



Virtual prototyping of maritime systems and operations: applications of distributed co-simulations

Stian Skjong¹ · Martin Rindarøy² · Lars T. Kyllingstad² · Vilmar Æsøy³ · Eilif Pedersen¹

Received: 24 May 2017 / Accepted: 2 December 2017 / Published online: 8 December 2017
© JASNAOE 2017

Abstract

In this work, we demonstrate the use of co-simulation technology in the maritime industry through four relevant examples of applications based on the outcome of the knowledge-building project *Virtual Prototyping of Maritime Systems and Operations* (ViProMa). Increasing computational capabilities opens for extended use of simulators in the design processes. Even complex systems can now be analyzed at an early stage of the design process and even in real time using distributed simulation technology. We conduct an assessment of the need for co-simulation technology in the industry, present a short background in co-simulation technology, and provide a short summary of the major findings and deliverables in the ViProMa project (<http://viproma.no>). The four case studies presented in this work pinpoint different advantages of using co-simulations in the industry, such as combining different modeling and simulation tools, improving collaboration without revealing sensitive information by using black-box models, testing conceptual designs in a fast and consistent manner before initiating building processes, and verifying the interplay between hardware and software in the simulation environment in *hardware in the loop* (HIL) tests. All the case studies are simulated using the open source co-simulation software *Coral* developed in the project, using the *Functional Mock-up Interface* (FMI) standard, and the co-simulation software can be downloaded from the project's web site.

Keywords Co-simulation · Marine systems and operations · FMI · Hardware-in-the-loop · Virtual prototyping

1 Introduction

The maritime industry has been through a major shift in the last decades in terms of how research and development are conducted, and is still changing. Nowadays, computers are used for a great variety of tasks, from controlling complex operations such as an offshore vessel in dynamic operations to numerical analysis of complex system dynamics in research and development projects.

Research and development activities in the maritime industry are characterized by specialists working from different angles on joint projects using specialized computer

software to optimize designs before any prototypes are built. Since the costs of building a prototype are significant, often only a single one is scheduled, if one is even considered at all. This is especially the case when designing a new ship, where the prototype is in fact the ship delivered to the customer at the end of the project. This increases the expectations and sets high requirements for the specialists as well as the software, trying to realize the product properties specified by the customer in the short lifetime of the project. On the other hand, the project manager expects the project group to oblige the customer and deliver a satisfying product within the agreed-upon time frame in order to obtain a financial surplus rather than large financial penalties and a dissatisfied customer.

1.1 Research, development and collaboration

Despite the fact that the number of prototypes is nowadays significantly reduced in research and development projects, the iterative process of obtaining the best design more or less remains the same [1]. One major difference is that the

✉ Stian Skjong
stian.skjong@ntnu.no

¹ Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway

² SINTEF Ocean, Trondheim, Norway

³ Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Ålesund, Norway

iterations have moved from physical models in the workshop floors to mathematical models in the engineer's computer. Not only does this speed up the development, it also increases the number of opportunities for optimizing the design. It is now possible to integrate detailed subsystems of the design, from various engineering disciplines, into the optimization process. This enables the vendor to optimize the design for the product environment described by the customer, not only local optimization of the design itself. This also enables virtual proofing and validation of design concepts and reduces the risk of not meeting the design requirements. However, such optimization tasks introduce new and challenging problems. In the design process, often a variety of different specialized software tools are used and must be interfaced in the optimization process. If the optimization process is not done manually, which is time consuming and often the case, the optimization must be performed by some algorithm. The interface between the different software tools must be automated in order to save time, as well as being platform independent in order to remove unnecessary restrictions and enable connection to standard hardware such as control systems. Since combining such software and equipment from scratch in a generic way requires lots of resources, shortcuts are often taken, making it difficult to reuse the couplings between models and software tools in later projects.

When optimizing a design with respect to a specific working environment, external expertise is often needed. Such expertise can in some cases be found in-house, outside the project group, which makes collaboration simpler with regards to confidentiality issues. However, challenges arise when external expertise is obtained from outside the business. This is especially the case for shipyards, which rely on many different third-party vendors that deliver customized equipment for new vessels. In such joint projects, confidentiality is important and each contributor wants to keep its know-how hidden from the competitors which may take part in the same project. Earlier, this restricted the contributors' ability to work together in optimizing the total design. However, recent projects, such as the knowledge building project *Virtual Prototyping of Maritime Systems and Operations* (ViProMa), have made some contributions to enable new technologies that facilitate reduction in project costs and development time, as well as making collaboration for the greater good among competitors easier.

1.2 The ViProMa project

The Norwegian maritime industrial cluster is a world leader in developing complex, customized ships and offshore vessels for the global market—in particular ships for demanding and complex operations, where safety and environment are in focus. Industrial value chains for these products are

also very complex and inter-organizational, where logistics, communication and interface challenges must be handled. Project lead times are constantly decreasing, and mistakes or system malfunctions may cause fatal incidents, project delays and costs overruns. To remain a world leader, the knowledge building project ViProMa was initiated back in 2013¹ with high ambitions, even though it is a small joint research project including industrial partners from the Norwegian maritime industry, the research institute SINTEF Ocean and The Norwegian University of Science and Technology (NTNU).

Simulation of system performance will be even more important in the future. Installation of heavy subsea units at several thousand meters depth requires accuracy and control. Such operations demand tremendous power, interaction and timing. To meet performance, safety and environmental issues and cost targets, engineers must understand how the equipment will behave. Evaluating multiple design concepts can be done effectively using simulation tools, where trade-offs and many alternatives can be evaluated within a short time.

It is commonly accepted that new ship designs should be optimized with respect to operational performance rather than the performance of individual components and systems. In recent years, several large, international research projects have taken different approaches towards this goal: VRS (2001–2005) aimed to develop a virtual platform for design of ROPAX vessels by integrating design and simulation tools [2]. VIRTUE (2005–2008) addressed the hydrodynamic aspects of ship behaviour and worked on the integration of different computational fluid dynamics (CFD) tools to create “virtual test tanks” [3]. The recently completed JOULES project (2013–2017) focused on onboard energy systems, aiming to develop a holistic approach for simulating vessel energy grids [4]. Finally, the currently ongoing HOLISHIP project (2016–) aims to develop an integrated design software platform that takes into account the ship's entire life cycle [5].

Despite these efforts, the most existing simulation tools for maritime applications are developed for research and optimization of components and sub-systems. Some are designed for analysis of total energy system performance, such as DNV GL's *COSSMOS* [6, 7], TNO's *Geïntegreerde Energiesystemen* [8], and the University of Trieste's *Italian Integrated Power Plant Ship Simulator* [9], but these typically do not include operational aspects such as seakeeping, manoeuvrability, stability and capability assessment.

Hence, the main goal of the ViProMa project was to develop a framework for overall system design, allowing configuration of ships and verification of operational

¹ The project expired by the end of 2016.

performance as a part of the design process. A variety of general-purpose software and frameworks for system simulations exist, but before the project was launched, there were no commonly adopted simulation frameworks that supported total systems integration and analysis of operational performance. General software solutions for system simulations were not considered suitable for the purpose, mainly due to very time-consuming model development. Because decreasing project lead times require rapid model development and configuration with sufficient accuracy, of which general software is not capable, the ViProMa project aimed to close this knowledge gap.

1.3 Co-simulation

Co-simulation technology has been used for a few decades already and enables the use of *black-box models*: models compiled to machine code such that internal implementation details are hidden from users. Put differently, it allows in-house modelling secrets to be hidden from competitors. Hence, when utilizing co-simulation technology in joint projects, the goal of optimizing a design in a specific working environment is possible with all the third-party vendors and competitors around the same table.

Co-simulation technology has long been utilized in the aerospace industry [10], primarily using the HLA standard [11–13], as well as in the automotive industry [14–16], primarily using the FMI standard. The maritime industry is slowly starting to follow [4, 17–19]. However, it is not without reasons that the maritime industry is running late when it comes to utilizing co-simulation technology. In contrast to the automotive industry, where the majority market share for a given product segment is often held by a single vendor—Bosch's dominance in micro-electromechanical systems (MEMS) market being a prime example [20]—the maritime industry has many third-party vendors where none have the market majority. This makes the companies keep their cards close to their chests, unwilling to share sensitive, but important information about their products. Another reason for running late in this digital working environment is that, while the automotive industry can invest a lot of resources in one prototype since it lays the ground work for mass production, in the maritime industry a vessel is tailored in each case and is rarely mass produced. Therefore, the industry is reluctant to utilize new technology before it has been thoroughly tested since the potential risks of failure are quite costly. Nevertheless, the development of co-simulation technology has reached a level where its benefits are clear and the risk of failure is decreased. The use of co-simulation enables multi-domain simulations, which makes it possible to test a vessel design, including all its subsystems and equipment, using different modeling and simulation software suited for specific systems. The total co-simulation model

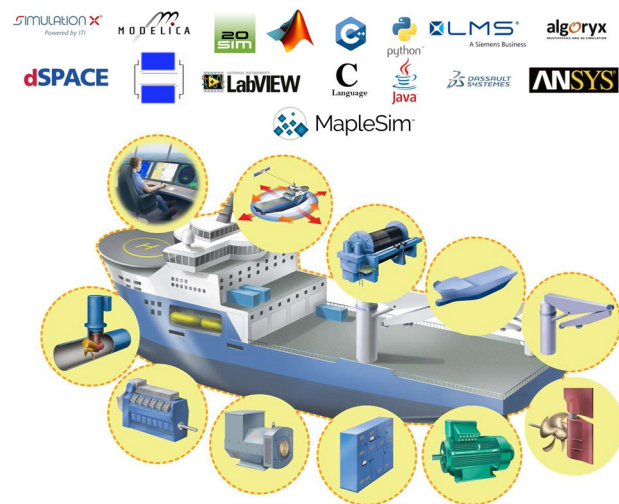


Fig. 1 A vessel and its equipment can be modeled using different software in combination using co-simulation technology [21]

of a vessel is also useful after commissioning since it can be used in a training simulator (Fig. 1).

1.4 Scope of work

This article aims to present some of the main findings in the ViProMa project and to illustrate the applicability of distributed simulation technologies in the maritime industry through several simple, but relevant case studies. Although the focus of the article is the maritime industry, the presentation should be of interest to researchers in other engineering disciplines as well. Hence, the presentation will focus mainly on application of the co-simulation strategy from the ViProMa project for the maritime industry as well as illustrating future possibilities based on the findings in the project. However, some background theory regarding distributed simulation technology and the FMI standard are given in order to provide the reader with some context and to improve readability. Hence, some of the presented topics will overlap with the presentation given in [22].

1.5 Outline

In the next section, we provide some background on co-simulation technology. Then, the co-simulation software Coral and the ViProMa project are presented in more detail in Sect. 2 before results from the research conducted in the ViProMa project are presented in Sect. 3. Thereafter, four use cases and demonstrators developed in the ViProMa project are presented in order to illustrate the applicability and promote the use of co-simulations in the maritime industry in Sect. 4. Finally, a conclusion is made in Sect. 5,

where we also discuss future opportunities afforded using co-simulation technology in the maritime industry.

2 Background on co-simulation

Co-simulation is a simulation technique in which the computations associated with different subsystems are performed independently from each other, and data exchange between subsystems is restricted to discrete *communication points* (sometimes called *synchronization points*). Each subsystem is then free to use the solver strategy and internal “micro” time step size which is deemed most suited. The time between communication points, the “macro” time steps, will generally be significantly longer than the micro time steps of most subsystems. A co-simulation is driven by a dedicated software or algorithm which determines the macro step sizes and routes data between subsystems according to a chosen output-input variable mapping. This software is variously called a *co-simulation software*, *master algorithm*, *co-simulation master* or *run-time infrastructure* (RTI).

The subsystems in a co-simulation can vary greatly in both complexity, fidelity and type, ranging from simple input–output mappings, like signal gains or empirical algebraic equations that do not require any local numerical solvers, to complex differential equations with varying time constants and sophisticated solvers. In fact, the subsystems do not need to be based on model equations at all; they can just as well be interfaces to hardware such as sensors, actuators and human input devices, or observers such as data loggers and visualization systems.

In order for the co-simulation software to be able to communicate with the different subsystems, some kind of common interface or communication protocol is needed. Several such exist, and the most prominent one is probably the *High-Level Architecture* (HLA) [23]. While it has its origin in military applications such as wargaming, the HLA standard describes a general-purpose co-simulation architecture [24] and has been used for a variety of civilian purposes, including systems engineering. Multiple HLA implementations exist, both commercial and free.

Another standard which has been gaining traction in the engineering community in recent years—in particular in the automotive sector—is the *Functional Mock-up Interface* (FMI), which we describe in the next section. Both HLA and FMI were considered as the preferred co-simulation interface in ViProMa, but the choice eventually fell on the latter. A comparison between them and a rationale for the choice of FMI is given in [22], and we will not dwell on it here.

2.1 Functional mock-up interface

The *Functional Mock-up Interface* (FMI) is a tool-independent standard for the exchange of dynamical models and for co-simulation [25]. The first version of the standard was published in 2010 as a result of the ITEA2 project MODELISAR. Since 2011, maintenance and development of the standard have been performed by the Modelica Association, and a second major version, FMI 2.0, was released in 2014 [26].

The FMI standard describes how models may be packaged into mostly self-contained units called *functional mock-up units* (FMU). An FMU is an archive (ZIP) file that contains metadata, machine code and data files, and optionally documentation and source code. FMI specifies an XML schema for the metadata as well as a C programming language interface for the model code. This allows simulation tools to obtain information about the model from the metadata—such as the names and types of its variables, and even advanced information such as the relationships between different variables—and to use the contained model by calling its C functions. These functions perform different predefined tasks such as initializing the model, setting and retrieving variable values, carrying out the model computations for a time step, and so on.

There are two aspects of the FMI standard: The first is *FMI for Model Exchange*, which is intended for models that consist of a set of differential equations which do not come with their own solver, and which therefore must be imported into a tool that supplies a general-purpose solver. The other is *FMI for Co-simulation*, which is used when a solver is either not needed or is bundled within the model code. For the remainder of this paper, when we refer to FMI, we shall exclusively use it in context of co-simulations. The overall structure of an FMI-based co-simulation is shown in Fig. 2.

FMI for co-simulation is based on a master/slave control paradigm, where the submodels are slaves which are controlled by a master simulation algorithm. An FMU serves as a “blueprint” for slaves, meaning that a simulation may contain several slaves which are separate instances of the model contained in one FMU.

The master algorithm decides the length of the communication intervals and when each time step is carried out, and it determines how to route data between the slaves’ output variables and input variables. The slaves simply receive inputs, perform computations based on those inputs and their internal state, and produce outputs based on the results. Aside from input/output values, they are otherwise completely isolated from the system, and have no information about the origin of their input values or where or how their output values will be used. Thus, FMI by design helps to minimize model interdependencies, which has very positive

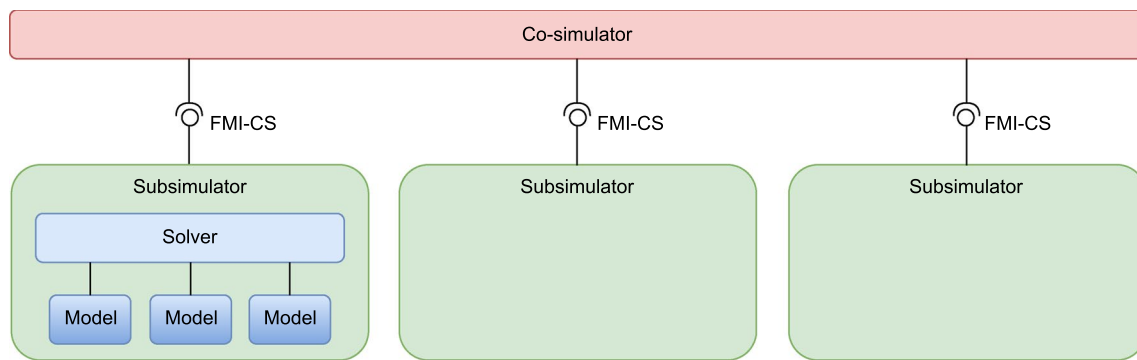


Fig. 2 Structure of a co-simulation that uses FMI [21]

effects both on the scalability of full-system simulations and the reusability of individual models.

3 Results from ViProMa

The research results from ViProMa are concentrated around co-simulation, covering both general technology and methods as well as their applications to maritime systems and operations. Both the co-simulator side and specific subsimulators have been studied. Some of the most significant research and central results are summarized in the following sections.

3.1 Virtual prototyping framework

The primary goal of the ViProMa project was to advance and facilitate the use of simulation and virtual prototyping as a tool for collaboration, innovation, and rapid design and development in the maritime industry. To that end, the project developed the *Virtual Prototyping Framework* (VPF): a set of practical guidelines for simulation of maritime systems aimed at engineers and designers rather than experts in simulation theory. The guidelines cover simulation methods, model coupling, simulator interfaces and more. They are supplemented with research-backed explanations and rationale, as well as software tools such as the co-simulation software *Coral*, which is described in the next section. All of this has been published on a dedicated website: <https://viproma.no>. The aim is for this website to become a living, continuously evolving and up-to-date repository of information and software for the simulation community. Its content is freely available and usable by anyone; no payment or even login is required.

A more comprehensive and academically oriented presentation of the VPF is given in [22], and we will not go into any further details here.

3.2 Coral: distributed co-simulation software

Coral is a free and open-source (FOSS) co-simulation software built from the ground up with support for FMI and distributed simulations in mind. It is primarily a software library that can be embedded into any application that needs to perform co-simulations. However, some simple command-line applications have been developed for testing, research and demonstration purposes, and these allow Coral to be used as a stand-alone co-simulation system as well.

Being designed around FMI, Coral is based on the same master/slave model of communication and control. It is a fully distributed system, where the master communicates with its slaves over network connections. This allows users to perform simulations where different slaves run on different machines just as easily as if they were all running on the same computer, thus enabling workload distribution as well as simultaneous use of multiple hardware/software platforms.

Coral is available for download,² both in source form and compiled form, from the ViProMa website [21].

3.3 Power connections between submodels

One of the first topics discussed in the ViProMa project was connectivity and model standards concerning inputs and outputs in submodels. If a model standard for domain models could be established, it would simplify collaboration in the industry, since domain models could then be interchanged without further explanation about connectivity and units. However, one of the major obstacles in this discussion was model fidelity, since many domain models with different model fidelities require different model inputs and

² A fair word of warning is that, at the time of writing, Coral is still in a pre-1.0 stage where it is under heavy development and likely to undergo several changes before it matures to a stable state.

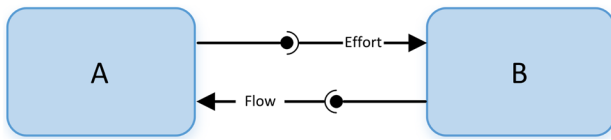


Fig. 3 Power connection between the two subsystems A and B

Table 1 Energy domains and power variables

Energy domain	Effort	Flow
Mechanical (trans.)	Force [N]	Linear velocity [m/s]
Mechanical (rot.)	Torque [Nm]	Angular velocity [s ⁻¹]
Electrical	Electromotive force [V]	Current [A]
Hydraulic	Pressure [Pa]	Volumetric flow rate [m ³ /s]
Thermal	Temperature [K]	Entropy flow rate [W/K]

outputs, as well as parameter sets. Hence, in some situations, replacing one low fidelity model with a higher fidelity model would not become a trivial task. However, some work regarding model standards was done and can be found on the project's web page [21].

Instead of focusing on making fixed model standards for domain models with different model fidelities, the ViProMa project adopted a model connection standard from the bond graph modeling theory [27], a modeling theory that focuses on the power exchange between dynamical effects in a system in an object-oriented scheme [28]. In bond graphs, different submodels and dynamical effects interact with each other through power which is a good connection quantity since it is defined equally in all energy domains. As it turns out, the exchange of power between submodels is closely related to both stability theory and simulation accuracy; we discuss this later on. Further, the power connection is divided into two connection variables denoted *effort* and *flow* which catches the *action* and the *reaction* between two connected systems, as can be seen in Fig. 3.

If the action of system A is to set the effort for system B, it gets a flow in return. The same can be said about system B, the action is to force a flow on system A and gets an effort in feedback, and the product of the effort and the flow is power. Hence, the subsystems exchange power through the connected power variables. Assuming that systems A and B are in the translatory mechanical energy domain, system A sets a force input to system B and receives a velocity in feedback, and the product of the force and the velocity is power. Connecting power variables for other energy domains are summarized in Table 1.

The ViProMa project goes as far as to recommend the use of power variables in the input–output (*I/O*) mapping when

making FMUs. However, some exceptions do exist. When working with control systems, it is difficult to use power connections. One example that illustrates this is the speed controller, the governor, controlling a diesel engine. It receives a reference speed and the measured speed while giving fuel injection rate in feedback to the engine. The rule of thumb used in the ViProMa project is to model each dynamical connection between equipment in real life with power connections. For example, the dynamical connection between a diesel engine and a generator is the engine shaft; therefore, the engine and the generator should exchange data through power variables in a co-simulation. The use of power connections between submodels in a co-simulation also introduces some nice features when studying the accuracy and the stability of co-simulations, as will be discussed in more detail in Sect. 3.5.

Even though power connecting variables are used to exchange data between submodels in a co-simulation, connectivity in general can not always be ensured. This has to do with the *causality* of the models—that is, whether a power variable is given as input or as an output in a submodel—and is crucial when connecting it to a model environment. This is discussed in more detail in the following.

3.4 Causality and connectivity

In a mathematical model representing the dynamics of a physical system, the causality of the model gives away the structure of the dynamical equations contained in the model. In general, a differential equation represents an integral causality form of the respective dynamics, and a differential algebraic equation represents a differential causality form of the respective dynamics. In other words, the causality of a model highly influences the states in the model and, thus, the *I/O* mapping of that model. To illustrate this, a mass-damper-spring system is used and its integral causality form is given as:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -\frac{k}{m}x - \frac{b}{m}v + \frac{1}{m}F(t) \end{aligned} \quad (1)$$

where x is the position of the mass, v is the velocity, m is the mass, k is the spring stiffness, b is a damping parameter and $F(t)$ is a driving force. In this model representation, $F(t)$ is input to the model and typically x or v is given as output from the model. Note that if connecting power variables are used, v would be given as output according to Table 1. The differential causality model of the mass–damper–spring system is given as:

$$\begin{aligned} \dot{x} &= v \\ F(t) &= m\dot{v} + bv + kx \end{aligned} \quad (2)$$

Here, the velocity is given as a model input and the force $F(t)$ is given as a model output, according to the power-based

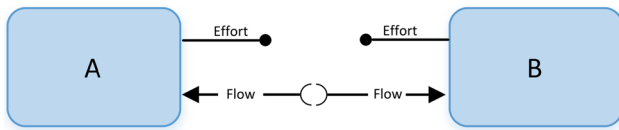


Fig. 4 Connection problem between the two subsystems A and B

submodel connections discussed in Sect. 3.3. Note that the differential causality has removed one of the states in the model and replaced it with a differential algebraic equation. This is problematic when analyzing the stability of the model.

In some cases, the causality orientations of two submodels that are to be connected do not fit. One classical example in the maritime industry is a deck crane model that is to be connected to a hull model. Both these models are quite similar to the mass–damper–spring system, and when having integral causality they both require external forces as model inputs and sets velocities and angular rates as model outputs, as illustrated in Fig. 4.

Hence, one of the models should change causality in order to ensure connectivity. However, as for the mass–damper–spring system, the differential algebraic equations become port dependent, meaning that the models are also connected by algebraic loops, since the derivative of an input signal is needed. Such systems are characterized as tightly coupled systems, and are not recommended for distribution. In [29], a generic method for combining a crane and a vessel into one single model is presented, but this reduces the modularity of the models since it is not straightforward to replace one crane design with another.

One might be tempted to calculate the derivative of the model inputs numerically, but the numerical errors would become too significant since they are sampled signals. To overcome the problems related to causality, connectivity and tightly coupled systems, some research on *hybrid causality models* were conducted and are presented in [30]. The idea of a hybrid causality model is that differential causalities can be reformulated to differential equations by applying a filter with differential properties. Hence, the model is reformulated to a full state-space model without differential algebraic equations in such a way that connectivity is ensured. Also, the causality can be formulated in a hybrid setting such that it is possible to change causality, and thus *I/O*-mapping of the model, online during a simulation. This is quite useful when working with discrete dynamics, as illustrated in [31], where a marine power plant model is presented. There, the generators were implemented as hybrid causality models, which allows switching between outputting current or voltage to the power grid. This is necessary when connecting and disconnecting generators to the power grid, if the power grid itself does not provide the voltage.

Since the reformulated differential causality method converts differential algebraic equations into differential equations, it is also possible to analyze the stability of the model in a co-simulation. This is elaborated in the following.

3.5 Stability and accuracy

In general, the stability of a system and the stability of a simulation are often divided and analyzed separately in non-distributed systems. However, in distributed systems these two stability considerations are more closely connected. This can be explained by considering the following: If the global communication time step is increased such that it approaches infinity, the subsystems in the co-simulation never interact with each other and are considered solved separately with constant inputs. Hence, the eigenvalues of the total co-simulation system are the union of all local eigenvalues. On the other hand, if the global communication time step is decreased such that it approaches zero, the subsystems in the total co-simulation interact with each other continuously. Hence, the eigenvalues of the total system would be dependent on all the connected subsystems. This means that the eigenvalues in a co-simulation are not equal to the eigenvalues in the total continuous system, nor the union of eigenvalues of each separate subsystem, but rather something in between. In addition, numerical errors from local solvers come on top of this and complicate things further.

Since the eigenvalues in a co-simulation system are highly dependent on the global communication time step, a combined stability analysis is recommended to ensure both dynamical and numerical stability. In [30], a combined stability analysis method based on the Euler integration method as a test function is proposed and is similar to Dahlquist test equations [32]. The use of the Euler integration method can be argued for since the stability region of this method is contained in most stability regions for explicit numerical solvers. In general, for linear systems the stability analysis method steps through each local subsystem between two global time-steps in order to calculate the local solutions of each subsystem in the distributed system using the Euler integration method. Then, each local solution is put into a global solution according to the system connections in the distributed system. Then, if the magnitude of each eigenvalue in the total solution system is less than one, the total solution converges and the co-simulation is stable. Moreover, if each local solver in the distributed system is the Euler integration method and the total system only contains linear dynamics, the proposed stability analysis method would be exact. If higher order numerical solvers are used instead, the stability analysis would be more conservative. The method also works for non-linear systems, but then operation regions for each state in each non-linear subsystem must be defined and used in the analysis as a maximal-minimal eigenvalue

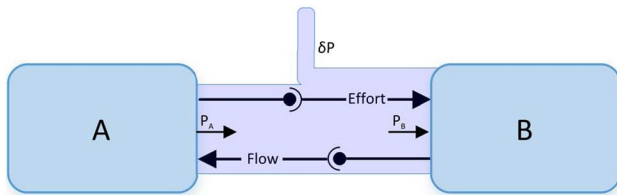


Fig. 5 Simulation accuracy control using the ECCO algorithm

study and will result in conservative stability results. Other relevant numerical stability and convergence results for co-simulations are presented in [33], including both explicit and implicit co-simulation methods.

Closely related to the stability of a distributed simulation is the accuracy of the simulation results. Since each input to a submodel in a co-simulation is normally held constant between each global communication time step (zero-order hold sampling), the energy transport between subsystems, through the connecting power variables (see Sect. 3.3), would not be correct. Hence, a subsystem would either receive/transmit too much power or too little power to the submodel environment in transient simulation regions, due to the fundamentals in the co-simulation strategy, and this affects the accuracy of the simulation results as well as the stability of the system. This is thoroughly studied in [34], which proposes an energy-conservation-based co-simulation method (ECCO) that aims to reduce the power discrepancy between submodels.

The main idea of the ECCO algorithm is to calculate the power level from the inputs and outputs of each connected submodel (P_A and P_B) and making them converge using a simple control law that adds or removes the residual power ($\delta P \equiv P_B - P_A$) in the connection, as shown in Fig. 5. One of the main advantages of this method is that it does not require retaking of global time steps. This makes it ideal for practical use, as re-stepping is often not supported by models, especially the custom-made models commonly used in industrial and research settings. For more details about ECCO, the reader is referred to [34].

Stability and accuracy in co-simulations have been minor research topics in the ViProMa project and more research should be devoted to these topics in the future.

4 Application of co-simulation technology in the maritime industry

During the ViProMa project period, different use cases and demonstrators were made mainly for research purposes. These case studies show a wide range of the use of co-simulations in the maritime industry, as well as in research projects. They include Hardware-In-the-Loop (HIL) in

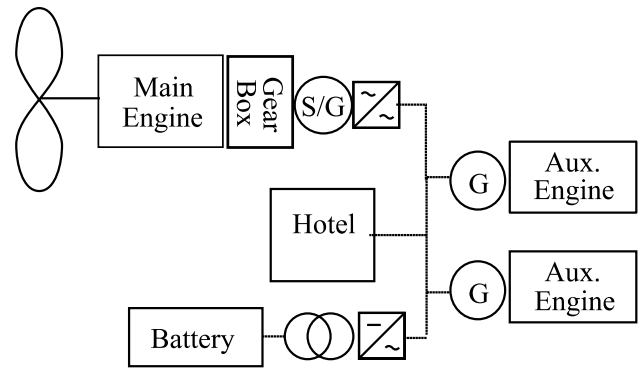


Fig. 6 Total system overview of vessel in transit operation affected by facing wave loads [35]

co-simulations, collaboration between researchers using co-simulations, optimizing system integration using co-simulation and testing different vessel configurations using co-simulation. Some of these cases are presented in the following. Note that the main focus in these case studies is not the simulation results themselves, but the applicability and advantages of utilizing co-simulation technology in complex engineering tasks in the maritime industry, although the simulation results also have research value in themselves. Since the ECCO algorithm presented in Sect. 3.5 has not yet been implemented in the Coral co-simulation software, the co-simulation case studies presented in this section will be solved with a constant communication time step size. Also, each connection between subsystems in the presented case studies is explicit, meaning that no port-dependent algebraic equations or relations are present.

4.1 Research collaboration

When researching complex systems that grow large because of many subsystems with high fidelity, specialized software for subsystems is hard to combine in a generic fashion. However, by utilizing co-simulations, researchers can work on different subsystems in their preferred software without being concerned about compatibility except for model interfaces. In [35], five researchers looked into using a shaft generator to reduce the transients of a two stroke maritime engine powering a vessel in a transit operation caused by significant wave loads. In such operations, the propeller might ventilate causing varying loads on the propeller and, hence, the propulsion system. While two of the researchers were researching wave loads and ventilation of propellers, the three other researchers were looking into the power systems. The total power system including the propeller was as illustrated in Fig. 6.

In addition, the vessel dynamics were included in the study as well. The power plant including the auxiliary

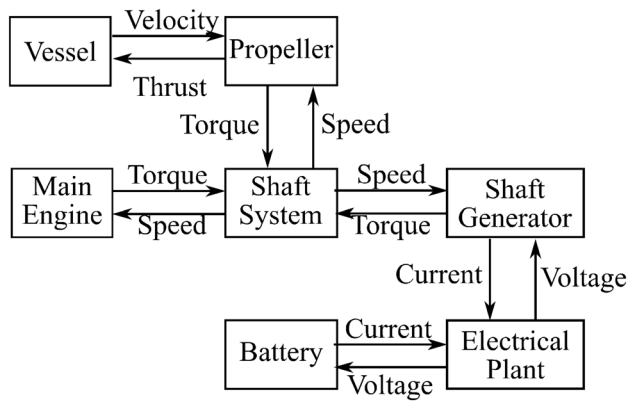


Fig. 7 Simulation setup. Note that each block represents a FMU in the total co-simulation and that control signals and measurements are neglected [35]

Table 2 Subsystems and integration methods

Subsystem	Integration method	Time step
Hull	Euler, 1st order	100 ms
Propeller	Euler, 1st order	10 ms
Shaft system	Euler, 1st order	10 ms
Diesel engine	Runge–Kutta, 4th order	2 ms
Power Plant	Euler, 1st order	0.1 ms
Shaft generator	Runge–Kutta, 4th order	0.05 ms
Battery system	Euler, 1st order	10 ms
Control system	Euler, 1st order	10 ms

engines, the generators and the hotel load was exported as one FMU, while the two stroke engine model, the vessel model, the propeller model, the shaft model, the shaft generator model and the battery power pack model were each exported as separate FMUs. The models were mainly constructed in the software 20-Sim and Simulink, and the total system was simulated as a co-simulation using Coral as shown in Fig. 7. A more detailed discussion about the model connections including the control signals and measurements is given in [35].

Note that all connections between submodels shown in the figure are power bonds, according to the bond graph modeling theory as presented in Sect. 3.3, since the product of inputs and outputs for each submodel is power. Because of the amount of computational power needed to solve the total system, and the fact that different modeling software was used to make the dynamical models, such a simulation study would have been difficult and time consuming to perform in a traditional manner as a non-distributed system.

In the co-simulation, the local numerical solvers and corresponding time steps are shown in Table 2 and the global communication time step was set to 10ms.

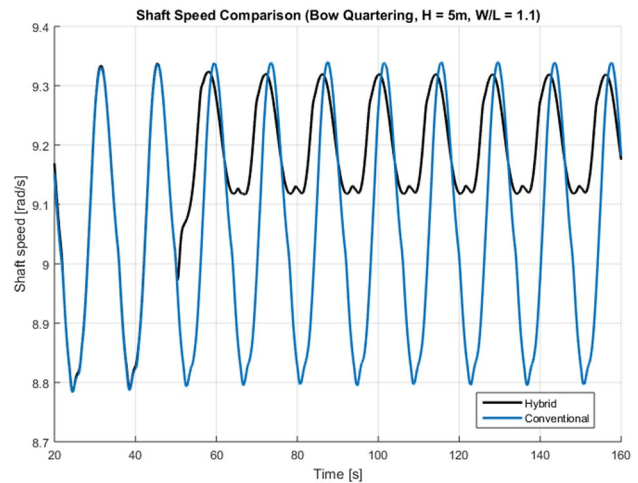


Fig. 8 Simulation results showing the comparison of shaft speed between the hybrid propulsion system, including a shaft generator to reduce the transient wave loads, and a conventional propulsion system [35]

The co-simulation results were compared to a conventional propulsion system, e.g., the two stroke engine powering the propeller without any shaft generator, and the comparison of the shaft speed is shown in Fig. 8 for waves with a significant wave height of 5 m and a wave length of 352 m. Note that the conventional propulsion system has also been simulated as a co-simulation and that the shaft generator is activated after 50 s in the simulation for the hybrid propulsion system.

The results showed that by applying a suited overall control system, the shaft generator was able to reduce the transient wave induced loads on the controller and, hence, smooth the operational conditions for the two stroke engine.

In this particular case study, the total system was also implemented as one non-distributed system for comparison and verification reasons. Nevertheless, such a verification of the co-simulation results is not considered here since it is thoroughly presented in [35] where the main results are that the co-simulation results converge to the results from the non-distributed system.

When it comes to dynamical stability and combined stability of the total co-simulation, as described in Sect. 3.5, it will be a tremendous work proving overall stability of the total system with the stability criterion derived in [30]. However, when working with *passive systems* [36], systems that dissipate energy, a more practical stability observation can be utilized. By ensuring that each subsystem that produce energy is stable by itself, and that all other systems in the co-simulation are only dissipating energy, storing energy or transforming energy from one energy domain to another, it is possible to sum the amount of produced energy with the amount of dissipated energy, in a source-sink analysis

approach. Hence, if the total system is able to dissipate all the produced energy, the system would be stable. This is also why power bonds are recommended when modeling, since it simplifies such practical analysis. This approach is in fact a practical interpretation of the passivity theorems presented in [36]. As it turns out, if the total system is passive according to the theorems, the system will be stable independently of the global communication time step. Hence, the only concern is the accuracy of the simulation results which also can be improved by the ECCO-algorithm presented in Sect. 3.5.

When it comes to control systems, it is often infeasible to design the I/O according to power bonds. However, the stability criterion derived in [30] is still applicable, as well as for example sampling theory and passivity theory. In general, care must be taken when choosing the global communication time step when integrator effects in control systems are considered.

For larger systems often more complex control system structures are required, and often the control system consists of multiple layers. This is especially the case for maritime vessels in Dynamic Positioning (DP) systems where a DP-control law controls the position and the orientation of the vessel and feeds local propulsion systems with reference signals. If the vessel is over-actuated, a more sophisticated thrust allocation algorithm is also present between the DP controller and the local propulsion controllers. When having multiple layers with controllers, it is important to optimize the total performance in order to obtain the best possible response of the vessel. This is often referred to as system integration and is often optimized by proper control system tuning. Such a case is presented in the following.

4.2 Optimizing system integration

When it comes to optimizing a vessel's performance, the largest potential lies in system integration and is often related to control system integration on different layers [37]. One typical case is to tune the DP controller, filters, thrust allocation algorithm and local thruster control systems such that the performance of the total control system results in a fast and stable response of the vessel that minimizes the power consumption. Since such control layers in real life are affected by sampling dynamics and sampling delays, the use of co-simulation works perfectly in simulating the interaction between the different parts of the total control system. One such case was studied in the ViProMa project, where the vessel model derived in [29], with the main parameters as given in Table 3, was connected to the power plant model derived in [31]. In addition, electrical motors and propulsors, constituting each thruster, two azimuths at the stern and one tunnel thruster in the bow, were connected to the total system, as shown in Fig. 9, constituting the mechanical and electrical models in the total system.

Table 3 Main parameters in vessel model

Parameter	Description	Value
L	Length of ship	107 m
B	Width of ship	22 m
D	Draught of ship	5 m
v_{cN}	Northward current	- 0.1 m/s
H_s	Significant wave height	1.0 m
T_p	Wave peak period	8 s
N_w	Number of wave components	50
γ	Jonswap-spectrum parameter	3
T_d	Lower wave spectra period	0.2 s
T_u	Upper wave spectra period	50 s

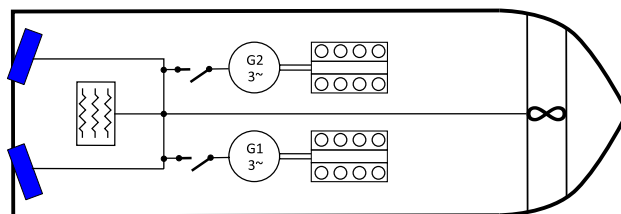


Fig. 9 Overview of vessel in DP operation including power plant and thruster configuration

The power management system controlling the power plant and auxiliary power grid load was implemented in the power plant itself while the generators were implemented as hybrid causality models, as described in Sect. 3.4. This, because the power grid was considered weak such that even a small disturbance in power grid load will affect the power grid voltage. Hence, one of the generators must set the power grid voltage while other active generators contribute with currents. When considering starting, stopping, synchronizing of the generators, as well as load sharing, it is of interest to keep the generator models as generic as possible as well as not fixing which generator should set the power grid voltage. Hence, a generator model with hybrid causality enables the model to alter the I/O mapping online during a simulation which means that one can change which generator sets the power grid voltage. A practical approach of implementing such generator models is presented in [31].

In this case study, the wave filter filtered the position and heading measurements from the vessel to be used in the DP controller. The DP controller output was then fed to the thrust allocation algorithm derived in [37], giving thruster control reference signals to local controllers controlling each thruster. Note that in order to increase the response of the azimuth thrusters, thruster biasing for the two main thrusters is considered, meaning that the thrusters should slightly cancel each other in order to increase the response of the propulsion system since they then can store some extra thrust. In

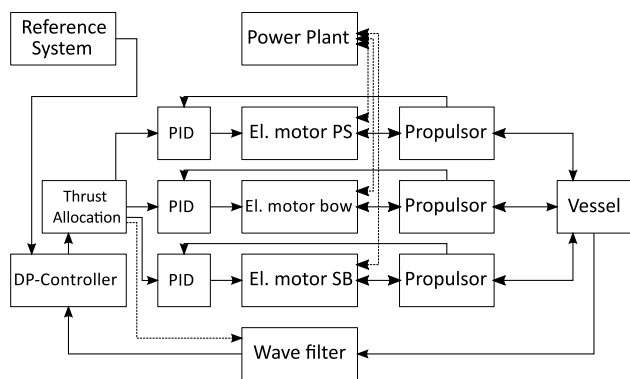


Fig. 10 Overview of vessel in DP operation including power plant and thruster configuration

second. The length of the co-simulation was set to 6000 s and two different tuning cases of the control systems were tested.

The simulation results showing the vessel in a north-east plot compared to the desired position overlap in the two tuning cases are given in Fig. 11.

As can be seen in the figure, the vessel follows its reference quite well, even though there is more noise in the position of the vessel when the vessel faces the waves with the heel. Note that in each corner in the position trajectory the vessel changes heading while trying to keep a fixed north-east position.

The simulation results from the propulsion system as well as the power plant for the first tuning case are shown in Fig. 12.

Table 4 Connections between subsystems

Connections	Input	Vessel	Propulsor	El. motor	Power plant	PID	Thrust alloc.	DP	Wave filter
Output	↗	↑	↑	↑	↑	↑	↑	↑	↑
Vessel	→		m/s						\mathcal{M}
Propulsor	→	N, rad		rad/s		\mathcal{M}			
El. motor	→		Nm		A				
Power plant	→			V					
PID	→			\mathcal{R}					
Thrust alloc.	→					\mathcal{R}			
DP	→						\mathcal{R}		
Wave filter	→							\mathcal{M}	
Ref. system	→							\mathcal{R}	

Note that the connecting variables between subsystems are given in SI-units except for \mathcal{M} , \mathcal{R} and τ which are abbreviations for measurement, reference and thrust forces and torques, respectively

this case, the biasing angle was set to $\pm 20^\circ$, meaning that if the thrusters are to produce thrusts in surge direction, one thruster has a biasing angle of -20° while the other a biasing angle of 20° . The total co-simulation setup is shown in Fig. 10 and the subsystem connections are given in Table 4.

As can be seen in the figure, the local thrusters were controlled by simple PID control laws and a separate reference system FMU set the position and orientation references for the DP controller. The connections between each dynamical system, excluding the controllers, the thrust allocation, the wave filter and the reference system, are power bond connections as explained in Sect. 3.3. The total co-simulation consisted of 15 FMUs. The vessel was to move in a square wave pattern affected by an irregular wave with significant wave height of 1 m and a current of 0.1 m/s, both coming from the north. The auxiliary power grid load was set to 100 kW, the global time step in the co-simulation was set to 10 ms and the local numerical solvers and the corresponding time steps for each subsystem are given in Table 5.

Note that the DP controller and the thrust allocation algorithm only communicated with the connected systems every

Table 5 Subsystems and integration methods

Subsystem	Integration method	Time step
Hull	Runge–Kutta, 2st order	1 ms
Propulsors	Euler, 1st order	1 ms
Propulsor drives	Runge–Kutta, 4th order	0.05 ms
Power Plant	Euler, 1st order	0.1 ms
Non-linear passive observer	Runge–Kutta, 2st order	5 ms
PID-controllers	Euler, 1st order	10 ms
Reference system	Euler, 1st order	10 ms
Thrust allocation	Euler, 1st order	10 ms
DP controller	Euler, 1st order	10 ms

Plot (a) shows the azimuth angles for the two main thrusters at the stern. As can be seen, the thruster angles stay between $\pm 180^\circ$ and one can clearly see the thruster biasing angle for the two thrusters. The second plot, (b) shows the thrust produced by the two azimuth thrusters. The three regions with a lot of noise are because the vessel moves in

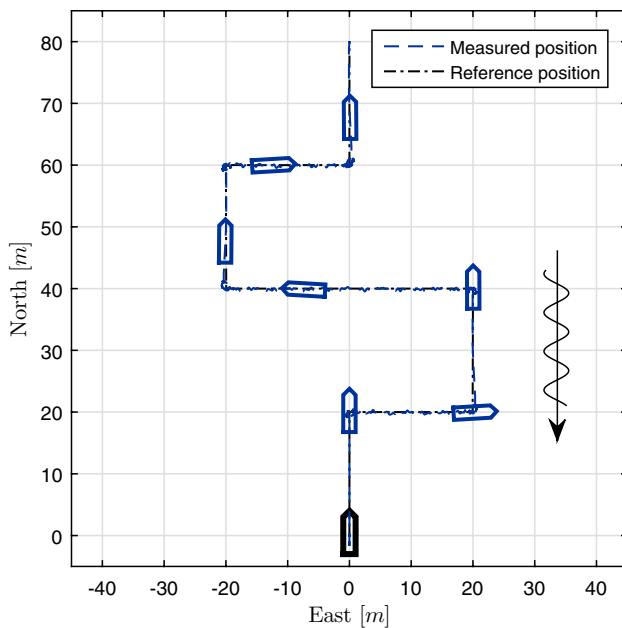


Fig. 11 North-east position and heading of vessel in square wave trajectory manoeuvre. Black vessel outline in the plot denotes initial position and orientation

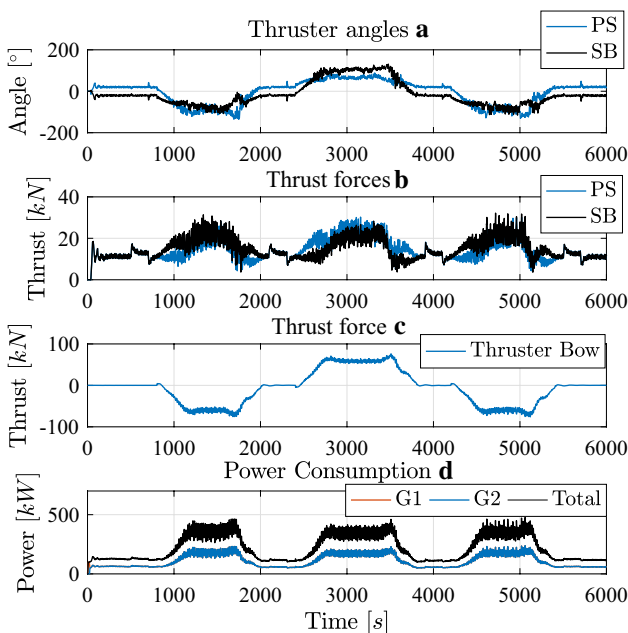


Fig. 12 Simulation results showing the thruster azimuth angles for the two main thrusters placed at the stern, the corresponding thrust, the thrust of the bow thruster and the power produced by the two generators as well as the total power consumption for the first tuning case

the east- or west direction, facing the waves with the heel. The third plot, (c) shows the thrust produced by the bow thruster. Also here, there are some oscillations present due to the wave effects. The last plot, (d) shows the produced power

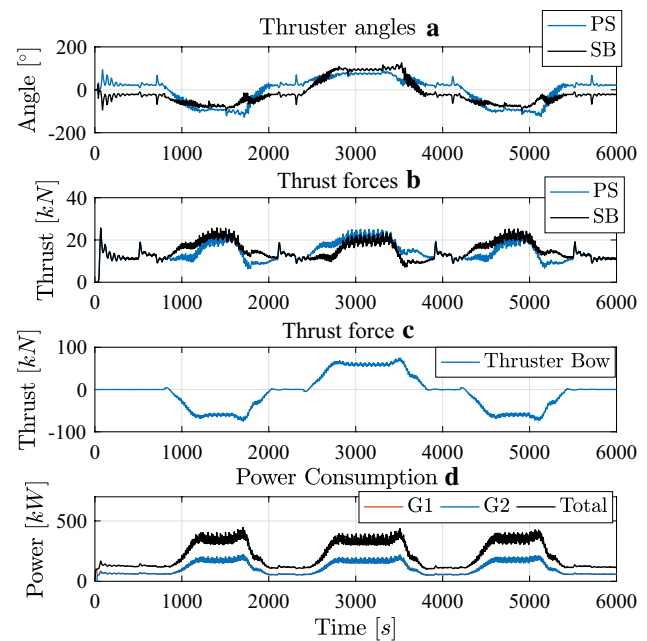


Fig. 13 Simulation results showing the thruster azimuth angles for the two main thrusters placed at the stern, the corresponding thrust, the thrust of the bow thruster and the power produced by the two generators as well as the total power consumption for the second tuning case

by generator 1 (G1) and generator 2 (G2), which overlap, and the total vessel power consumption. Since the produced thrusts from the three thrusters oscillate when the waves encounter the heel of the vessel, it is not surprising that the total power consumption oscillates as well. However, by tuning the different control systems properly altogether it is possible to obtain a smoother power consumption as well as smoother operation of the thruster systems. This has been done and the corresponding results are shown in Fig. 13.

As can be seen in the figure, both the azimuth angles and the produced thrusts oscillate less in this case compared to the previous one, when neglecting the initial oscillations for the azimuth angles and the corresponding thrusts. Also, the oscillations in the power consumption are reduced and result in a slightly lower power consumption as well as reducing wear of the propulsion system and the power plant.

Another crucial requirement in order for the control system to perform properly is the choice of sampling frequency of the different components in the overall vessel control system, and is strongly related to both dynamical stability of the total system and combined stability of the entire co-simulations, see Sect. 3.5. In general, each control layer should be tuned such that the outer control layers are slower than the inner control layers. They may also have a lower sampling frequency. Here, the outer control layers consist of the DP controller and the thrust allocation algorithm, which have a sampling frequency of 1 Hz, while the inner control

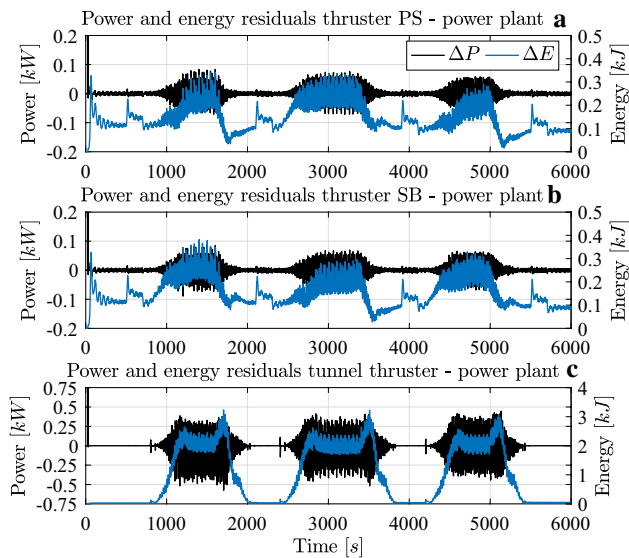


Fig. 14 Power residuals (ΔP) and energy residuals (ΔE) between the thrusters and the power plant in the co-simulation

layers consist of the wave filter and the local thruster control systems, having a sampling frequency of 100 Hz. Note that care must be taken when tuning the DP controller since it contains integration effects and since it has such a low sampling frequency.

Large co-simulation systems that contain both dynamical connections and control connections are in general hard to analyze when it comes to stability. If such analysis is even possible, it would require a huge amount of work. However, another way of quantifying the stability in a co-simulation system is to look at the power and energy residuals in connections between subsystems in the co-simulation, which are utilized in the ECCO algorithm as discussed in Sect. 3.5. This is because the power and energy residuals are closely related to convergence in the co-simulation and will also provide some information about the accuracy of the simulation results due to the discrete communication points between the subsystems. As an example, the power and energy residuals between the electrical motors driving the thrusters and the power plant are shown in Fig. 14.

In the figure, the power and energy residuals for the port-side main thruster are shown in plot (a), the residuals for the starboard thruster in plot (b) and the residuals for the tunnel thruster in plot (c). The results show that the residuals are quite small in comparison to the power and energy transmitted through the model connections. The power residuals for the two main thrusters are lower than 0.4% and for the tunnel thruster lower than 1.7% of the instantaneous transmitted power. Hence, since the power residuals are small, the corresponding subsystems are stable and the simulation results have good convergence properties. Note that if the subsystems were unstable the power residuals would also

become unstable. The accuracy of the simulation results can be discussed from the power residuals too, since low power residuals mean high accuracy in simulation results due to the discrete communication point. The simulation results from the control systems and the vessel motion in this case study are also converging to the simulation results in [37] where almost the same system is simulated as a continuous system except that the power plant model and the electrical motors are idealized.

The power residuals shown in Fig. 14 also argue for stable thruster control systems since the energy residuals seem to be bounded, in contrast to the uncontrolled quarter-car co-simulation system studied in [34] where the energy residuals keep growing during the co-simulation. Hence, co-simulations can also be used as a tool for tuning the control systems before being installed in real processes, which are also affected by sampling characteristics and sampling delays. Such a case study is presented in Sect. 4.3 where a DP controller is implemented on a hardware microcontroller and connected to the co-simulation loop.

This case has shown how co-simulations can be utilized to optimize the total response of multiple layers of control systems for a vessel. Such cases can be quite difficult to study in single modeling and simulation software since the models are made in different software. For example, in this case the control systems are implemented in the C++ programming language as separate units while the mechanical models are made in the 20-Sim modeling and simulation software. Also, the different models may require different local solvers and local solver time steps for stability reasons and, if they were to be implemented as one total model, it would be quite time consuming to solve since the solver and the local time step would have to be chosen based on the largest eigenvalues in the total system.

4.3 Hardware in the loop (HIL)

A small study of hardware in the loop (HIL) in co-simulations was initiated in the ViProMa project and is thoroughly elaborated in [38]. Therefore, only a short presentation is given here.

When including hardware in a simulation loop, proper communication between the hardware and the simulation is important. Because the FMI standard has predefined functions that are called by the simulation master, such as the function *fmiDoStep()*, it is possible to make an FMU with suited functionality such as reading and sending data in a consistent manner through a serial port on the computer running the co-simulation. An Arduino microcontroller [39] was used as hardware and a DP control law for controlling a vessel in DP operations was implemented and uploaded to the microcontroller. In this case, the focus was the interaction between the DP controller and the wave filter; hence,

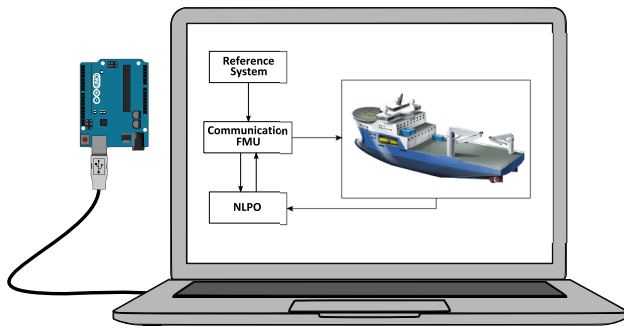


Fig. 15 Simulation setup using hardware in the simulation loop

Table 6 Connections between subsystems

Connections	Input	Vessel	Hardware	Comm. FMU	Wave filter
Output	↗	↑	↑	↑	↑
Vessel	→				\mathcal{M}
Hardware	→			S	
Comm. FMU	→	τ	S		τ
Wave filter	→			\mathcal{M}	
Ref. system	→			\mathcal{R}	

Note that the connecting variables \mathcal{M} , \mathcal{R} , S and τ are abbreviations for measurement, reference, data string and thrust forces and torques, respectively

simplified thruster models and a static thrust allocation were implemented directly into the vessel model, the same as derived in [29]. Since a static thrust allocation algorithm is used, meaning that the main thrusters placed at the stern have fixed azimuth angles, the vessel was to keep its heading northwards during the whole study. Also, the power plant derived in [31] was omitted in this study and the DP controller tuning parameters that were used in Sect. 4.2 are used here as well. The total co-simulation setup is shown in Fig. 15 and the connections between the subsystems are given in Table 6.

As in the previous case study, both the waves and the current encountered the vessel from north, but the significant wave height was set to 1.5 m and the southward current was set to 0.2 m/s. The global communication time step was set to 50 ms, the local numerical solver time steps for each subsystem and corresponding integration method are listed in Table 7 and the hardware DP controller was to communicate with the rest of the simulation every second.

The total simulation time was set to 2000 s. In the simulation, the vessel was set to face the encountering waves while moving in a square-like trajectory, meaning that the heading reference was always zero while the north-east references changed. A nonlinear passive observer (NLPO) [40] was used as a wave-filter and the simulation results for the

Table 7 Subsystems and integration methods

Subsystem	Integration method	Time step
Vessel	Euler, 1st order	10 ms
Non-linear passive observer	Euler, 1st order	10 ms
Reference system	Euler, 1st order	10 ms
Communication FMU	Euler, 1st order	1000 ms

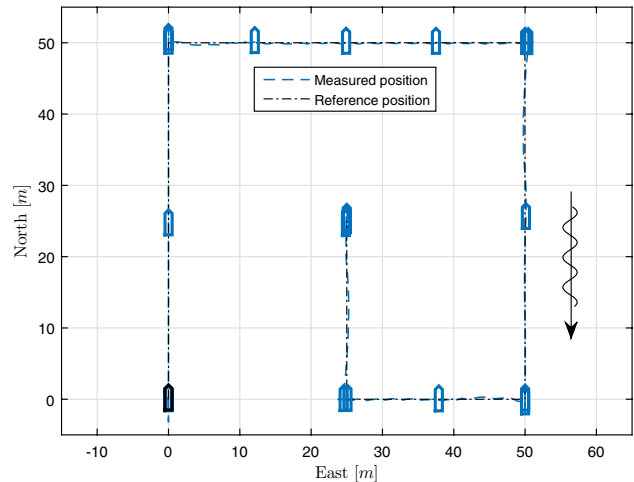


Fig. 16 North-east position and heading of vessel in DP-operation. Black vessel outline in the plot denotes initial position and orientation

position and orientation of the vessel are shown in a north-east plot in Fig. 16.

As can be seen in the figure, the vessel seems to keep its position and orientation also in this case when the DP controller is placed on a microcontroller. The DP control law is a simple PID control law including the rotational matrix related to the heading of the vessel. The rates for the north-east position as well as the heading are estimated by the NLPO which feeds the DP controller with both the rates, the position and the orientation of the vessel. These rates (\hat{N} , \hat{E} and $\hat{\psi}$ —north, east and heading rate, respectively) are shown in Fig. 17 in comparison to the actual rates (\dot{N} , \dot{E} and $\dot{\psi}$) and the reference rates (\dot{N}_d , \dot{E}_d and $\dot{\psi}_d$).

The first plot, (a) in the figure, compares the estimated north rate with the actual north rate and the desired north rate; the second plot, (b) in the figure, compares the estimated east rate with the actual east rate and the desired east rate; while the last plot, (c) in the figure, shows the estimated heading rate compared to the actual heading rate and the desired heading rate. As can be seen in the figure, the wave filter is able to filter out most of the wave induced motions as well as generate good position and orientation rates. It can also be seen from the figure that the rates have biases in the beginning of the simulation. This has to do with the fact that the wave filter needs some time to update the biases

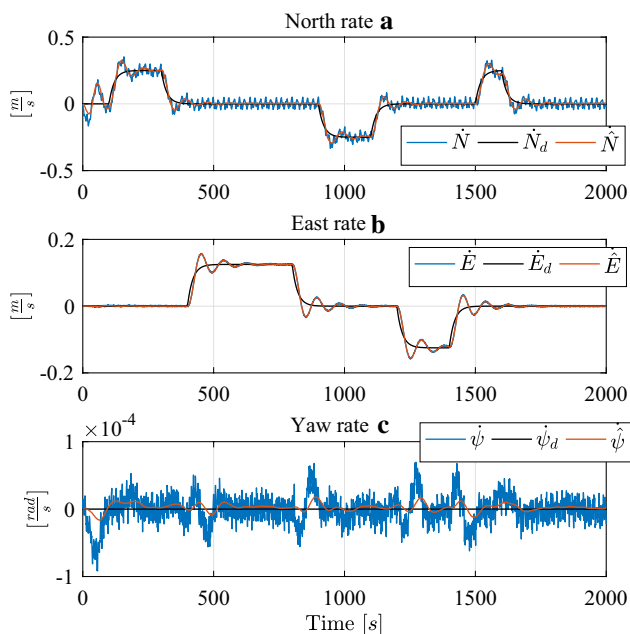


Fig. 17 Simulation results comparing the estimated north, east and heading rates (\hat{N}, \hat{E} and $\hat{\psi}$) with the actual rates (\dot{N}, \dot{E} and $\dot{\psi}$) and the commanded rates (\dot{N}_d, \dot{E}_d and $\dot{\psi}_d$)

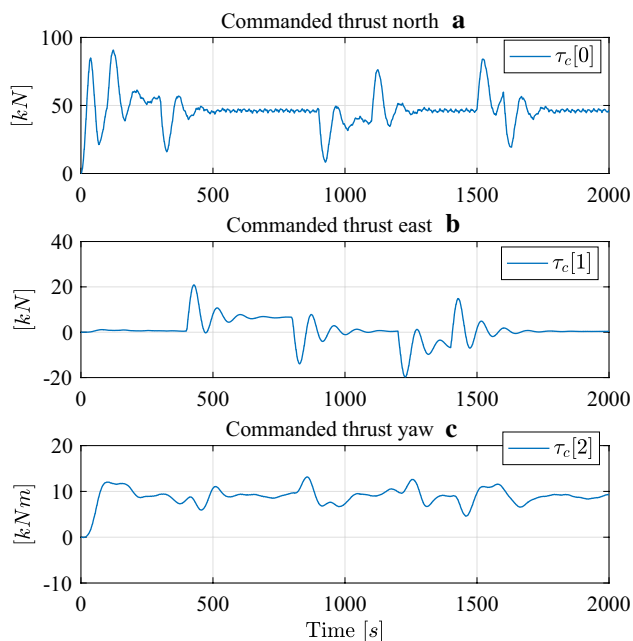


Fig. 18 DP controller commands fed to the co-simulation from the microcontroller through the communication FMU

that represent the slowly varying drift forces caused by the second-order wave effects and the current. Nevertheless, the results show clearly that the wave filter works properly. Figure 18 shows the commands from the DP controller in north, east and yaw for the vessel.

The first plot, (a) in the figure, shows the commanded thrust force in surge, the second plot, (b) in the figure, shows the commanded thrust force in sway and the last plot, (c) in the figure, shows the commanded thrust torque in yaw. As can be seen in the figure, the DP controller seems to be stable and control the vessel to its desired position despite being implemented on a microcontroller and only able to communicate with the rest of the simulation once per second. This means that the DP controller implemented on the microcontroller has the same characteristics and the same sampling properties as the DP controller implemented as an FMU in Sect. 4.2. Hence, the only difference of any significance is the communication protocol used to communicate with the microcontroller as well as real-time limitations related to hardware.

In general, dynamical interactions between systems do not suffer from sampling characteristics in realistic systems. This is one of the drawbacks using co-simulations. However, by applying suited co-simulation algorithms that minimize these sampling characteristics, such as the ECCO algorithm presented in [34], or by manually setting the communication time step small in comparison to the smallest dynamical time constant in the co-simulation [41], the related numerical simulation errors and the corresponding power residuals can be reduced. However, in this particular case study the idea is to mimic the sampling characteristics and the sampling delays that are present in realistic controlled systems. This is why the vessel with all its dynamical systems are considered as one FMU and solved by the same numerical solver in the co-simulation. This can also be seen in Fig. 15 where all connections (arrows) point in only one direction. Hence, the arguments for using co-simulation as a prototyping tool as well as a virtual laboratory for tuning sampled system integrations are strengthened. This also argues for the possibility to have different communication time step sizes between dynamical systems and control systems. However, this has not been implemented in the Coral co-simulation software at this stage.

When considering simulations that are solved in real time, as is required in most simulations that involve hardware in the loop, the complexity tends to increase. This is because high model fidelities, which the quality of the simulation results depend strongly on, do not go well with real-time criteria in continuous systems due to limited available computational power. When the total system becomes large, it is often necessary to reduce the model fidelities in order to reach such real-time criteria. Simulation models used for prediction purposes, such as observers and estimators, often need more and better measurements in order to produce high-quality results when the model fidelities are reduced. To ensure that the measurements have the required quality, they possibly need to be preprocessed as well. One such example is a vessel’s position and orientation measurements,



Fig. 19 NTNU’s research vessel R/V Gunnerus

which need to be filtered before entering the control loop, as was illustrated in this case study. Furthermore, sending large amounts of measurement data in real time can be quite challenging in itself. Some of these challenges can be reduced by utilizing co-simulations, which potentially increases the available computational power and allows the use of numerical solvers and time steps tailored to different parts of the system. However, even though these topics are important for real-time simulations they are considered out of scope here and should be treated in a separate publication.

Another interesting aspect with co-simulation systems is subsystem modularity. For example, it would be straight forward to replace the DP controller in Sect. 4.2 with the communication FMU and the microcontroller. This is of particular interest when designing new vessels, where one would like to test different vessel configurations and equipment in a virtual setting before actually building the vessel. We discuss this in more detail in the following.

4.4 Testing different vessel configurations

NTNU’s research vessel *R/V Gunnerus*, see Fig. 19, is a multi-purpose vessel used in research projects, spanning from developing DP controllers, autopilots, autonomous vessel operations, sub-sea operations using ROVs, surveillance using UAVs, testing fishing equipment and mapping the seabed. The main parameters describing the vessel are given in Tables 8 and 9.

The vessel is equipped with two propellers and rudders at the stern and one tunnel thruster in the bow. In the ViProMa project, a demonstrator case based on Gunnerus was developed based on the specifications of the vessel and in-house mathematical models obtained from different modeling and simulation software, e.g., the hull model is a VeSim model developed in the “Sea Trials and Model Tests to Validate Shiphhandling Simulation Models” (SimVal) project, funded by The Research Council of Norway (RCN) [42–44], the

Table 8 R/V Gunnerus main parameters

Parameter	Description	Value
L_{OA}	Overall Length of ship	31.25 m
L_{PP}	Length between perpendiculars	22 m
B	Moulded breadth	9.6 m
dwt	Dead weight	75t
D	Draft	2.7 m
P	Generator power	3×450 kW

Table 9 Parameters describing the old and the new propulsion system

Parameter	Description	Old	New
α	Rudder rate limit	5.85 deg/s	5.85 deg/s
D	Propeller diameter	2.0 m	1.9 m
P	Propeller power	2×500.0 kW	2×500.0 kW
Z	Number of blades	5	4

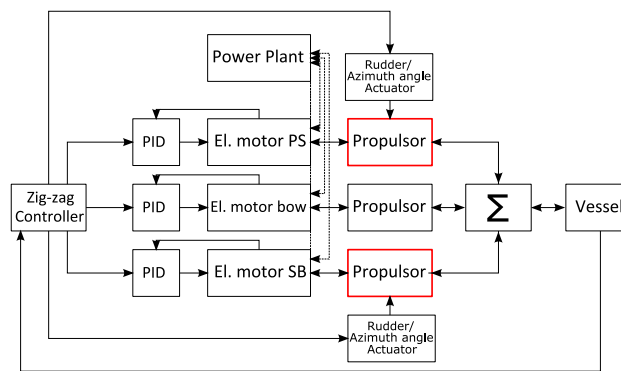


Fig. 20 Simulation setup of Gunnerus

zig-zag controller is derived in Matlab, the PID-controllers in C++ and the electrical motors and the power plant in 20-Sim. The focus in the case was to study the effect of replacing the main propulsors and the rudders with azimuths. The azimuth models were developed by Rolls-Royce Marine in the ViProMa project, while the propeller and rudder models are generic models developed by SINTEF Ocean, parametrized to fit Gunnerus. The total co-simulation setup is as shown in Fig. 20.

In the figure, the main propulsor units placed at the stern are outlined in red colour in order to illustrate that these are the only models that need to be replaced when changing the main propulsors. To compare the two different propulsion configurations, a $10^\circ/10^\circ$ zig-zag test in calm sea is conducted, meaning that the rudder/azimuth angles are given a command of 10° and when the heading of the vessel reaches 10° the sign of the rudder/azimuth angle commands is changed. As key parameters, the surge speed, heading

Table 10 Connections between subsystems

Connections	Input	Vessel	Σ	Propulsor	Rudder act.	El. motor	Power plant	PID	Zig-zag
Output	↗	↑	↑	↑	↑	↑	↑	↑	↑
Vessel	→		m/s						\mathcal{M}
Σ	→	τ		m/s					
Propulsor	→		N, rad			Nm			
Rudder act.	→			rad					
El. motor	→			rad/s			A	\mathcal{M}	
Power plant	→					V			
PID	→					\mathcal{R}			
Zig-zag	→				\mathcal{R}			\mathcal{R}	

Note that the connecting variables between subsystems are given in SI units except for \mathcal{M} , \mathcal{R} and τ which are abbreviations for measurement, reference and thrust forces and torques, respectively

Table 11 Subsystems and integration methods

Subsystem	Integration method	Time step
Hull	Euler, 1st order	10 ms
Propulsor units	Quasi-static, no solver	10 ms
Σ	Static, no solver	10 ms
Rudder/azimuth angle actuators	Rate limiters, no solver	10 ms
Zig-Zag controller	Only logics, no solver	10 ms
Power plant	Euler, 1st order	0.1 ms
Propulsor drives	Runge–Kutta, 4th order	0.05 ms
PID-controllers	Euler, 1st order	10 ms

response and power consumed by each main thruster are compared. Initially, the power plant is started and after 30 s the main propulsors are initiated. The ship is to reach a steady state surge velocity of about 9 kn before the zig-zag manoeuvre is initiated, which happens after 100 s with good margins. The total simulation time is set to 200 s and the global communication time step is set to 10 ms and the local numerical solver time steps and corresponding integration methods are listed in Tables 10 and 11.

The simulation results are shown in Fig. 21.

The leftmost plot in the figure, (a) shows a north-east-orientation comparison of the two propulsion configurations, the upper rightmost plot, (b) shows a comparison of the surge speed, the second, (c) a comparison of vessel heading and the last, (d) a comparison of consumed power in a magnified region for the port-side main propulsion unit. As the results indicate, the surge speed is slightly less oscillating throughout the zig-zag manoeuvre for the case including azimuths, as well as the overshooting heading angle and the consumed power are slightly lower for this case in comparison to the conventional propulsion system. The amount of consumed power is also in the expected range, as argued for by the sea trails conducted on Gunnerus that are presented in [45].

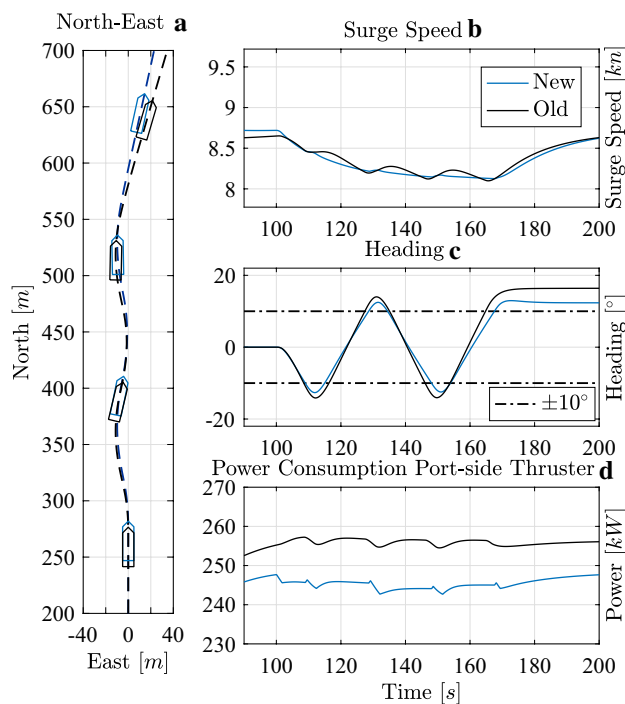


Fig. 21 Results from zig-zag test comparing the old and the new propulsion systems. Note that *New* denotes the simulation case including azimuth thrusters while *Old* denotes the conventional propulsion system including propellers and rudders

This case illustrates the easiness of replacing models in a co-simulation, which is quite interesting when testing different concepts in a fast and virtual setting. This is especially the case when designing new vessels where different vessel equipment or hull designs should be verified to meet the requirement set by the customer.

5 Conclusion

Besides giving a short summary of the ViProMa project, including a short presentation of the major findings and deliverables, such as the open source co-simulation master software *Coral*, the main focus in this work is to demonstrate the use of co-simulation technology in the maritime industry. Four different use cases and demonstrators have been presented in Sect. 4. These include collaboration between researchers and different modeling and simulation software, global system optimizing and tuning, the inclusion of hardware in the simulation loop and testing different concepts in a virtual prototyping fashion in an effective and consistent manner. These cases, in addition to the research conducted in the project, have brought into light new opportunities in the maritime industry by utilizing co-simulation technology. The use of co-simulations in the maritime industry enables

1. the use of black-box models which keep secrets related to systems and equipment hidden from competitors. This makes it possible for ship yards to obtain mathematical black-box models of equipment from third party vendors for testing purposes together with the vessel design before determining which equipment to install and the shipyard is able to compare different design concepts before building the vessel.
2. the vessel designer to design the vessel together with the customer on the fly by choosing different concepts from a model library containing many different vessel designs, systems and equipment. It is also expected in the near future that optimization algorithms taking predefined vessel KPIs into consideration can be implemented as a layer on top of the co-simulation platform in order to conduct simulations and choose different equipment suited the KPIs from a larger model library.
3. simulation-based commissioning of vessels and virtual sea trials to remove design flaws and implementation errors at an early stage. It is also expected that the ship yards can demand black-box models from third party vendors in the near future when choosing to buy their equipment. This would enable them to test the vessel performance before building it, as well as being able to deliver a complete vessel simulator to the customer that can be used for, e.g., operation planning and vessel fleet optimization. It is also believed that the entire maritime cluster would benefit from working in a *maritime cluster cloud* utilizing co-simulation technology in future research.

However, these topics should be devoted more attention and are beyond the scope of the ViProMa project.

Acknowledgements The authors would like to thank the partners involved in the ViProMa project, Research Council of Norway project no. 225322, for providing financial support for researching distributed systems for maritime systems and operations. The authors would also like to thank especially Rolls-Royce Marine for the contributions in the ViProMa Project.

References

1. Evans J (1959) Basic design concepts. *J Am Soc Naval Eng* 71(4)
2. Whitfield RI, Duffy AHB, Wu Z, Meehan J (2005) An overview of the VRS virtual platform. In: 15th International Conference on Engineering Design (ICED'05), 2005. [Online]. <https://strathprints.strath.ac.uk/6369/>
3. Hamburgische Schiffbau-Versuchsanstalt GmbH (2005) VIRTUE: the virtual tank utility in Europe. http://cordis.europa.eu/project/rcn/74944_en.html. European Commission CORDIS web site. Accessed 25 Oct 2017
4. Nagel R (2015) A holistic approach for energy flow simulations in early ship design. In: International Conference on Computer Applications in Shipbuilding 2015. The Royal Institution of Naval Architects, 2015
5. HOLISHIP (2016) Holistic optimisation of ship design and operation for life cycle. <http://www.holiship.eu>. Accessed 25 Oct 2017
6. Dimopoulos GG, Kakalis NMP (2010) An integrated modelling framework for the design, operation and control of marine energy systems. In: Proceedings of the 26th CIMAC World Congress
7. Kakalis NMP, Dimopoulos G (2012) Managing the complexity of marine energy systems. Det Norske Veritas, Research and Innovation, Høvik, Norway, Position paper, Nov. 2012. <http://www.dnvgl.com>
8. TNO. Geïntegreerde energiesystemen (GES). https://www.tno.nl/media/8083/ge%C3%AFntegreerde_energiesystemen_ges.pdf. Accessed 26 Oct 2017
9. Giadrossi G, Menis R, Sulligoi G, Tessarolo A (2009) Voltage stability analysis of all-electric cruise liners. *Electr Eng Res Rep* 1(1)
10. Sogandares F (2002) Stone axes and warhammers: A decade of distributed simulation in aviation research. In: Proceedings of the 16th Workshop on Parallel and Distributed Simulation, PADS 2002
11. Lehmer R, Malsom S (2004) Distributed system architecture in VAST-RT for real-time airspace simulation. In: Collection of Technical Papers—AIAA Modeling and Simulation Technologies Conference, vol. 2
12. Bruzzone A, Mosca R, Orsoni A, Revetria R (2002) Simulation-based VV&A methodology for HLA federations: an example from the Aerospace industry. In: Proceedings—Simulation Symposium, vol. 2002-Janua
13. Hu Li B, Xiao T, Zhang L, Ming Z, Hong N, Xiao-hui W, Xiaomei Q, Enzhi Z (2009) Modeling and simulation of aircraft anti-skid braking and steering using co-simulation method. *COMPEL Int J Comput Math Electr Electron Eng* 28(6):1471–1488
14. Krammer M, Martin H, Radmilovic Z, Erker S, Karner M (2015) Standard compliant co-simulation models for verification of automotive embedded systems. In: Forum on Specification and Design Languages, vol. 2015
15. Winter M, Taube J, Froeschl J, Herzog HG (2015) From simulation to testbench using the FMI-standard. In: 2015 IEEE Vehicle Power and Propulsion Conference, VPPC 2015—Proceedings
16. Abel A, Blochwitz T, Eichberger A, Hamann P, Rein U. Functional mock-up interface in mechatronic gearshift simulation for commercial vehicles

17. Hassani V, Rindarøy M, Kyllingstad LT, Nielsen JB, Sadjina SS, Skjong S, Fathi D, Johnsen T, Æsøy V, Pedersen E (2016) Virtual prototyping of maritime systems and operations. In: Proceedings of the ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering, Busan, South Korea, 2016
18. Chu Y, Hatledal LI, Sanfilippo F, Schaathun HG, Æsøy V, Zhang H (2015) Virtual prototyping system for maritime crane design and operation based on functional mock-up interface. In: Proceeding of the MTS/IEEE Oceans '15 Conference, Genova, Italy. MTS/IEEE, 2015, pp 1–4
19. Paquin JN, Li W, Belanger J, Schoen L, Peres I, Olariu C, Kohmann H (2009) A modern and open real-time digital simulator of All-Electric Ships with a multi-platform co-simulation approach. In: Electric Ship Technologies Symposium, 2009. ESTS 2009. IEEE, apr 2009, pp 28–35
20. Laursen L (2014) Top 10 suppliers of car mems sensors [Online]. <http://spectrum.ieee.org/cars-that-think/transportation/sensors/top-10-suppliers-of-car-mems-sensors>
21. NTNU and Sintef Ocean (2017) <http://viproma.no>
22. Sadjina SS, Kyllingstad LT, Rindarøy M, Skjong S, Æsøy V, Fathi D, Hassani V, Johnsen T, Nielsen JB, Pedersen E (2017) Distributed co-simulation of maritime systems and operations. [arXiv:1701.00997v1](https://arxiv.org/abs/1701.00997v1)
23. Dahmann JS, Fujimoto RM, Weatherly RM (1997) Department of defense high level architecture. In: Winter Simulation Conference Proceedings
24. Simulation Interoperability Standards Organization (2010) IEEE standard for modeling and simulation (M&S) High Level Architecture (HLA)—framework and rules. IEEE Std. 1516–2010
25. Blochwitz T, Otter M, Arnold M, Bausch C, Clauss C, Elmqvist H, Junghanns A, Mauss J, Monteiro M, Neidhold T, Neumerkel D, Olsson H, Peetz JV, Wolf S, GmbH AS, Berlin Q, Scai F, Augustin S (2011) The functional mockup interface for tool independent exchange of simulation models. In: In Proceedings of the 8th International Modelica Conference
26. Blochwitz T, Otter M, Åkesson J, Arnold M, Clauss C, Elmqvist H, Friedrich M, Junghanns A, Mauss J, Neumerkel D, Olsson H, Viel A (2012) Functional mockup interface 2.0: the standard for tool independent exchange of simulation models. In: Proceedings of the 9th International Modelica Conference. The Modelica Association, pp 173–184
27. Karnopp DC, Margolis DL, Rosenberg RC (2006) System dynamics: modeling and simulation of mechatronic systems. Wiley, New York, NY, USA
28. Borutzky W (1999) Bond graph modeling from an object oriented modeling point of view. *Simul Pract Theory* 7(5–6):439–461
29. Rokseth B, Skjong S, Pedersen E (2016) Modeling of generic offshore vessel in crane operations with focus on strong rigid body connections. *IEEE J Ocean Eng PP(99)*: 1–23
30. Skjong S, Pedersen E (2016) The theory of bond graphs in distributed systems and simulations. In: 2016 International Conference on Bond Graph Modeling and Simulation (ICBGM 2016). Montreal, Canada: Society for Modeling and Simulation International
31. Skjong S, Pedersen E (2017) A real-time simulator framework for marine power plants with weak power grids. *Elsevier Mechatron* 47:24–36
32. Dahlquist G (1956) Convergence and stability in the numerical integration of ordinary differential equations. *Math Scand* 4:33–53
33. Schweizer B, Li P, Lu D (2015) Explicit and implicit cosimulation methods: stability and convergence analysis for different solver coupling approaches. *J Comput Nonlinear Dyn* 10(5):51007
34. Sadjina S, Kyllingstad LT, Skjong S, Pedersen E (2016) Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation. *Eng Comput*
35. Yum KK, Skjong S, Tasker B, Pedersen E, Steen S (2016) Simulation of a hybrid marine propulsion system in waves. Helsinki: CIMAC
36. Khalil HK (2002) Nonlinear systems. Prentice Hall PTR
37. Skjong S, Pedersen E (2017) Non-angular MPC-based thrust allocation algorithm for marine vessels—a study of optimal thruster commands. *IEEE Trans Transp Electr PP(99)*: 1–17
38. Skjong S, Pedersen E (2017) Co-simulation of a marine offshore vessel in dp-operations including hardware-in-the-loop (HIL). In: ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2017), Trondheim, Norway
39. Arduino. <https://www.arduino.cc/>. [Online]. <https://www.arduino.org/>
40. Fossen TI, Strand JP (1999) Passive nonlinear observer design for ships using lyapunov methods: full-scale experiments with a supply vessel. *Automatica* 35(1):3–16
41. Belanger J, Venne P, Paquin J-N (2010) The what, where and why of real-time simulation. *Planet RT* 1:37–49
42. Ross A, Hassani V, Ø, Ringen E, Fathi D (2015) Identification of nonlinear manoeuvring models for marine vessels using planar motion mechanism tests. In: 34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2015), St. John's, Newfoundland, Canada, 2015
43. Hassani V, Ross A, Selvik Ø, Fathi D, Sprenger F, Berg T (2015) Time domain simulation model for research vessel Gunnerus. 34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2015). St. John's, Newfoundland, Canada, p 2015
44. Sintef Ocean. <https://www.sintef.no/en/software/vesim/>
45. Selvik Ø, Berg TE, Gavrilin S (2014) Sea trials for validation of shiphandling simulation models—a case study. In: Maritime-Port Technology and Development. CRC Press, pp 141–146