An Object-Oriented Modeling Approach to Virtual Prototyping of Marine Operation Systems Based on Functional Mock-up Interface Co-simulation

Yingguang Chu[1]*, **Lars Ivar Hatledal**[1], **Vilmar Æsøy**[1], **Sören Ehlers**[2], **Houxiang Zhang**[1]
[1]Norwegian University of Science and Technology
Postboks 1517, 6025, Aalesund, Norway
Email: {yingguang.chu, laht, vilmar.aesoy, hozh}@ntnu.no
[2]Technical University of Hamburg
Room 4.008 Schwarzenbergstrae 95 (C) D-21073, Hamburg, Germany
Email: ehlers@tuhh.de

This paper presents an Object-Oriented Modeling (OOM) approach to model development of marine operation systems, specifically the hydraulic systems of marine cranes. Benefited from the rapid development of computation technology, many modeling and simulation techniques and software tools have proved to be very useful during the product and system development process. However, due to the increasing complexity of the physical systems, many challenges still exist regarding model flexibility, model integration, simulation accuracy, stability and efficiency. The goal of introducing OOM to complex dynamic systems is to provide flexible, effective and efficient models for different simulation applications. Previous work presented a virtual prototyping framework based on the Functional Mock-up Interface (FMI) standard. The advantage of using FMI co-simulation is that modeling and simulation of stiff and strongly-coupled systems can be distributed. As a result, the modeling tradeoffs between simulation accuracy and efficiency can be evaluated. The essential features of OOM and its application within dynamic operation system domain are highlighted through a case study. These features include model causality, model encapsulation and inheritance that facilitate the decomposition and coupling of complex system models for co-simulation. The simulation results based on the proposed virtual prototyping framework showed speedups in the computation efficiency at the cost of moderate accuracy loss.

Keywords: object-oriented modeling, virtual prototyping, marine crane operation, hydraulic, co-simulation

1 Introduction

Modern dynamic operation systems are characterized by high complexity, high customization, low production volume, high cost and short life cycle. Currently, the product and system design process of marine operation systems still follows the traditional way. Automated design approaches, so-called expert systems, have been proposed to support the decision making particularly during early design stages [1]. However, the impacts of expert systems in industry are rather limited. The reasons include that most of these systems are application dependent, which makes it difficult to develop or maintain generic design criteria. Moreover, the designers and engineers tend to be reluctant to adapt to such systems due to skepticism and conservatism [2].

Since the last few years, virtual prototyping (VP) technology has been developing rapidly in many industrial sectors. Simulation in virtual environment provide the users comprehensive and cost-efficient insights into the system behavioral characteristics. But model development of the complex multi-domain systems and handling of the simulation of these dynamic models is challenging. For example, modeling of marine operation systems involves such as mechanical design, kinematics and dynamics, hydraulic systems, and control algorithms. A plethora of software tools have been used for modeling and simulation of these various subsystems. On one hand, there will never be one single approach or software tool that can be perfect for the entire system. On the other hand, partitioning the complete system introduces new challenges for data exchange and communication between the simulators.

Recent advance in exchange of simulation models and VP has been significant. The automotive industry has been particularly innovative with projects such as the European Modelisar project which led to the establishment of the Modelica language [3] and the Functional Mock-up Interface (FMI) standard [4]. The FMI standard defines a tool independent standard to support both model exchange and co-simulation of dynamic models. With this shared format, component models can be developed using different

*Yingguang Chu is with the Department of Ocean Operations and Civil Engineering, NTNU, Norwegian University of Science and Technology, Aalesund, Norway, e-mail: yingguang.chu@ntnu.no.

tools depending on their disciplines and the designers preferences. These models are wrapped in as a so-called Functional Mock-up Unit (FMU) to be reused whenever and wherever it is needed. Model integration, or rather coupling of simulation, requires a separate tool or platform.

The FMI standard solved a long-existing challenge in simulation coupling, hence was quickly accepted by the academy and industry. To date, over 150 commercialized and open-source software tools support the FMI standard. A VP framework based on FMI co-simulation for marine crane system design and operations was proposed and presented [5]. In brief, the VP system for marine crane operation systems aims to solve the following problems:

1. To enable rapid simulation and visualization of concept design alternatives for new product and system development [6].
2. To provide an integrated tool-independent platform for the simulation and co-simulation of complex multi-domain dynamic systems, specifically real-time simulation for operations.
3. To allow for flexible model manipulation and reuse of models in different simulation scenarios.

Developing effective, efficient and flexible models of the dynamic systems lay the foundation the VP system. This paper introduces an Object-Oriented Modeling (OOM) approach to model development of dynamic operation systems. A component model is defined as an object described by its structure and behavior. The separation in modeling and abstraction of interfaces allows for flexible model manipulation and reuse. Further more, it requires less understanding of all the implementation details of a specific sub-model for integration, but only its interfaces to the other interrelated sub-models. Depending on the complexity level, different model implementations contain different sets of properties describing the behavior of the physical system. Usually, a more complex model implementation provides more realistic results close to the physical system, but it also requires more parameters and state variables for computation. As a result, the user interfaces become more complex and the simulation time increases, which is problematic for real-time simulation of stiff and strongly-coupled systems.

Model implementation of a hydraulic crane system is presented as a case study. Based on the proposed VP framework, the main issues with co-simulation of strongly-coupled systems were addressed, specifically the weighting between the simulation accuracy and efficiency. The requirement to real-time simulation for operations imposes an intractable challenge to the micro-steps of the simulators which decide the simulation accuracy and efficiency. Meanwhile, the convergence of the co-simulation process depends on the macro-step size of the simulation. In another word, divergence and poor accuracy can be minimized by taking small macro-step sizes, but also increases the computation time [7].

The rest of the paper is organized as follows: Section 2 discusses the available modeling approaches for dynamic operation systems from the OOM point of view. The chosen Bond Graph (BG) method can be seen as a special form of OOM technique for multi-domain system modeling based on the energy structure of the physical system. Section 3 describes model implementations of the hydraulic crane system, specifically the hydraulic power system of the actuators and model setup for co-simulation. Section 4 shows the simulation results based on different model implementations and comparison to FMI co-simulation. The benefits and drawbacks of using FMI co-simulation for strongly-coupled systems were discussed. At last, Section 5 concludes the work and outlines the future work.

2 Related Work on Model Development of Dynamic Operation Systems

VP is a broad concept that not only includes the modeling and simulation of the physical systems, but also embraces the simulation environment and user interactions. Before the advent of VP, Model-based Design and System Engineering (MBD/MBSE) is defined as a systematic model-centric approach for complex interdisciplinary systems and systems of systems [8–10]. The Wymorian theory defines a model as which consists of two parts, i.e., the ”inside” of the system described by its states, and the ”outside” of the system defines the interfaces to its outer world [9]. The separation of the states and interfaces of the model is one of the most important features shared by several other definitions. Modeling is simplification and representation of the physical system behavior. Virtual prototypes, or digital models representing the physical system should contain all properties and reflect all the aspects that physical prototypes could provide.

The architecture of a component model based on the OOM paradigm for simulation of dynamic systems is shown in Fig. 1. The behavior of an object is separated from its structure. Specifically, the structure model defines the representation and properties, including the parameters and user interfaces for model manipulation. The behavior model describes the dynamics of the physical system and the simulation interfaces for model interaction. This includes the parameters, state variables, input and output of the component model to others where any physical connection exists. Model classification is established based on the different complexity levels of modeling in describing the behavioral characteristics of the system. For one component model, several implementations share the same set of input and output to its outside world, but contain different parameters and state variables.

OOM originated from software model development and resulted in the establishment of the Unified Modeling Language (UML) in the 1990s [11]. Later, SysML as an extension of UML, is a graphical modeling language that provides a general purpose modeling language to support specification, analysis, design and verification of dynamic systems [12]. Known as the four pillars of SysML: the requirements, the behavior, the structure and the parametric, in total nine diagrams are used to reflects the various aspects of a system model. Recently, the INTO-CPS project has developed INTO-SysML models for co-simulation of a cyber-physical system [13]. The INTO-SysML profile customizes SysML

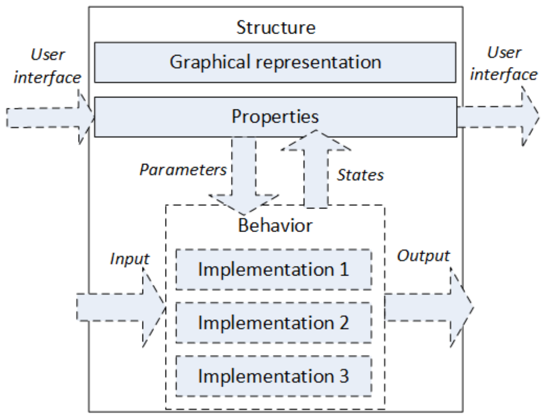


Fig. 1. Model architecture of an object-oriented component model

for architectural modeling in a setting of multi-modeling and FMI co-simulation. It introduced specializations of SysML blocks to represent different types of CPS components, constituting the building blocks that enable a hierarchical description of the CPS architecture. SysML is a powerful, but formidable, technical language to master, especially for the designers and system engineers [14]. Its extensions, such as INTO-SysML, could be a better alternative for modeling if matured. Up to date, the application of OOM within dynamic operation systems is still at the initiative stage. The purpose of introducing the OOM method for dynamic operation systems is to facilitate the modularization and integration of model development. Thus, all different users can form an understanding on the same matter (model) and contribute their expertise on one sub-system without worrying about the details of others.

More recently, the Modelica language has drawn the attention of both the academic and industrial users. Modelica is a non-proprietary, object-oriented, equation-based language designed for the convenient modeling of complex physical systems [3]. Basic models of physical components are defined in a declarative manner by their constitutive equations and connected with the outside world without implied causality. This makes the description of the physical systems more flexible and understandable than using causal or block-oriented modeling languages. It is also easier to use than directly writing simulation code using procedural languages like C or FORTRAN [15]. Complex models can be built by connecting the basic models where the physical connections exist. Since the models are written in terms of generic differential-algebraic equations, components that define interactions between different physical domains can be combined without any restriction.

Before the establishment of OOM and Modelica, the BG method was one of the most effective approaches for modeling multi-domain dynamic systems. It describes the physical system by identifying its energetic structure and represents the energy flow using generalized variables: the effort e, the flow f, the displacement q and the momentum p. For example, in the hydraulic domain these variables represent the pressure P, the volumetric flow $\dot{V}$, the volume V, and the pressure momentum p_p. As shown in Fig. 2, the so-

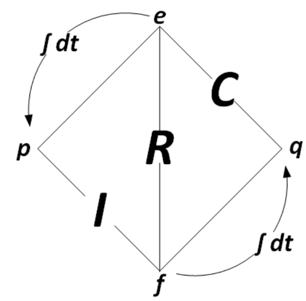


Fig. 2. Tetrahedron of state of the three single-port elements

called tetrahedron of state depicts the interrelations of the three basic single-port BG elements, including the inertia $\boldsymbol{I}$, the capacitor $\boldsymbol{C}$ and the resistor $\boldsymbol{R}$ [16]. Multi-ports BG elements, special types of BG fields and equation-based iconic diagrams for specific applications can be created following the same principles. These BG elements and iconic diagrams provide generalized forms for representations of the component models in different energy domains.

BG modeling can be seen as a special form of OOM modeling [17]. A Modelica BG library, in effect a translation of BG elements in Modelica has been presented by Broenink and Cellier [18, 19]. Later, an export filter for BG models implemented in a software tool called 20-sim was developed to generate Modelica code [20]. The generation of Modelica code from the existing basic BG elements and block diagram appears quite straightforward. As stated by Broenink, since BG method exists before OOM applied to physical system modeling, BG can be seen as a form of OOM approach as it shares the key features of OOM paradigm as follows:

1. A-causal modeling: Computation causality is not needed in modeling, but needs to be decided in compiling the model into computable code. A-causal modeling means that the equations are written as true equations that state the mathematical principles rather than determining the actual computational sequence. In this way, the interfaces of the sub-model can be defined by variables that are not committed to any role for computation. This is essential if the internals of the sub-models are to be completely encapsulated.
2. Encapsulation: The interaction with a model is only accessible via well defined interfaces with its ”outside” world. The use of the model is not constrained by the internal specifications. This allows for the reuse of the model or certain parts of model while keeping the rest untouched.
3. Hierarchy: A physical system is hierarchical in nature and can be decomposed into sub-systems. Thus, the model of a system is composed of sub-models that may contain lower level of sub-models as well.
4. Inheritance: Based on a hierarchical relationship, the properties of a sub-model are inherited by its upper class of sub-models and shared by the lower class of sub-models.

Modelica is a textual modeling language, while BG diagrams are equation-based graphical representations of the physical properties and phenomena. In this sense, neither Modelica nor BG provide easy comprehension for those who are not familiar with the modeling language or graphical diagrams. Modelica allows for more flexibility than BG for modeling. Currently, many Modelica-based software programs are in use such as SimulationX, MapleSim, JModelica, OpenModelica. For system engineers, graphical modeling tools take less effort to understand given well-defined model libraries. BG offers a convenient classification of models. The basic BG elements by themselves are graphical representations of the behavioral properties of the physics and dynamics. Component model representations can be customize-made using basic BG elements, special types of BG elements, iconic diagrams, block diagrams or combined.

In this paper, the BG method was used for modeling of the marine crane systems. As one of the benefits, comparison can be made easily between the simulation results based on integrated BG models and co-simulation based on the proposed VP system. Model implementation and handling was realized in 20-sim which provides the basic BG elements as well as domain model libraries. One model can have several implementations, which allows for the encapsulation of different alternatives of the behavior models. For example, model classification of a double-acting hydraulic cylinder is shown in Fig. 3, including two model implementations, namely the simplified and the extended. The representation of the component model uses iconic diagram and the implementation alternatives use BG elements indicating different properties of the behavior models. Construction of the behavior model of the crane system will be elaborated in Section 3.

The proposed OOM approach introduces model hierarchy and encapsulation for different applications. Separating the structure model from the behavior model allows for convenient model manipulation and reuse. Furthermore, co-simulation based on the FMI standard provides an effective solution to the long existing challenge with the interfacing and data exchange for simulation coupling, especially real-time simulation of complex stiff systems.

3 Model Development of the Knuckle Boom Crane

The considered physical system is a common type of offshore crane called Knuckle Boom Crane (KBC) as shown in Fig.4. It consists of a base seated on a pedestal and two booms connected by two rotational joints resembling the finger knuckles. The crane base and the booms are driven by the hydraulic power system via the slew bearing and the cylinders respectively.

The complete BG model of the kinematics and multi-body dynamics of the KBC is presented in [21]. The approach describes the multi-body crane using the Lagrange's equations in the Hamiltonian form. This derives the equations of motions using one single mass-inertia matrix, thus to avoid the intractable problems with algebraic loops and derivative causalities. In this way, the models of the hydraulic system can be integrated directly by a transformation of the cylinder piston velocity to the crane boom angular velocity. However, this approach is not desired when modeling of the hydraulic system and the crane bodies needs to be handled separately, which is the case in the proposed VP system based on the FMI co-simulation. Alternatively, this can be done by introducing artificial spring-damper mechanisms to enforce constraints between the actuators and the crane, and also the joints of the crane links. However, it will result in large forces and high frequency vibrations at the strongly-coupled constraints which is problematic for simulation. Consequently, it necessitate small simulation time steps for numerical computation. The key for this approach is thus to find the lowest stiffness that yields the longest possible time steps, consequently the shortest simulation time. Nevertheless, to achieve real-time simulation is challenging and easily becomes impossible when the system gets complex involving more and more objects in the simulator [22].

As a case study for testing the performance of FMI co-simulation, the paper presents modeling of the hydraulic system and the crane in one degree of freedom: that is the main boom and the boom cylinder of the KBC. Fig. 5 shows a block diagram of the model architecture. A PI-controller is used to control the volumetric flow into the cylinder in order to form a closed-loop control of the cylinder's movements, and consequently the crane boom. The causality of the cylinder model prefers to be sending the cylinder velocity out and receiving the acting force from the crane boom back. The FMUs for exportation is imploded at the sub-system level instead of component level. This is simply for the convenience of exporting and coupling of co-simulation FMUs.

3.1 Behavior Model of the Hydraulic Cylinder Implementation 1 - Simplified

The hydraulic cylinder converts hydraulic power into translational mechanical power through pressurized liquid in the cylinder chambers and the piston-rod connected to the crane boom via a constraint. Due to the inherent nature of fluid dynamics that pressurized liquids are nearly incompressible, solving the dynamic equations requires very small time steps, hence long simulation time. For real-time simulation of crane operations, a simplified behavior model of the hydraulic system is usually preferred. Assuming that the liquid being incompressible, the behavior of the hydraulic cylinder can be simply represented by the conversion of the hydraulic power to the mechanical translation power. The first BG model implementation representing the simplified hydraulic cylinder is shown in Fig. 6.

According to the preferred causality, Eq. 1 holds for the pressure at the piston side P_a. The pressure at the rod side P_b is given by the return line pressure represented by an effort source.

$$P_a \cdot A_a = P_b \cdot A_b + F \tag{1}$$

where A_a, A_b denotes the area of the piston at the either side, F is the external force acting on the cylinder piston.

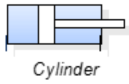
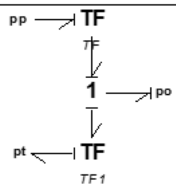
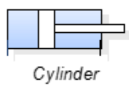
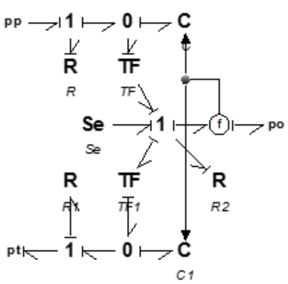
Model Implementation	Model Representation: Icon Diagram / Bond Graph	Interface: Input & Output / Causality	Interface: Parameters & Variables	Model Behavior: Variables {unit}
Double-Acting - Simplified		pp, pt, po	Diameter d	$A = \frac{\pi \cdot d^2}{4}$ $F = A \cdot P$
		pressure in, force out		Force {kN}
Double-Acting - Extended		pp, pt, po	Density ρ Inlet Diameter d_i Discharge Coefficient cd Bulk Modulus β Initial Pressure P_0 Piston Diameter d_{pstn} Rod Diameter d_{rod} Friction Factor f Dead Volume V_0 Initial Displacement x_0 Stroke s Bumper Stiffness k Bumper Damping Factor c	$A = \frac{\pi \cdot d^2}{4}$ $\dot{V} = cd \cdot A \cdot \sqrt{\frac{2 \cdot \lvert P \rvert}{\rho}} \cdot sgn(P)$ $V = x \cdot A \subseteq [V_0, (x_0 + s) \cdot A]$ $P = \frac{\beta}{V} \int \Delta\dot{V} \cdot dt$ $F = A \cdot P$ $F_f = f \cdot v$ $F_b = \begin{cases} -k \cdot (\delta - x) - c \cdot v, x < \delta \\ 0, \delta < x < \delta + s \\ -k \cdot (x - s - \delta) - c \cdot v, x > \delta + s \end{cases}$
		flow in, force out		Pressure Loss {bar} Pressure {bar} Force {kN} Friction Force {kN}

Fig. 3. Model classification of a double-acting hydraulic cylinder according to OOM

The modulated flow source represents the liquid flow from the pump into the cylinder. The volumetric flow rate is given by Eq.2.

$$\dot{V}_a = f \cdot \forall \cdot \omega_{cyl} \qquad (2)$$

where f is the flow gain of the pump flow, $\forall$ is the displacement of the pump per revolution, and ω_{cyl} is the speed of the pump in revolution per second.

Two transformers represent the power conversion at both sides of the piston. Since the liquid is incompressible, the velocity of the cylinder piston is given by Eq. 3.

$$v = \frac{\dot{V}_a}{A_a} \qquad (3)$$

Accordingly, the volumetric flow rate at the tank side of the cylinder piston can be derived by Eq. 4.

$$\dot{V}_b = v \cdot A_b \qquad (4)$$

3.2 Behavior Model of the Hydraulic Cylinder Implementation 2 - Extended

The fluid dynamics of the pressurized flow is established in the extended model, including the liquid compressibility, inertia of the flow flux, friction loss though the pipeline, and pressure drops over the valves and flow restrictions. The extended model describes the behavior of all components in Fig. 7, including the hydraulic power unit (HPU), pressure control valve (PCV), direction control valve (DCV), counterbalance valve (CBV), and the cylinder. The BG implementation of the extended hydraulic cylinder model is shown in

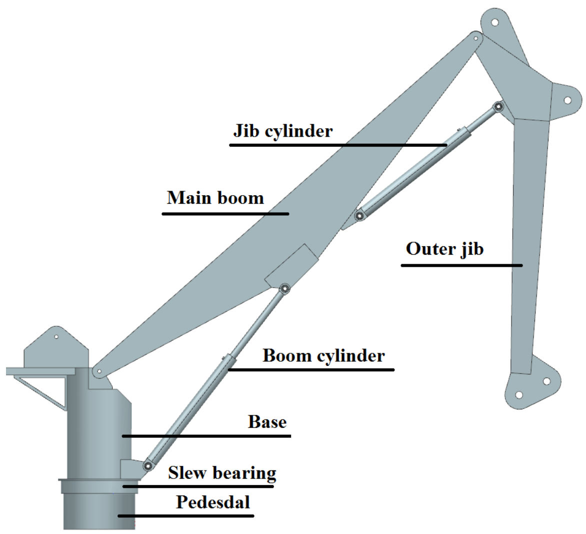


Fig. 4. The structure of the KBC

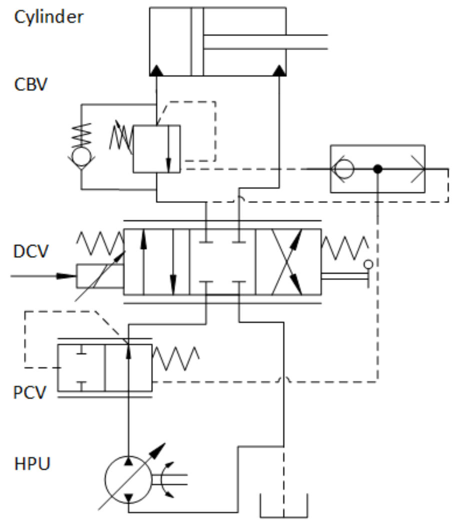


Fig. 7. Simplified hydraulic diagram of the KBC

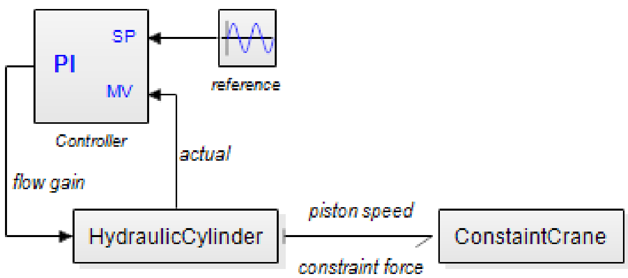


Fig. 5. Model architecture of the hydraulic cylinder, constraint and crane boom

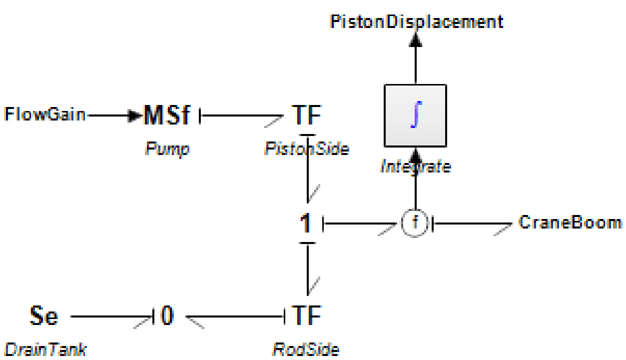


Fig. 6. Simplified behavior model of the hydraulic cylinder

Fig. 8. The sub-model shares the same interfaces as the simplified model implementation. In the simplified model, the flow gain is applied directly to the flow source representing the pump flow. In the extended model, valve control is used. The flow to the cylinder is controlled by a four-way-three-position DCV, a pressure compensator for the DCV and a load control valve (CBV) at the supply line when the cylinder is exposed to negative loads (when the piston velocity and loading force have the same direction). The constitutive equations describing the behavior of these components are presented as follows.

The power supply of the HPU is defined as a pressure compensated displacement pump. The flow rate from the pump is given by Eq. 5.

$$\dot{V} = \frac{P_{set} - P}{\Delta P} \cdot \dot{V}_{max} = \frac{P_{max} - P}{\Delta P} \cdot \forall \cdot \omega_{cyl} \tag{5}$$

where P_{set} is the set point pressure of the pump, ΔP is the pressure deviation from set point for full pump flow $\dot{V}_{max}$.

The pipe flow represents a lumped model of the fluid dynamics through the pipeline including the compressibility of the liquid, the inertia of the flow flux and the friction loss. The control volumes represents the compression of the small volumes of liquid between the hydraulic components.

The compressibility of the liquid is given by Eq.6.

$$\Delta P = \frac{B}{V} \cdot \Delta V \tag{6}$$

where B is defined as the bulk modulus the liquid and considered as constant, V denotes the original volume of the liquid which can be the physical volume of the pipe segment or the control volumes, ΔV represents a small decrease in the volume of the liquid.

The expressions for the compressibility of the liquid in the cylinder chambers are slightly different because the original volumes of the liquid change when the cylinder piston moves. The pressures in chamber A and B are given by Eq.7 and Eq.8 respectively.

$$\Delta P_a = \frac{B}{V_{a0} + A_a \cdot (x_0 + x)} \cdot \Delta V_a \tag{7}$$

$$\Delta P_b = \frac{B}{V_{b0} + A_b \cdot (s + x_0 - x)} \cdot \Delta V_b \tag{8}$$

where V_{a0} and V_{b0} denote the dead volumes of the cylinder chambers, x_0 and s denote the initial displacement and stroke of the cylinder piston.

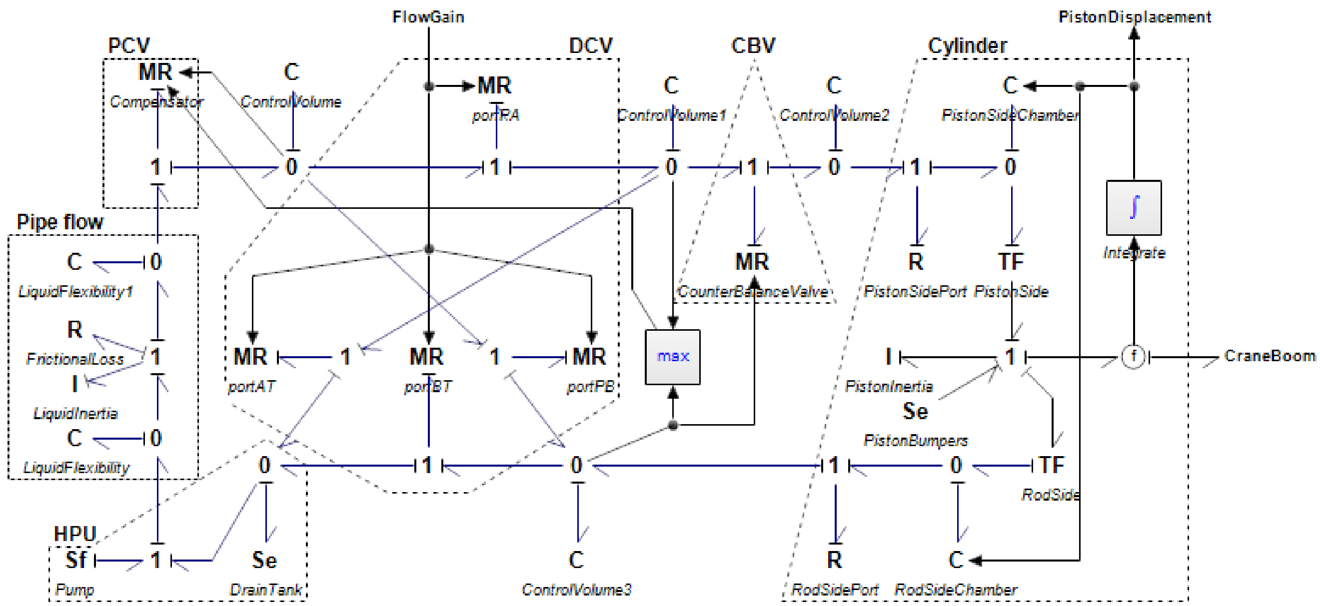


Fig. 8. Extended behavior model of the hydraulic cylinder

The inertia effects of the liquid flow through the pipe is analogous to the mass for a body in translation. Assuming that the flux of the liquid flow in the pipe moves as a rigid body, the Newton's law becomes as written in Eq. 9.

$$\Delta P \cdot A = (\rho \cdot A \cdot l) \cdot \frac{\ddot{V}}{A} \tag{9}$$

where ρ is density of the liquid, A is the effective section area of the pipe, l is the length of the pipe segment, $\dot{V}$ represents the volumetric flow rate through the pipe, hence, the velocity of the liquid slug becomes $\dot{V}/A$, and its acceleration becomes $\ddot{V}/A$.

The pressure momentum is defined as given by Eq. 10. As can be seen, the inertia coefficient is proportional to the mass density and length of the fluid flux, but inversely proportionate to the area.

$$p_P = \int (\Delta P \cdot A) dt = \frac{\rho \cdot l}{A} \cdot \dot{V} \tag{10}$$

The friction loss of the flow through a circular pipe is implied by a pressure drop related to the volume flow rate as written in Eq. 11.

$$\Delta P = \lambda \cdot \frac{l}{d} \cdot \frac{\rho}{2} \cdot (\frac{\dot{V}}{A})^2 \tag{11}$$

where λ is the friction factor of the liquid flow in the pipe and d is the diameter of the pipe.

The friction factor λ can be written as in Eq. 12 depending on the flow pattern of the pipe flow, whether it's laminar flow or turbulent flow.

$$\lambda = \begin{cases} \frac{64}{Re}, & Re < 2000 \\ \frac{0.25}{(\log(\varepsilon/(3.7 \cdot d) + 5.74/Re^{0.9})^2}, & Re > 2000 \end{cases} \tag{12}$$

where Re is the Reynold's number, ε is the absolute roughness of the pipe.

The liquid flow through any valve or port causes a pressure drop in the direction of the flow. The relationship of the pressure drop and the volumetric flow rate can be established following approximately the nozzle equation, as given by Eq. 13. The direction of the flow is implied by the sign of the pressure drop ΔP.

$$\dot{V} = C_d \cdot A \cdot \sqrt{\frac{2 \cdot \Delta P}{\rho}} \tag{13}$$

where C_d is the discharge coefficient of the nozzle, A denotes the open area of the nozzle.

As shown in Fig. 8, the PCV, DCV and CBV are represented by modulated resistors. The opening area of the valves is controlled by modulated signals. more specifically, the PCV opens when the pressure difference across the DCV is bigger than the compensation pressure. The opening of the four ports of the DCV are controlled by the flow gain from the PI-controller, which determines the opening area of each port. The flow through the CBV is dependent on the pressures at both sides of the cylinder piston. The CBV is fully open when hoisting the load, and partly open when lowering the load in case of loose control of the load and cause negative pressure in the cylinder.

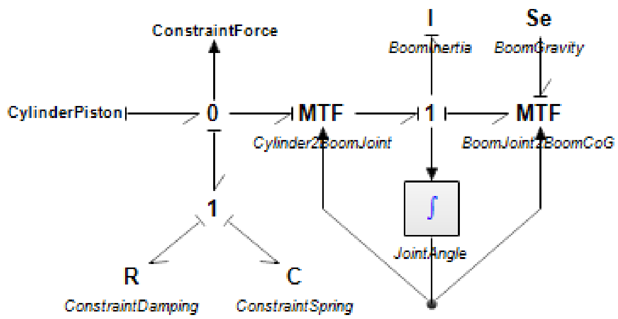


Fig. 9. BG implementation of the constraint and crane boom

The velocity of the cylinder piston can be calculated by its momentum according to Newton's second law.

$$p = \int F dt = m \cdot v \tag{14}$$

where m is the mass of the cylinder piston.

The effort source in the cylinder model represents the bumpers of the piston, which generates the reaction forces upon contact with the piston. The reaction force of the bumper is given by Eq. 15.

$$F = \begin{cases} -k \cdot x - c \cdot \dot{x}, & x <= -\delta x \\ 0, & 0 < x < s \\ -k \cdot (x - s) - c \cdot \dot{x}, & x >= s \end{cases} \tag{15}$$

where k and c are the spring stiffness and critical damping factor of the bumper. x and s are the displacement and stroke of the piston.

3.3 Behavior Model of the Constraint and Crane Boom

The reason of adopting the constraint joint is that the hydraulic system can be separated from the mechanical part. As a result, model development of the hydraulic system and the multi-body dynamics of the crane can be done using different tools. In the case study, we present comparison of two different alternatives of the crane boom model. The first one is a BG model developed in 20-sim, as shown in Fig. 9. The other is created and handled by a physics engine called AgX which will be described in Section 4.2 on co-simulation.

The constraint force connecting the cylinder piston and the crane boom is given by Eq.16. The compliance of the constraint needs to be tried out which can yield adequate small displacement gap δd and the longest simulation time steps in order to have the shortest simulation time.

$$F = -k \cdot \delta d - c \cdot \dot{\delta d} \tag{16}$$

As illustrated in Fig. 10, the transformation of the cylinder piston and the center of gravity of the boom to the joint can be derived, as written in Eq.17 and Eq.18.

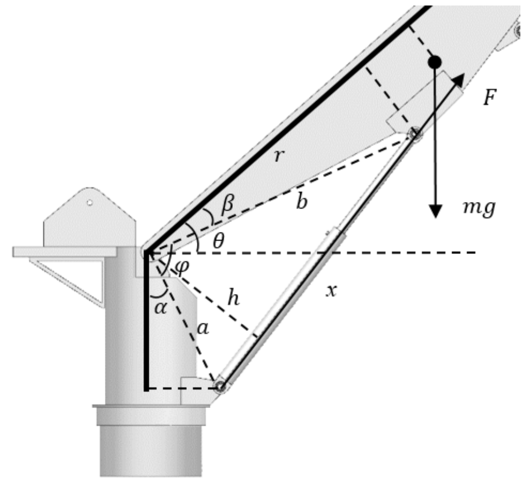


Fig. 10. Transformation from the cylinder to the boom joint

$$T_{cyl} = F \cdot h = F \cdot \frac{a \cdot b \cdot sin(\theta + \pi/2 - \alpha - \beta)}{x} \tag{17}$$

$$T_g = m \cdot g \cdot r \cdot cos\theta \tag{18}$$

The angular velocity of the boom $\dot{\theta}$ can be calculated from the angular momentum of the boom p_L, as in Eq.19.

$$p_L = \int \sum T dt = I \cdot \dot{\theta} \tag{19}$$

where I is the moment of inertia of the boom about the rotational joint.

4 Simulation Results and Discussion

The testing of the models was performed in two steps. Firstly, the simulation results based on the different behavior models of the hydraulic system were shown. Depending on the applications, models with different complexity levels are engaged. The priority of the criteria to the simulation performance vary for concept design, system analysis, and operation training. Secondly, the simulation results were compared with the results using FMI co-simulation. The co-simulation results also include two parts to show the flexibility of the VP system for model development. First is co-simulation of the two FMUs of the separated BG models of the hydraulic system and the crane boom. The other is co-simulation of the FMU of the hydraulic system and the crane boom built and handled by the physics engine AgX.

4.1 Simulation Results Based on Different Behavior Models of the Hydraulic System

Given a regular sine function as the reference of the piston displacement, the actual displacement based on different behavior models is shown in Fig. 11. The results indicate

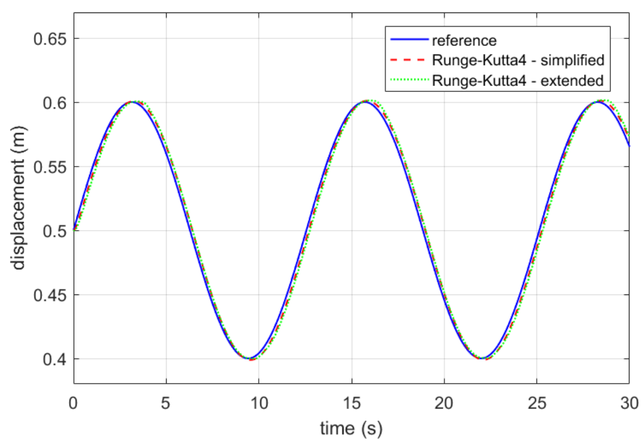


Fig. 11. Displacement of the cylinder piston based on different behavior model implementations

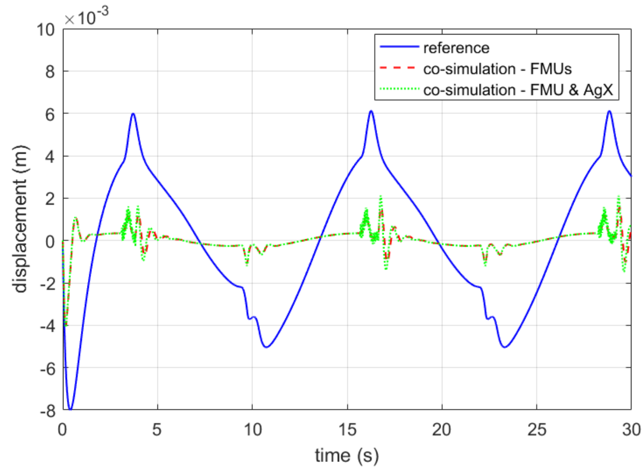


Fig. 12. Deviation of the cylinder piston displacement

Table 1. Simulation based on different behavior models

Test	RTF	RMSE
RK4 - extended	≈ 2.36	3.30e-3
RK4 - simplified	≈ 0.73	2.38e-3
VA - extended	≈ 0.067	3.80e-3
VA - simplified	≈ 0.0003	2.73e-3

Table 2. Simulation time based on FMI co-simulation

Test	RTF	RMSE
RK4 - FMUs by 20-sim	≈ 0.19	0.52e-3
RK4 - FMU&AgX	≈ 0.17	0.55e-3
VA - FMUs by 20-sim	≈ 0.03	0.61e-3
VA - FMU&AgX	≈ 0.05	0.62e-3

that both models are effective in simulating the behavior of the piston's movement. However, the simulation time is different depending on the behavior models and the chosen integration methods of the solver for computation.

The RTF and the Root-Mean-Square Error (RMSE) of the piston displacement using a fixed time step integration method (Runge-Kutta 4) and a variable step integration method (Vode Adams) are shown in Tab. 1. Real-time factor (RTF) is defined as the real world time for each step of simulation. For example, if the RTF is 0.1, each simulation step takes 0.1s. To ensure the requirement to the performance of real-time simulation, the RTF must be less than 1, and the smaller the better considering the added time for complex systems that involve many objects, as well as the time for data saving and sampling for plotting and visualization. It is obvious that the fixed step test fails to manage real-time simulation with the extended behavior model. It is also expected to fail with the simplified model when the complexity of the whole system increases.

4.2 Simulation Results Based on the VP System using FMI Co-simulation

The hydraulic cylinder connected to a rigid body via a constraint is a fairly simple system. However, it reveals several challenges regarding simulation of stiff and strongly-coupled multi-domain systems. The real-time performance of the simulation imposes an intractable challenge to the computation time steps (micro-step) which decides the simulation accuracy and efficiency. The convergence of the co-simulation process depends on the size of the co-simulation time steps (macro-step). In another word, divergence and poor accuracy can be minimized by taking small co-simulation time steps; however doing this also increases the computation time [23].

Co-simulation of the hydraulic crane was carried out in two steps. Firstly, two FMUs handled by 20-sim were implemented in the VP framework. Next, the sub-model of the crane boom with the constraint was replaced by a model built and handled by AgX which is the chosen tool of the proposed VP simulator for mechanical systems. This allows for more flexible and efficient modeling of strongly-coupled rigid body systems. Both sub-models can be developed using other software tools that support the FMI co-simulation standard. The macro-step of co-simulation is defined as 0.01s. The deviation of the piston displacement is minimal, as shown in Fig. 12. The reference of the deviation is from the complete BG model in 20-sim with the extended implementation of the hydraulic system. The RTF and RMSE from both of the two model tests suggests that the simulation time is reduced using co-simulation, as shown in Tab. 2. Furthermore, using a variable step solver takes less time than using a fixed time step solver.

The extended model implementation exhibits more behavioral characteristics of the hydraulic system than the simplified model implementation. For example, the volumetric flow rate of the cylinder piston side chamber is shown in Fig.

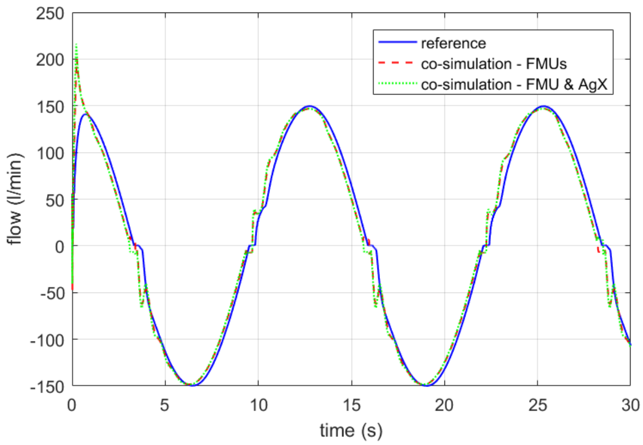


Fig. 13. Volumetric flow rate to the cylinder piston side

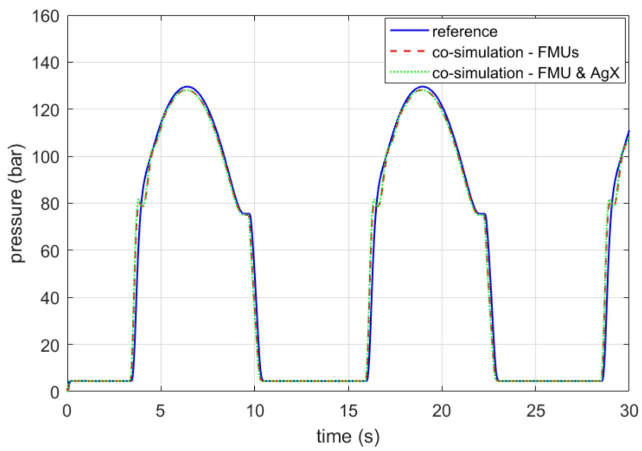


Fig. 14. Pressure at the cylinder piston side

13. The pressure at the cylinder rod side chamber is shown in Fig. 14. The reference of the result is from the complete BG model in 20-sim. The co-simulation results have shown minimal deviations to the reference that mainly because of different model setups, micro-steps of co-simulation FMUs and different solvers of 20-sim and AgX.

4.3 Discussion

The simplified model implementation showed valid results of the behavior of the cylinder regarding the movement of the cylinder piston. The simplified model implementation contains fewer parameters and state variables for modeling and computations, hence the decrease in model complexity and simulation time compared to the extended model implementation. For concept design and operation training applications, it maybe sufficient to use simplified models since real-time performance is placed at higher priority than the responding characteristics of every single component of the entire system.

The alternative solution, by describing the whole system by one single equation set is not as flexible and efficient for modeling of complex systems. Using co-simulation, it is possible to break the strongly-coupled system and simulate them separately. Using co-simulation one could include complex behavior models of the physical systems with reduced simulation time by increasing the macro-step at modest loss of accuracy.

5 Conclusions

This paper introduces an OOM approach for model development of marine operation systems. The BG method as a form of OOM modeling approach supports co-simulation of complex dynamic models using the FMI co-simulation standard. A component model is encapsulated provided the interfaces to the user defining its properties interfaces interacting to other component models. Modification and interaction with the FMUs is only accessible by their interfaces, hence the standardization of the interfaces to the FMUs is essential. These include the definition of parameters for model manipulation, state variables and the input and output variables for co-simulation. The behavior model of a component can include several implementations in different complexity levels tailored to different applications. For concept design and operation training applications, using a simplified behavior model requires less effort for model setup and also reduces the simulation time. However, an extended model provides more realistic characteristics of the dynamic system and simulation efficiency can be enhanced by FMI co-simulation at modest loss of accuracy.

The proposed marine crane VP system is developed based on the application of the FMI co-simulation standard. The current implementation uses the physic engine AgX to handle the rigid body dynamics of the crane. The component models of the hydraulic systems are developed using the BG method and handled by 20-sim. This distributive solution is the future direction considering the overall marine operation system is more complex than a single crane. The key challenge is ensure both accuracy and efficiency of the simulation due to the significant amount of exchanging data, not only between the communication between the FMUs, but also sampling and pulling the data for visualization.

The future work for model development includes defining the guidelines for model classification and categorization, for example, the RTF and the RMSE as the index for simulation efficiency and accuracy. However, establishing and maintaining robust component model libraries is an extensive undertaking that requires continuous contributions of different users. It is not always easy to avoid the stiff elements in the model for real-time simulation. Co-simulation provides a feasible solution for computation speedups at modest loss of accuracy. The overall simulation performance is also dependent on the co-simulation master [24]. Advanced Co-simulation Open System Architecture is an ongoing project that will address the questions this dissertation leaves unanswered [25]. It is dedicated to real-time system co-simulation to develop both a non-proprietary advanced co-simulation interface for real-time systems integration and an according integration methodology which shall be a substantial contribution to the international standardization, e.g., the FMI standard.

6 FUNDING

The project is financially sponsored by the Research Council of Norway (The Innovation Program for Marine Activities and Offshore Operations, MAROFF 235309, and Centres for Research-based Innovation, SFI 237929).

References

[1] Bye, R. T., Osen, O. L., and Pedersen, B. S., 2017. "A computer-automated design tool for intelligent virtual prototyping of offshore cranes". In Proc. ECMS 31st European Conference on Modelling and Simulation, Budapest, Hungary, pp. 147–156.

[2] Bak, M. K., 2014. "Model based design of electro-hydraulic motion control systems for offshore pipe handling equipment". PhD Thesis, University of Agder, Kristiansand, Norway.

[3] Modelica, 2017. *A Unified Object-Oriented Language for Systems Modeling - Language Specification*, version 3.4 ed. The Modelica Association, Linkoping, Sweden. See also URL http://www.modelica.org/.

[4] Blochwitz, T., Otter, M., kesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., Olsson, H., and Viel, A., 2012. "Functional mockup interface 2.0: The standard for tool independent exchange of simulation models". In Proc. 9th International Modelica Conference, The Modelica Association, pp. 173–184. See also URL https://www.fmi-standard.org/.

[5] Chu, Y., Hatledal, L., Zhang, H., Æsøy, V., and Ehlers, S., submitted in Nov. 2015. accccepted in Oct. 2017. "Virtual prototyping for marine crane design and operations". *Journal of Marine Science And Technology*.

[6] Chu, Y., Deng, Y., Pedersen, B. S., and Zhang, H., 2016. "Parameterization and visualization of marine crane concept design". In Proc. ASME 35th International Conference on Ocean, Offshore and Arctic Engineering - Volume 7: Ocean Engineering., American Society of Mechanical Engineers, p. V007T06A095.

[7] Bastian, J., Clauß, C., Wolf, S., and Schneider, P., 2011. "Master for co-simulation using fmi". In Proc. 8th International Modelica Conference, Dresden, Citeseer, pp. 115–120.

[8] Paterno, F., 1999. *Model-Based Design and Evaluation of Interactive Applications*, 1st ed. Springer-Verlag, London, UK, pp. 11–30.

[9] Wymore, A. W., 1993. *Model-based systems engineering*, Vol. 3. CRC press, Boca Raton, Florida, p. 58.

[10] Estefan, J. A., 2007. "Survey of model-based systems engineering (mbse) methodologies". *Incose MBSE Focus Group,* **25**(8).

[11] Rumbaugh, J., Jacobson, I., and Booch, G., 2004. *Unified Modeling Language Reference Manual*, 2nd ed. Pearson Higher Education, Boston, MA, pp. 3–11.

[12] Friedenthal, S., Moore, A., and Steiner, R., 2014. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, Waltham, MA, pp. 29–49.

[13] Amálio, N., Payne, R., Cavalcanti, A., and Woodcock, J., 2016. "Checking SysML Models for Co-simulation BT - Formal Methods and Software Engineering: ". In Proc. 18th International Conference on Formal Engineering Methods, ICFEM Tokyo, Japan, Springer International Publishing, pp. 450–465.

[14] Ramos, A. L., Ferreira, J. V., and Barceló, J., 2012. "Model-based systems engineering: An emerging approach for modern systems". *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* **42**(1), pp. 101–111.

[15] Pulecchi, T., Casella, F., and Lovera, M., 2010. "Object-oriented modelling for spacecraft dynamics: Tools and applications". *Simulation Modelling Practice and Theory,* **18**(1), pp. 63–86.

[16] Karnopp, D. C., Margolis, D. L., and Rosenberg, R. C., 2012. *System dynamics: modeling, simulation, and control of mechatronic systems*, 5th ed. John Wiley & Sons, Hoboken, New Jersey, pp. 37–48.

[17] Borutzky, W., 1999. "Bond graph modeling from an object oriented modeling point of view". *Simulation practice and theory,* **7**(5), pp. 439–461.

[18] Broenink, J. F., 1997. "Bond-graph modeling in modelica". In Proc. 9th European simulation symposium, Passau Germany, Oct, pp. 19–22.

[19] Cellier, F. E., and Nebot, À., 2005. "The modelica bond graph library". In Proc. 4th International Modelica Conference, Hamburg, Germany, pp. 57–65.

[20] Broenink, J. F., 1999. "Object-oriented modeling with bond graphs and modelica". *SIMULATION SERIES,* **31**, pp. 163–168.

[21] Chu, Y., and Æsøy, V., 2015. "A multi-body dynamic model based on bond graph for maritime hydraulic crane operations". In Proc. ASME 34th International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers, p. V001T01A010.

[22] Chu, Y., Æsøy, V., Ehlers, S., and Zhang, H., 2015. "Integrated multi-domain system modelling and simulation for offshore crane operations". *Ship Technology Research,* **62**(1), pp. 36–46.

[23] Viel, A., and Imagine, L., 2014. "Implementing stabilized co-simulation of strongly coupled systems using the functional mock-up interface 2.0". In Proc. 10th International Modelica Conference, Lund, Sweden, pp. 213–223.

[24] Mengist, A., Asghar, A., Pop, A., Fritzson, P., Braun, W., Siemers, A., and Fritzson, D., 2015. "An Open-Source Graphical Composite Modeling Editor and Simulation Tool Based on FMI and TLM Co-Simulation". In Proc. 11th International Modelica Conference, Versailles, France, Linköping University Electronic Press, pp. 181–188.

[25] Krammer, M., Marko, N., and Benedikt, M., 2016. "Interfacing real-time systems for advanced co-simulation - The ACOSAR approach". In CEUR Workshop Proceedings, Vol. 1675, pp. 32–39.