

**OMAE2016-54448**

## PARAMETERIZATION AND VISUALIZATION OF MARINE CRANE CONCEPT DESIGN

**Yingguang Chu**

Norwegian University of Science and Technology  
Aalesund, Norway

**Yuxiang Deng**

Norwegian University of Science and Technology  
Aalesund, Norway

**Birger Skogeng Pedersen**

Norwegian University of Science and Technology  
Aalesund, Norway

**Houxiang Zhang**

Norwegian University of Science and Technology  
Aalesund, Norway

### ABSTRACT

The paper presents a flexible and lightweight fast prototyping approach for marine crane concept design based on WebGL. The goal is to develop a user-friendly tool that facilitates the communication between designers, system engineers and customers during the early product and system design stage. In this study, we focus on the design of the mechanical parts of the crane in addition with the actuation systems, typically the hydraulic power systems. The degrees of freedom, main geometric dimensions of the crane links and joints, and the configurations of the actuators can be evaluated according to the requirements of workspace, load capacity, etc. The implementation and user interface of the designer tool are developed based on the application of WebGL. In this way, it also enables remote communications between departments in different locations over the web. The development of the parameterization and visualization tool will bring in the customer directly into the product and system design process, thus to improve the work efficacy and reduce the overall time and cost. The results from the designer tool also provide primary information and data for later design processes and simulations in the virtual prototyping environment.

**Keywords:** marine crane, fast prototyping, workspace, load capacity

### INTRODUCTION

Nowadays, marine crane design is characterized by high customization, high complexity, high price and low production volume. For crane development companies, it requires a cost- and time- efficient

solution to survive with a tight profit margin in the market. Poor design practice in early design stage affects all the follow-up processes, which can be devastating.

Mechanical design usually starts with collecting the design requirements from the customers, and thereafter defining the design specifications. The interpretation from the customer requirements to design specifications is particularly crucial for the decision making. Traditionally, direct interacting with Voice of the Customer (VOC) is carried out by the marketing department, lacking the roles of designers and engineers [1]. As a consequence, designers and engineers have to continuously improve their work in order to reach the optimum trade-offs between performance and cost.

Over the last few decades, computer-aided modelling and simulation tools have proved to be very useful for design verifications, testing and analysis, exchanging data and information, product promotion and presentation, etc. Virtual Prototyping (VP) has been a hot research topic in many industries, in particular, the aerospace and aircraft, and automobile industries [2] [3]. Given a VP platform that can cope with the inputs and outputs from all the interested parties, the product development process would become more efficient and effective. Currently, VP applications in marine industry are found rather limited. Commercial simulation centers for marine crane operations, such as Offshore Simulation Center, K-sim by Kongsberg, and Vortex by CM Labs, are developed mainly for training purposes, which emphasize the virtual reality environment, but with limited physical system simulations. Our mechatronics lab with Norwegian University of Science and

Technology in Aalesund has been in close collaboration with the local industry partners on marine crane control, modelling and simulations [4] [5], and developing automated virtual prototyping systems [6]. Other attempts on developing multi-domain simulator related to offshore crane applications are found with parts of the whole picture implemented. Terashima presented a virtual plant of shipboard crane combines Computer Fluid Dynamics (CFD) with mechanical dynamics [7]. Ku developed a dynamics kernel for multi-body systems considering external hydrostatic and hydrodynamic effects for offshore crane operations [8].

From the author's point of view, the general common research challenges are highlighted as follows:

- Lacking a standard for interfacing between the heterogeneous simulation models, usually favored by different implementation tools.
- The complexity and flexibility of the models and sub-models suit to different purposes.
- The interfacing for integration and communication for visualizations including 3D animation and user interfaces.

The proposal of VP for marine crane system aims to bring the different parties together including the customer, designers, system engineers and even marketing and after sales departments [9]. Modelling of the interdisciplinary systems for simulations is the foundation for developing the virtual simulator. In particular, implementing the models for simulation purposes in respect to mechanical design is difficult. This is because a rigid body model built by current available software tools contains too many mechanical properties including the geometric dimensions, the mass and inertia, the constrains, and graphical properties for visualization, etc. The simplification process for simulation purposes can be tedious for presenting an initiative design concept. Many of these data must be obtained and transferred in many various ways and formats for later use.

In this paper, we present an approach based on the application of WebGL technology for marine crane concept design, specifically, the mechanical design of the crane structure. The objectives of the study include:

- To generate mesh models and geometric models for visualization and simulation directly on the web with a set of predefined parameters that determine the main geometric dimensions of the crane.
- To generate and visualize the workspace and load chart according to the joint constrains determined by the actuation systems.

The implementation of the proposed approach was illustrated through a knuckle boom crane, which is a typical offshore crane for lifting and transporting operations [10]. The topology and classification of the design specifications for the crane were defined. The main advantages of the web-based tool are the flexibility for design concept verification by defining

the main geometric dimensions, and the user interfaces through the web-browser for data exchange and information sharing. The generated models can be wrapped and directly used for simulations in the VP environment.

## METHODS AND TOOLS

3D solid models of the crane are often built in Computer Aided Design (CAD) software packages such as Solidworks, NX, CATIA, etc. FEA and static analysis can be achieved via these programs. However, to integrate mechanical design in real-time simulations is hard for the current CAD tools, partly because of a complete solid model contains all the design details, which are overwhelming for real time simulations. Simplifications of the 3D solid models must be done in order to perform the kinematic or dynamic simulations. But manual transcription is time consuming and error prone. During the concept design stage, in particular for new product development, it would benefit the latter design processes to be able to check the main geometrical and functional features by simulations and visualizations of the design.

The "Crane Designer" tool includes a Workspace Editor (WE) and a Mechanical Editor (ME) as shown in Fig. 1. The WE interprets the requirements from the customer including the crane type, work space and load capability into initiative technical specifications, i.e. the first level specifications, and send to the mechanical editor. After the main parameters of the links are defined, the designer can interact with the ME to modify the geometries of the parts. Other configurations, like the attachment positions of cylinder joints, winch and sheaves are determined at this step. Finally, the ME can export and wrap up the mesh model, kinematic properties and mass properties of each link, to be used in simulations for design verification.

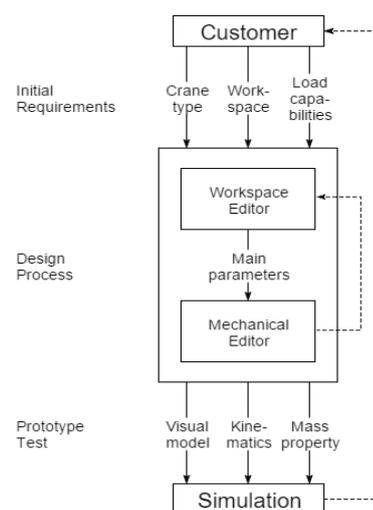


Fig. 1. The designer tool for fast prototyping of marine crane

The mesh models are generated by creating vertices and faces with JavaScript from a very low

level in WebGL. The mass and inertia properties are then calculated by the total volume of the mesh model in fair approximations given a material density. These properties can also be set manually. The crane workspace and load chart can then be computed by defining the joint constraints, and visualized directly in the browser. In order to provide the flexibility of the proposed approach for model reuse and modification, the specifications of defining a parametrical model in WebGL are divided into two levels. The first level defines the kinematic parts, i.e., the link bodies and joints. The second level is a collection of the parameters for defining the main geometries of each part. The appearances and volumes of the parts are then determined, as well as the mass and inertia properties. Mesh models are generated and provide the 3D representations of the parts.

Both editors utilize the three.js WebGL JavaScript library [11] to visualize the parts of the crane. The various components are defined as three-dimensional objects of a three.js scene. Objects can be hierarchically added to each other, forming a “parent-child” relationship, where a child’s position and orientation is defined within the frame of its parent. In this way, the objects inherit the transformations of their parent(s) recursively, and we can re-position a child object by transforming its parent. Three.js creates a HTML canvas element [12], on which the projections of a scene are drawn. Other, two-dimensional graphical features, like the legends of load charts (discussed in the next section) are drawn using another HTML canvas element with custom drawing functions. For parameter input dialogs, a lightweight user Interface, called dat.gui is utilized [13].

## THE WORKSPACE EDITOR

When designing a new crane, the user will firstly decide a crane type. This means specifying the degrees-of-freedom (DoF) and the joint types being rotational or prismatic. This process can be simplified by providing the user with some initiative suggestions. For instance, one alternative is a knuckle boom crane with 3-DoF constrained via three rotational joints. Given the crane type, the WE generates the workspace provided with some initial parameters. This is a three-dimensional space constrained by the possible positions of the crane’s end effector that determined by the crane parts and the joint limits. These positions are computed through an iteration process of each joint, i.e., re-positioning its corresponding joint from lower to upper limit sequentially, and computing the end effector position at each iteration step. For example, a crane with two rotational joints in a 2D plane yields 4 rotation operations, as illustrated in Fig. 2. Each joint is firstly rotated from its lower to upper limit, and vice-versa. The numbered arrows illustrate one of the alternative sequence. At each rotation step, the cranes end effector coordinate is computed. Each coordinate is saved as a vertex, making up a mesh which is added

to the Three.js scene, thus visualizing the crane workspace in the editor.

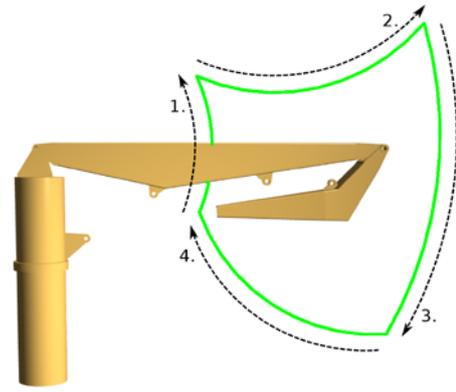


Fig. 2. Generate the workspace of a 2-DoF crane

For cranes with more than two rotational joints, the algorithm needs to be more “thorough” in order to identify the boundaries of the workspace. This means that we want to iterate not only through each rotational operation, but through each combination of the rotational operations. For example, a crane with 3 rotational joints in 3D space needs a total of 18 rotation operations, as shown in Fig. 3.

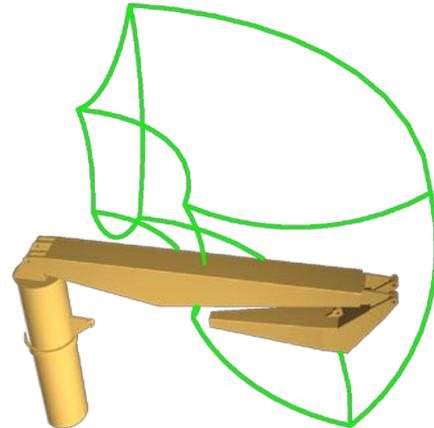


Fig. 3. Generate the workspace of a 3-DoF crane

To define the payload capabilities of the crane, the user can variate a “payload torque” of the joints, hereby termed as  $\tau$ . These values will later be used to compute the required capabilities of the crane cylinders. To compute the payload capabilities of the crane, a process of permuting through the possible joint configurations is started. For each permutation, the maximum payload is computed considering the maximum “payload torque”. Thus, the maximum payload is computed considering the “payload torque” of each joint.

The formula for the maximum “payload torque” of a joint  $n$  is calculated by the following formula:

$$\tau_n = r_n \times F_n \quad (1)$$

where  $r$  represents the arm from the joint  $n$  to the payload position, while  $F$  represents a force produced by the mass and gravity of the payload calculated, e.g. in the vertical direction:

$$F = m * g \quad (2)$$

For simplicity, we allow each joint to have a torque defined for only one of its rotational axes. The formula for computing the maximum payload  $m$  from maximum payload torques about an axis  $a$  considering a joint  $n$  can be obtained:

$$m_n = \tau_{An} / (r_{An} * g) \quad (3)$$

Once the maximum payload has been computed considering each joint, the smallest value is accepted as the maximum payload of the crane. The payload capabilities are computed and illustrated as the “load chart”, which splits the entire workspace into areas of different payloads. Each area is assigned a color, which represents the maximum payload scale the crane can hoist in that area, as shown in Fig. 4. By varying the torque limitations, the payload capabilities of the crane change. The “load capacity chart” is re-computed and re-visualized. By clicking and dragging the joints of the crane, the distance between the joints, i.e., the boom length, can be modified. Similarly, the joint limits can also be clicked and dragged directly in the editor. This alters the shape of the workspace and re-compute the load chart in real-time, thus providing the user with an easy way of “molding” the crane workspace and load chart.

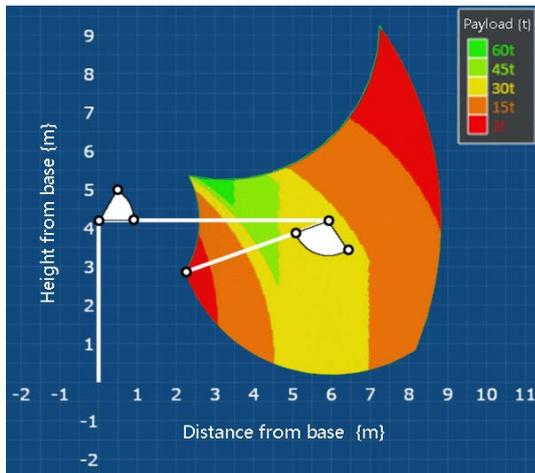


Fig. 4. Screenshot of a knuckle boom crane in the Workspace Editor

## THE MECHANICAL EDITOR

The ME is a 3D scene editor based on Three.js JavaScript library/API using WebGL to display 3D computer graphics in browser. A parametric component library containing a variety of basic components (e.g. main bodies, joints, cylinder attachments) is pre-defined. The components in the library are 3D mesh

models composed of vertices and faces. The arrangement of each component’s vertices is controlled by a set of parameters, which means, by manipulating those parameters, the shape of each component can be modified parametrically. For the purpose of fast prototyping, the mesh models should be simplified, but they should retain the sufficient features for both 3D visualization and dynamics simulation for the future stages.

Through selecting and assembling the components from the library, the designer can easily build a parametric link with desired configurations. Then different types of cranes can be built by varying the permutations and combinations of different links. For instance, a typical knuckle boom crane consists of three links, as shown in Fig. 5. Thus, in this tool, a link will be regarded as the basic unit for both editing and exporting. A link consists of several components, which could have components of their own as well.

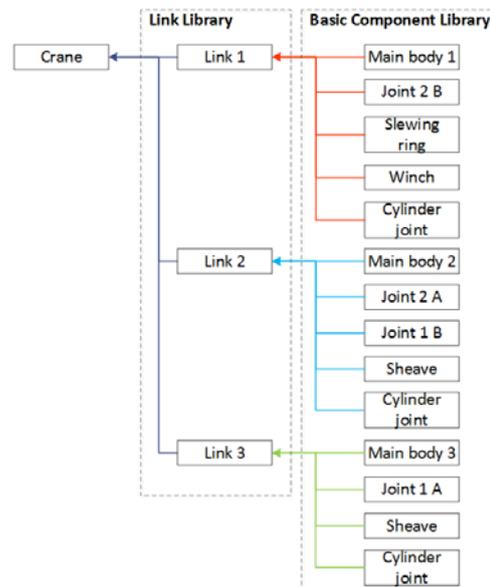


Fig. 5 Decomposing of the crane mechanical parts

Furthermore, we take one boom link of the knuckle boom crane as example to describe how a body part is modeled. The boom link is decomposed into 5 basic components: one main body, two cylinder attachment parts and two rotational joints. By coupling the dependent parameters of the basic components, the extracted parameters of the links are exposed to the users as shown in Fig. 6. The parameters in the first group are utilized for altering the shape of the main body. The second and third groups define the positions of two cylinder attachment parts. For the attachment position of the second cylinder, the Y-axis value is dependent on the X-axis value, because it should be always attached to the bottom of main body. Similarly, with the parameters in the last two groups, the users are allowed to modify the defined geometric features of the two rotational joints.

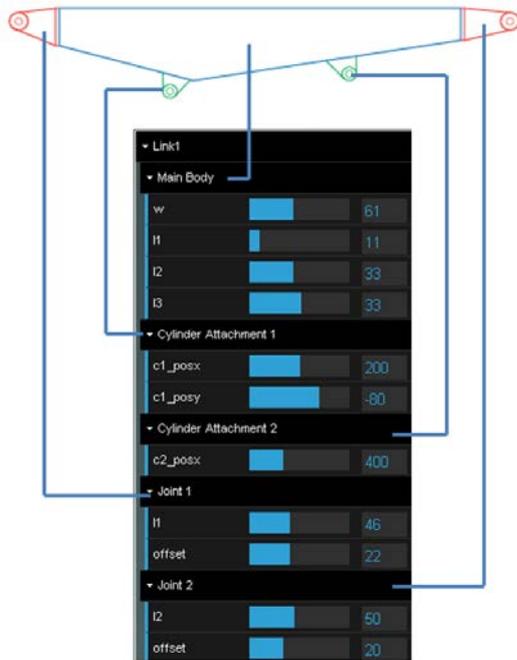


Fig. 6 Decomposition and parameterization of the boom link

A link or component consists of several Basic Component Mesh (BCM) models. A part of the program code of scripting a BCM model is presented. A BCM is defined by a “geometry” with a set of vertices and faces. The positions of the vertices are defined by the dimensional parameters. Three vertices then form a face. At last, the geometry is merged back to its upper level component or link, and disposed to avoid memory leak.

```
BasicComponentDemo = function (geo) {
  //Pre-defined dimensional parameters
  this.l1 = 0.5;
  ...
  bcm.rot = [0.12, 0, 0];
  bcm.pos = [0, 2.1, 0];
  //Function for generating the mesh
  this.generate = function () {
    //Create a temporary Geometry
    var geometry = new THREE.Geometry();
    //Generate vertices based on the parameters
    geometry.vertices.push(new THREE.Vector3(-
    this.l1, this.l2, this.t / 2));
    ...
    //Generate faces from the vertices
    geometry.faces.push(new THREE.Face3(i, i + a +
    1, i + a));
    ...
    //Define the BCM position and orientation
    geometry.rotateX(this.rot[0]);
    geometry.rotateY(this.rot[1]);
    geometry.rotateZ(this.rot[2]);
    geometry.applyMatrix(new
    THREE.Matrix4().makeTranslation(this.pos[0],
    this.pos[1], this.pos[2]));
    //Merge new geometry into former geometry
    geo.merge(geometry, geometry.matrix);
    //Dispose the temporary geometry
    geometry.dispose();
  };
};
```

When constructing a new link or component with more than one BCM, a new geometry is created in the first place. Then the BCMs are added and interrelated by defining their positions and orientations. The following program code illustrates how a link is generated, of which two BCMs are defined.

```
//Define the component or link geometry
var geometry = new THREE.Geometry();

//Define the first BCM
var bcm1 = new BasicComponentDemo(geometry);
bcm1.l1 = 0.5;
...
bcm1.rot = [0.12, 0, 0];
bcm1.pos = [0, 2.1, 0];
bcm1.generate();

//Define the second BCM
var bcm2 = new BasicComponentDemo(geometry);
bcm2.l1 = 1;
...
bcm2.rot = [1.57, 0, 0];
bcm2.pos = [0, 1, 0];
bcm2.generate();
...

```

For a pre-selected crane type, a 3D model with default parameters will be shown in the ME along with the WE. The main parameters of each link, e.g. the length, are adjustable via the WE. Other parameters, like geometric attributes and actuator positions will be exposed to the users through the GUI. By manipulating those parameters, the corresponding 3D model will be modified and re-rendered in real-time. This provides the users an efficient and intuitive way of interacting with the parametric models. Fig. 6 shows the screenshot of the implementation of the “Crane Designer” tool in the web-browser for designing the knuckle boom crane.

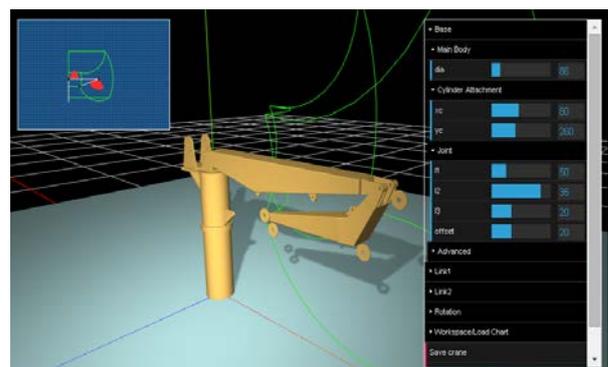


Fig. 7 Simulation of the knuckle boom crane in the web-browser by the “Crane Designer”

The products of each crane link can be saved after the evaluation of the workspace and load capacity of the design requirements. This packs in the mesh model, kinematic properties and mass properties specifications to a single data collection, which can be exported (as a .ZIP-file) and implemented directly into the virtual

prototyping framework, as shown in Fig. 8. The data is saved as .JSON-formatted objects, which enables the developers or other advanced users to read and interpret the crane data for other design and simulation practices. The .JSON-formatted file also provides the crane designer the data to be parseable for third-party simulators and modelling editors.



Fig. 8 The crane in the virtual crane simulator

## CONCLUSION

Integrating simulation of the mechanical design tools into the virtual prototyping process of product and system design is challenging due to the complexity of mechanical design and the interfacing and communication between the software tools. In the above sections, a flexible approach based on WebGL for marine crane concept design was presented. The topology and classification of the design specifications were described via designing a knuckle boom crane case. The parameterization and visualization tool rendered in the web-browser provided a flexible platform that brings together the customers and designers together in the early design stage for the product development. This improves the general efficiency of the product development and reduces the risk of re-work, thus saves time and cost.

The usability of the design tool is dependent on the pre-defined variables of the libraries. The editable parameters can be as many as needed, which increases the detail level for the design as well as the complexity of the modelling. Future work of the project includes the formalization of the component library and the GUI of the editors. In the case of 3D CAD models are provided and ready to use for simulations, an alternative of reusing the data for simulations is to export the CAD models as object files (OBJ) and then import to WebGL.

## ACKNOWLEDGMENT

The project work of this paper is financially sponsored by the Norwegian National Research Foundation (Innovation Projects for the Industrial Sector, MAROFF, 525564, and Centre for Research-based Innovation, SFI, 237929).

## REFERENCES

- [1]. K. Yang. (2007). "Voice of the customer: capture and analysis," McGraw-Hill Professional, New York.
- [2]. G. R. Bennett, (1997). The application of virtual prototyping in the development of complex aerospace products. *Journal of Aircraft Engineering and Aerospace Technology*, 69(1), pp.19-25.
- [3]. H. Y. K. Lau, K. L. Mak, and M. T. H. Lu. (2003). A virtual design platform for interactive product design and visualization. *Journal of Materials Processing Technology*, 139(1), pp. 402-407.
- [4]. Y. Chu, V. Æsøy, H. Zhang, "Integrated Multi-domain System Modelling and Simulation for Offshore Crane Operations," *J. of Ship Technology Research*, vol.1. 2015, pp. 36-46.
- [5]. Y. Chu, V. Æsøy, "A multi-body dynamic model based on bond graph for marine hydraulic crane operation," in *Proc. of the 34th International Conference on Ocean, Offshore and Arctic Engineering*, St. John's, Newfoundland, Canada, May 2015.
- [6]. R. T. Bye, O. L. Osen and B. S. Pedersen, "A computer-automated design tool for intelligent virtual prototyping of offshore cranes," in *proceedings of 29th European Conference on Modelling and Simulation*, May. 2015, Sofia, Bulgaria.
- [7]. K. Terashima, R. Ito, Y. Noda, Y. Masui, and T. Iwasa, "Innovative integrated simulator for agile control design on shipboard crane considering ship and load sway," in *Proc. of IEEE CCA*, Yokohama, Japan, 2010, pp.1293-1300.
- [8]. N. Ku, S. Ha and M. Roh. (2014). Crane modeling and simulation in offshore structure building industry, *Journal of Computer Theory & Engineering*, 6(3), pp. 278-284.
- [9]. Y. Chu, L. I. Hatledal, S. Ehlers, V. Æsøy and H. Zhang, "Virtual Prototyping for Marine Crane Design and Operations," *J. of Marine Science and Technology*, submitted, Nov. 2015.
- [10]. M. K. Bak and M. R. Hansen, "Analysis of Offshore Knuckle Boom Crane - Part One: Modeling and Parameter Identification," *Journal of Modelling, Identification and Control*, 34(4), pp.157-174, Jan. 2013.
- [11]. three.js, Available online at: <http://threejs.org/>, Mar. 2016.
- [12]. HTML/Elements/canvas, Available online at: <https://www.w3.org/wiki/HTML/Elements/canvas>, Mar. 2016.
- [13]. dat.gui, Available online at: <https://github.com/dataarts/dat.gui>, Mar. 2016.