, codes and standards.

The accuracy of our crane calculator has been verified against other crane calculators and spreadsheets currently in use in the industry, and, as a result, Seaonics AS has already adopted the CPT server and web GUI client for manual crane design.

A block diagram depicting how the crane calculator can be used for manual crane design is shown in Figure 4.

Web Graphical User Interface (GUI)

To simplify practical use of the crane calculator, we have created a web graphical user interface (GUI) that can be used to interact with the crane calculator via WS/JSON communication. Using the web GUI to manually adjust the 120 design parameters in the crane calculator by trial-and-error, the effect of the parameters on a number of KPIs and other design criteria can be investigated numerically, with the possibility for exporting to text files, and visually, by depicting the main components of the crane and its 2D SWL load chart.

The GUI consists of three parts aligned horizontally: a left column with "drawers" where design parameters can be modified; the visualised crane and load chart in the middle; and a right column that shows numerical results, including 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).

The load chart is a graphical representation of the lifting capacity in the workspace of a particular crane design as determined by the crane calculator. The workspace is divided into zones, where each zone is highlighted by a colour indicating the maximum SWL in tonnes in that particular crane configuration.

Due to space consideration, we refer to Bye et al. (2015) for a screenshot of the GUI.

Artificial Intelligence for Product Optimisation (AIPO)

The manual design process using the web GUI together with the CPT is cumbersome. Indeed, there are more

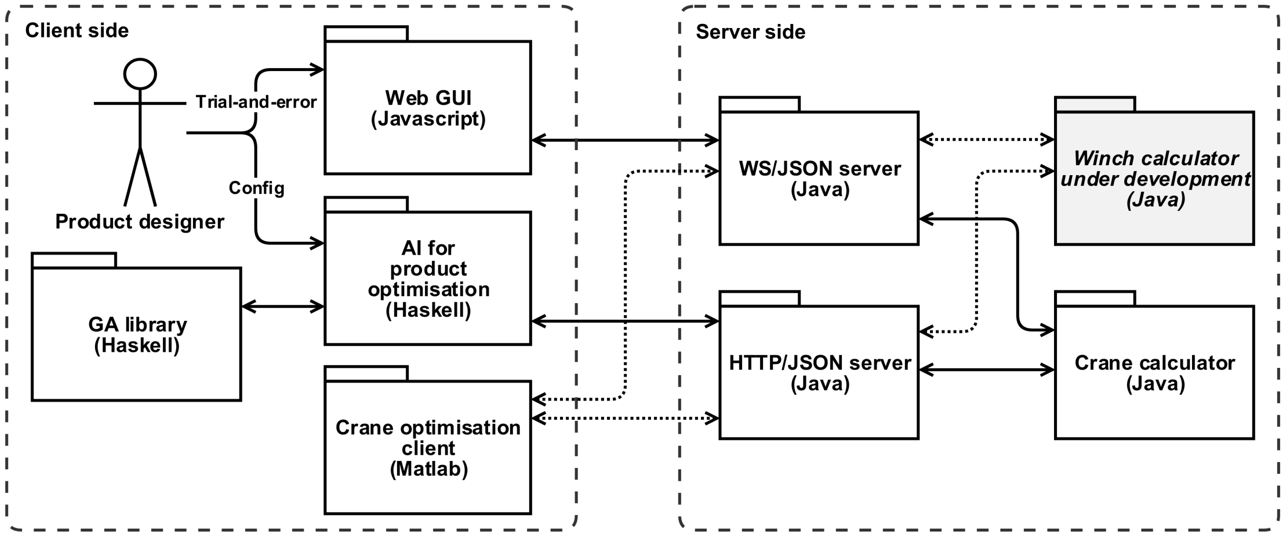

Figure 3: Software architecture for intelligent CautoD of offshore cranes, winches, or other products. The winch calculator is shown in grey because it is still under development. Solid and dashed lines indicate whether modules and their interconnections are inside or outside the scope of this paper, respectively.

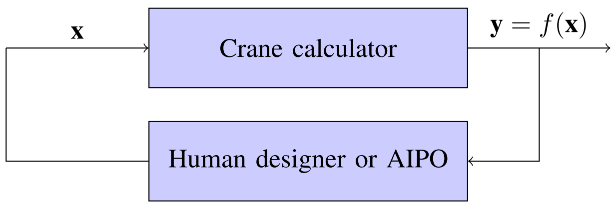

Figure 4: Human crane designer using a manual trial-and-error approach with the crane calculator to tune the input design parameters $\mathbf{x}$ until the resulting design $\mathbf{y}$ matches the desired design criteria. Our aim is automate this process by a CautoD solution, namely AIPO.

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 a manual trial-and-error approach will necessarily be both time-consuming and cost-inefficient and lead to suboptimal designs. Hence, we have developed an AI for product optimisation (AIPO) software module replacing the human operator in Figure 4 in order to automate and optimise the design process.

The product designer must configure the AIPO module with one or several objective functions based on the KPIs and design criteria of the product to be designed, be it an offshore crane, a winch, or some other product. In addition, the product designer must configure the AI algorithm to be used for optimisation. For example, if using a GA, the designer must set certain parameter values such as population size, mutation rate, and choose methods for selection and crossover.

Using AI optimisation methods such as a GA in this case, the module interrogates the CPT with an objective function until an optimised design solution is obtained. A short description of GAs and objective functions are provided in the following sections.

Genetic Algorithms (GAs)

The GA is inspired by natural evolution, with elements such as inheritance, mutation, selection, and crossover. It is well suited for hard optimisation problems where solutions are difficult or impossible to obtain in polynomial time. Additionally, an advantage of GAs is that constraints can be handled with ease. For crane design, the most important components are the hydraulic cylinders and the slewing ring. 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. In the case of the cylinders, their maximum pressure and buckling limit is of major importance. These and other constraints can easily be incorporated in a GA optimisation procedure.

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 different disciplines (e.g., Haupt and Haupt, 2004).

The authors and colleagues have used GAs for a number real-world optimisation problems (e.g., Bye, 2012; Bye and Schaathun, 2014, 2015; Bye et al., 2015; Alaliyat et al., 2014; Sanfilippo et al., 2013). Here, we let the AIPO module use a GA in conjunction with the CPT for optimising the design phase of offshore cranes. Nevertheless, with only minor adjustments, we are able to adopt the same process for CautoD of winches (currently under development) or other products. More details about the GA we use here can be found in a previous paper (Bye et al., 2015).

Objective Functions

In order for the GA to succeed in optimising design solutions, a suitable objective function must be selected that incorporates the design criteria that we wish to optimise. That is, the GA must determine values for a number of input parameters $\mathbf{x}$ such that the selected objective function $f(\mathbf{x})$ is optimised.

Nevertheless, choosing an appropriate objective function is not trivial. In addition to adhering to laws, regulations and standards, offshore cranes must be designed in accordance with the specific needs of the customer. 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). Using a GA for MOO, the GA will not return a single solution but 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).

Choice of Implementation Languages

The web GUI was implemented in JavaScript, the CPT in Java, and the AIPO module and the GA library in Haskell.

JavaScript has stayed popular for more then a decade and has a huge user base and easy access to resources. Most web client supports JavaScript out of the box. Since only a text editor and web browser is needed for creating programs, modifications can be tested immediately by refreshing the browser, thus enabling rapid development and testing. Initially, therefore, the crane calculator was part of a JavaScript web client, but this choice had several drawbacks. Most importantly, the source code was open for everyone to see, which was unacceptable to our industrial partners. In addition, processing speed was dependent of the client's hardware. Thus, to ensure sufficient processing power independent of user hardware, the JavaScript implementation was moved "as is" to the server-side. Unfortunately, there were speed issues and further development was needed in order to develop a suitable application programme interface (API) for the AIPO module. We therefore decided to opt for a different implementation language able to solve these issues, and also chose to separate the calculator from the handling of communication with clients. In order to keep migration costs low, a language with syntax similar to the C-like syntax of JavaScript was desired. Java was therefore chosen, with the additional advantage of its portability to different platforms. If even higher performance will be needed in the future, we might use compiled languages such as C, C++ or C#, or a functional language like Haskell, which is well suited for parallellisation.

For the AIPO module and its interconnected GA library, we chose Haskell. 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 software framework 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 still required.

Using Haskell for AIPO and GA implementation makes this part of our framework very modular and extensible, something we believe is necessary in order to expand the framework and design tools in the future.

Parallel Computing

The most computationally expensive part of the GA calculations is the evaluation of the objective function. Fortunately, calculating objective 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 objective 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 optimised for parallelism.

The GA library we have developed already supports parallel processing, thus, we should be able to extend our framework to apply parallel computing with little or no modification. Note, however, that we currently do not take advantage of parallel processing, since the evaluation of the cost function is performed by the server-side CPT, and not the client-side GA. In future work, we may consider improving the server-side CPT implementation by parallel processing, for example by recoding in Haskell and employing GPGUs for parallel evaluation of objective functions.

Case Study

To test the ability of our software framework to perform optimised CautD of an offshore crane, we used a real knuckleboom crane as a nominal benchmark against which our optimised crane design could be compared. The nominal crane has been designed, sold, and delivered by Seaonics AS to a company in Baku, Azerbaijan. The crane had a total delivery price of approximately 2.9 million EUR.

Two KPIs were chosen as components of an objective function to be optimised: (i) the maximum safe working load $\mathrm{SWL}_{\mathrm{max}}$ and (ii) the total crane weight W. Whilst the total crane delivery price is of great concern, we do not currently have price estimates as a function of crane design implemented in the CPT. Nevertheless, the total weight can to some extent be used as a proxy for price, because price will be correlated to weight, and one wants to minimise both measures. Moreover, these cranes are installed onboard vessels and reduced crane weight allows a higher deadweight tonnage (DWT). Hence, weight is important for both capital and operating expenditure.

The maximum SWL, on the other hand, is a measure of the maximum lifting capacity of the crane in the entire workspace. In a particular crane configuration, with the tip of the crane in a particular position of the workspace where SWL is maximum, all other configurations with the crane tip in surrounding positions will have a SWL less than or equal to the maximum SWL.

The goal of the crane optimisation was thus to maximise $\mathrm{SWL}_{\max}$ while simultaneously minimising W. To achieve this goal, we let the optimisation function be a fitness function f_1 given by

$$f_1 = \frac{\mathrm{SWL}_{\max}}{\mathrm{W}}$$

The evaluation of f_1 will increase when $\mathrm{SWL}_{\max}$ increases and/or W decreases, and vice versa.

There are several other possible choices for objective functions, including handling with MOO, some of which are studied in our accompanying paper (Hameed et al., 2016). Here, we are mainly concerned with showing proof-of-concept, namely that our complete software framework performs as intended. A simple case study and a single objective function suffices for this purpose.

As optimisation variables, we chose four design parameters that greatly affect both $\mathrm{SWL}_{\max}$ and W: (i) the boom length; (ii) the jib length; (iii) the maximum pressure of the boom cylinder; and (iv) the maximum pressure of the jib cylinder. The parameter values were constrained to a range with minimum and maximum limits. All other design parameters were identical to those of the nominal crane.

RESULTS

Using the GA with a population size (set of candidate design solutions) of 100 and running for 50 generations, giving a grand total of 5,000 evaluated designs, took 98.4 minutes, including transfer times between the AIPO client and CPT server. The best design solution found is presented in Table 1, and shows that compared with the nominal benchmark crane, the maximum SWL improved from 100.0 tonnes to 142.1 tonnes, or an improvement of 42.2%, while simultaneously, the weight of crane was reduced from 50.8 tonnes to 44.0 tonnes, or an improvement of 13.5%. The objective function evaluated at the optimised design parameters was improved by 64.3%.

A graphical representation of the load charts of the nominal crane and the optimised crane design is shown in Figure 5. From the diagram, it is clear that the optimised design has resulted in a a crane with a much smaller workspace than the nominal crane but with much higher lifting capacity.

DISCUSSION

In this paper, we have presented our first version of a generic and modular software framework for intelligent computer-automated product design. To demonstrate proof-of-concept of the fully functioning complete system, we did a case study where our CautoD solution was able to improve the existing design of a real and delivered 50-tonnes, 2.9 million EUR knuckleboom crane with respect to the objectives of maximising the SWL and minimising the total crane weight. Whilst the results are sufficient for the purpose of proof-of-concept, we wish to emphasise that for real crane design, much more sophisticated objective functions must be constructed, able to handling a variety of different, possibly competing, design criteria. For example, our optimised design solution in the case study above outperforms the real crane on the objectives we have chosen but may fail on other design criteria not encompassed by our choice of objective function. For instance, the workspace of the optimised crane is much smaller compared to the real crane (see Fig. 5) and may not satisfy the needs of the company buying the crane.

Generic and Modular Software Architecture

We have implemented several software modules (see Figure 3) that together constitute the framework: (a) the server-side CPT calculates a fully specified crane design and a number of key performance indicators based on a set of about 120 input design parameters; (b) the client-side web GUI facilitates manually setting the design parameters of the CPT as well as providing an immediate visualisation of the resulting crane and its 2D workspace SWL load chart; (c) the client-side AIPO module uses a GA library for optimising the design parameters to achieve a crane design with desired performance. Communication between clients and server is achieved by means of the HTTP and WS protocols, and with standardised JSON messages as the data format.

Our framework being generic and modular, both client-side and server-side modules can easily be extended or replaced. On the client-side, crane designers may choose to create their own custom-made computer-automated design modules or GUIs that can communicate with the CPT to obtain desired crane designs and visualisations. We demonstrate the feasibility of this concept in an accompanying paper submitted concurrently (Hameed et al., 2016), in which we create a simple product optimisation client in Matlab that connects to the CPT and optimises various crane designs by means of a GA. On the server-side, crane designers can provide other product prototyping tools to which our existing AIPO module can connect and optimise the product design.

Furthermore, our research team is currently developing a winch prototyping tool to which our existing AIPO module can connect and optimise winch designs with little or no modification. This work will be published in the near future.

Complexity and Sensitivity

We have not analysed algorithm complexity nor sensitivity with respect to the design parameters. A core of $N = 120$ design parameters has been implemented in the crane calculator but only $n = 4 \ll N$ parameters were

measure	units	nominal	limits (min, max)	optimised	difference	change
boom length	mm	15,800	(12,000, 26,000)	12038	-3,762	-23.8%
jib length	mm	10,300	(6,000, 16,000)	6124	-4,176	-40.5%
boom cylinder max pressure	bar	315	(100, 400)	383	68	21.6%
jib cylinder max pressure	bar	215	(50, 300)	262	47	21.8%
SWL_{max}	kg	99,978	-	142,138	42,160	42.2%
W	kg	50,856	-	44,014	-6,842	-13.5%
$f_1 = \text{SWL}_{\text{max}}/\text{W}$	-	1.97	-	3.23	1.26	64.3%

Table 1: Optimised crane design compared with a nominal benchmark crane.

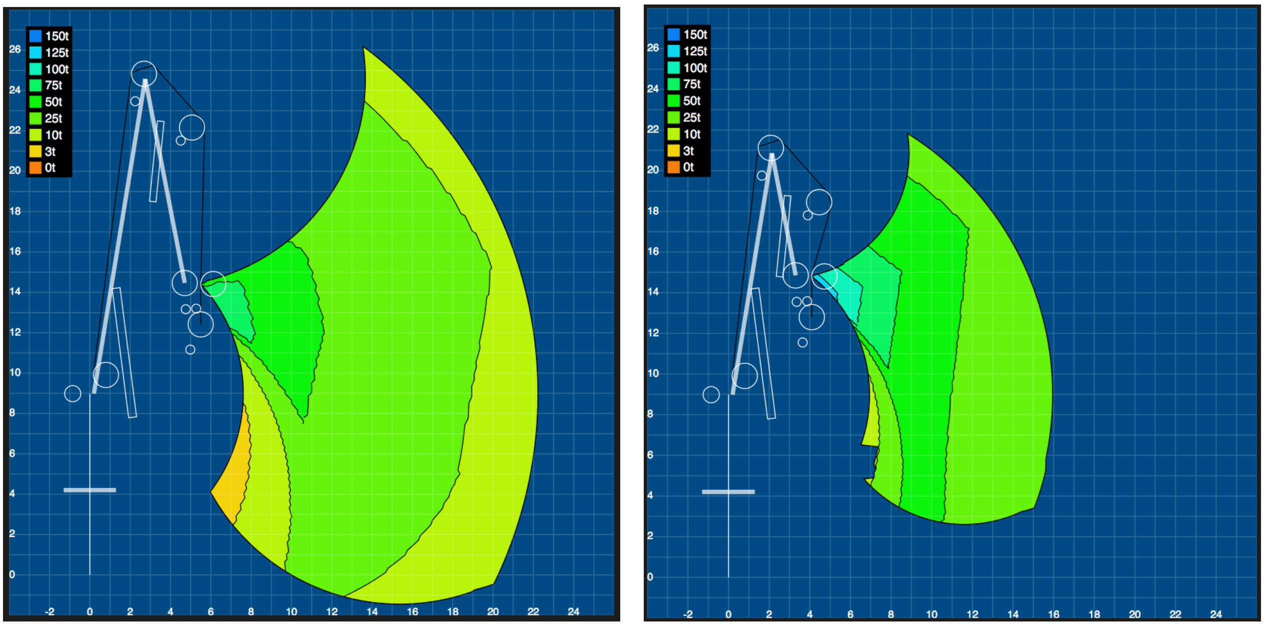

Figure 5: 2D SWL load charts for a nominal benchmark crane (left) and the optimised crane (right).

optimised by the GA in our case study. Now that we have demonstrated proof-of-concept of our framework with 4 parameters, we are in the process of investigating both the effects of varying both the choice of parameters and their values (sensitivity) and expanding our solution and let the number n grow (complexity). As the number of design parameters grows, the search space grows exponentially. Hence, parallel computing may be needed in order to find optimal solutions within reasonable time.

Intellectual Property Protection and Licensing

In its current implementation, our software framework partially supports protection of intellectual property as required by our industrial partners. Specifically, product designers who wants to use the web GUI to design offshore cranes will have no knowledge of the inner workings of the black box server-side CPT. Designers can also write their own customised client software to connect to the CPT, such as the Matlab crane optimisation client we have developed concurrently (Hameed et al., 2016).

The CautoD solution, on the other hand, consists of the AIPO module and its GA library and does not have a GUI frontend yet, which would have enabled black box usage. Thus, in it current state of implementation, users must modify the code directly to obtain crane design results, and intellectual property is therefore not protected. We describe the possibility of extending the AIPO module with a web GUI in the next section.

For licensing, we have currently developed a username/password-protected version of the web GUI, which uses the WS/JSON interface. This can easily be extended to the HTTP/JSON interface as well. Using usernames/passwords enables the owner of the software to license usage of the CPT server on a time-limited basis. We may implement licensing for the CautoD solution, as well as more advanced licensing mechanisms, in future work.

AIPO GUI and Added Functionality

Product designers are experts in their own professional domains but will often lack prior knowledge of AI or programming. It is therefore essential to enable these domain experts to use our software framework while having to acquire as little as possible of this knowledge. Accordingly, we plan to develop a GUI for configuration of the AIPO module, possibly absorbing the functionality of the existing web GUI in Figure 3. This has several advantages. First, the code and intellectual property of the

AIPO module and related AI algorithms can be hidden and used as a black box. Second, the GUI could be used for configuration purposes, such as defining and using new objective functions or setting parameters or weights of a set of predefined objective functions. Moreover, such a GUI should enable the user to choose which AI algorithm should be used for optimisation and set its parameters. Third, the GUI culd be used for visualisation, not only of the load chart, but of other relevant design aspects, thus enabling the product designer to quickly obtain an overview of the proficiency of the design and possible tradeoffs. Fourth, the GUI could let the user investigate existing product designs stored in a library, and possibly use these as a starting point for optimisation. Finally, the GUI can provide an intuitive and user-friendly method for doing all of the above when compared with text-based alternatives and direct programming of the software.

ACKNOWLEDGEMENTS

The SoftICE lab at NTNU in Ålesund 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 (RFF) Midt-Norge and the Research Council of Norway through the VRI research projects *Artificial Intelligence for Crane Design (Kunstig intelligens for krandesign (KIK))*, grant no. 241238 and *Artificial Intelligence for Winch Design (Kunstig intelligens for vinsjdesign (KIV))*, grant no. 249171.

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., Osen, O. L. and Pedersen, B. S. (2015), A computer-automated design tool for intelligent virtual prototyping of offshore cranes, Proceedings of the 29th European Conference on Modelling and Simulation (ECMS'15), pp. 147–156.

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.

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.

Chaves, O., Nickelsen, M. and Gaspar, H. (2015), Enhancing Virtual Prototype in Ship Design Using Modular Techniques, *Proceedings 29th ECMS* .

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., Hatledal, L. I., Sanfilippo, F., Asoy, V., Zhang, H. and Schaathun, H. G. (2015), Virtual prototyping system for maritime crane design and operation based on functional mock-up interface, OCEANS 2015-Genova, IEEE, pp. 1–4.

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.

Erbatur, F., Hasançebi, O., Tütüncü, I. and Kılıç, H. (2000), Optimal design of planar and space structures with genetic algorithms, *Computers & Structures* 75(2), 209–224.

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.

Hameed, I. A., Osen, O. L., Bye, R. T., Pedersen, B. S. and Schaathun, H. G. (2016), Intelligent computer-automated crane design using an online crane prototyping tool, Proceedings of the 30th European Conference on Modelling and Simulation (ECMS'16). Accepted for publication.

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 visualising the workspace of offshore cranes, ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers, pp. V001T01A012–V001T01A012.

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

He, B., Tang, W. and Cao, J. (2014), Virtual prototyping-based multibody systems dynamics analysis of offshore crane, *The International Journal of Advanced Manufacturing Technology* 75(1-4), 161–180.

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.

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.

Mujber, T., Szecsi, T. and Hashmi, M. (2004), Virtual reality applications in manufacturing process simulation, *Journal of materials processing technology* 155, 1834–1838.

Narayan, K. L., Rao, K. M. and Sarcar, M. (2008), *Computer aided design and manufacturing*, PHI Learning Pvt. Ltd.

Park, H.-S. and Le, N.-T. (2012), Modeling and controlling the mobile harbour crane system with virtual prototyping technology, *International Journal of Control, Automation and Systems* 10(6), 1204–1214.

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., Zhang, H., Rekdalsbakken, W. and Pettersen, K. (2015), A wave simulator and active heave compensation framework for demanding offshore crane operations, Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on, IEEE, pp. 1588–1593.

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.

Schoning, L.-C. and Li, Y. (2013), Multivariable optimisation of a Homogeneous Charge Microwave Ignition system, Automation and Computing (ICAC), 2013 19th International Conference on, IEEE, pp. 1–5.

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.

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.

Zhang, J., Zhan, Z., Lin, Y., Chen, N., Gong, Y.-j., Zhong, J.-h., Chung, H., Li, Y. and Shi, Y. (2011), Evolutionary computation meets machine learning: A survey, *Computational Intelligence Magazine, IEEE* 6(4), 68–75.

AUTHOR BIOGRAPHIES

ROBIN T. BYE[6] 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 NTNU in Ålesund (formerly Aalesund University College) 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 NTNU in Ålesund.

BIRGER SKOGENG PEDERSEN graduated from NTNU in Ålesund with a BE in automation engineering and is a former employee at ICD Software AS, during which time he worked on the research projects described in this paper. He is currently a MSc student of simulation and visualisation as well as a employed as a project manager in maritime technology at NTNU in Ålesund.

IBRAHIM A. HAMEED has a BSc and a MSc in Industrial Electronics and Control Engineering, Menofia University, Egypt, a PhD in Industrial Systems and Information Engineering from Korea University, S. Korea, and a PhD in Mechanical Engineering, Aarhus University, Denmark. He has been working as an associate Professor at NTNU in Ålesund since 2015. His research interests includes artificial Intelligence, optimization, control systems and robotics.

HANS GEORG SCHAATHUN Hans Georg Schaathun graduated from the University of Bergen with cand.mag. in 1996 (Mathematics, Economics, and Informatics), cand.scient. in 1999 (Industrial and Applied Mathematics and Informatics), and dr.scient. in 2002 (Informatics – Coding Theory), all from the University of Bergen, Norway. He was a lecturer in coding and cryptography at the University of Bergen 2002 and a postdoc in 2003-2006. As a lecturer and senior lecturer in computer science at the University of Surrey, England 2006-2010, his research focused on multimedia security including applications of coding theory and steganalysis using machine learning. He joined Aalesund University College (now NTNU in Ålesund) and became a professor in 2011. His current research focus is software engineering and pedagogy.

[6]www.robinbye.com